

Disclaimer

Die Übungen die hier gezeigt werden stammen aus der Vorlesung *Algorithmen, Sprachen und Komplexität!* Für die Richtigkeit der Lösungen wird keine Gewähr gegeben.

Übung 00

Aufgabe 1

Das Schubfachprinzip besagt: Wenn n Objekte auf m Schubladen verteilt werden mit $n > m > 0$, dann gibt es eine Schublade, die mindestens zwei Objekte enthält. Zeigen Sie, dass es mindestens zwei Personen in Deutschland mit gleich vielen Haaren gibt.

(b) Beweisen Sie das verschärfte Schubfachprinzip: Verteilt man n Objekte auf m Schubladen mit $n > m > 0$, dann gibt es eine Schublade, die mindestens $\lceil \frac{n}{m} \rceil$ Objekte enthält

Aufgabe 2

Zeigen Sie per vollständiger Induktion über $n \geq 0$, dass es in jedem Binärbaum mit mindestens 2^n Blättern einen Pfad der Länge mindestens n von der Wurzel zu einem Blatt gibt.

Aufgabe 3

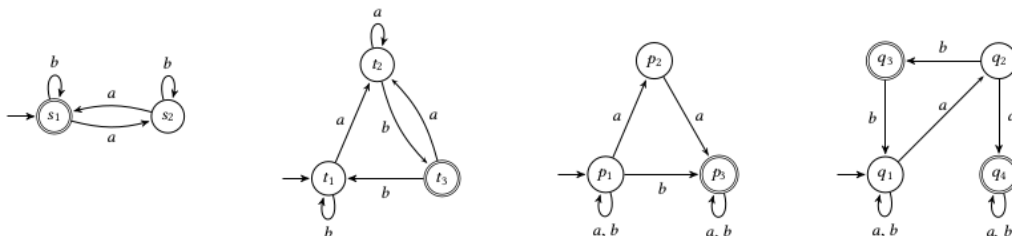
Eine Menge A heißt gleichmächtig zu einer Menge B , wenn es eine Bijektion von A nach B gibt. Zeigen Sie:

- (a) Die Menge der natürlichen Zahlen \mathbb{N} ist nicht gleichmächtig zur Menge der reellen Zahlen \mathbb{R} .
- (b) Keine Menge ist gleichmächtig zu ihrer Potenzmenge. (Satz von Cantor)

Übung 01

Aufgabe 1

Es sind die DFAs M_1 und M_2 und die NFAs M_3 und M_4 (von links nach rechts) gegeben.



Bearbeiten Sie die folgenden Teilaufgaben für alle $i \in \{1, 2, 3, 4\}$.

(a) Geben Sie jeweils zwei Wörter an, die von M_i akzeptiert bzw. nicht akzeptiert werden.

(b) Geben Sie analog zu Aufgabe 2 eine kurze aber präzise Beschreibung der Sprache $L(M_i)$ an.

Aufgabe 2

Betrachten Sie die nachfolgenden Sprachen über dem Alphabet $\Sigma = \{a, b\}$.

- $L_1 = \{w \in \Sigma^* \mid w \text{ enthält die Zeichenfolge } baba\}$
- $L_2 = \Sigma^* \setminus \{aa, ab, aab\}$
- $L_3 = \{w \in \Sigma^* \mid \text{es existiert } k \geq 1, \text{ so dass } w \text{ mit } a(ab)^k \text{ beginnt}\}$
- $L_4 = \{w \in \Sigma^* \mid w \text{ endet auf } aab\}$

Geben Sie für alle $i \in \{1, 2, 3, 4\}$ jeweils einen DFA M_i mit $L(M_i) = L_i$ grafisch an. Wählen Sie jeweils zwei Wörter aus L_i und $\Sigma^* \setminus L_i$ aus und überprüfen Sie, ob M_i auf diesen korrekt arbeitet.

Aufgabe 3

Konstruieren Sie mit der Potenzmengenkonstruktion einen DFA, der die gleiche Sprache akzeptiert, wie M_3 aus Aufgabe 1.

Aufgabe 4

Sei $\Sigma = \{a, b, c\}$. Unter den folgenden 16 Sprachen über Σ befinden sich acht Paare gleicher Sprachen. Finden Sie heraus, welche Sprachen gleich sind und begründen Sie jeweils in maximal zwei Sätzen, warum die entsprechende Gleichheit gilt.

$$L_1 = \{w \in \Sigma^* \mid |w|_a = |w|_b = |w|_c\}$$

$$L_2 = \{w \in \Sigma^* \mid |w|_a = |w|_b\}$$

$$L_3 = \{w \in \Sigma^* \mid |w|_a = 0\}$$

$$\begin{aligned}
L_4 &= \{w \in \sum^* \mid |w|_a = 2\} & L_{11} &= (L_3 L_4)^2 \\
L_5 &= \{w \in \sum^* \mid |w|_a = 4\} & L_{12} &= \sum^* \setminus L_3 \\
L_6 &= \{b, c\}^* \{a\} \{b, c\}^* \{a\} \{b, c\}^* & L_{13} &= L_2^3 \\
L_7 &= \{a\} \{ba\}^* \{b\} & L_{14} &= \{ab\}^+ \\
L_8 &= \{a^n b^n \mid n \in \mathbb{N}\} & L_{15} &= \{b, c\}^* \\
L_9 &= L_2 \cap \{a\}^* \{b\}^* & L_{16} &= \sum^* \{a\} \sum^* \\
L_{10} &= L_2 \cap \{w \in \sum^* \mid |w|_b = |w|_c\}
\end{aligned}$$

Aufgabe 5

Sei $\sum = \{a, b\}$. Für $n \in \mathbb{N}$ sei $\sum^{\leq n} = \bigcup_{i \leq n} \sum^i$ die Menge der Wörter in \sum deren Länge höchstens n ist. Zeigen Sie per vollständiger Induktion über $n \in \mathbb{N}$, dass $|\sum^{\leq n}| = 2^{n+1} - 1$.

Aufgabe 6

Gegeben sei die Grammatik $G = (\{S, A, B, C\}, \{a, b, c\}, P, S)$, wobei P genau die folgenden Produktionen enthält:

$$\begin{array}{lll}
S \rightarrow A|C & A \rightarrow Aa|a & Bc \rightarrow bbc \\
& Bb \rightarrow bb & C \rightarrow BCc|c.
\end{array}$$

- (a) Geben Sie eine Ableitung von $bbcccc$ an.
(b) Geben Sie eine möglichst kurze aber präzise Beschreibung von $L(G)$ an. Begründen Sie Ihre Antwort.

Aufgabe 7

Konstruieren Sie Grammatiken G_1, G_2 und G_3 so, dass folgende Sprachen erzeugt werden.

- (a) $L(G_1) = \sum^* \{a\} \sum^* \cup \sum^* \{b\} \sum^*$ für $\sum = \{a, b, c\}$
(b) $L(G_2) = \{ww^R \mid w \in \{a, b\}^* : w \text{ startet mit einem } b\}$ Hinweis: Für $w = w_1 w_2 \dots w_{n-1} w_n$ sei $w^R = w_n w_{n-1} \dots w_2 w_1$ das umgekehrte Wort.
(c) $L(G_3)$ ist die Menge der Polynomgleichungen über den Variablen x, y . Hinweis: Ein Polynom über den Variablen x, y ist induktiv wie folgt definiert: $0, 1, x, y$ sind Polynome und falls f, g Polynome sind, so auch $(f + g)$ und $(f * g)$.

Übung 02

Aufgabe 1

Geben Sie zu den Sprachen L_a, L_b reguläre Ausdrücke α, β so an, dass $L(\alpha) = L_a$ und $L(\beta) = L_b$.

- (a) $L_a = \{w \in \{a, b, c\}^* \mid \text{entweder kommen } a \text{ und } b \text{ in } w \text{ vor oder weder } a \text{ noch } b\}$
- (b) $L_b = \{w \in \{a, b, c\}^* \mid w \text{ enthält nicht das Infix } bc\}$

Aufgabe 2

Zeigen Sie, dass die Klasse der regulären Sprachen nicht unter unendlicher Vereinigung abgeschlossen ist.

Aufgabe 3

Sei $L \subseteq \Sigma^*$ eine Sprache. Unter der Verdopplung von L verstehen wir die Sprache $2 * L := \{ww \mid w \in L\}$. Überprüfen Sie, ob die Klasse der regulären Sprachen unter Verdopplung abgeschlossen ist. Beweisen Sie Ihre Behauptung!

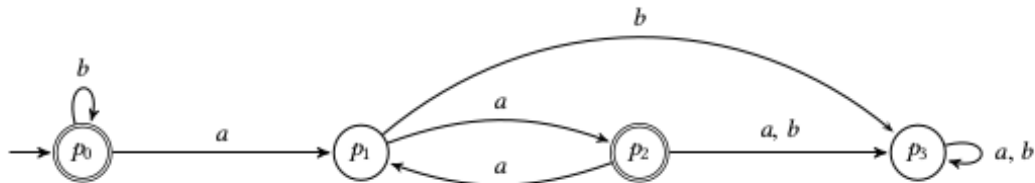
Aufgabe 4

Sei Σ ein Alphabet (eine endliche Menge). Zeigen Sie, dass Σ^* abzählbar ist.

Aufgabe 5

Bearbeiten Sie folgende Teilaufgaben:

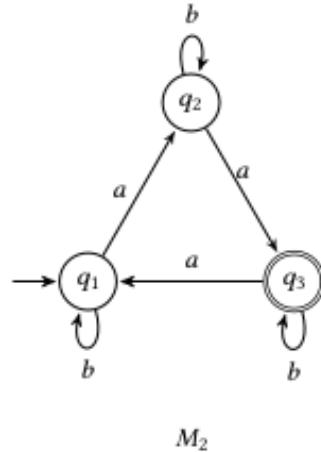
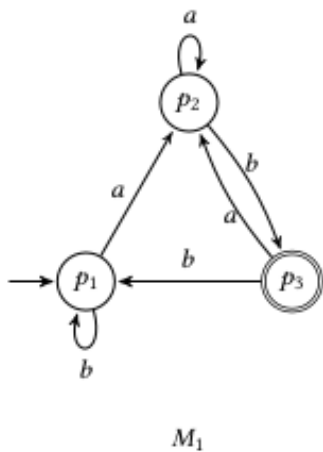
- (a) Beschreiben Sie die Sprache des folgenden Automaten kurz und präzise



- (b) Sei $\Sigma = \{a, b, c\}$. Geben Sie einen DFA an, der die Sprache $L = \{w \in \Sigma^* \mid |w|_a \leq 1 \text{ und } |w|_b = 0\}$ akzeptiert. Dabei steht für $x \in \Sigma, w \in \Sigma^*$ der Ausdruck $|w|_x$ für die Anzahl der x in w .

Aufgabe 6

Gegeben seien die folgenden DFAs M_1 und M_2 .



Konstruieren Sie folgende Automaten:

- (a) einen DFA M_\cap mit $L(M_\cap) = L(M_1) \cap L(M_2)$,
- (b) einen NFA M_\cdot mit $L(M_\cdot) = L(M_1) * L(M_2)$ und
- (c) einen NFA M_* mit $L(M_*) = L(M_1)^*$.

Aufgabe 7

Zeigen Sie die folgenden Aussagen:

(a) Für jeden NFA $M = (Z, \Sigma, S, \delta, E)$ existiert ein NFA $M_0 = (Z_0, \Sigma, S_0, \delta_0, E_0)$ mit $L(M) = L(M_0)$ und $|E_0| = 1$.

(b) Für jeden NFA $M = (Z, \Sigma, S, \delta, E)$ existiert ein NFA $M_0 = (Z_0, \Sigma, S_0, \delta_0, E_0)$ mit $L(M) = L(M_0)$, $|S_0| = 1$ und $|Z_0| = |Z| + 1$.

Aufgabe 8

Die Spiegelung eines Wortes $w = a_1 a_2 \dots a_n \in \Sigma^*$ sei $w^R := a_n a_{n-1} \dots a_1$ für $a_i \in \Sigma$ für alle $1 \leq i \leq n$. Die Spiegelung einer Sprache L sei $L^R := \{w^R \mid w \in L\}$. Zeigen Sie, dass die Klasse der regulären Sprachen unter Spiegelung abgeschlossen ist.

Übung 03

Aufgabe 1

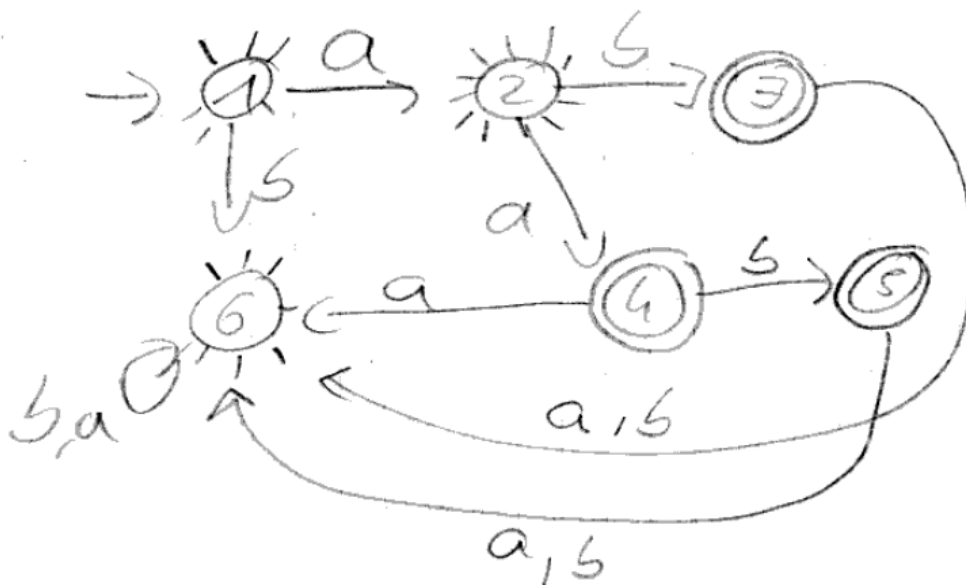
Betrachten Sie die nachfolgenden Sprachen über dem Alphabet $\Sigma = \{a, b\}$.

- $L_1 = \{w \in \Sigma^* \mid \text{der vorletzte Buchstabe von } w \text{ ist ein } a\}$
- $L_2 = \Sigma^* \setminus \{aa, ab, aab\}$
- $L_3 = \{w \in \Sigma^* \mid \text{in } w \text{ kommt ein Buchstabe zweimal direkt hintereinander vor}\}$
- $L_4 = \Sigma^* \setminus L_3$

Konstruieren Sie für alle $i \in \{1, 2, 3, 4\}$ jeweils einen regulären Ausdruck r_i mit $L(r_i) = L_i$.

$$L_1 = (a + b)^* a (a + b)$$

$$L_2 = (b(a + b)^*) + ab(a + b)(a + b)^* + aaa(a + b)^* + aab(a + b)(a + b)^* + a + \lambda$$



Idee: Zuerst den Automaten der die aa,ab,aab Sprache akzeptiert aufzeichnen. Dann alle Endzustände (die doppelt umkreisten) zu normalen Zuständen machen und dann die früheren Nicht-Endzustände zu Endzuständen machen (symbolisiert durch die sonnenähnlichen Gebilde um Zustand 1,2,6)

$$L_3 = ((a + b)^* (aa + bb) (a + b)^* (aa + bb) (a + b)^*)$$

$$L_4 = (ab)^* + (ba)^* + a + b + \lambda$$

Aufgabe 2

Zeigen Sie direkt mit dem Pumping-Lemma, dass die Sprache $L = \{a^i b^j \mid i, j \in \mathbb{N}, i > j\}$ nicht regulär ist.

Behauptung: Die Sprache L ist nicht regulär.

0. Beweis: indirekt. Angenommen L wäre regulär. Nach dem Pumping-Lemma gibt es ein $n \geq 1$, sodass die folgende Aussage gilt:

Für jedes $x \in L$, $|x| \geq n$ gibt es $u, v, w \in \Sigma^*$ mit

- i $x = uvw$
- ii $|uv| \leq n$
- iii $|v| \geq 1$
- iv $uv^i w \in L \forall i \geq 0$

1. Wir wählen ein Wort $x \in L$. Sei $x = a^n b^j$, wobei n nach Definition der Sprache echt größer j ist.
2. Nach der Aussage (*) gibt es $u, v, w \in \Sigma^*$, welche die Eigenschaften (i)-(iv) erfüllen.
3. Sei $x = a^n b^j$ mit $n > j$, wir definieren $j = n - 1$. Wir wählen $|uv| \leq n$ mit $|v| \geq k$. Es gilt $v \in \{a\}^+$. Nun sei $i = 0$, damit ist $x = a^{n-k} b^j$, da $|v| \geq 1$ ist, und da nun $j = n - k$ gilt, ist $n = j$, was allerdings der Bedingung $n > j$ widerspricht.
Wählen wir $|v| = k$ mit $k \in \mathbb{N}$. so gilt: $uv = a^{(n-k)} b^j$
4. Dieser Widerspruch von $n = j \neq n > j$ ist ein Widerspruch zu Aussage (iv) des Pumping Lemmas.
Somit ist die Aussage bewiesen, dass die Sprache L nicht regulär sein kann. q.e.d

Aufgabe 3

Zeigen Sie mit dem Spielschema des Pumping-Lemmas, dass die Sprache $L = \{a^{2^n} \mid n \in \mathbb{N}\}$ nicht regulär ist.

1. Runde: G wählt eine Zahl $n \geq 1$
2. Runde: B wählt $x \in L$ mit $|x| \geq n$. Sei $x = a^{(2^n)}$.
3. Runde: G wählt u, v, w mit i) $x = uvw$ ii) $|uv| \leq n$ iii) $|v| \geq 1$
4. Runde: B wählt $i = 2$ und zeigt, dass $uv^i w \notin L$
Sei n beliebig. Wir wählen wie in Runde 2 bereits gesagt $x = a^{2^n}$. Es gilt $x \in L$ und $|x| \geq n$.
Alle möglichen Stückelungen des Worts sind gemäß der Form: $u = a^p$ $v = a^q$ $w = a^{2^n - a^q - a^p}$ mit $p + q \leq n$ und $q \geq 1$.
Wir wählen $i = 2$, es gilt $uv^i w = a^{2^n + q}$. Es gilt $2^n \geq n \rightarrow p + q < 2^n$ und es gilt weiterhin $0 < q < 2^n$.
Dies bedeutet:

$$2^n < 2^n + q < 2^n + 2^n = 2 * 2^n = 2^{n+1}$$

Hieraus folgt, dass $2^n + q$ keine Zweierpotenz ist, dies wiederum verletzt die Eigenschaften der Sprache und somit ist $uv^i w \notin L$ q.e.d

Aufgabe 4

Beweisen Sie die folgende verschärfte Version des Pumping-Lemmas: Sei $L \in \Sigma^*$ eine reguläre Sprache. Dann existiert ein $n > 0$, so dass für alle $x \in L$ und alle $x_0, x_1, x_2 \in \Sigma^*$ mit $x = x_0x_1x_2$ und $|x_1| \geq n$ Wörter $u, v, w \in \Sigma^*$ existieren mit (a) $x_1 = uvw$, (b) $|v| \geq 1$ und (c) $x_0uv^iwx_2 \in L$ für alle $i \in \mathbb{N}$.

Sei $L \subseteq \Sigma^*$ eine reguläre Sprache. Dann existiert ein $n > 0$, sodass für alle $x \in L$ und alle $x_0, x_1, x_2 \in \Sigma^*$ mit $x = x_0x_1x_2$ und $|x_1| \geq n$ Wörter $u, v, w \in \Sigma^*$ existieren mit:

1. $x_1 = uvw$
2. $|v| \geq 1$ und
3. $x_0uv^iwx_2 \in L$ für alle $i \in \mathbb{N}$.

Beweis: Sei $n = |Z|$, wobei Z die Zustände des zugehörigen NFAs $M = (Z, \Sigma, S, \delta, E)$ sind. Ist $x_0x_1x_2 \in L$, so gibt es Zustände $m, n, o \in Z$ mit:

$$z_0 \xrightarrow{x_0} m \xrightarrow{x_1} n \xrightarrow{x_2} o \in E$$

Die Transition von m nach n kann in $|x_1| \geq |v| + 1 \geq n$ Schritten, also durch Begehung von so vielen Zuständen geschehen. Nach der Aussage des Schubkastenprinzips ist dies gleichbedeutend damit, dass zwei der Zustände gleich sein müssen. Nun folgt der Beweis analog dem des einfachen Pumping Lemmas.

Es gibt also in x_1 Zustände $z_0, z_1, \dots, z_m \in Z$ mit: $z_0 \in S$ (von x_1), $z_j \in \delta(z_{j-1}, a_j)$ für $1 \leq j \leq m$, und $z_m \in E$ (von x_1). Setze $u = a_1 \dots a_j, v = a_{j+1} \dots a_k, w = a_{k+1} \dots a_m$. Dann gilt:

- i $x_1 = a_1 \dots a_j a_{j+1} \dots a_k a_{k+1} \dots a_m = uvw$ und $x = x_0x_1x_2 = x_0a_1 \dots a_j a_{j+1} \dots a_k a_{k+1} \dots a_mx_2 = x_0uvw x_2$
- ii $|uv| = |a_1 \dots a_k| = k \leq n$
- iii $|v| = k - (j + 1) + 1 = k - j > 0$, da $(j < k)$
- iv Sei $i \geq 0$ beliebig. Es gelten: Es führt ein Weg von $z_0 \in x_0$ zu dem $z_0^1 \in x_1$ und ein Weg von $z_m^1 \in E$ von x_1 nach z_m von x_2 . Modellieren sozusagen die drei Teilwörter als eigenständige NFAs, bei deren die Überführungen auf die Endzustände der einzelnen NFAs auf die Startzustände des nächsten führen. Betrachten wir nun den NFA zu x_1 , so folgt nun wie im anderen Beweis auch, dass $uv^i w \in L(x_1)$ ist. Und dies in Kombi mit den weiteren Übergängen = L ist.

Aufgabe 5

Sei $\Sigma = \{a, b\}$. Wir betrachten die Sprache $L = \{w \in \Sigma^* \mid |w| \text{ ist gerade und } |w|_a \geq 1\}$. Bearbeiten Sie folgende Teilaufgaben:

- (a) Bestimmen Sie die Myhill-Nerode Äquivalenzklassen von L .
- (b) Geben Sie den Automaten M_L an

Aufgabe 6

Geben Sie einen Algorithmus an, der bei Eingabe eines DFAs M die Größe von $L(M)$ (also $|L(M)|$) berechnet (entweder eine natürliche Zahl n oder ∞).

Aufgabe 7

Sei Σ ein Alphabet. Zeigen Sie, dass für alle regulären Sprachen $K_1, K_2 \subseteq \Sigma^*$ und ihre Vereinigung $L = K_1 \cup K_2$ gilt, dass $\text{Index}(R_L) \leq \text{Index}(R_{K_1}) * \text{Index}(R_{K_2})$.

Übung 04

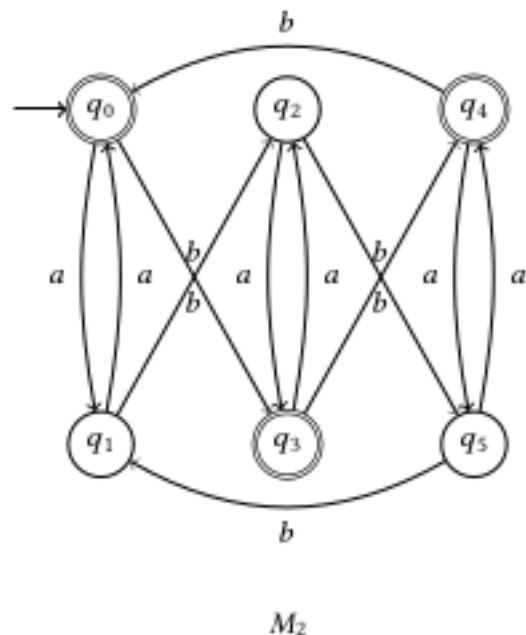
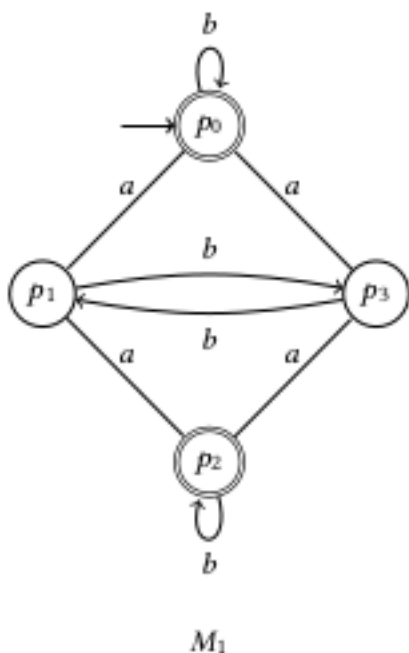
Aufgabe 1

Seien $u_1, u_2 \in \Sigma$ zwei Wörter und $L \subseteq \Sigma$ eine Sprache. Ein trennendes Wort für die Myhill-Nerode Äquivalenzklassen $[u_1]_L, [u_2]_L$ ist ein Wort $w \in \Sigma^*$, so dass $u_1w \in L, u_2w \notin L$ oder umgekehrt. Bearbeiten Sie die folgenden Teilaufgaben: Wir betrachten die paarweise verschiedenen Myhill-Nerode Äquivalenzklassen $[\epsilon], [a], [c]$ der Sprache $L_a = \{w \in \{a, b, c\}^* \mid |w|_a \text{ ist gerade oder } |w|_c \geq 1\}$. Geben Sie für jedes Paar von unterschiedlichen Äquivalenzklassen ein trennendes Wort an.

Wir betrachten die Myhill-Nerode Äquivalenzklassen der Sprache $L_b = \{0^l 10^m 10^{l+m} \mid l, m \in \mathbb{N}\}$. Geben Sie für $l \in \mathbb{N}, m \neq m'$ ein trennendes Wort für die Äquivalenzklassen $[0^l 10^m]$ und $[0^{l'} 10^{m'}]$ an.

Aufgabe 2

Wenden Sie das in der Vorlesung vorgestellte Verfahren an, um zu entscheiden, ob die beiden dargestellten DFAs M_1 und M_2 die gleiche Sprache akzeptieren.



Aufgabe 3

Wir betrachten das Universalitätsproblem:

Eingabe NFA $M = (Z, \Sigma, S, \delta, E)$.

Frage Gilt $L(M) = \Sigma^*$?

Geben Sie ein Verfahren an, welches das Universalitätsproblem löst. Begründen Sie Ihre Antwort.

Aufgabe 4

In Übung 2 Aufgabe 7 a) haben wir gezeigt, dass es für jeden NFA einen äquivalenten NFA mit genau einem Endzustand gibt. In dieser Aufgabe zeigen wir, dass dies für DFAs nicht der Fall ist. Bearbeiten Sie dazu folgende Teilaufgaben:

(a) Geben Sie einen DFA M an, sodass jeder DFA M_0 mit $L(M_0) = L(M)$ mindestens zwei akzeptierende Zustände hat.

(b) Beweisen Sie, dass Ihr Automat M diese Eigenschaft hat. Hinweis: Es gibt einen Automaten M , der diese Eigenschaft und eine endliche Sprache akzeptiert.

Aufgabe 5

In dieser Aufgabe betrachten wir Sprachen für die ein DFA wesentlich mehr Zustände haben muss als ein NFA. Sei $n \in \mathbb{N}$. Wir betrachten die Sprache $K_n = \{w \in \{a, b\}^* \mid |w| \geq n \text{ und der } n\text{-letzte Buchstabe von } w \text{ ist ein } a\}$.

(a) Geben Sie einen NFA mit minimaler Anzahl an Zuständen für K_n an.

(b) Bestimmen Sie den Index der Myhill-Nerode Äquivalenz von K_n , $\text{Index}(R_{K_n})$. Begründen Sie Ihre Antwort.

Aufgabe 6

Sei $\Sigma = \{a, b\}$. Geben Sie für die folgenden Sprachen jeweils eine kontextfreie Grammatik an.

(a) $L_a = \{a_n b_n \mid n \in \mathbb{N}\}$

(b) $L_b = \{w \in \Sigma^* \mid |w|_a = |w|_b\}$

(c) $L_c = \Sigma^* \setminus \{ww \mid w \in \Sigma^*\}$

Übung 05

Aufgabe 1

Aufgabe 2

Aufgabe 3

Übung 06

Aufgabe 1

Aufgabe 2

Aufgabe 3

Übung 07

Aufgabe 1

Aufgabe 2

Aufgabe 3

Übung 08

Aufgabe 1

Aufgabe 2

Aufgabe 3

Übung 09

Aufgabe 1

Aufgabe 2

Aufgabe 3

Übung 10

Aufgabe 1

Aufgabe 2

Aufgabe 3

Übung 11

Aufgabe 1

Aufgabe 2

Aufgabe 3

Übung 12

Aufgabe 1

Aufgabe 2

Aufgabe 3

Übung 13

Aufgabe 1

Aufgabe 2

Aufgabe 3

Übung 14

Aufgabe 1

Aufgabe 2

Aufgabe 3

Zusätzliche Aufgaben