

# Grundlagen der Theoretischen Informatik

## 4. Kellerautomaten und kontextfreie Sprachen (II)

19.05.2021

Viorica Sofronie-Stokkermans

e-mail: [sofronie@uni-koblenz.de](mailto:sofronie@uni-koblenz.de)

# Übersicht

---

1. Motivation
2. Terminologie
3. Endliche Automaten und reguläre Sprachen
4. Kellerautomaten und kontextfreie Sprachen
5. Turingmaschinen und rekursiv aufzählbare Sprachen
6. Berechenbarkeit, (Un-)Entscheidbarkeit
7. Komplexitätsklassen P und NP

# Bis jetzt

---

## Kellerautomaten und kontextfreie Sprachen

- Kontextfreie Grammatiken
  - Ableitungsbäume
  - Links- und Rechtsableitung
  - Mehrdeutigkeit
- Umformung von Grammatiken
  - Startsymbol nur links
  - Eliminierung nutzloser Symbole:
    - \* Definition: Erreichbare/co-erreichbare Symbole; nutzlose Symbole
    - \* Algorithmus zur Berechnung der co-erreichbaren Variablen
    - \* Algorithmus zur Berechnung der erreichbaren Symbole

# Normalform für Regeln

---

# Normalform für Regeln

---

## Theorem (Normalform)

Zu jeder Grammatik  $G$  (beliebigen Typs) existiert eine äquivalente Grammatik  $G'$ , bei der für alle Regeln  $P \rightarrow Q \in R'$  gilt:

- $Q \in V^*$  und  $P$  beliebig
- $Q \in T$  und  $P \in V$

Für alle Typen außer den linearen hat  $G'$  denselben Typ wie  $G$ .

# Normalform für Regeln

---

## Beweis

Für jedes Terminal  $t \in T$  erzeuge man eine neue Variable  $V_t$ .

- $V' = V \cup \{V_t \mid t \in T\}$
- $R'$  entsteht aus  $R$ , indem für jede Regel  $P \rightarrow Q \in R$  in  $Q$  alle Vorkommen eines Terminals  $t$  durch die zugehörige Variable  $V_t$  ersetzt werden. Außerdem enthält  $R'$  für jedes  $t \in T$  eine neue Regel  $V_t \rightarrow t$ .

# Beispiel 1

---

$G = (\{S\}, \{a, b, c\}, R, S)$  wobei

$$R = \left\{ \begin{array}{l} S \rightarrow aSb \\ S \rightarrow c \end{array} \right\}$$

$\mapsto$

$G' = (\{S, V_a, V_b, V_c\}, \{a, b, c\}, R', S)$  mit

$$R' = \left\{ \begin{array}{l} S \rightarrow V_a S V_b \\ S \rightarrow V_c \\ V_a \rightarrow a \\ V_b \rightarrow b \\ V_c \rightarrow c \end{array} \right\}$$

## Beispiel 2

---

$G = (\{S\}, \{a, c\}, R, S)$  wobei

$$R = \left\{ \begin{array}{l} S \rightarrow aS \\ S \rightarrow c \end{array} \right\}$$

$\mapsto$

$G' = (\{S, V_a, V_c\}, \{a, c\}, R', S)$  mit

$$R' = \left\{ \begin{array}{l} S \rightarrow V_a S \\ S \rightarrow V_c \\ V_a \rightarrow a \\ V_c \rightarrow c \end{array} \right\}$$

# Normalform für Regeln

---

## Beweis

Für jedes Terminal  $t \in T$  erzeuge man eine neue Variable  $V_t$ .

- $V' = V \cup \{V_t \mid t \in T\}$
- $R'$  entsteht aus  $R$ , indem für jede Regel  $P \rightarrow Q \in R$  in  $Q$  alle Vorkommen eines Terminals  $t$  durch die zugehörige Variable  $V_t$  ersetzt werden. Außerdem enthält  $R'$  für jedes  $t \in T$  eine neue Regel  $V_t \rightarrow t$ .

Also  $L(G') = L(G)$ ,

und für alle Sprachklassen außer  $\mathcal{L}_3$  hat  $G'$  denselben Typ wie  $G$ .

# Elimination von $\varepsilon$ -Regeln

---

**Idee:** Variablen, aus denen  $\varepsilon$  ableitbar ist, sollten eliminiert werden

# Elimination von $\varepsilon$ -Regeln

---

**Idee:** Variablen, aus denen  $\varepsilon$  ableitbar ist, sollten eliminiert werden

## Definition ( $\varepsilon$ -Regel, nullbare Variablen)

Eine Regel der Form

$$P \rightarrow \varepsilon \quad (P \text{ eine Variable})$$

heißt  $\varepsilon$ -Regel.

Eine Variable  $A$  heißt **nullbar**,  
falls

$$A \Longrightarrow^* \varepsilon$$

# Elimination von $\varepsilon$ -Regeln

---

## Theorem ( $\varepsilon$ -Regeln sind eliminierbar)

Zu jeder cf-Grammatik  $G$  existiert eine äquivalente cf-Grammatik  $G'$

- ohne  $\varepsilon$ -Regeln und nullbare Variablen, falls  $\varepsilon \notin L(G)$ ,
- mit der einzigen  $\varepsilon$ -Regel  $S \rightarrow \varepsilon$  und der einzigen nullbaren Variablen  $S$ , falls  $\varepsilon \in L(G)$  und  $S$  das Startsymbol ist.

# Elimination von $\varepsilon$ -Regeln

---

## Algorithmus zur Berechnung der nullbaren Variablen

**Input:** Grammatik  $G = (V, T, R, S)$        $S$  o.B.d.A. in keiner Regel rechts

**Output:** nullbare Variablen

$Alt := \emptyset$

$Neu := \{A \in V \mid A \rightarrow \varepsilon \in R\}$

**while**  $Alt \neq Neu$

{  $Alt := Neu$

**für alle**  $(P \rightarrow Q) \in R$  **do**

  { **if**  $Q = A_1 \dots A_n$  **and**  $A_i \in Neu$  für  $1 \leq i \leq n$  **and**  $P \notin Neu$ ,  
  **then**  $Neu := Neu \cup \{P\}$

  }

}

output  $Neu$

# Elimination von $\varepsilon$ -Regeln: Beispiel

---

$R :$

$S \rightarrow ABD$

$A \rightarrow ED \mid BB$

$B \rightarrow AC \mid \varepsilon$

$C \rightarrow \varepsilon$

$D \rightarrow d$

$E \rightarrow e$

Alt :=  $\emptyset$

New :=  $\{B, C\}$       [ $B \rightarrow \varepsilon \in R$

$C \rightarrow \varepsilon \in R$ ]

Alt  $\neq$  New

Alt :=  $\{B, C\}$

New :=  $\{B, C, A\}$       [ $A \rightarrow BB \in R$ ]

Alt  $\neq$  New

Alt :=  $\{B, C, A\}$

Alt = New

Für die Regelmengemenge  $R$  sind die Variablen  $A, B, C$  nullbar.

# Elimination von $\varepsilon$ -Regeln

---

Beweis (Forts.)

Ausgangsgrammatik  $G$  habe die Normalform, bei der für jede Regel  $P \rightarrow Q$ :  
 $Q \in V^*$  oder  $Q \in T$ .

Für jede Regel  $P \rightarrow A_1 \dots A_n$  generiere alle möglichen Kombinationen

$$P \rightarrow \alpha_1 \dots \alpha_n$$

mit

- $\alpha_i \in \{\varepsilon, A_i\}$  falls  $A_i$  nullbar
- $\alpha_i = A_i$  falls  $A_i$  nicht nullbar

Dann

- Füge alle diese neuen Regeln zur Grammatik hinzu
- Entferne alle Regeln der Form  $A \rightarrow \varepsilon$  mit  $A \neq S$

# Elimination von $\varepsilon$ -Regeln: Beispiel

---

| $R :$                               | $R' :$                                     |
|-------------------------------------|--|
| $S \rightarrow ABD$                 | $S \rightarrow ABD \mid AD \mid BD \mid D$ |
| $A \rightarrow ED \mid BB$          | $A \rightarrow ED \mid BB \mid B$          |
| $B \rightarrow AC \mid \varepsilon$ | $B \rightarrow AC \mid A \mid C$           |
| $C \rightarrow \varepsilon$         |  |
| $D \rightarrow d$                   | $D \rightarrow d$                          |
| $E \rightarrow e$                   | $E \rightarrow e$                          |

Für die Regelmenge  $R$  in der linken Spalte sind die Variablen  $A, B, C$  nullbar.

# Elimination von $\varepsilon$ -Regeln: Beispiel

---

| $R :$                               | $R' :$                                     |
|-------------------------------------|--|
| $S \rightarrow ABD$                 | $S \rightarrow ABD \mid AD \mid BD \mid D$ |
| $A \rightarrow ED \mid BB$          | $A \rightarrow ED \mid BB \mid B$          |
| $B \rightarrow AC \mid \varepsilon$ | $B \rightarrow AC \mid A \mid C$           |
| $C \rightarrow \varepsilon$         |  |
| $D \rightarrow d$                   | $D \rightarrow d$                          |
| $E \rightarrow e$                   | $E \rightarrow e$                          |

Für die Regelmenge  $R$  in der linken Spalte sind die Variablen  $A, B, C$  nullbar.

Der obige Algorithmus erzeugt aus  $R$  die rechts aufgeführte Regelmenge  $R'$ .

# Elimination von $\varepsilon$ -Regeln

---

Beweis ((Forts.))

**Zu zeigen:**

Für die neue Grammatik  $G'$  gilt:  $L(G') = L(G)$

Vorgehen:

- $G$  hat die Normalform:

Für jede Regel  $P \rightarrow Q$  gilt  $Q \in V^*$  oder  $Q \in T$ .

- Wir beweisen die etwas stärkere Behauptung

für alle  $A \in V$  für alle  $w \in (V \cup T)^* - \{\varepsilon\}$

$$\left( (A \Longrightarrow_G^* w) \quad \underline{\text{gdw}} \quad (A \Longrightarrow_{G'}^* w) \right),$$

- Daraus folgt sofort  $L(G') = L(G)$ .

# Elimination von $\varepsilon$ -Regeln

---

Beweis (Forts.) Wir beweisen die Behauptung

für alle  $A \in V$  für alle  $w \in (V \cup T)^* - \{\varepsilon\}$

$$((A \Longrightarrow_G^* w) \quad \underline{\text{gdw}} \quad (A \Longrightarrow_{G'}^* w)),$$

” $\Rightarrow$ ” Wir zeigen: Aus  $A \Longrightarrow_G^* w$  folgt  $A \Longrightarrow_{G'}^* w$  (Induktion über Länge einer Ableitung von  $A$  nach  $w$  in  $G$ ).

**Induktionsanfang:** Länge = 0.

Dann ist  $w = A$ , und  $A \Longrightarrow_{G'}^* A$  gilt immer.

**Induktionsschritt:** Es sei schon gezeigt: Wenn in  $G$  in  $n$  Schritten eine Ableitung  $B \Longrightarrow_G^* u$  durchgeführt werden kann, dann folgt, dass in  $G'$  die Ableitung  $B \Longrightarrow_{G'}^* u$  möglich ist.

# Elimination von $\varepsilon$ -Regeln

---

Beweis (Forts.)

Außerdem gelte in der Ausgangsgrammatik  $G$ :  $A \Longrightarrow_G^* w \neq \varepsilon$  in  $n + 1$  Schritten.

Dann gilt:

- $A \Longrightarrow_G w' \Longrightarrow_G^* w$ ,
- $w' = A_1 \dots A_\ell \Longrightarrow_G^* w_1 \dots w_\ell = w$ ,
- und es wird jeweils  $A_i$  zu  $w_i$  in höchstens  $n$  Schritten für geeignete  $w', A_1, \dots, A_\ell, w_1, \dots, w_\ell$ .
- Für  $1 \leq i \leq \ell$  gilt:
  - Entweder  $w_i \neq \varepsilon$  und  $A_i \Longrightarrow_G^* w_i$   
also (per Induktionsvoraussetzung)  $A_i \Longrightarrow_{G'}^* w_i$
  - oder  $w_i = \varepsilon$  und  $A_i \Longrightarrow_G^* w_i$ .

# Elimination von $\varepsilon$ -Regeln

---

Beweis (Forts.)

**Fall 1:**  $w_i = \varepsilon$ ,  $A_i$  ist nullbar.

Dann gibt es in  $G'$  eine Regel  $A \rightarrow A_1 \dots A_{i-1} A_{i+1} \dots A_\ell$  nach der obigen Konstruktionsvorschrift für  $G'$ , falls  $A_1 \dots A_{i-1} A_{i+1} \dots A_\ell \neq \varepsilon$ . Das ist der Fall, denn sonst hätten wir:  $A \Longrightarrow w' = \varepsilon \Longrightarrow^* w = \varepsilon$  (aus nichts wird nichts), aber  $w = \varepsilon$  ist ausgeschlossen.

**Fall 2:**  $w_i \neq \varepsilon$ . Dann gilt nach Induktionsvoraussetzung

$$A_i \Longrightarrow_{G'}^* w_i.$$

# Elimination von $\varepsilon$ -Regeln

---

Beweis (Forts.)

Wir haben also folgendes gezeigt:

Sei  $I = \{i \in \{1 \dots \ell\} \mid w_i \neq \varepsilon\} \neq \emptyset$ .

Dann gibt es in  $R'$  eine Regel  $A \rightarrow A_{i_1} \dots A_{i_m}$  mit  $I = \{i_1, \dots, i_m\}$ , und die  $A_i$  sind so angeordnet wie in der ursprünglichen Regel  $A \rightarrow A_1 \dots A_\ell$ .

Mit dieser neuen Regel können wir  $w$  so ableiten:

$$A \Longrightarrow_{G'} A_{i_1} \dots A_{i_m} \Longrightarrow_{G'}^* w_{i_1} \dots w_{i_m} = w$$

# Elimination von $\varepsilon$ -Regeln

---

Beweis (Forts.)

" $\Leftarrow$ " Wir zeigen: Aus  $A \Longrightarrow_{G'}^* w$  folgt  $A \Longrightarrow_G^* w$  (Induktion über Länge einer Ableitung von  $A$  nach  $w$  in  $G'$ ):

**Induktionsanfang:** Länge = 0. Dann ist  $w = A$ , und  $A \Longrightarrow_G^* A$  gilt immer.

**Induktionsschritt:** Es gelte für alle Ableitungen  $A \Longrightarrow_{G'}^* w$  einer Länge von höchstens  $n$ , dass  $A \Longrightarrow_G^* w$ .

Ist  $A \Longrightarrow_{G'}^* w$  eine Ableitung der Länge  $n + 1$ , so gibt es ein  $\ell$ , Wörter  $w_1, \dots, w_\ell$  und Variablen  $A_1, \dots, A_\ell$  mit  $A \Longrightarrow_{G'} A_1 \dots A_\ell \Longrightarrow_{G'}^* w = w_1 \dots w_\ell$ . Es gilt jeweils  $A_i \Longrightarrow_{G'}^* w_i$  in höchstens  $n$  Schritten, und  $w_i \neq \varepsilon$ .

# Elimination von $\varepsilon$ -Regeln

---

Beweis (Forts.)

Nach der Induktionsvoraussetzung folgt daraus:

- für die Originalgrammatik  $G$  gibt es Ableitungen  $A_i \Longrightarrow_G^* w_i$
- damit gibt es auch eine Ableitung  $A_1 \dots A_\ell \Longrightarrow_G^* w$ .

# Elimination von $\varepsilon$ -Regeln

---

Beweis (Forts.)

Nach der Induktionsvoraussetzung folgt daraus:

- für die Originalgrammatik  $G$  gibt es Ableitungen  $A_i \Longrightarrow_G^* w_i$
- damit gibt es auch eine Ableitung  $A_1 \dots A_\ell \Longrightarrow_G^* w$ .

Da es in  $G'$  eine Ableitung  $A \Longrightarrow_{G'} A_1 \dots A_\ell$  gibt, gibt es in  $R'$  eine Regel  $A \rightarrow A_1 \dots A_\ell$ . Wie ist diese Regel aus  $R$  entstanden?

# Elimination von $\varepsilon$ -Regeln

---

Beweis (Forts.)

Nach der Induktionsvoraussetzung folgt daraus:

- für die Originalgrammatik  $G$  gibt es Ableitungen  $A_i \Longrightarrow_G^* w_i$
- damit gibt es auch eine Ableitung  $A_1 \dots A_\ell \Longrightarrow_G^* w$ .

Da es in  $G'$  eine Ableitung  $A \Longrightarrow_{G'} A_1 \dots A_\ell$  gibt, gibt es in  $R'$  eine Regel  $A \rightarrow A_1 \dots A_\ell$ . Wie ist diese Regel aus  $R$  entstanden?

Eine Regel in  $R'$  entsteht aus einer Regel in  $R$ , indem einige nullbare Variablen gestrichen werden. Es gab also in  $G$  nullbare Variablen  $B_1$  bis  $B_m$ , so dass  $R$  die Regel

$$A \rightarrow A_1 \dots A_{\ell_1} B_1 A_{\ell_1+1} \dots A_{\ell_2} B_2 \dots A_m B_m A_{m+1} \dots A_\ell$$

enthält. ( $m$  kann auch 0 sein, dann war die Regel selbst schon in  $R$ .)

$$\begin{aligned} \text{Dann: } A &\Longrightarrow_G A_1 \dots A_{\ell_1} B_1 A_{\ell_1+1} \dots A_{\ell_2} B_2 \dots A_m B_m A_{m+1} \dots A_\ell \\ &\Longrightarrow_G^* w_1 \dots w_{\ell_1} \varepsilon w_{\ell_1+1} \dots w_{\ell_2} \varepsilon \dots w_m \varepsilon w_{m+1} \dots w_\ell = w \end{aligned}$$

# Elimination von $\varepsilon$ -Regeln: Beispiel

---

| $R :$                               | $R' :$                                     |
|-------------------------------------|--|
| $S \rightarrow ABD$                 | $S \rightarrow ABD \mid AD \mid BD \mid D$ |
| $A \rightarrow ED \mid BB$          | $A \rightarrow ED \mid BB \mid B$          |
| $B \rightarrow AC \mid \varepsilon$ | $B \rightarrow AC \mid A \mid C$           |
| $C \rightarrow \varepsilon$         |  |
| $D \rightarrow d$                   | $D \rightarrow d$                          |
| $E \rightarrow e$                   | $E \rightarrow e$                          |

Für die Regelmenge  $R$  in der linken Spalte sind die Variablen  $A, B, C$  nullbar.

Der obige Algorithmus erzeugt aus  $R$  die rechts aufgeführte Regelmenge  $R'$ .

# Elimination von $\varepsilon$ -Regeln

---

## Beobachtung

- Der Algorithmus lässt nutzlose Variablen zurück, die nicht in Prämissen auftauchen (und deshalb nicht co-erreichbar sind).  
Hier:  $C$ .
- Der Algorithmus lässt nutzlose Regeln zurück.  
Hier:  $B \rightarrow AC \mid C$ .

# Elimination von $\varepsilon$ -Regeln: Beispiel

---

Nach Elimination der nutzlosen Variablen/Regeln

$R :$

$$S \rightarrow ABD$$

$$A \rightarrow ED \mid BB$$

$$B \rightarrow AC \mid \varepsilon$$

$$C \rightarrow \varepsilon$$

$$D \rightarrow d$$

$$E \rightarrow e$$

$R'' :$

$$S \rightarrow ABD \mid AD \mid BD \mid D$$

$$A \rightarrow ED \mid BB \mid B$$

$$B \rightarrow A$$

$$D \rightarrow d$$

$$E \rightarrow e$$

# Elimination von $\varepsilon$ -Regeln

---

**Korollar.**

$$\mathcal{L}_2 \subseteq \mathcal{L}_1$$

Das heißt, jede kontextfreie Sprache ist auch kontextsensitiv

# Elimination von $\varepsilon$ -Regeln

---

## Korollar.

$$\mathcal{L}_2 \subseteq \mathcal{L}_1$$

Das heißt, jede kontextfreie Sprache ist auch kontextsensitiv

**Beweis.** Regeln einer kontextsensitiven Grammatik müssen folgende Form haben:

- entweder  $uAv \rightarrow u\alpha v$   
mit  $u, v, \alpha \in (V \cup T)^*$ ,  $|\alpha| \geq 1$ ,  $A \in V$
- oder  $S \rightarrow \varepsilon$   
und  $S$  kommt in keiner Regelconclusio vor.

Diesen Bedingungen genügt die kontextfreie Grammatik nach Elimination der  $\varepsilon$ -Regeln.

# Elimination von Kettenproduktionen

---

# Elimination von Kettenproduktionen

---

**Definition.** Eine Regel der Form

$$A \rightarrow B \quad \text{mit } A, B \in V$$

heißt **Kettenproduktion**.

**Theorem.** (Kettenproduktionen sind eliminierbar)

Zu jeder cf-Grammatik existiert eine äquivalente cf-Grammatik ohne Kettenproduktionen.

# Elimination von Kettenproduktionen

---

Beweis.

Sei  $G = (V, T, R, S)$  eine kontextfreie Grammatik ohne  $\varepsilon$ -Regeln, außer ggf.  $S \rightarrow \varepsilon$ .

Konstruiere neue Grammatik wie folgt:

1. Für alle

- Variablenpaare  $A, B \in V$ ,  $A \neq B$  mit  $A \Longrightarrow^* B$
- Regeln  $B \rightarrow \alpha \in R$ ,  $\alpha \notin V$

füge zu  $R$  hinzu:

$$A \rightarrow \alpha$$

2. Lösche alle Kettenproduktionen

# Beispiel

---

$G = (\{S, A, B, C\}, \{a, b, c\}, R, S)$  mit

$$R = \left\{ \begin{array}{l} S \rightarrow B|C|AA, \\ A \rightarrow BC|B|a, \\ B \rightarrow b|AA|C, \\ C \rightarrow BC|c \end{array} \right\}$$

# Beispiel

---

$G = (\{S, A, B, C\}, \{a, b, c\}, R, S)$  mit

$$R = \left\{ \begin{array}{l} S \rightarrow B|C|AA, \\ A \rightarrow BC|B|a, \\ B \rightarrow b|AA|C, \\ C \rightarrow BC|c \end{array} \right\}$$

## Kettenproduktionen

$$S \rightarrow B$$

$$S \rightarrow C$$

$$A \rightarrow B$$

$$B \rightarrow C$$

# Beispiel

---

$G = (\{S, A, B, C\}, \{a, b, c\}, R, S)$  mit

$$R = \left\{ \begin{array}{l} S \rightarrow B|C|AA, \\ A \rightarrow BC|B|a, \\ B \rightarrow b|AA|C, \\ C \rightarrow BC|c \end{array} \right\}$$

## Kettenproduktionen

$$S \rightarrow B$$

$$S \Rightarrow^* B$$

$$S \rightarrow C$$

$$S \Rightarrow^* C$$

$$A \rightarrow B$$

$$A \Rightarrow^* B$$

$$B \rightarrow C$$

$$B \Rightarrow^* C$$

$$A \Rightarrow^* C$$

# Beispiel

---

$G = (\{S, A, B, C\}, \{a, b, c\}, R, S)$  mit

$$R = \left\{ \begin{array}{l} S \rightarrow B|C|AA, \\ A \rightarrow BC|B|a, \\ B \rightarrow b|AA|C, \\ C \rightarrow BC|c \end{array} \right\}$$

$$R' = \left\{ \begin{array}{l} S \rightarrow b|AA|BC|c, \\ A \rightarrow BC|b|AA|c|a, \\ B \rightarrow b|AA|BC|c, \\ C \rightarrow BC|c \end{array} \right\}$$

## Kettenproduktionen

$$S \rightarrow B$$

$$S \rightarrow C$$

$$A \rightarrow B$$

$$B \rightarrow C$$

$$S \Rightarrow^* B$$

$$S \Rightarrow^* C$$

$$A \Rightarrow^* B$$

$$B \Rightarrow^* C$$

$$A \Rightarrow^* C$$

# Normalform für cf-Grammatiken

---

**Theorem.** Zu jeder cf-Grammatik existiert eine äquivalente cf-Grammatik

- ohne  $\varepsilon$ -Regeln  
(bis auf  $S \rightarrow \varepsilon$ , falls  $\varepsilon$  zur Sprache gehört;  
in diesem Fall darf  $S$  in keiner Regelconclusio vorkommen),
- ohne nutzlose Symbole,
- ohne Kettenproduktionen,
- so dass für jede Regel  $P \rightarrow Q$  gilt: entweder  $Q \in V^*$  oder  $Q \in T$ .

# Normalform für cf-Grammatiken

---

Beweis.

1. Man teste zunächst, ob  $S$  nullbar ist. Falls ja, dann verwende man  $S_{neu}$  als neues Startsymbol und füge die Regeln  $S_{neu} \rightarrow S \mid \varepsilon$  zum Regelsatz hinzu.
2. Man eliminiere nutzlose Symbole.
3. Man bringe die Grammatik in die Normalform, bei der für jede Regel  $P \rightarrow Q$  gilt: entweder  $Q \in V^*$  oder  $Q \in T$ .
4. Man eliminiere alle  $\varepsilon$ -Regeln außer  $S_{neu} \rightarrow \varepsilon$ .
5. Man eliminiere Kettenproduktionen.
6. Zum Schluss eliminiere man noch einmal alle nutzlosen Symbole (wg. Schritt 3)

# Normalformen

---

## Unterschied: Grammatiktypen und Normalformen

**Gemeinsamkeit:** Sowohl Grammatiktypen als auch Normalformen schränken die Form von Grammatikregeln ein.

## Unterschied:

- Grammatiktypen (rechtslinear, kontextfrei usw.) führen zu **unterschiedlichen Sprachklassen**
- Normalformen führen zu **den selben Sprachklassen**

# Normalformen

---

## Wozu dann Normalformen?

- Weniger Fallunterscheidungen bei Algorithmen, die mit Grammatiken arbeiten.
- Struktur von Grammatiken einfacher zu “durchschauen”

# Normalformen

---

## Wozu dann Normalformen?

- Weniger Fallunterscheidungen bei Algorithmen, die mit Grammatiken arbeiten.
- Struktur von Grammatiken einfacher zu „durchschauen“

## Zwei Normalformen

**Chomsky-Normalform:** Baut auf den Umformungen des vorigen Teils auf.

**Greibach-Normalform:** Ähnlich den rechtslinearen Grammatiken.

# Chomsky-Normalform

---

**Definition.** Eine cf-Grammatik  $G = (V, T, R, S)$  ist in **Chomsky-Normalform (CNF)**, wenn gilt:

- $G$  hat nur Regeln der Form

$$A \rightarrow BC \quad \text{mit } A, B, C \in V \text{ und}$$

$$A \rightarrow a \quad \text{mit } A \in V, a \in T \quad (\text{nicht } \varepsilon!)$$

- Ist  $\varepsilon \in L(G)$ , so darf  $G$  zusätzlich die Regel  $S \rightarrow \varepsilon$  enthalten. In diesem Fall darf  $S$  in keiner Regelconclusio vorkommen.
- $G$  enthält keine nutzlosen Symbole.

# Chomsky-Normalform

---

**Theorem.** Zu jeder cf-Grammatik existiert eine äquivalente cf-Grammatik in Chomsky-Normalform.

# Chomsky-Normalform

---

**Theorem.** Zu jeder cf-Grammatik existiert eine äquivalente cf-Grammatik in Chomsky-Normalform.

Beweis.

**Schritt 1:** Wende auf  $G$  die Umformungen des letzten Abschnitts an.

**Ergebnis:**

- $G$  hat keine nutzlosen Symbole
- Alle Regeln haben die Form
  1.  $A \rightarrow \alpha$  mit  $A \in V$  und  $\alpha \in V^*$ ,  $|\alpha| \geq 2$ , und
  2.  $A \rightarrow a$  mit  $A \in V$ ,  $a \in T$

# Chomsky-Normalform

---

Beweis (Forts.)

**Schritt 2:** Regeln so umformen, dass keine Conclusio eine Länge größer 2 hat.

Ersetze jede Regel

$$A \rightarrow A_1 \dots A_n \text{ mit } A, A_i \in V, n \geq 3$$

durch:

$$\begin{aligned} A &\rightarrow A_1 C_1 \\ C_1 &\rightarrow A_2 C_2 \\ &\vdots \\ C_{n-2} &\rightarrow A_{n-1} A_n \end{aligned}$$

Dabei sind die  $C_i$  neue Variablen in  $V$ .

□

# Beispiel

---

$G = (\{S, A, B, C\}, \{a, b, c\}, R, S) \mapsto G' = (\{S, A, B, C, C_1, C_2, C_3, C_4, C_5\}, \{a, b, c\}, R', S)$

|          |                              |           |                               |
|----------|------------------------------|-----------|-------------------------------|
| $R = \{$ | $S \rightarrow ABCA BCB AB,$ | $R' = \{$ | $S \rightarrow AC_1 BC_3 AB,$ |
|          | $A \rightarrow BCA a$        |           | $A \rightarrow BC_4 a$        |
|          | $B \rightarrow BAB b$        |           | $B \rightarrow BC_5 b$        |
|          | $C \rightarrow c$            |           | $C \rightarrow c$             |
|          | $\}$                         |           | $C_1 \rightarrow BC_2$        |
|          |                              |           | $C_2 \rightarrow CA$          |
|          |                              |           | $C_3 \rightarrow CB$          |
|          |                              |           | $C_4 \rightarrow CA$          |
|          |                              |           | $C_5 \rightarrow AB$          |
|          |                              |           | $\}$                          |

# Greibach-Normalform

---

## Definition.

Eine cf-Grammatik  $G = (V, T, R, S)$  ist in **Greibach-Normalform (GNF)**, wenn gilt:

- $G$  hat nur Regeln der Form
$$A \rightarrow a\alpha \text{ mit } A \in V \text{ und } a \in T \text{ und } \alpha \in V^*$$
- Ist  $\varepsilon \in L(G)$ , so darf  $G$  zusätzlich die Regel  $S \rightarrow \varepsilon$  enthalten. In diesem Fall darf  $S$  in keiner Regelconclusio vorkommen.
- $G$  enthält keine nutzlosen Symbole.

# Übersicht

---

- **Umformung von Grammatiken**

  - Startsymbol nur links

  - Keine nutzlose Symbole

- **Normalform für Regeln**

  - Zu jeder Grammatik  $G$  (beliebigen Typs) existiert eine äquivalente Grammatik  $G'$ , bei der für alle Regeln  $P \rightarrow Q \in R'$  gilt:

    - $Q \in V^*$  und  $P$  beliebig

    - $Q \in T$  und  $P \in V$

- Elimination von  $\varepsilon$ -Regeln

- Elimination von Kettenproduktionen

- **Chomsky Normalform**

- **Greibach Normalform**