Folgerungen

7.22 Folgerung

a) Rechts-lineare Sprachen sind abgeschlossen gegenüber Komplement und Durchschnitt.
$A = (Q, \Sigma, \Pi, q_0, F)$ DEA $L = L(A)$.
$A' = (Q, \Sigma, \Pi, q_0, Q - F)$ DEA mit $L(A') = \neg L$.
$L_1 \cap L_2 = \overline{\bar{L}_1 \cup \bar{L}_2}$ oder direkt mit Produktautomaten.
$A_1 \times A_2 = (Q_1 \times Q_2, \Sigma, \Pi_1 \times \Pi_2, (q_{0_1}, q_{0_2}), F_1 \times F_2)$.

b) Jede Typ-3 Sprache kann von Typ-3 Grammatik G erzeugt werden mit: Π enthält für $X \in N, a \in \Sigma$ $X \to aY$ oder $X \to a$ (genau eine Produktion $X \to aY$). D. h. G ist eindeutig und somit ist jede Typ-3 Sprache eindeutig.

c) Das WP für Typ-3 Grammatiken ist in linearer Zeit entscheidbar.

d) **Pumping-Lemma** für Typ-3 Sprachen.
Zu jeder Typ-3 Sprache L gibt es ein $n \in \mathbb{N}$, so dass für alle $y \in L$ gilt: Ist $|y| \geq n$. Dann lässt sich y zerlegen in $y = uvw$ mit $0 < |uv| \leq n$, so dass für alle $i \in \mathbb{N}$ $uv^iw \in L$.
Beweis:
Sei A DEA mit $L(A) = L$ und $n := |Q|$. Ist $y \in L(A)$, $|y| \geq n$. Betrachte
$q_0 y \overset{1}{\vdash} q_1 y_1 \vdash \cdots \overset{1}{\vdash} q_{n-1} y_{n-1} \overset{1}{\vdash} q_n y_n \vdash \cdots \overset{1}{\vdash} q \in F$, $\{q_0, \ldots, q_n\} \subseteq Q$. Es gibt Zustand q', der zweimal vorkommt $q_0 uvw \underset{A}{\vdash} q'vw \underset{A}{\vdash} q'w \vdash q_0, |uv| \leq n$. Dann aber $q_0 uv^iw \vdash q$ für alle $i \geq 0$.

Beispiel

7.23 Beispiel

$L = \{w \in \{a,b\}^* : |w|_a = |w|_b\}$ nicht Typ 3 Sprache.

Angenommen, L ist rechts-linear, sei n Konstante für L.
Betrachte $y = a^n b^n \in L$
Pumping-Lemma $\rightsquigarrow a^{k_0}(a^k)^i a^{k_1} b^n \in L$ für alle i $(k_0 + k + k_1 = n, k > 0)$ ↯

Oder: $L \cap \{a\}^*\{b\}^* = \{a^n b^n \mid n \geq 0\}$ wäre rechts-linear, falls L es ist. ↯

e) Für eine Typ-3 Sprache sind folgende Probleme entscheidbar. Dabei soll L durch eine Typ-3 Grammatik, oder durch einen DEA, oder durch einen NEA gegeben sein.
   - Ist L leer?
   - Ist $L = \Sigma^*$?
   - Ist L endlich?
   - Ist $L = L_1$ für eine Typ-3 Sprache L_1?

Es gibt weitere Charakterisierungen von rl-Sprachen, z.B. durch rechtsinvariante Äquivalenzrelationen auf Σ^* von endlichen Index (d.h. nur endlich viele Äquivalenzklassen) oder etwa durch reguläre Ausdrücke.

Andere Charakterisierung von Typ-3 Sprachen

Reguläre Ausdrücke über Σ : $REG(\Sigma)$

Wörter über $\Sigma \cup \{\Lambda, \varepsilon, \cup, *, (,)\}$ (oft $+$ für $\cup$).

Kalkül:

$$\frac{}{\Lambda},\quad \frac{}{\varepsilon},\quad \frac{}{a} \text{ für } a \in \Sigma,\quad \frac{\alpha, \beta}{(\alpha\beta)},\quad \frac{\alpha, \beta}{(\alpha \cup \beta)},\quad \frac{\alpha}{\alpha^*}$$

Semantik: Reguläre Sprachen, die durch reg. Ausdrücke über Σ dargestellt werden: $\langle\ \rangle$: reg. Ausdruck $\longrightarrow$ Sprachen über Σ

- $\langle \Lambda \rangle = \varnothing$
- $\langle \varepsilon \rangle = \{\varepsilon\}$
- $\langle a \rangle = \{a\} \quad a \in \Sigma$
- $\langle (\alpha\beta) \rangle = \langle \alpha \rangle \circ \langle \beta \rangle$
- $\langle (\alpha \cup \beta) \rangle = \langle \alpha \rangle \cup \langle \beta \rangle$
- $\langle \alpha^* \rangle = \langle \alpha \rangle^*$

7.24 Satz

L ist Typ-3 Sprache gdw L ist reguläre Sprache, d. h.
es gibt $\alpha \in REG(\Sigma)$: $\langle \alpha \rangle = L$.

Beweis:

„$\Leftarrow$“ Typ-3 Sprachen enthalten $\varnothing$,$\{\varepsilon\}$, $\{a\}$ für $a \in \Sigma$ und sind abgeschlossen gegen $\cdot, \cup, *$.

„$\Rightarrow$“ Sei $A = (Q, \Sigma, \Pi, q_1, F)$, $Q = \{q_1, \ldots, q_n\}$ DEA mit $L(A) = L$. Für $i, j \in \{1, \ldots, n\}$ und $t \in \{0, \ldots, n\}$ definiere

$$L_{ij}^t = \{y \in \Sigma^* : q_i y \overset{1}{\vdash} q_{i_1} y_1 \overset{1}{\vdash} \cdots \overset{1}{\vdash} q_{i_k} y_k \overset{1}{\vdash} q_j \text{ mit Zwischenzuständen } q_{i_1}, \ldots, q_{i_k} \in \{q_1, \ldots, q_t\}\}$$

Behauptung: Jedes L_{ij}^t ist durch regulären Ausdruck darstellbar. Insbesondere auch $L(A)$.

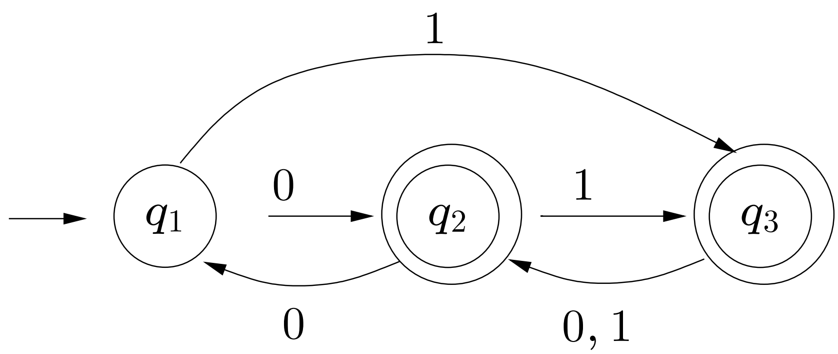
Beweis: Induktion nach t:

$L_{ij}^0 = \{y \in \Sigma^* : q_i y \overset{1}{\vdash} q_j\}$ ist endlich.

$L_{ij}^{t+1} = L_{ij}^t \cup L_{it+1}^t (L_{t+1t+1}^t)^* L_{t+1j}^t$

$L(A) = \bigcup\limits_{q_j \in F} L_{1j}^n$

7.25 Beispiel



i	j	$t=$ 0	1	2	3
1	1	ε	ε	$(00)^*$	
1	2	0	0	$0(00)^*$	
1	3	1	1	0^*1	
2	1	0	0	$0(00)^*$	
2	2	ε	$\varepsilon + 00$	$(00)^*$	
2	3	1	$1 + 01$	0^*1	
3	1	$\varnothing$	$\varnothing$	$(0+1)(00)^*0$	
3	2	$0+1$	$0+1$	$(0+1)(00)^*$	
3	3	ε	ε	$\varepsilon + (0+1)0^*1$	

Variante + Verallgemeinerungen EA

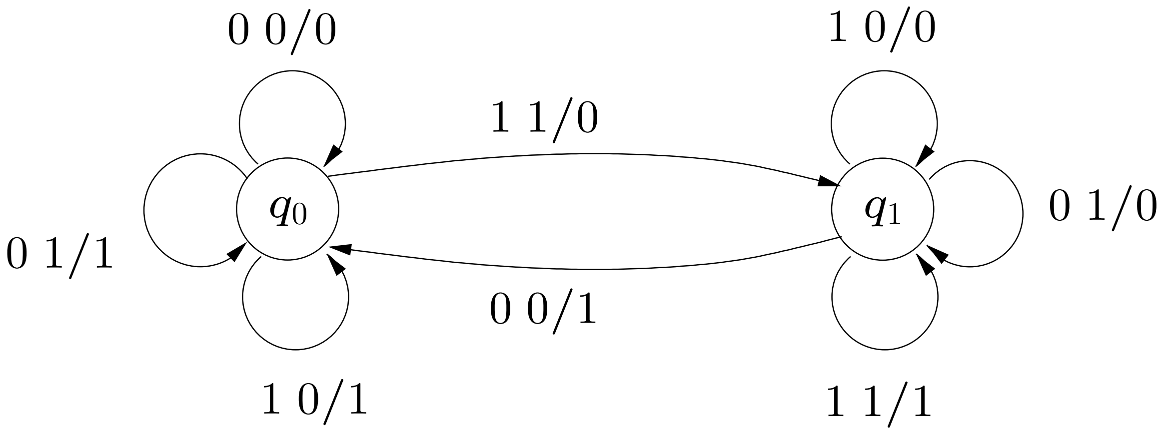
Endliche Automaten mit Ausgaben Mealy und Moore Automaten

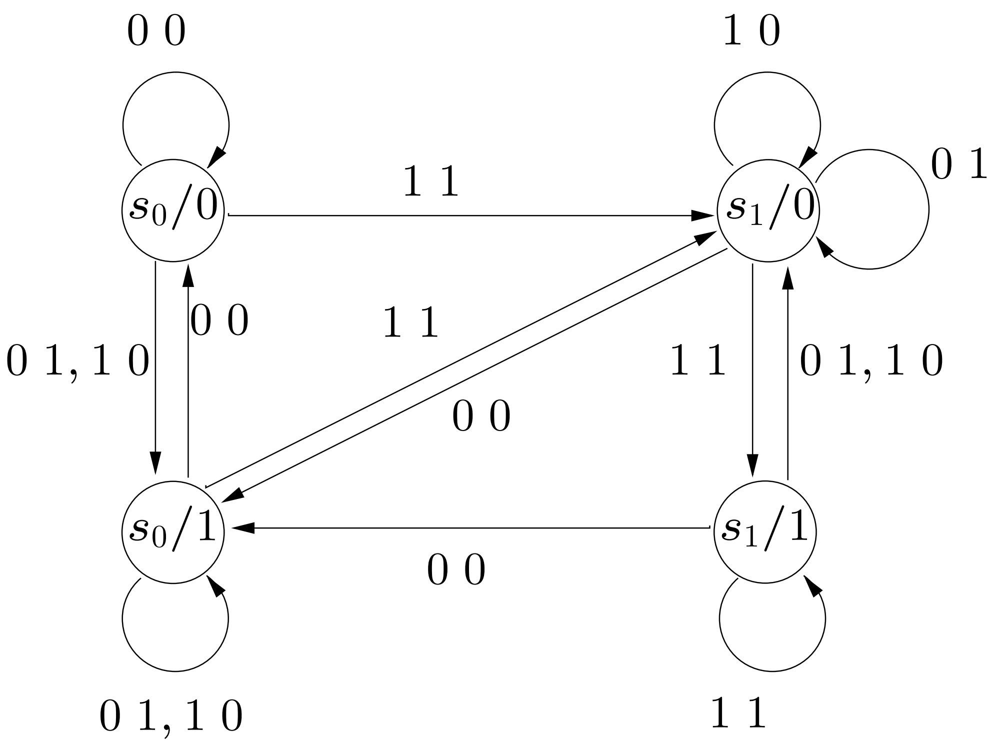
$\Sigma = \{0, 1\} \times \{0, 1\}$

mod 2 Addierer.

0 1 0 1 1

0 0 1 1 0   1 0 0 0 1





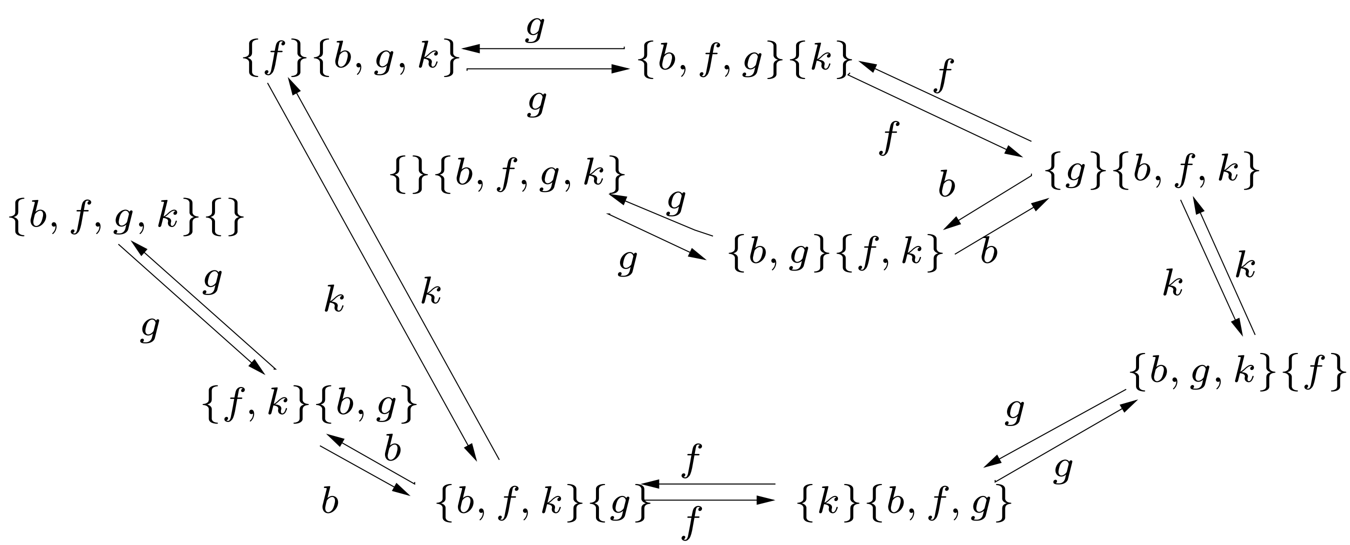
Spezifikation von Prozessen
Dynamisches Verhalten

Statecharts, Petri-Netze, SDL

UML Verhaltensdiagramme (Statecharts, Activity diagrams, MSC)

Event-Condition-Action: $e[c]$ Action: Übergänge.

Prozess: Bauer/Boot /Fluss, Gans/Fuchs/Korn.



7.4 Kontextfreie Sprachen - Typ2-Sprachen

Erinnerung Sei $G = (N, T, \Pi, Z)$ Grammatik.

G ist vom Typ 2 (kontextfrei), falls $l \to r \in \Pi$, so $l = A, r = z, A \in N, z \in (N \cup T)^*$.

Eine Sprache heißt kontextfrei, falls sie durch eine kontextfreie Grammatik erzeugt werden kann.

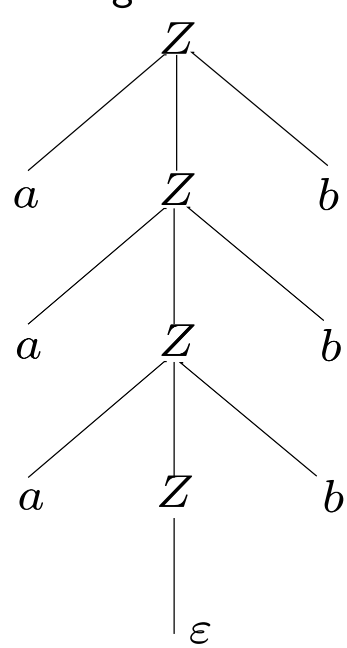
Beispiel: $G = (N, T, \Pi, Z)$, $T = \{a, b\}$, $N = \{Z\}$.

$\Pi : Z \to aZb \mid \varepsilon \qquad L(G) = \{a^n b^n \mid n \in \mathbb{N}\}$

Behauptung: $L(G)$ ist nicht rechtslinear. Sei n Konstante für L
$y = a^n b^n$. Pumping-Lemma $\rightsquigarrow (a^{k_0})(a^k)^i(a^{k_1})b^n \in L$
für alle $i \in \mathbb{N}$ $(k_0 + k + k_1 = n, k > 0)$ ↯

Gibt es auch ein Pumping-Lemma für kontextfreie Sprachen?

Es ist $aaabbb \in L(G)$. Ableitung als Baum:



Ableitungsbäume - Strukturbäume

7.26 Definition

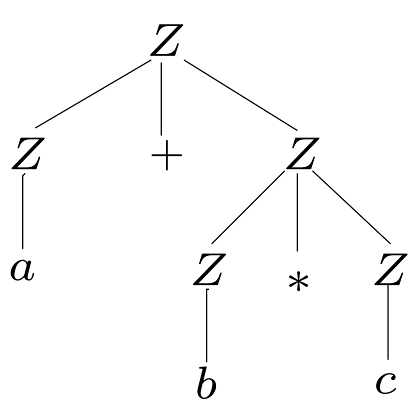
Sei G eine kontextfreie Grammatik und $(Z, u_1, \ldots, u_n)$ eine Ableitung in G. Der Strukturbaum zu dieser Ableitung wird induktiv über n definiert:

1. Der Strukturbaum zur Ableitung (Z) besteht aus einem einzigen mit Z beschrifteten Knoten. Blattwort ist Z.
2. Es sei die Ableitung $(Z, u_1, \ldots, u_n, u_{n+1})$ mit $u_n = uAv$, $u_{n+1} = ub_1 \ldots b_m v$ und eine Produktion $A \to b_1 \ldots b_m$ von G mit einzelnen Zeichen b_i gegeben. Sei weiter der Strukturbaum von $(Z, u_1, \ldots, u_n)$ schon konstruiert. Erweitere in diesem Baum den $(|u| + 1)$-ten Knoten (mit dem zu ersetzenden A beschriftet) mit m Folgeknoten, die mit $b_1, \ldots, b_m$ beschriftet sind. (ε als Zeichen erlaubt). Blattwort ist u_{n+1}.

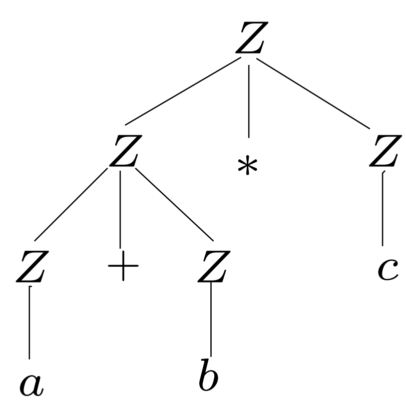
7.27 Beispiel

$G = (N, T, \Pi, Z)$ mit $N = \{Z\}, T = \{a, b, c, +, *\}$,
$\Pi : Z \to Z + Z$, $Z \to Z * Z$, $Z \to a|b|c$

a)

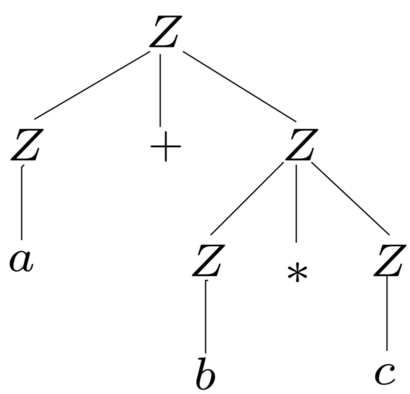


b)



Strukturbäume

a)



Es gibt zu $a + b * c$ verschiedene Ableitungen:

(i) $(\underset{\uparrow}{Z}, \underset{\uparrow}{Z} + Z, a + \underset{\uparrow}{Z}, a + \underset{\uparrow}{Z} * Z, a + b * \underset{\uparrow}{Z}, a + b * c)$

(ii) $(\underset{\uparrow}{Z}, Z + \underset{\uparrow}{Z}, Z + Z * \underset{\uparrow}{Z}, Z + \underset{\uparrow}{Z} * c, \underset{\uparrow}{Z} + b * c, a + b * c)$

Die Ableitungen (i) und (ii) sind unterschiedlich, erzeugen aber denselben Strukturbaum: a).

Desweiteren wird in Ableitung (i) immer das am weitesten links stehende Nichtterminalzeichen ersetzt. (siehe $\uparrow$).

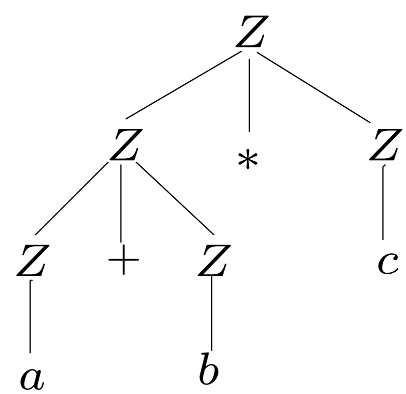
Betrachte die Ableitungen:

(iii) $(\underset{\uparrow}{Z}, \underset{\uparrow}{Z} * Z, \underset{\uparrow}{Z} + Z * Z, a + \underset{\uparrow}{Z} * Z, a + b * \underset{\uparrow}{Z}, a + b * c)$

(iv) $(\underset{\uparrow}{Z}, Z * \underset{\uparrow}{Z}, \underset{\uparrow}{Z} * c, Z + \underset{\uparrow}{Z} * c, \underset{\uparrow}{Z} + b * c, a + b * c)$

Strukturbäume

b)



Ableitungen (iii) und (iv) erzeugen Strukturbaum b).

Insgesamt:

1. Ein Strukturbaum repräsentiert eine Menge von Ableitungen.
2. Ein ableitbares Wort kann verschiedene Ableitungen haben, die nicht durch **einen** Strukturbaum dargestellt werden können.

Punkt 2 kann Schwierigkeiten bereiten, wenn einem ableitbaren Ausdruck eine Semantik (etwa ein Wert) zugeordnet werden soll.

Eindeutigkeit der Termsyntax geht verloren, wenn auf Klammern verzichtet wird. Was ist der Wert von $1 + 2 * 3$?
$(1 + 2) * 3 = 6$
$1 + (2 * 3) = 7$

Eindeutigkeit

7.28 Definition

Eine kontextfreie Grammatik G heißt eindeutig, falls für jedes $w \in L(G)$ gilt: Alle Ableitungen von w besitzen denselben Strukturbaum.

7.29 Beispiel Betrachte Grammatik $G = (N, T, \Pi, Z)$ mit
$N = \{Z\}$, $T = \{a, b, c, +, *, (,)\}$,

$$\begin{aligned} \Pi : \quad & Z \to (Z + Z) \\ & Z \to (Z * Z) \\ & Z \to a|b|c \end{aligned}$$

G ist eindeutig. $\rightsquigarrow$ Übung.

7.30 Definition

Sei G eine kontextfreie Grammatik und $(u_0, u_1, \ldots, u_n)$ eine Ableitung in G. Die Ableitung heißt **Linksableitung** in G, falls für alle $i < n$ $\;u_{i+1}$ aus u_i durch Ersetzen des am weitesten links stehende Nichterminalzeichen mit Hilfe einer Regel in G entsteht.
(**Rechtsableitung** analog).

7.31 Beispiel G aus vorherigem Beispiel

$$(\underset{\uparrow}{Z}, (\underset{\uparrow}{Z} * Z), ((\underset{\uparrow}{Z} + Z) * Z), ((a + \underset{\uparrow}{Z}) * Z),$$
$$((a + b) * \underset{\uparrow}{Z}), ((a + b) * c))$$

Ableitung für $((a + b) * c)$ $\rightsquigarrow$ Linksableitung.

Eindeutigkeit k.f. Grammatiken

7.32 Lemma

Eine kontextfreie Grammatik ist genau dann eindeutig, wenn jedes durch die Grammatik erzeugte Wort genau eine Linksableitung (bzw. Rechtsableitung) besitzt.

Beweis: Übung.

Beachte:

1. Ist $w \in L(G)$, so gibt es eine Linksableitung zu w.
2. Jede rechtslineare Grammatik ist eindeutig.
3. Es gibt sogenannte ererbt mehrdeutige kontextfreie Sprachen, etwa
   $$\begin{aligned} L = \quad & \{a^n b^n c^m d^m \mid n, m \geq 1\} \cup \\ & \{a^n b^m c^m d^n \mid n, m \geq 1\} \end{aligned}$$
   Man kann zeigen:
   Jede kontextfreie Grammatik, die $L(G)$ erzeugt, ist mehrdeutig.

Problem: Wie kann man möglichst effizient testen, ob ein Wort aus einer kontextfreien Grammatik ableitbar ist?

$\rightsquigarrow$ Konstruiere Automaten, der den Strukturbaum einer Ableitung in einer bestimmten Weise aufbaut: Top-Down, Preorder.

LL-Automaten zu einer k.f. Grammatik

7.33 Definition

Sei $G = (N, T, \Pi, Z)$ eine kontextfreie Grammatik. Der **LL-Automat** zu G ist das folgende Tupel

$$A_{LL}(G) = (\{\#\}, N, T, \Pi_{LL}(G), Z\#, \{\#\})$$

Mit folgenden Produktionen in $\Pi_{LL}(G)$:

Für alle $t \in T$ und alle Produktionen
$A \to B_1 \dots B_n \in \Pi$ mit einzelnen Zeichen B_i

$A\# \to B_n \dots B_1\#$	(Produce) (Beachte die Reihenfolge der B's)
$t\#t \to \#$	(Compare)

Ableitbarkeit in A_{LL} bedeutet Ableitbarkeit in diesem Wortersetzungssystem. Die von A_{LL} akzeptierte Sprache ist die Menge

$$\{x \in T^* : Z\#x \underset{\Pi_{LL}(G)}{\vdash} \#\}$$

Initialkonfiguration bei Eingabe $x \in T^* : Z\#x$, d. h.
$i(X) = Z\#x$.

Finalkonfigurationen: $\{\#\}$

7.34 Lemma Sei G eine kontextfreie Grammatik.

Es ist $x \in L(G) \text{ gdw } x \in L(A_{LL}(G))$.

Beispielkonstruktion

7.35 Beispiel G aus vorherigem Beispiel,

$$
\begin{aligned}
\Pi_{LL}(G): \quad & Z\# \to \ )Z + Z(\# \\
& Z\# \to \ )Z * Z(\# \\
& Z\# \to a\#|\ b\#|\ c\# \\
& a\#a \to \# \\
& b\#b \to \# \\
& \vdots \\
&)\#) \to \#
\end{aligned}
$$

Wir wissen $((a+b)*c) \in L(G)$.

Betrachte Ableitung

$$
\begin{aligned}
(\quad & Z\#((a+b)*c, \\
&)Z * Z(\#((a+b)*c), \\
&)Z * Z\#(a+b)*c), \\
&)Z*)Z + Z(\#(a+b)*c), \\
&)Z*)Z + Z\#a+b)*c), \\
&)Z*)Z + a\#a+b)*c), \\
&)Z*)Z + \#+ +b)*c), \\
&)Z*)Z\#b)*c), \\
& \vdots \\
& \#)
\end{aligned}
$$

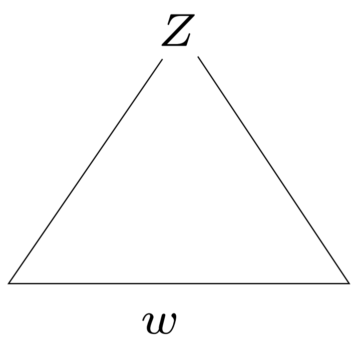
Spezielle Eigenschaften kontextfreier Sprachen
Pumping-Lemma

Erinnerung: Syntaxanalyse: G Typ-2 Grammatik.

- $w \in L(G)$, so gibt es eine Linksherleitung (Ableitung) für w aus z, d. h.

$$Z \underset{G}{\overset{1}{\vdash}} \alpha_1 \underset{G}{\overset{1}{\vdash}} \alpha_2 \underset{G}{\overset{1}{\vdash}} \cdots \vdash \alpha_n = w$$

- LL-Automat akzeptiert w (simuliert die Linksableitung).
- Zugehöriger Strukturbaum (geordneter markierter Baum, mit Blattwort w).



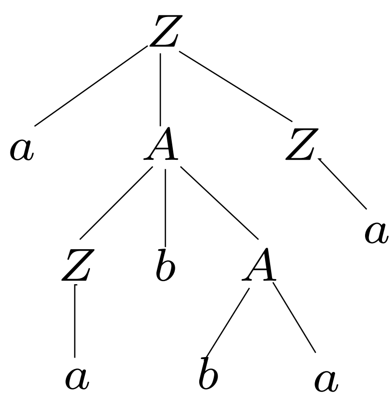
- G ist eindeutig gdw für kein $w \in L(G)$ gibt es zwei verschiedene Strukturbäume.
  gdw keine zwei verschiedene Linksableitungen.

  Es gibt kontextfreie Sprachen, die nicht von eindeutiger kontextfreier Grammatik erzeugt werden können.
  z. B. $\{b^m c^m d^l : m, l \geq 1\} \cup \{b^l c^n d^n : l, n \geq 1\}$
  Alle Wörter der Form $b^i c^i d^i \;\; i \geq 1$ sind mehrdeutig.

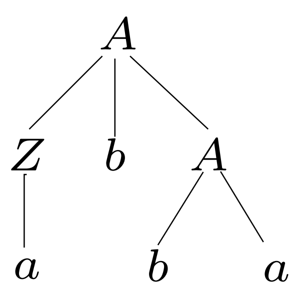
Beispiel: Pumping Eigenschaft

7.36 Beispiel $G = (\{Z, A\}, \{a, b\}, \Pi, Z)$ mit
$\Pi : Z \to aAZ \mid a \qquad A \to ZbA \mid ZZ \mid ba$

- $Z \vdash aAZ \vdash aZbAZ \vdash aabAZ \vdash aabbaZ \vdash aabbaa$
- Strukturbaum für $aabbaa$



Teilbaum mit Wurzel
A ist Strukturbaum
für Begrenzung vom Teilbaum
$A \underset{G}{\vdash} abba$



Beachte
$A \underset{G}{\vdash} abA \underset{G}{\vdash} (ab)^n A \underset{G}{\vdash} (ab)^n ba$ oder
$Z \underset{G}{\vdash} aabbaZ \underset{G}{\vdash} (aabba)^n Z \vdash (aabba)^n a$

„Aufpumpen“ von Teilwörter bei Wiederholung nichtterminaler Buchstaben.

Pumping Lemma für k.f. Sprachen

7.37 Lemma $G = (N, T, \Pi, Z)$ kontext-freie Grammatik.

Sei $p = \max\{|\beta_i| : \alpha_i \to \beta_i \in \Pi\}$. Ist $\mathcal{B}$ Strukturbaum für $\alpha \in (N \cup T)^*$ der Tiefe h, so gilt $|\alpha| \leq p^h$.
(Da Anzahl der Blätter $\leq p^h$).

7.38 Satz $uvwxy$-**Theorem** (Bar-Hillel, Perles, Shamir).

Sei L eine kontext-freie Sprache. Dann gibt es ein $n \in \mathbb{N}$, so dass für jedes Wort $z \in L(G)$ mit $|z| \geq n$ gilt:
Es gibt eine Zerlegung von z in $uvwxy$ mit $0 < |vx|$ und $|vwx| \leq n$ und für jedes $i \in \mathbb{N}$ ist auch $uv^iwx^iy \in L(G)$.

- (Beachte: Insbesondere ist auch $uwy \in L(G)$).

Beweis-Idee: o.B.d.A. sei L erzeugt von kontext-freier Grammatik G ohne ε-Regeln (bis auf $Z \to \varepsilon$).

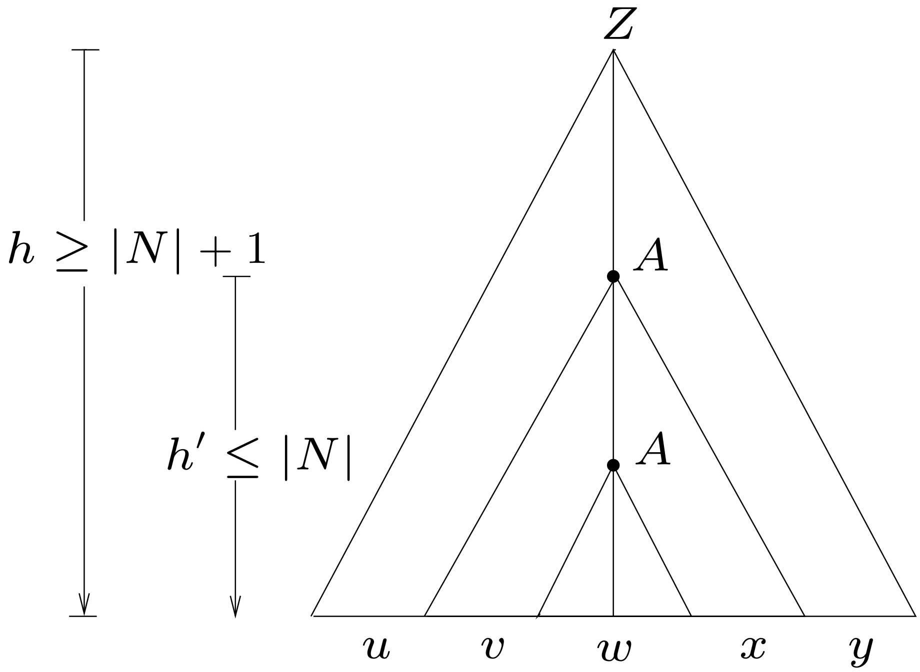
Sei $p = \max\{|\beta| : A \to \beta \in \Pi_G\}$. Betrachte $p^{|N|}$ und $z \in L(G)$ mit $|z| > p^{|N|}$. Ist $\mathcal{B}$ Strukturbaum für z, so ist die Tiefe von $\mathcal{B}$ mindestens $|N| + 1$. Sei $\mathcal{B}$ gewählt von minimaler Tiefe h.

Behauptung: Es gibt $A \in N$ mit

$Z \underset{G}{\vdash} uAy \underset{G}{\vdash} uvAxy \underset{G}{\vdash} uvwxy = z$, wobei
$u, v, w, x, y \in \Sigma^*$, $vx \neq \varepsilon$, $|vwx| \leq p^{|N|}$.
Dann $A \underset{G}{\vdash} vAx$, $A \underset{G}{\vdash} w$, wähle $n = p^{|N|} + 1$.

Beweisargument

Beachte: Analoges Argument führt zu Beweis des Pumping-Lemmas für RL-Grammatiken.



Z kommt auf keiner rechten Seite vor.

keine ε-Regeln.

- Innere Knoten sind mit Nichtterminalsymbolen (NT) markiert.
- Da $h \geq |N|+1$, gibt es eine Weg zu Blatt der Länge $\geq |N|+1$
- NT-Symbol (verschieden von Z) wiederholt sich.
- Wähle NT A maximaler Tiefe, d.h. Teilbaum unter A hat Tiefe $\leq |N|$ und $|vwx| \leq p^{|N|}$.
- Dann $vx \neq \varepsilon$, da $\mathcal{B}$ minimaler Tiefe.

$\rightsquigarrow$ Behauptung.

Anwendungen

7.39 Folgerung und Anwendungen

a) Die Sprache $L = \{a^m b^m c^m \mid m > 0\}$ ist **nicht kontextfrei**. Angenommen L ist kontextfrei, n die Konstante vom $uvwxy$-Theorem. Wähle $m > n/3$.
$z = a^m b^m c^m = uvwxy$, $vx \neq \varepsilon$, $|vwx| \leq n$
Enthält v oder x mindestens zwei Buchstaben aus $\{a, b, c\}$, so $uv^2wx^2y \notin L$, da falsche Reihenfolge der Buchstaben.
Falls v und x nur aus a's, b's oder c's, so falsche Anzahl, da nur zwei gekoppelt.

b) $L = \{a^n : n \text{ Primzahl }\} \subseteq a^*$ ist **nicht kontextfrei**. Angenommen ja. Dann ist L RL-Sprache (warum?). Sei n Konstante des Pumping-Lemmas für RL-Sprachen $a^p \in L$ mit $p > n$. Dann ist $a^p = a^i a^j a^k$, $j > 0$, $a^{i+l \cdot j+k} \in L$, $l \geq 0$. D. h. $i + l \cdot j + k$ ist Primzahl für alle l, insbesondere für $l = i + k$
$\lightning$

c) Kontextfreie-Sprachen (Typ-2 Sprachen) sind nicht abgeschlossen gegen $\cap$ und $\neg$.
Beweis:
$L_1 = \{a^n b^n c^m : n, m \geq 1\}$, $L_2 = \{a^m b^n c^n : n, m \geq 1\}$ sind kontextfrei, aber $L_1 \cap L_2 = \{a^n b^n c^n : n \geq 1\}$ ist nicht kontextfrei, wegen $L_1 \cap L_2 = \Sigma^* - ((\Sigma^* - L_1) \cup (\Sigma^* - L_2))$ folgt Behauptung.

Anwendungen (Forts.)

d) Sei $G = (N, T, \Pi, Z)$ kontextfreie Grammatik $p = \max\{|\beta| : A \to \beta \in \Pi\}$, $n = p^{|N|}$. $L(G)$ ist unendlich gdw es gibt $z \in L(G) : n < |z| \leq n \cdot (p+1)$.

Beweis:

„$\Leftarrow$“ Pumping-Lemma.

„$\Rightarrow$“ $z \in L(G)$ minimale Länge mit $|z| > n$. Angenommen $|z| > n \cdot (p+1)$, dann $z = uvwxy \in L(G)$, $0 < |vx| \leq |vwx| \leq n$ und $uwy \in L(G)$ nach Pumping-Lemma. Dann ist $n < |uwy| < |z|$ ↯

Insbesondere ist es entscheidbar, ob $L(G)$ unendliche Sprache für G Typ-2 Grammatik.

e) Beachte: Pumping-Lemma liefern notwendige, jedoch nicht hinreichende Bedingungen für L Typ-2 (3) Sprache:
$\{a^p b^n : p\text{-Primzahl}, n \geq p\}$ ist nicht kontextfrei, dies kann aber nicht mit Pumping-Lemma bewiesen werden.

LL-Automat für G $(\{\#\}, N, T, \Pi_{LL}(G), Z\#, \{\#\})$ kann als Kellerautomat aufgefasst werden. Nur ein Zustand $\#$.

Kontextfreie Sprachen und Kellerautomaten

7.40 Definition

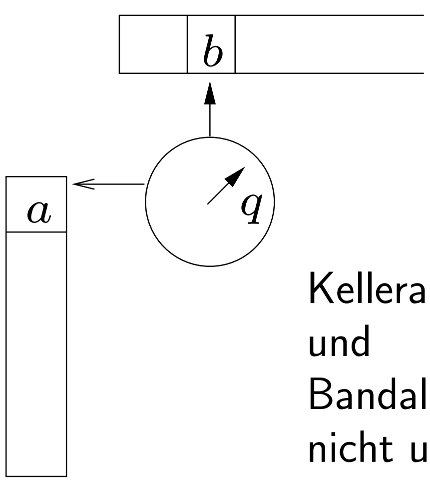
Ein Kellerautomat $K = (Q, N, T, \Pi, iq_0, F)$ mit Q Zustandsmenge, T Eingabealphabet, N Kelleralphabet, $i \in N, q_0 \in Q$, $F \subset Q$. Anfangskonfiguration: Für $x \in T^*$ $i(x) = iq_0x$,
Π Produktionen der Form

$$\begin{aligned} aqb &\rightarrow xq' \quad \text{(Lesen eines Zeichens)} \\ aq &\rightarrow xq' \quad \text{(Spontanübergang)} \end{aligned}$$

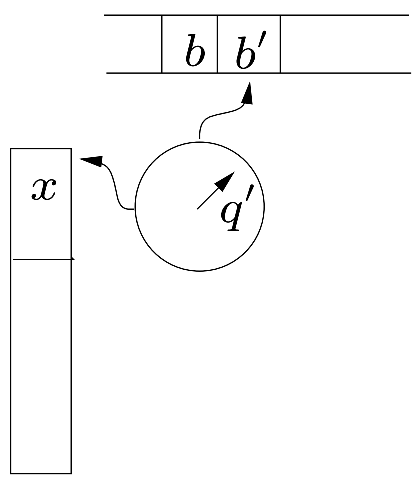
mit $x \in N^*, a \in N, q, q' \in Q$ und $b \in T$.
Die von K akzeptierte Sprache ist die Menge

$$L(K) = \{x \in T^* : iq_0x \underset{\Pi}{\vdash} f \text{ für ein } f \in F\}$$

Lesen eines Zeichens und Spontanübergänge erzeugen in Abhängigkeit eines gewissen Buchstabens im Keller ein neues Wort.



Kelleralphabet
und
Bandalphabet
nicht unbedingt
disjunkt



Beispiele

Deterministische Kellerautomaten:
Für $(a, q) \in N \times Q$ gibt es entweder genau eine Produktion der Form $aq \to xq'$ oder für jedes $b \in T$ genau eine Produktion der Form $aqb \to xq'$. $\rightsquigarrow$ **Deterministische kontextfreie Sprachen**.

7.41 Beispiel

1. $L = \{w \not{c}\, w^{mi} : w \in \{a, b\}^*\}$
   k.f. Grammatik für L: $Z \to aZa \mid bZb \mid \not{c}$
   $K = (\{q_0, q_1\}, \{Z, a, b\}, \{a, b, \not{c}\}, \Pi, Zq_0, F = \{q_1\})$
   $$\begin{array}{llll} \Pi :: & zq_0a \mapsto zaq_0 & zq_0b \to zbq_0 & z \in \{Z, a, b\} \\ & zq_0\ \not{c} \to zq_1 & & z \in \{Z, a, b\} \\ & aq_1a \to q_1 & bq_1b \to q_1 & \\ & Zq_1 \to q_1 & & \end{array}$$
   K ist deterministischer Kellerautomat $L(K) = L$. Also ist L eine deterministische k.f. Sprache.

2. $G = (N, T, \Pi, Z)$, $I = \{a, b\}$, $\Pi : Z \to aZa \mid bZb \mid \varepsilon$
   Dann gilt $L(G) = \{ww^{mi} : w \in T^*\}$.
   Sei K mit $Q = \{q\}$, $N = \{Z, a, b\}$, $q_0 = q$, $i = Z$, und Π_K:
   $aqa \to q$, $bqb \to q$
   $Zq \to aZaq \mid bZbq \mid q$
   (nicht deterministische Produktionen).

Beispiele (Fort.)

Behauptung: $L(K) = L(G)$

„$\supseteq$" klar.

„$\subseteq$" $Zqw \vdash q \rightsquigarrow Z$ muss vom Keller gelöscht werden., d. h.
$Zqw \vdash uZqv \overset{1}{\vdash} uqv \vdash q$
$uqv \vdash q$, wobei Z in u nicht enthalten ist.
$\rightsquigarrow$ nur Vergleiche, also $|u| = |v| \wedge u^{mi} = v$ (Ind. $|u|$). v ist Endwort von w, d. h. $w = xv = xu^{mi}$ und $Zq_0w \vdash uZqu^{mi}$, d. h. $2|u|$ Schritte und $w = uu^{mi}$
Induktion nach $|u|$.