

# **Theoretische Informatik III (T3INF2002)**

Formale Sprachen und Automaten | Einführung Compilerbau

Vorlesung im Wintersemester 2022/23

# Formale Sprachen und Automaten

- Grammatiken
- Chomsky-Hierarchie

# Grammatiken

# Grammatiken

Eine Grammatik wird spezifiziert durch 4 Angaben:

$$G = (V, \Sigma, P, S)$$

$V$  = endliche Menge der **Variablen**  
 $\Sigma$  = endliche Menge der **Terminalzeichen**  
 $S \in V$  = die **Startvariable**  
 $P$  = endliche Menge der **Regeln**  
(oder Produktionen)

auch üblich:  $(V, A, P, S)$ ,  $(N, T, P, S)$ ,  $(S_N, S_T, P, w_S)$

- Es gilt:  $V \cap \Sigma = \emptyset$
- Regeln „Produktionsregeln“ haben die Form: linke Seite  $\rightarrow$  rechte Seite
- linke und rechte Seite können aus Variablen  
(= Nichtterminalzeichen) und Terminalzeichen zusammengesetzt sein

# Grammatiken

Beispiel:

$$V = \{S\}$$

$$\Sigma = \{a, b\}$$

Regeln

1)  $S$

2)  $S \rightarrow ab$

lies: „S erzeugt aSb“  
oder „aus S folgt aSb“


$$S \Rightarrow ab$$


$$S \Rightarrow aSb \Rightarrow aabb = a^2b^2$$

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb = a^3b^3$$

d. h. Bei dieser Grammatik sind ableitbar alle Wörter der Form  $a^n b^n$ ,  $n \geq 1$

# Definition Grammatiken

- Eine endliche, nicht-leere Menge von Terminalzeichen nennt man auch *Alphabet*.
- Elemente eines Alphabets heißen *Symbole*.
- Falls  $\Sigma$  ein Alphabet ist, so bezeichnet  $\Sigma^*$  die Menge aller *Worte* (*aus endlichen Folgen*) bestehend aus Buchstaben  $\in \Sigma$ .



z. B.:  $\Sigma^* = \{ \varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, \dots \}$   
 $\Sigma^+ = \Sigma^* \setminus \{ \varepsilon \}$

→  $\Sigma^+$  Bezeichnet eine nicht-leere Menge

# Beispiele: Alphabete

$A = \{a, b, \dots, z\}; \quad B = \{0, 1\}$

$C = \{ \text{for, end, begin, if, then, else, ...} \}$

Durch Hintereinanderschalten entstehen Wörter (endliche Folgen)

$abbcad \in A^*; \quad 01101 \in B^*; \quad \text{begin if end} \in C^*;$

# Grammatiken

**Die Länge eines Wortes ist die Anzahl seiner Buchstaben:**

$abbcda \in \Sigma^*$  , mit  $|abbcda| = 6$

$|\varepsilon|=0$

Es gilt:

$$|w_1 w_2| = |w_1| + |w_2|$$

— Für  $w \in \Sigma^*$  bezeichnet  $|w|$  die Länge von  $w$ .

— Sei  $w \in \Sigma^*$  ein Wort,  $n \in \mathbb{N}$

Dann ist  $w^n \underbrace{\quad}_{n\text{-mal}} \dots w$  ein Wort der Länge  $|w^n| = n \cdot |w|$

Sei  $\Sigma$  ein Alphabet, dann heißt  $L \subseteq \Sigma^*$  eine (formale) Sprache



# Grammatiken

## Problem:

- Sprachen enthalten i. a. unendlich viele Wörter

## Ziel:

- Endlichen Formalismus angeben, der in der Lage ist, unendlich viele Sprachen zu bezeichnen.

## Beispiel:

- Grammatik aus vorangegangenem Beispiel war kontextfrei, d. h. auf der linken Seite der Regeln steht nur eine Variable.

# Grammatiken

- Beispiel für eine nicht-kontextfreie (kontextsensitive) Grammatik

**$V = \{ S, B \}$**

**$\Sigma = \{ a, b, c \}$**

**S: Startvariable**

*Regeln:*

1)  $S \rightarrow aSBc$

2)  $S \rightarrow abc$

3)  $cB \rightarrow Bc$

4)  $bB \rightarrow bb$

- Eine mögliche Ableitung eines Wortes  $\in \Sigma^*$

$S \Rightarrow aSBc \Rightarrow aabcBc \Rightarrow aabBcc \Rightarrow aabbcc = a^2b^2c^2$

1)

2)

3)

4)  $\in \Sigma^*$

- Bei dieser Grammatik sind ableitbar: alle Wörter der Form  $a^n b^n c^n$ ,  $n \geq 1$

# Beispiel-Übung n=3

$V = \{ S, B \}$

$\Sigma = \{ a, b, c \}$

S: Startvariable

Regeln:

1)  $S \rightarrow aSBc$

2)  $S \rightarrow abc$

3)  $cB \rightarrow Bc$

4)  $bB \rightarrow bb$

# Beispiel-Übung n=4

$V = \{ S, B \}$

$\Sigma = \{ a, b, c \}$

S: Startvariable

Regeln:

1)  $S \rightarrow aSBc$

2)  $S \rightarrow abc$

3)  $cB \rightarrow Bc$

4)  $bB \rightarrow bb$

# Grammatiken–Definition

Die von einer Grammatik  $G = (V, \Sigma, P, S)$  erzeugte (definierte) Sprache ist

$$L(G) := \{w \in \Sigma^* \mid S \Rightarrow \dots \Rightarrow w\}$$

$w_1 \Rightarrow w_2$  ist eine Ableitung:

- Wort  $w_1$  enthält die linke Seite einer Grammatik-Regel.
- Diese linke Seite wurde in  $w_2$  durch die rechte Seite ersetzt.
- Eine Ableitung endet, wenn  $w$  nur noch Terminalsymbole enthält.

# Grammatiken-Sequenz

$S \Rightarrow \dots \Rightarrow w$  bedeutet, dass es eine Ableitung (eine endliche Folge) von Regelanwendungen gibt, die von  $S$  auf  $w$  führt.

Diese Folge ist nicht zwingend; es kann passieren, dass bestimmte ( $\rightarrow$  "schlechte") Folgen zu keiner Sequenz aus  $\Sigma^*$  führen.

# Grammatiken-Sequenz Beispiel

$V = \{S, B\}, \Sigma = \{a, b, c\}$

$S \rightarrow aSBc$

$S \rightarrow abc$

$cB \rightarrow Bc$

$bB \rightarrow bb$

$aS \rightarrow aB$

$S \Rightarrow aSBc \Rightarrow aBBc$  [ende]

keine Sequenz nur aus  
Terminalsymbolen entstanden

# Chomsky-Hierarchie

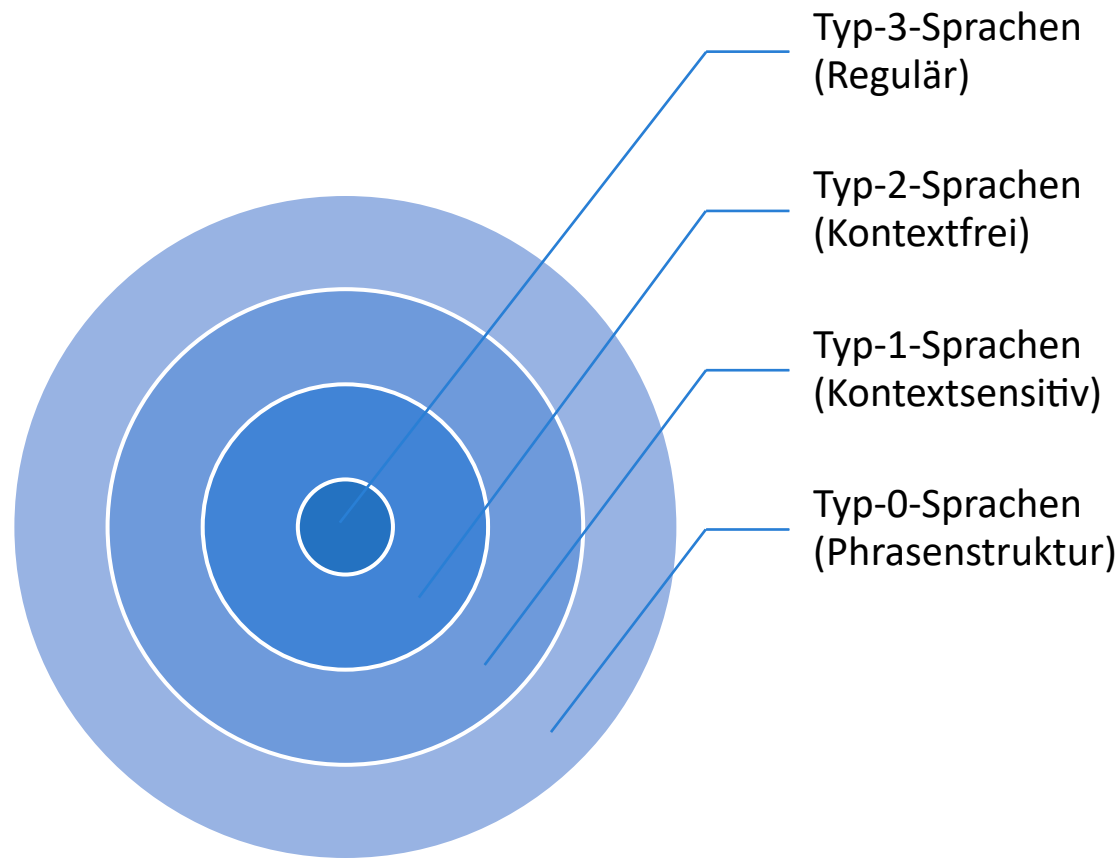


# Chomsky-Hierarchie

- Formale Grammatiken sind ein mächtiges Werkzeug für die Erzeugung unterschiedlichster Sprachen.
- Spanne reicht von einelementigen Wortmengen bis hin zu komplexen Sprachgebilden, die dem gesprochenen Wort gleichen können.
- Struktur der Produktionen einer Grammatik  $G$  hat dabei einen großen Einfluss auf die Eigenschaften der erzeugten Sprache  $L(G)$ .

-> 1957 veröffentlichte der amerikanische Sprachwissenschaftler Noam Chomsky ein Regelwerk, mit dessen Hilfe formale Grammatiken in vier Klassen eingeteilt werden können.

# Chomsky-Hierarchie



Chomsky-Hierarchie teilt die Menge der formalen Sprachen in vier Typklassen ein.

Eine Sprache  $L$  ist eine Typ- $n$ -Sprache, wenn eine Typ- $n$ -Grammatik existiert, die  $L$  erzeugt.

Zwischen den Sprachklassen besteht eine echte Inklusionsbeziehung, d. h., für alle  $n$  mit  $0 \leq n < 3$  gilt  $L_n \supset L_{n+1}$  und  $L_n \neq L_{n+1}$ .

# Chomsky-Hierarchie Typ-0

## Phrasenstrukturgrammatiken(Typ-0-Grammatiken)

Jede Grammatik ist per Definition immer auch eine Typ-0-Grammatik. Insbesondere unterliegt die Struktur der Produktionen keinen weiteren Einschränkungen.

$L_{C0} := \{\omega \mid \omega \text{ codiert eine terminierende Turing-Maschine}\}$  ist eine Typ-0-Sprache, aber keine Typ-1-Sprache.

# Chomsky-Hierarchie Typ-1

## Kontextsensitive Grammatiken(Typ-1-Grammatiken)

Eine Grammatik heißt *kontextsensitiv*, wenn jede Produktionsregel  $l \rightarrow r$  entweder die Beziehung  $|r| \geq |l|$  erfüllt oder die Form  $S \rightarrow \varepsilon$  aufweist. Ist die Regel  $S \rightarrow \varepsilon$  enthalten, so darf  $S$  in keiner anderen rechten Seite einer Regel vorkommen.

$L_{C1} := \{anbncn \mid n \in \mathbb{N}^+\}$  ist eine Typ-1-Sprache, aber keine Typ-2-Sprache.

# Chomsky-Hierarchie Typ-2

## Kontextfreie Grammatiken (Typ-2-Grammatiken)

Typ-2-Grammatiken sind dadurch charakterisiert, dass die linke Seite einer Produktionsregel ausschließlich aus einer einzigen Variablen besteht. Für alle Produktionen  $l \rightarrow r$  gilt also  $l \in V$ .

$L_{C2} := \{anbn \mid n \in \mathbb{N}^+\}$  ist eine Typ-2-Sprache, aber keine Typ-3-Sprache.

# Chomsky-Hierarchie Typ-3

## Reguläre Grammatiken (Typ-3-Grammatiken)

Reguläre Grammatiken sind kontextfrei und besitzen die zusätzliche Eigenschaft, dass die rechte Seite einer Produktion entweder aus dem leeren Wort  $\varepsilon$  oder einem Terminalsymbol, gefolgt von einem Nonterminal, besteht. Formal gesprochen besitzt jede Produktion die Form  $l \rightarrow r$  mit  $l \in V$  und  $r \in \{\varepsilon\} \cup \Sigma V$ .

$L_{C3} := \{ (ab)^n \mid n \in \mathbb{N}^+ \}$  ist eine Typ-3-Sprache.

# Beispiel für reguläre Sprache (Typ 3)

$V = \{ S, A, B \}$

$\Sigma = \{ a, b \}$

$P = \{$   
1.  $S \rightarrow bS,$   
2.  $S \rightarrow aA,$   
3.  $A \rightarrow bS,$   
4.  $A \rightarrow aB,$   
5.  $A \rightarrow a,$   
6.  $B \rightarrow bB,$   
7.  $B \rightarrow aB,$   
8.  $B \rightarrow b,$   
9.  $B \rightarrow a \}$

Die Grammatik ist vom Typ 3, da...

...auf der linken Seite bei jeder Regel nur eine Variable steht.

... auf der rechten Seite nur ein einziges Terminalsymbol (Regeln 5, 8, 9) oder ein Terminalsymbol gefolgt von einer Variablen (Regeln 1, 2, 3, 4, 6, 7) steht.

# Chomsky Hierarchie in der Programmierung

- Programmiersprachen liegen zwischen Typ-2 (kontextfrei) und Typ-1 Grammatiken (kontextsensitiv).
- Das hat zur Folge, dass die „meisten“ Anweisungen formale Wörter einer kontextfreien Sprache sind, aber „einige“ Anweisungen sind formale Wörter einer darüber hinaus gehenden Sprache.