

Grundlagen der Theoretischen Informatik

4. Kellerautomaten und kontextfreie Sprachen (III)

17.05.2018

Viorica Sofronie-Stokkermans

e-mail: 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

Umformung von Grammatiken

- **Startsymbol nur links**

Ist das bei einer Grammatik nicht gegeben, kann man es wie folgt erreichen:

- Führe ein neues Startsymbol S_{neu} ein
- Füge die Regel $S_{neu} \rightarrow S$ hinzu.

- **Keine nutzlose Symbole**

Theorem (cf-Grammatik ohne nutzlose Symbole)

Ist $G = (V, T, R, S)$ eine cf-Grammatik mit $L(G) \neq \emptyset$,
dann existiert eine cf-Grammatik $G' = (V', T', R', S')$ mit:

- G' ist äquivalent zu G .
- Jedes $x \in (V \cup T)$ ist erreichbar und co-erreichbar.

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 .

Beweis: Für jedes $t \in T$ 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 $V_t \rightarrow t$.

Elimination von ε -Regeln

Definition (ε -Regel, nullbare Variablen)

Eine Regel der Form $P \rightarrow \varepsilon$ (P eine Variable) heißt ε -Regel.

Eine Variable A heißt **nullbar**, falls $A \Longrightarrow^* \varepsilon$

Theorem (ε -Regeln sind eliminierbar)

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

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

Beweis: 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

$$P \rightarrow \alpha_1 \dots \alpha_n \text{ mit } \begin{array}{l} \alpha_i \in \{\varepsilon, A_i\} \quad \text{falls } A_i \text{ nullbar} \\ \alpha_i = A_i \quad \text{falls } A_i \text{ nicht nullbar} \end{array}$$

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

Elimination von ε -Regeln

Korollar.

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

Das heißt, jede kontextfreie Sprache ist auch kontextsensitiv

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 ε -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

Normalform für cf-Grammatiken

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

- ohne ε -Regeln
(bis auf $S \rightarrow \varepsilon$, falls ε 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 ε -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 **denselben 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 .

□

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.

Pumping-Lemma für kontextfreie Sprachen

Wiederholung

Theorem (Pumping-Lemma für \mathcal{L}_3 -Sprachen)

Sei $L \in \mathbf{RAT}$.

Dann existiert ein $n \in \mathbb{N}$, so dass:

Für alle

$$x \in L \quad \text{mit} \quad |x| \geq n$$

existiert eine Zerlegung

$$x = uvw \quad u, v, w \in \Sigma^*$$

mit

- $|v| \geq 1$
- $|v| < n$
- $uv^m w \in L$ für alle $m \in \mathbb{N}$

Pumping-Lemma für kontextfreie Sprachen

Theorem (Pumping-Lemma für kontextfreie Sprachen)

Sei L kontextfrei.

Dann existiert ein $n \in \mathbb{N}$, so dass:

Für alle

$$z \in L \quad \text{mit} \quad |z| \geq n$$

existiert eine Zerlegung

$$z = uvwxy \quad u, v, w, x, y \in \Sigma^*$$

mit

- $|vx| \geq 1$
- $|vwx| < n$
- $uv^mwx^my \in L$ für alle $m \in \mathbb{N}$

Pumping-Lemma für kontextfreie Sprachen

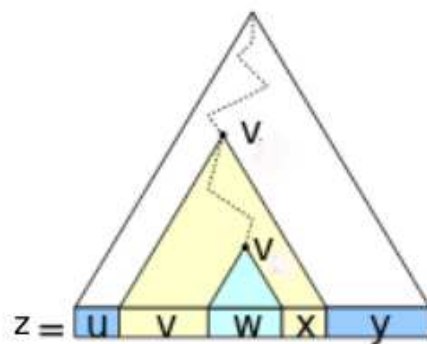
Beweisidee:

Bei der Ableitung eines hinreichend langen Wortes muss es eine Variable geben, die mehr als einmal auftaucht.

Dies führt zu einer Schleife in der Ableitung, die aufgepumpt werden kann.

Pumping-Lemma für kontextfreie Sprachen

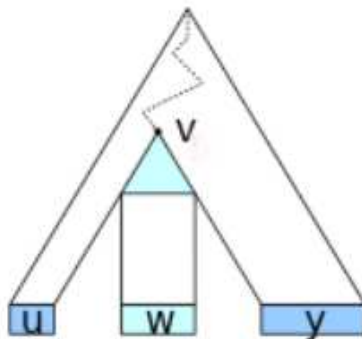
Beweisidee:



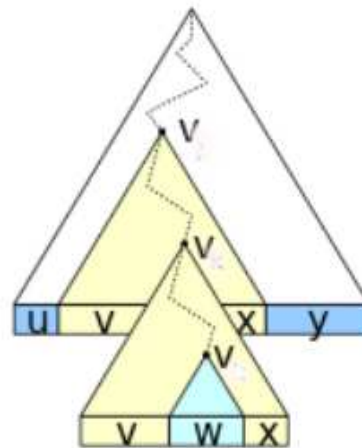
} h

Gegeben: Wort $z \in L$
mit $|z| \geq n$

Ableitungsbaum T für z
mit Höhe $h \geq N$



Erzeugen von uv^0wx^0y



Erzeugen von uv^2wx^2y

Pumping-Lemma für kontextfreie Sprachen

Anwendung des Pumping-Lemmas für cf-Sprachen

Wenn das cf-Pumping-Lemma für eine Sprache nicht gilt,
dann kann sie nicht kontextfrei sein.

Pumping-Lemma für kontextfreie Sprachen

Anwendung des Pumping-Lemmas für cf-Sprachen

Wenn das cf-Pumping-Lemma für eine Sprache nicht gilt, dann kann sie nicht kontextfrei sein.

Beispiel (Sprachen, die nicht kontextfrei sind)

Für folgende Sprachen kann man mit Hilfe des cf-Pumping-Lemmas zeigen, dass sie nicht kontextfrei sind:

- $\{a^p \mid p \text{ prim}\}$
- $\{a^n b^n c^n \mid n \in \mathbb{N}\}$
- $\{zzz \mid z \in \{a, b\}^*\}$.

Pumping-Lemma für kontextfreie Sprachen

$L_1 = \{a^p \mid p \text{ prim}\}$ ist nicht kontextfrei.

Beweis: Wir nehmen an, L_1 sei kontextfrei.

Sei dann n die zugehörige Konstante aus dem Pumping-Lemma.

Wir betrachten das Wort $z = a^p$, wobei p prim und $p \geq n + 2$.

Es muss dann eine Zerlegung $z = uvwxy$ geben, so dass:

$|vx| \geq 1$, $|vwx| < n$, $uv^iwx^iy \in L_1$ für alle $i \geq 0$.

Dann $u = a^{i_1}$, $v = a^{i_2}$, $w = a^{i_3}$, $x = a^{i_4}$, $y = a^{i_5}$ mit

- $i_1 + i_2 + i_3 + i_4 + i_5 = p$
- $i_2 + i_4 \geq 1$, $i_2 + i_3 + i_4 < n$
- $i_1 + mi_2 + i_3 + mi_4 + i_5$ prim für alle $m \geq 0$.

Sei $m = i_1 + i_3 + i_5$. Dann kann $i_1 + mi_2 + i_3 + mi_4 + i_5 = (i_1 + i_3 + i_5)(1 + i_2 + i_4)$ nicht prim sein, da $i_1 + i_3 + i_5 = p - (i_2 + i_4) \geq p - n \geq 2$ und $1 + i_2 + i_4 \geq 2$.

Also $uv^mwx^my \notin L_1$. Widerspruch.

Also kann L_1 nicht kontextfrei sein.