Eti Vorlesungsnotizen

Fabio Oesch

5. Semester (HS 2014)

Inhaltsverzeichnis

1 Repetition

Alphabet: endliche Menge von Zeichen (Buchstaben)
Bsp: $\Sigma := 0, 1$
Wort über Σ: endliche Folge von Buchstaben aus Σ
Σ^* = Menge oder Wörter über Σ
$\Sigma^* = \Sigma^* - \varepsilon$, ε: leeres Wort
Bsp: $\Sigma = 0, 1$, $\Sigma^* = \varepsilon \cup 0, 1 \cup 0.0, 0.1, 0.2$
Länge von $\omega \in \Sigma^*$, $\mid \omega \mid :=$ Anzahl Buchstaben im Wort: **Bsp:** $\omega = 01011$, $\mid \omega \mid = 5$
Def: Eine Teilmenge L von Σ^*, $L \subset \Sigma$, heisst eine **Sprache** über Σ.

1.1 Entscheidungsproblem (EP)

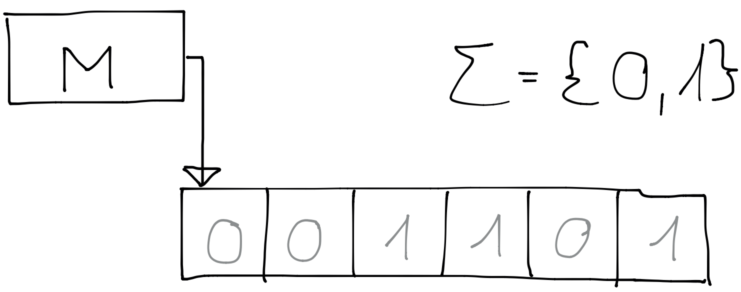
Gegeben:
Σ Alphabet
$L \subset \Sigma^*$ Sprache
$\omega \in \Sigma^*$ Wort
Bsp:

1. (Primalität)
   $\Sigma := 0, 1$
   $L := \{\text{binären Darstellungen der Primzahlen}\} = \{10, 111, 101, 111, \dots\}$
   $\omega = 11 \dots 1 \in L$?
2. **Progammiersprachen**
   if, else, do, ...
3. Natürliche Sprache

1.2 Beschreibung von Sprachen

- Aufzählung von Wörtern (endliche Sprache)
- Generieren von Wörtern (Grammatiken)
- Erkennen von Wörtern (Algorithmen, Automaten)
- Konstruiren von Wörtern (mathematische Operationen)

1.3 Endliche Automaten



Def: Ein endlicher Automat (deterministisch) ist ein 5-Tupel $A = (Q, \Sigma, \delta, q_0, F)$

Q Menge der Zustände

Σ Alphabet

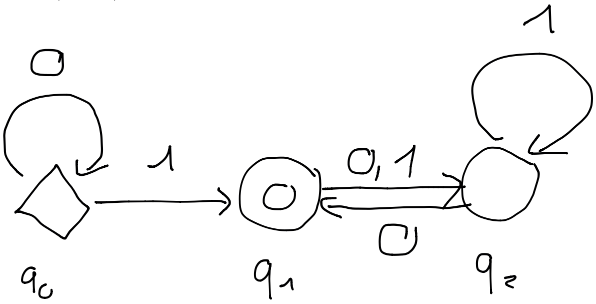
δ $Q \times \Sigma \to Q$ Übergangsfunktion

q_0 Startzustand

$F \subset Q$ akzeptierende Zustände

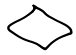
Bsp: Sei $A = (Q, \Sigma, \delta, q_0, F)$
mit $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, $F = \{q_1\}$

δ	0	1
q_0	q_0	q_1
q_1	q_2	q_2
q_2	q_1	q_2



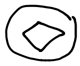
Antwortfunktion von A
$r_A : \Sigma^* \to Q$
Bsp: $r_A(0, 0, 1, 0) = q_2 \notin F \Rightarrow 0010 \notin L(A)$
$L(A) = \{\omega \in \Sigma^* \mid r_A(\omega) \in F\}$

1.3.1 Notation

 : Startzustand

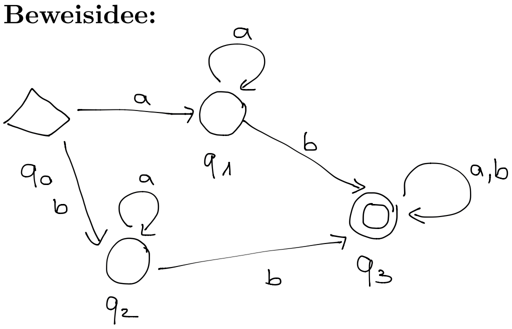
 : akzept Zustand

 : "gewöhnlicher" Zustand

 : akzept. Startzustand

1.4 Satz 1

Vor: $A \in DFA$
Beh: $L(A)$ ist regulär
Beweisidee:



$\Sigma = \{0, 1\}$
$G = (N, T, Q, S)$ regulär mit $L(G) = L(A)$
$T = \Sigma = \{a, b\}$
$N = \{S = Sq_0, Sq_1, Sq_2, Sq_3\}$
$R = \{Sq_0 \to aSq_1 \mid bSq_2, Sq1 \to aSq1 \mid bSq3, Sq2 \to aSq2 \mid bSq3, Sq3 \to aSq3 \mid bSq3 \mid a \mid b\}$

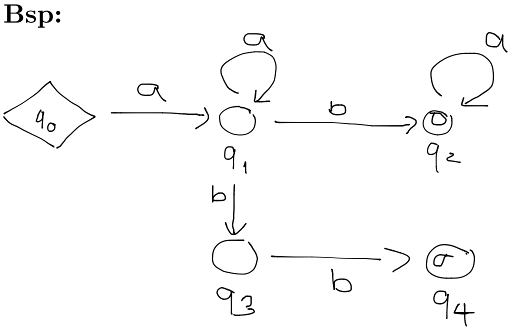
1. Zuordnung: Zustand $\mapsto$ Nichtterminalsymbol

2. Jedem Pfeil im Diagramm ordnen wir eine oder zwei Regeln zu
   (a) $q_i \xrightarrow{a} q_j$ mit $a_j \notin F$
   $\Rightarrow Sq_i \to aSq_j$
   (b) $q_i \xrightarrow{a} a_j$ mit $q_j \in F$
   $Sq_i \to aSq_j$
   $Sq_i \to a$

Bsp:
a) Das Wort aab akzeptieren. $q_0 \to q_1 \to q_1 \to q_3$
b) Das Wort aab generieren. $Sq_0 \Rightarrow aSq_1 \Rightarrow aaSq_1 \Rightarrow aab$

1.5 Nicht-deterministische, endliche Automaten (NFA)

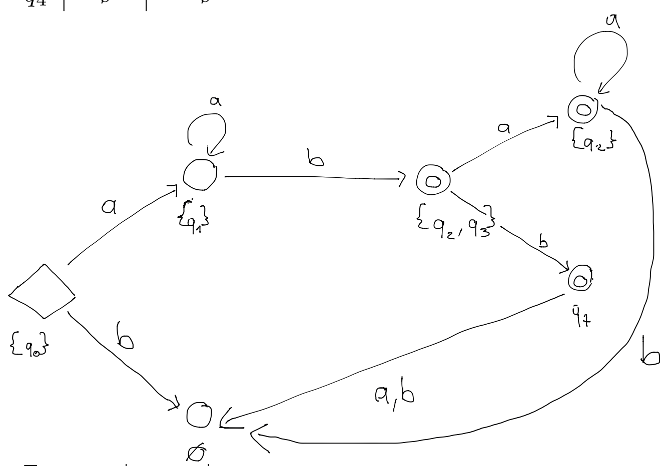
Bsp:



$baa \notin L(A)$, $aab \in L(A)$

δ	a	b
q_0	$\{q_1\}$	$\emptyset$
q_1	$\{q_1\}$	$\{q_2, q_3\}$
q_2	$\{q_2\}$	$\emptyset$
q_3	$\emptyset$	$\{q_4\}$
q_4	$\emptyset$	$\emptyset$

Satz: Vor. $A \in NFA$ **Beh.** $\exists B \in DFA : L(A) = L(B)$



$\overline{\delta}$	a	b
$\{q_0\}$	$\{q_1\}$	$\emptyset$
$\{q_1\}$	$\{q_1\}$	$\{q_2, q_3\}$
$\{q_2, q_3\}$	$\{q_2\}$	$\{q_4\}$
$\{q_2\}$	$\{q_2\}$	$\emptyset$
$\{q_4\}$	$\emptyset$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$

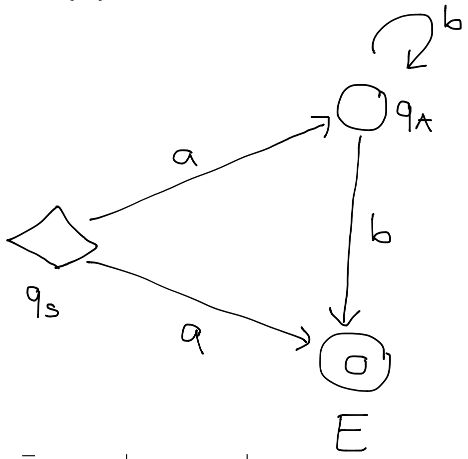
$\overline{F} = \{a_2, a_3\}, \{a_2\}, \{a_4\}$
Satz: Vor. $L \subset \Sigma^*$ regulär, **Beh.** $\exists A \in NFA : L(A) = L$
Beweis regulär $\Rightarrow \exists$ reguläre Grammatik $G = (N, T, R, S)$

mit $N = \{S, A, B, \dots\}$, $T = \Sigma$, $R = \{\dots A \to aB,\ A \to a \dots\}$

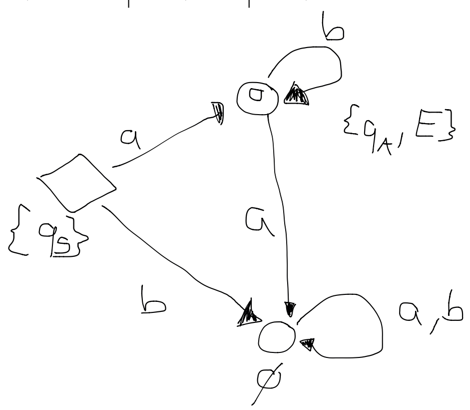
1. Jedem Nichtterminalsymbol ordnen wir einen Zustand zu. z.B. $A \mapsto q_A$
2. Jeder Regel vom Typ $A \to aB$ ordnen wir einen Pfeil im Diagramm zu: $q_A \xrightarrow{a} q_B$
3. Wir fügen einen **neuen** akzeptierenden Zustand E zu Q hinzu und für jede Regel $A \to b$ ein Pfeil $q_A \xrightarrow{b} E$
   $Q = \{q_S, q_A, q_B, \dots, E\}$

1.6 DFA

Bsp: $\Sigma = \{a, b\}$, $G = (N, T, R, S)$
mit $N = \{S, A\}$, $T = \Sigma$, $R = \{S \to aA, S \to a, A \to bA, A \to b\}$
$\mathbf{A} := (Q, \Sigma, \delta, q_S, F)$
$Q = \{q_S, q_A, E\}$
$F = \{E\}$



$\bar{\delta}$	a	b
$\{q_S\}$	$\{q_A, E\}$	$\emptyset$
$\{q_A, E\}$	$\emptyset$	$\{q_A, E\}$
$\emptyset$	$\emptyset$	$\emptyset$

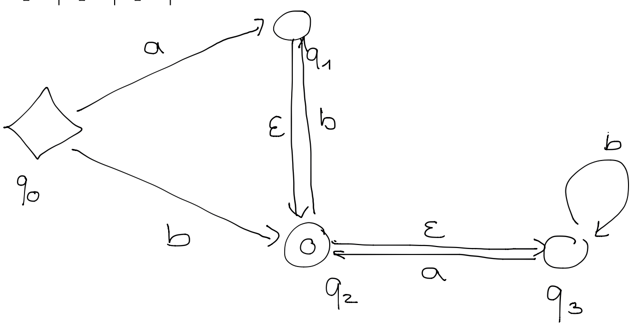


Def. Zwei endliche Automaten A und B heisen **äquivalent**: $\Leftrightarrow L(A) = L(B)$

1.7 NFA/ε

Bsp: (*) $A = (Q, \Sigma, \delta, q_0, F)$
$\Sigma = \{a, b\}$, $Q = \{q_0, q_1, q_2, q_3\}$, $F = \{q_2\}$

δ	a	b	ε
q_0	q_1	q_2	$\emptyset$
q_1	$\emptyset$	$\emptyset$	q_2
q_2	$\emptyset$	q_1	q_3
q_3	q_2	q_3	$\emptyset$



1.7.1 ε-Hüllen der Zustände

$[q]_\varepsilon^* := \{r \in Q \mid q \overset{\varepsilon^*}{\to} r\}$
Bsp anhand Bild 9
$[q_0]_\varepsilon^* = \{q_0\}$, $[q_1]_\varepsilon^* = \{q_1, q_2, q_3\}$, $[q_2]_\varepsilon^* = \{q_2, q_3\}$, $[q_3]_\varepsilon^* = \{q_3\}$
NFA
$B = (\overline{Q}, \Sigma, \overline{\delta}, \overline{q_0}, \overline{F})$

$$\overline{\delta}(q, a) = \bigcup_{r \in [q]_\varepsilon^*} \delta(r, a) \tag{1}$$

$\overline{\delta}$	a	b
q_0	$\{q_1\}$	$\{q_2\}$
q_1	$\{q_2\}$	$\{q_1, q_3\}$
q_2	$\{q_2\}$	$\{q_1, q_3\}$
q_3	$\{q_2\}$	$\{q_3\}$

$\overline{F} = \{q \in Q \mid [q]_\varepsilon^* \cap F \neq \emptyset\}$

1.8 2.4 Eigenschaften regulärer Sprachen

Sei Σ ein Alphabet
$C \subset P(\Sigma*)$ Menge von Sprachen
Frage: Führen Operationen auf den Elementen von C aus C heraus?
Bsp: von Operationen
$\overline{L} := \Sigma * \backslash L$
$L_1 \cup L_2$
$L_1 \cap L_2$
$L_1 \cdot L_2 = \{\omega_1\omega_2 | \omega_1 \in L_1, \omega_2 \in L_2\}$
Bsp: $L_1 = \{0, 1\}$,$L_2 = \{\varepsilon, 1\} \Rightarrow L_1 \cdot L_2 = \{0\varepsilon, 1\varepsilon, 01, 11\}$ **Notation:** $L^0 := \{\varepsilon\}$
$L^1 := L$
$L^2 := L \cdot L$ (Konkatenation)
$L^3 := L \cdot L^2 = L^2 \cdot L$

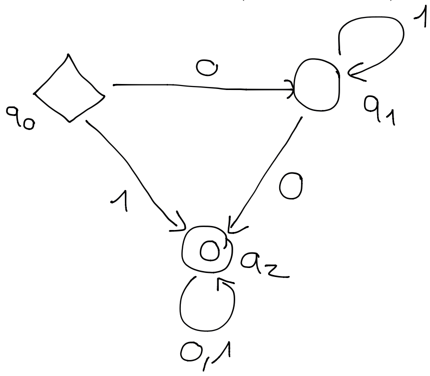
1.8.1 *- oder Kleene-Operation

$L^* := L^0 \cup L^1 \cup L^2 \cup L^3 \cup \ldots$
Bsp: $\Sigma = \{0, 1\}$, $\Sigma^* = \{\varepsilon\} \cup \{0, 1\} \cup \{00, 01, 10, 11\} \cup \ldots$ **Notation:** $Reg_\Sigma :=$ Menge der regulären Sprachen

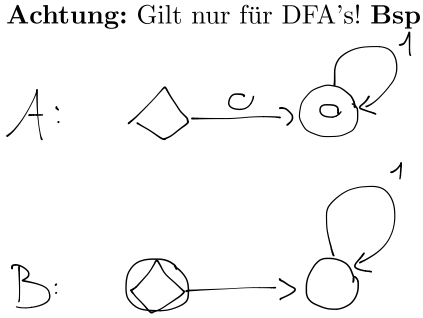
über Σ.
Satz: Vor. $L_1, L_2 \in Reg_\Sigma$
Beh. $\overline{L_1}, L_1 \cup L_2, L_1 \cap L_2, L_1 \cdot L_2, L_1^* \in Reg_\Sigma$

Bew.

1. $I_1 \in Reg_\Sigma \Rightarrow \exists A = (Q, \Sigma, \delta, q_0, F) \in DFA$ mit $L(A) = L$



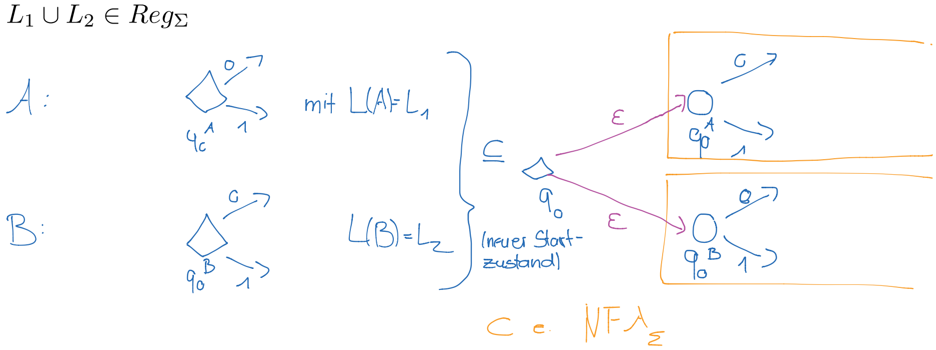
$L(A) = \{1w, 01^*0\omega | \omega \in \Sigma^*\}$
$\overline{L(A)} = \{01^*, \varepsilon\}$
$\overline{A} := (Q, \Sigma, \delta, q_0, \overline{F} := Q \backslash F)$
$L(\overline{A}) = I$
Achtung: Gilt nur für DFA's! **Bsp:**



: $L(A) = \{01^n \mid n \in \mathbb{N}\}$

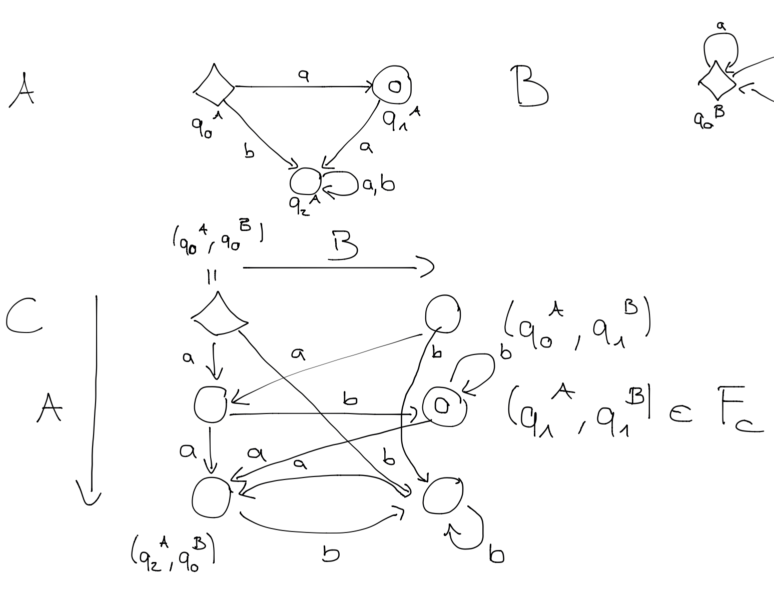
: $L(B) = \{\varepsilon\}$

2. $L_1 \cup L_2 \in Reg_\Sigma$

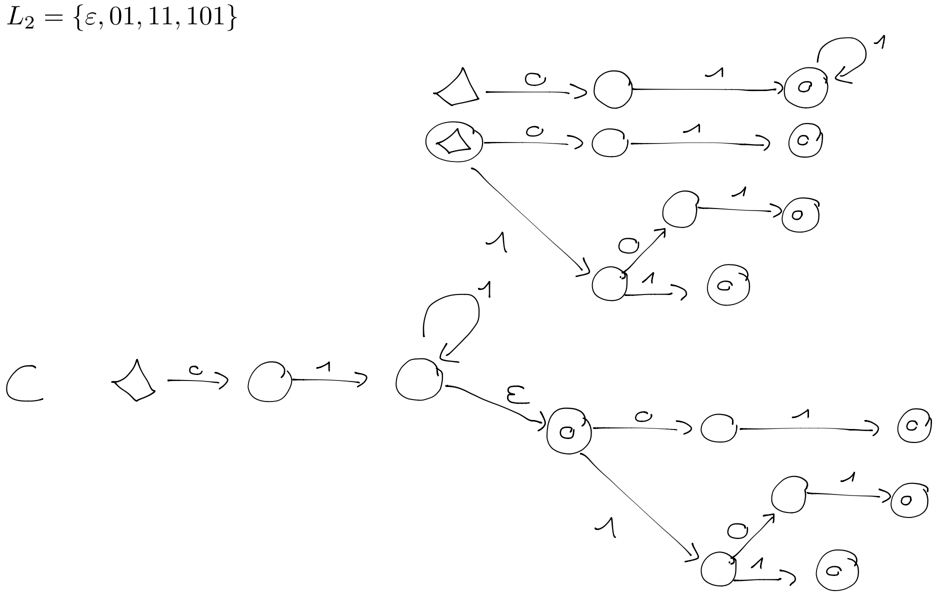


$L(C) = L_1 \cup L_2$

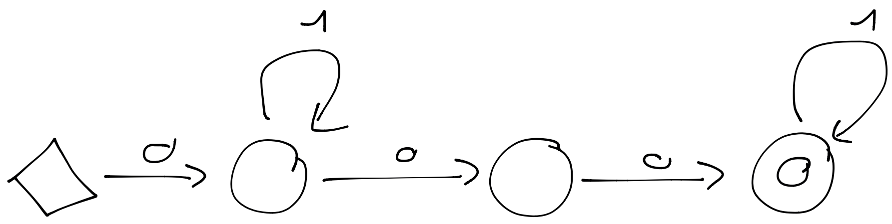
3. $L_1 \cap L_2 \in Reg_\Sigma$
1. Beweis De Morgan: $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$
Bsp:

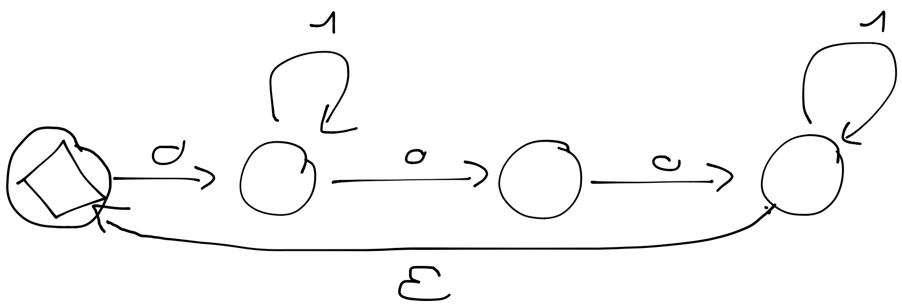


4. $L_1 \cdot L_2 \in Reg_\Sigma$
   $L_1 = \{01^+\}$
   $L_2 = \{\varepsilon, 01, 11, 101\}$



5. $L_1^* \in Reg_\Sigma$
   $L = \{01^*001^*\}$





1.9 2.5 Das Pumping-Lemma und der Satz von Myhill-Nerode

Frage: Wie zeigen wir, dass eine Sprache $L \notin Reg_\Sigma$?
Gegeben: $L \in Reg_\Sigma \Rightarrow \exists A = (Q, \Sigma, \delta, q_0, F) \in DFA$
$n := \mid Q \mid$ (Anzahl der Zustände)

1.9.1 Pumping Lemma

Vor. $L \in Reg_\Sigma$
Beh. $\exists n \in \mathbb{N}^* : \forall \omega \in L, \mid \omega \mid \geq n \exists x, y, z \in \Sigma^*$
x ist Weg vom Anfangszustand zum wiederholenden Zustand, y ist der Loop (vom wiederholenden zum wiederholenden Zustand), z der Weg vom wiederholenden Zustand zum akzeptierenden Zustand

1. $\omega = xyz$
2. $\mid y \mid \geq 1$
3. $\mid xy \mid \leq n$
4. $\forall i \in \mathbb{N} : xy^i z \in L$

Bsp:

1. $\Sigma = \{0, 1\}$, $L = \{0^k 1^k \mid k \in \mathbb{N}^*\}$
   $G = (N, T, R, S)$, $N = \{S\}$, $T = \{0, 1\}$, $R = \{S \to 01, S \to 0S1\}$ (kontextfrei) $\} = L(G) = L$
   Annahme L ist regulär
   $\forall n \in \mathbb{N}^*$: Wählen wir ein Wort $\omega_n \in L$ mit $\mid \omega_n \mid \geq n$.
   $\omega_n = 0^n 1^n = \left| \begin{array}{l|l} 0 \ldots\ldots 0 & 1 \ldots\ldots 1 \\ \hline \multicolumn{2}{l}{x \mid y \mid z \Rightarrow \text{3. Bedingung: } y = 0 \ldots 0 \geq 1} \end{array} \right| (\mid \omega_n \mid = 2n)$
   Bedingung 4: $xyyz \in L$ geht nicht, da mindestens eine weitere 0 hinzugefügt wird.
   $\Rightarrow$ Die Sprache ist kontextfrei denn $xz \notin L$

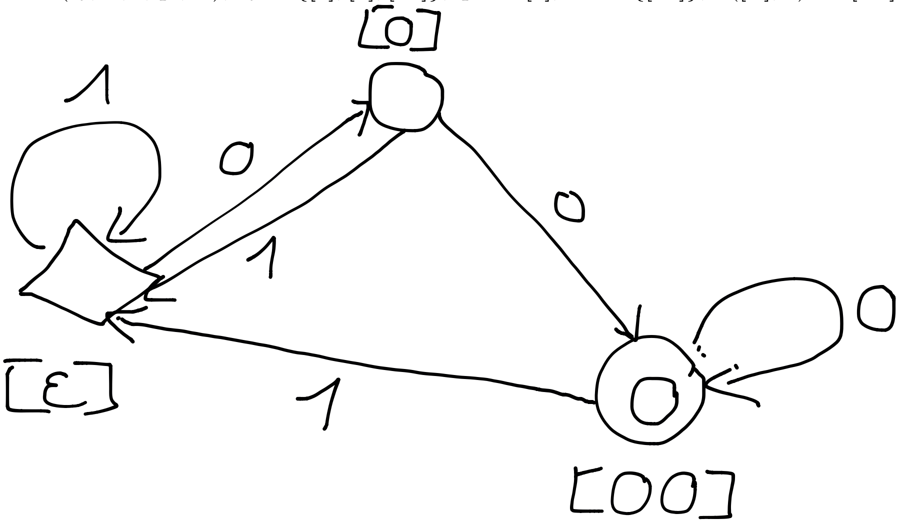
Bsp. $G^n_{arith} = (N, T, R, < arithmAusdruecke >)$ // $< AA >$
$N = \{< AA >, < Vor >\}$, $T = \{(,), +, -, x_1, x_2, \ldots, x_n\}$
$R = \{< AA > \to < Var >, < Var > \to x_1, < Var > \to x_2, \ldots, < Var > \to x_n, < AA > \to (< AA > + < AA >),$
$< AA > \to (< AA > \cdot < AA >)\}$
Augfaben:

1. Gegeben für Bsp für Wörter aus $L(G^n_{arith})$
   $x_1, x_2, x_n, (x_1 + x_2)$

2. Welcher Klasse gehört G^n_{arith} an.
   kontextfrei

3. In welchek Klasse liegt $L(G^n_{arith})$
   Es wird angenommen, dass die Sprache nicht regulär ist, da man Zählen muss, wieviele Klammern geöffnet worden sind.
   Annahme: $L(G^n_{arith}) \in Reg$
   $n \in \mathbb{N}^*$ ω_n
   $\omega_n = \underbrace{(((((\dots(}_{n} x_1 + x_2) \dots)$
   Da $\mid xy \mid \leq n$ und $y \neq \varepsilon \Rightarrow y = (^n \mid n \in \mathbb{N}$
   $\omega_n = xyz$, $\tilde{\omega} = xyyz \notin L(G^n_{arith})$

Falls man Zählen muss ist die Sprache mit grosser Wahrscheinlichkeit keine reguläre Sprache. Zählen bedeutet zum Beispiel, dass bei $0^n 1^n$ man die Nullen zählen muss da es genau gleich viele Einsen haben muss.
Bsp. $\Sigma = \{0, 1\}$
$L = \{\omega \in \Sigma^* \mid \omega \text{ endet auf } 00\}$
Erste Frage: Was sind die Äquivalezklassen? $R_L = [\varepsilon], [0], [00]$, $\Sigma^* = [\varepsilon] \cup [0] \cup [00]$
$A = (Q, \Sigma, \delta, q_0, F)$, $Q = \{[\varepsilon], [0], [00]\}$, $q_0 := [\varepsilon]$, $F := \{[00]\}$, $\delta([\omega], a) := [\omega a]$



1.10 Minimierung endlicher Automaten (gilt nur für DFA)

Problem: Gegeben: $A = (Q, \Sigma, \delta, q_0, F)$ mit $L(A) = L$
Gesucht: Minimaler DFA, der L akzeptiert.
Notation: Sei $q \in Q$
$L(A, q) := \{\omega \in \Sigma^* \mid r_A(q, \omega) \in F\}$
Bsp. $L(A, q_0) = L(A)$
Wir führen auf Q eine Relation RL_A ein:
Seien $q_l, q_j \in Q$
$(q_l, q_j) \in RL_A :\Leftrightarrow L(A, q_i) = L(A, q_j)$
Bem. RL_A ist eine Äquivalenzrelation

1.10.1 Minimaler Automat

1. Elimination von aus q_0 nicht erreichbaren Zuständen

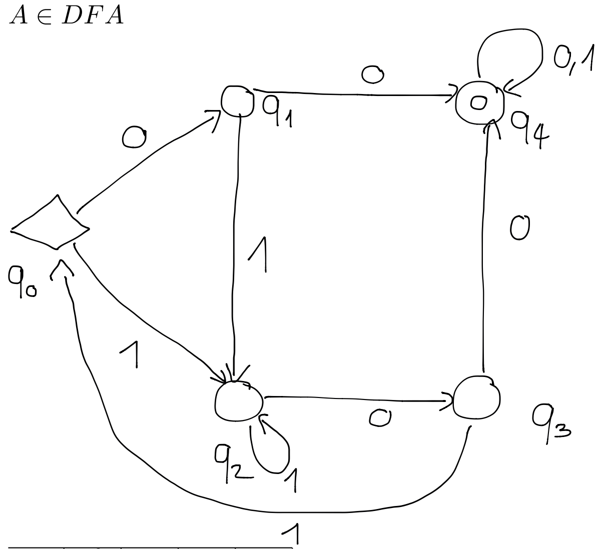
2. Bestimmen der Äquivalezklassen von RL_A

3. $A_{Min} = (\bar{Q}, \Sigma, \bar{\delta}, \bar{q_0}, \bar{F})$
   mit $\bar{Q} := \{[q] \mid q \in Q\}$, $\bar{q_0} := [q_0]$, $\bar{F} := \{[q] \mid q \in F\}$, $\bar{\delta} := [\delta(q, a)]$

1.10.2 Algorithmus zur Bestimmung der Äquivalezklassen von RL_A

1. $\forall q_i, q_j \in Q$ mit $q_i \in Q \backslash F$ und $a_j \in F \Rightarrow [q_i] \neq [q_j]$
   Beweis. $\varepsilon \notin L(A, q_i)$ und $\varepsilon \in L(A, q_j)$
2. Sei $[q_i] \neq [q_j]$ und $\tilde{q_k}, \tilde{q_e} \in Q$
   $\exists a \in \Sigma : \left\{ \begin{array}{c} \delta(\tilde{q_k}, a) = q_i \\ \delta(\tilde{q_e}, a) = q_j \end{array} \right\} \Rightarrow [\tilde{q_k}] \neq [\tilde{q_e}]$
   $L(A, q_i) \neq L(A, q_j) \Rightarrow L(A, \tilde{q_k} \neq L(A, \tilde{q_e})$

$A \in DFA$



q_1	$\star_1^0$	-	-	-
q_2		$\star_1^0$	-	-
q_3	$\star_1^0$		$\star_1^0$	-
q_4	$\star_0$	$\star_0$	$\star_0$	$\star_0$
	q_0	q_1	q_2	q_3

$RL_A = \{[q_0], [q_1], \cancel{[q_2]}, \cancel{[q_3]}, [q_4]\}$