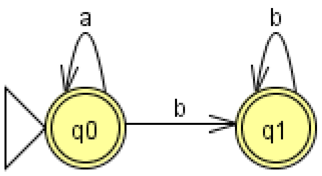
Aufgabe 1

a) $L(A) = \{ a^n b^m \mid n, m \in N_0 \}$
$A = (\Sigma, S, s_0, F, R)$
$\Sigma = \{ a, b \}$
$S = \{ q_0, q_1 \}$
$s0 = q_0$
$F = \{ q_0, q_1 \}$

R als Graph

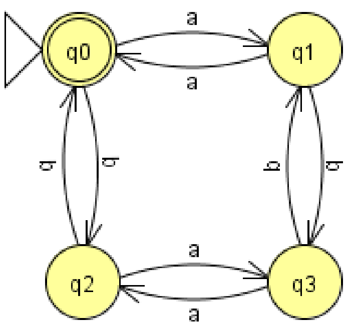


R als Tabelle

	a	b
q_0	q_0	q_1
q_1	-	q_1

b) $L(A) = \{ w \in \Sigma^* \mid$ w enthält eine gerade Anzahl a's und eine gerade Anzahl b's $\}$ (0 sei gerade)
$A = (\Sigma, S, s_0, F, R)$
$\Sigma = \{ a, b \}$
$S = \{ q_0, q_1, q_2, q_3 \}$
$s0 = q_0$
$F = \{ q_0 \}$

R als Graph



R als Tabelle

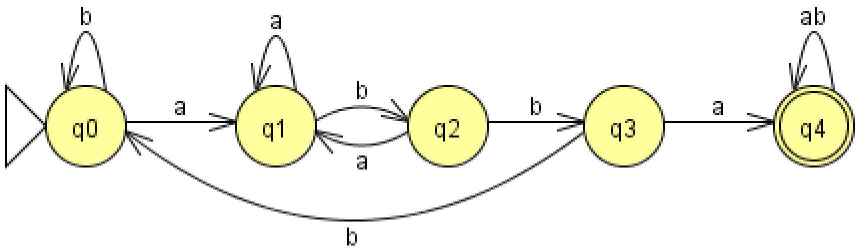
	a	b
q_0	q_1	q_2
q_1	q_0	q_3
q_2	q_3	q_0
q_3	q_2	q_1

c) aabaa: q_0 (a) $\rightarrow$ q_1 (a) $\rightarrow$ q_0 (b) $\rightarrow$ q_2 (a) $\rightarrow$ q_3 (a) $\rightarrow$ q_2
Wird nicht akzeptiert, da q_2 kein Endzustand ist.

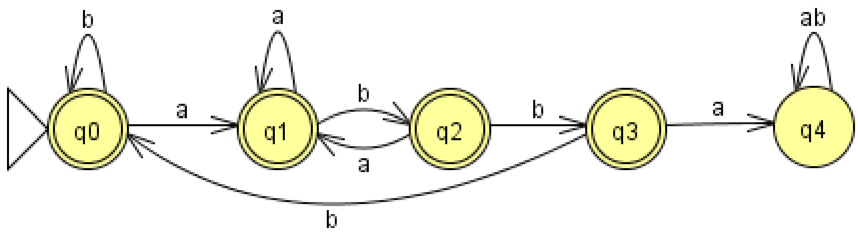
ababbb: q_0 (a) $\rightarrow$ q_1 (b) $\rightarrow$ q_3 (a) $\rightarrow$ q_2 (b) $\rightarrow$ q_0 (b) $\rightarrow$ q_2 (b) $\rightarrow$ q_0
Wird akzeptiert, da q_0 Endzustand ist.

Aufgabe 2

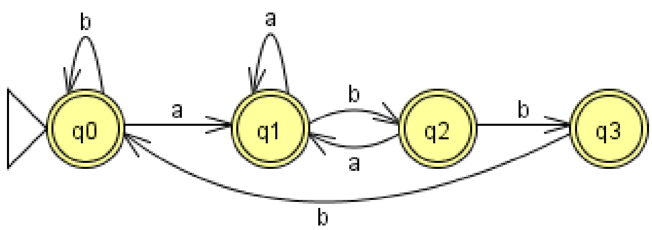
a) L(A) = { w ∈ Σ^* | w enthält „irgendwo“ die Kombination abba }



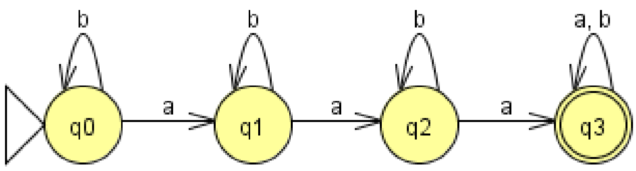
b) L(A) = { w ∈ Σ^* | w enthält nirgends die Kombination abba }



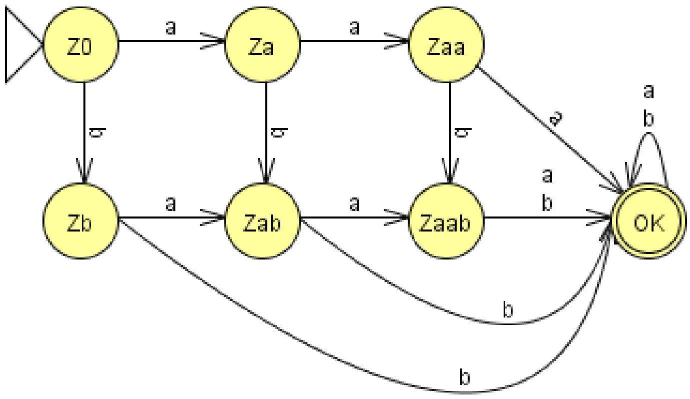
D.h. der gleiche Automat wie in a), nur dass alle Endzustände und Nicht-Endzustände vertauscht wurden. Oder, etwas einfacher:



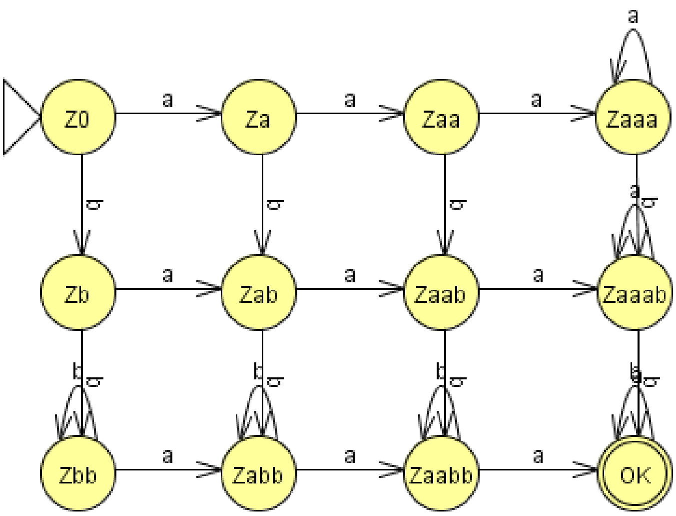
c) L(A) = { w ∈ Σ^* | w enthält mindestens 3 a's }



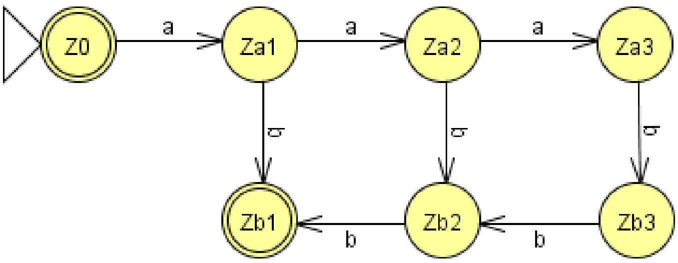
d) L(A) = { w ∈ Σ^* | w enthält mindestens 3 a's **oder** mindestens 2 b's (beides zulässig) }



e) $L(A) = \{ w \in \Sigma^* \mid$ w enthält mindestens 3 a's **und** mindestens 2 b's $\}$



f) $L(A) = \{ a^n b^n \mid n \in \mathbb{N}, n < 10 \}$ (d.h. die gleiche Anzahl a's und b's, bis zu 9 Stück)



(der Automat wird analog nach rechts weiter fortgesetzt)

g) $L(A) = \{ a^n b^n \mid n \in \mathbb{N}_0 \}$ (d.h. die gleiche Anzahl a's und b's, beliebig viele)

Wie in f) zu sehen, benötigt ein DEA n Zustände, um n a's zu zählen, damit anschließend die gleiche Anzahl an b's abgezählt werden kann. Für n = 20 muss der Automat also 40 Zustände haben (20 für die a's und entsprechend viele für die b's).

Wenn n beliebig groß sein darf, müsste der DEA unendliche viele Zustände haben. Denn wenn der DEA eine endliche Zustandsmenge hätte, könnte er maximal bis zur Anzahl der Zustände zählen. Sobald er ein Wort liest, das mehr Buchstaben hat, könnte er nicht weit genug zählen.

Endliche Automaten dürfen aber nur **endlich** viele Zustände haben. Daher gibt es keinen DEA für diese Aufgabe.

Eine Folgerung ist, das DEAs nicht als Compiler geeignet sind, also nicht prüfen können ob ein Text ein korrektes Programm in einer Programmiersprache ist.
Begründung: Programme sind geklammerte Texte – in Java beispielsweise mit { bzw. }.
Ein Programm kann beliebig viele öffnende Klammern haben, für die es die gleiche Anzahl schließender Klammern geben muss, sonst ist das Programm syntaktisch nicht korrekt.

Wie Aufgabe g) zeigt, gibt es keinen solchen DEA.