

Automaten und Formale Sprachen

SoSe 2013 in Trier

Henning Fernau

Universität Trier

fernau@uni-trier.de

24. Juli 2013

Automaten und Formale Sprachen

Gesamtübersicht

- Organisatorisches
- Einführung
- Endliche Automaten und reguläre Sprachen
- **Kontextfreie Grammatiken und kontextfreie Sprachen**
- Chomsky-Hierarchie

Kontextfreie Grammatiken und kontextfreie Sprachen

1. Automaten mit unendlichem Speicher
2. Kontextfreie Grammatiken
3. **Normalformen**
4. Nichtkontextfreie Sprachen
5. **Algorithmen für kontextfreie Grammatiken**

Eine **kontextfreie Grammatik** ist ein Quadrupel $G = (\Sigma, N, R, S)$ mit:

- Σ ist das *Terminalalphabet*,
- N ist das *Nonterminalalphabet* (die *Variablenmenge*),
- $R \subset N \times (\Sigma \cup N)^*$ ist das Alphabet der *Regeln* oder *Produktionen*;
übliche Schreibweise: $A \rightarrow v$ anstelle von $(A, v) \in R$, wobei $A \in N$ und $v \in (\Sigma \cup N)^*$ auch *linke Seite* bzw. *rechte Seite* der Regel heißen.
- $S \in N$ ist das *Startsymbol* oder *Anfangszeichen*.

Eine **kontextfreie Grammatik in Chomsky-Normalform** ist ein Quadrupel $G = (\Sigma, N, R, S)$ mit:

- Σ ist das *Terminalalphabet*,
- N ist das *Nonterminalalphabet* (die *Variablenmenge*),
- $R \subset (N \times N^2) \cup (N \times \Sigma)$ ist das Alphabet der *Regeln* oder *Produktionen*;
- $S \in N$ ist das *Startsymbol* oder *Anfangszeichen*.

Ziel der Vorlesung: o.B.d.A.: Chomsky-Normalform

Lemma: Zu der kfG G gibt es eine kfG G' mit $L(G) = L(G')$, bei der Regeln, die Terminalzeichen enthalten, alleinig von der Form $A \rightarrow a$ sind mit $a \in \Sigma$.

Beweis: Es sei $G = (\Sigma, N, R, S)$.

Führe für jedes Terminalzeichen a ein neues Nichtterminalzeichen X_a ein.

$$N' = N \cup \{X_a \mid a \in \Sigma\}.$$

Betrachte Morphismus $h : (N \cup \Sigma)^* \rightarrow (N')^*$, gegeben durch $a \mapsto X_a$ für $a \in \Sigma$ und $A \mapsto A$ für $A \in N$.

$$R' := \{A \rightarrow h(w) \mid A \rightarrow w \in R\} \cup \{X_a \rightarrow a \mid a \in \Sigma\}.$$

Setze $G' := (\Sigma, N', R', S)$.

Einzelheiten zur Korrektheit als Übung. □

Elimination von λ -Regeln, d.i., Regeln der Form $A \rightarrow \lambda$

Lemma: Zu jeder kfG G gibt es eine kfG G' ohne λ -Regeln mit $L(G) \setminus \{\lambda\} = L(G')$.

Beweis: Es sei $G = (\Sigma, N, R, S)$.

$N^\lambda := \{A \in N \mid A \xrightarrow{*}_G \lambda\}$ (Zeichen, die gelöscht werden könnten).

$h : (\Sigma \cup N)^* \rightarrow 2^{(\Sigma \cup N)^*}$ Morphismus, definiert auf $(\Sigma \cup N)$ durch $h(A) = \{A, \lambda\}$, falls $A \in N^\lambda$ und $h(A) = \{A\}$ sonst.

$R' = \{A \rightarrow u \mid A \rightarrow w \in R \wedge u \in h(w) \wedge u \neq \lambda\}$

Setze $G' := (\Sigma, N, R', S)$.

Einzelheiten zur Korrektheit als Übung. □

Wie erhält man G' aus G algorithmisch ?

Problem: $N^\lambda := \{A \in N \mid A \xrightarrow{*}_G \lambda\}$.

Induktive / rekursive Konstruktion:

$N_0^\lambda := \{A \in N \mid A \rightarrow_G \lambda\}$.

$N_k^\lambda := \{A \in N \mid A \rightarrow_G X \wedge X \in (N_{k-1}^\lambda)^*\}$ für $k > 0$.

Beachte: $N_0^\lambda \subseteq N_1^\lambda \subseteq N_2^\lambda \dots$

Lemma: $A \in N^\lambda$ gdw. $A \in \bigcup_{k=0}^{\infty} N_k^\lambda =: N_*^\lambda$ gdw. $A \in N_{\#N}^\lambda$.

Lemma: $A \in N^\lambda$ gdw. $A \in \bigcup_{k=0}^{\infty} N_k^\lambda =: N_*^\lambda$ gdw. $A \in N_{\#N}^\lambda$.

Beweis: Gilt $A \in N^\lambda$, so gibt es Ableitungsfolge

$$A = u_0 \Rightarrow_G u_1 \Rightarrow_G u_2 \dots \Rightarrow_G u_k = \lambda.$$

“Offenbar” ist stets $u_i \in N^*$, genauer gilt: $u_i \in (N_*^\lambda)^*$.

$\leadsto A \in N_*^\lambda$.

Betrachte $M_k^\lambda := N_k^\lambda \setminus N_{k-1}^\lambda$.

In der induktiven Definition gilt: $M_k^\lambda = \emptyset \implies M_{k+i}^\lambda$ für alle $i \geq 0$.

M_k^λ kann höchstens für $k = 0, \dots, \#N - 1$ nicht leer sein.

Also folgt: $A \in N_*^\lambda$ impliziert $A \in N_{\#N}^\lambda$.

Falls $A \in N_{\#N}^\lambda$, so gibt es umgekehrt eine Ableitungsbaum mit A als Wurzel und λ an allen Blättern, der $A \in N^\lambda$ belegt. □

Algorithmus für N^λ

1. $E := N_0^\lambda$

2. WHILE $(\exists A \rightarrow B_1 \dots B_n \in R : A \notin E \wedge (\forall i \in \{1, \dots, n\} : B_i \in E))$ DO:

$E := E \cup \{A\}$

3. $N^\lambda := E$ (Ausgabe)

Alternative Sicht: Finde Wege in einem Graphen mit Knotenmenge 2^N .

Algorithmus für λ -Elimination

1. Berechne N^λ ; dies definiert den Morphismus h .

2. Setze $R' := \emptyset$.

3. FORALL $A \rightarrow w \in R$ DO:

Bestimme $R(A \rightarrow w) = \{A \rightarrow u \mid u \in h(w) \wedge u \neq \lambda\}$.

Hinweis: Gilt $w \in ((N \cup \Sigma) \setminus N^\lambda)^*(N^\lambda((N \cup \Sigma) \setminus N^\lambda)^*)^k$, so gilt $\#R(A \rightarrow w) \leq 2^k$.

$R' := R' \cup R(A \rightarrow w)$.

4. Hinweis: R' ist die neue Regelmenge als (wesentliche) Ausgabe.

Elimination von Kettenregeln, d.i., Regeln der Form $A \rightarrow B$, $B \in N$

Lemma: Zu jeder kfG G gibt es eine kfG G' ohne Kettenregeln mit $L(G) = L(G')$.

Beweis: Es sei $G = (\Sigma, N, R, S)$.

Betrachte $H = (\Sigma, N, R \cap N \times N, S)$.

$N(A) := \{B \in N \mid B \xrightarrow{*}_H A\}$.

$R' = \{B \rightarrow w \mid A \rightarrow w \in R \setminus (N \times N) \wedge B \in N(A)\}$

Setze $G' := (\Sigma, N, R', S)$.

Einzelheiten zur Korrektheit als Übung. □

Algorithmus für $N(A)$ Hinweis: Wegeproblem im zu H gehörigen Graphen.

Algorithmus für Elimination von Kettenregeln

1. Berechne $N(A)$ für alle A .
2. Setze $R' := \emptyset$.
3. FORALL $A \rightarrow w \in R \setminus (N \times N)$ DO:
 Bestimme $R(A) = \{B \rightarrow w \mid B \in N(A)\}$
 $R' := R' \cup R(A)$.
4. Hinweis: R' ist die neue Regelmenge als (wesentliche) Ausgabe.

Chomsky-Normalform

Satz: Zu jeder kfG G gibt es eine kfG G' in Chomsky-Normalform mit $L(G) \setminus \{\lambda\} = L(G')$.

Beweis: Zunächst können wir in der angegebenen Reihenfolge die Lemmata anwenden.

Hinweis: Durch das zweite und dritte Lemma wird die im ersten Lemma sichergestellte Eigenschaft nicht wieder zerstört, und durch das dritte Lemma nicht die im zweiten Lemma hergestellte Eigenschaft.

Dann sind noch möglicherweise noch rechte Regelseiten länger als zwei.

Dies wird durch den Algorithmus auf der nächsten Folie repariert. □

Algorithmus zur Vermeidung zu langer rechter Regelseiten

Eingabe: Regelmenge R (die im Folgenden modifiziert wird)

WHILE $(\exists A \rightarrow w \in R : \ell(w) > 2)$ DO:

1. Stelle $w = Bu$ dar.
2. Erzeuge neues Nichtterminalzeichen X .
3. Ersetze $A \rightarrow w$ durch $A \rightarrow BX$ und $X \rightarrow u$ in R .

Ein Beispiel

Regelmenge $S \rightarrow aSb, S \rightarrow \lambda, S \rightarrow c$.

1. Lemma $\rightsquigarrow S \rightarrow X_aSX_b, S \rightarrow \lambda, S \rightarrow X_c, X_a \rightarrow a, X_b \rightarrow b, X_c \rightarrow c$.

2. Lemma $\rightsquigarrow S \rightarrow X_aSX_b, S \rightarrow X_aX_b, S \rightarrow X_c, X_a \rightarrow a, X_b \rightarrow b, X_c \rightarrow c$.

3. Lemma $\rightsquigarrow S \rightarrow X_aSX_b, S \rightarrow X_aX_b, S \rightarrow c, X_a \rightarrow a, X_b \rightarrow b, X_c \rightarrow c$.

Konstruktion aus Satz $\rightsquigarrow S \rightarrow X_aX, X \rightarrow SX_b, S \rightarrow X_aX_b, S \rightarrow c, X_a \rightarrow a, X_b \rightarrow b, X_c \rightarrow c$.

Satz: Es gibt einen Algorithmus, der zu jeder vorgelegten kfG $G = (\Sigma, N, R, S)$ und jedem $w \in \Sigma^*$ entscheidet, ob $w \in L(G)$ gilt.

Beweis: Wird $w = \lambda$ angefragt, so berechnen wir N^λ wie im Beweis zum 2. Lemma. Es gilt: $\lambda \in L(G)$ gdw. $S \in N^\lambda$.

Wird $w \neq \lambda$ angefragt, so überführen wir G in eine Chomky-Normalform kfG G' mit der Eigenschaft $w \in L(G)$ gdw. $w \in L(G')$.

Da G' keine λ -Regeln enthält, kann man “ $w \in L(G)$?” durch Berechnen aller aus G' ableitbaren Satzformen der Länge höchstens $\ell(w)$ beantworten. □

Cocke, Younger und Kasami haben gezeigt, wie man die Komplexität des beschriebenen Verfahrens durch **dynamisches Programmieren** erheblich vermindern kann.

Das Verfahren von Cocke, Younger und Kasami (*CYK-Algorithmus*)

Satz: Ist eine kfG G in Chomsky-Normalform fixiert, so lässt sich die Frage “ $w \in L(G)$?” in einer Zeit beantworten, die sich durch ein kubisches Polynom in $\ell(w)$ abschätzen lässt.

Beweis: Betrachte $w = a_1 \dots a_n$.

Lege 2–dimensionale dreiecksförmige Tabelle T an mit Nonterminalmengen als Einträgen so dass $T[i, j] = \{A \in N \mid A \xrightarrow{*} a_i \dots a_j\}$. T kann man wie folgt “von oben nach unten” berechnen:

$$T[i, i] = \{A \in N \mid A \rightarrow a_i \in R\}$$

$$T[i, j] = \{A \in N \mid A \rightarrow BC \in R \wedge \exists i \leq k < j : B \in T[i, k], C \in T[k + 1, j]\}$$

Die CYK-Tabelle

a_1	a_2	\dots	a_n
$T[1, 1]$	$T[2, 2]$	\dots	$T[n, n]$
$T[1, 2]$	$T[2, 3]$	\dots	
\vdots	\vdots		
$T[1, n-1]$	$T[2, n-1]$		
$T[1, n]$			

$a_1 \dots a_n \in L(G)$ gdw. das Startsymbol erscheint in $T[1, n]$.

Der CYK-Algorithmus

FOR $j = 1, \dots, n$

$T[i, i] = \{A \in N \mid A \rightarrow \alpha_i \in R\}$

FOR $j = 1, \dots, n - 1$

FOR $i = 1, \dots, n - j$

$T[i, j + i] = \emptyset$

FOR $k = i, \dots, j + i - 1$

IF $\exists A \rightarrow BC \in R : B \in T[i, k], C \in T[k + 1, j + i]$

THEN $T[i, j + i] = T[i, j + i] \cup \{A\}$

Ein Beispiel

$S \rightarrow AX, X \rightarrow SB, S \rightarrow c, A \rightarrow a, B \rightarrow b$ erzeugt $L = \{a^n cb^n \mid n \geq 0\}$.

a	a	c	b	b
A	A	S	B	B
	S	X		
	X			
S				

Satz: **KF** ist unter Vereinigung abgeschlossen.

Beweis: L_1, L_2 seien durch kfGs G_1 bzw. G_2 spezifiziert.

Genauer sei, für $i = 1, 2$, $G_i = (\Sigma_i, N_i, R_i, S_i)$. O.E. nehmen wir an: $N_1 \cap N_2 = \emptyset$.

Betrachte $G = (\Sigma_1 \cup \Sigma_2, N_1 \cup N_2 \cup \{S\}, R_1 \cup R_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}, S)$ mit $S \notin N_1 \cup N_2$.

Beh.: $L(G) = L(G_1) \cup L(G_2)$.

$w \in L(G)$ gdw. es gibt Ableitungsfolge $S \Rightarrow S_i \xrightarrow{*}_G w$ ($S \notin N_1 \cup N_2$)

gdw. $S_1 \xrightarrow{*}_{G_1} w$ oder $S_2 \xrightarrow{*}_{G_2} w$ ($N_1 \cap N_2 = \emptyset$) gdw. $w \in L(G_1)$ oder $w \in L(G_2)$

gdw. $w \in L(G_1) \cup L(G_2)$. □

Satz: **KF** ist unter Konkatenation abgeschlossen.

Beweis: L_1, L_2 seien durch kfGs G_1 bzw. G_2 spezifiziert.

Genauer sei, für $i = 1, 2$, $G_i = (\Sigma_i, N_i, R_i, S_i)$. O.E. nehmen wir an: $N_1 \cap N_2 = \emptyset$.

Betrachte $G = (\Sigma_1 \cup \Sigma_2, N_1 \cup N_2 \cup \{S\}, R_1 \cup R_2 \cup \{S \rightarrow S_1 S_2\}, S)$ mit $S \notin N_1 \cup N_2$.

Beh.: $L(G) = L(G_1)L(G_2)$.

$w \in L(G)$ gdw. es gibt Ableitungsfolge $S \Rightarrow S_1 S_2 \xrightarrow{*}_G w$ ($S \notin N_1 \cup N_2$)

gdw. $S_1 \xrightarrow{*}_{G_1} u$ und $S_2 \xrightarrow{*}_{G_2} v$ mit $uv = w$ ($N_1 \cap N_2 = \emptyset$) gdw. $u \in L(G_1)$ und $v \in L(G_2)$

gdw. $w \in L(G_1)L(G_2)$ mit $w = uv$. □

Satz: **KF** ist unter Kleene Stern abgeschlossen.

Beweis: L sei durch kfG $G = (\Sigma, N, R, S)$ spezifiziert.

Betrachte $G' = (\Sigma, N \cup \{\hat{S}\}, R \cup \{\hat{S} \rightarrow \hat{S}S, \hat{S} \rightarrow \lambda\}, \hat{S})$ mit $\hat{S} \notin N$.

Beh.: $L(G') = L(G)^*$.

$w \in L(G')$ gdw. es gibt Ableitungsfolge $\hat{S} \Rightarrow^k \hat{S}S^k \Rightarrow S^k \xrightarrow{*}_{G'} w_1 \dots w_k = w$

gdw. $S^k \xrightarrow{*}_G w_1 \dots w_k = w$ (denn $\hat{S} \notin N$)

gdw. $S \xrightarrow{*}_G w_i$ für $1 \leq i \leq k$

gdw. $w \in L(G)^*$. □

Satz: **KF** ist unter Durchschnitt mit regulären Sprachen abgeschlossen.

Beweis: L sei durch Kellerautomaten $A = (Q, \Sigma, \Gamma, q_0, \Delta, F)$ akzeptiert.

R sei akzeptiert durch NEA $A' = (Q', \Sigma, \delta, q'_0, F')$.

Betrachte Produktkellerautomat $\hat{A} = (\hat{Q}, \Sigma, \Gamma, \hat{q}_0, \hat{\Delta}, \hat{F})$ mit

$$\hat{Q} = Q \times Q', \hat{q}_0 = (q_0, q'_0), \hat{F} = F \times F'$$

Für $a \in \Sigma$: $((q, q'), a, X), ((p, p'), Y)) \in \hat{\Delta}$ gdw. $((q, a, X), (p, Y)) \in \Delta$ und $(q', a, p') \in \delta$.

Für $a = \lambda$: $((q, q'), a, X), ((p, p'), Y)) \in \hat{\Delta}$ gdw. $((q, a, X), (p, Y)) \in \Delta$ und $q' = p'$.

Beh.: $L(\hat{A}) = L(A) \cap L(A')$. □