

Musterlösung zu Übungsblatt 4

Aufgabe 1.

Sind die folgenden Aussagen wahr oder falsch?

- (a) Wenn L eine reguläre Sprache ist und $u, v \in L$, dann gilt $uv \in L$.
- (b) Wenn L eine reguläre Sprache ist, dann ist auch die Menge $\text{Pref}(L)$ aller Präfixe von Wörtern aus L regulär.
- (c) Wenn L eine reguläre Sprache ist, dann existiert ein NFA mit genau einem Startzustand und genau einem Endzustand, der L erkennt.
- (d) Wenn K und L regulär sind, dann ist auch $K \setminus L$ regulär.
- (e) Wenn L^* regulär ist, dann ist auch L regulär.
- (f) Wenn L regulär ist, dann ist auch $\{\text{rev}(w) \mid w \in L\}$ regulär, wobei $\text{rev}(a_1 \dots a_n) = a_n \dots a_1$.

Lösung zu Aufgabe 1.

- (a) falsch, sei z.B. $L = L(a|b)$, dann ist $a \in L, b \in L$ aber $ab \notin L$
- (b) wahr

Begründung

Sei M ein DFA für die reguläre Sprache L und F die Endzustandsmenge von M . Wir konstruieren aus M einen DFA für die Sprache $\text{Pref}(L)$ indem wir die Endzustandsmenge F um alle Zustände erweitern, von denen aus ein Endzustand aus F erreichbar ist. Somit werden im neuen Automaten alle Wörter akzeptiert, die sich zu einem Wort aus L verlängern lassen. Somit akzeptiert der modifizierte DFA genau die Sprache $\text{Pref}(L)$ und folglich ist $\text{Pref}(L)$ regulär.

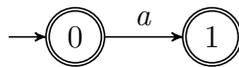
Beachten Sie, dass hier die Sprache $\text{Pref}(L)$ auch den leeren Präfix ε enthält. Wenn man nur nicht-leere Präfixe haben möchte, so muss man zusätzlich einen neuen, nicht-akzeptierenden Anfangszustand hinzufügen, der keine eingehenden Transitionen hat und zu den gleichen Zuständen ausgehende Transitionen hat wie der ursprüngliche Anfangszustand.

- (c) wahr, wenn ε -Übergänge erlaubt sind oder $\varepsilon \notin L$ gilt (sonst falsch)

Begründung

Sei M ein NFA, der die reguläre Sprache L erkennt. M hat möglicherweise mehrere Anfangs- und Endzustände. Man fügt nun einen neuen, einzelnen Anfangszustand hinzu, der ausgehende ε -Übergänge zu allen ursprünglichen Anfangszuständen hat. Außerdem fügt man einen neuen, einzelnen Endzustand hinzu, so dass aus den alten Endzuständen ε -Übergänge in den neuen Endzustand führen. Dieser Automat erkennt ebenfalls L und hat nur einen Anfangs- und Endzustand.

Ohne ε -Übergänge gilt die Behauptung im Allgemeinen nicht. Betrachte z.B. die reguläre Sprache $L = \{\varepsilon, a\}$.



Für diese Sprache gibt es keinen NFA ohne ε -Übergänge mit nur einem Endzustand.

Falls die Sprache L das leere Wort ε nicht enthält ($\varepsilon \notin L$), dann ist die Behauptung wiederum auch ohne ε -Übergänge wahr. Man modifiziert in diesem Fall den NFA so, dass ein neuer, einzelner (nicht-akzeptierender) Anfangszustand hinzugefügt wird, der keine eingehenden Transitionen hat und zu den gleichen Zuständen ausgehende Transitionen hat wie alle ursprünglichen Anfangszustände. Da der Automaten ein NFA ist, stört es nicht dass der neue Anfangszustand möglicherweise viele ausgehende Transitionen mit den gleichen Buchstaben hat. Außerdem erzeugt man einen neuen, einzelnen Endzustand, der keine ausgehenden Transitionen hat. Für jeden Übergang, der im ursprünglichen NFA in einen Endzustand geführt hat, fügen wir nun einen zusätzlichen Übergang vom gleichen Zustand hinzu, der in den neuen Endzustand führt.

(d) wahr

Begründung

Aus der Vorlesung ist bekannt, dass für reguläre Sprachen L_1, L_2 auch $L_1 \cup L_2, L_1 \cap L_2, L_1 L_2$ und $\overline{L_1}$ (Komplement) regulär sind. Außerdem gilt

$$\begin{aligned} K \setminus L &= \{w \in \Sigma^* \mid w \in K \wedge w \notin L\} \\ &= \{w \in \Sigma^* \mid w \in K \wedge w \in \overline{L}\} \\ &= K \cap \overline{L}. \end{aligned}$$

$K \setminus L$ ist äquivalent zu $K \cap \overline{L}$ und somit auf Grund der oben genannten Abschlusseigenschaften regulär.

(e) falsch

Begründung

Sei $L = \{a^n b^n \mid n \geq 0\} \cup \{a, b\}$. L ist nicht regulär. In der Vorlesung wird mit Hilfe des Pumping Lemma bewiesen, dass $\{a^n b^n \mid n \geq 0\}$ nicht regulär ist und der gleiche Beweis funktioniert auch für L .

Aus $a, b \in L$ folgt $L^* = L((a|b)^*)$, d.h. L^* enthält alle Wörter über dem Alphabet $\{a, b\}$ und diese Sprache ist regulär.

(f) wahr

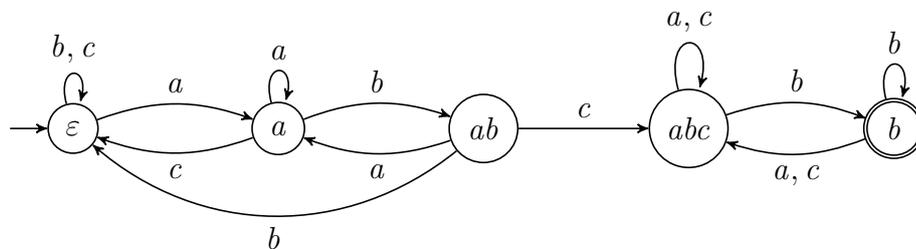
Begründung

Wir konstruieren aus einem endlichen Automaten M für L einen neuen nicht-deterministischen Automaten für $\{\text{rev}(w) \mid w \in L\}$ und zeigen so, dass $\{\text{rev}(w) \mid w \in L\}$ regulär ist. Dazu modifizieren wir M so, dass die Start- und Endzustände vertauscht werden und zusätzliche alle Übergänge umgekehrt werden. Für jedes Wort w welches von M akzeptiert wird, gilt nun dass $\text{rev}(w)$ vom modifizierten Automaten akzeptiert wird. Beachten Sie, dass selbst wenn der ursprüngliche Automat ein DFA war, so ist der resultierende Automat auf Grund der Umkehrung der Transitionen im Allgemeinen ein NFA.

Aufgabe 2. Geben Sie endliche Automaten und reguläre Ausdrücke an, die die folgenden Sprachen über dem Alphabet $\Sigma = \{a, b, c\}$ erkennen bzw. definieren.

- (a) $\{w \in \Sigma^* \mid w \text{ endet mit } b \text{ und enthält das Teilwort } abc\}$.
- (b) $\{w \in \Sigma^* \mid |w| \text{ ist gerade}\}$.
- (c) $\{w \in \Sigma^* \mid w \text{ enthält nicht das Teilwort } aba\}$.

Lösung zu Aufgabe 2. (a) $L = L((a|b|c)^* abc(a|b|c)^* b)$

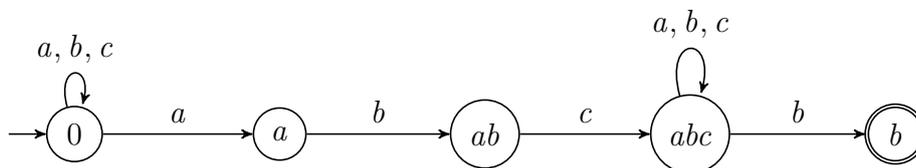


Erklärung

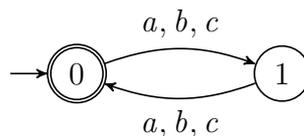
Wir verwenden die Zustände ε , a , ab , und abc , um festzuhalten, welchen maximalen Präfix von abc wir derzeit eingelesen haben (die Grundidee ist die gleiche wie bei Übungsblatt 2, Aufgabe 3(a)).

Ist der Zustand abc erreicht, wird anschließend noch unterschieden ob das Wort mit b endet.

Alternativer NFA:



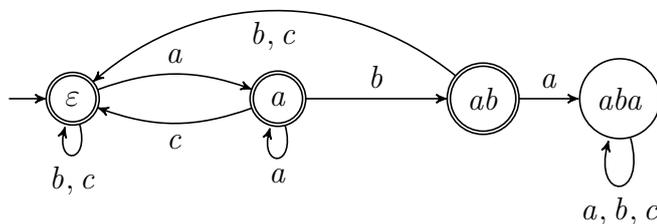
- (b) $L = L((aa|ab|ac|ba|bb|bc|ca|cb|cc)^*)$
 Alternativ: $L = L(((a|b|c)(a|b|c))^*)$



Erklärung

Im reguläre Ausdruck werden alle möglichen Paaren von Symbolen beliebig oft wiederholt um alle Wörter gerader Länge zu erhalten. Beim Automaten verwenden wir zwei Zustände 0 und 1 um die Länge modulo 2 zu zählen.

- (c) $L = (b|c|aa^*bb|aa^*bc|aa^*c)^*(aa^*b|aa^*|\varepsilon)$



Erklärung

Ähnlich wie bei (a) verwenden wir die Zustände ε , a , ab und aba um festzuhalten, welchen maximalen Präfix von aba wir derzeit eingelesen haben, bzw. ob das Wort bereits aba enthält.

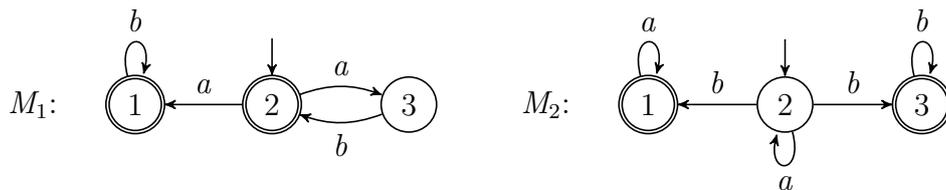
Im Gegensatz zu (a) sollen nur Wörter akzeptiert werden, die aba nicht enthalten, daher sind alle Zustände außer aba Endzustände.

Einen passenden regulären Ausdruck können wir mit dem Verfahren zur Umwandlung eines Automaten in einen regulären Ausdruck erzeugen, dies ist allerdings sehr aufwendig.

Alternativ können wir das sog. "State Elimination" Verfahren anwenden und erhalten damit genau den oben angegebenen regulären Ausdruck. Dieses wird z.B. in <https://courses.cs.washington.edu/courses/cse311/14sp/kleene.pdf> beschrieben.

Insgesamt ist es eher schwierig, einen regulären Ausdruck zu finden, wenn man ein Teilwort nicht haben möchte. Die Idee des oben stehenden Ausdrucks ist, dass im ersten Teil Wiederholungen erzeugt werden, so dass nach jedem a -Block entweder ein c kommt (aa^*c) oder, falls ein b folgt, so folgen direkt zwei b 's (aa^*bb) oder wiederum ein c (aa^*bc). Somit wird vermieden, dass aba entsteht. Der hintere Teil des Automaten regelt dann das Ende. Entweder endet das Wort ohne einen echten Präfix von aba , so kann dies schon im ersten Teil erzeugt werden und im zweiten Teil kann man das ε verwenden. Oder man endet mit einem Wort der Form aa^* oder aa^*b . Diese lassen sich im ersten Teil nicht erzeugen, da sie sonst zu aba erweitert werden könnten.

Aufgabe 3. Gegeben sind die folgenden NFAs M_1, M_2 .



Konstruieren Sie NFAs, die die folgenden Sprachen erkennen:

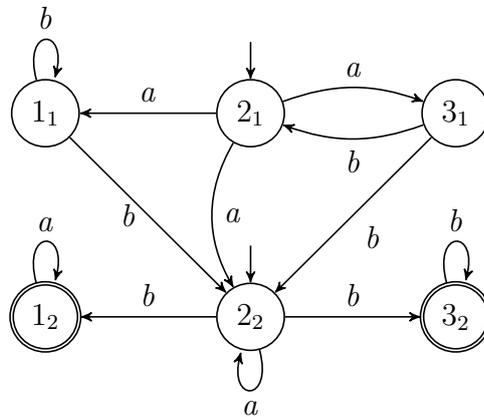
- (a) $T(M_1) \cdot T(M_2)$ (Konkatenation)
- (b) $T(M_2)^*$
- (c) $T(M_1)^+$
- (d) $T(M_1) \cap T(M_2)$ (Kreuzproduktautomat)
- (e) $T(M_1) \cup T(M_2)$

Lösung zu Aufgabe 3. (a) $T(M_1) \cdot T(M_2)$

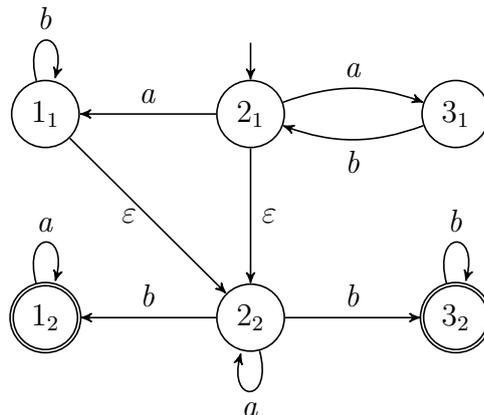
Wir verwenden das Verfahren aus der Vorlesung.

Für alle Übergänge zu Endzuständen von M_1 fügen wir Übergänge zu den Startzuständen von M_2 hinzu.

Da schon der Startzustand von M_1 ein Endzustand ist, ist auch der Startzustand von M_2 ein Startzustand im konstruierten Automaten.



Falls ε -Übergänge erlaubt wären (was wir im Allgemeinen nicht zulassen), können wir einen etwas einfacheren Automaten konstruieren:



(b) $T(M_2)^*$

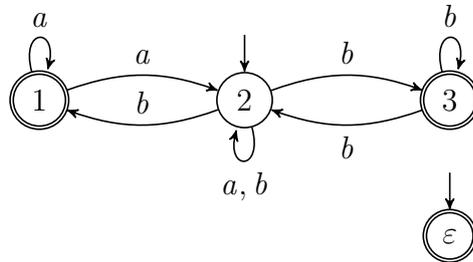
Wir verwenden das Verfahren aus der Vorlesung.

Für alle Übergänge zu Endzuständen von M_2 fügen wir Übergänge zu den Startzuständen von M_2 hinzu.

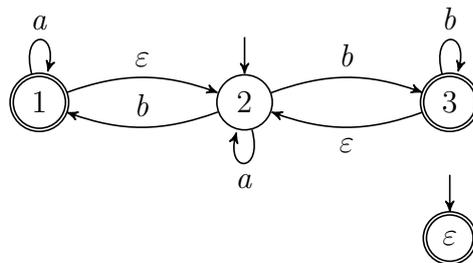
Da der Startzustand von M_2 kein Endzustand ist fügen wir einen neuen Endzustand hinzu, um auch das leere Wort zu erkennen.

Die Übergänge die wir neu hinzufügen sind:

- 2 zu 2 mit b
- 1 zu 2 mit a
- 3 zu 2 mit b



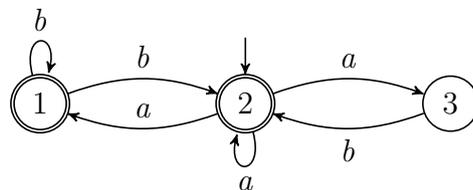
Alternativ, falls ε -Übergänge erlaubt wären, könnten wir den Automaten für $T(M_2)^*$ konstruieren, indem wir von jedem Endzustand einen ε -Übergang in den Startzustand hinzufügen:



- (c) $T(M_1)^+ = T(M_1) \cdot T(M_1)^* = \{w_1 \dots w_n \mid n \geq 1, w_i \in T(M_1) \text{ für alle } i\}$
 Da $\varepsilon \in T(M_1)$ gilt $T(M_1)^+ = T(M_1) \cdot T(M_1)^* = T(M_1)^*$ und somit können wir das gleiche Verfahren anwenden wie bei (b).

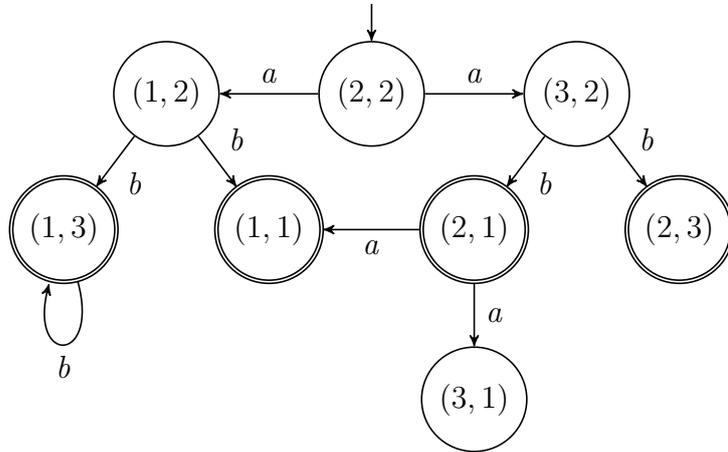
Die neuen Übergänge sind:

- 2 zu 2 mit a
- 1 zu 2 mit b



- (d) $T(M_1) \cap T(M_2)$

Es folgt der Kreuzproduktautomat, wie in der Vorlesung beschrieben:



Mit Blick auf beiden Automaten M_1 und M_2 und die entsprechenden regulären Ausdrücke $(ab)^* \mid (ab)^*ab^*$ und $a^*ba^* \mid a^*bb^*$ (siehe dazu Übungsblatt 2) sehen wir, dass nur Wörter der Form aba und abb^* von beiden Ausdrücken erkannt werden.

Wir können leicht verifizieren, dass dies genau die Wörter sind, die der konstruierte Automat akzeptiert.

(e) $T(M_1) \cup T(M_2)$

Hier können wir einfach beide Automaten kombinieren ohne neue Übergänge hinzuzufügen, da NFAs mehrere Startzustände haben können.

