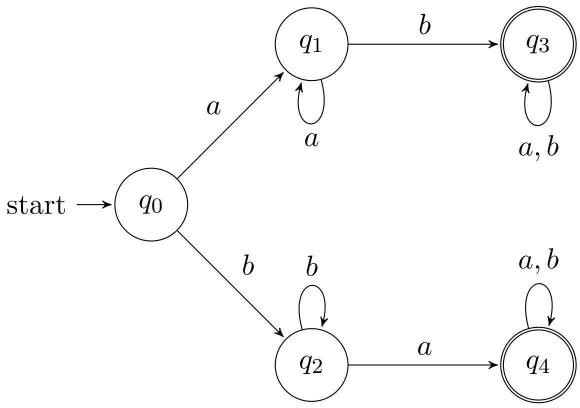
Gruppe $\boxed{3}$
Formale Systeme und Automaten

Phil Pützstück, 377247
Benedikt Gerlach, 376944
Sebastian Hackenberg, 377550

Hausaufgabe 6

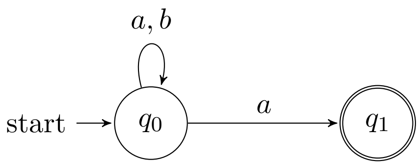
Aufgabe 5

a) Wir konstruieren zuerst einen DFA $\mathcal{A}$ mit $L(\mathcal{A}) = \{a,b\}^* \setminus (\{a^n \mid n \in \mathbb{N}\} \cup \{b^n \mid n \in \mathbb{N}\})$, welcher also nur alle Wörter über $\{a,b\}$, welche sowohl b als auch a enthalten, erkennt.
Dieser könnte z.B. wie folgt aussehen:



Dann wandeln wir den gegebenen NFA zu einem DFA $\mathcal{B}$ um und berechnen das Inklusionsproblem $L(\mathcal{B}) \subseteq L(\mathcal{A})$.

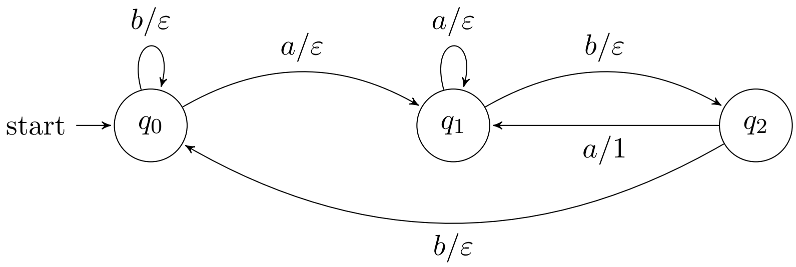
b) Wir konstruieren den NFA $\mathcal{A}$ mit $L(\mathcal{A}) = \{va \mid v \in \Sigma^*\}$. Der also alle Wörter über Σ^* erkennt, welche mit a enden. Dieser könnte z.B. wie folgt aussehen:



Da FA's nach Satz 2.40 unter dem Schnitt abgeschlossen sind, können wir den NFA $\mathcal{A} \cap \mathcal{B}$ berechnen, wobei $\mathcal{B}$ der gegebene NFA ist. Dieser erkennt nun alle Wörter aus $\mathcal{B}$ welche mit einem a aufhören. Nun können wir das Unendlichkeitsproblem auf dem NFA $\mathcal{A} \cap \mathcal{B}$ lösen. Wenn dies also unendlich ist, so kann es keine obere Schranke geben, ist er jedoch endlich so gibt es ein $k = \max\{|w| \mid w \in L(\mathcal{A} \cap \mathcal{B})\}$ welches eine obere Schranke der Länge aller Wörter aus $\mathcal{B}$, welche mit a aufhören, darstellt. Es ist $K := \{|w| \mid w \in L(\mathcal{A} \cap \mathcal{B})\} \subseteq \mathbb{N}$ und damit bzgl. $\leq$ totalgeordnet, was bei endlichkeit von K die Existenz eines Maximums aus K impliziert.

Aufgabe 6

a)



b)

Seien $\mathcal{M}_1 = (Q_1, \Sigma_1, \Sigma_2, \delta_1, q_{01}, \lambda_1), \mathcal{M}_2 = (Q_2, \Sigma_2, \Sigma_3, \delta_2, q_{02}, \lambda_2)$ die Mealy-Automaten, welche jeweils die Funktionen G_1, G_2 berechnen. Die Idee ist, beide Automaton gleichzeitig laufen zu lassen, wir machen also eine Produktkonstruktion. Wir übergeben die Ausgabe von $\mathcal{M}_1$ an die Transitionsfunktion von $\mathcal{M}_\in$, wechseln intern in beiden Automaton entsprechend die Zustände und geben nur das berechnete Symbol von $\mathcal{M}_2$ bei der Transition aus.

Der Mealy-Automat $\mathcal{M}$ mit $G_\mathcal{M} = G_2 \circ G_1$ ist also wie folgt definiert:

$$\mathcal{M} := (Q_1 \times Q_2, \Sigma_1, \Sigma_3, \delta, (q_{01}, q_{02}), \lambda)$$

wobei für $q_1 \in Q_1, q_2 \in Q_2, a \in \Sigma_1$ gilt:

$$\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, \lambda_1(q_1, a))) \quad \text{und} \quad \lambda((q_1, q_2), a) = \lambda_2(q_2, \lambda_1(q_1, a))$$

Aufgabe 7

a)

Wir haben $\mathcal{G} = (\{S, A\}, \{a, b, c\}, P, S)$ wobei P folgende Regeln hat:

$$S \to aSa \mid Ac \qquad\qquad A \to bAc \mid \varepsilon$$

Wir machen zuerst die n a's am Anfang und ende des Wortes, können dann zu einem anderen "Zustand" A übergehen wobei wir ein c so platzieren, dass es rechts der Mitte des Wortes vorkommt. Dann können wir in A die m b's und c's wie gewünscht erschaffen.

b)

Wir haben $\mathcal{G} = (\{S, A\}, \{a, b, c\}, P, S)$ wobei P die folgenden Regeln hat:

$$S \to aSc \mid A \qquad\qquad A \to bbAc \mid \varepsilon$$

Wir erschaffen zuerst die n a's am Anfang des Wortes bzw. n c's am Ende des Wortes. Dann können wir zum "Zustand" A übergehen und dort die weitern m bb's und c's entsprechend erschaffen.

c)

Wir haben $\mathcal{G} = (\{S\}, \{a, b, c\}, P, S)$ wobei P die folgenden Regeln hat:

$$S \to bS \mid Sb \mid aSc \mid \varepsilon$$

Wir erschaffen analog zu a) und b) die Teilworte u und v von w jeweils von der Mitte von w aus. Also gilt für eine Satzform xSy stets später $x \in (a + b)^*$, $y \in (b + c)^*$ und $|x|_a = |y|_c$. Wir können durch beliebiges einschieben von b's die gestalt von x und y um die Mitte von w beliebig formen um dann a sowie c gleichzeitig hinzuzufügen, ohne Worte aus der angegebenen Sprachen nicht abzudecken.

d)

Wir haben $\mathcal{G} = (\{S\}, \{a, b, c\}, P, S)$ wobei P die folgenden Regeln hat:

$$S \to cS \mid bS \mid bSaS \mid \varepsilon$$

Wir können also stets beliebig viel c's und b's überall einfügen. Um ein a in das Wort zu schreiben müssen wir jedoch irgendwo vorher mindestens ein b schreiben. Dies entspricht eben den Anforderungen.