Tutorium Theorie I, 24.11.2009

Note Title 11/3/2009

Reguläre Ausdruck -> Sprache

Übung NBA mit Umformung 3.6

5.4.

Zustandselimination (z.B. 5.3)

$$L\left((1+0)^* 1 0^* + 1\right)$$

$$= L((1+0)^*) \circ L(1) \circ L(0^*) \cup L(1)$$

$$= L(1+0)^* \circ \{1\} \circ L(0)^* \cup \{1\}$$

$$= (L(1) \cup L(0))^* \circ \{1\} \circ \{0\}^* \cup \{1\}$$

$= (\{1\} \cup \{0\})^* \circ \{1\} \circ \{0\}^* \cup \{1\}$

$= \{0,1\}^* \circ \{1\} \circ \{0\}^* \cup \{1\}$

$\{0,1\}^* = \{w \mid w = w_1 .. w_n$ mit $w_i \in \{0,1\}\}$

$= \{w \mid (\exists u, \cancel{v}, x : u \in \{0,1\}^* \land \cancel{v = 1} \land x \in \{0\}^*$
$\land\ w = u\overset{1}{\cancel{v}}x)$ oder $w = 1\}$

$= \{w \mid w = u1x$ oder $w = 1$ für ein $u \in \{0,1\}^*, x \in \{0\}^*\}$

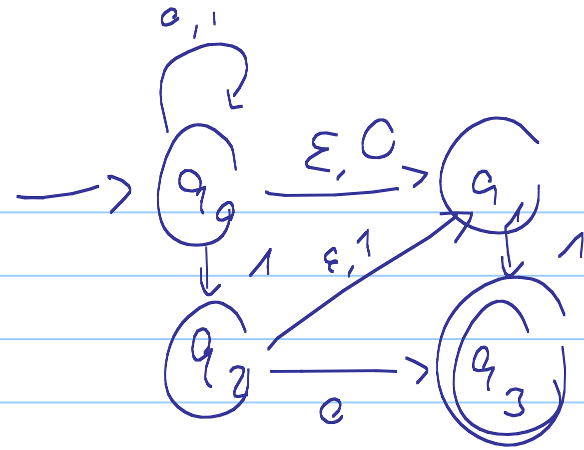
Argument: u, x können ε sein damit
ist $w = 1$ abgedeckt

$= \{w \mid w = u1x$ für ein $u \in \{0,1\}^*, x \in \{0\}^*\}$

Argument: eine 1 muß vorkommen
und dieser 1 können nur Nullen
davor beliebig 0, 1
also muß es eine letzte 1 geben

$= \{w \mid w$ enthält mindestens eine $1\}$

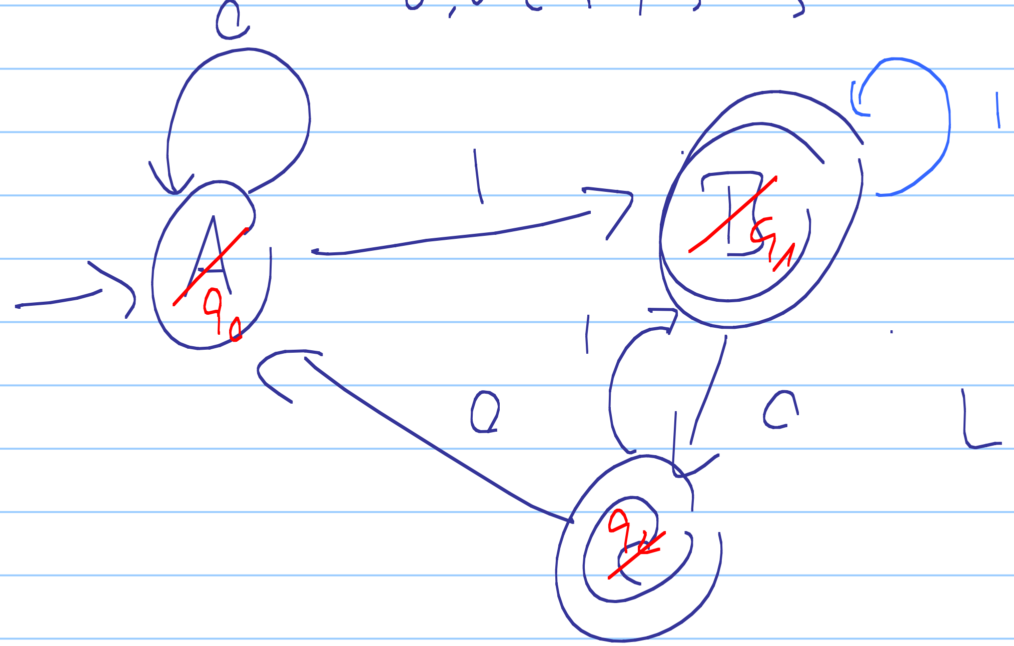
3.6



	0	1
→ A $q_{0,1}$	A q_{01}	B q_{0123}
* B q_{0123}	C q_{013}	B q_{0123}
* C q_{013}	A q_{01}	B q_{0123}

$L = \{ w \mid w = u1$ oder $w = v10$ für $u, v \in \{0,1\}^* \}$

A'



$L = \{ w \mid$ die letzte 2 Symbole enthalte keine 1 $\}$

Automaten sind äquivalent

Beweis mit A':

$L(A') = \{ w \mid (q_0, w) \vdash^* (q_1, \varepsilon)$ oder
$(q_0, w) \vdash^* (q_2, \varepsilon) \}$

Behauptung: für alle $w, v \in \{0,1\}^*$ gilt

(1) $(q_0, wv) \vdash^* (q_0, v) \iff$ letzte zwei Symbole von w enthalten keine 1

(2) $(q_0, wv) \vdash^* (q_1, v) \iff$ $w = u1$ für ein $u \in \{0,1\}^*$

(3) $(q_0, wv) \vdash^* (q_2, v) \iff$ $w = u10$ für ein $u \in \{0,1\}^*$

Induktion über Struktur von w

Basisfall $w = \varepsilon$: (2) (3) gilt "trivialerweise" denn wenn $w = \varepsilon$ dann

$$(q_0, wv) \not\vdash^* (q_1, v)$$

$$(q_0, wv) \not\vdash^* (q_2, v)$$

und $w \neq v1$ $w \neq v10$

Schrittfall (2) und (3) gelte für ein $y \in \{0,1\}^*$

a) Sei $w = y0$:

$\vdash^* (q_i, 0v)$

(2) $(q_0, wv) = (q_0, y\underline{0}v) \vdash^* (q_{q_1}, v)$

dann "falsch" da kein Übergang mit 0 nach q_1

es gibt kein q_i

Umgekehrt hat w nicht die Gestalt $v1$

damit falsch $\Leftrightarrow$ falsch ✓

(3) wenn $(q_0, wv) = \boxed{(q_0, y\underline{0v}) \vdash^* (q_i, \underline{0v})} \vdash (q_2, v)$

dann $q_i = q_1$

also per Annahme (2) $y = u1$ für ein $u \in \{0,1\}^*$

also $w = u10$ " " ✓

umgekehrt: wenn $w = u10$

dann $y = u1$ und per Annahme (2)

$$(q_0, y\underline{0v}) \vdash^* (q_1, 0v) \vdash (q_2, v) \checkmark$$

b) Sei $w = y1$

(2) wenn $(q_0, wv) = (q_0, y1v) \vdash^* (q_1, v)$

dann $w = u1$ für $u = y \in \{0,1\}^*$

Gegenrichtung:

wenn $w = u1$

dann $(q_0, u1v) \vdash^* (q_j, 1v)$ für ein q_j

und $(q_j, 1v) \vdash (q_1, v)$

also $(q_0, wv) \vdash^* (q_1, v)$

(3) wenn $(q_0, wv) = (q_0, y1v) \vdash^* (q_j, 1v) \vdash (q_2, v)$

dann falsch

und $w \neq u10$

also falsch $\Leftrightarrow$ falsch

Damit gilt (2), (3) für alle w, v

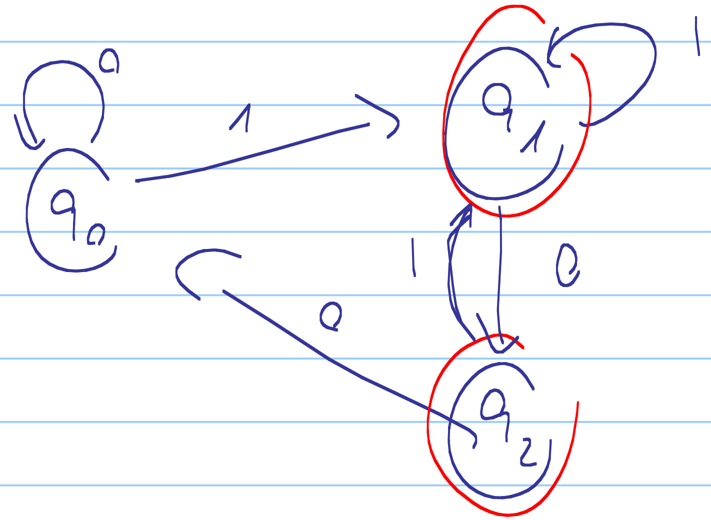
also $L(A') = \{ w \mid (q_0, w) \vdash^* (q_1, \varepsilon)$ oder $(q_0, w) \vdash^* (q_2, \varepsilon) ?$

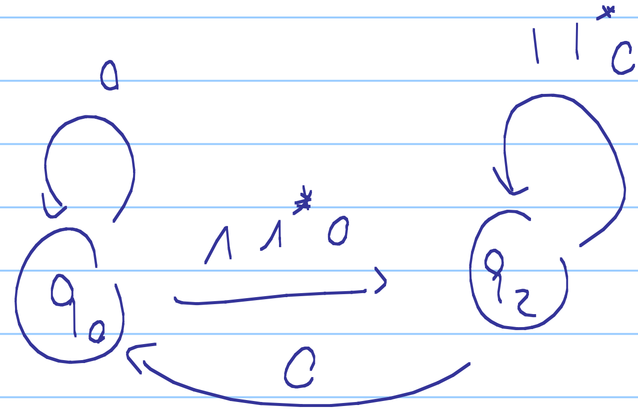
(2) (3)

$= \{ w \mid w = u1 \text{ oder } w = u10 \text{ für ein } u \in \{0,1\}^* \}$

$R = (0+1)^* 1 + (0+1)^* 10$

$= (0+1)^* 1 (0+\varepsilon)$

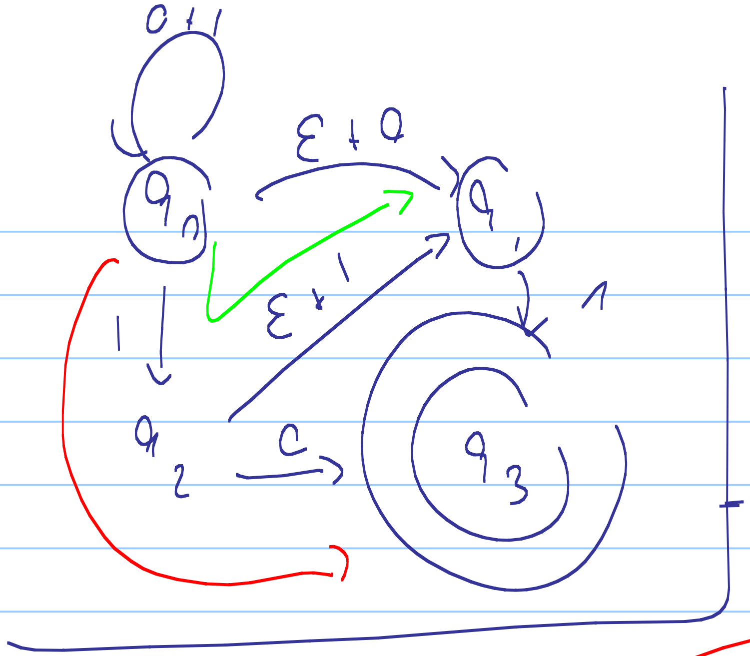




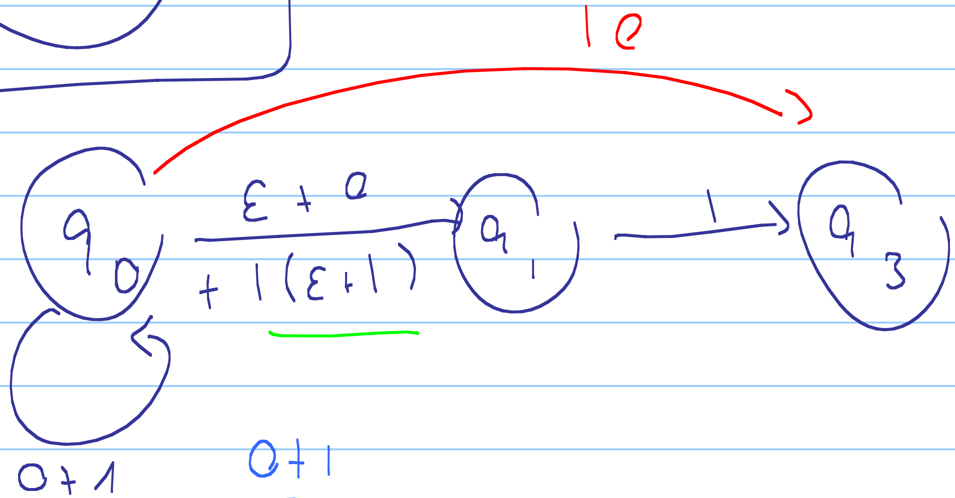
$\left(0^* (11^*0)(11^*0)^* 0 \right)^*$

$11^*0 (11^*0)^*$

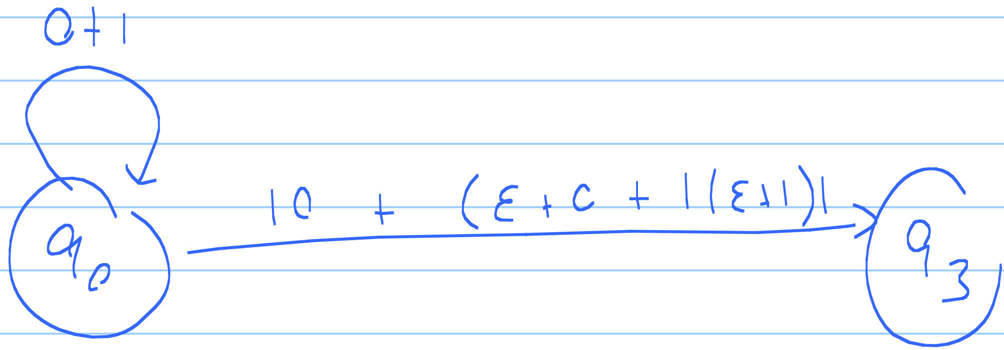
Eliminiere q_1 bzw q_2



eliminiere q_2



Eliminiere q_1



$$R = (0+1)^* (10 + 1 + 01 + 1(\varepsilon+1)1)$$

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

$$+ (0+1)^* 01$$
$$+ (0+1)^* 11$$
$$+ (0+1)^* 111$$

subsumiert

nehm

$$(0+1)^* 1 (0+\varepsilon)$$

Leichte da wenig Schleife und ein Endzustand

5.4 Algorithmus: Eingabe DEA A, Zahl n

Ausgabe $|\{ w \in \Sigma^n \mid w \in L(A) \}|$

Randbedingung: polynomieller Aufwand in $|A|$ und n

Problem: Durchzählen geht nicht da Σ^n Worte zu testen

Idee: wir brauchen nur die Zahl, nicht die Wörter

- Zähle wieviel Wörter der Länge k in Zustand q_i enden

Tabelle $\#_{k,i}$ wobei $Q = \{q_0, \dots, q_m\}$

$k \leq n$

$i \leq m$

Beobachtung: $\#_{0,0} = 1 = |\{w \mid |w| = 0 \wedge \hat{\delta}(q_0, w) = q_0\}|$

$\#_{0,i} = 0$ für $i \neq 0$

gibt erste Zeile der Tabelle

$$\#_{k+1,i} = \sum_{j=1}^{m} \#_{k,j} \cdot |\{a \mid \delta(q_j, a) = q_i\}|$$

liefert Zeile $k+1$ (für alle i)

Wiederhole bis $k+1 = n$

Tabellen Konstruktion mit iterativem Algorithmus

Berechne Zeile 0
und iterativ Zeilen 1 .. n

Aufwand ist: $n \cdot m$ Tabelleneinträge

und jeder Schritt verlangt m Summationen
$\cdot \; |\Sigma|$ Elemente

Insgesamt

$$n \cdot \underbrace{m^2 \cdot |\Sigma|}_{|A|^2} \quad \text{Schritte (Polynomial)}$$

Bei Länge 10, 10 Zustände 2 Symbole 2000
$> 2^{10}$

Länge 20 " "
4000 Schritte statt $2^{20} = 1000\,000$

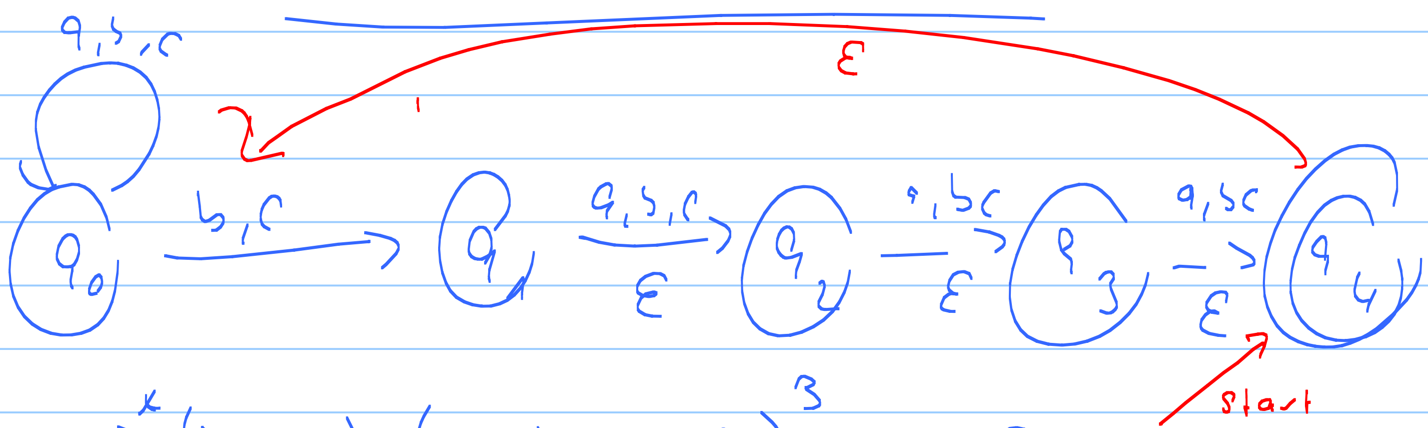
Ausdruck für $\{\}$ = "die leere Menge"

Standard $\quad L(\boxed{\emptyset}) = \{\} = \emptyset$

$$L(\emptyset \circ abc) = \{\}$$

Achtung $\quad L(\emptyset^*) = \{\varepsilon\} \qquad (\hat{=}\ 0^0 = 1)$

4.6



$$(a+b+c)^*(b+c)(a+b+c+\varepsilon)^3 + \varepsilon$$

Fehlt ε als zulässiges Wort: mache q_4 zum Startzustand und ergänze ε-Übergang nach q_0