Automatentheorie und Formale Sprachen

– LV 4110 –

Grammatiken und Formale Sprachen

- Kennenlernen der Begriffe: Formale Sprachen und Ableitung
- Definition eines allgemeinen Erzeugungs- bzw. Ableitungssystems
- Klärung, was man unter der Sprache einer Grammatik versteht
- Definition und Einteilung der Chomsky-Grammatik (Chomsky-Hierarchie bzw. -Typen 0 bis 3)
- Festlegung, was das allgemeine Wortproblem ausdrückt
- Konstruktion einer Grammatik aus gegebenem Automaten und umgekehrt
- Kennenlernen des Vorgehens bei der Erstellung von Ableitungsbäumen

III. Grammatiken und Formale Sprachen

1. Semi-Thue-Systeme
   - 1.1 Idee und Definition des Systems
   - 1.2 Ableitung und Überführung
2. Chomsky-Grammatiken
   - 2.1 Idee und Definition
   - 2.2 Sprache einer Chomsky-Grammatik
3. Chomsky-Hierarchie
4. Endliche Automaten und RL-Grammatiken
   - 4.1 Konstruktion einer Grammatik aus einem Automaten
   - 4.2 Konstruktion eines Automaten aus einer Grammatik

<u>bisher</u>:

Sprachen mit Hilfe von **Automaten** identifiziert und analysiert

<u>jetzt</u>:

Kennenlernen von Formalismen zur **Erzeugung** von Sprachen

→ Erzeugungssysteme

Solche Erzeugungssysteme, die auf sog. **REGELN**, **PRODUKTIONEN** oder **GRAMMATIKEN** basieren, sind Untersuchungsgegenstand der <u>Theorie der Formalen Sprachen</u>.

III. Grammatiken und Formale Sprachen

1. Semi-Thue-Systeme

1.1 Idee und Definition des Systems

1.2 Ableitung und Überführung

2. Chomsky-Grammatiken

2.1 Idee und Definition

2.2 Sprache einer Chomsky-Grammatik

3. Chomsky-Hierarchie

4. Endliche Automaten und RL-Grammatiken

4.1 Konstruktion einer Grammatik aus einem Automaten

4.2 Konstruktion eines Automaten aus einer Grammatik

Idee:

- Sprache nicht rein als eine Menge von Wörtern ansehen, sondern definieren, wie man Wörter verändern und manipulieren kann.
- Diese Manipulationen müssen kontrollierbar und nachvollziehbar sein und deshalb nach festen Regeln ablaufen.

Definition:

Ein **Semi-Thue-System** (Norwegischer Mathematiker A. Thue, 1914) wird durch eine endliche Teilmenge **P** $\subseteq \Sigma^* \times \Sigma^*$ bestimmt. Jedes Wortpaar $(\alpha, \beta) \in$ **P** ist eine **Regel** in dem Sinne, dass in einem vorhandenen Ausgangswort **w** ein Teilwort α durch β ersetzt werden kann. Durch die einseitige Ersetzungsrichtung $\alpha \to \beta$ erklärt sich die Bezeichnung "Semi" .

III. Grammatiken und Formale Sprachen

Definition:

Ein Wort **w** heißt aus einem Wort **v** ***ableitbar***, wenn es durch endlich viele **Ersetzungsschritte** aus **v** entsteht.

Notation:

$v \Rightarrow^* w: \quad v(\alpha 1 \to \beta 1) \Rightarrow v1(\alpha 2 \to \beta 2) \Rightarrow \ldots \Rightarrow v_{n-1}(\alpha n \to \beta n) \Rightarrow v_n = w$

kurz:

$$\mathbf{v \Rightarrow^* w} \quad := \quad$$

w aus **v** **ableitbar** oder

v in **w** **überführbar**

Ableitung

Beispiel:

$\Sigma = \{a, b, c, d, e\}$ $v = \{abc\}$ $w = \{aeb\}$

Regeln des Semi-Thue-Systems **R**:

(1) $ab \rightarrow ad$

(2) $dc \rightarrow ee$

(3) $e \rightarrow b$

(4) $ad \rightarrow ae$

(5) $eb \rightarrow b$

(6) $abc \rightarrow e$

Ableitung: $v \Rightarrow^* w$

$v = \underline{abc}$ $(1)\Rightarrow^*$ $a\underline{dc}$ $(2)\Rightarrow^*$ $ae\underline{e}$ $(3)\Rightarrow^*$ $aeb = w$ q.e.d.

III. Grammatiken und Formale Sprachen

Amerikanischer Sprachwissenschaftler **Noam Chomsky**, 50. Jahren

Idee:

- Das Alphabet Σ besteht aus zwei disjunkten Teilmengen **T** und **N**.
- **T** steht für die Menge der Terminalsymbole, d. h. Menge atomarer Zeichen, aus denen letztlich alle Wörter einer Sprache aufgebaut sind.
  Bsp.: Zeichen a, b, ..., z und 0, 1, ..., 9
- **N** steht für die Menge der **N**onterminalsymbole, die zur Abstraktion und Klassenbildung von Wörtern dienen; sog. syntaktische Kategorien
  Bsp.: Begriffe wie <Name>, <Buchstabe>, <Ziffern>

Ableitung

Definition

<u>Beispiel</u>: „Namensgebung in PASCAL“
hier: erweiterte Backus-Naur-Form (BNF)

<Name>	::=	<Buchstabe> { <Buchstabe> | <Ziffer> }
<Buchstabe>	::=	a | b | c | ... | z
<Ziffer>	::=	0 | 1 | 2 | ... | 9

::=	Überführung oder Ersetzung
|	Auswahl unter mehreren Alternativen
{ ... }	beliebige Wiederholung

Ergebnis: Ein korrekter Name muss mit einem Buchstaben beginnen, anschließend können weitere Buchstaben und Ziffern in beliebiger Zahl folgen.

Definition:

Unter einer ***Chomsky-Grammatik*** verstehen wir ein **Quadrupel** G = (**N**, **T**, **P**, S), wobei **P** ein **Semi-Thue-System** über dem Alphabet $\Sigma^* = (\mathbf{N} \cup \mathbf{T})^*$ ist. Die einzelnen Komponenten haben folgende Bedeutung:

- **N** Menge der Nonterminalsymbole A,B,C, ...
- **T** Menge der Terminalsymbole a,b,c, ...
- **P** Menge der Produktionen (Regeln):
  $\mathbf{P} \subset \{ \varphi \rightarrow \psi \mid \varphi, \psi \in (\mathbf{N} \cup \mathbf{T})^*, \varphi \neq \varepsilon \}$
- S Startsymbol $\in$ **N**, das in mindestens einer Regel links vorkommt

Interpretation:

- Man hat endliche Menge von Produktionen **P** ($\rightarrow$ auch Regeln genannt), die Terminal- und Nonterminalsymbole beinhalten können.
- Erzeugung von Wörtern beginnt immer mit dem Startsymbol S.
- Linke Seite einer Regel darf nicht das leere Wort ε sein.
- Die erzeugenden Wörter bestehen letztendlich nur aus Terminalzeichen T.

III. Grammatiken und Formale Sprachen

Definition:

Die Sprache einer Chomsky-Grammatik besteht aus allen Worten aus **T***, die aus S ableitbar sind, d. h. für die gilt:

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

heißt:

$$S \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \dots \Rightarrow u_n = w$$

mit $n \in \mathbb{N}_0$

$u_i \in (\mathbf{N} \cup \mathbf{T})^*$

$w \in \mathbf{T}^*$

kurz: $S \Rightarrow^* w$

Ableitung

Beispiel: „Namensgebung in PASCAL“

Mit

N = { <Name>, <Buchstabe>, <Ziffer> }

T = { a, b, c, ..., z, 0, 1, 2, …, 9 }

P = { <Name> → <Name> <Ziffer>, <Name> → <Buchstabe>,
<Buchstabe> → a | b | c | ... | z, <Ziffer> → 0 | 1 | 2 | ... | 9 }

S = <Name>

gehört das Wort „**a12**“ zur Sprache L(G) obiger Grammatik G = (**N**, **T**, **P**, S), weil sich folgende Ableitung angeben lässt:

<Name> → <Name> <Ziffer> → <Name> <Ziffer> <Ziffer>
→ <Buchstabe> <Ziffer> <Ziffer> → **a12**

Grundsätzliche Bemerkung:

Die bisherige Definition einer Chomsky-Grammatik ist so allgemein, dass nicht entscheidbar ist, ob ein Wort zur Sprache einer vorgegebenen Grammatik gehört oder nicht (vgl. allgemeines Wortproblem).

Deshalb ist es sinnvoll, Einschränkungen zu treffen, so dass dieses Problem lösbar wird.

$\Rightarrow$ vernünftige Hierarchie von Einschränkungen!

Im folgenden wollen wir uns einer entsprechenden Einteilung (Typ 0 bis Typ 3) zuwenden.

<u>Definition</u>:

Eine Chomsky-Grammatik G = (**N**, **T**, **P**, S) heißt

- **Typ 0** oder **allgemeine** Chomsky-Grammatik, wenn alle Regeln die nicht eingeschränkte Form

  $\alpha \rightarrow \beta$ mit $\alpha \neq \varepsilon$ ε = leeres Element

  $\alpha, \beta \in (\mathbf{N} \cup \mathbf{T})^*$

  haben (→ Ursprungs-Definition!).

Definition:

Eine Chomsky-Grammatik G = (**N**, **T**, **P**, S) heißt

- **Typ 1** oder **kontextsensitive** Grammatik, wenn alle Regeln die Form

$$\alpha A \beta \rightarrow \alpha \gamma \beta \quad \text{mit} \quad \gamma \neq \varepsilon$$

$$A \in \mathbf{N}\,; \qquad \alpha, \beta, \gamma \in (\mathbf{N} \cup \mathbf{T})^*$$

haben mit der Ausnahme, dass S $\rightarrow \varepsilon$ dazugehören darf, wenn S in keiner Regel auf der rechten Seite auftritt.

Definition:

Eine Chomsky-Grammatik G = (**N**, **T**, **P**, S) heißt

- **Typ 2** oder **kontextfreie** Grammatik, wenn alle Regeln die Form

  $$\mathbf{A \to \gamma} \quad \text{mit} \quad \mathbf{A \in N\,;} \quad \mathbf{\gamma \in (N \cup T)^*}$$

  $$\text{oder} \quad \mathbf{\gamma = \varepsilon}$$

  haben.

Definition:

Eine Chomsky-Grammatik G = (**N**, **T**, **P**, S) heißt

- **Typ 3** oder **rechtslineare***) Grammatik, wenn alle Regeln die Form

$$A \to \varepsilon \;;\; A \to a \quad \text{oder} \quad A \to aB \;;$$

mit $a \in \mathbf{T}$; $A, B \in \mathbf{N}$

haben.

*) d. h. nur ein **N**onterminalsymbol ganz rechts!

Anmerkung:

Jede Typ-3-Grammatik ist auch vom Typ 2, und jede Typ-1-Grammatik auch vom Typ 0.

Definition:

Zwei Grammatiken heißen ***äquivalent***, wenn sie die gleiche Sprache erzeugen.

Definition:

Eine Sprache L(G) heißt vom ***Chomsky-Typ i*** (i=0,1,2,3) wenn G vom Typ i ist. Die Familie der Sprachen vom Typ i wird mit $\mathbf{L}_i$ bezeichnet.

Satz:

Es gelten die Beziehungen: $\mathbf{L}_3 \subset \mathbf{L}_2 \subset \mathbf{L}_1 \subset \mathbf{L}_0$ und

$$\mathbf{L}_3 = \{ L(A) \mid A \text{ ist endlicher Automat} \}$$

III. Grammatiken und Formale Sprachen

1. Semi-Thue-Systeme
   - 1.1 Idee und Definition des Systems
   - 1.2 Ableitung und Überführung
2. Chomsky-Grammatiken
   - 2.1 Idee und Definition
   - 2.2 Sprache einer Chomsky-Grammatik
3. Chomsky-Hierarchie
4. **Endliche Automaten und RL-Grammatiken**
   - 4.1 Konstruktion einer Grammatik aus einem Automaten
   - 4.2 Konstruktion eines Automaten aus einer Grammatik

Rechts- und linkslineare Grammatiken

Die Regeln der rechts- (oder links-) linearen Grammatiken sind innerhalb der Chomsky-Hierarchie am stärksten eingeschränkt.

rechtslinear: **$A \rightarrow aB$**

Nonterminal-zeichen	nur ersetzbar durch	Terminal-zeichen	+ ggf.	Nonterminal-zeichen
A		**a**		**B**

Bei Rechtslinearität erfolgt Einsetzung immer **rechts**, d. h. am Ende des bereits abgeleiteten Wortes.

Da immer nur ein zusätzliches Terminalzeichen hinzugefügt wird, wächst das entstehende Wort in einer Richtung **linear** mit der Anzahl der Ersetzungsschritte.

Satz:

Zu jeder **rechtslinearen** Grammatik gibt es eine **linkslineare**, die die gleiche Sprache erzeugt.

Satz:

Zu jeder rechtslinearen Grammatik mit der Sprache L(G) gibt es einen endlichen Automaten mit

$$L(A) = L(G)$$

und umgekehrt.

III. Grammatiken und Formale Sprachen

1. Semi-Thue-Systeme
   - 1.1 Idee und Definition des Systems
   - 1.2 Ableitung und Überführung
2. Chomsky-Grammatiken
   - 2.1 Idee und Definition
   - 2.2 Sprache einer Chomsky-Grammatik
3. Chomsky-Hierarchie
4. Endliche Automaten und RL-Grammatiken
   - **4.1 Konstruktion einer Grammatik aus einem Automaten**
   - 4.2 Konstruktion eines Automaten aus einer Grammatik

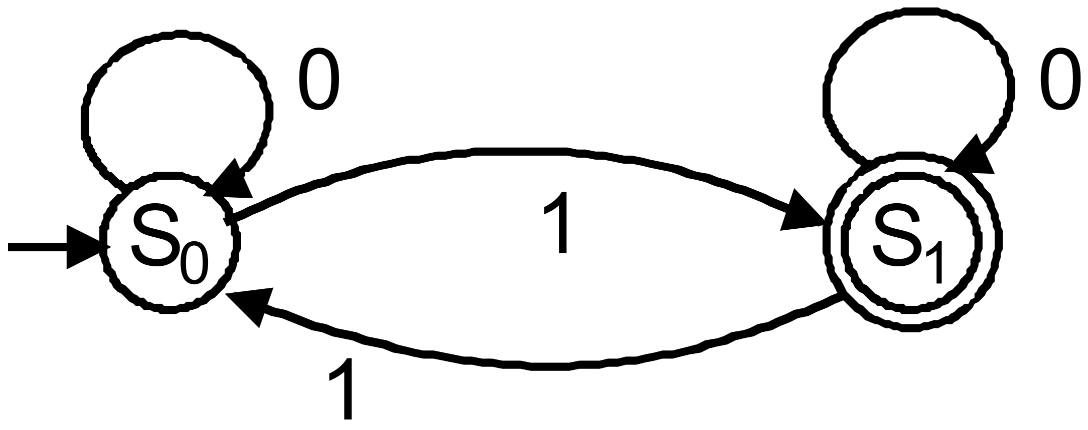
Konstruktion einer RL G aus einem A = (**S**, s_0, **F**, Σ, δ):

- verwende die Eingabezeichen Σ als Terminalzeichen **T**
- verwende die Menge **S** der Zustände als Nonterminal-zeichen **N**
- verwende s_0 als Startsymbol S
- ersetze jeden Funktionswert $\boldsymbol{\delta(s_1,a) = s_2}$ durch die Regel $\boldsymbol{s_1 \rightarrow a\, s_2}$
- erzeuge für jeden Endzustand s_f eine zusätzliche Regel $\boldsymbol{s_f \rightarrow \varepsilon}$

Ungerade Anzahl von Einsen:

Wir betrachten folgenden DFA, der alle Worte akzeptiert, die eine ungerade Anzahl von Einsen enthalten.

DFA $\mathbf{A} = (\Sigma, \mathbf{S}, S_0, \delta, \mathbf{F})$ mit $\Sigma = \{0, 1\}$, $\mathbf{S} = \{S_0, S_1\}$, $\mathbf{F} = \{S_1\}$ und δ gemäß folgendem Zustandsdiagramm:



Ungerade Anzahl von Einsen:

Hierauf folgt unter Anwendung der Regeln die Grammatik
G = (**N**, **T**, **P**, S) mit

- Eingabezeichen Σ als Terminalzeichen **T** → $\mathbf{T} = \Sigma = \{0, 1\}$
- Zustandsmenge **S** als Nonterminalzeichen **N** → $\mathbf{N} = \{S_0, S_1\}$
- Anfangszustand S_0 als Startsymbol S → $S = S_0$
- Jeder Funktionswert $\boldsymbol{\delta(s_1,a) = s_2}$ eine Regel der Form $\boldsymbol{s_1 \to a\ s_2}$ →

  $\mathbf{P} = \{ S_0 \to 1\ S_1 ,\ S_0 \to 0\ S_0 ,\ S_1 \to 1\ S_0 ,\ S_1 \to 0\ S_1 \}$
- Jeden Endzustand s_f eine zusätzliche Regel $\boldsymbol{s_f \to \varepsilon}$ → $S_1 \to \varepsilon$

⇒ **P** und δ sind **ähnlich ausdrucksstarke** Konzepte!

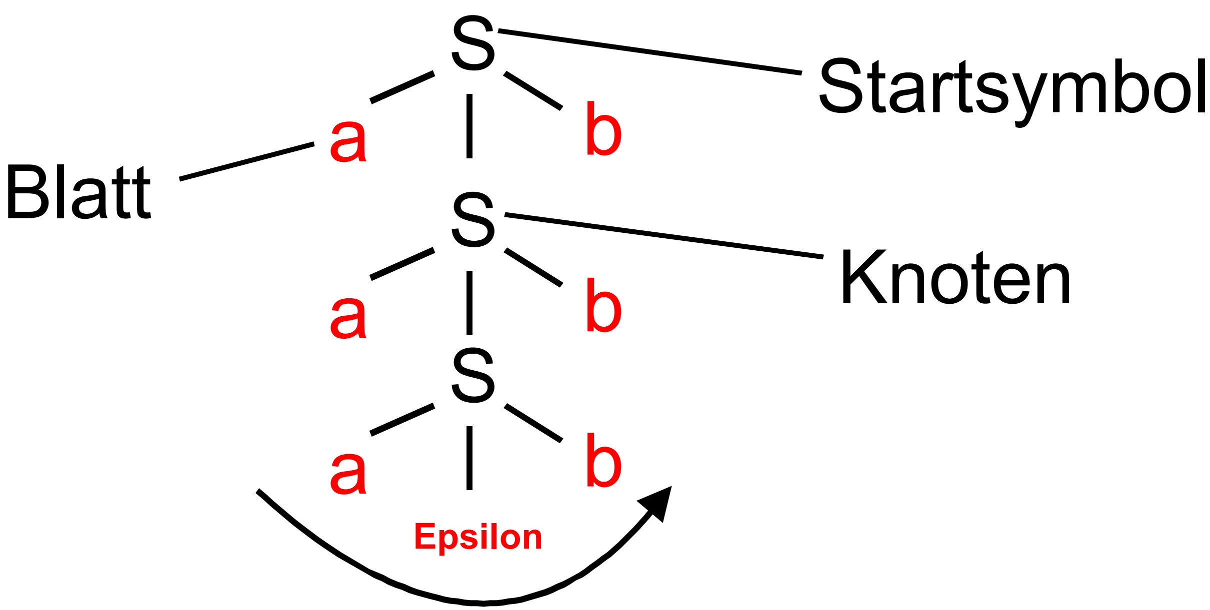
III. Grammatiken und Formale Sprachen

1. Semi-Thue-Systeme
   - 1.1 Idee und Definition des Systems
   - 1.2 Ableitung und Überführung
2. Chomsky-Grammatiken
   - 2.1 Idee und Definition
   - 2.2 Sprache einer Chomsky-Grammatik
3. Chomsky-Hierarchie
4. Endliche Automaten und RL-Grammatiken
   - 4.1 Konstruktion einer Grammatik aus einem Automaten
   - **4.2 Konstruktion eines Automaten aus einer Grammatik**

Konstruktion eines A aus einer RL G = (**N**, **T**, **P**, S):

- verwende die Terminalzeichen **T** als Eingabezeichen Σ
- verwende die Nonterminalzeichen **N** als Zustände **S**
- verwende das Startsymbol S als Ausgangszustand s_0
- verwende jedes A, für das $\mathbf{A} \rightarrow \varepsilon$ gilt, als Endzustand s_f
- ersetze jede Produktion $\mathbf{A} \rightarrow \mathbf{aB}$ durch einen Funktionswert $\boldsymbol{\delta}\mathbf{(A,a) = B}$
- führe für ***alle*** Produktionen $\mathbf{A} \rightarrow \mathbf{a}$ *einen* Endzustand s_f ein und
- bilde für ***jede*** solche Produktion den Funktionswert $\delta(A,a) = s_f$

Ableitungsbaum des Wortes **aaabbb**:



Blätter von links nach rechts gelesen ergeben das Wort w.