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 kontrextfreie Sprachen

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

Ein **Kellerautomat**, engl. Pushdown automaton, ist ein Sextupel $A = (Q, \Sigma, \Gamma, q_0, \Delta, F)$:

- Q ist das *Zustandsalphabet*,
- Σ ist das *Eingabealphabet*,
- Γ ist das *Kelleralphabet*,
- $q_0 \in Q$ ist der *Startzustand (Anfangszustand)*,
- $\Delta \subset (Q \times (\Sigma \cup \{\lambda\}) \times \Gamma^*) \times (Q \times \Gamma^*)$ ist die *Übergangsrelation*,
- $F \subseteq Q$ ist die *Endzustandsmenge*.

Konfigurationsübergänge

Konfiguration: $C \in (Q \times \Sigma^* \times \Gamma^*)$
Erweiterte NEA-Konfiguration, die dritte Komponente modelliert den Kellerinhalt.
Für zwei Konfigurationen C_1 und C_2 mit $C_i = (q_i, w_i, \gamma_i)$ definieren wir $C_1 \vdash C_2$ *(Einschrittkonfigurationsübergang)* gdw.
es gibt Transitionen $((q_1, a, \alpha_1), (q_2, \alpha_2))$ mit $w_1 = aw_2, \gamma_i = \alpha_i\beta$ für ein $\beta \in \Gamma^*$.
Bei dieser Formalisierung steht also der “erreichbare” Teil des Kellers “links.”
α_1 wird durch α_2 ersetzt; also $\texttt{pop}(\alpha_1)$ [mit Test]; $\texttt{push}(\alpha_2)$.
Beachte: $a \in \Sigma \cup \{\lambda\}$!

$$L(A) = \{w \in \Sigma^* \mid \exists f \in F : (q_0, w, \lambda) \vdash^* (f, \lambda, \lambda)\}$$

Sprachfamilie: **PDA**

Ein Beispiel Betrachte

$$A = (\{q_0, q, f\}, \{a, b\}, \{a\}, \Delta, q_0, \{q_0, f\})$$

mit folgenden Übergängen:

- $((q_0, a, \lambda), (q, a))$
- $((q, a, \lambda), (q, a))$
- $((q, b, a), (f, \lambda))$
- $((f, b, a), (f, \lambda))$

Lemma: $L(A) = \{a^n b^n \mid n \geq 0\} =: L$.

Beweis: Induktion über $k > 0$ zeigt $a^k b^k \in L(A)$: Betrachte hierzu

$$
\begin{aligned}
(q_0, a^k, \lambda) &\vdash (q, a^{k-1}, a) \vdash^{k-1} (q, \lambda, a^k) \text{ und} \\
(q, b^k, a^k) &\vdash (f, b^{k-1}, a^{k-1}) \vdash^{k-1} (f, \lambda, \lambda)
\end{aligned}
$$

q_0 und f akzeptieren $\rightsquigarrow L \subseteq L(A)$.

Betrachte umgekehrt $w \in L(A)$.

Induktion über die Länge k einer Ableitung liefert:

Beh. 1: $(q, w, \lambda) \vdash^* (q, \lambda, x) \Rightarrow w = x \in \{a\}^*$

Bew.: $k = 0$: $(q, w, \lambda) \vdash^0 (q, \lambda, x) \rightsquigarrow w = x = \lambda$ ✓

$k > 0$: $(q, w, \lambda)(\vdash \circ \vdash^{k-1})(q, \lambda, x)$ gdw. $(q, w, \lambda) \vdash (q', w', x') \vdash^{k-1} (q, \lambda, x)$.

Inspektion der Übergänge ergibt: $q' = q$ oder $q' = f$.

Im zweiten Fall gibt es keinen Rückweg nach q. $\rightsquigarrow q' = q$ und $aw' = w$, $x' = a$.

IH liefert: $(q, w', a) \vdash^{k-1} (q, \lambda, x) \Rightarrow w' \in \{a\}^*$ und $x = aw'$. $\rightsquigarrow x = w \in \{a\}^*$.

Ähnlich sieht man:

Beh. 2: $(f, w, x) \vdash^* (f, \lambda, y) \Rightarrow \exists k\, w = b^k$ und $x = a^k y$.

Diskutiere $w \in L(A)$. Falls $w \neq \lambda$, so $w = aw'$ und

$$(q_0, aw', \lambda) \vdash (q, w', a)$$

(durch Betrachten der möglichen Übergänge).
Eine weitere Betrachtung offenbart: $w \in L(A)$ heißt $w' = ubv$ mit

$$(q, w', a) \vdash^* (q, bv, au) \vdash (f, v, u) \vdash^* (f, \lambda, \lambda)$$

Gemäß Beh. 1 bedeutet dies $u = a^k$ (für irgendein $k \geq 0$) und
gemäß Beh. 2 $v = b^k$.
$\rightsquigarrow w = a^{k+1}b^{k+1} \rightsquigarrow w \in L$.

Palindrome Betrachte $A = (\{q, f\}, \{a, b\}, \{a, b\}, \Delta, q, \{f\})$ mit Transitionen

- $((q, a, \lambda), (q, a))$
- $((q, b, \lambda), (q, b))$
- $((q, \lambda, \lambda), (f, \lambda))$
- $((f, a, a), (f, \lambda))$
- $((f, b, b), (f, \lambda))$

Lemma: $L(A) = \{ww^R \mid w \in \{a, b\}^*\}$

Frage: Kellerautomat / Zählerautomat für $\{w \mid w \in \{a, b\}^* \wedge w = w^R\}$?

Kellerautomaten erzeugen kontextfreie Sprachen

Sei $G = (\Sigma, N, R, S)$ eine kfG. Betrachte folgende Transitionen:

- $((s, \lambda, \lambda), (f, S))$,
- für jede Regel $C \to w$: $((f, \lambda, C), (f, w))$,
- für jedes Terminalzeichen a: $((f, a, a), (f, \lambda))$.

f ist der einzige Endzustand und s der Startzustand.

$\leadsto$ Satz: **KF** $\subseteq$ **PDA.**

Die Konstruktion am Beispiel

$G = (\{a, b\}, \{S\}, R, S)$ mit den Regeln $r_1 = S \to aSb$ und $r_2 = S \to \lambda$.

Der entsprechende Kellerautomat hat folgende Regeln:
$((s, \lambda, \lambda), (f, S))$, $((f, \lambda, S), (f, aSb))$, $((f, \lambda, S), (f, \lambda))$, $((f, a, a), (f, \lambda))$, $((f, b, b), (f, \lambda))$.

Die Ableitung $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$ wird wie folgt simuliert:

1. Initialisierungsphase:

$(s, aabb, \lambda) \vdash (f, aabb, S)$

2. Simulieren der kfG und 3. Überprüfen der Eingabe:

$(f, aabb, S) \vdash (f, aabb, aSb) \vdash (f, abb, Sb) \vdash (f, abb, aSbb)$
$\vdash (f, bb, Sbb) \vdash (f, bb, bb) \vdash (f, b, b) \vdash (f, \lambda, \lambda)$

Normalformen

Satz: (1) O.E.: Akzeptierung nur durch Endzustände:

$$L_{fs}(A) = \{w \in \Sigma^* \mid \exists f \in F, x \in \Gamma^* : (q_0, w, \lambda) \vdash^* (f, \lambda, x)\}$$

Satz: (2) O.E.: Akzeptierung nur durch leeren Keller:

$$L_{es}(A) = \{w \in \Sigma^* \mid \exists q \in Q : (q_0, w, \lambda) \vdash^+ (q, \lambda, \lambda)\}$$

Satz: (3) O.E.: Akzeptierung durch leeren Keller und mit $|Q| = 1$.

Satz: (4) O.E.: Alle Normalformen mit *Kellerbodenzeichen* $\lhd$, d.h.,

$$L_{es}(A) = \{w \in \Sigma^* \mid \exists q \in Q : (q_0, w, \lhd) \vdash^* (q, \lambda, \lambda)\}$$

Beweisideen

Satz (4), Grundversion: (a) Jede $L \in$ **PDA** ist so beschreibbar: lösche zum Schluss $\triangleleft$ und gehe dann in den einzigen akzeptierenden Zustand.
(b) Rückrichtung: Erzeuge Kellerbodenzeichen ganz am Anfang.
Satz (4) mit leerem Keller geht ähnlich.

Satz (1): (a) Jede $L \in$ **PDA** ist so beschreibbar: Der Leerkellertest ist mit dem Kellerbodenzeichen implementierbar.
(b) Rückrichtung: Lösche Kellerinhalt, sofern Endzustand erreicht ist und Terminieren nichtdeterministisch entschieden wird.

Satz (2): (a) Jede $L \in$ **PDA** ist so beschreibbar: führe Kellerbodenzeichen ein und gestatte seine Löschung nur, falls Endzustand erreicht.
(b) Rückrichtung: Merke in endlicher Kontrolle, ob wenigstens ein Schritt gemacht wurde.

Satz (3): Speichere endliche Kontrolle auf Keller $\rightsquigarrow$ neues Kelleralphabet $Q \times \Gamma \times Q$;
die Semantik von $[p, x, p'] \in Q \times \Gamma \times Q$ die folgende:
Der zu simulierende Kellerautomat ist im Zustand p, x steht oben auf dem Keller, und p' ist der angestrebte "Zielzustand", der nach Auskellern von x erreicht werden soll. Konkret:
(i) simuliere $((q_0, a, b), (q_1, cd))$ durch $((s, a, [q_0, b, q_2]), (s, [q_1, c, q][q, d, q_2]))$;
(ii) simuliere $((q_0, a, b), (q_1, \lambda))$ durch $((s, a, [q_0, b, q_1]), (s, \lambda))$;
(iii) starte mit $[q_0, \triangleleft, q]$ für bel. Zustand q (es-Version) oder Endzustand q.

kfGs erzeugen PDA-Sprachen

Erinnerung an Normalformensatz:

$L \in$ **PDA** gdw. L wird durch Kellerautomaten mit Kellerbodensymbol $\triangleleft$ und nur einem Zustand per Leerkellerakzeptanz beschrieben.

Damit wird die Simulation einfach:

$((s, a, A), (s, \lambda)) \rightsquigarrow$ Regel $A \to a$ (inbesondere für Kellerbodenzeichen).

$((s, a, A), (s, B_1 \ldots B_k)) \rightsquigarrow$ Regel $A \to aB_1 \ldots B_k$.

Startsymbol ist neues Zeichen S mit Regeln $S \to aX$ für Kellerzeichen X mit Transition $((s, a, \lambda), (s, X)$.

(Hierbei weiterhin benutzt:

(a) O.E. werden beim Kellerautomat nur (und stets) das oberste Kellerzeichen beachtet.

(b) O.E. sind Eingabe- und Kelleralphabet disjunkt.)

$\rightsquigarrow$ Satz: **KF** $=$ **PDA.**

Hinweis: ähnlich sog. Greibach-Normalform (“lediglich” Zusatz: $a \in \Sigma$)

Die Konstruktionen an einem Beispiel

Welche Sprache akzeptiert der folgende Kellerautomat mit f als Endzustand und s als Startzustand ?

$((s, a, \lambda), (s, A))$
$((s, b, \lambda), (f, \lambda))$
$((f, a, AAA), (f, \lambda))$

Die Konstruktionen an einem Beispiel

Welche Sprache akzeptiert der folgende Kellerautomat A mit f als Endzustand und s als Startzustand ?

$((s, a, \lambda), (s, A))$
$((s, b, \lambda), (f, \lambda))$
$((f, a, AAA), (f, \lambda))$

$L(A) = \{a^{3n}ba^n \mid n \in \mathbb{N}\}$.

Beispielakzeptierung:
$(s, aaaba, \lambda) \vdash (s, aaba, A) \vdash (s, aba, AA) \vdash (s, ba, AAA) \vdash (f, a, AAA) \vdash (f, \lambda, \lambda)$.

Die Konstruktionen an einem Beispiel: Der Weg zur kfG

Was ist “falsch” ?

(a) Einlesen von ganzen Wörtern vom Keller
(b) kein Kellerbodenzeichen / Nichtbeachten des Kellerinhalts
(c) falsches Akzeptanzkriterium
(d) mehr als ein Zustand

Die Konstruktionen an einem Beispiel:
(a) Einlesen von ganzen Wörtern vom Keller

Lösung: Führe Zwischenzustände ein, die nicht akzeptieren:

$((s, a, \lambda), (s, A))$
$((s, b, \lambda), (f, \lambda))$
$((f, \lambda, A), (f_1, \lambda))$
$((f_1, \lambda, A), (f_2, \lambda))$
$((f_2, a, A), (f, \lambda))$

Beispielakzeptierung: $(s, aaaba, \lambda) \vdash (s, aaba, A) \vdash (s, aba, AA) \vdash (s, ba, AAA) \vdash (f, a, AAA) \vdash (f_1, a, AA) \vdash (f_2, a, A) \vdash (f, \lambda, \lambda)$.

Die Konstruktionen an einem Beispiel:
(b) kein Kellerbodenzeichen (sowie (c) falsches Akzeptanzkriterium)

$[((s_0, \lambda, \lambda), (s, \triangleleft)]$
$((s, a, \triangleleft), (s, A\triangleleft))$
$((s, a, A), (s, AA))$
$((s, b, \triangleleft), (f, \triangleleft))$
$((s, b, A), (f, A))$
$((f, \lambda, A), (f_1, \lambda))$
$((f_1, \lambda, A), (f_2, \lambda))$
$((f_2, a, A), (f, \lambda))$
$((f, \lambda, \triangleleft), (s_f, \lambda))$

Beispielakzeptierung: $[(s_0, aaaba, \lambda) \vdash](s, aaaba, \triangleleft) \vdash (s, aaba, A\triangleleft) \vdash (s, aba, AA\triangleleft) \vdash (s, ba, AAA\triangleleft) \vdash (f, a, AAA\triangleleft) \vdash (f_1, a, AA\triangleleft) \vdash (f_2, a, A\triangleleft) \vdash (f, \lambda, \triangleleft) \vdash (s_f, \lambda, \lambda)$.

[] Klammern deuten an: Kann bei expliziter Einführung von $\triangleleft$ als KBZ unbeachtet bleiben (Keller startet mit KBZ).

Die Konstruktionen an einem Beispiel:

(d) mehr als ein Zustand; im Folgenden sind r, r' beliebige Zustände des vorigen Automaten.

(Wir nehmen explizites KBZ an, dieses heiße $\triangleleft$.)

$((q, \lambda, \triangleleft), (q, [s, \triangleleft, s_f]\triangleleft)$
$((q, a, [s, \triangleleft, r]), (q, [s, A, r'][r', \triangleleft, r]))$
$((q, a, [s, A, r]), (q, [s, A, r'][r', A, r]))$
$((q, b, [s, \triangleleft, r]), (q, [f, \triangleleft, r]))$
$((q, b, [s, A, r]), (q, [f, A, r]))$
$((q, \lambda, [f, A, f_1]), (q, \lambda))$
$((q, \lambda, [f_1, A, f_2]), (q, \lambda))$
$((q, a, [f_2, A, f]), (q, \lambda))$
$((q, \lambda, [f, \triangleleft, s_f]), (q, \lambda))$
$((q, \lambda, \triangleleft), (q, \lambda))$

Beispielakzeptierung:

$(q, aaaba, \triangleleft) \vdash (q, aaaba, [s, \triangleleft, s_f]\triangleleft)$
$\vdash (q, aaba, [s, A, f][f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, aba, [s, A, f_2][f_2, A, f][f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, ba, [s, A, f_1][f_1, A, f_2][f_2, A, f][f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, a, [f, A, f_1][f_1, A, f_2][f_2, A, f][f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, a, [f_1, A, f_2][f_2, A, f][f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, a, [f_2, A, f][f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, \lambda, [f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, \lambda, \triangleleft)$
$\vdash (q, \lambda, \lambda).$

Die Konstruktionen an einem Beispiel:

Die zugehörige Grammatik; im Folgenden sind r, r' beliebige Zustände des vorvorigen Automaten. (Wir können auf das Kellerbodenzeichen verzichten.)

$S \to [s, \lhd, s_f]$

$[s, \lhd, r] \to a[s, A, r'][r', \lhd, r]$

$[s, A, r] \to a[s, A, r'][r', A, r]$

$[s, \lhd, r] \to b[f, \lhd, r]$

$[s, A, r] \to b[f, A, r]$

$[f, A, f_1] \to a$

$[f_1, A, f_2] \to a$

$[f_2, A, f] \to a$

$[f, \lhd, s_f] \to \lambda$

Beispiellinksableitung:

$S \Rightarrow [s, \lhd, s_f]$

$\Rightarrow a[s, A, f][f, \lhd, s_f]$

$\Rightarrow aa[s, A, f_2][f_2, A, f][f, \lhd, s_f]$

$\Rightarrow aaa[s, A, f_1][f_1, A, f_2][f_2, A, f][f, \lhd, s_f]$

$\Rightarrow aaab[f, A, f_1][f_1, A, f_2][f_2, A, f][f, \lhd, s_f]$

$\Rightarrow aaab[f_1, A, f_2][f_2, A, f][f, \lhd, s_f]$

$\Rightarrow aaab[f_2, A, f][f, \lhd, s_f]$

$\Rightarrow aaaba[f, \lhd, s_f]$

$\Rightarrow aaaba.$

Die Konstruktionen an einem Beispiel:

Links die Arbeitsweise des Normalform-Kellerautomaten,

rechts die der daraus konstruierten Grammatik.

Beispielakzeptierung:

$(q, aaaba, \triangleleft)$
$\vdash (q, aaaba, [s, \triangleleft, s_f]\triangleleft)$
$\vdash (q, aaba, [s, A, f][f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, aba, [s, A, f_2][f_2, A, f][f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, ba, [s, A, f_1][f_1, A, f_2][f_2, A, f][f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, a, [f, A, f_1][f_1, A, f_2][f_2, A, f][f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, a, [f_1, A, f_2][f_2, A, f][f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, a, [f_2, A, f][f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, \lambda, [f, \triangleleft, s_f]\triangleleft)$
$\vdash (q, \lambda, \triangleleft)$
$\vdash (q, \lambda, \lambda)$.

Beispiellinksableitung:

S
$\Rightarrow [s, \triangleleft, s_f]$
$\Rightarrow a[s, A, f][f, \triangleleft, s_f]$
$\Rightarrow aa[s, A, f_2][f_2, A, f][f, \triangleleft, s_f]$
$\Rightarrow aaa[s, A, f_1][f_1, A, f_2][f_2, A, f][f, \triangleleft, s_f]$
$\Rightarrow aaab[f, A, f_1][f_1, A, f_2][f_2, A, f][f, \triangleleft, s_f]$
$\Rightarrow aaab[f_1, A, f_2][f_2, A, f][f, \triangleleft, s_f]$
$\Rightarrow aaab[f_2, A, f][f, \triangleleft, s_f]$
$\Rightarrow aaaba[f, \triangleleft, s_f]$

$\Rightarrow aaaba$.