AFS

1)
M1 = { a, b, c } ; M2 = { b, d, e, f }
Vereinigung: M3 = M1 oder M2 = { a, b, c, d, e, f }
Durchschnitt: M3 = M1 und M2 = { b }
Differenz: M3 = M1 \ M2 = { a, c }
Kreuzprodukt: M3 = M1 x M2 = { ab, ad, ae, af, bb, bd, be, bf, cb, cd, ce, cf }

Eine Menge M heißt abzählbar unendlich, wenn sie zur Menge N der natürlichen Zahlen gleichmächtig ist. Alle anderen unendlichen Mengen sollen überabzählbar unendlich heißen.
Die abzählbare Unendlichkeit einer Menge M bedeutet also nichts anderes, als dass M mit den natürlichen Zahlen durchnummeriert werden kann, quasi abgezählt werden kann.

2)
Alphabet (Sigma): Eine endliche nicht-leere Menge von Symbolen

Wort: Eine Aneinanderreihung von Symbolen eines Alphabets

Sprache (T): Die Menge aller Eingabefolgen, die von einem Automaten A akzeptiert werden, nennt man die Sprache T des Automaten

endlicher Automat:
-Verfügt über eine endliche Menge von Zuständen
-Folgezustand ist von Eingabezeichen und dem momentaten Zustand abhängig

Eingabe: dient zur Versorgung des Automaten mit Eingabedaten ei aus den sog. Eingabealphabet Sigma

Interne Zustände: werden als Reaktion auf die Eingabe durchlaufen

Ausgabe: sind die vom Automaten i.d.R. produzierten Ausgabedaten mit A = Ausgabealphabet

2 unterschiedliche Automatenmodelle sind äquivalent (s =~ s') wenn die Menge der Worte, die in einen Endzustand führen, für beide identisch ist delta(s,w) Element von F <=> delta(s', w) Element von F für alle w Element von Sigma*.

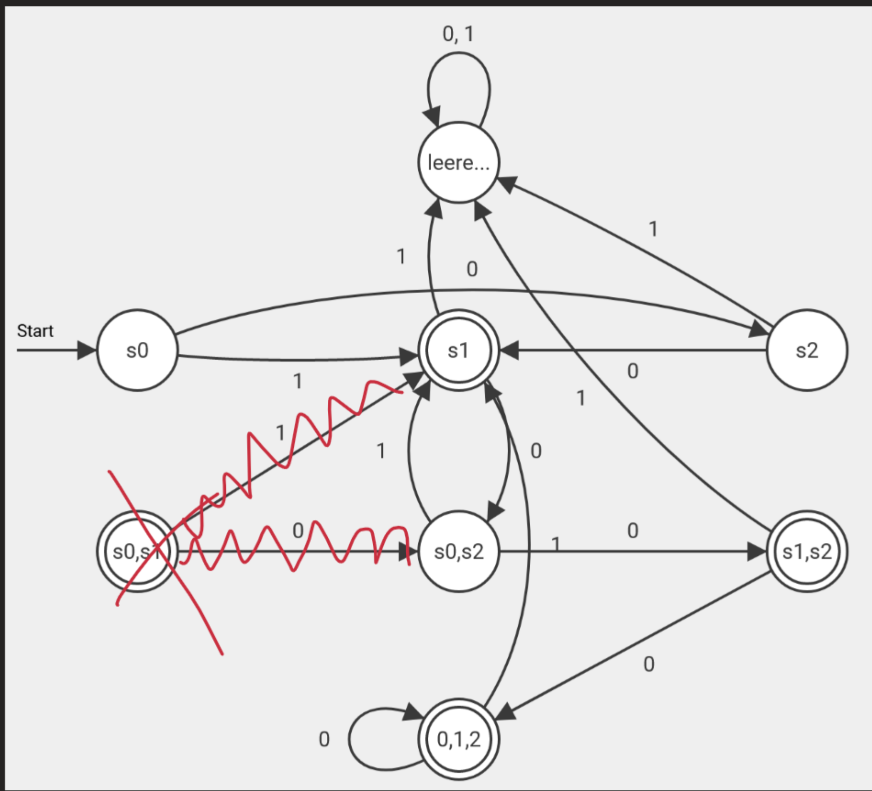
Eine Sprache ist regulär wenn sie von einem endlichen Automaten akzeptiert wird

Sigma = {a, b} ; S = Knoten ; F = Endknoten ; S0 = Startknoten => A = (S, S0, F, Sigma, S)

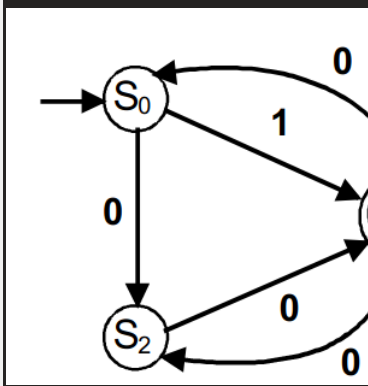
3)
-Die Vereinigung zweier entscheidbarer Mengen ist entscheidbar => wahr
-Jede von einem DFA akzeptierte Sprache wird auch von einem äquivalenten NFA akzeptiert, der möglicherweise weniger Zustände hat => wahr
-Jede Sprache enthält das leere Wort => falsch
-Seinen La, Lb ∈ Typ3 zwei beliebige reguläre Sprachen. Dann gilt: La ∪ Lb ∈ Typ3. => wahr
-Es gibt reguläre Sprachen, die nicht von einem nichtdeterministischen endlichen Automaten mit ε-Übergängen akzeptiert werden können. => falsch
-Alle regulären Sprachen sind endlich => falsch

4)
Teilmengenkonstruktion

4a)

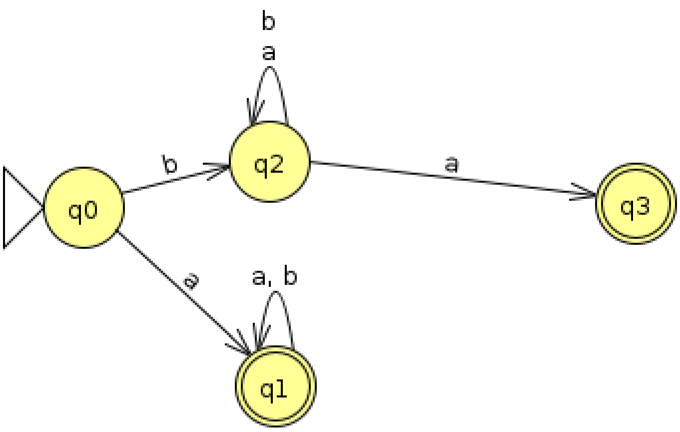


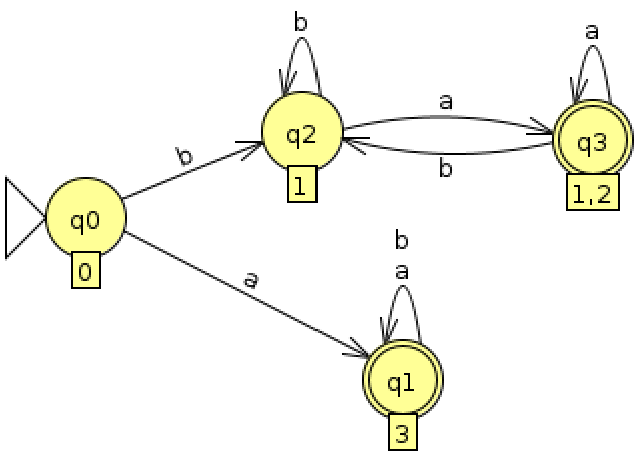
	0	1
Leere Menge	Leere Menge	Leere Menge
S0	S2	S1
S1	S0,S2	Leere Menge
S2	S1	Leere Menge
~~S0,S1~~	~~S0,S2~~	~~S1~~
S0,S2	S1,S2	S1
S1,S2	S0,S1,S2	Leere Menge
S0,S1,S2	S0,S1,S2	S1



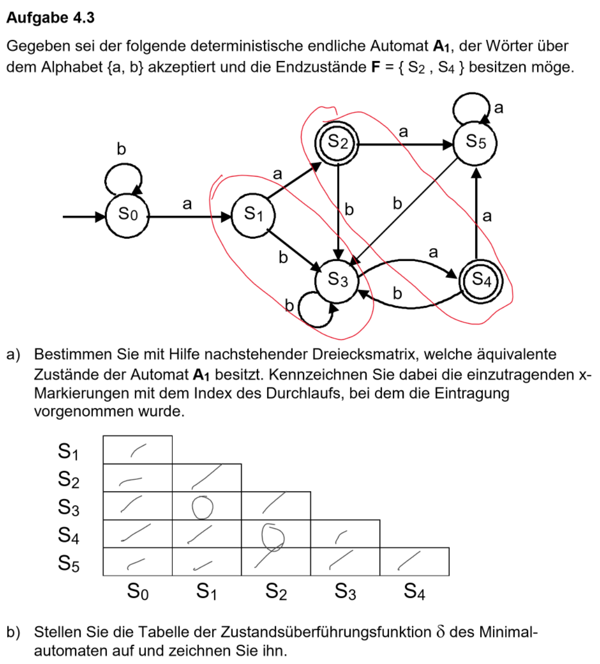
c)

$\alpha = \{\omega \in \sum^* \mid (1 \mid 00)\ ((01)^* \mid (0000^*1)^*)^*\ (\varepsilon \mid (000^*))\}$





DEA minimieren

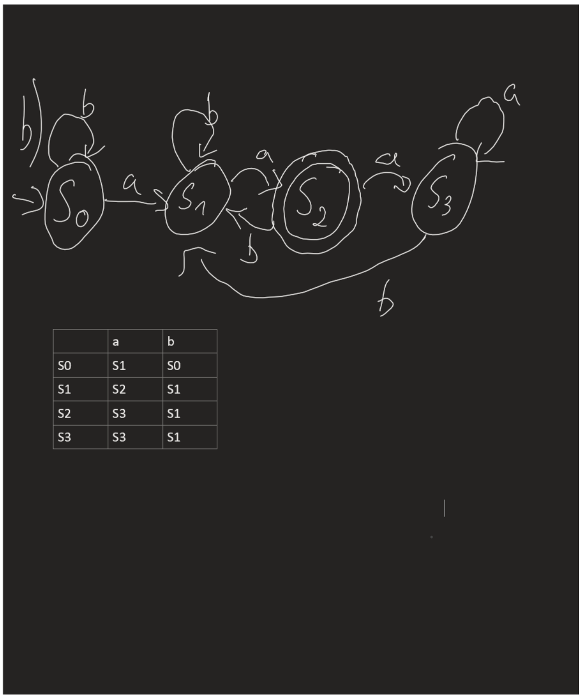


Aufgabe 4.3

Gegeben sei der folgende deterministische endliche Automat **A₁**, der Wörter über dem Alphabet {a, b} akzeptiert und die Endzustände **F** = { S_2 , S_4 } besitzen möge.

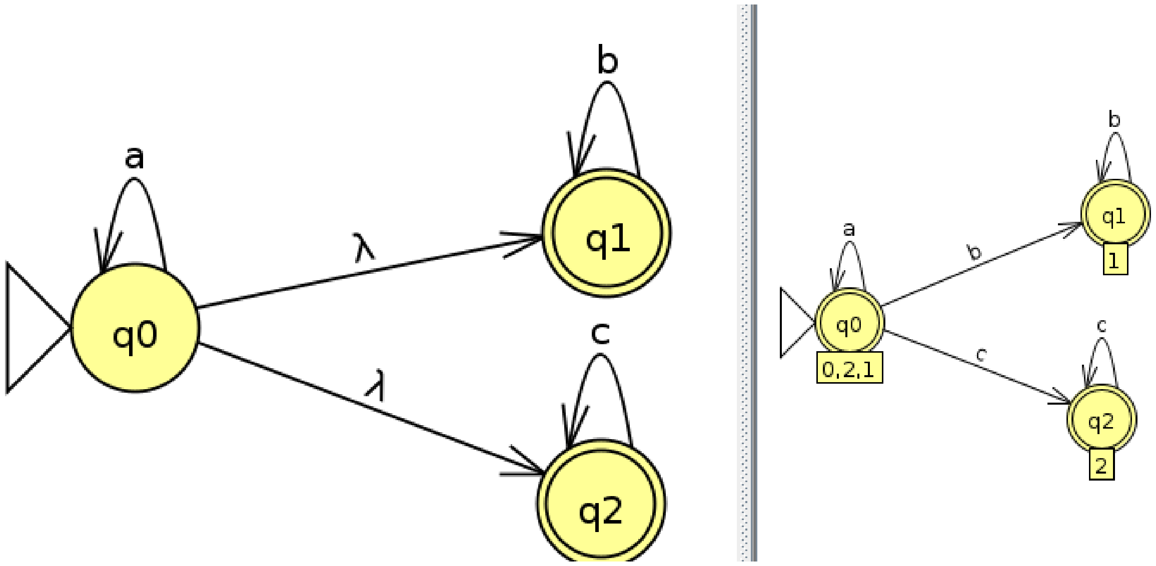
a) Bestimmen Sie mit Hilfe nachstehender Dreiecksmatrix, welche äquivalente Zustände der Automat **A₁** besitzt. Kennzeichnen Sie dabei die einzutragenden x-Markierungen mit dem Index des Durchlaufs, bei dem die Eintragung vorgenommen wurde.

b) Stellen Sie die Tabelle der Zustandsüberführungsfunktion δ des Minimalautomaten auf und zeichnen Sie ihn.



	a	b
S0	S1	S0
S1	S2	S1
S2	S3	S1
S3	S3	S1

5)



6) Chomsky

G = (N, T, P, S) ; Nichtterminale, Terminale, Produktion, Syntaxregeln (der Produktion)

Typ 0 (allgemein, nicht eingeschränkt): $\alpha \rightarrow \beta$ mit $\alpha \neq \varepsilon$ [$\alpha, \beta \in (N \cup T)^*$]

Typ 1 (kontextsensitiv): $\alpha A \beta \rightarrow \alpha \gamma \beta$ mit $\gamma \neq \varepsilon$ [$A \in N$; $\alpha, \beta, \gamma \in (N \cup T)^*$]

Typ 2 (kontextfrei): $A \rightarrow \gamma$ mit $A \in N$; oder $\gamma = \varepsilon$ [$\gamma \in (N \cup T)^*$]

Typ 3 (rechtslineare – N ist ganz rechts!): $A \rightarrow \varepsilon$; $A \rightarrow a$ oder $A \rightarrow aB$; mit $a \in T$; $A, B \in N$

Pumping Lemma

7)
CYK

8)
Kellerautomat:

$\delta(s\ 0, a, k0) = (s0, ak0)$; $\delta(s0, a, a) = (s0, aa)$;
$\delta(s0, b, a) = (s1, \varepsilon)$; $\delta(s\ 1, b, a) = (s1, \varepsilon)$

"aaabb"

k0	a	a	a	a	a
	k0	a	a	a	k0
		k0	a	k0	
			k0		

"aabbb"

k0	a	a	a	k0	?
	k0	a	k0		
		k0			

"abab"

k0	a	k0	?
	k0		

"aabb"

k0	a	a	a	k0
	k0	a	k0	
		k0		