Bemerkung:
Das uniforme wie auch das nicht-uniforme Wortproblem ist für Typ-0-Sprachen (also die rekursiv-aufzählbare Sprachen) nicht entscheidbar. Wir werden später sehen, dass es zum Halteproblem für Turingmaschinen äquivalent ist.

Es gilt jedoch

Satz 16

Für kontextsensitive Grammatiken ist das Wortproblem entscheidbar.
Genauer: Es gibt einen Algorithmus, der bei Eingabe einer kontextsensitiven Grammatik $G = (V, \Sigma, P, S)$ und eines Wortes w in endlicher Zeit entscheidet, ob $w \in L(G)$.

Beweisidee:
Angenommen $w \in L(G)$. Dann gibt es eine Ableitung

$$S = w^{(0)} \to_G w^{(1)} \to_G \cdots \to_G w^{(n)} = w$$

mit $w^{(i)} \in (\Sigma \cup V)^*$ für $i = 1, \ldots, n$.
Da aber G kontextsensitiv ist, gilt (falls $w \neq \epsilon$)

$$|w^{(0)}| \leq |w^{(1)}| \leq \cdots \leq |w^{(n)}| \ ,$$

d.h., es genügt, alle Wörter in $(\Sigma \cup V)^*$ der Länge $\leq |w|$ zu erzeugen.

Beweis:

Sei o.B.d.A. $w \neq \epsilon$ und sei

$$\begin{aligned} T_m^n \quad := \quad & \{w' \in (\Sigma \cup V)^*;\ |w'| \leq n \text{ und} \\ & w' \text{ lässt sich aus } S \text{ in } \leq m \text{ Schritten ableiten}\} \end{aligned}$$

Diese Mengen kann man für alle n und m induktiv wie folgt berechnen:

$$\begin{aligned} T_0^n \quad &:= \quad \{S\} \\ T_{m+1}^n \quad &:= \quad T_m^n \cup \{w' \in (\Sigma \cup V)^*;\ |w'| \leq n \text{ und} \\ & \qquad\qquad w'' \to w' \text{ für ein } w'' \in T_m^n\} \end{aligned}$$

Beachte: Für alle m gilt: $|T_m^n| \leq \sum_{i=1}^{n} |\Sigma \cup V|^i$.
Es muss daher immer ein m_0 geben mit

$$T_{m_0}^n = T_{m_0+1}^n = \ldots =: T_n \ .$$

Beweis (Forts.):

Algorithmus:

$n := |w|$
$T := \{S\}$
$T' := \emptyset$
while $T \neq T'$ **do**
$T' := T$
$T := T' \cup \{w' \in (V \cup \Sigma)^+;\ |w'| \leq n, (\exists w'' \in T')[w'' \to w']\}$
od
if $w \in T$ **return** "ja" **else return** "nein" **fi**

□

Beispiel 17

Gegeben sei die Typ-2-Grammatik mit den Produktionen

$$S \to ab \text{ und } S \to aSb$$

sowie das Wort $w = abab$.

$$\begin{aligned}
T_0^4 &= \{S\} \\
T_1^4 &= \{S, ab, aSb\} \\
T_2^4 &= \{S, ab, aSb, aabb\} \quad aaSbb \text{ ist zu lang!} \\
T_3^4 &= \{S, ab, aSb, aabb\}
\end{aligned}$$

Also lässt sich das Wort w mit der gegebenen Grammatik nicht erzeugen!

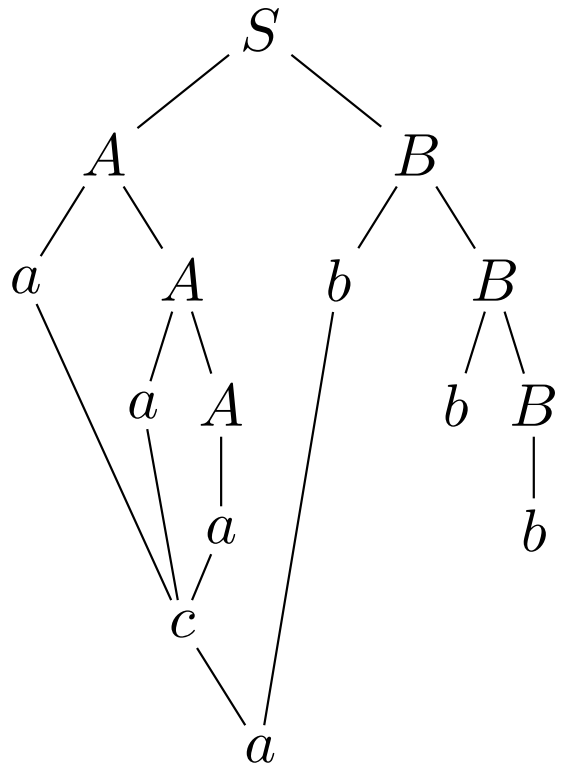
Bemerkung:
Der angegebene Algorithmus ist nicht sehr effizient! Für kontextfreie Grammatiken gibt es wesentlich effizientere Verfahren, die wir später kennenlernen werden!

2.4 Ableitungsgraph und Ableitungsbaum

Grammatik:

$$
\begin{aligned}
S &\rightarrow AB \\
A &\rightarrow aA \\
A &\rightarrow a \\
B &\rightarrow bB \\
B &\rightarrow b \\
aaa &\rightarrow c \\
cb &\rightarrow a
\end{aligned}
$$

Beispiel:

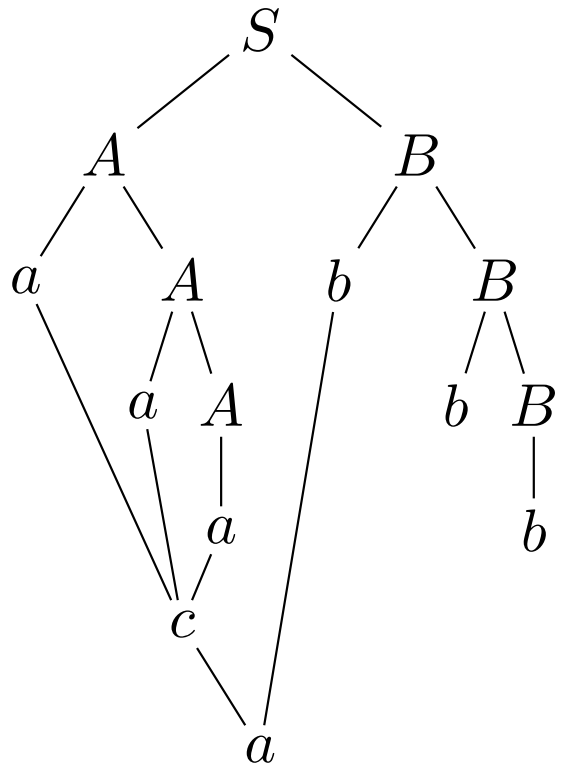


Die Terminale ohne Kante nach unten entsprechen, von links nach rechts gelesen, dem durch den Ableitungsgraphen dargestellten Wort.

Grammatik:

$$
\begin{aligned}
S &\rightarrow AB \\
A &\rightarrow aA \\
A &\rightarrow a \\
B &\rightarrow bB \\
B &\rightarrow b \\
aaa &\rightarrow c \\
cb &\rightarrow a
\end{aligned}
$$

Beispiel:



Dem Ableitungsgraph entspricht z.B. die Ableitung

$$
\begin{aligned}
S &\rightarrow AB \rightarrow aAB \rightarrow aAbB \rightarrow aaAbB \rightarrow aaAbbB \rightarrow \\
&\rightarrow aaabbB \rightarrow aaabbb \rightarrow cbbb \rightarrow abb
\end{aligned}
$$

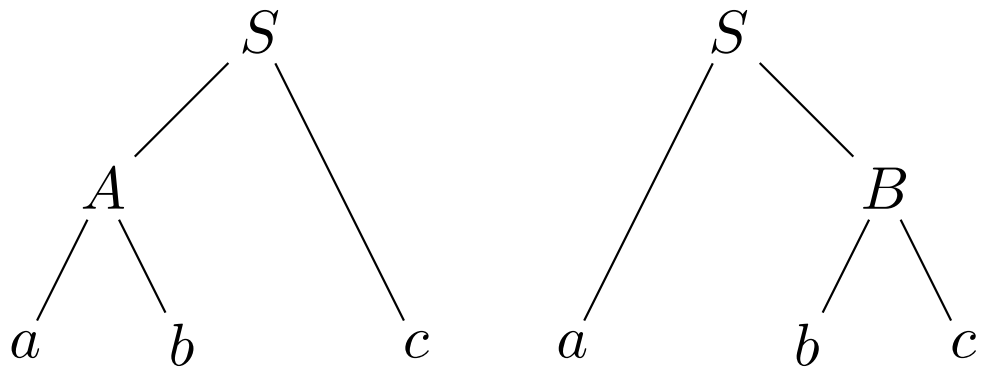
Beobachtung:
Bei kontextfreien Sprachen sind die Ableitungsgraphen immer Bäume.

Beispiel 18

Grammatik:

$$
\begin{aligned}
S &\to aB \\
S &\to Ac \\
A &\to ab \\
B &\to bc
\end{aligned}
$$

AbleitungsbaumAbleitungsbäume:



Für das Wort abc gibt es zwei verschiedene Ableitungsbäume.

Definition 19

- Eine Ableitung

$$S = w^{(0)} \rightarrow w^{(1)} \rightarrow \cdots \rightarrow w^{(n)} = w$$

  eines Wortes w heißt Linksableitung, wenn für jede Anwendung einer Produktion $\alpha \rightarrow \beta$ auf $w^{(i)} = x\alpha z$ gilt, dass sich keine Regel der Grammatik auf ein echtes Präfix von $x\alpha$ anwenden lässt.
- Eine Grammatik heißt eindeutig, wenn es für jedes Wort $w \in L(G)$ genau eine Linksableitung gibt. Nicht eindeutige Grammatiken nennt man auch mehrdeutig.
- Eine Sprache L heißt eindeutig, wenn es für L eine eindeutige Grammatik gibt. Ansonsten heißt L mehrdeutig.

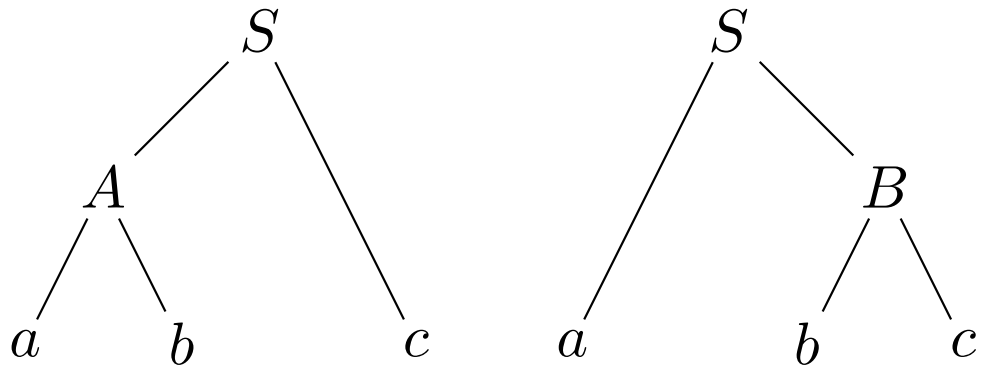
Bemerkung: Für eindeutige kontextfreie Grammatiken ist das Wortproblem sehr einfach lösbar.

Beispiel 20

Grammatik:

$$
\begin{aligned}
S &\rightarrow aB \\
S &\rightarrow Ac \\
A &\rightarrow ab \\
B &\rightarrow bc
\end{aligned}
$$

Ableitungsbäume:



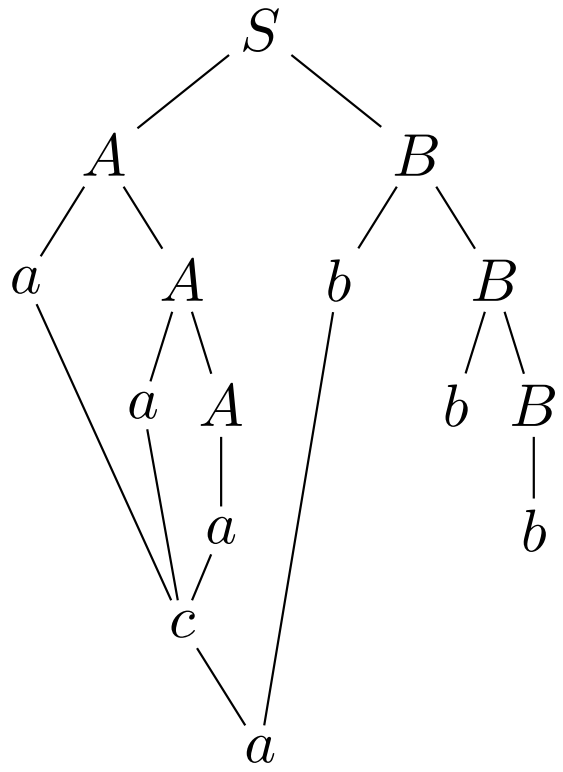
Beide Ableitungsbäume für das Wort abc entsprechen Linksableitungen.

Beispiel 21

Grammatik:

$$
\begin{aligned}
S &\rightarrow AB \\
A &\rightarrow aA \\
A &\rightarrow a \\
B &\rightarrow bB \\
B &\rightarrow b \\
aaa &\rightarrow c \\
cb &\rightarrow a
\end{aligned}
$$

Ableitung:



Eine Linksableitung ist

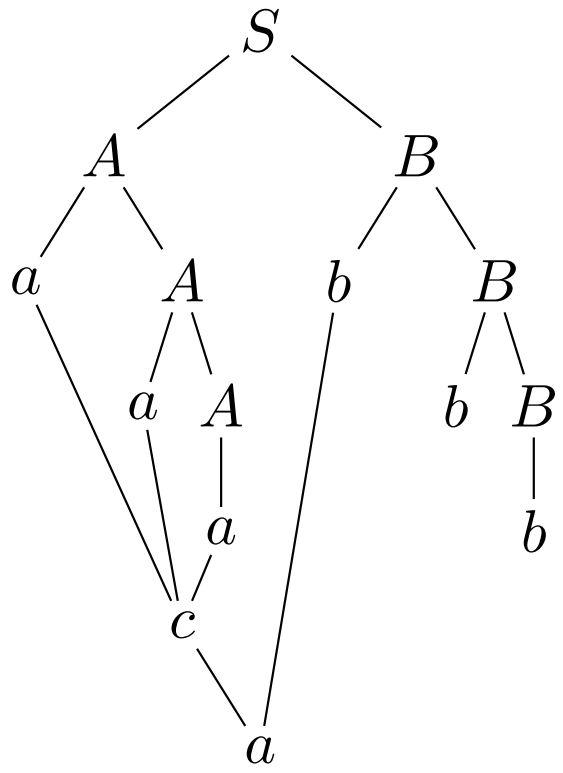
$$
\begin{aligned}
S &\rightarrow AB \rightarrow aAB \rightarrow aaAB \rightarrow aaaB \rightarrow cB \rightarrow \\
&\rightarrow cbB \rightarrow aB \rightarrow abB \rightarrow abb
\end{aligned}
$$

Beispiel 21

Grammatik:

$$
\begin{aligned}
S &\rightarrow AB \\
A &\rightarrow aA \\
A &\rightarrow a \\
B &\rightarrow bB \\
B &\rightarrow b \\
aaa &\rightarrow c \\
cb &\rightarrow a
\end{aligned}
$$

Ableitung:



Eine andere Linksableitung ist

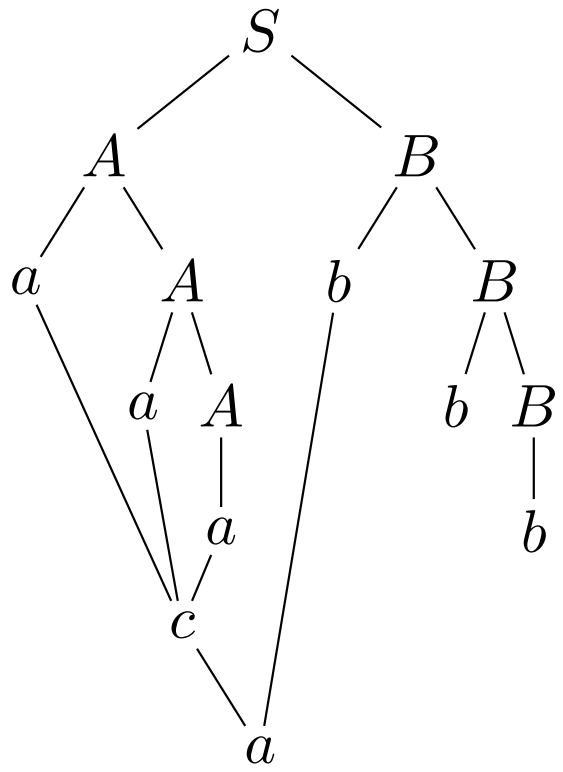
$$S \rightarrow AB \rightarrow aB \rightarrow abB \rightarrow abb\ .$$

Beispiel 21

Grammatik:

$$
\begin{aligned}
S &\rightarrow AB \\
A &\rightarrow aA \\
A &\rightarrow a \\
B &\rightarrow bB \\
B &\rightarrow b \\
aaa &\rightarrow c \\
cb &\rightarrow a
\end{aligned}
$$

Ableitung:



Die Grammatik ist also mehrdeutig.

3. Reguläre Sprachen

3.1 Deterministische endliche Automaten

Definition 22

Ein deterministischer endlicher Automat (englisch: deterministic finite automaton, kurz DFA) wird durch ein 5-Tupel $M = (Q, \Sigma, \delta, q_0, F)$ beschrieben, das folgende Bedingungen erfüllt:

1. Q ist eine endliche Menge von Zuständen.
2. Σ ist eine endliche Menge, das Eingabealphabet, wobei $Q \cap \Sigma = \emptyset$.
3. $q_0 \in Q$ ist der Startzustand.
4. $F \subseteq Q$ ist die Menge der Endzustände.
5. $\delta : Q \times \Sigma \to Q$ heißt Übergangsfunktion.

Die von M akzeptierte Sprache ist

$$L(M) := \{w \in \Sigma^*;\ \hat{\delta}(q_0, w) \in F\}\ ,$$

wobei $\hat{\delta} : Q \times \Sigma^* \to Q$ induktiv definiert ist durch

$$\begin{array}{rcll} \hat{\delta}(q, \epsilon) & = & q & \text{für alle } q \in Q \\ \hat{\delta}(q, ax) & = & \hat{\delta}(\delta(q, a), x) & \text{für alle } q \in Q, a \in \Sigma \\ & & & \text{und } x \in \Sigma^* \end{array}$$

Bemerkung: Endliche Automaten können durch (gerichtete und markierte) Zustandsgraphen veranschaulicht werden:

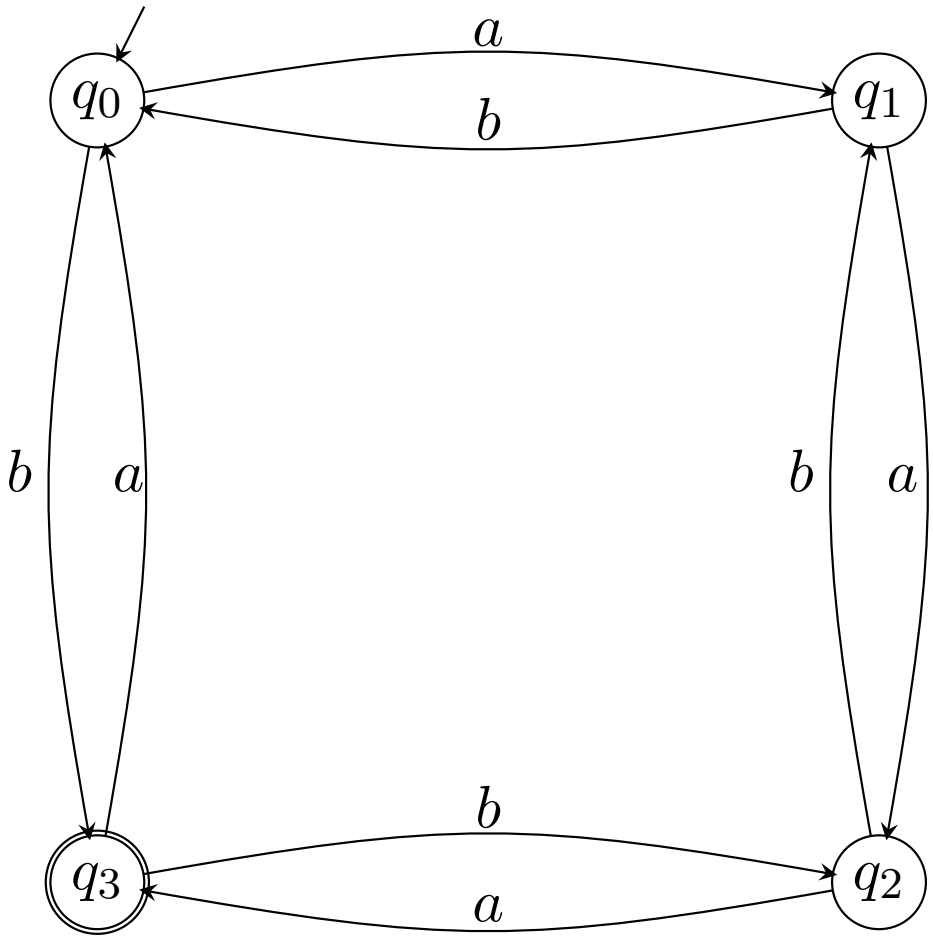
- Knoten $\widehat{=}$ Zuständen
- Kanten $\widehat{=}$ Übergängen
- genauer: die mit $a \in \Sigma$ markierte Kante (u, v) entspricht $\delta(u, a) = v$

Der Anfangszustand wird durch einen Pfeil, Endzustände werden durch doppelte Kreise gekennzeichnet.

Beispiel 23

Sei $M = (Q, \Sigma, \delta, q_0, F)$, wobei

$$
\begin{aligned}
Q &= \{q_0, q_1, q_2, q_3\} \\
\Sigma &= \{a, b\} \\
F &= \{q_3\} \\
\delta(q_0, a) &= q_1 \\
\delta(q_0, b) &= q_3 \\
\delta(q_1, a) &= q_2 \\
\delta(q_1, b) &= q_0 \\
\delta(q_2, a) &= q_3 \\
\delta(q_2, b) &= q_1 \\
\delta(q_3, a) &= q_0 \\
\delta(q_3, b) &= q_2
\end{aligned}
$$



Satz 24

Ist $M = (Q, \Sigma, \delta, q_0, F)$ ein deterministischer endlicher Automat, so ist die durch

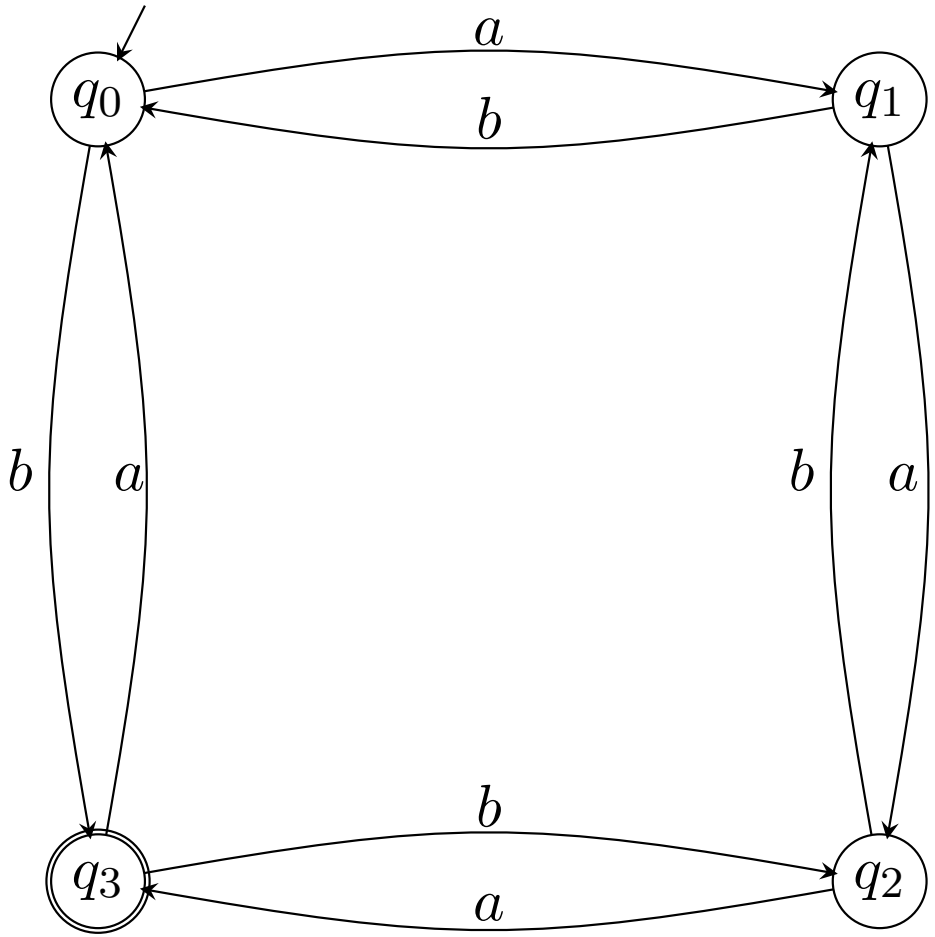
$$P \quad := \quad \{q \to aq';\ \delta(q,a) = q'\} \cup \{q \to a;\ \delta(q,a) \in F\}$$

gegebene Grammatik $G = (Q, \Sigma, P, q_0)$ regulär.

Beweis:

Offensichtlich! □

Beispiel 25



Produktionen:

$$
\begin{array}{llllll}
q_0 & \to & aq_1 & q_0 & \to & bq_3 \\
q_1 & \to & aq_2 & q_1 & \to & bq_0 \\
q_2 & \to & aq_3 & q_2 & \to & bq_1 \\
q_3 & \to & aq_0 & q_3 & \to & bq_2 \\
q_2 & \to & a & q_0 & \to & b
\end{array}
$$

$$
\begin{aligned}
q_0 &\to aq_1 \to abq_0 \\
&\to abaq_1 \to abaaq_2 \\
&\to abaaa \in L(G)
\end{aligned}
$$

Satz 26

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein endlicher deterministischer Automat. Dann gilt für die soeben konstruierte reguläre Grammatik G

$$L(G) = L(M) \; .$$

Beweis:

Sei $w = a_1 a_2 \cdots a_n \in \Sigma^*$. Dann gilt gemäß Konstruktion:

$$\begin{aligned}
& w \in L(M) \\
\Leftrightarrow \quad & \exists q_0, q_1, \ldots, q_n \in Q\text{: } q_0 \text{ Startzustand von } M, \\
& \forall i = 0, \ldots, n-1\text{: } \delta(q_i, a_{i+1}) = q_{i+1},\ q_n \in F \\
\Leftrightarrow \quad & \exists q_0, q_1, \ldots, q_{n-1} \in V\text{: } q_0 \text{ Startsymbol von } G \\
& q_0 \to a_1 q_1 \to a_1 a_2 q_2 \to \cdots \to a_1 \cdots a_{n-1} q_{n-1} \to \\
& \to a_1 \cdots a_{n-1} a_n \\
\Leftrightarrow \quad & w \in L(G)
\end{aligned}$$

□

3.2 Nichtdeterministische endliche Automaten

Definition 27

Ein nichtdeterministischer endlicher Automat (englisch: nondeterministic finite automaton, kurz NFA) wird durch ein 5-Tupel $N = (Q, \Sigma, \delta, S, F)$ beschrieben, das folgende Bedingungen erfüllt:

1. Q ist eine endliche Menge von Zuständen.
2. Σ ist eine endliche Menge, das Eingabealphabet, wobei $Q \cap \Sigma = \emptyset$.
3. $S \subseteq Q$ ist die Menge der Startzustände.
4. $F \subseteq Q$ ist die Menge der Endzustände.
5. $\delta : Q \times \Sigma \to \mathcal{P}(Q) \setminus \{\emptyset\}$ heißt Übergangsrelation.

Die von N akzeptierte Sprache ist

$$L(N) := \{w \in \Sigma^*;\ \hat{\delta}(S, w) \cap F \neq \emptyset\}\ ,$$

wobei $\hat{\delta} : \mathcal{P}(Q) \times \Sigma^* \to \mathcal{P}(Q)$ wieder induktiv definiert ist durch

$$\begin{aligned}
\hat{\delta}(Q', \epsilon) &= Q' \quad \forall Q' \subseteq Q, Q' \neq \emptyset \\
\hat{\delta}(Q', ax) &= \hat{\delta}(\bigcup_{q \in Q'} \delta(q, a), x) \quad \forall \emptyset \neq Q' \subseteq Q, \forall a \in \Sigma, \forall x \in \Sigma^*
\end{aligned}$$

Satz 28

Ist $G = (V, \Sigma, P, S)$ eine reguläre (rechtslineare) Grammatik, so ist $N = (V \cup \{X\}, \Sigma, \delta, \{S\}, F)$, mit

$$F := \begin{cases} \{S, X\}, & \text{falls } S \to \epsilon \in P \\ \{X\}, & \text{sonst} \end{cases}$$

und, für alle $A, B \in V$, $a \in \Sigma$,

$$\begin{aligned} B \in \delta(A, a) \quad &\Longleftrightarrow \quad A \to aB \qquad \text{und} \\ X \in \delta(A, a) \quad &\Longleftrightarrow \quad A \to a \end{aligned}$$

ein nichtdeterministischer endlicher Automat.

Beispiel 29

Produktionen:

$$
\begin{aligned}
S &\rightarrow aA \\
S &\rightarrow bB \\
A &\rightarrow aA \\
A &\rightarrow a \\
B &\rightarrow bB \\
B &\rightarrow b
\end{aligned}
$$