

## Übungsblatt 5

### 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  $\{w^r \mid w \in L\}$  regulär, wobei wenn  $w = a_1 \dots a_n$  gilt, dann ist  $w^r = a_n \dots a_1$  (siehe Blatt 2, Aufgabe 1).
- (g) Für jede Sprache  $L$  sind  $L \cap \bar{L}$  und  $L \cup \bar{L}$  regulär.

### Lösung zu Aufgabe 1.

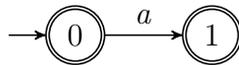
- (a) Falsch, sei z.B.  $L = L(a|b)$ , dann ist  $a \in L, b \in L$  aber  $ab \notin L$ .
- (b) Wahr. Sei  $M$  ein DFA für die reguläre Sprache  $L$  und  $F$  die Endzustandsmenge von  $M$ . Wir konstruieren aus  $M$  einen DFA für die Sprache  $\text{Pref}(L)$  indem wir die Endzustandsmenge  $F$  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  $\varepsilon$  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  $\varepsilon$ -Übergänge erlaubt sind oder  $\varepsilon \notin L$  gilt (sonst falsch).

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  $\varepsilon$ -Übergänge zu allen ursprünglichen Anfangszuständen hat. Außerdem fügt man einen neuen, einzelnen Endzustand hinzu, so dass aus den alten Endzuständen  $\varepsilon$ -Übergänge in den neuen Endzustand führen. Dieser Automat erkennt ebenfalls  $L$  und hat nur einen Anfangs- und Endzustand.

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



Für diese Sprache gibt es keinen NFA ohne  $\varepsilon$ -Übergänge mit nur einem Endzustand.

Falls die Sprache  $L$  das leere Wort  $\varepsilon$  nicht enthält ( $\varepsilon \notin L$ ), dann ist die Behauptung wiederum auch ohne  $\varepsilon$ -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, kann der neue Anfangszustand möglicherweise viele ausgehende Transitionen mit gleichen Buchstaben haben. 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.

Bemerkung: Mittels Potenzmengenkonstruktion bekommen wir aus jedem NFA einen DFA, der nur einen Anfangszustand hat. Da jeder DFA auch ein NFA ist, ist die erste Bedingung (genau ein Startzustand) also immer erfüllbar.

- (d) Wahr. Aus der Vorlesung ist bekannt, dass für reguläre Sprachen  $L_1, L_2$  auch  $L_1 \cap L_2$  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.

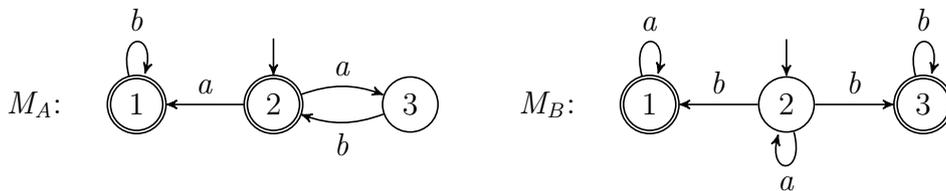
- (e) Falsch. Sei  $L = \{a^n b^n \mid n \geq 0\} \cup \{a, b\}$ .  $L$  ist nicht regulär. In der Vorlesung wird später mit Hilfe des Pumping Lemmas bewiesen, dass  $\{a^n b^n \mid n \geq 0\}$  nicht regulär ist und der gleiche Beweis funktioniert auch für  $L$ .

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. Wir konstruieren aus einem endlichen Automaten  $M$  für  $L$  einen neuen nicht-deterministischen Automaten für  $\{w^r \mid w \in L\}$  und zeigen so, dass  $\{w^r \mid w \in L\}$  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  $w^r$  vom modifizierten Automaten akzeptiert wird. Man beachte, dass selbst wenn der ursprüngliche Automat ein DFA war, der resultierende Automat auf Grund der Umkehrung der Transitionen im Allgemeinen ein NFA ist.

- (g) Wahr, denn es gilt stets  $L \cap \overline{L} = \emptyset$  und  $L \cup \overline{L} = \Sigma^*$ .

**Aufgabe 2.** Gegeben sind die folgenden NFAs  $M_A, M_B$ .



- (a) Geben Sie die Sprachen  $T(M_A)$  und  $T(M_B)$  an.  
 (b) Geben Sie DFAs für  $T(M_A)$  und  $T(M_B)$  an.  
 (c) Konstruieren Sie NFAs, die die folgenden Sprachen erkennen:

- (i)  $T(M_A) \cdot T(M_B)$  (Konkatenation)
- (ii)  $T(M_B)^*$
- (iii)  $T(M_A)^+$
- (iv)  $T(M_A) \cap T(M_B)$  (Kreuzproduktautomat)
- (v)  $T(M_A) \cup T(M_B)$

**Lösung zu Aufgabe 2.** (a)

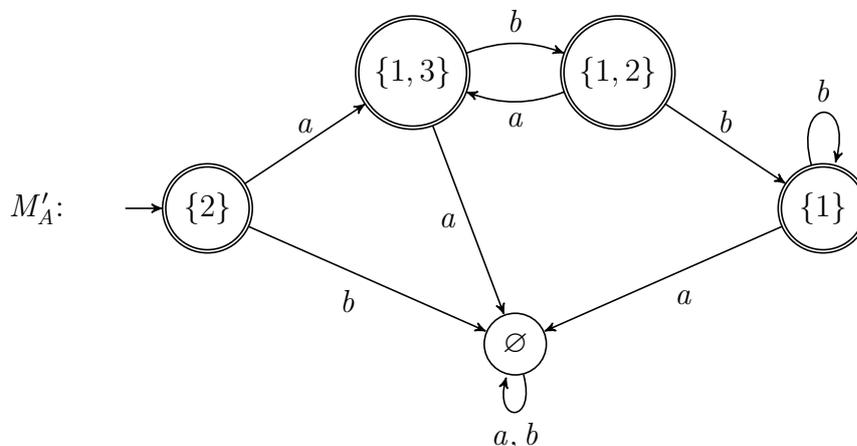
$$T(M_A) = \{(ab)^n \mid n \in \mathbb{N}\} \cup \{(ab)^n ab^m \mid n, m \in \mathbb{N}\}$$

Vom Startzustand 2 erreicht man den Endzustand 2 ausschließlich in dem man (beliebig oft) mit  $a$  nach 3 geht und anschließend mit  $b$  wieder zurück nach 2 geht. Daher werden Wörter der Form  $(ab)^n$  für beliebige  $n \in \mathbb{N}$  akzeptiert. Zusätzlich kann man aber anschließend (oder sofort) vom Zustand 2 auch noch mit  $a$  in den Endzustand 1 wechseln und dort beliebig viele  $b$ 's lesen. So lassen sich Wörter der Form  $(ab)^n ab^m$  für  $n, m \in \mathbb{N}$  erzeugen.

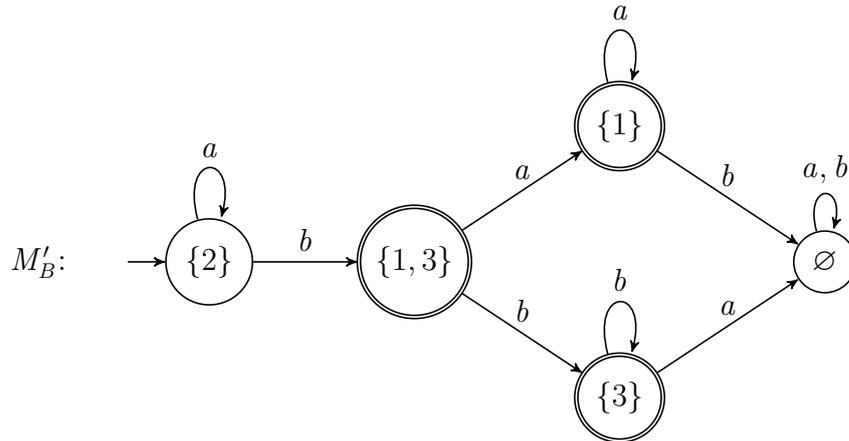
$$T(M_B) = \{a^n ba^m \mid n, m \in \mathbb{N}\} \cup \{a^n bb^m \mid n, m \in \mathbb{N}\}$$

Vom Startzustand 2 erreicht man den Endzustand 1 in dem man beliebig oft mit  $a$  im Zustand 2 bleibt, anschließend mit  $b$  in den Zustand 1 wechselt und dort beliebig viele  $a$ 's liest. Somit werden Wörter der Form  $a^n ba^m$  für  $n, m \in \mathbb{N}$  akzeptiert. Alternativ erreicht man vom Startzustand 2 den Endzustand 3 in dem man beliebig oft mit  $a$  im Zustand 2 bleibt, anschließend mit  $b$  in den Zustand 3 wechselt und dort beliebig viele  $b$ 's liest. Somit werden Wörter der Form  $a^n bb^m$  für  $n, m \in \mathbb{N}$  akzeptiert.

- (b) Mittels Potenzmengenkonstruktion erhalten wir DFA's aus NFA's. DFA für  $T(M_A)$ :



DFA für  $T(M_B)$ :



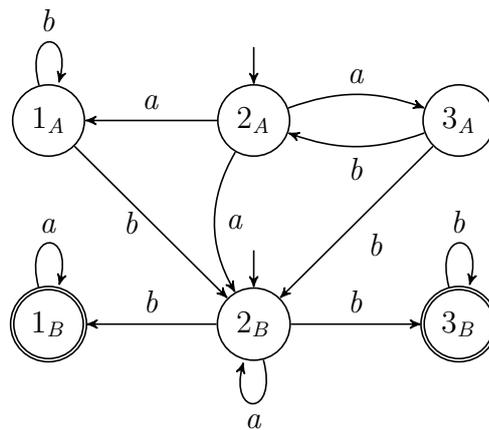
(c)

(i)  $T(M_A) \cdot T(M_B)$

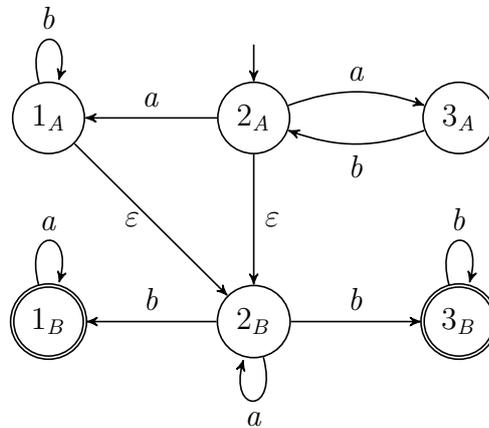
Wir verwenden das Verfahren aus der Vorlesung.

Für alle Übergänge zu Endzuständen von  $M_A$  fügen wir Übergänge zu den Startzuständen von  $M_B$  hinzu.

Da schon der Startzustand von  $M_A$  ein Endzustand ist, ist auch der Startzustand von  $M_B$  ein Startzustand im konstruierten Automaten.



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



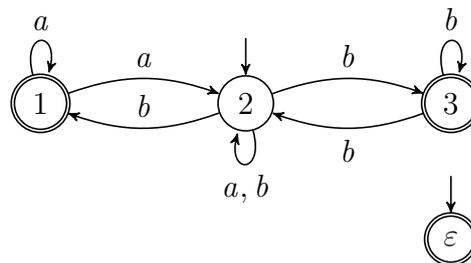
(ii)  $T(M_B)^*$ : Wir verwenden das Verfahren aus der Vorlesung.

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

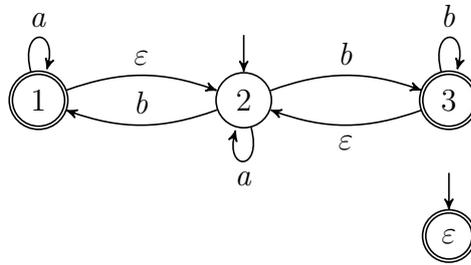
Da der Startzustand von  $M_B$  kein Endzustand ist fügen wir einen neuen Endzustand hinzu, um auch das leere Wort zu erkennen.

Die Übergänge die wir neu hinzufügen sind:

- $1 \xrightarrow{a} 2$
- $2 \xrightarrow{b} 2$
- $3 \xrightarrow{b} 2$



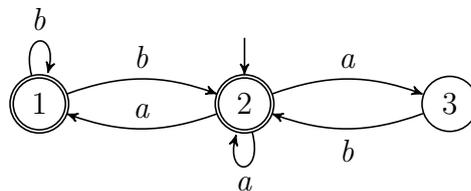
Alternativ, falls  $\varepsilon$ -Übergänge erlaubt wären, könnten wir den Automaten für  $T(M_B)^*$  konstruieren, indem wir von jedem Endzustand einen  $\varepsilon$ -Übergang in den Startzustand hinzufügen:



- (iii)  $T(M_A)^+ = T(M_A) \cdot T(M_A)^* = \{w_1 \dots w_n \mid n \geq 1, w_i \in T(M_A) \text{ f\"ur alle } i\}$   
 Da  $\varepsilon \in T(M_A)$  gilt  $T(M_A)^+ = T(M_A) \cdot T(M_A)^* = T(M_A)^*$  und somit  
 k\"onnen wir das gleiche Verfahren anwenden wie bei (ii).

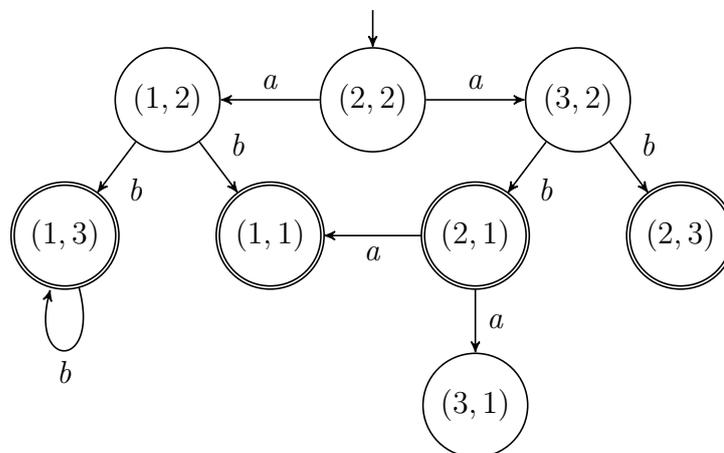
Die neuen \"Uberg\"ange sind:

- $1 \xrightarrow{b} 2$
- $2 \xrightarrow{a} 2$



- (iv)  $T(M_A) \cap T(M_B)$

Es folgt der Kreuzproduktautomat, wie in der Vorlesung beschrieben:



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

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

(v)  $T(M_A) \cup T(M_B)$

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

