

# Formale Grundlagen der Informatik 1

## Kapitel 5

### Kontextfreie Sprachen

Frank Heitmann  
heitmann@informatik.uni-hamburg.de

15. April 2014

# Motivation

Bisher hatten wir

- **Automaten**, die Wörter **akzeptieren**

Wir werden nun ein neues Modell kennenlernen:

- **Grammatiken**, die Wörter **generieren**

# Motivation

Bisher hatten wir

- **Automaten**, die Wörter **akzeptieren**

Wir werden nun ein neues Modell kennenlernen:

- **Grammatiken**, die Wörter **generieren**

# Informales Beispiel

Ihr kennt vielleicht schon Beispiele bei der Definition von Programmiersprachen! Hier z.B. um mögliche Identifier abzuleiten, die Ziffern und Buchstaben enthalten dürfen, aber mit einem Buchstaben beginnen müssen:

$$\langle \text{identifier} \rangle ::= \langle \text{letter} \rangle \mid \langle \text{identifier} \rangle \langle \text{letter} \rangle \mid \langle \text{identifier} \rangle \langle \text{digit} \rangle$$
$$\langle \text{letter} \rangle ::= A \mid B \mid C$$
$$\langle \text{digit} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4$$

oder kompakter:

$$I \longrightarrow L \mid IL \mid ID$$
$$L \longrightarrow a \mid b \mid c$$
$$D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4$$

# Informales Beispiel

Ihr kennt vielleicht schon Beispiele bei der Definition von Programmiersprachen! Hier z.B. um mögliche Identifier abzuleiten, die Ziffern und Buchstaben enthalten dürfen, aber mit einem Buchstaben beginnen müssen:

$$\langle \textit{identifier} \rangle ::= \langle \textit{letter} \rangle \mid \langle \textit{identifier} \rangle \langle \textit{letter} \rangle \mid \langle \textit{identifier} \rangle \langle \textit{digit} \rangle$$
$$\langle \textit{letter} \rangle ::= A \mid B \mid C$$
$$\langle \textit{digit} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4$$

oder kompakter:

$$I \longrightarrow L \mid IL \mid ID$$
$$L \longrightarrow a \mid b \mid c$$
$$D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4$$

# Informales Beispiel

Ihr kennt vielleicht schon Beispiele bei der Definition von Programmiersprachen! Hier z.B. um mögliche Identifier abzuleiten, die Ziffern und Buchstaben enthalten dürfen, aber mit einem Buchstaben beginnen müssen:

$$\langle \text{identifier} \rangle ::= \langle \text{letter} \rangle \mid \langle \text{identifier} \rangle \langle \text{letter} \rangle \mid \langle \text{identifier} \rangle \langle \text{digit} \rangle$$
$$\langle \text{letter} \rangle ::= A \mid B \mid C$$
$$\langle \text{digit} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4$$

oder kompakter:

$$I \longrightarrow L \mid IL \mid ID$$
$$L \longrightarrow a \mid b \mid c$$
$$D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4$$

# Informales Beispiel

$$I \longrightarrow L \mid IL \mid ID$$

$$L \longrightarrow a \mid b \mid c$$

$$D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4$$

Eine mögliche Ableitung eines Identifiers (Ziel:  $abc1$ ):

$$I \Rightarrow ID \Rightarrow ILD \Rightarrow ILLD \Rightarrow LLLD$$

# Informales Beispiel

$$I \longrightarrow L \mid IL \mid ID$$

$$L \longrightarrow a \mid b \mid c$$

$$D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4$$

Eine mögliche Ableitung eines Identifiers (Ziel: *abc1*):

$$I \Rightarrow ID \Rightarrow ILD \Rightarrow ILLD \Rightarrow LLLD$$



# Informales Beispiel

$$I \longrightarrow L \mid IL \mid ID$$

$$L \longrightarrow a \mid b \mid c$$

$$D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4$$

Eine mögliche Ableitung eines Identifiers (Ziel:  $abc1$ ):

$$I \Rightarrow ID \Rightarrow ILD \Rightarrow ILLD \Rightarrow LLLD$$

# Informales Beispiel

$$I \longrightarrow L \mid IL \mid ID$$

$$L \longrightarrow a \mid b \mid c$$

$$D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4$$

Eine mögliche Ableitung eines Identifiers (Ziel:  $abc1$ ):

$$I \Rightarrow ID \Rightarrow ILD \Rightarrow ILLD \Rightarrow LLLD$$

# Informales Beispiel

$$I \longrightarrow L \mid IL \mid ID$$

$$L \longrightarrow a \mid b \mid c$$

$$D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4$$

Eine mögliche Ableitung eines Identifiers (Ziel:  $abc1$ ):

$$I \Rightarrow ID \Rightarrow ILD \Rightarrow ILLD \Rightarrow LLLD$$

# Informales Beispiel

$$I \longrightarrow L \mid IL \mid ID$$

$$L \longrightarrow a \mid b \mid c$$

$$D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4$$

Eine mögliche Ableitung eines Identifiers (Ziel:  $abc1$ ):

$$I \Rightarrow ID \Rightarrow ILD \Rightarrow ILLD \Rightarrow LLLD$$

$$\Rightarrow aLLD \Rightarrow abLD \Rightarrow abL1 \Rightarrow abc1$$

# Informales Beispiel

$$I \longrightarrow L \mid IL \mid ID$$

$$L \longrightarrow a \mid b \mid c$$

$$D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4$$

Eine mögliche Ableitung eines Identifiers (Ziel:  $abc1$ ):

$$I \Rightarrow ID \Rightarrow ILD \Rightarrow ILLD \Rightarrow LLLD$$

$$\Rightarrow aLLD \Rightarrow abLD \Rightarrow abL1 \Rightarrow abc1$$

# Informales Beispiel

$$I \longrightarrow L \mid IL \mid ID$$

$$L \longrightarrow a \mid b \mid c$$

$$D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4$$

Eine mögliche Ableitung eines Identifiers (Ziel:  $abc1$ ):

$$I \Rightarrow ID \Rightarrow ILD \Rightarrow ILLD \Rightarrow LLLD$$

$$\Rightarrow aLLD \Rightarrow abLD \Rightarrow abL1 \Rightarrow abc1$$

# Informales Beispiel

$$I \longrightarrow L \mid IL \mid ID$$

$$L \longrightarrow a \mid b \mid c$$

$$D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4$$

Eine mögliche Ableitung eines Identifiers (Ziel:  $abc1$ ):

$$I \Rightarrow ID \Rightarrow ILD \Rightarrow ILLD \Rightarrow LLLD$$

$$\Rightarrow aLLD \Rightarrow abLD \Rightarrow abL1 \Rightarrow abc1$$

# Hin zu Grammatiken

Dies ist genau die Idee bei Grammatiken!

- Die Großbuchstaben, die ersetzt werden können, sind die **Nonterminale**.
- Die Kleinbuchstaben, aus denen sich das abgeleitete Wort zusammensetzt, sind die **Terminale**
- Die  $I \rightarrow IL$  usw. sind die **Regeln** oder **Produktionen**

Wir definieren nun zunächst die Grammatik *sehr allgemein* und schränken diese später ein...



# Hin zu Grammatiken

Dies ist genau die Idee bei Grammatiken!

- Die Großbuchstaben, die ersetzt werden können, sind die **Nonterminale**.
- Die Kleinbuchstaben, aus denen sich das abgeleitete Wort zusammensetzt, sind die **Terminale**
- Die  $I \rightarrow IL$  usw. sind die **Regeln** oder **Produktionen**

Wir definieren nun zunächst die Grammatik *sehr allgemein* und schränken diese später ein...

# Hin zu Grammatiken

Dies ist genau die Idee bei Grammatiken!

- Die Großbuchstaben, die ersetzt werden können, sind die **Nonterminale**.
- Die Kleinbuchstaben, aus denen sich das abgeleitete Wort zusammensetzt, sind die **Terminale**
- Die  $I \rightarrow IL$  usw. sind die **Regeln** oder **Produktionen**

Wir definieren nun zunächst die Grammatik *sehr allgemein* und schränken diese später ein...

# Hin zu Grammatiken

Dies ist genau die Idee bei Grammatiken!

- Die Großbuchstaben, die ersetzt werden können, sind die **Nonterminale**.
- Die Kleinbuchstaben, aus denen sich das abgeleitete Wort zusammensetzt, sind die **Terminale**
- Die  $I \rightarrow IL$  usw. sind die **Regeln** oder **Produktionen**

Wir definieren nun zunächst die Grammatik *sehr allgemein* und schränken diese später ein...

# Grammatiken. Formale Definition

## Definition (Grammatik)

Eine **Grammatik** ist ein Quadrupel  $G = (V_N, V_T, P, S)$  mit

- 1 Dem endlichen Alphabet von **Nonterminalen**  $V_N$ .
- 2 Dem endlichen Alphabet von **Terminalen**  $V_T$  mit  $V_T \cap V_N = \emptyset$ . Das *Gesamtalphabet* wird mit  $V := V_T \cup V_N$  bezeichnet.
- 3 Der endlichen Menge von **Produktionen** (oder *Regeln*)  $P \subseteq (V^* \setminus V_T^*) \times V^*$ .
- 4 Dem **Startsymbol**  $S \in V_N$ .

# Grammatiken. Formale Definition

## Definition (Grammatik)

Eine **Grammatik** ist ein Quadrupel  $G = (V_N, V_T, P, S)$  mit

- 1 Dem endlichen Alphabet von **Nonterminalen**  $V_N$ .
- 2 Dem endlichen Alphabet von **Terminalen**  $V_T$  mit  $V_T \cap V_N = \emptyset$ . Das *Gesamtalphabet* wird mit  $V := V_T \cup V_N$  bezeichnet.
- 3 Der endlichen Menge von **Produktionen** (oder *Regeln*)  $P \subseteq (V^* \setminus V_T^*) \times V^*$ .
- 4 Dem **Startsymbol**  $S \in V_N$ .

# Grammatiken. Formale Definition

## Definition (Grammatik)

Eine **Grammatik** ist ein Quadrupel  $G = (V_N, V_T, P, S)$  mit

- 1 Dem endlichen Alphabet von **Nonterminalen**  $V_N$ .
- 2 Dem endlichen Alphabet von **Terminalen**  $V_T$  mit  $V_T \cap V_N = \emptyset$ . Das *Gesamtalphabet* wird mit  $V := V_T \cup V_N$  bezeichnet.
- 3 Der endlichen Menge von **Produktionen** (oder *Regeln*)  $P \subseteq (V^* \setminus V_T^*) \times V^*$ .
- 4 Dem **Startsymbol**  $S \in V_N$ .

# Grammatiken. Formale Definition

## Definition (Grammatik)

Eine **Grammatik** ist ein Quadrupel  $G = (V_N, V_T, P, S)$  mit

- 1 Dem endlichen Alphabet von **Nonterminalen**  $V_N$ .
- 2 Dem endlichen Alphabet von **Terminalen**  $V_T$  mit  $V_T \cap V_N = \emptyset$ . Das *Gesamtalphabet* wird mit  $V := V_T \cup V_N$  bezeichnet.
- 3 Der endlichen Menge von **Produktionen** (oder *Regeln*)  $P \subseteq (V^* \setminus V_T^*) \times V^*$ .
- 4 Dem **Startsymbol**  $S \in V_N$ .

# Grammatiken. Formale Definition

## Definition (Grammatik)

Eine **Grammatik** ist ein Quadrupel  $G = (V_N, V_T, P, S)$  mit

- 1 Dem endlichen Alphabet von **Nonterminalen**  $V_N$ .
- 2 Dem endlichen Alphabet von **Terminalen**  $V_T$  mit  $V_T \cap V_N = \emptyset$ . Das *Gesamtalphabet* wird mit  $V := V_T \cup V_N$  bezeichnet.
- 3 Der endlichen Menge von **Produktionen** (oder *Regeln*)  $P \subseteq (V^* \setminus V_T^*) \times V^*$ .
- 4 Dem **Startsymbol**  $S \in V_N$ .



# Grammatiken. Anmerkungen

## Bemerkung

- Eine Regel  $(u, v) \in P$  wird meist als  $u \rightarrow v$  notiert.
- Mehreren Regeln  $u \rightarrow v$  und  $u \rightarrow w$  werden als  $u \rightarrow v \mid w$  abgekürzt.
- Auf der linken Seite einer Regel steht stets mindestens ein Nonterminal!

$$P \subseteq (V^* \setminus V_T^*) \times V^*$$

# Grammatiken. Anmerkungen

## Bemerkung

- Eine Regel  $(u, v) \in P$  wird meist als  $u \rightarrow v$  notiert.
- Mehreren Regeln  $u \rightarrow v$  und  $u \rightarrow w$  werden als  $u \rightarrow v \mid w$  abgekürzt.
- Auf der linken Seite einer Regel steht stets mindestens ein Nonterminal!

$$P \subseteq (V^* \setminus V_T^*) \times V^*$$

# Grammatiken. Anmerkungen

## Bemerkung

- Eine Regel  $(u, v) \in P$  wird meist als  $u \rightarrow v$  notiert.
- Mehreren Regeln  $u \rightarrow v$  und  $u \rightarrow w$  werden als  $u \rightarrow v \mid w$  abgekürzt.
- Auf der linken Seite einer Regel steht stets mindestens ein Nonterminal!

$$P \subseteq (V^* \setminus V_T^*) \times V^*$$

# Ableitung

## Definition (Ableitung)

Die **einschrittige Ableitung** eines Wortes  $v$  aus einem Wort  $u$  mittels einer Produktion einer Grammatik  $G$  wird notiert als

$u \xrightarrow{G} v$ . Dabei ist die Relation  $\xrightarrow{G} \subseteq V^* \times V^*$  für alle

$u, v \in V^*$  definiert durch:  $u \xrightarrow{G} v$  gdw.

$$\exists u_1, u_2 \in V^* \exists (w_l, w_r) \in P : u = u_1 w_l u_2 \text{ und } v = u_1 w_r u_2$$

Ist der Kontext klar, wird das tief gestellte  $G$  weggelassen. Ferner bedienen wir uns wieder der reflexiven, transitiven Hülle  $\xrightarrow{*}_G$  für **mehrschrittige Ableitungen**.

# Ableitung

## Definition (Ableitung)

Die **einschrittige Ableitung** eines Wortes  $v$  aus einem Wort  $u$  mittels einer Produktion einer Grammatik  $G$  wird notiert als

$u \xrightarrow{G} v$ . Dabei ist die Relation  $\xrightarrow{G} \subseteq V^* \times V^*$  für alle

$u, v \in V^*$  definiert durch:  $u \xrightarrow{G} v$  gdw.

$$\exists u_1, u_2 \in V^* \exists (w_l, w_r) \in P : u = u_1 w_l u_2 \text{ und } v = u_1 w_r u_2$$

Ist der Kontext klar, wird das tief gestellte  $G$  weggelassen. Ferner bedienen wir uns wieder der reflexiven, transitiven Hülle  $\xrightarrow{*}_G$  für **mehrschrittige Ableitungen**.

# Ableitung

## Definition (Ableitung)

Die **einschrittige Ableitung** eines Wortes  $v$  aus einem Wort  $u$  mittels einer Produktion einer Grammatik  $G$  wird notiert als

$u \xrightarrow{G} v$ . Dabei ist die Relation  $\xrightarrow{G} \subseteq V^* \times V^*$  für alle

$u, v \in V^*$  definiert durch:  $u \xrightarrow{G} v$  gdw.

$$\exists u_1, u_2 \in V^* \exists (w_l, w_r) \in P : u = u_1 w_l u_2 \text{ und } v = u_1 w_r u_2$$

Ist der Kontext klar, wird das tief gestellte  $G$  weggelassen. Ferner bedienen wir uns wieder der reflexiven, transitiven Hülle  $\xrightarrow{G}^*$  für **mehrschrittige Ableitungen**.

# Grammatiken. Beispiel

## Vorgehen

Eine Ableitung funktioniert also so:

- Das aktuelle Wort  $w$  betrachten
- Treten in  $w$  linke Seiten von Regeln auf?
- Falls ja, wähle eine dieser Regeln und
- ersetze die in  $w$  auftretende linke Seite der Regel
- durch die rechte Seite dieser Regel

## Beispiel

Mit  $S \rightarrow ASB \mid AB$ ,  $AB \rightarrow c$ ,  $AcB \rightarrow c$  gilt

$$S \Rightarrow ASB \Rightarrow AAB \Rightarrow AcB \Rightarrow c$$

# Grammatiken. Beispiel

## Vorgehen

Eine Ableitung funktioniert also so:

- Das aktuelle Wort  $w$  betrachten
- Treten in  $w$  linke Seiten von Regeln auf?
- Falls ja, wähle eine dieser Regeln und
- ersetze die in  $w$  auftretende linke Seite der Regel
- durch die rechte Seite dieser Regel

## Beispiel

Mit  $S \rightarrow ASB \mid AB$ ,  $AB \rightarrow c$ ,  $AcB \rightarrow c$  gilt

$$S \Rightarrow ASB \Rightarrow AAB \Rightarrow AcB \Rightarrow c$$



# Grammatiken. Beispiel

## Vorgehen

Eine Ableitung funktioniert also so:

- Das aktuelle Wort  $w$  betrachten
- Treten in  $w$  linke Seiten von Regeln auf?
- Falls ja, wähle eine dieser Regeln und
  - ersetze die in  $w$  auftretende linke Seite der Regel
  - durch die rechte Seite dieser Regel

## Beispiel

Mit  $S \rightarrow ASB \mid AB$ ,  $AB \rightarrow c$ ,  $AcB \rightarrow c$  gilt

$$S \Rightarrow ASB \Rightarrow AAB \Rightarrow AcB \Rightarrow c$$

# Grammatiken. Beispiel

## Vorgehen

Eine Ableitung funktioniert also so:

- Das aktuelle Wort  $w$  betrachten
- Treten in  $w$  linke Seiten von Regeln auf?
- Falls ja, wähle eine dieser Regeln und
- ersetze die in  $w$  auftretende linke Seite der Regel
- durch die rechte Seite dieser Regel

## Beispiel

Mit  $S \rightarrow ASB \mid AB$ ,  $AB \rightarrow c$ ,  $AcB \rightarrow c$  gilt

$$S \Rightarrow ASB \Rightarrow AAB \Rightarrow AcB \Rightarrow c$$

# Grammatiken. Beispiel

## Vorgehen

Eine Ableitung funktioniert also so:

- Das aktuelle Wort  $w$  betrachten
- Treten in  $w$  linke Seiten von Regeln auf?
- Falls ja, wähle eine dieser Regeln und
- ersetze die in  $w$  auftretende linke Seite der Regel
- durch die rechte Seite dieser Regel

## Beispiel

Mit  $S \rightarrow ASB \mid AB$ ,  $AB \rightarrow c$ ,  $AcB \rightarrow c$  gilt

$$S \Rightarrow ASB \Rightarrow AABBB \Rightarrow AcB \Rightarrow c$$

# Grammatiken. Beispiel

## Vorgehen

Eine Ableitung funktioniert also so:

- Das aktuelle Wort  $w$  betrachten
- Treten in  $w$  linke Seiten von Regeln auf?
- Falls ja, wähle eine dieser Regeln und
- ersetze die in  $w$  auftretende linke Seite der Regel
- durch die rechte Seite dieser Regel

## Beispiel

Mit  $S \rightarrow ASB \mid AB$ ,  $AB \rightarrow c$ ,  $AcB \rightarrow c$  gilt

$$S \Rightarrow ASB \Rightarrow AABBB \Rightarrow AcB \Rightarrow c$$

# Generierte Sprache

## Definition (Generierte Sprache)

Sei  $G = (V_N, V_T, P, S)$  eine Grammatik. Die von  $G$  **generierte** oder **erzeugte** Sprache ist

$$L(G) := \{w \in V_T^* \mid S \xrightarrow[G]{*} w\}$$

# Fragen...

Wenn man zwei Regeln mit gleichen linken Seite hat, wie wird ausgewählt welche Regel benutzt wird?

- 1 Das darf es nicht geben!
- 2 Darf man frei wählen!
- 3 Die, die am weitesten links steht!
- 4 Es werden alle benutzt!

# Fragen...

Kann man zwei Regeln mit gleicher rechten Seite haben?

- 1 Ja!
- 2 Nein!

# Fragen...

Kann es mehrere Möglichkeiten geben, um das gleiche Wort abzuleiten?

- 1 Ja!
- 2 Nein!



# Zur Nachbereitung

## Zur Nachbereitung

Richtige Antworten sind:

- ① 2
- ② 1
- ③ 1

# Kontextfreie Grammatiken

Die bisher eingeführten Grammatiken sind sehr mächtig.

Wir wollen sie jetzt zunächst einschränken und bei den eingeschränkten Grammatiken gucken, was man damit alles machen kann...

# Kontextfreie Grammatiken. Formale Definition

## Definition (Kontextfreie Grammatik (CFG))

Eine **kontextfreie Grammatik** (CFG) ist ein Quadrupel  $G = (V_N, V_T, P, S)$  mit

- 1 Dem endlichen Alphabet von **Nonterminalen**  $V_N$ .
- 2 Dem endlichen Alphabet von **Terminalen**  $V_T$  mit  $V_T \cap V_N = \emptyset$ . Das *Gesamtalphabet* wird mit  $V := V_T \cup V_N$  bezeichnet.
- 3 Der endlichen Menge von **Produktionen** (oder *Regeln*)  $P \subseteq V_N \times V^*$ .
- 4 Dem **Startsymbol**  $S \in V_N$ .

Kontextfreie Grammatiken werden auch als **Typ-2**-Grammatiken bezeichnet.

# Kontextfreie Grammatiken. Formale Definition

## Definition (Kontextfreie Grammatik (CFG))

Eine **kontextfreie Grammatik** (CFG) ist ein Quadrupel  $G = (V_N, V_T, P, S)$  mit

- 1 Dem endlichen Alphabet von **Nonterminalen**  $V_N$ .
- 2 Dem endlichen Alphabet von **Terminalen**  $V_T$  mit  $V_T \cap V_N = \emptyset$ . Das *Gesamtalphabet* wird mit  $V := V_T \cup V_N$  bezeichnet.
- 3 Der endlichen Menge von **Produktionen** (oder *Regeln*)  $P \subseteq V_N \times V^*$ .
- 4 Dem **Startsymbol**  $S \in V_N$ .

Kontextfreie Grammatiken werden auch als **Typ-2**-Grammatiken bezeichnet.

# Kontextfreie Grammatiken. Formale Definition

## Definition (Kontextfreie Grammatik (CFG))

Eine **kontextfreie Grammatik** (CFG) ist ein Quadrupel  $G = (V_N, V_T, P, S)$  mit

- 1 Dem endlichen Alphabet von **Nonterminalen**  $V_N$ .
- 2 Dem endlichen Alphabet von **Terminalen**  $V_T$  mit  $V_T \cap V_N = \emptyset$ . Das *Gesamtalphabet* wird mit  $V := V_T \cup V_N$  bezeichnet.
- 3 Der endlichen Menge von **Produktionen** (oder *Regeln*)  $P \subseteq V_N \times V^*$ .
- 4 Dem **Startsymbol**  $S \in V_N$ .

Kontextfreie Grammatiken werden auch als **Typ-2**-Grammatiken bezeichnet.

# Kontextfreie Grammatiken. Formale Definition

## Definition (Kontextfreie Grammatik (CFG))

Eine **kontextfreie Grammatik** (CFG) ist ein Quadrupel  $G = (V_N, V_T, P, S)$  mit

- 1 Dem endlichen Alphabet von **Nonterminalen**  $V_N$ .
- 2 Dem endlichen Alphabet von **Terminalen**  $V_T$  mit  $V_T \cap V_N = \emptyset$ . Das *Gesamtalphabet* wird mit  $V := V_T \cup V_N$  bezeichnet.
- 3 Der endlichen Menge von **Produktionen** (oder *Regeln*)  $P \subseteq V_N \times V^*$ .
- 4 Dem **Startsymbol**  $S \in V_N$ .

Kontextfreie Grammatiken werden auch als **Typ-2**-Grammatiken bezeichnet.

# Kontextfreie Grammatiken. Formale Definition

## Definition (Kontextfreie Grammatik (CFG))

Eine **kontextfreie Grammatik** (CFG) ist ein Quadrupel  $G = (V_N, V_T, P, S)$  mit

- 1 Dem endlichen Alphabet von **Nonterminalen**  $V_N$ .
- 2 Dem endlichen Alphabet von **Terminalen**  $V_T$  mit  $V_T \cap V_N = \emptyset$ . Das *Gesamtalphabet* wird mit  $V := V_T \cup V_N$  bezeichnet.
- 3 Der endlichen Menge von **Produktionen** (oder *Regeln*)  $P \subseteq V_N \times V^*$ .
- 4 Dem **Startsymbol**  $S \in V_N$ .

Kontextfreie Grammatiken werden auch als **Typ-2**-Grammatiken bezeichnet.

# Kontextfreie Grammatiken. Formale Definition

## Definition (Kontextfreie Grammatik (CFG))

Eine **kontextfreie Grammatik** (CFG) ist ein Quadrupel  $G = (V_N, V_T, P, S)$  mit

- 1 Dem endlichen Alphabet von **Nonterminalen**  $V_N$ .
- 2 Dem endlichen Alphabet von **Terminalen**  $V_T$  mit  $V_T \cap V_N = \emptyset$ . Das *Gesamtalphabet* wird mit  $V := V_T \cup V_N$  bezeichnet.
- 3 Der endlichen Menge von **Produktionen** (oder *Regeln*)  $P \subseteq V_N \times V^*$ .
- 4 Dem **Startsymbol**  $S \in V_N$ .

Kontextfreie Grammatiken werden auch als **Typ-2**-Grammatiken bezeichnet.



# Grammatiken vs. kontextfreie Grammatiken

## Anmerkung

Der Unterschied zwischen (allgemeinen) Grammatiken und kontextfreien Grammatiken liegt lediglich in der Menge der Produktionen  $P$ :

- bei allgemeinen Grammatiken:

$$P \subseteq (V^* \setminus V_T^*) \times V^*$$

- bei kontextfreien Grammatiken:

$$P \subseteq V_N \times V^*$$

Bei einer kontextfreien Grammatik ist bei einer Regel also stets genau ein Nonterminal auf der linken Seite! Daher auch der Name kontextfrei. Das Nonterminal wird unabhängig von dem Kontext, in dem es steht, ersetzt.

# Grammatiken vs. kontextfreie Grammatiken

## Anmerkung

Der Unterschied zwischen (allgemeinen) Grammatiken und kontextfreien Grammatiken liegt lediglich in der Menge der Produktionen  $P$ :

- bei allgemeinen Grammatiken:

$$P \subseteq (V^* \setminus V_T^*) \times V^*$$

- bei kontextfreien Grammatiken:

$$P \subseteq V_N \times V^*$$

Bei einer kontextfreien Grammatik ist bei einer Regel also stets genau ein Nonterminal auf der linken Seite! Daher auch der Name kontextfrei. Das Nonterminal wird unabhängig von dem Kontext, in dem es steht, ersetzt.

# CFG - das $\lambda$

## Definition ( $\lambda$ -Produktionen, $\lambda$ -frei)

- Eine kontextfreie Produktion  $(A, \lambda)$  wird als  $\lambda$ -**Produktion** bezeichnet.
- Besitzt eine CFG keine  $\lambda$ -Produktionen, so heißt sie  $\lambda$ -**frei**.

# Sprachfamilie CF

## Definition (Sprachfamilie CF)

Ableitung und akzeptierte Sprache ist wie bei Grammatiken definiert. Die **Familie der kontextfreien Sprachen** ist dann jene Familie von Sprachen, für die es eine kontextfreie Grammatik, die sie generiert. Abgekürzt wird diese Sprachfamilie mit CF.

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

Die Grammatik  $G$  ist gegeben durch die Produktionen

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

Der Vollständigkeit halber:

$$V_N := \{S, A, B\}$$

$$V_T := \{a, b\}$$

Das Startsymbol ist  $S$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

Die Grammatik  $G$  ist gegeben durch die Produktionen

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

Der Vollständigkeit halber:

$$V_N := \{S, A, B\}$$

$$V_T := \{a, b\}$$

Das Startsymbol ist  $S$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

Die Grammatik  $G$  ist gegeben durch die Produktionen

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

Der Vollständigkeit halber:

$$V_N := \{S, A, B\}$$

$$V_T := \{a, b\}$$

Das Startsymbol ist  $S$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$M \subseteq L(G)$ . Sei  $w = a^n b^m \in M$ . In  $G$  können wir nun wie folgt ableiten:

$$\begin{aligned} S &\Rightarrow AB \Rightarrow aAB \Rightarrow \dots \Rightarrow a^n AB \Rightarrow a^n B \\ &\Rightarrow a^n bB \Rightarrow \dots \Rightarrow a^n b^m B \Rightarrow a^n b^m \end{aligned}$$

Also gilt auch  $w \in L(G)$ .



# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$M \subseteq L(G)$ . Sei  $w = a^n b^m \in M$ . In  $G$  können wir nun wie folgt ableiten:

$$\begin{aligned} S &\Rightarrow AB \Rightarrow aAB \Rightarrow \dots \Rightarrow a^n AB \Rightarrow a^n B \\ &\Rightarrow a^n bB \Rightarrow \dots \Rightarrow a^n b^m B \Rightarrow a^n b^m \end{aligned}$$

Also gilt auch  $w \in L(G)$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$M \subseteq L(G)$ . Sei  $w = a^n b^m \in M$ . In  $G$  können wir nun wie folgt ableiten:

$$\begin{aligned} S &\Rightarrow AB \Rightarrow aAB \Rightarrow \dots \Rightarrow a^n AB \Rightarrow a^n B \\ &\Rightarrow a^n bB \Rightarrow \dots \Rightarrow a^n b^m B \Rightarrow a^n b^m \end{aligned}$$

Also gilt auch  $w \in L(G)$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$M \subseteq L(G)$ . Sei  $w = a^n b^m \in M$ . In  $G$  können wir nun wie folgt ableiten:

$$\begin{aligned} S &\Rightarrow AB \Rightarrow aAB \Rightarrow \dots \Rightarrow a^n AB \Rightarrow a^n B \\ &\Rightarrow a^n bB \Rightarrow \dots \Rightarrow a^n b^m B \Rightarrow a^n b^m \end{aligned}$$

Also gilt auch  $w \in L(G)$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$L(G) \subseteq M$ . An den Produktionen erkennen wir zunächst, dass  $w \in \{a\}^*$  für jedes  $w$  mit  $A \xRightarrow{*} w$  gilt, denn  $i$ -malige Anwendung von  $A \rightarrow aA$  führt zu der Satzform  $a^i A$ . Dann kann mit  $A \rightarrow \lambda$  das  $A$  gelöscht werden. Ebenso gilt  $v \in \{b\}^*$  für jedes  $v$  mit  $B \xRightarrow{*} v$ .

Eine Ableitung eines Wortes  $u \in L(G)$  muss nun mit  $S \Rightarrow AB$  beginnen. Anschließend kann wie eben erläutert  $A$  zu  $a^i$  und  $B$  zu  $b^j$  mit  $i, j \in \mathbb{N}$  abgeleitet werden, d.h.  $u$  hat die Form  $a^i b^j$ . Dies zeigt  $u \in M$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$L(G) \subseteq M$ . An den Produktionen erkennen wir zunächst, dass  $w \in \{a\}^*$  für jedes  $w$  mit  $A \xRightarrow{*} w$  gilt, denn  $i$ -malige Anwendung von  $A \rightarrow aA$  führt zu der Satzform  $a^i A$ . Dann kann mit  $A \rightarrow \lambda$  das  $A$  gelöscht werden. Ebenso gilt  $v \in \{b\}^*$  für jedes  $v$  mit  $B \xRightarrow{*} v$ .

Eine Ableitung eines Wortes  $u \in L(G)$  muss nun mit  $S \Rightarrow AB$  beginnen. Anschließend kann wie eben erläutert  $A$  zu  $a^i$  und  $B$  zu  $b^j$  mit  $i, j \in \mathbb{N}$  abgeleitet werden, d.h.  $u$  hat die Form  $a^i b^j$ . Dies zeigt  $u \in M$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$L(G) \subseteq M$ . An den Produktionen erkennen wir zunächst, dass  $w \in \{a\}^*$  für jedes  $w$  mit  $A \xRightarrow{*} w$  gilt, denn  $i$ -malige Anwendung von  $A \rightarrow aA$  führt zu der Satzform  $a^i A$ . Dann kann mit  $A \rightarrow \lambda$  das  $A$  gelöscht werden. Ebenso gilt  $v \in \{b\}^*$  für jedes  $v$  mit  $B \xRightarrow{*} v$ .

Eine Ableitung eines Wortes  $u \in L(G)$  muss nun mit  $S \Rightarrow AB$  beginnen. Anschließend kann wie eben erläutert  $A$  zu  $a^i$  und  $B$  zu  $b^j$  mit  $i, j \in \mathbb{N}$  abgeleitet werden, d.h.  $u$  hat die Form  $a^i b^j$ . Dies zeigt  $u \in M$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$L(G) \subseteq M$ . An den Produktionen erkennen wir zunächst, dass  $w \in \{a\}^*$  für jedes  $w$  mit  $A \xRightarrow{*} w$  gilt, denn  $i$ -malige Anwendung von  $A \rightarrow aA$  führt zu der Satzform  $a^i A$ . Dann kann mit  $A \rightarrow \lambda$  das  $A$  gelöscht werden. Ebenso gilt  $v \in \{b\}^*$  für jedes  $v$  mit  $B \xRightarrow{*} v$ .

Eine Ableitung eines Wortes  $u \in L(G)$  muss nun mit  $S \Rightarrow AB$  beginnen. Anschließend kann wie eben erläutert  $A$  zu  $a^i$  und  $B$  zu  $b^j$  mit  $i, j \in \mathbb{N}$  abgeleitet werden, d.h.  $u$  hat die Form  $a^i b^j$ . Dies zeigt  $u \in M$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$L(G) \subseteq M$ . An den Produktionen erkennen wir zunächst, dass  $w \in \{a\}^*$  für jedes  $w$  mit  $A \xRightarrow{*} w$  gilt, denn  $i$ -malige Anwendung von  $A \rightarrow aA$  führt zu der Satzform  $a^i A$ . Dann kann mit  $A \rightarrow \lambda$  das  $A$  gelöscht werden. Ebenso gilt  $v \in \{b\}^*$  für jedes  $v$  mit  $B \xRightarrow{*} v$ .

Eine Ableitung eines Wortes  $u \in L(G)$  muss nun mit  $S \Rightarrow AB$  beginnen. Anschließend kann wie eben erläutert  $A$  zu  $a^i$  und  $B$  zu  $b^j$  mit  $i, j \in \mathbb{N}$  abgeleitet werden, d.h.  $u$  hat die Form  $a^i b^j$ . Dies zeigt  $u \in M$ .



# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$L(G) \subseteq M$ . An den Produktionen erkennen wir zunächst, dass  $w \in \{a\}^*$  für jedes  $w$  mit  $A \xrightarrow{*} w$  gilt, denn  $i$ -malige Anwendung von  $A \rightarrow aA$  führt zu der Satzform  $a^i A$ . Dann kann mit  $A \rightarrow \lambda$  das  $A$  gelöscht werden. Ebenso gilt  $v \in \{b\}^*$  für jedes  $v$  mit  $B \xrightarrow{*} v$ .

Eine Ableitung eines Wortes  $u \in L(G)$  muss nun mit  $S \Rightarrow AB$  beginnen. Anschließend kann wie eben erläutert  $A$  zu  $a^i$  und  $B$  zu  $b^j$  mit  $i, j \in \mathbb{N}$  abgeleitet werden, d.h.  $u$  hat die Form  $a^i b^j$ . Dies zeigt  $u \in M$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$L(G) \subseteq M$ . An den Produktionen erkennen wir zunächst, dass  $w \in \{a\}^*$  für jedes  $w$  mit  $A \xrightarrow{*} w$  gilt, denn  $i$ -malige Anwendung von  $A \rightarrow aA$  führt zu der Satzform  $a^i A$ . Dann kann mit  $A \rightarrow \lambda$  das  $A$  gelöscht werden. Ebenso gilt  $v \in \{b\}^*$  für jedes  $v$  mit  $B \xrightarrow{*} v$ .

Eine Ableitung eines Wortes  $u \in L(G)$  muss nun mit  $S \Rightarrow AB$  beginnen. Anschließend kann wie eben erläutert  $A$  zu  $a^i$  und  $B$  zu  $b^j$  mit  $i, j \in \mathbb{N}$  abgeleitet werden, d.h.  $u$  hat die Form  $a^i b^j$ . Dies zeigt  $u \in M$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$L(G) \subseteq M$ . An den Produktionen erkennen wir zunächst, dass  $w \in \{a\}^*$  für jedes  $w$  mit  $A \xrightarrow{*} w$  gilt, denn  $i$ -malige Anwendung von  $A \rightarrow aA$  führt zu der Satzform  $a^i A$ . Dann kann mit  $A \rightarrow \lambda$  das  $A$  gelöscht werden. Ebenso gilt  $v \in \{b\}^*$  für jedes  $v$  mit  $B \xrightarrow{*} v$ .

Eine Ableitung eines Wortes  $u \in L(G)$  muss nun mit  $S \Rightarrow AB$  beginnen. Anschließend kann wie eben erläutert  $A$  zu  $a^i$  und  $B$  zu  $b^j$  mit  $i, j \in \mathbb{N}$  abgeleitet werden, d.h.  $u$  hat die Form  $a^i b^j$ . Dies zeigt  $u \in M$ .

# Beispiel 1

$$M = a^*b^* = \{a^n b^m \mid n, m \in \mathbb{N}\}$$

$$S \rightarrow AB, A \rightarrow aA \mid \lambda, B \rightarrow bB \mid \lambda$$

$L(G) \subseteq M$ . An den Produktionen erkennen wir zunächst, dass  $w \in \{a\}^*$  für jedes  $w$  mit  $A \xrightarrow{*} w$  gilt, denn  $i$ -malige Anwendung von  $A \rightarrow aA$  führt zu der Satzform  $a^i A$ . Dann kann mit  $A \rightarrow \lambda$  das  $A$  gelöscht werden. Ebenso gilt  $v \in \{b\}^*$  für jedes  $v$  mit  $B \xrightarrow{*} v$ .

Eine Ableitung eines Wortes  $u \in L(G)$  muss nun mit  $S \Rightarrow AB$  beginnen. Anschließend kann wie eben erläutert  $A$  zu  $a^i$  und  $B$  zu  $b^j$  mit  $i, j \in \mathbb{N}$  abgeleitet werden, d.h.  $u$  hat die Form  $a^i b^j$ . Dies zeigt  $u \in M$ .

## Beispiel 2

$$M = \{a^n b^n \mid n \in \mathbb{N}\}$$

$$S \rightarrow aSb \mid \lambda$$

$M \subseteq L(G)$ . Sei  $w = a^n b^n \in M$ . In  $G$  können wir nun wie folgt ableiten:

$$S \Rightarrow aSb \Rightarrow \dots \Rightarrow a^n S b^n \Rightarrow a^n b^n$$

Also gilt auch  $w \in L(G)$ .

## Beispiel 2

$$M = \{a^n b^n \mid n \in \mathbb{N}\}$$

$$S \rightarrow aSb \mid \lambda$$

$M \subseteq L(G)$ . Sei  $w = a^n b^n \in M$ . In  $G$  können wir nun wie folgt ableiten:

$$S \Rightarrow aSb \Rightarrow \dots \Rightarrow a^n S b^n \Rightarrow a^n b^n$$

Also gilt auch  $w \in L(G)$ .

## Beispiel 2

$$M = \{a^n b^n \mid n \in \mathbb{N}\}$$

$$S \rightarrow aSb \mid \lambda$$

$M \subseteq L(G)$ . Sei  $w = a^n b^n \in M$ . In  $G$  können wir nun wie folgt ableiten:

$$S \Rightarrow aSb \Rightarrow \dots \Rightarrow a^n S b^n \Rightarrow a^n b^n$$

Also gilt auch  $w \in L(G)$ .

## Beispiel 2

$$M = \{a^n b^n \mid n \in \mathbb{N}\}$$

$$S \rightarrow aSb \mid \lambda$$

$M \subseteq L(G)$ . Sei  $w = a^n b^n \in M$ . In  $G$  können wir nun wie folgt ableiten:

$$S \Rightarrow aSb \Rightarrow \dots \Rightarrow a^n S b^n \Rightarrow a^n b^n$$

Also gilt auch  $w \in L(G)$ .



## Beispiel 2

$$M = \{a^n b^n \mid n \in \mathbb{N}\}$$

$$S \rightarrow aSb \mid \lambda$$

$L(G) \subseteq M$ . Sei  $w \in L(G)$ . Da wir nur zwei Regeln haben, ist die Argumentation recht einfach. Die einzige Möglichkeit zu einem Wort zu kommen ist die Produktion  $S \rightarrow \lambda$ . Vorher kann  $i$ -mal die Produktion  $S \rightarrow aSb$  benutzt werden.  $w$  kann damit nur die Form  $a^i b^i$  haben und damit gilt auch  $w \in M$ .

## Beispiel 2

$$M = \{a^n b^n \mid n \in \mathbb{N}\}$$

$$S \rightarrow aSb \mid \lambda$$

$L(G) \subseteq M$ . Sei  $w \in L(G)$ . Da wir nur zwei Regeln haben, ist die Argumentation recht einfach. Die einzige Möglichkeit zu einem Wort zu kommen ist die Produktion  $S \rightarrow \lambda$ . Vorher kann  $i$ -mal die Produktion  $S \rightarrow aSb$  benutzt werden.  $w$  kann damit nur die Form  $a^i b^i$  haben und damit gilt auch  $w \in M$ .

# Zur Konstruktion von Grammatiken

## Konstruktionstipps

Zwei Tipps bei der Konstruktion von Grammatiken:

- 1 Will man einen "linken" und einen "rechten" Teil miteinander in Beziehung setzen, wie bei  $a^n b^n$ , so klappt dies meist, indem das Nonterminal nach innen wandert:  $S \rightarrow aSb$
- 2 Will man eine Zeichenkette "sequentiell" aufbauen, wie bei  $a^k$ , so kann man das Nonterminal meist an den Rand setzen  $S \rightarrow aS$

# Zur Konstruktion von Grammatiken

## Konstruktionstipps

Zwei Tipps bei der Konstruktion von Grammatiken:

- 1 Will man einen “linken” und einen “rechten” Teil miteinander in Beziehung setzen, wie bei  $a^n b^n$ , so klappt dies meist, indem das Nonterminal nach innen wandert:  $S \rightarrow aSb$
- 2 Will man eine Zeichenkette “sequentiell” aufbauen, wie bei  $a^k$ , so kann man das Nonterminal meist an den Rand setzen  $S \rightarrow aS$

# Zur Konstruktion von Grammatiken

## Konstruktionstipps

Zwei Tipps bei der Konstruktion von Grammatiken:

- 1 Will man einen "linken" und einen "rechten" Teil miteinander in Beziehung setzen, wie bei  $a^n b^n$ , so klappt dies meist, indem das Nonterminal nach innen wandert:  $S \rightarrow aSb$
- 2 Will man eine Zeichenkette "sequentiell" aufbauen, wie bei  $a^k$ , so kann man das Nonterminal meist an den Rand setzen  $S \rightarrow aS$

# Typ-3 Grammatiken

## Definition (Kontextfreie Grammatik (CFG))

Eine **kontextfreie Grammatik**  $G = (V_N, V_T, P, S)$  heißt

- **linear**, falls  $P \subseteq V_N \times (V_T^* \cdot V_N \cdot V_T^* \cup V_T^*)$
- **linkslin**ear, falls  $P \subseteq V_N \times (V_N \cdot V_T^* \cup V_T^*)$
- **rechtslin**ear, falls  $P \subseteq V_N \times (V_T^* \cdot V_N \cup V_T^*)$

Wir werden uns auf rechtslineare Grammatiken beschränken. Diese heißen auch **Typ-3**-Grammatiken oder **reguläre** Grammatiken.

# Typ-3 Grammatiken

## Definition (Kontextfreie Grammatik (CFG))

Eine **kontextfreie Grammatik**  $G = (V_N, V_T, P, S)$  heißt

- **linear**, falls  $P \subseteq V_N \times (V_T^* \cdot V_N \cdot V_T^* \cup V_T^*)$
- **linkslin**ear, falls  $P \subseteq V_N \times (V_N \cdot V_T^* \cup V_T^*)$
- **rechtslin**ear, falls  $P \subseteq V_N \times (V_T^* \cdot V_N \cup V_T^*)$

Wir werden uns auf rechtslin

ear Grammatiken beschränken. Diese heißen auch **Typ-3**-Grammatiken oder **reguläre** Grammatiken.

# Typ-3 Grammatiken

## Definition (Kontextfreie Grammatik (CFG))

Eine **kontextfreie Grammatik**  $G = (V_N, V_T, P, S)$  heißt

- **linear**, falls  $P \subseteq V_N \times (V_T^* \cdot V_N \cdot V_T^* \cup V_T^*)$
- **linkslinear**, falls  $P \subseteq V_N \times (V_N \cdot V_T^* \cup V_T^*)$
- **rechtslinear**, falls  $P \subseteq V_N \times (V_T^* \cdot V_N \cup V_T^*)$

Wir werden uns auf rechtslineare Grammatiken beschränken. Diese heißen auch **Typ-3**-Grammatiken oder **reguläre** Grammatiken.



# Typ-3 Grammatiken

## Definition (Kontextfreie Grammatik (CFG))

Eine **kontextfreie Grammatik**  $G = (V_N, V_T, P, S)$  heißt

- **linear**, falls  $P \subseteq V_N \times (V_T^* \cdot V_N \cdot V_T^* \cup V_T^*)$
- **linkslinear**, falls  $P \subseteq V_N \times (V_N \cdot V_T^* \cup V_T^*)$
- **rechtslinear**, falls  $P \subseteq V_N \times (V_T^* \cdot V_N \cup V_T^*)$

Wir werden uns auf rechtslineare Grammatiken beschränken. Diese heißen auch **Typ-3**-Grammatiken oder **reguläre** Grammatiken.

# Grammatiken und Automaten

## Satz

*Eine Sprache  $R \subseteq \Sigma^*$  ist regulär genau dann, wenn es eine rechtslineare Grammatik  $G$  gibt mit  $L(G) = R$ .*

## Satz

*Eine Sprache  $C \subseteq \Sigma^*$  ist kontextfrei genau dann, wenn es einen PDA gibt, der  $C$  mit leerem Keller akzeptiert.*

## Bemerkung

Der erste Satz stellt einen Zusammenhang zwischen DFAs und rechtslinearen Grammatiken her. Der zweite Satz stellt einen Zusammenhang zwischen PDAs und kontextfreien Grammatiken her. **DFAs und rechtslineare Grammatiken sind hiernach äquivalent, ebenso wie PDAs und kontextfreie Grammatiken!**

# Grammatiken und Automaten

## Satz

*Eine Sprache  $R \subseteq \Sigma^*$  ist regulär genau dann, wenn es eine rechtslineare Grammatik  $G$  gibt mit  $L(G) = R$ .*

## Satz

*Eine Sprache  $C \subseteq \Sigma^*$  ist kontextfrei genau dann, wenn es einen PDA gibt, der  $C$  mit leerem Keller akzeptiert.*

## Bemerkung

Der erste Satz stellt einen Zusammenhang zwischen DFAs und rechtslinearen Grammatiken her. Der zweite Satz stellt einen Zusammenhang zwischen PDAs und kontextfreien Grammatiken her. **DFAs und rechtslineare Grammatiken sind hiernach äquivalent, ebenso wie PDAs und kontextfreie Grammatiken!**

# Grammatiken und Automaten

## Satz

*Eine Sprache  $R \subseteq \Sigma^*$  ist regulär genau dann, wenn es eine rechtslineare Grammatik  $G$  gibt mit  $L(G) = R$ .*

## Satz

*Eine Sprache  $C \subseteq \Sigma^*$  ist kontextfrei genau dann, wenn es einen PDA gibt, der  $C$  mit leerem Keller akzeptiert.*

## Bemerkung

Der erste Satz stellt einen Zusammenhang zwischen DFAs und rechtslinearen Grammatiken her. Der zweite Satz stellt einen Zusammenhang zwischen PDAs und kontextfreien Grammatiken her.

**DFAs und rechtslineare Grammatiken sind hiernach äquivalent, ebenso wie PDAs und kontextfreie Grammatiken!**

# Ausblick

Das machen wir nächstes Mal... :)