Übungsblatt 3

Hausaufgabe 5

(a) $r_1 = a^* + b^* + ((a + b)^* \cdot (a + bb))^*$

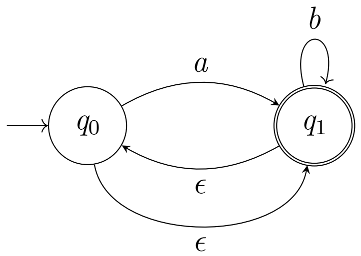
Ein Wort aus dem Alphabet $\{a, b\}^*$, welches nicht das Suffix ab hat kann neben den Basisfällen von ϵ, a, b entweder mit ba, bb oder aa enden. Der angegebene Ausdruck deckt also einmal diese Basisfälle, und im Fall eines längeren Wortes mit abwechselnden as und bs (also $(a + b)^*$) stellt er sicher, dass das Wort entweder mit a oder bb terminiert, sodass der Suffix ab nicht zustande kommen kann.

(b) $r_2 = b^* \cdot (ab^*ab^*)^* + a^*b \cdot (a^*ba^*ba^*)$

Im ersten Fall ($|w|_a$ gerade) können wir an jeder Stelle des Wortes beliebig viele bs lesen, allerdings muss die Menge an as immer gerade bleiben, also ist im umklammerten Ausdruck eine gerade Menge an as enthalten, welche beliebig oft mit beliebig vielen bs konkateniert werden können. Im zweiten Fall ($|w|_b$ ungerade) muss mindestens ein b gelesen werden, allerdings können davor und dazwischen wieder beliebige as gelesen werden, und es soll die Möglichkeit geben nach dem ersten b weiter eine gerade Menge an bs zu schreiben, welche durch das erste b immer eine ungerade Anzahl ergeben werden.

Hausaufgabe 6

$\mathcal{A}$



Hausaufgabe 7

$L(\mathcal{A}) = \bigcup_{q_f \in F} L^Q_{q_0,q_f} = L^Q_{1,1}$

Induktionsanfang

$L^{\varnothing}_{1,2} = b, L^{\varnothing}_{1,1} = b + \epsilon$
$L^{\varnothing}_{2,1} = a + b, L^{\varnothing}_{2,2} = \epsilon$

Induktionsschritt

$$L^Q_{1,1} = L^{\{1\}}_{1,1} \cup L^{\{1\}}_{1,1} \cdot (L^{\{1\}}_{1,1})^* \cdot L^{\{1\}}_{1,1}$$

$$\begin{aligned} L^{\{2\}}_{1,1} &= L^{\varnothing}_{1,1} \cup L^{\varnothing}_{1,2} \cdot (L^{\varnothing}_{2,2})^* \cdot L^{\varnothing}_{2,1} \\ &= b + \epsilon + b \cdot (\epsilon)^* \cdot a + b \\ &= \epsilon + ba + b \end{aligned}$$

$$\begin{aligned} \Rightarrow L^Q_{1,1} &= (\epsilon + ba + b) + (\epsilon + ba + b)(\epsilon + ba + b)^*(\epsilon + ba + b) \\ &= (\epsilon + ba + b)^* \\ &= (ba + b)^* \end{aligned}$$

$\Rightarrow r = (ba + b)^*, L(r) = L(\mathcal{A})$

Hausaufgabe 8

Zu beweisen:
Es gibt einen DEA $\mathcal{A}, L(\mathcal{A}) = L$
$\Rightarrow$ Es gibt einen DEA $\mathcal{A}', L(\mathcal{A}') = \text{minimal}(L)$

Wenn ein beliebiger Automat $\mathcal{A}$ die Sprache L erkennt, dann lässt sich aus diesem Automaten ein Automat $\mathcal{A}'$ konstruieren, welcher die Sprache minimal(L) erkennt. Das Prinzip funktioniert wie folgt: wenn ein Automat $\mathcal{A} = \{Q, \Sigma, q_0, \Delta, F\}$ einen akzeptierenden Zustand hat, welcher hinter bzw. nach einem anderen akzeptierenden Zustand liegt, dann wird dieser im Automat $\mathcal{A}' = \{Q, \Sigma, q_0, \Delta, F'\}$ aus der Menge F der akzeptierenden Zuständen von $\mathcal{A}$ entfernt. In dem Fall würde das bedeuten, dass dieser entfernte akzeptierende Zustand in $\mathcal{A}$ ein Wort akzeptieren würde, welches ein Präfix hätte, was ebenfalls in der Sprache $L(\mathcal{A})$ liegt und somit nicht in der Sprache minimal(L) enthalten sein darf.

Die neue Menge F' der akzeptierenden Zustände kann wie folgt ausgedrückt werden:

$F = \{f \in F \mid$ Für alle $w \in \Sigma^*$ gibt es keinen Pfad von
q_0 nach f wo ein weiterer Finalzustand f' im Lauf enthalten ist$\}$

Das Wort aab aus dem geg. Beispiel hat als Präfix a, welches in L enthalten ist und somit nicht in minimal(L) enthalten ist.
Somit lässt sich also aus jedem Automaten $\mathcal{A}$ mit $L(\mathcal{A}) = L$ ein Automat $\mathcal{A}'$ mit $L(\mathcal{A}') = \text{minimal}(L)$ konstruieren.

Es gilt also: L erkennbar $\Rightarrow$ minimal(L) erkennbar.