Automatentheorie und Formale Sprachen

gehalten von Prof. Peter Rossmanith
im Sommersemester 2005 an der RWTH Aachen

eine studentische Mitschrift von
Florian Heller
florian@heller-web.net
Diese Mitschrift erhebt keinen Anspruch auf Richtigkeit oder Vollständigkeit.
No RISC - No Fun!

11. September 2005

Diese Mitschrift erhebt keinen Anspruch auf Korrektheit oder Vollständigkeit.
Kommentare und Korrekturvorschläge sind ausdrücklich erwünscht.

Changelog:

Seite 19: $L/_{\equiv_L} = \{0^*, 0^*1^+, (0+1)^*10(0+1)^*\}$

Seite 23: A ist kein Aktz. Zustand

Seite 35: A unerreichbar $\Leftrightarrow\ S \notin pre^*((N \cup T)^* A (N \cup T)^*)$

Seite 35: $\varepsilon \notin L(G)$

Inhaltsverzeichnis

0.1 Organisatorisches 4

1 Alphabete, Wörter und Sprachen **5**

1.1 Alphabete, Wörter und Sprachen 5
1.2 Reguläre Ausdrücke 8

2 Endliche Automaten **10**

2.1 Deterministischer endlicher Automat 10
2.2 Nichtdeterministische endliche Automaten 12
2.2.1 Potenzmengenkonstruktion: 13
2.3 NFAs mit ε-Übergängen 15
2.4 Minimale DFAs und der Satz von Myhill-Nerode 19
2.4.1 Berechnung des minimalen DFA 22
2.5 Regulärer Ausdruck aus einem NFA mit ε-Kanten II 24
2.6 Pumping-Lemma 26
2.7 Entscheidungsproblem für reguläre Sprache 27

3 Kontextfreie Sprachen und Grammatiken **29**

3.1 Ableitungen 29
3.2 Ableitungsbäume 30
3.3 Die pre*-Operation 32
3.4 Entscheidungsprobleme für CFG 34
3.5 Chomsky-Normalform 35
3.6 Greibach-Normalform 37
3.7 Pumping-Lemma für CFL's 40
3.8 Kellerautomaten 41
3.8.1 Deterministische Kellerautomaten 44
3.9 Chomsky-Hierarchie 48

0.1 Organisatorisches

Vorlesung: findet Dienstags und Freitags im Hörsaal Gr statt.

Tutorgruppen:

Montag	8:15 - 9:45	SG12
Montag	10:00 - 11:30	AH2
Montag	11:45 - 13:15	AH1
Montag	11:45 - 13:15	5052 (Spezialübung)
Montag	14:00 - 15:30	SG203
Dienstag	15:45 - 17:15	5052
Mittwoch	11:45 - 13:15	5052

Sprechstunde: 11:00 - 12:00

Literatur

- Hopcroft,Motwani, Ullman
- Algorithmics beauty of plants

1 Alphabete, Wörter und Sprachen

1.1 Alphabete, Wörter und Sprachen

Definition (Alphabet)
Ein Alphabet ist eine endliche Menge von Symbolen

Beispiel:

- $\{a, \ldots, z\}$
- $\{0, 1\}$
- $ASCII$

Was ist ein Wort, was ist eine Sprache?
Informelle Antwort:

1. Ein Wort ist eine Aneinanderkettung von Symbolen.
2. Eine Sprache ist eine Menge von Wörtern.

Beispiele:
01,101001,ε sind Wörter über dem Alphabet $\{0, 1\}$.

Definition (Halbgruppe)
Eine Halbgruppe $(H, \circ)$ besteht aus einer Menge H und einer assoziativen Operation $\circ : H \times H \to H$.

Definition (Monoid)
Ein Monoid ist eine Halbgruppe mit einem neutralen Element

Es sei $(M, \circ)$ ein Monoid und $E \subseteq M$.
E ist ein **Erzeugendensystem** von $(M, \circ)$, falls jedes $m \in M$ als $m = e_1 \circ \cdots \circ e_n$ mit $e_i \in E$ dargestellt werden können.

Beispiele:

- $(\mathbb{Z}, +)$ ist ein Monoid
- $\{-1, 1\}$ ist ein Erzeugendensystem

Frage: Ist $\{-16, 13, 12\}$ ein Erzeugendensystem für $(\mathbb{Z}, +)$?
Ein Erzeugendensystem E für ein Monoid$(M, \circ)$ ist **frei**, falls jedes $m \in M$ auf nur eine Art als $m = e_1 \circ \cdots \circ e_n$ mit $e_i \in E$ dargestellt werden kann.

Beispiele:
$(\mathbb{Z}, +)$ ist von $\{-1, 1\}$ nicht frei erzeugt. $(2 = 1 + 1 = 1 + 1 + (-1) + 1)$
$(\mathbb{N}_0, +)$ ist von $\{1\}$ frei erzeugt.

$$0 = \sum_{i \in \emptyset} 1 \left(= \sum_{i=0}^{-1} a_n = 0 \right)$$

Satz 1.1: Es sei Σ ein Alphabet. Dann ist das von Σ frei erzeugte Monoid bis auf Isomorphismus eindeutig.

Zwei Monoide $(M_1, \cdot)$ und $(M_2, \circ)$ sind isomorph, falls es eine Abbildung $h : M_1 \to M_2$ gibt mit

1. h ist bijektiv
2. h ist ein Homomorphismus d.h. $h(u \cdot v) = h(u) \circ h(v)$ für alle $u, v \in M_1$

Beweis:
Es seien $(M_1, \cdot)$ und $(M_2, \circ)$ zwei von Σ frei erzeugte Monoide.
Wir definieren eine Abbildung $h : M_1 \to M_2$ auf folgende Weise: $h(w) = h(w_1 \cdot \ldots \cdot w_n) = w_1 \circ \ldots \circ w_n$ wobei $w_i \in \Sigma$ ist.
Behauptung: h ist ein Isomorphismus.
Wir zeigen zuerst, dass h bijektiv ist.
Definiere $g : M_2 \to M_1$ vermöge
$g(v) = v_1 \cdot \ldots \cdot v_n$ wobei $v = v_1 \circ \ldots \circ v_n$ mit $v_i \in \Sigma$

$$\begin{aligned} g(h(w)) &= g(h(w_1 \cdot \ldots \cdot w_n)) \\ &= g(w_1 \circ \ldots \circ w_n) \\ &= w_1 \cdot \ldots \cdot w_n = w \end{aligned}$$

Nun ist ausserdem:

$$\begin{aligned} h(u \cdot v) &= h(u_1 \cdot \ldots \cdot u_n \cdot v_1 \cdot \ldots \cdot v_m) \\ &= u_1 \circ \ldots \circ u_n \circ v_1 \circ \ldots \circ v_m \\ &= (u_1 \circ \ldots \circ u_n) \circ (v_1 \circ \ldots \circ v_m) \\ &= h(u) \circ h(v) \end{aligned}$$

Definition
Es sei Σ ein Alphabet. Dann ist $(\Sigma^*, \cdot)$ das von Σ frei erzeugte Monoid.
Die Elemente von Σ^* nennen wir **Wörter** (über Σ).
Falls $L \subseteq \Sigma^*$, dann nennen wir L eine **Sprache** (über Σ)
Das neutrale Element von $(\Sigma^*, \cdot)$ nennen wir ε.

Satz 1.2: Es seien Σ und Γ Alphabete.
Jede Abbildung $\Sigma \to \Gamma^*$ lässt sich eindeutig auf einen Homomorphismus $\Sigma^* \to \Gamma^*$ erweitern.

Beweis:
Es sei $h : \Sigma^* \to \Gamma^*$ ein Homomorphismus.
Dann ist

$$\begin{aligned} h(w) &= h(w_1 \ldots w_n) \text{ mit } w_i \in \Sigma \\ &= h(w_1) \ldots h(w_n) \text{ weil } h \text{ homomorph.} \end{aligned}$$

Beispiel:
$\Sigma = \{a, b\}$
$h(a) = b$
$h(b) = ab$
$h(abba) = bababb$

Definition
Die Länge $|w|$ eines Wortes $w \in \Sigma^*$ ist n falls $w = w_1 \dots w_n$ mit $w_i \in \Sigma$

Definition
Es seien $L_1, L_2 \subseteq \Sigma^*$ Sprachen über Σ.
Dann ist $L_1L_2 := \{uv | u \in L_1, v \in L_2\}$

Definition
Falls $L \subseteq \Sigma^*$, dann ist

- $L^0 := \{\varepsilon\}$
- $L^1 := L$
- $L^n := LL^{n-1}$ für $n \geq 1$
- $$L^* := \bigcup_{n \geq 0} L^n$$
  L^* wird die **Kleenesche Hülle** genannt.
- $r^+ = rr^*$
- $$L^+ = \bigcup_{n \geq 1} L^n = LL^*$$

Beispiele

1. $\emptyset^* = \{\varepsilon\}$
2. $\emptyset\{ab, \varepsilon\} = \emptyset$
3. $L = \emptyset^* \qquad L = \{\varepsilon\}$
4. $\varepsilon\emptyset = \emptyset$
5. $(a + b)^* = (a^*b^*)^*$

1.2 Reguläre Ausdrücke

Ein regulärer Ausdruck ist eine kurze, prägnante Schreibweise für "einfache" Sprachen. Reguläre Ausdrücke eignen sich besonders für Ein-/Ausgabe.
Wir definieren reguläre Ausdrücke über Σ induktiv:
Es sei Σ ein Alphabet.

1. $\emptyset$ ist ein regulärer Ausdruck.
2. ε ist ein regulärer Ausdruck
3. a ist ein regulärer Ausdruck für $a \in \Sigma$
4. rs ist ein regulärer Ausdruck, falls r, s reguläre Ausdrücke sind.
5. $r + s$ ist ein regulärer Ausdruck, falls r, s reguläre Ausdrücke sind.
6. r^* ist ein regulärer Ausdruck, falls r ein regulärer Ausdruck ist.

Wir ordnen jedem regulären Ausdruck r eine Sprache $L(r)$ zu.

1. $L(\emptyset) = \emptyset$
2. $L(\varepsilon) = \{\varepsilon\}$
3. $L(a) = \{a\}$
4. $L(rs) = L(r)L(s)$
5. $L(r + s) = L(r) \vee L(s)$
6. $L(r^*) = L(r)^*$

Beispiele:

1. $(A + \cdots + Z + a + \cdots + z)^* \ (O + o)tto(A + \cdots + Z + a + \cdots + z)^*$
2. $0^*(10^*1)^*0^*$
   Wörter über $\{0, 1\}$ mit gerader Anzahl von 1en (der Stern in der Klammer bezieht sich nur auf die Null (bindet stärker))
   $= 0^*(10^*10^*)^*$
3. 1^*0^*
   Wörter, die nicht 01 als Unterwort enthalten
4. $(0 + 11 + (10(1 + 00)^*01)^*$ Binärzahlen, die durch 3 teilbar sind.

19.04.05

Satz 1.3: Es seien $A, B, C \subseteq \Sigma^*$
$A(BC) = (AB)C$
$\{\varepsilon\}A = A\{\varepsilon\} = A$
$(A^*)^* = A^*$
$A(B \cup C) = AB \cup AC$
$(A \cup B)C = AC \cup BC$
$A^* = \{\varepsilon\} \cup A^+$

Definition (reguläre Sprache)
Eine Sprache $L \subseteq \Sigma^*$ ist **regulär**, falls es einen regulären Ausdruck r über Σ gibt mit $L(r) = L$.
Die Klasse der regulären Sprachen besteht aus allen regulären Sprachen über allen Alphabeten.

Satz 1.4: Reguläre Sprachen sind unter Vereinigung, Konkatenation, Kleene'scher Hülle und Homomorphismen abgeschlossen.
Das heißt: Falls A, B regulär sind, dann auch:

- $A \cup B$
- AB
- A^*
- $h(A)$ für Homomorphismus $h : \Sigma^* \to \Gamma^*$

Beweis:
A, B regulär
$\Rightarrow A = L(r_A),\ B = L(r_B)$

$$\begin{aligned} A \cup B &= L(r_A + r_B) \\ AB &= L(r_A r_B) \\ A^* &= L(r_A)^* \end{aligned}$$

Nun zu h:

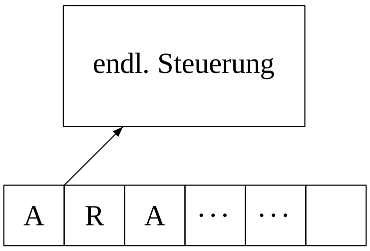
1. $r_A = \emptyset \Rightarrow A = \emptyset \Rightarrow h(A) = \emptyset$ (klar)
2. $r_A = \varepsilon \Rightarrow A = \{\varepsilon\} \Rightarrow h(A) = \{\varepsilon\}$ (klar)
3. $r_A = a \in \Sigma \Rightarrow A = \{a\} \Rightarrow h(A) = \{b_1, \ldots, b_n\}$
4. $r_A = r_B + r_C \Rightarrow h(A) = h(B) \cup h(C)$
   I.V. $\Rightarrow h(B), h(C)$ regulär
5. ...
6. ...

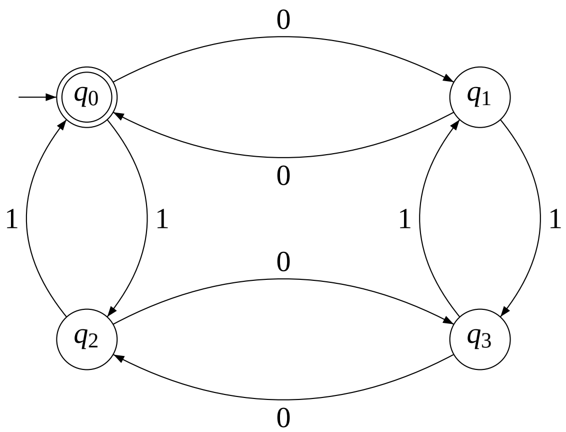
Abschluss unter Schnitt lässt sich so nicht beweisen.

Fragestellungen:

1. **Wortproblem:**
   Eingabe: reguläre Sprache L, Wort w.
   Frage: $w \in L$?
2. **Sprachequivalenz**
   Eingabe: reguläre Sprachen L_1, L_2
   Frage: $L_1 \stackrel{?}{=} L_2$
3. **Schnittleerheitsproblem:**
   Eingabe: reguläre Sprachen L_1, L_2
   Frage: $L_1 \cap L_2 = \emptyset$

2 Endliche Automaten





2.1 Deterministischer endlicher Automat

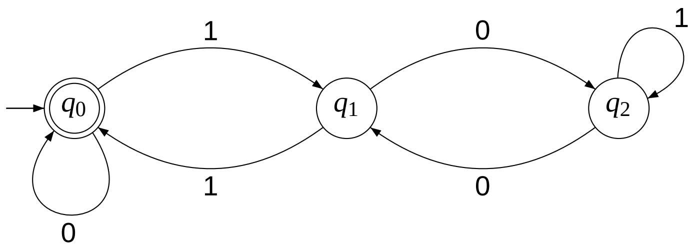
Definition (Deterministischer endlicher Automat)
Ein deterministischer endlicher Automat (DFA) ist ein 5-Tupel $M = (Q, \Sigma, \delta, q_0, F)$ wobei

- Q eine endliche Menge von Zuständen
- Σ das Eingabealphabet
- $\delta : Q \times \Sigma \to Q$ die Übergangsfunktion
- $q_0 \in Q$ der Startzustand und
- $F \subseteq Q$ die Endzustände sind.

Erweitere δ zu $\hat{\delta} : Q \times \Sigma^* \to Q$ mittels

- $\hat{\delta}(q, \varepsilon) = q$
- $\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a)w)$ (mit $a \in \Sigma, w \in \Sigma^*$)

Die Sprache, die M akzeptiert, ist $L(M) := \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}$



M

$M = (Q, \Sigma, \delta, q_0, F)$
$Q = \{q_0, q_1, q_2\}$
$F = \{q_0\}$
$\Sigma = \{0, 1\}$
$\delta : Q \times \Sigma \to Q$
$(q_0, 0) \mapsto q_0 \quad (q_0, 1) \mapsto q_1$
$(q_1, 0) \mapsto q_2 \quad (q_1, 1) \mapsto q_0$
$(q_2, 0) \mapsto q_1 \quad (q_2, 1) \mapsto q_2$

Zum Beispiel:
$\hat{\delta}(q_0, 010) = \hat{\delta}(\delta(q_0, 0), 10) = \hat{\delta}(q_0, 10) = \hat{\delta}(q_1, 0) = \hat{\delta}(q_2, \varepsilon) = q_2$ 22.4.05

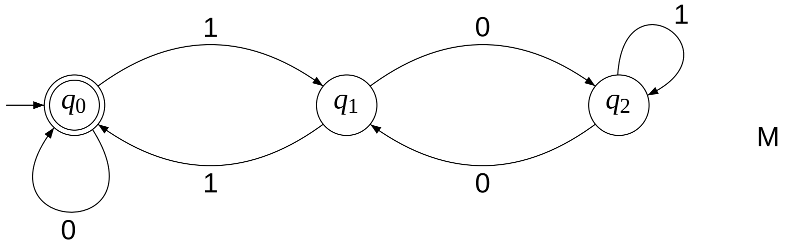
Satz 2.1: Es sei M ein endlicher Automat (DFA).
Dann ist $L(M)$ regulär.

Beweis:
$M = (Q, \Sigma, \delta, q_0, F)$ mit o.B.d.A. $Q = \{q_0, \ldots, q_{n-1}\}$
$L^k_{ij} := \{w \in \Sigma^* \mid \hat{\delta}(q_i, w) = q_j$ und $\hat{\delta}(q_i, u) = q_m$ mit $m \leq k \ \ \forall u = w_1 \ldots w_r$ mit $0 < r < |w|\}$
”=” Wörter die M von q_i nach q_j führen über Zwischenzustände $\in \{q_0, \ldots, q_k\}$.
Dann ist L(M)= $\bigcup_{q_j \in F} L^n_{0j}$
Beispiel: $L^0_{00} = 0^*$ $\quad L^1_{00} = (0 + 11)^*$ $\quad L^0_{02} = \emptyset$ $\quad L^1_{02} = (0 + 1)^* 10$
Behauptung: $L^k_{ij} = L^{k-1}_{ij} \cup L^{k-1}_{ik} \left(L^{k-1}_{kk}\right)^* L^{k-1}_{kj}$
$R^k_{ij} = R^{k-1}_{ij} + R^{k-1}_{ik} \left(R^{k-1}_{kk}\right)^* R^{k-1}_{kj}$
$L^{-1}_{ij} = \{a \in \Sigma \mid \delta(q_i, a) = q_j\}$ für $i \neq j$
$L^{-1}_{ii} = \{a \in \Sigma \mid \delta(q_i, a) = q_i\} \cup \{\varepsilon\}$

$$R^{-1}_{ij} = \begin{cases} \sum_{a \in \Sigma, \delta(q_i, a) = q_j} a & \text{falls } i \neq j \\ \sum_{a \in \Sigma, \delta(q_i, a) = q_j} a + \varepsilon & \text{falls } i = j \end{cases}$$

$R = \sum_{q_j \in F} R^n_{0j} \Rightarrow L(R) = L(M)$
26.4.05

Beispiel:



$$L(M) = L^2_{00} = L^1_{00} \cup L^1_{02}(L^1_{22})^* L^1_{20}$$

$$\begin{aligned}
L^1_{00} &= L^0_{00} \cup L^0_{01}(L^0_{11})^* L^0_{10} = 0^* \cup 0^*1(10^*1)^*10^* \\
L^0_{00} &= 0^* \\
L^0_{01} &= 0^*1 \\
L^0_{11} &= 10^*1 + \varepsilon \\
L^0_{10} &= 10^*
\end{aligned}$$

$$\begin{aligned}
L^1_{02} &= L^0_{02} \cup L^0_{01}(L^0_{11})^* L^0_{12} = 0^*1(10^*1)^*0 \\
L^0_{02} &= \emptyset \\
L^0_{01} &= 0^*1 \\
L^0_{12} &= 0
\end{aligned}$$

$$\begin{aligned}
L^1_{22} &= L^0_{22} \cup L^0_{21}(L^0_{11})^* L^0_{12} = \varepsilon + 1 \cup 0(10^*1)^*0 \\
L^0_{22} &= 1 + \varepsilon \\
L^0_{21} &= 0
\end{aligned}$$

$$\begin{aligned}
L^1_{20} &= L^0_{20} \cup L^0_{21}(L^0_{11})^* L^0_{10} = 0(10^*1)^*10^* \\
L^0_{20} &= \emptyset
\end{aligned}$$

$$L(M) = 0^* + 0^*1(10^*1)^*10^* + 0^*1(10^*1)^*0(1 + 0(10^*1)^*0)^*0(10^*1)^*10^*$$

2.2 Nichtdeterministische endliche Automaten

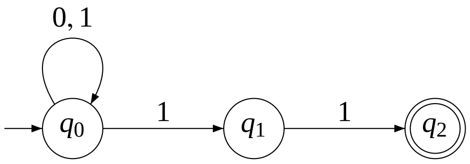
Definition
Ein NFA ist ein 5-Tupel $M = (\Sigma, Q, \delta, q_0, F)$ mit

- Σ: Eingabealphabet
- Q: endliche Menge von Zuständen

- $\delta : Q \times \Sigma \to 2^Q$
- $q_0 \in Q$ Startzustand
- $F \subseteq Q$ Endzustände

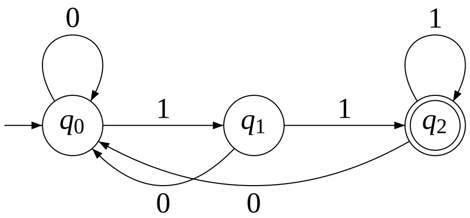
$\hat{\delta} : Q \times \Sigma^* \to 2^Q$
Definition durch:
$\hat{\delta}(q, \varepsilon) = \{q\}$
$\hat{\delta}(q, aw) = \{p \mid$ Es gibt $r \in Q$ mit $r \in \delta(q, a)$ und $p \in \hat{\delta}(r, w)\}$
$L(M) := \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$

Beispiel:

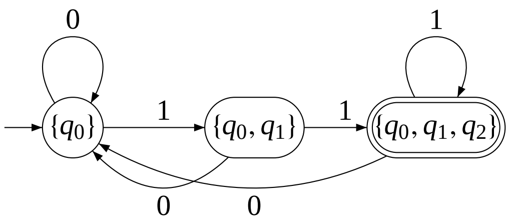


$\delta(q_0, 1) = \{q_0, q_1\}$
$\delta(q_2, 0) = \emptyset$
$\hat{\delta}(q_0, 1011011001) = \{q_0, q_1\}$
$\hat{\delta}(q_1, 0) = \emptyset$
$L(M) = (0 + 1)^* 11$

Gibt es DFA M' mit $L(M') = L(M)$?



1011
$\{q_0\}$ 1 $\{q_0, q_1\}$ 0 $\{q_0\}$ 1 $\{q_0, q_1\}$ 1 $\{q_0, q_1, q_2\}$



29.4.05

2.2.1 Potenzmengenkonstruktion:

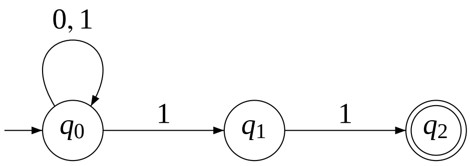
Es sei $M = (Q, \Sigma, \delta, q_o, F)$ ein NFA.
Der zugehörige **Potenzautomat** M' ist so aufgebaut:

$M' = (2^Q, \Sigma, \delta', \{q_0\}, F')$
$\delta' : 2^Q \times \Sigma \to 2^Q \qquad (S, a) \mapsto \bigcup_{q \in S} \delta(q, a)$
$F' = \{S \subseteq Q \mid S \cap F \neq \emptyset\}$

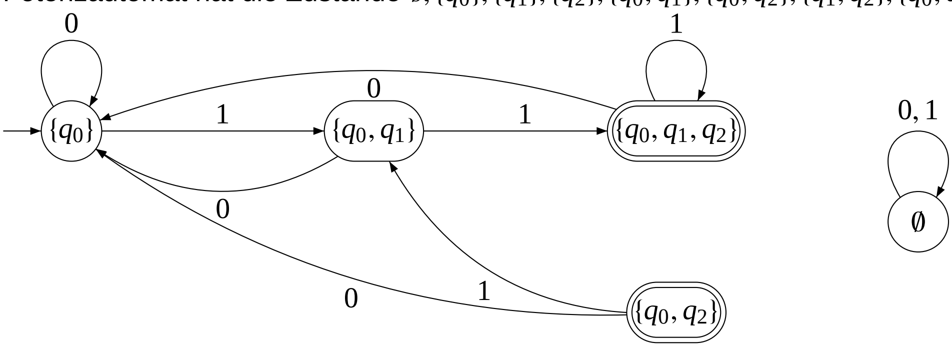
Satz 2.2: Zu jedem NFA M gibt es einen DFA M' mit $L(M) = L(M')$

Beweis (idee):
$\hat{\delta}(q_0, w) \in F \Leftrightarrow \hat{\delta}'(\{q_0\}, w) \in F'$ zu zeigen mit Induktion über $|w|$
I.A. $|w| = 0, w = \varepsilon$.
Linke Seite: $q_0 \in F$ Rechte Seite: $\{q_0\} \in F' \Leftrightarrow \{q_0\} \cap F \neq \emptyset \Rightarrow q_0 \in F$
I.S.: ...

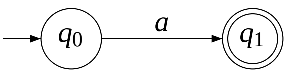
Beispiel:



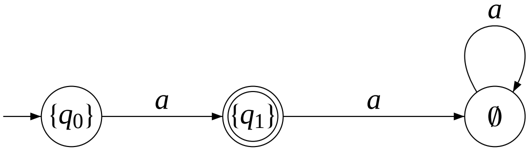
Der Potenzautomat hat die Zustände $\emptyset, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}$.



Beispiel:



$\delta(q_0, a) = \{q_1\}$
$\delta(q_1, a) = \emptyset$



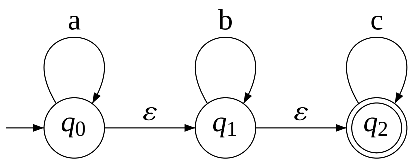
Vorteile des DFA:

- Effizient simulierbar

Vorteile des NFA:

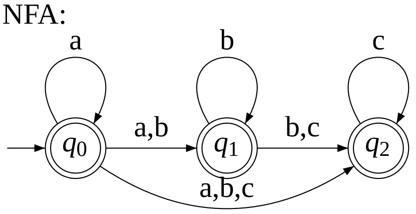
- Oft kleiner als DFA
- Einfacher zu entwerfen 6.5.05

2.3 NFAs mit ε-Übergängen



M

$L(M) = a^*b^*c^*$
Als NFA:



M'

Definition (NFA mit ε-Übergängen)
Ein NFA mit ε-Übergängen ist ein 5-Tupel $M = (Q, \Sigma, \delta, q_0, F)$ mit

- $\delta : Q \times \Sigma \cup \{\varepsilon\}$
- ε-Hülle$(q) := \{p \in Q \mid$ es gibt mit εs markierten Pfad von q nach $p\}$
- Wir definieren $\hat{\delta} : Q \times \Sigma^* \to 2^Q$:
  $\hat{\delta}(q, \varepsilon) := \varepsilon$-Hülle(q)
  $\hat{\delta}(q, wa) := P$
  mit $P = \{p \in Q \mid$ es gibt $r \in \hat{\delta}(q, w)$ und $p \in \delta(r, a)\}$

Beispiel:
$\hat{\delta}(q_0, b) = \varepsilon$-Hülle$(P) = \{q_1, q_2\}$
mit $P = \{p \in Q \mid$ es gilt $r \in \hat{\delta}(q_0, \varepsilon\} = \varepsilon$-Hülle$(q_0) = \{q_0, q_1, q_2\}$ und $p \in \delta(r, b)\}$

Wir können ε-Kanten entfernen, ohne die akzeptierte Sprache zu verändern, falls wir δ durch δ' und F durch F' ersetzen:

$$\delta'(q,a) = \bigcup_{p \in \varepsilon\text{-Hülle}(q)} \varepsilon\text{-Hülle}(\delta(p,a))$$

$$F' = \{p \in Q \mid \varepsilon\text{-Hülle}(p) \cap F \neq \emptyset\}$$

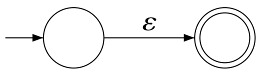
Satz 2.3: Zu jeder regulären Sprache L gibt es einen NFA mit ε-Übergängen, der L akzeptiert.

Beweis:
Spezielle Form: Nur ein Endzustand ohne ausgehende Kanten
Es sei $L = L(r)$ für einen regulären Ausdruck r

1. $r = \emptyset$

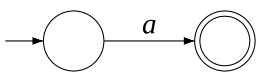
$$L(M) = \emptyset$$

2. $r = \varepsilon$



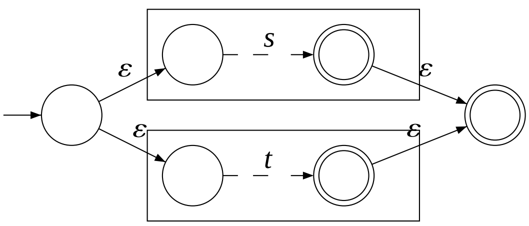
$$L(M) = \{\varepsilon\}$$

3. $r = a$



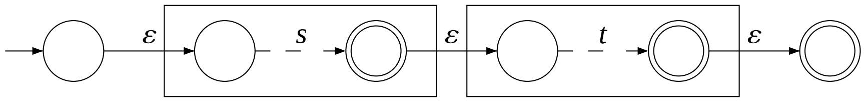
$$L(M) = \{a\}$$

4. $r = s + t$



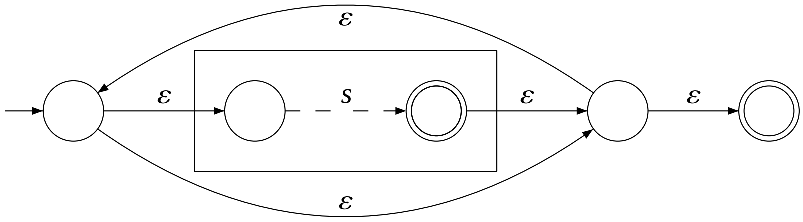
$$L(M) = L(s) \vee L(t)$$

5. $r = st$



$L(M) = L(s)L(t)$

6. $r = s^*$



Korollar 1 *Reguläre Ausdrücke, DFAs, NFAs und NFAs mit ε-Übergängen charakterisieren alle genau die reg. Sprachen.*

Satz 2.4: Reguläre Sprachen sind unter Komplement, Differenz und Schnitt abgeschlossen.

Beweis:
Komplement: Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA, $M' = (Q, \Sigma, \delta, q_0, Q - F)$
Es gilt:

$$\begin{aligned} L(M') &= \Sigma^* - L(M) \\ w \in L(M') &\Leftrightarrow \hat{\delta}(q_0, w) \in Q - F \\ &\Leftrightarrow \hat{\delta}(q_0, w) \notin F \\ &\Leftrightarrow w \notin L(M) \\ &\Leftrightarrow w \in \Sigma^* - L(M) \end{aligned}$$

Schnitt: Es seien $M' = (Q', \Sigma, \delta', q_0', F')$ und $M'' = (Q'', \Sigma, \delta'', q_0'', F'')$ zwei DFAs.
Wir definieren den **Produktautomaten** $M = M' \times M''$ vermöge

$$M = \left(\underbrace{Q' \times Q''}_{\text{Zustände sind Tupel}}, \Sigma, \delta, (q_0', q_0''), F' \times F'' \right)$$

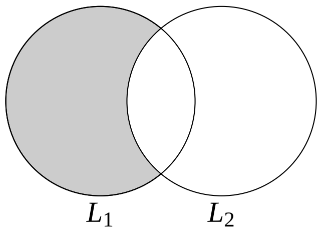
$$\delta((q, p), a) := (\delta'(q, a), \delta''(p, a))$$

Offensichtlich gilt:

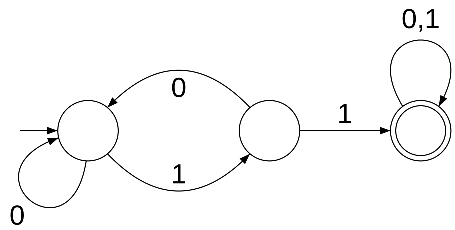
$$\begin{aligned} \hat{\delta}((q_0', q_0''), w) &= (\hat{\delta}'(q_0', w), \hat{\delta}''(q_0'', w)) \\ \text{und damit } w \in L(M) &\Leftrightarrow \hat{\delta}((q_0', q_o''), w) \in F' \times F'' \\ &\Leftrightarrow (\hat{\delta}'(q_0', w), \hat{\delta}''(q_0'', w)) \in F' \times F'' \\ &\Leftrightarrow \hat{\delta}'(q_0', w) \in F' \text{ und } \hat{\delta}''(q_0'', w) \in F'' \\ &\Leftrightarrow w \in L(M') \text{ und } w \in L(M'') \\ &\Leftrightarrow w \in L(M') \cap L(M'') \end{aligned}$$

Differenz: folgt aus Komplement und Schnitt:

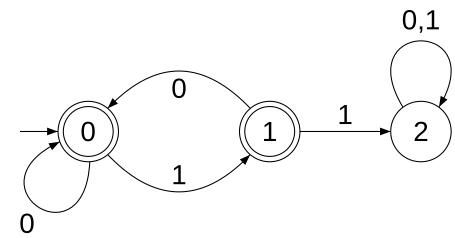
$$L_1 - L_2 = L_1 \cap (\Sigma^* - L_2)$$



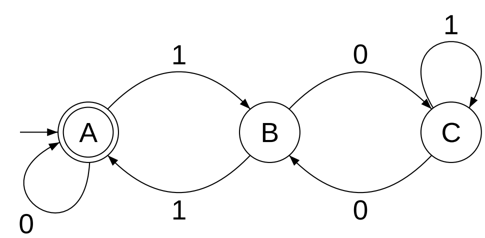
Beispiel:
Binärwörter die durch 3 teilbar sind, aber 11 nicht als Unterwort enthalten.
11 als Unterwort:



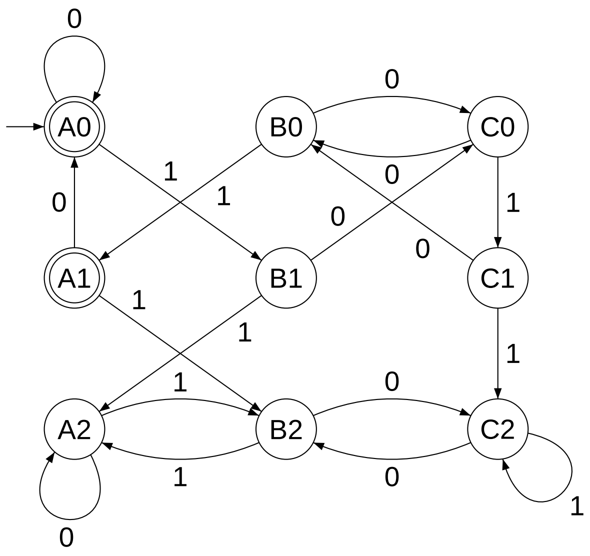
11 nicht als Unterwort M_1:



Durch 3 teilbar M_2:



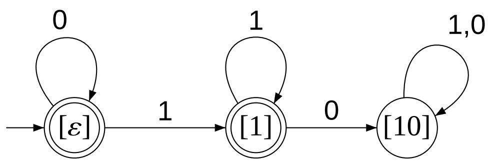
Produktautomat $M_1 \times M_2$:



2.4 Minimale DFAs und der Satz von Myhill-Nerode

Definition
Es sei $L \subseteq \Sigma^*$ eine (nicht zwingend reguläre) Sprache.
Wir definieren die Äquivalenzrelation $\equiv_L \subseteq \Sigma^* \times \Sigma^*$ auf Wörtern vermöge
$u \equiv_L v$ g.d.w. $uw \in L \Leftrightarrow vw \in L$ für alle $w \in \Sigma^*$.
Der Index einer Äquivalenzrelation ist die Anzahl der Äquivalenzklassen

Beispiel:
Es sei $L = 0^*1^*$



$L/_{\equiv_L} = \{0^*, 0^*1^+, (0+1)^*10(0+1)^*\}$

$$\begin{aligned} 001 &\equiv_L 0111 \\ 010 &\not\equiv_L 0111 \text{ , da } 010 \notin L \\ 00 &\not\equiv_L 00001 \text{, da } 00|0 \in L\text{, aber } 00001|0 \notin L \end{aligned}$$

Was ist der Index von L?

$$\begin{aligned} 0^* &= [\varepsilon]_{\equiv_L} \\ 0^*1^+ &= [1]_{\equiv_L} \\ 0^*1^+0(0+1)^* &= [10]_{\equiv_L} \end{aligned}$$

Satz 2.5 (Myhill-Nerode): $L \subseteq \Sigma^*$ ist genau dann regulär wenn $\equiv_L$ endlichen Index hat.

Beweis:

"$\Rightarrow$" Es sei $L \subseteq \Sigma^*$ regulär. Dann gibt es DFA $M = (Q, \Sigma, \delta, q_0, F)$ mit $L(M) = L$.
Wir definieren $u \sim v$ gdw. $\hat{\delta}(q_0, u) = \hat{\delta}(q_0, v)$
Offensichtlich $u \sim v \Rightarrow u \equiv_L v$, denn

$$\begin{aligned} uw \in L &\Leftrightarrow \hat{\delta}(q_0, uw) \in F \\ &\Leftrightarrow \hat{\delta}(\hat{\delta}(q_0, u), w) \in F \\ &\Leftrightarrow \hat{\delta}(\hat{\delta}(q, v), w) \in F \\ &\Leftrightarrow \hat{\delta}(q_0, vw) \in F \\ &\Leftrightarrow vw \in L \end{aligned}$$

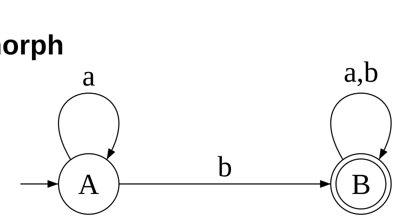
Also hat $\sim$ mindestens so viele Äquivalenzklassen wie $\equiv_L$.
Der Index von $\sim$ ist endlich (höchstens $|Q|$), daher auch der Index von $\equiv_L$

"$\Leftarrow$" $L \subseteq \Sigma^*$ und Index von $\equiv_L$ endlich.
Wir konstruieren folgenden DFA: $M = (Q, \Sigma, \delta, [\varepsilon]_{\equiv_L}, F)$ mit $Q = \{[w]_{\equiv_L} | w \in \Sigma^*\}$.
$\delta : Q \times \Sigma \to Q$, $([w]_{\equiv_L}, a) \mapsto [wa]_{\equiv_L}$
$F = \{[w]_{\equiv_L} | w \in L\}$
Q ist endlich!
δ wohldefiniert, denn $[u]_{\equiv_L} = [v]_{\equiv_L} \Rightarrow [ua]_{\equiv_L} = [va]_{\equiv_L}$.
Behauptung: $L = L(M)$.

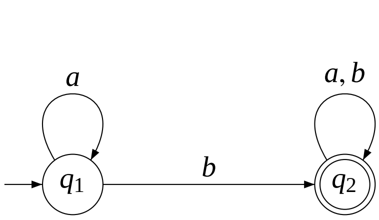
$$\begin{aligned} w \in L(M) &\Leftrightarrow \hat{\delta}([\varepsilon]_{\equiv_L}, w) \in F \\ &= [w]_{\equiv_L} \in F \\ &\Leftrightarrow w \in L \\ &\Rightarrow L \text{ regulär} \end{aligned}$$

Sprachäquivalenz und Isomorphie

Isomorph



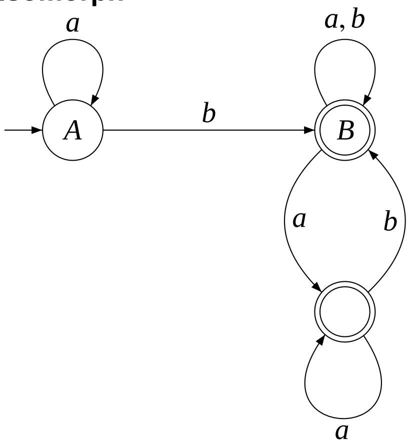
$M_1 = (\{A, B\}, \{a, b\}, \dots)$

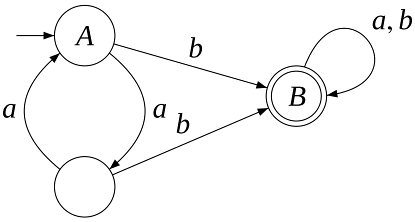


$M_2 = (\{q_1, q_2\}, \dots)$

M_1, M_2 isomorph

Nicht Isomorph





$a^* b(a+b)^*$

Definition
Seien $M = (Q, \Sigma, \delta, q_0, F)$ und $M' = (Q', \Sigma, \delta', q_0', F')$ zwei DFAs. M und M' sind **isomorph** gdw. es eine Abbildung $h : Q \to Q'$ gibt mit

- $h(q_0) = q_0'$
- $h(q) \in F' \Leftrightarrow q \in F$
- $h(\delta(q, a)) = \delta'(h(q), a)$
- h bijektiv

Satz 2.6: Falls $L(M_1) = L(M_2)$ und M_1, M_2 haben eine minimale Anzahl von Zuständen, dann sind M_1 und M_2 zueinander isomorph.

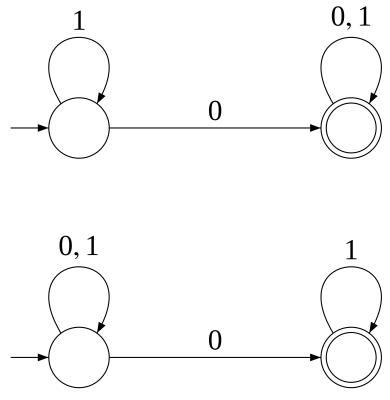
Beweis:
Beh.: M ist isomorph zum Myhill-Nerode-DFA von $L(M)$, falls M minimal ist.
Falls $\hat{\delta}(q_0, w) = q$, dann $h(q) = [w]_{\equiv_L}$
h ist Isomorphismus.

Mit Myhill-Nerode können wir zeigen, dass eine Sprache nicht regulär ist.

Beispiel:
$L = \{a^n b^n | n \geq 0\} = \{\varepsilon, ab, aabb, aaabbb, \dots\}$
$a^n \not\equiv_L a^m$,falls $n \neq m$
a^n, a^m in verschiedene Äquivalenzklassen
$\Rightarrow$ Index von $\equiv_L = \infty$

- $L = \{a^{2^n} | n \geq 0\}$
  $a^{2^n} \not\equiv_L a^{2^m}$ mit $n < m$, denn $a^{2^n} a^{2^m} \notin L$, aber $a^{2^m} a^{2^m} \in L$
- $L = a^* bb$
  $a^n \not\equiv_L a^m b$, denn $a^n bb \in L, a^m bbb \notin L$
  $\Rightarrow [a^n]_{\equiv_L} \neq [a^m b]_{\equiv_L}$

Bemerkung:



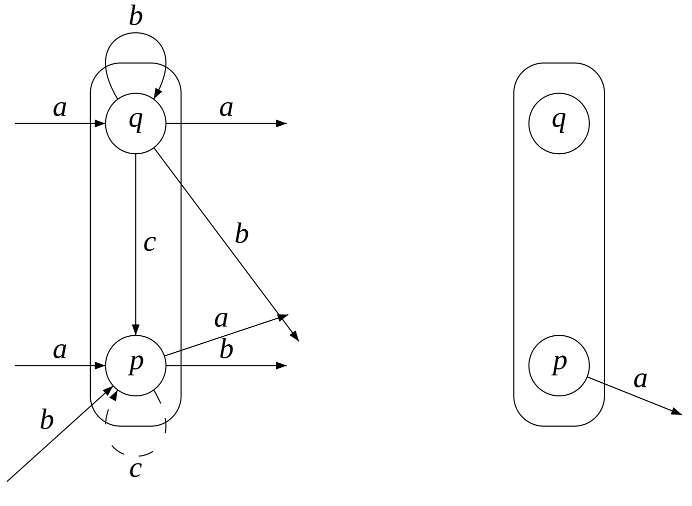
NFA

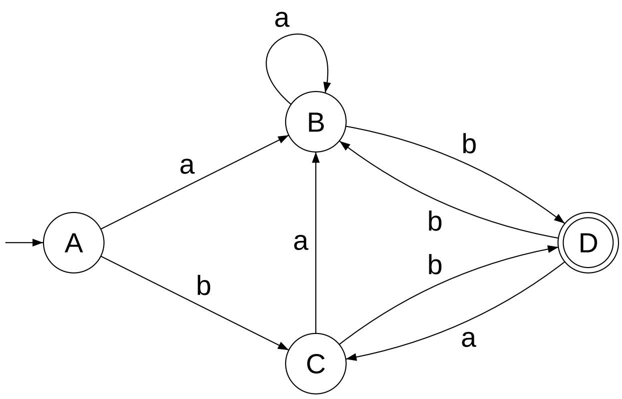
2.4.1 Berechnung des minimalen DFA

(Myhill-Nerode nicht Konstruktiv)
Problem: Minimierung von DFAs
Eingabe: DFA M
Ausgabe: Zugehörigen minimalen DFA

Definition
Zwei Zustände q und p (bez. eines DFAs) sind **unterscheidbar**, falls $q \in F \not\Leftrightarrow p \in F$ oder falls $\delta(q, a)$ und $\delta(p, a)$ für ein $a \in \Sigma$ unterscheidbar.

Beweis:
Offensichtlich sind p, q genau dann unterscheidbar, wenn es ein $w \in \Sigma^*$ gibt, mit $\hat{\delta}(q, w) \in F \not\Leftrightarrow \hat{\delta}(p, w) \in F$.
Dann gilt auch $u \not\equiv_L v$ für $u, v \in \Sigma^*$ mit $\hat{\delta}(q_0, u) = q$ und $\hat{\delta}(q_0, v) = p$
Umgekehrt ist $u \equiv_L v$, falls q und p nicht unterscheidbar sind.
$\Rightarrow$ Nicht unterscheidbare Zustände können "verschmolzen" werden.

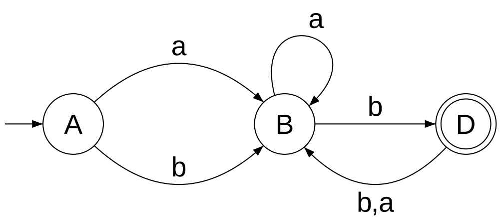




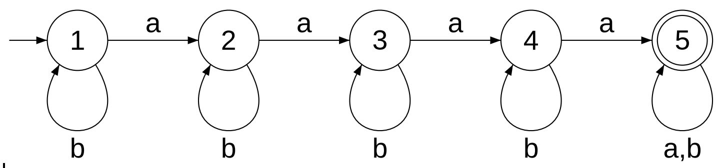
Beispiel:
Markierungsalgorithmus

A				
B	X			
C	X			
D	X	X	X	
	A	B	C	D

Wird zu :



Beispiel:

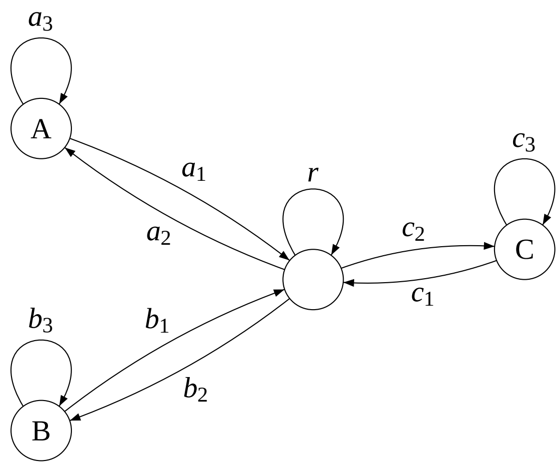


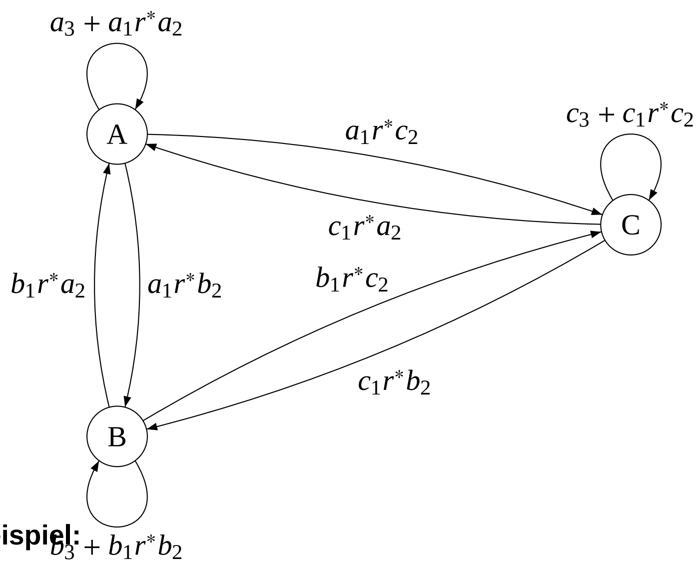
1					
2	X				
3	X	X			
4	X	X	X		
5	X	X	X	X	
	1	2	3	4	5

oder Optimiert:

1					
2	X				
3	X	X $\{2,1\}$			
4	X	X $\{3,1\}$	X $\{3,2\}$		
5	X	X	X	X	
	1	2	3	4	5

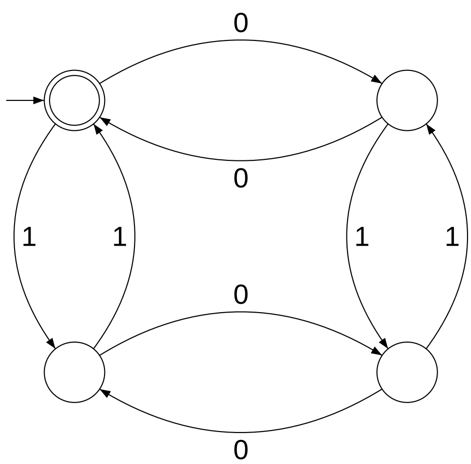
2.5 Regulärer Ausdruck aus einem NFA mit ε-Kanten II



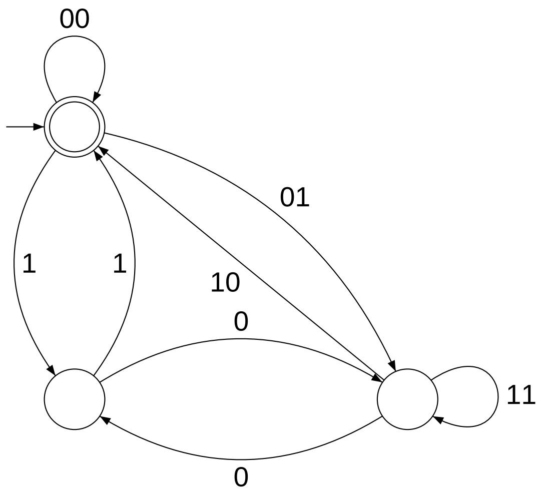


31.5.05

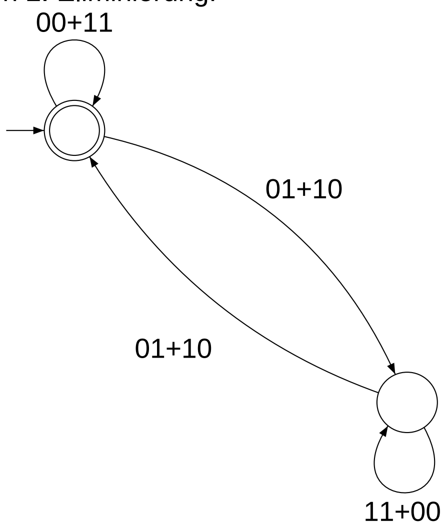
Beispiel:



Nach 1. Eliminierung:

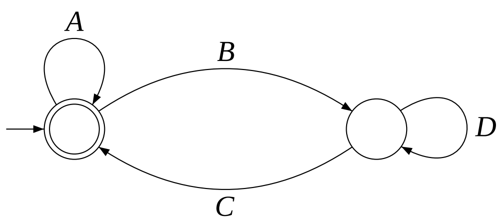


Nach 2. Eliminierung:

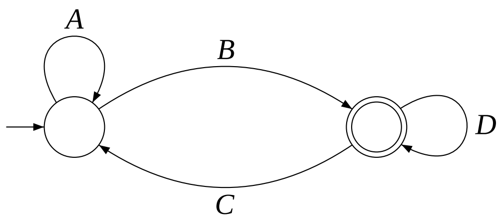


$((00 + 11)^* + (01 + 10)(11 + 00)^*(01 + 10))^*$
$= (00 + 11 + (01 + 10)(00 + 11)^*(01 + 10))^*$

Beispiel:



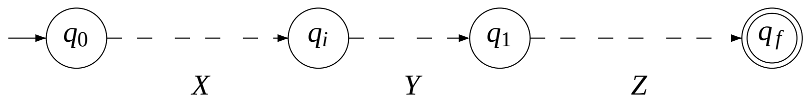
$= (A + BD^*C)^*$



$= A^*B(D + CA^*B)^*$

2.6 Pumping-Lemma

Sei M ein DFA mit n Zuständen und $w \in L(M)$ mit $|w| > n + 1$



$w = XYZ$
$w \in L(M)$, aber auch $XZ, XYZ, XY^2Z, XY^3Z, \cdots \in L(M)$

Lemma
Sei M ein DFA mit n Zuständen, dann kann man jedes Wort $w \in L(M)$ mit $|w| > n + 1$ in $w = XYZ$ Zerlegen mit:

1. $|XY| \leq 2n$
2. $|Y| > 0$
3. $XY^iZ \in L(M)$ für alle $i > 0$

Satz 2.7 (Pumping-Lemma): Sei L eine reguläre Sprache. Dann gibt es eine Zahl n, so dass jedes Wort $w \in L$ mit $|w| \geq n$ in Wörter $w = XYZ$ zerlegt werden kann, mit

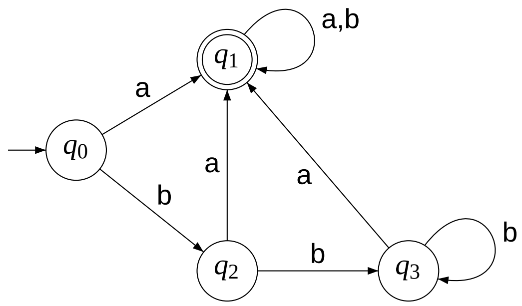
1. $|XY| \leq n$
2. $|Y| > 0$
3. $XY^iZ \in L$ für alle $i \geq 0$

Beispiel:
Sei $L = \{a^kb^k|k \geq 0|\}$
Annahme: Sei n die Konstante vom Pumping-Lemma.
Wähle $w = a^nb^n$ oder $a^{100}b^{100}$ falls $n < 100$
Sei $w = xyz$ die Zerlegung, die es angeblich gibt. $x = a^j$, $y = a^k$, $z = a^{n-j-k}b^n$ für gewisse j, k, denn $|xy| \leq n$
$xy^{n^2}z = a^{j+kn^2+n-j-k}b^n \notin L$ denn $kn^2 + n - k > n$ denn $k > 0$

Beispiel:
$L = \{a^p | p$ Primzahl$\}$
Annahme: L ist regulär.
Sei n die Konstante des Pumping-Lemmas. Wähle $w = a^{p_n}$ mit p_n ist die n-te Primzahl.
Sei $w = xyz$ mit $|xy| \leq n, |y| > 0, xy^iz \in L$
$x = a^{n_x}, y = a^{n_y}, z = a^{n_z}$
$w = a^{n_x+n_y+n_z}, p_n = n_x + n_y + n_z$
$\Rightarrow n_x + in_y + n_z$ ist prim für jedes $i \geq 0$
Sei $i = n_x + n_z$, d.h. $n_x + (n_x + n_z)n_y + n_z = (n_x + n_z)(n_y + 1)$

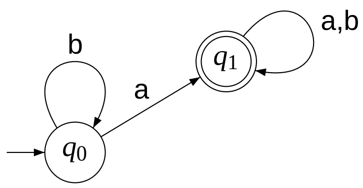
3.6.05

Hinweis: Der Daniel Mölle hat gerade gesagt, es wäre äußerst sinvoll an den Übungen Teilzunehmen.
Beispiel:
Aus der Großübung



q_0				
q_1	X			
q_2		X		
q_3		X	$\{0,2\},\{0,3\}$	
	q_0	q_1	q_2	q_3

X: unterscheidbar



Minimierter Automat:

2.7 Entscheidungsproblem für reguläre Sprache

Theorem 1 *Folgende Probleme sind in polynomieller Zeit lösbar falls M_1 und M_2 DFAs sind:*

1. *Ist $L(M_1) = L(M_2)$?*

2. *Ist* $L(M_1) \cap L(M_2) = \emptyset$*?*

3. *Ist* $L(M_1) \subseteq L(M_2)$*?*

Beweis:

1. Minimieren, isomorph?
2. Produktautomat
3. Test $L(M_1) \cap \overline{L(M_2)} = \emptyset$

Für NFAs leider schwieriger, z.B. ist 1. PSPACE-vollständig

3 Kontextfreie Sprachen und Grammatiken

Definition
Eine Kontextfreie Grammatik ist ein 4-Tupel (N, T, P, S), wobei

N Alphabet, Nonterminale

T Alphabet, Terminalsymbole

P Menge von Produktionen der Form $A \to \alpha$ mit $A \in N, \alpha \in (N \cup T)^*$

S $S \in N$, das Startsymbol

Satzformen: $(N \cup T)^*$ wobei $N \cap T = \emptyset$
Notation: Wir verwenden oft

a,b,c,. . .	für Terminalsymbole
A,B,C, . . .	für Nonterminale
u,v,w,. . .	für Terminalwörter $\in T^*$
$\alpha, \beta, \gamma, \ldots$	für Satzformen

Beispiel:
$G = (\{E, A\}, \{0, 1, +, -, (,)\}, P, E)$ wobei

$$
\begin{aligned}
P = \{ \quad & E \to (E + E) \\
& E \to (E - E) \\
& E \to A \\
& A \to 0 \\
& A \to 1 \\
& A \to AA\}
\end{aligned}
$$

3.1 Ableitungen

Wir definieren die Relation $\Rightarrow_G \subseteq (N \cup T)^* \times (N \cup T)^*$
vermöge $\alpha A\beta \Rightarrow_G \alpha\gamma\beta$ falls $A \to \gamma \in P$ z.B. $E \Rightarrow (E+E) \Rightarrow (E+A) \Rightarrow (A+A) \Rightarrow (AA+A) \Rightarrow (A0+A) \Rightarrow (A0+1) \Rightarrow (10+1)$
$\overset{*}{\Rightarrow}$ ist die reflexiv-transitive Hülle, d.h. $\alpha \overset{*}{\Rightarrow} \beta$ falls $\alpha \Rightarrow \cdots \Rightarrow \beta$ gilt im Besonderen $\alpha \overset{*}{\Rightarrow} \alpha$

Wir definieren $L(G) = \{w \in T^* | S \overset{*}{\Rightarrow}_G w\}$ für $G = (N, T, P, S)$. 7.6.05

$S \to aA$
$aASb \Rightarrow aAaAb$
$S \overset{*}{\Rightarrow} w \in T^*$
$w \in L(G)$
$S \to aSSb, S \to \varepsilon$

$$\begin{aligned} S &\Rightarrow aSSb \\ &\Rightarrow aaSSbSb \\ &\Rightarrow aaSbSb \\ &\Rightarrow aabSb \\ &\Rightarrow aabb \end{aligned}$$

Linksableitung.

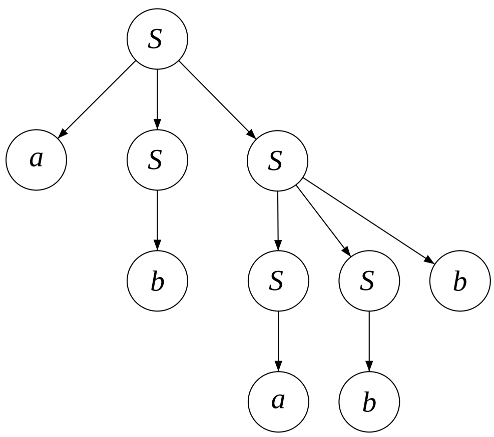
$$\begin{aligned} S &\Rightarrow aSSb \\ &\Rightarrow aSb \\ &\Rightarrow aaSSbb \\ &\Rightarrow aaSbb \\ &\Rightarrow aabb \end{aligned}$$

Rechtsableitung.

3.2 Ableitungsbäume

Ein **Ableitungsbaum** ist ein gewurzelter, orientierter Baum, dessen innere Knoten mit Nonterminalen und dessen Blätter mit Terminalsymbolen beschriftet sind. Die Kinder eines Knotens A müssen die rechte Seite einer Produktion $A \to \alpha$ sein. Die Wurzel ist mit dem Startsymbol beschriftet.

Beispiel:
$S \to aSS \mid SSb \mid a \mid b$

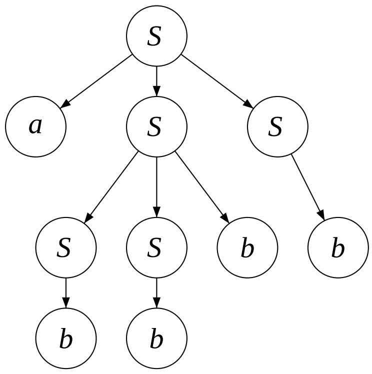


Zu jedem Ableitungsbaum gibt es eindeutig eine Linksableitung:

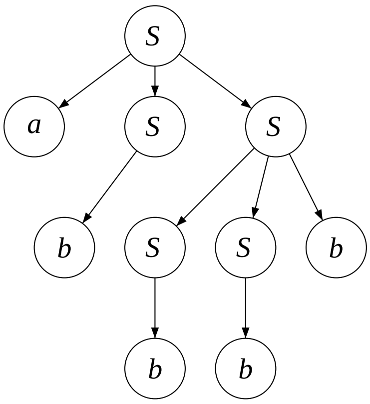
$$\begin{aligned} S &\Rightarrow aSS \\ &\Rightarrow abS \\ &\Rightarrow abSSb \\ &\Rightarrow abaSb \\ &\Rightarrow ababb \end{aligned}$$

Wir finden sie durch eine Tiefensuche im Ableitungsbaum.

Definition
Eine CFG (Kontextfreie Grammatik) $G = (N, T, P, S)$ ist **eindeutig**, wenn jedes $w \in L(G)$ genau einen Ableitungsbaum besitzt.



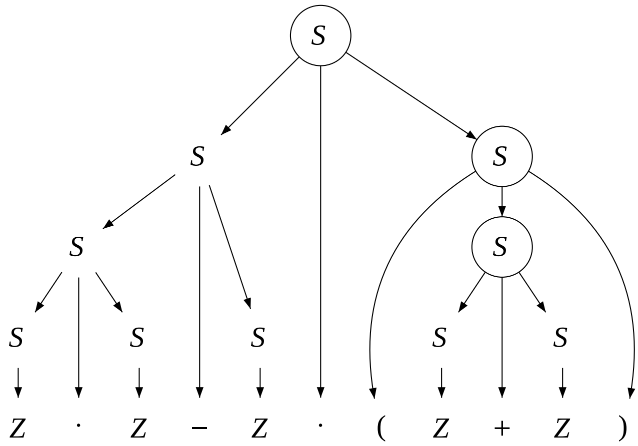
$S \to aSS \mid SSb \mid a \mid b$



$$\begin{aligned} S \Rightarrow aSS &\Rightarrow aSSbS \Rightarrow abSbS \Rightarrow abbbS \Rightarrow abbbb \\ &\Rightarrow abS \Rightarrow abSSb \Rightarrow abbSb \Rightarrow abbbb \end{aligned}$$

$S \to S + P \mid S - P \mid P$
$P \to P \cdot A \mid P/A \mid A$
$A \to (S) \mid Z$
$Z \to 0 \mid 1 \mid Z0 \mid Z1$

$S \rightarrow (S) \mid S + S \mid S - S \mid S \cdot S \mid S/S \mid Z$



$Z \rightarrow 0 \mid 1 \mid Z0 \mid Z1$

Definition
Eine Sprache L ist **kontextfrei**, wenn es eine Kontextfreie Grammatik G gibt mit $L = L(G)$
L ist eindeutig kontextfrei, wenn $L = L(G)$ für eine eindeutige CFG G

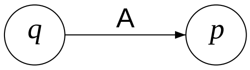
Folgende Sprache ist kontextfrei, aber nicht eindeutig kontextfrei: $L = \left\{a^i b^j c^k d^l \mid i = j \text{ oder } k = l\right\}$
$S \rightarrow AX \mid YB,\ Y \rightarrow aY \mid Yb \mid \varepsilon,\ B \rightarrow cBd \mid \varepsilon$
$A \rightarrow aAb,\ X \rightarrow cX \mid Xd \mid \varepsilon$

3.3 Die pre*-Operation

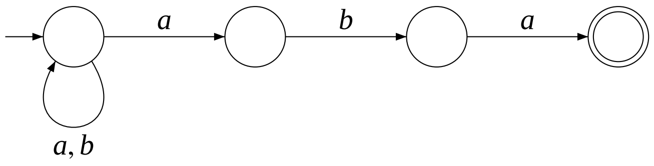
Sei $G = (N, T, P, S)$ eine CFG und $L \subseteq (N \cup T)^*$ eine beliebige Sprache.
Wir definieren: $\mathrm{pre}^*_G(L) = \{\alpha \in (N \cup T)^* \mid \alpha \overset{*}{\Rightarrow}_G \beta \text{ für ein } \beta \in L\}$
$S \in pre^*(L) \Leftrightarrow L \cap L(G) \neq \emptyset$
10.6.05 $S \in pre^*(\{w\})$ mit $w \in T^* \Leftrightarrow w \in L(G)$

Satz 3.1: Falls L regulär ist, dann ist auch $pre^*(L)$ regulär.

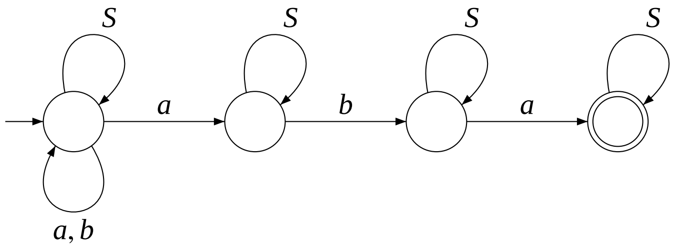
Beweis:
Sei $L = L(M)$ für einen NFA M. Wir transformieren M in einen NFA M' mit $L(M') = pre^*(L)$ durch folgende Regeln:

Falls $A \rightarrow \alpha \in P$ und $p \in \hat{\delta}(q, \alpha)$, dann füge den Übergang  hinzu, d.h. $\delta(q, A) := \delta(q, A) \cup \{p\}$. Wiederhole dies, solange wie möglich.

Beispiel:
$G : S \rightarrow aSa \mid bSb \mid aSb \mid \varepsilon$

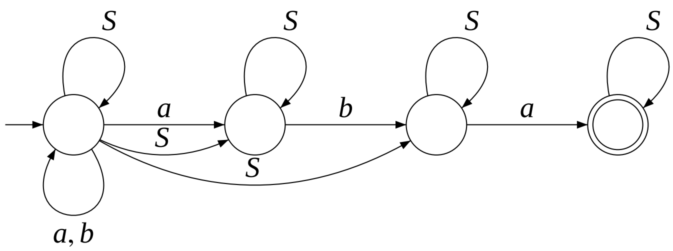


M_0 :

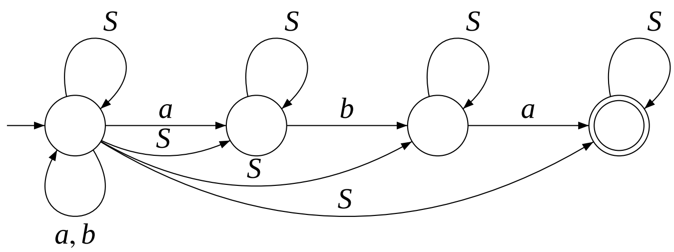


M_1 :

$M_0 := M$
M_{k+1} := „Sättigung“von M_K



M_2 :



M_3:

$M_4 = M_3$ $L(M') = pre^*(L) = pre^*(L(M))$
$L = (a+b)^*aba$

14.06.05

$pre^*(L) = \{\alpha | \alpha \overset{*}{\Rightarrow} \beta$ für ein $\beta \in L\}$
$pre(L) = \{\alpha | \alpha \Rightarrow \beta$ für ein $\beta \in L\}$
$pre^0(L) = L$, $pre^{k+1}(L) = pre(pre^k(L))$
Sei M ein NFA:
$M_0 = M$, M_{k+1} = ”Sättigung” von M_k
$q_i \in \hat{\delta}_{M_{k+1}}(q_j, A)$ falls $q_i \in \hat{\delta}_{M_k}(q_j, \alpha), A \to \alpha \in P$
$q_i \longrightarrow^{\alpha}_{M_k} q_j$ g.d.w. $q_j \in \hat{\delta}_{M_k}(q_i, \alpha)$

$L(M_{k+1}) = L(M_k) \cup pre(L(M_k))$
durch Induktion folgt: $L(M_k) = L(M) \cup pre(L(M)) \cup pre^2(L(M)) \cup \cdots \cup pre^k(L(M))$

Beweis:

- $L(M_{k+1}) \supseteq L(M_k) \cup pre(L(M_k))$
  Sei $w \in L(M_k) \cup pre(L(M_k))$

1. Fall $w \in L(M_k) \Rightarrow w \in L(M_{k+1})$ da $L(M_{k+1}) \supseteq L(M_k)$.
2. Fall $w \in pre(L(M_k))$ und $w \notin L(M_k)$ d.h. $\delta = \alpha A \beta$ und $A \to \gamma \in P$ mit $\alpha\gamma\beta \in L(M_k)$
   denn $\delta \Rightarrow \alpha\gamma\beta \in L(M_k)$
   aber $q_0 \longrightarrow^{\alpha}_{M_k} q_i \longrightarrow^{\gamma}_{M_k} q_j \longrightarrow^{\beta}_{M_k} q_f$ mit $q_f \in F$ daher $q_0 \longrightarrow^{\alpha}_{M_{k+1}} q_i \longrightarrow^{A}_{M_{k+1}} q_j \longrightarrow^{\beta}_{M_{k+1}} q_f$
   und daher $\delta = \alpha A \beta \in L(M_{k+1})$.

- $L(M_{k+1}) \subseteq L(M_k) \cup pre^*(L(M_k))$
  Sei $\delta \in L(M_{k+1})$

1. Fall $\delta \in L(M_k)$
   Glück gehabt
2. Fall $\delta \notin L(M_k)$
   Wir wissen: $\delta \in L(M_{k+1})$, aber $\delta \notin L(M_k) \Rightarrow q_0 \xrightarrow[M_{k+1}]{\delta} q_f$ für ein $q_f \in F$.
   Genauer:$q_0 \xrightarrow[M_k]{\alpha_0} q_1 \xrightarrow[M_{k+1}]{A_1} q_1' \xrightarrow[M_k]{\alpha_1} q_2 \xrightarrow[M_{k+1}]{A_2} q_2' \longrightarrow \ldots \longrightarrow q_f$
   α_0 = Längster Präfix von δ, für den es diese Zerlegung gibt.
   Insbesondere: $q_i \xrightarrow[M_{k+1}]{A_i} q_i'$ aber $q_i \cancel{\xrightarrow[M_k]{A_i}} q_i'$
   $\Rightarrow q_i \xrightarrow[M_k]{\beta_i} q_i'$ mit $A_i \to \beta_i \in P$
   $\Rightarrow q_0 \xrightarrow[M_k]{\alpha_0} q_1 \xrightarrow[M_k]{\beta_1} q_1' \xrightarrow[M_k]{\alpha_1} q_2 \xrightarrow[M_k]{\beta_2} q_2' \longrightarrow \ldots \longrightarrow q_f$ d.h. $\alpha_0\beta_1\alpha_1\beta_2\alpha_2 \cdots \in L(M_k)$ $\delta \stackrel{*}{\Rightarrow} \alpha_0\beta_1\alpha_1\beta_2\alpha_2 \cdots \in L(M_k)$
   denn:
   $\delta = \alpha_0 A_1 \alpha_1 A_2 \alpha_2 \ldots$
   und $A_1 \to \beta_1 \in P$ $\quad \Rightarrow \delta \in pre^*(L(M_k))$
   $A_2 \to \beta_2 \in P$
   $\vdots$

1. $L(M_{k+1}) \subseteq L(M_k) \cup pre^*(L(M_k))$
2. $L(M_{k+1}) \supseteq L(M_k) \cup pre(L(M_k))$

$L(M') = pre^*(L(M))$

2.	1.
$L(M_1) \supseteq L(M) \cup pre(L(M))$	$L(M_1) \subseteq pre^*(L(M_0)) = pre^*(L(M))$
$L(M_2) \supseteq L(M) \cup pre(L(M)) \cup pre^2(L(M))$	$L(M_2) \subseteq pre^*(L(M_1)) = pre^*(L(M))$
$\vdots$	$\vdots$
$L(M') \supseteq pre^*(L(M))$	$L(M') \subseteq pre^*(L(M))$

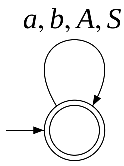
3.4 Entscheidungsprobleme für CFG

Folgende Probleme lassen sich in polynomieller Zeit mit pre^* lösen:

1. Wortproblem
   Eingabe: $G = (N, T, P, S), w \in T^*$
   Frage: Ist $w \in L(G)$?
   Lösung Ist $S \in pre^*(\{w\})$?

2. Unproduktive Symbole

   Eingabe: $G = (N, T, P, S)$
   Ausgabe: Menge der unproduktiven Symbole $U \subseteq N$
   $A \in U$ g.d.w. es kein $w \in T^*$ mit $A \stackrel{*}{\Rightarrow} w$.
   $S \to aA \mid bB$ 17.06.05
   $B \to SB \mid aB$
   $A \to SB \mid a$
   Lösung: A unproduktiv g.d.w. $A \notin pre^*(T^*)$



3. Leerheitsproblem
   Eingabe: CFG
   Frage: Ist $L(G) = \emptyset$
   Lösung Ist $S \in pre^*(\{T^*\})$?

4. Unerreichbare Symbole
   Eingabe: CFG G
   Ausgabe: Menge der unerreichbaren Symbole

   A ist unerreichbar, falls es keine Satzform $\alpha A \beta$ gibt mit $S \stackrel{*}{\Rightarrow} \alpha A \beta$
   A unerreichbar $\Leftrightarrow$ $S \notin pre^*((N \cup T)^* A (N \cup T)^*)$
   A **unnütz** gdw. A unproduktiv oder A unerreichbar.
   Unnütze Symbole können aus G entfernt werden.

5. Unendlichkeit.
   Eingabe: CFG G
   Frage: Ist $L(G)$ endlich ?
   1. Entferne unnütze Symbole
   jetzt:$|L(G)| \to \infty$ gdw. $A \stackrel{*}{\Rightarrow} \alpha A \beta$ für $A \in N$ und $|\alpha\beta| > 0$ Lösung: Test ob $A \in pre^*((N \cup T)^+ A (N \cup T)^* \cup (N \cup T)^* A (N \cup T)^+)$ für ein $A \in N$

6. Nullierbare Symbole
   A ist **nullierbar**, falls $A \stackrel{*}{\Rightarrow} \varepsilon$. Es kann in G nullierbare Symbole geben, obwohl $\varepsilon \notin L(G)$. z.B. $S \to Aa, A \to \varepsilon, A \to AA$
   Die nullierbaren Symbole sind $N \cap pre^*(\{\varepsilon\})$

21.06.05

3.5 Chomsky-Normalform

$A \stackrel{*}{\Rightarrow} \varepsilon$ A nullierbar
$A \to \varepsilon$ ε-Produktion
ε-Produktion eliminieren:

$A \to \alpha_1 \beta_1 \alpha_2 \beta_2 \alpha_3 \ldots \alpha_k \beta_k \alpha_{k+1}$
β_i seinen die nullierbaren Symbole der Rechten Seite.
Füge diese Regeln hinzu:

$$\begin{aligned} A &\rightarrow \alpha_1\alpha_2 \ldots \alpha_k \\ A &\rightarrow \alpha_1\beta_1\alpha_2 \ldots \alpha_k \\ &\vdots \end{aligned}$$

Eine Regel für jede Teilmenge von $\{1 \ldots k\}$

Beispiel:

$$\begin{aligned} S &\rightarrow SA|BS|a \\ A &\rightarrow AB|a|c \\ B &\rightarrow AS|b|\varepsilon \end{aligned}$$

Äquivalente Grammatik ohne ε-Produktionen

$$\begin{aligned} S &\rightarrow SA|BS|a \\ A &\rightarrow AB|a|B \\ B &\rightarrow AS|b|s \end{aligned}$$

Definition
Eine CFG G ist in Chomsky-Normalform (CNF), falls alle Regeln die Form $A \rightarrow BC$ oder $A \rightarrow a$ haben.

Satz 3.2: Zu jeder ε-freien CFL gibt es eine Grammatik in CNF.

Beweis:
Sei G eine CFG ohne ε-Produktionen und ohne unnütze Symbole.

1. Wähle neue Nonterminals R_a für jedes $a \in T$ und eine neue Regel $R_a \rightarrow a$. Ersetze in anderen Regeln a durch R_a.

2. Ersetze $A \rightarrow B_1B_2 \ldots B_k$ durch

$$\begin{aligned} A &\rightarrow B_1B_{2\ldots k} \\ B_{2\ldots k} &\rightarrow B_2B_{3\ldots k} \\ B_{3\ldots k} &\rightarrow B_3B_{4\ldots k} \\ &\vdots \\ B_{k-1\ldots k} &\rightarrow B_{k-1}B_k \end{aligned}$$

   wobei $B_{i\ldots k}$ neu ist.

3. Eliminiere Kettenregeln $A \rightarrow B$ Finde Paare A, B mit $A \stackrel{*}{\Rightarrow} B$
   Falls $C \rightarrow \alpha A\beta$ neue Regel $C \rightarrow \alpha\beta\beta$.
   Falls $\beta \rightarrow \gamma$ dann neue Regel $A \rightarrow \gamma$. Dann Kettenregeln entfernen.

$$
\begin{array}{lcl}
S & \rightarrow & aSa|bSb|a|b| \\
\hline
S & \rightarrow & R_aSR_a|R_bSR_b|a|b \\
R_a & \rightarrow & a \\
R_b & \rightarrow & b \\
\hline
S & \rightarrow & R_aX \\
X & \rightarrow & SR_a \\
S & \rightarrow & R_bY \\
Y & \rightarrow & SR_b \\
S & \rightarrow & a|b \\
R_a & \rightarrow & a \\
R_b & \rightarrow & b
\end{array}
$$

Beispiel:
$S \rightarrow S + T|T$
$T \rightarrow T \cdot A|A$

$$
\begin{array}{llcl}
 & S & \rightarrow & S + T \\
 & S & \rightarrow & T + T \\
 & S & \rightarrow & A + T \\
 & S & \rightarrow & S + A \\
A \rightarrow a & S & \rightarrow & T + A \\
 & S & \rightarrow & A + A \\
 & T & \rightarrow & T \cdot A \\
 & T & \rightarrow & A \cdot A \\
 & A & \rightarrow & a
\end{array}
$$

3.6 Greibach-Normalform

Eine Regel $A \rightarrow A\alpha$ ist linksrekursiv.
Linksrekursion lässt sich eliminieren.
Für $A \in N$ seien $A \rightarrow A\alpha_1|\dots|A\alpha_k$ die linksrekursiven und $A \rightarrow \beta_1|\dots|\beta_l$ die anderen Regeln. Ersetze $A \rightarrow A\alpha_1|\dots|A\alpha_k$ durch
$A \rightarrow \beta_1 Z|\dots|\beta_l Z|\beta_1|\dots|\beta_l$
$Z \rightarrow \alpha_1 Z|\dots|\alpha_k Z|\alpha_1|\dots|\alpha_k$

Beispiel:

$$
\begin{array}{lcl}
S & \rightarrow & SS|AS|SB \\
\hline
S & \rightarrow & ASZ|AS \\
Z & \rightarrow & SZ|BZ|S|B \\
\hline
S & \rightarrow & S + T|T \\
\hline
S & \rightarrow & T|TZ \\
Z & \rightarrow & +TZ|+T
\end{array}
$$

Definition
Eine CFG ist in GNF falls jede Regel von der Form $A \rightarrow aBC\dots$ ist, d.h. die rechte Seite ist aus TN^*

Satz 3.3: Es sei G eine CFG mit $\varepsilon \notin L(G)$. Dann gibt es eine CFG G' in GNF mit $L(G) = L(G')$.

Linksrekursion entfernen:

$$\begin{array}{lcl} A & \rightarrow & A\alpha|\ldots|A\alpha_k \\ A & \rightarrow & \beta_1|\ldots|\beta_l \end{array}$$

$$\begin{array}{lcl} A & \rightarrow & \beta_1|\ldots\beta_l \\ A & \rightarrow & \beta_1 Z|\ldots|\beta_l Z \\ Z & \rightarrow & \alpha_1 Z|\ldots|\alpha_k Z \\ Z & \rightarrow & \alpha_1|\ldots|\alpha_k \end{array}$$

Beweis:
Wir beginnen mit einer CFG in CNF ohne unnütze Symbole, ohne ε-Regeln. O.b.d.A. $N = \{A_1, \ldots, A_n\}$

1. Wir verändern die Grammatik so, dass es keine Regeln $A_i \rightarrow A_j\alpha$ mit $j \leq i$ gibt.
   Nehmen wir an, dies gilt schon für $i = 1, \ldots, k$ Sei $A_{k+1} \rightarrow A_{l\alpha}$ eine Regel mit $l \leq k+1$ und l die kleinste Zahl, für die es eine solche Regel gibt.
   Ersetze $A_{k+1} \rightarrow A_l$ durch $A_k + 1 \rightarrow \beta_1\alpha|\ldots|\beta_m\alpha$ wobei $A_l \rightarrow \beta_1|\ldots|\beta_m$ falls $l \leq k$
   Falls $l = k+1$, dann $A_{k+1} \rightarrow A_{k+1}\alpha \;\Rightarrow$ Linksrekursion eliminieren.
   Wiederhole...

2. Alle Regeln haben jetzt die Form $A_i \rightarrow A_j\alpha$ mit $j > i$ oder $A_i \rightarrow a\alpha$
   Wir ersetzen $A_i \rightarrow A_j\alpha$ durch $A_i \rightarrow \beta_1\alpha|\ldots|\beta_k\alpha$ falls $A_j \rightarrow \beta_1|\ldots|\beta_k \in P$ bis alle rechten Seiten mit Terminal beginnen.

Beispiel:
$S < A < B < Z_1 < Z_2 < \ldots$

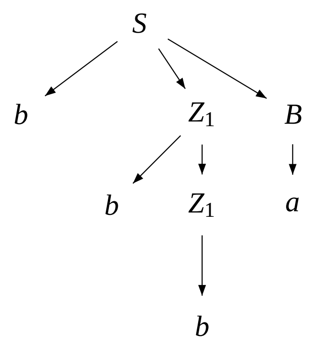
$$\begin{array}{lcl} S & \rightarrow & AB|a \\ A & \rightarrow & AA|b \\ B & \rightarrow & AB|a \end{array}$$

$$\begin{array}{lcl} S & \rightarrow & AB|a \\ A & \rightarrow & b|bZ_1 \\ B & \rightarrow & AB|a \\ Z_1 & \rightarrow & A|AZ_1 \end{array}$$

$$\begin{array}{lcl} S & \rightarrow & AB|a \\ A & \rightarrow & b|bZ_1 \\ B & \rightarrow & bB|bZ_1B|a \\ Z_1 & \rightarrow & A|AZ_1 \end{array}$$

$$\begin{array}{rcl} S & \rightarrow & AB|a \\ A & \rightarrow & b|bZ_1 \\ B & \rightarrow & bB|bZ_1B|a \\ Z_1 & \rightarrow & b|bZ_1|bZ_1|bZ_1Z_1 \end{array}$$

$$\begin{array}{rcl} S & \rightarrow & bB|bZ_1B|a \\ A & \rightarrow & b|bZ_1 \\ B & \rightarrow & bB|bZ_1B|a \\ Z_1 & \rightarrow & b|bZ_1|bZ_1Z_1 \end{array}$$



$$\begin{array}{rcl} S & \Rightarrow & bZ_1B \\ & \Rightarrow & bbZ_1B \\ & \Rightarrow & bbbB \\ & \Rightarrow & bbba \end{array}$$

Beispiel:

$$\begin{array}{rcl} S & \rightarrow & SAS|SBS|A \\ A & \rightarrow & ABA|a \\ B & \rightarrow & BBB|b \end{array}$$

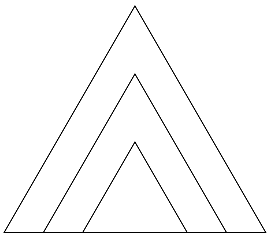
$$\begin{array}{rcl} S & \rightarrow & A|AZ_1 \\ Z_1 & \rightarrow & AS|BS|ASZ_1|BSZ_1 \\ A & \rightarrow & a|aZ_2 \\ Z_2 & \rightarrow & BA|BAZ_2 \\ B & \rightarrow & b|bZ_3 \\ Z_3 & \rightarrow & BB|BBZ_3 \end{array}$$

3.7 Pumping-Lemma für CFL's

Satz 3.4: Für jede CFL L gibt es eine Zahl N, für die gilt: Jedes Wort $z \in L$ mit $|z| > N$ hat eine Zerlegung $z = uvwxy$ mit

1. $|vwx| \leq N$
2. $|vx| > 0$
3. $uv^iwx^iy \in L$ für alle $i \geq 0$

Beweis:
Sei $\varepsilon \notin L$.
G sei eine CFG für L in CNF ohne unnütze Symbole.



Beispiel:
$L = \{a^nb^nc^n | n \geq 0\}$ **Annahme:** L CFL
Sei N die Konstante des Pumping-Lemma.
Wähle $z = a^Nb^Nc^N$.
Dann gibt es $z = uvwxy$ mit 1),2),3)

1) $\Rightarrow$ vx enthält kein a oder vx enthält kein c.
$\Rightarrow uv^2wx^2y \notin L$ Widerspruch zu 3)

1.7.05

Beispiel:
(Aus der Großübung) $A \cancel{< B} < S < B$ Umwandlung in GNF:

$$\begin{array}{rcl} S & \rightarrow & AB \\ A & \rightarrow & BB|a \\ B & \rightarrow & BA|b \end{array}$$

$$\begin{array}{rcl} S & \rightarrow & BBB|aB \\ A & \rightarrow & BB|a \\ B & \rightarrow & BA|b \end{array}$$

$$\begin{array}{rcl} S & \rightarrow & BBB|aB \\ A & \rightarrow & BB|a \\ B' & \rightarrow & AB'|A \\ B & \rightarrow & b|bB' \end{array}$$

$$\begin{aligned} S &\to aB|bBB|bB'BB \\ A &\to a|bB|bB'B \\ B &\to b|bB' \\ B' &\to aB'|bBB'|bB'BB|a|bB|bB'B \end{aligned}$$

3.8 Kellerautomaten

Beispiel:
Phase 1 a lesen $\to$ a auf den Keller
b lesen $\to$ b auf den Keller

Phase 2 a lesen $\to$ nimm Zeichen vom Keller falls es kein a ist, hör auf.
b lesen $\to$ nimm Zeichen vom Keller falls es kein b ist, hör auf.

Phase 3 Kein Zeichen mehr + Keller leer $\to$ Glücklich

Definition
Ein **Kellerautomat** (PDA) ist ein 7-Tupel: $M = (Q, \Sigma, \Gamma, \delta, q_0, \Gamma_0, F)$ mit:

- Q Menge der Zustände (endl.)
- Σ Eingabealphabet
- Γ Kelleralphabet
- $\delta : Q \times (\Sigma \cup \varepsilon) \times \Gamma \to 2^{Q\times\Gamma^*}$
- q_0 Startzustand
- Γ_0 Kellerbodensymbol
- $F \subseteq Q$ Menge der Endzustände.

$\delta(q, a, X) = \{(p, B), \ldots\}$

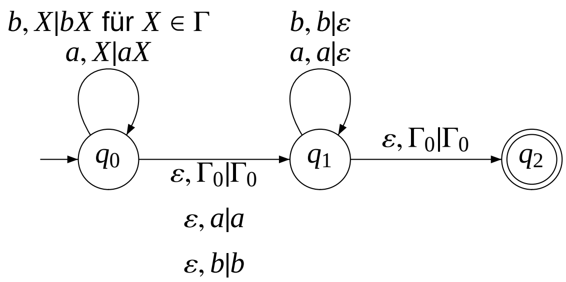
(q, w, γ)
$(q, aw, X\beta) \vdash (p, w, \alpha\beta)$ falls $(p, \alpha) \in \delta(q, a, X)$ wobei $a \in \Sigma \cup \{\varepsilon\}$

$L(M) = \{w \in \Sigma^* \mid (q_0, w, \Gamma_0) \overset{*}{\vdash} (q, \varepsilon, \gamma)$ mit $\gamma \in \Gamma^*, q \in F\}$

5.7.05

Definition
Eine **Konfiguration** eines PDA ist ein Tripel (q, w, γ) wobei
$q \in Q$ ein Zustand
$w \in \Sigma^*$ das noch zu lesende Wort
$\gamma \in \Gamma^*$ der Kellerinhalt ist. Wir schreiben $(q, aw, X\beta) \underset{M}{\vdash} (p, w, \alpha\beta)$, falls $(p, \alpha) \in \delta(q, a, X)$ wobei $q, p \in Q$, $a \in \Sigma \cup \{\varepsilon\}, w \in \Sigma^*, X \in \Gamma, \alpha, \beta \in \Gamma^*$
Die **Startkonfiguration** ist (q_0, w, Γ_0) für $w \in \Sigma^*$. $\overset{*}{\vdash}$ ist die reflexiv-transitive Hülle von $\vdash$.
Die von M durch Endzustand akzeptierte Sprache $L(M) = \{w \in \Sigma^* | (q_0, w, \Gamma_0) \overset{*}{\vdash} (p, \varepsilon, \gamma)\}$

Beispiel:

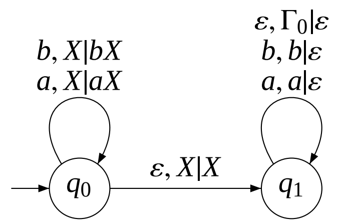


$$\begin{array}{ccccc}
(q_0, abba, \Gamma_0) & \vdash & (q_0, bba, a\Gamma_0) & \vdash & (q_1, bba, a\Gamma_0) \\
 & & \downarrow & & \\
 & & (q_0, ba, ba\Gamma_0) & \vdash & (q_1, ba, ba\Gamma_0) \\
 & & \downarrow & & \downarrow \\
 & & (q_0, a, bba\Gamma_0) & & (q_1, a, a\Gamma_0) \\
 & & & & \downarrow \\
 & & & & (q_1, \varepsilon, \Gamma_0) \\
 & & & & \downarrow \\
 & & & & (q_2, \varepsilon, \Gamma_0)
\end{array}$$

$L(M) = \{ww^R | w \in \{a, b\}^*\}$

Definition
Sei $M = (Q, \Sigma, \Gamma, \delta, q_0, \Gamma_0, F)$ ein PDA. Die von M durch leeren Keller akzeptierte Sprache $N(M)$ ist $N(M) = \{w \in \Sigma^* | (q_0, w, \Gamma_0) \overset{*}{\vdash} (p, \varepsilon, \varepsilon)$ mit $p \in Q$

Beispiel:



$X \in \Gamma = \{a, b, \Gamma_0\}$
$N(M) = \{ww^R | w \in \{a, b\}^*\}$ $(L(M) = \emptyset)$

Satz 3.5: Für ein PDA $A = (Q, \Sigma, \Gamma, \delta, q_0, \Gamma_0)$ gibt es einen PDA B mit $L(B) = N(A)$

> **Beweis:**
> Verwende: $B = (Q \cup \{q_0^*, q_f^*\}, \Sigma, \Gamma \cup \{\Gamma_0^*\}, \delta_B, q_0^*, \Gamma_0^*, \{q_f^*\})$
> A: $(q_0, w, \Gamma_0) \overset{*}{\underset{A}{\vdash}} (p, \varepsilon, \varepsilon)$
> B: $(q_0^*, w, \Gamma_0^* \underset{B}{\vdash} (q_0, w, \Gamma_0, \Gamma_0^*) \overset{*}{\underset{B}{\vdash}} (p, \varepsilon, \Gamma_0^*) \vdash (q_f^*, \varepsilon, \Gamma_0^*)$

Satz 3.6: Für einen PDA $A = (Q, \Sigma, \Gamma, \delta, q_0, \Gamma_0, F)$ gibt es einen PDA B mit $N(B) = L(A)$

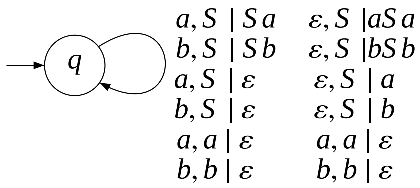
Beweis:
$(q_0, w, \Gamma_0) \overset{*}{\underset{A}{\vdash}} (p, \varepsilon, \varepsilon)$ mit $p \notin F$
$(q_0, w, \Gamma_0) \overset{*}{\underset{B}{\vdash}} (p, \varepsilon, \varepsilon)$

8.7.05

Satz 3.7: Es sei $G = (N, T, P, S)$ eine CFG ohne ε- Produktionen. Dann gibt es einen PDA M mit $N(M) = L(G)$

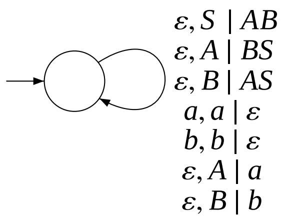
Beweis:
$M = (\{q\}, T, N \cup T, \delta, q, S)$ wobei
$\delta(q, a, a) = \{(q, \varepsilon)\}$ für alle $a \in T$
$\delta(q, \varepsilon, A) = \{(q, \alpha)|A \to \alpha \in P\}$ für alle $A \in N$ Wir wollen $(q, w, \gamma) \overset{*}{\vdash} q, \varepsilon, \varepsilon)$ gdw. $\gamma \overset{*}{\Rightarrow} w$. Insbesondere $w \in N(M) \Leftrightarrow (q, w, S) \overset{*}{\vdash} (q, \varepsilon, \varepsilon) \Leftrightarrow S \overset{*}{\Rightarrow} w$

Beispiel:
$S \to aSa \mid bSb \mid a \mid b$



Links: $(q, aba, S) \vdash (q, ba, Sa) \vdash (q, a, a) \vdash (q, \varepsilon, \varepsilon)$
Rechts: $(q, aba, S) \vdash (q, aba, aSa) \vdash (q, ba, Sa) \vdash (q, ba, ba) \vdash (q, a, a) \vdash (q, \varepsilon, \varepsilon)$

Beispiel:
$S \to AB$
$A \to a|BS$
$B \to b|AS$



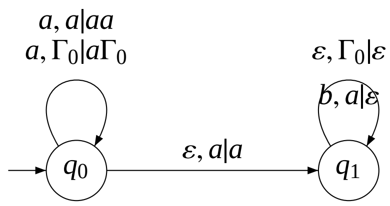
Satz 3.8: Es sei M ein PDA. Dann gibt es eine CFG G mit $L(G) = N(M)$

Beweis:
Es sei $M = (Q, \Sigma, \Gamma, \delta, q_0, \Gamma_0)$
Es sei $G_p = (N, \Sigma, P, [q_0, \Gamma_0, p])$ wobei $p \in Q$ und $N = \{[q, Z, r]|q, r \in Q, Z \in \Gamma\}$
Idee: $[q, Z, r] \overset{*}{\Rightarrow} w$ gdw. $(q, w, Z) \overset{*}{\vdash} (r, \varepsilon, \varepsilon)$ P enthält die Regeln: Falls $(r, \gamma) \in \delta(q, a, Z)$ $(a \in \Sigma \cup \{\varepsilon\})$sind $[q, Z, q_k] \to a[r, \gamma_1, q_1][q_1, \gamma_2, q_2][q_2, \gamma_3, q_3] \ldots [q_{k-1}, \gamma_k, q_k]$ in P, wobei $\gamma = \gamma_1 \ldots \gamma_k$

12.07.05

Info: Einmalige Fragestunde: Do. 14.7.05 ab 23:59 am Dreiländereck.

Beispiel:



$L(M) = \{a^n b^n | n \geq 1\}$

$$\begin{aligned}
[q_0, \Gamma_0, q_1] &\rightarrow a[q_0, a, q_1][q_1, \Gamma_0, q_1] \\
[q_0, a, q_1] &\rightarrow [q_1, a, q_1] \\
[q_1, a, q_1] &\rightarrow b \\
[q_1, \Gamma_0, q_1] &\rightarrow \varepsilon \\
[q_0, a, q_1] &\rightarrow a[q_0, a, q_1][q_1, a, q_1]
\end{aligned}$$

$$\begin{aligned}
[q_0, \Gamma_0, q_1] &\Rightarrow a[q_0, a, q_1][q_1, \Gamma_0, q_1] \\
&\stackrel{*}{\Rightarrow} ab[q_1, \Gamma_0, q_1] \\
&\Rightarrow ab
\end{aligned}$$

$$\begin{aligned}
[q_0, \Gamma_0, q_1] &\Rightarrow a[q_0, a, q_1][q_1, \Gamma_0, q_1] \\
&\Rightarrow aa[q_0, a, q_1][q_1, a, q_1][q_1, \Gamma_0, q_2] \\
&\stackrel{*}{\Rightarrow} aab[q_1, a, q_2][q_1, \Gamma_0, q_1] \\
&\Rightarrow aabb[q_1, \Gamma_0, q_1] \\
&\Rightarrow aabb
\end{aligned}$$

Idee: $\{w \mid [q, X, p] \stackrel{*}{\Rightarrow} w\} = \{w \mid (q, w, X) \stackrel{*}{\vdash} (p, \varepsilon, \varepsilon)\}$

3.8.1 Deterministische Kellerautomaten

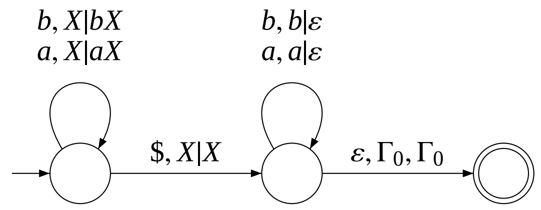
Definition
Ein PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, \Gamma_0, F)$ ist ein DPDA falls

1. $|\delta(q, a, X)| \leq 1$ für alle $q \in Q, a \in \Sigma \cup \{\varepsilon\}, X \in \Gamma$

2. Falls $\delta(q, a, X) \neq \emptyset$ für ein $a \in \Sigma$ dann muss $\delta(q, \varepsilon, X) = \emptyset$ sein.

Eine Sprache L ist eine DCFL falls $L = L(M)$ für einen DPDA M

Beispiel:



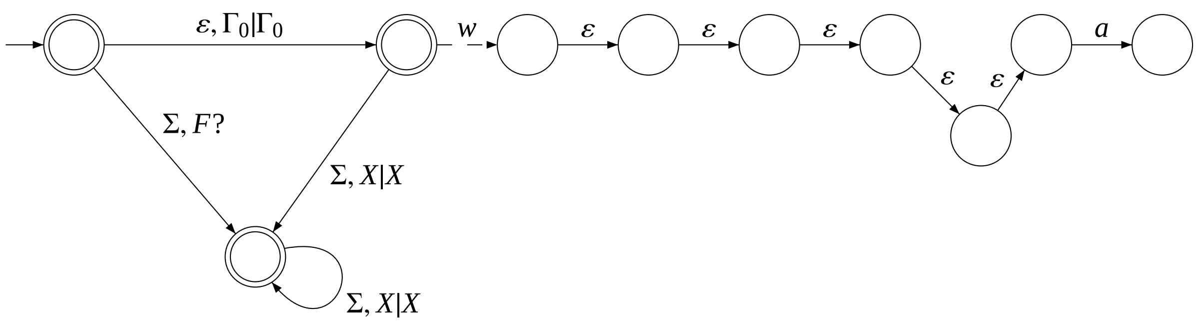
$L(M) = \{w\$w^R \mid w \in \{a, b\}^*\}$ für alle $X \in \Gamma = \{a, b, \Gamma_0\}$

Satz 3.9: DCFL ist unter Komplement abgeschlossen.
d.h. $L \in DCFL \Rightarrow \Sigma^* \setminus L \in DCFL$ 15.07.05

Beweis:
$M = (Q, \Sigma, \Gamma, \delta, q_0, \Gamma_0, F) DPDA$
Konstruiere M' mit $L(M') = \Sigma^* - L(M)$
$L(M) = \{w \in \Sigma^* \mid (q_0, w, \Gamma_0) \overset{*}{\vdash} (p, \varepsilon, \gamma)$ für $p \in F, \gamma \in \Gamma^*\}$

1. Führe ein neues Kellerbodensymbol und einen Fangzustand ein.
2. Falls $(q, \varepsilon, Z) \vdash (q', \varepsilon, \gamma') \vdash (q'', \varepsilon, \gamma'') \vdash \ldots$ unendlich lang ist, dann setze $\delta(q, \varepsilon, Z) := (q_\emptyset, Z)$ wobei $q_\emptyset$ der Fangzustand ist, falls $q, q', q'' \cdots \notin F$. Setze $\delta(q, \varepsilon, Z) := (q_f, Z)$ wobei q_f ein neuer Endzustand ist und $\delta(q_f, a, X) := (q_\emptyset, X)$ für alle $a \in \Sigma, X \in \Gamma$, falls eines der $q, q', \cdots \in F$ Ersetze Q durch

Alternative:



Problem: Endzustände/Nicht-Endzustände tauschen $\rightarrow$ akzeptiert noch immer das leere Wort!

Ersetze Q durch $Q' = Q \times \{1, 2, 3\}$
$F' = Q \times \{3\}$

Falls $\delta(q, \varepsilon, Z) = \{p, \gamma)\}$ setze $\delta'((q,1), \varepsilon, Z) := \begin{cases} \{((p,1),\gamma)\} & \text{falls } p \notin F \\ \{((p,2),\gamma)\} & \text{sonst} \end{cases}$

$\delta'((q,2), \varepsilon, Z) := \{((p,2),\gamma)\}$

$(q,1)$ kein Endzustand wurde gesehen.

$(q,2)$ Endzustand wurde gesehen.

Falls $\delta(q, a, Z) = \{(p, \gamma)\}$ für $a \in \Sigma$, setze $\delta'((q,2), \varepsilon, Z) := \{(q,3), Z\}$ und

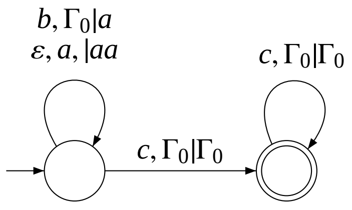
$\delta'((q,1), a, Z) = \delta'((q,3), a, Z) := \begin{cases} \{(p,1), \delta\} & \text{falls } p \notin F \\ \{(p,2), \gamma\} & \text{falls } p \in F \end{cases}$

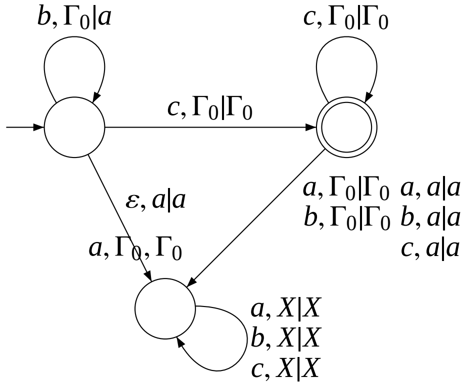
Neuer Startzustand: $(q_0, 1)$ falls $q_0 \in F$
$(q_0, 3)$ falls $q_0 \notin F$

"1": kein Endzustand in M gesehen seit dem ein Zeichen gelesen wurde.
"2": doch

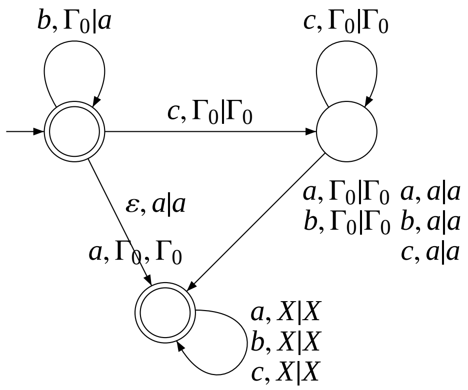
19.7.05

Beispiel:
$\Sigma = \{a, b, c\}$
$\Gamma = \{\Gamma_0, a\}$
$L(M) = c^+$
M:





$\bar{M}$



Satz 3.10: Seien L, L' CFLs und R regulär. Dann sind $L \cap R, L - R, L \cup L', LL'$ wieder CFL's

Beweis:

$\cap$: durch Produktautomat.

LL': $L(G_1) = L, L(G_2) = L'$
$G_1 = (N_1, T, P_1, S_1)$, $G_2 = (N_2, T, P_2, S_2)$
$G = (N_1 \cup N_2, T, P_1 \cup P_2 \cup \{S \to S_1S_2\}, S)$
$L(G) = L(G_1)L(G_2)$ o.B.d.A.: $N_1 \cap N_2 = \emptyset$

$L \cup L'$: analog, nur $\ldots S \to S_1|S_2 \ldots$

Satz 3.11: CFL ist **nicht** unter Komplement abgeschlossen.

Beweis:

$L = \{a^ib^jc^k \mid i \neq \ joder j \neq k\}$
Offensichtlich: $L \in CFL$ z.B.

$$\begin{aligned}
S &\to AaS_{ab}C \mid S_{ab}BbC \mid ABbS_{bc} \mid AS_{bc}Cc \\
S_{ab} &\to \varepsilon \mid aS_{ab}b \\
S_{bc} &\to \varepsilon|bS_{bc}c \\
A &\to \varepsilon \mid aA \\
B &\to \varepsilon \mid bB \\
C &\to \varepsilon \mid cC
\end{aligned}$$

Falls $\overline{L} \in$ CFL, dann auch $\overline{L} \cap a^*b^*c^*$. $\overline{L} \cap a^*b^*c^* = \{a^nb^nc^n \mid n \geq 0\} \notin CFL$
$\Rightarrow \overline{L} \notin CFL$

Korollar 2 $DCFL \subsetneq CFL$

Satz 3.12: Folgendes Problem ist nicht rekursiv (unentscheidbar):
Eingabe: zwei CFG G_1, G_2
Frage: Ist $L(G_1) \cap L(G_2) = \emptyset$?

Beweis:
Wir reduzieren folgendes Problem:
Eingabe: Einband-TM M, Wort w
Frage: $w \in L(M)$?

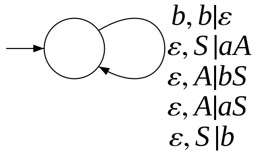
$L(G_1) = \{c_1\#c_1'^R\$c_2\#c_2'^R\$\dots\$c_n\#c_n'^R \mid c_i$ Konfiguration von M,
c_i' Nachfolgekonfiguration von $c_i\}$
$L(G_2) = \{c_1\#c_2^R\$c_2\#c_3^R\$\dots\$c_{n-1}\#c_n^R \mid c_i$ Konfiguration von M,
c_1 Startkonfiguration von M auf w,
c_{n-1} akz. Konfiguration$\}$
$L(G_1) \cap L(G_2) = \{c_1\#c_2^R\$c_2\#c_2^R\$\dots\$c_{n-1}\#c_n^R \mid c_1$ Startkonfiguration
c_{n-1} Endkonf. akz.
c_{i+1} Nachf. von $c_i\} = \emptyset \Rightarrow w \notin L(M)$

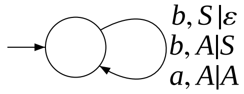
22.07.05

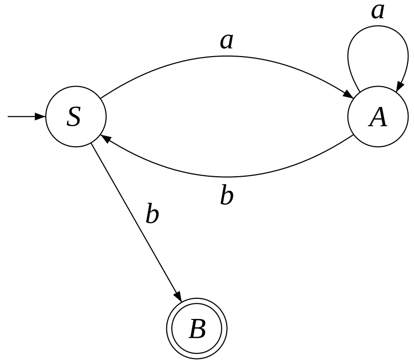
3.9 Chomsky-Hierarchie

Chomsky 0	Phrase-structure	Semi-Thue
Chomsky 1	Kontextsensitiv	$CAD \to CBCD$
Chomsky 2	$A \to bAC$	CFL
Chomsky 3	$A \to aB, B \to C$	Reguläre

Optional:
$S \to aA$
$A \to bS \mid aA$
$S \to aA|bB$
$A \to aA|bS$
$B \to \varepsilon$







Chomsky-1
$NSPACE(n)$ LBA

$NSPACE(n) \stackrel{?}{=} DSPACE(n)$
$NSPACE(n) = Co - NSPACE(n)$ (wobei Co: Komplement)

Ende

Stichwortverzeichnis

ε-Automat, 15
ε-Übergang, 15
pre*, 32

Ableitungen, 29
Ableitungsbäume, 30
Abschluss einer regulären Sprache, 8
akzeptierte Sprache eines PDA, 41
Alphabet, 5
Automat, deterministischer, 10
Automat, nichtdeterministisch, 12

Chomsky-Normalform, 35
CNF, 35

DCFL, 45
Deterministischer endlicher Automat, 10
Deterministischer Kellerautomat, 44

Erzeugendensystem, 5
Erzeugendensystem, freies, 5

GNF, 37
Greibach-Normalform, 37

Halbgruppe, 5

Isomorph, 21

Kelleralphabet, 41
Kellerautomat, 41
Kellerautomat, deterministisch, 44
Kellerbodensymbol, 41
Klasse der regulären Sprachen, 8
Kleenesche Hülle, 7
Konfiguration, eines PDA, 41
Kontextfreie Sprache, 32

Länge, eines Wortes, 7
Leerheitsproblem, 35
linksrekursiv, 37

Minimalisierung von DFAs, 22
Monoid, 5
Myhill-Nerode, Satz von, 20

NFA mit ε-Übergägen, 15
Normalform - Chomsky, 35
Normalform, Greibach, 37
Nullierbar, 35

Pontenzautomat, 13
Potenzmengenkostruktion, 13
Prodduktautomat, 17
Pumping-Lemma, 26
Pumping-Lemma für CFL’, 40

reguläre Sprache, 8
Regulärer Ausdruck, 7

Schnittleerheitsproblem, 9
Sprachäquivalenz, 21
Sprachequivalenz, 9

Unendlichkeitsproblem, 35
Unerreichbar, 35
Unnütz, 35
Unproduktive Symbole, 35

Wort, 5
Wortproblem, 9, 35