Universität Siegen
Lehrstuhl Theoretische Informatik
Markus Lohrey

Formale Sprachen und Automaten
SS 2025

Übungsblatt 4

Aufgabe 1. Sind die folgenden Aussagen wahr oder falsch?

(a) $L((a \mid bb)^*) = L(a^* \mid (bb)^*)$.

(b) $L(a \mid b \mid (a^* \mid b^*)^*) = \{a, b\}^*$.

(c) Jeder reguläre Ausdruck ohne den $*$-Operator erzeugt eine endliche Sprache.

Lösung zu Aufgabe 1.

(a) Falsch: z.B. $abb \in L((a \mid bb)^*)$ aber $abb \notin L(a^* \mid (bb)^*)$.

(b) Wahr. Formal müsste man beide Mengeninklusionen zeigen.

(c) Wahr (der $+$-Operator ist hier als Erweiterung von $*$ auch ausgeschlossen). Einen Beweis können wir induktiv über den Aufbau möglicher regulärer Ausdrücke führen.

$L(\varnothing)$, $L(\varepsilon)$ und $L(\alpha)$ für $\alpha \in \Sigma$ sind endlich.

Als Operatoren bleiben noch Vereinigung und Konkatenation. Falls $L(\alpha)$ und $L(\beta)$ endlich sind (Induktionsvoraussetzung), dann sind auch $L(\alpha\beta) = L(\alpha)L(\beta)$ und $L(\alpha|\beta) = L(\alpha) \cup L(\beta)$ endlich.

Aufgabe 2. Zur Erinnerung (Folie 16): Ein *Monoid* ist ein Paar $(M, \times)$, wobei M eine Menge und $\times : M \times M \to M$ eine assoziative Abbildung ist und es ein neutrales Element in $e \in M$ gibt, d.h. $a \times e = a = e \times a$ für alle $a \in M$.

(a) Zeigen Sie: $(L((ab)^*), \circ)$ ist ein Monoid (Erinnerung: $\circ$ ist die Konkatenation von Wörtern).

(b) Ist $(L, \circ)$ für jede reguläre Sprache L ein Monoid?

Ein *(Monoid-)Homomorphismus* zwischen zwei Monoiden $(M, \circ)$ und $(N, \cdot)$ ist eine Abbildung $h : M \to N$ mit den Eigenschaften $h(u \circ v) = h(u) \cdot h(v)$ und $h(e_M) = e_N$ für alle $u, v \in M$ und die neutralen Elemente $e_M \in M$ und $e_N \in N$. Ein Homomorphismus heißt *Isomorphismus*, falls er *bijektiv* ist, d.h. es gibt eine Abbildung $h^{-1} : N \to M$ mit $h(h^{-1}(n)) = n$ und $h^{-1}(h(m)) = m$ für alle $m \in M$ und $n \in N$.

(c) Zeigen Sie, dass es einen Monoid-Isomorphismus zwischen $(L((ab)^*), \circ)$ und $(\mathbb{N}, +)$ gibt.

(d) Zeigen Sie, dass es *keinen* Monoid-Isomorphismus zwischen $(\{\varepsilon\}, \circ)$ und $(\mathbb{N}, +)$ geben kann.

Lösung zu Aufgabe 2.

(a) Wir wissen, dass die Konkatenation von Wörtern assoziativ ist. Wir müssen also nur noch zeigen, dass $L((ab)^*)$ unter dieser Operation abgeschlossen ist und das ist ein neutrales Element gibt. Der reguläre Ausdruck $(ab)^*$ erzeugt die Sprache $\{(ab)^n \mid n \geq 0\}$, also ist $\varepsilon \in L((ab)^*)$ $(n = 0)$ und wir wissen bereits, dass das leere Wort ein neutrales Element bezüglich der Konkatenation ist.

Seien nun $u, v \in L((ab)^*)$, also gibt es $n, m \in \mathbb{N}$ mit $u = (ab)^m$ und $v = (ab)^n$. Dann gilt $u \circ v = (ab)^m \circ (ab)^n = (ab)^{m+n} \in L((ab)^*)$. Somit ist $(L((ab)^*), \circ)$ ein Monoid.

(b) Nein. Falls $\varepsilon \notin L$ ist, ist bereits eine Bedingung nicht erfüllt. Man könnte aber auch eine reguläre Sprache nehmen, die nicht unter $\circ$ abgeschlossen ist (zum Beispiel jede endliche Sprache ungleich $\{\varepsilon\}$).

(c) Wähle $h : L((ab)^*) \to \mathbb{N}$ mit $h((ab)^n) = n$. Diese Abbildung ist ein Homomorphismus, denn $h((ab)^m \circ (ab)^n) = h((ab)^{m+n}) = m + n = h((ab)^m) + h((ab)^n)$ und $h(\varepsilon) = h((ab)^0) = 0$.

Außerdem ist diese Abbildung bijektiv, denn mit der Umkehrabbildung $h^{-1} : \mathbb{N} \to L((ab)^*), h^{-1}(n) = (ab)^n$ gilt: $h(h^{-1}(n)) = h((ab)^n) = n$ und $h^{-1}(h((ab)^n)) = h^{-1}(n) = (ab)^n$ für alle $n \in \mathbb{N}$.

(d) Sei $h : \{\varepsilon\} \to \mathbb{N}$ ein Homomorphismus, also ist $h(\varepsilon) = 0$. Für eine Umkehrabbildung $h^{-1} : \mathbb{N} \to \{\varepsilon\}$ gilt dann: $h(h^{-1}(n)) = h(\varepsilon) = 0$, welche nur für $n = 0$ die Bijektivitätsbedingung erfüllt. Man sagt auch, dass h nicht *surjektiv* bzw. h^{-1} nicht *injektiv* ist. Als Mindestvoraussetzung für einen Isomorphismus müssen beide Mengen zumindest die gleiche Anzahl Elemente haben.

Aufgabe 3. Geben Sie zu jeder der folgenden Sprachen einen regulären Ausdruck an.

(a) $L_1 = \{w \in \{a, b\}^* \mid \text{Das Wort } w \text{ enthält mindestens ein } b.\}$

(b) $L_2 = \{w \in \{a, b\}^+ \mid \text{Der erste und letzte Buchstabe in } w \text{ stimmen überein.}\}$

(c) $L_3 = \left\{a^n b^m c^\ell \mid n \geq 0,\ m \geq 1,\ \ell \geq 2\right\}$

Lösung zu Aufgabe 3.

(a) $L_1 = L(a^* b(a|b)^*)$ **Erklärung:** Ein Wort w, dass mindestens ein b enthält, lässt sich aufteilen in $w = ubv$, wobei

  (i) $u \in \{a^k \mid k \geq 0\}$ der Teil des Worts w vor dem ersten b (der also nur aus a's besteht) ist, und

  (ii) $v \in \{a, b\}^*$ der Teil des Wortes w ist, der auf das erste b folgt (bestehend aus beliebig vielen a's und b's).

  Ein alternativer regulärer Ausdruck ist $(a|b)^* b(a|b)^*$.

(b) $L_2 = L(a \mid b \mid a(a|b)^* a \mid b(a|b)^* b)$

  Erklärung: Hier unterscheiden wir drei Fälle:

  (i) $|w| = 1$,

  (ii) $|w| \geq 2$ und w beginnt und endet mit a,

  (iii) $|w| \geq 2$ und w beginnt und endet mit b.

  Fall (i) erfassen wir mit dem regulären Ausdruck $a|b$.

  In Fall (ii) lässt sich w also schreiben als $w = axa$ mit $x \in \{a, b\}^*$, diesen Fall erfassen wir mit dem regulären Ausdruck $a(a|b)^* a$.

  In Fall (iii) lässt sich w also schreiben als $w = bxb$ mit $x \in \{a, b\}^*$, diesen Fall erfassen wir mit dem regulären Ausdruck $b(a|b)^* b$.

(c) $L_3 = L(a^* bb^* ccc^*)$ **Erklärung:**

  Die Sprache $\{a^n \mid n \geq 0\}$ lässt sich als regulärer Ausdruck a^* schreiben.

  Die Sprache $\{b^m \mid n \geq 1\}$ lässt sich durch den regulären Ausdruck bb^* darstellen. Alternativ kann man b^+ verwenden, da $L(bb^*) = L(b^+)$.

  Die Sprache $\{c^k \mid k \geq 2\}$ lässt sich durch den regulären Ausdruck ccc^* (oder alternativ: cc^+) darstellen.

  Die Sprache L_3 ist nun die Konkatenation der Sprachen $\{a^n \mid n \geq 0\}$, $\{b^m \mid n \geq 1\}$ und $\{c^k \mid k \geq 2\}$.

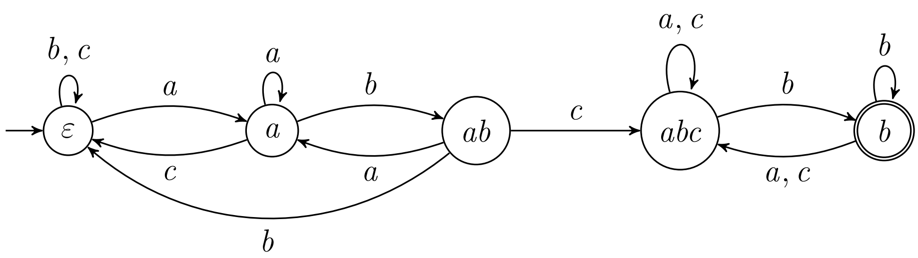
Aufgabe 4. Geben Sie endliche Automaten und reguläre Ausdrücke an, die die folgenden Sprachen über dem Alphabet $\Sigma = \{a, b, c\}$ erkennen bzw. definieren.

(a) $\{w \in \Sigma^* \mid w$ endet mit b und enthält das Teilwort $abc\}$.

(b) $\{w \in \Sigma^* \mid |w|$ ist gerade$\}$.

(c) $\{w \in \Sigma^* \mid w$ enthält nicht das Teilwort $aba\}$.

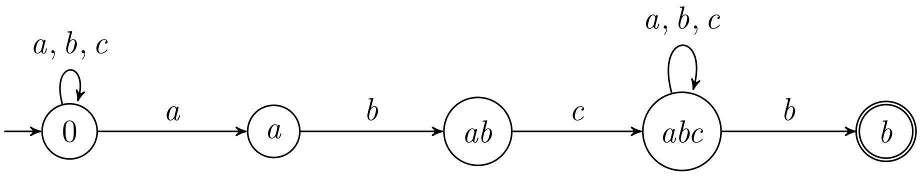
Lösung zu Aufgabe 4. (a) $L = L((a|b|c)^* abc(a|b|c)^* b)$



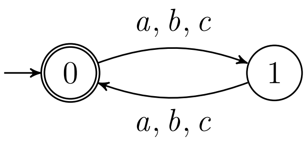
Erklärung

Wir verwenden die Zustände ε, a, ab, und abc, um festzuhalten, welchen maximalen Präfix von abc wir derzeit eingelesen haben. Ist der Zustand abc erreicht, wird anschließend noch unterschieden ob das Wort mit b endet.

Alternativer NFA:



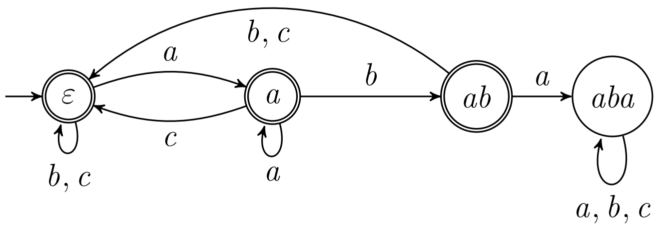
(b) $L = L((aa|ab|ac|ba|bb|bc|ca|cb|cc)^*)$
Alternativ: $L = L(((a|b|c)(a|b|c))^*)$



Erklärung

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

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



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 regulären Ausdrucks beschreibt, wie die letzten Zeichen eines Wortes aus L aussehen müssen. 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 Teilwörter lassen sich im ersten Teil nicht erzeugen, da sie sonst zu aba erweitert werden könnten. Man beachte, dass diese 3 Fälle genau den akzeptierenden Zuständen des Automaten entsprechen.