

Automaten und Formale Sprachen

SoSe 2007 in Trier

Henning Fernau

Universität Trier

fernau@informatik.uni-trier.de

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. **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*.

Ein Wort über dem *Gesamtalphabet* $(\Sigma \cup N)$ heißt auch *Satzform*.

Ein Beispiel

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

a, b : die Terminalzeichen

S : die Nonterminalzeichen; hier gleichzeitig das Startzeichen

$\{S, a, b\}$: das Gesamtalphabet

$R = \{r_1, r_2\}$: die Regelmeng

$abS, aaSbb, aSaSa$: mögliche Satzformen, **nicht alle ableitbar in G**

Der **Ableitungsmechanismus** einer kontextfreien Grammatik:

1-Schritt-Ableitungsrelation \Rightarrow_G zwischen zwei Satzformen u, v einer kfG G :

$u \Rightarrow_G v$ (manchmal kurz $u \Rightarrow v$) gdw.

es gibt Regel $A \rightarrow y$, sodass u und v wie folgt zerlegt werden können: $u = xAz$ und $v = xyz$; hierbei sind x und z wiederum Satzformen.

Etwas formaler, ggb. $G = (\Sigma, N, R, S)$:

$$\forall u, v \in (\Sigma \cup N)^* : u \Rightarrow_G v \iff$$

$$(\exists x, z \in (\Sigma \cup N)^* \exists (A \rightarrow y) \in R : u = xAz \wedge v = xyz)$$

\Rightarrow^n : n -Schritt-Ableitungsrelation.

\Rightarrow^* : Ableitung mit beliebig vielen Schritten.

Die von einer kfG G *erzeugte* oder *abgeleitete* Sprache ist gegeben durch:

$$L(G) := \{w \in \Sigma^* \mid S \xRightarrow{*} w\}.$$

Bequeme Schreibweise einer *Ableitung(sfolge)*:

$$u_0 \Rightarrow u_1 \Rightarrow u_2 \Rightarrow u_3$$

Gilt $u \xRightarrow{*} v$, so gilt für ein k : $u \Rightarrow^k v$, bezeugt durch die Ableitungsfolge $u = u_0 \Rightarrow u_1 \Rightarrow \dots \Rightarrow u_{k-1} \Rightarrow u_k = v$. Dieses k heißt auch *Länge der Ableitung*.

KF: Familie der *kontextfreien Sprachen* (der durch kontextfreie Grammatiken erzeugbaren Sprachen).

Ein Beispiel

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

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

Beweis: Offenbar gilt $\lambda \in L(G)$ und $\lambda = a^0 b^0$, sodass wir diesen Fall hinfort außer Acht lassen können.

Per Induktion beweisen wir die folgende Behauptung:

Die einzigen in $k > 0$ Ableitungsschritten ableitbaren Wörter sind $a^k S b^k$ und $a^{k-1} b^{k-1}$.

✓ für $k = 1$.

IH: Die Behauptung gilt für $k > 0$. Betrachte eine Ableitung der Länge $k + 1$: $S \Rightarrow^k v \Rightarrow w$.

IH $\rightsquigarrow v = a^k S b^k$ (Die Möglichkeit $v = a^{k-1} b^{k-1}$ gestattet keine Fortführung.).

Anwendung von r_1 liefert $v \Rightarrow a^{k+1} S b^{k+1}$.

Anwendung von r_2 liefert $v \Rightarrow a^k b^k$.

Dies war zu zeigen.

Rechtslinear und regulär

Eine kfG heißt *rechtslinear* gdw. alle rechten Regelseiten haben die Form $z\bar{A}$ oder z , wobei z ein Terminalwort ist und \bar{A} ein Nichtterminalzeichen.

Satz: L ist regulär gdw. L ist durch rechtslineare kfG erzeugbar.

Beweis: Es sei $A = (Q, \Sigma, \delta, s, F)$ ein DEA (mit $\Sigma \cap Q = \emptyset$). Für $\delta(q, a) = q'$ wird die Regel $q \rightarrow aq'$ in die simulierende Grammatik $G = (\Sigma, Q, R, s)$ aufgenommen; die einzigen anderen Regeln sind $q \rightarrow \lambda$ für alle $q \in F$.

Umgekehrt können die Nichtterminale einer rechtslinearen Grammatik als Zustände eines NEAs mit "Zustandsübergängen mit beliebigen Wörtern" gedeutet werden.

Folgerung: $\mathbf{REG} \subset \mathbf{KF}$.

Eine Anwendung von KfG:

Beschreibung der Syntax (grammatisches Gerüst) natürlicher Sprachen

Satz \rightarrow NP VP

NP \rightarrow Artikel Nomen

VP \rightarrow Verb

Artikel \rightarrow "der"

Artikel \rightarrow "die"

Nomen \rightarrow "Hund"

Nomen \rightarrow "Hunde"

Nomen \rightarrow "Katze"

Verb \rightarrow "beißen"

Satz \Rightarrow NP VP \Rightarrow^2 Artikel Nomen Verb \Rightarrow^3 "die" "Hunde" "beißen"

Satz \Rightarrow NP VP \Rightarrow^2 Artikel Nomen Verb \Rightarrow^3 "die" "Katze" "beißen"

Satz \Rightarrow NP VP \Rightarrow^2 Artikel Nomen Verb \Rightarrow^3 "der" "Katze" "beißen"

Eine Anwendung von KfG: Beschreibung arithmetischer Ausdrücke

$\Sigma = \{v, -, +, *, /, (,)\}$, $N = \{E\}$.

Die Regeln seien die folgenden:

$E \rightarrow (E)$

$E \rightarrow -(E)$

$E \rightarrow (E + E)$

$E \rightarrow (E - E)$

$E \rightarrow (E * E)$

$E \rightarrow (E/E)$

$E \rightarrow v$

Beispielableitung:

$E \Rightarrow -(E)$

$\Rightarrow -(E + E)$

$\Rightarrow -((E * E) + E)$

$\Rightarrow -((-E) * E + E)$

$\Rightarrow -((-E) * E + (E - E))$

$\xRightarrow{*} -((-v) * v + (v - v))$

Hinweis: Der *Scanner*, ein endlicher Automat, produziert idealerweise als Ausgabe die Eingabe für den *Parser* zwecks *syntaktischer Analyse* eines Programmtextes.

“Nebensächlichkeiten” wie Zahlen oder Zahlenvariablen werden in einen Statthalter v übersetzt.

Ein Wort hat viele Ableitungen

Linksableitung:

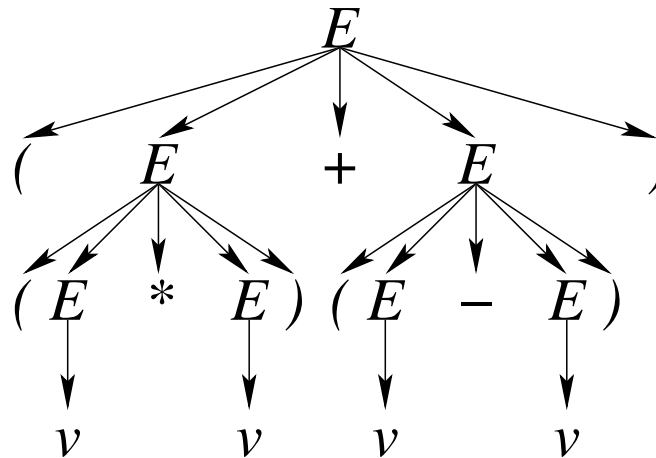
$$\begin{aligned} E &\Rightarrow -(E) \\ &\Rightarrow -(E + E) \\ &\Rightarrow -((E * E) + E) \\ &\Rightarrow -((-E) * E + E) \\ &\Rightarrow -((-v) * E + E) \\ &\Rightarrow -((-v) * v + E) \\ &\Rightarrow -((-v) * v + (E - E)) \\ &\Rightarrow -((-v) * v + (v - E)) \\ &\Rightarrow -((-v) * v + (v - v)) \end{aligned}$$

Rechtsableitung:

$$\begin{aligned} E &\Rightarrow -(E) \\ &\Rightarrow -(E + E) \\ &\Rightarrow -(E + (E - E)) \\ &\Rightarrow -(E + (E - v)) \\ &\Rightarrow -(E + (v - v)) \\ &\Rightarrow -((E * E) + (v - v)) \\ &\Rightarrow -((E * v) + (v - v)) \\ &\Rightarrow -((-E) * v + (v - v)) \\ &\Rightarrow -((-v) * v + (v - v)) \end{aligned}$$

Ein Syntaxbaum (oder *Ableitungsbaum*) für

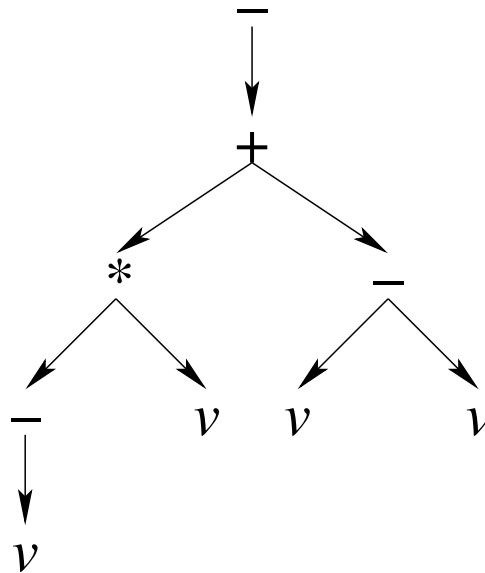
$$E \Rightarrow (E + E) \Rightarrow ((E * E) + E) \Rightarrow ((E * E) + (E - E)) \xRightarrow{*} ((v * v) + (v - v))$$



Linksableitung: Tiefensuche mit Linksabstieg durch Syntaxbaum

Rechtsableitung: Tiefensuche mit Rechtsabstieg durch Syntaxbaum

Operatorbaum: ein kompakter Syntaxbaum für das ursprüngliche Beispiel



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 $w_1 = aw_2$, $\gamma_1 = \alpha_1\beta$ für ein $\beta \in \Gamma^*$.

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

$k > 0$: $(q, w, \lambda) (\vdash \circ \vdash^{k-1}) (q, \lambda, x)$ 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 $C \rightarrow w$: $((f, \lambda, C), (f, w))$,
- für jedes Terminalzeichen a : $((f, a, a), (f, \lambda))$.

f ist der einzige Endzustand und s der Startzustand.

\rightsquigarrow **Satz: $\mathbf{KF} \subseteq \mathbf{PDA}$.**

Die Konstruktion am Beispiel

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

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* \triangleleft , d.h.,

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

Beweisideen

Satz (4), Grundversion: (a) Jede $L \in \mathbf{PDA}$ 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 $L \in \mathbf{PDA}$ 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 $L \in \mathbf{PDA}$ 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 $Q \times \Gamma \times Q$; z.B. simuliere $((q_0, a, b), (q_1, cd))$ durch $((s, a, [q_0, b, q_2]), (s, [q_1, c, q][q, d, q_2]))$; simuliere $((q_0, a, b), (q_1, \lambda))$ durch $((s, a, [q_0, b, q_1]), (s, \lambda))$; starte mit $[q_0, \triangleleft, q]$ für bel. Zustand q (es-Version) oder Endzustand q

kfGs erzeugen PDA-Sprachen

Erinnerung an Normalformensatz:

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

Damit wird die Simulation einfach:

$((s, a, A), (s, \lambda)) \rightsquigarrow \text{Regel } A \rightarrow a$ (insbesondere für Kellerbodenzeichen).

$((s, a, A), (s, B_1 \dots B_k)) \rightsquigarrow \text{Regel } A \rightarrow aB_1 \dots B_k$.

Startsymbol ist neues Zeichen S mit Regeln $S \rightarrow aX$ 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)

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

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

[] 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 heie \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 vor-
vorigen Automaten. (Wir können auf das Kellerbodenzeichen verzichten.)

$$S \rightarrow [s, \triangleleft, s_f]$$

$$[s, \triangleleft, r] \rightarrow a[s, A, r'][r', \triangleleft, r]$$

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

$$[s, \triangleleft, r] \rightarrow b[f, \triangleleft, r]$$

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

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

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

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

$$[f, \triangleleft, s_f] \rightarrow \lambda$$

Beispiellinksableitung:

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

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

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

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

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

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

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

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

$$\Rightarrow aaaba.$$

Die Konstruktionen an einem Beispiel:

Links die Arbeitsweise des Normalform-Kellerautomaten,
rechts die der daraus konstruierten Grammatik.

Beispielakzeptierung:

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

Beispiel**links**ableitung:

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

 $\Rightarrow aaaba.$