Abschnitt 5

Grammatiken

Definition Formale Grammatik

Eine **formale Grammatik** G ist eine 4-Tupel

$$G = (N, T, P, S)$$

mit

- einem Alphabet von **Nicht-Terminalsymbolen** N
- einem Alphabet von **Terminalsymbolen** T, wobei $V = N \cup T$ das **Gesamtalphabet** bezeichnet und $N \cap T = \emptyset$
- einer endlichen Mege von **Produktionen** P der Form

  $\alpha \longrightarrow \gamma$ (gesprochen “Alpha produziert Gamma”) mit
  $\alpha \in V^*N\ V^*$ und
  $\gamma \in V^*$

- einem **Startsymbol** $S \in N$

Beispiel einer Formalen Grammatik

$$
\begin{aligned}
G_1 &= (N, T, P, S) \text{ mit} \\
N &= \{S\} \\
T &= \{a, b\} \\
P &= \{S \rightarrow Sa,\ S \rightarrow b\} \\
S &= S
\end{aligned}
$$

Erzeugt z.B. *b, ba, baa, ...* also $ba^n, n \geq 0$

Ableiten Formaler Sprachen (1)

- Grammatiken legen “Regeln” fest um aus dem jeweiligen Startsymbol Wörter aus Terminalsymbolen zu bilden.
- Dieser Erzeugungsprozess wird durch die Relation $\Rightarrow \subseteq V^* \times V^*$ repräsentiert (gesprochen “ist direkt ableitbar nach”)

Seien $u, v, \gamma \in V^*$ und sei $\alpha \in V^*NV^*$, dann gilt

$$u\alpha v \Rightarrow u\gamma v \quad \text{genau dann wenn}$$
$$(\alpha \longrightarrow \gamma) \in P$$

Das heißt, die Ableitung von Wörtern über V^* richtet sich nach den Produktionen der jeweiligen Grammatik.

Ableiten Formaler Sprachen (2)

Wie bei allen Relationen gilt, dass

- $\Rightarrow^n$ die *n*-fache **Potenz**,
- $\Rightarrow^+$ die **transitive Hülle** und
- $\Rightarrow^*$ die **reflexive und transitive Hülle**

der Relation $\Rightarrow$ bezeichnet.

Ableiten Formaler Sprachen (3)

Sei $G = (N, T, P, S)$ eine formale Grammatik. Die von G erzeugte Sprache $L(G)$ ist definiert als

$$L(G) := \{\omega \in T^* | S \Rightarrow^* \omega\}$$

Beispiel

$$
\begin{aligned}
G_1 &= (N, T, P, S) \text{ mit} \\
N &= \{S\} \\
T &= \{a, b\} \\
P &= \{S \rightarrow Sa,\ S \rightarrow b\} \\
S &= S
\end{aligned}
$$

- dann $S \Rightarrow Sa$, $Sa \Rightarrow Saa$, ...
- $S \Rightarrow^2 Saa$, $S \Rightarrow^3 Saaa$, ...
- $(\Rightarrow^+) = (\Rightarrow^1 \cup \Rightarrow^2 \cup \Rightarrow^3 \cup \ldots)$
- $(\Rightarrow^*) = (\Rightarrow^0 \cup \Rightarrow^+)$
- $L(G_1) = \{ba^n | n \in \mathbb{N}_0\} = b\ a^n, n \geq 0$

Aufgabe

Schreiben Sie eine Grammatik G_a, die beliebige Wörter $a^n, n > 0$ vom Startsymbol S ableitet.
Geben Sie die Ableitungsschritte in der Form "$S \Rightarrow a$ mit $S \longrightarrow a$" an, die nötig sind, um das Wort *aaaa* aus dem Startsymbol S abzuleiten.

Typen Formaler Grammatiken

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

- $\alpha \in V^*NV^*$
- $A \in N$
- $\gamma \in V^*$
- $\gamma_l \in NT^* \cup T^*$
- $\gamma_r \in T^*N \cup T^*$

Dann unterscheidet man folgende Typen formaler Grammatiken, abhängig von den Beschränkungen der Produktionsregeln.

Typ	**Regeln**	**Erläuterung**
0	$\alpha \longrightarrow \gamma$	keine Beschränkung
1	$\alpha \longrightarrow \gamma$ mit $\lvert\alpha\rvert \leq \lvert\gamma\rvert$	keine Verkürzung
2	$A \longrightarrow \gamma$	links steht genau ein NT
3		rechts steht höchstens 1 NT...
linkslinear	$A \longrightarrow \gamma_l$	... entweder ganz links oder ...
rechtslinear	$A \longrightarrow \gamma_r$	... ganz rechts

Beispiele erlaubter Regeln: Typ-0-Grammatik

- $A \longrightarrow a$
- $A \longrightarrow AB$
- $AA \longrightarrow BC$
- $Aa \longrightarrow aBB$
- $AA \longrightarrow aBB$
- $AA \longrightarrow B$
- $A \longrightarrow Ba$
- $A \longrightarrow aB$
- $a \longrightarrow A$

Beispiele erlaubter Regeln: Typ-1-Grammatik (kontextsensitiv)

- $A \longrightarrow a$
- $A \longrightarrow AB$
- $AA \longrightarrow BC$
- $Aa \longrightarrow aBB$
- $AA \longrightarrow aBB$
- $AA \longrightarrow B$
- $A \longrightarrow Ba$
- $A \longrightarrow aB$
- $a \longrightarrow A$

Beispiele erlaubter Regeln: Typ-2-Grammatik (kontextfrei)

- $A \longrightarrow a$
- $A \longrightarrow AB$
- $AA \longrightarrow BC$
- $Aa \longrightarrow aBB$
- $AA \longrightarrow aBB$
- $AA \longrightarrow B$
- $A \longrightarrow Ba$
- $A \longrightarrow aB$
- $a \longrightarrow A$

Beispiele erlaubter Regeln: Typ-3-Grammatik (regulär, linkslinear)

- $A \longrightarrow a$
- $A \longrightarrow AB$
- $AA \longrightarrow BC$
- $Aa \longrightarrow aBB$
- $AA \longrightarrow aBB$
- $AA \longrightarrow B$
- $A \longrightarrow Ba$
- $A \longrightarrow aB$
- $a \longrightarrow A$

Beispiele erlaubter Regeln: Typ-3-Grammatik (regulär, rechtslinear)

- $A \longrightarrow a$
- $A \longrightarrow AB$
- $AA \longrightarrow BC$
- $Aa \longrightarrow aBB$
- $AA \longrightarrow aBB$
- $AA \longrightarrow B$
- $A \longrightarrow Ba$
- $A \longrightarrow aB$
- $a \longrightarrow A$

Zusammenhang von Sprachen, Grammatiken und Automaten

- Zu jedem Typ von Grammatik gibt es eine korrespondieren Klasse von Sprachen (Typ-0-Grammatiken erzeugen Typ-0-Sprachen, usw.)
- Umgekehrt gibt es für jede Typ-n-Sprache ($n \in [0,3]$) eine Typ-n-Grammatik, die genau die Wörter der Sprache produziert **und sonst keine**.
- Genauso gibt es für jeden Grammatik-Typ eine entsprechende Klasse von Automaten.
- **Endliche Automaten** entsprechen regulären (Typ-3-) Grammatiken. D.h. für jede reguläre Grammatik gibt es einen endlichen Automaten, der genau nur die Wörter akzeptiert, die die Grammatik produziert.

Aufgabe

1. Schreiben Sie eine Typ-2-Grammatik die (unter anderem) folgende Wörter erzeugt.
   Anton liebt Eiscreme
   Bruno hasst Regen
   Clara mag Fahrradfahren
2. Schreiben Sie eine Typ-2-Grammatik für arithmetische Ausdrücke, z.B.: '$1+24$', $(2+3)\times 7$.
3. Schreiben Sie eine Typ-3-Grammtik zu (1).