Disclaimer

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

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.
   (a) Zeigen Sie, dass es mindestens zwei Personen in Deutschland mit gleich vielen Haaren gibt.

   Solution:

   (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

   Solution:

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.

   Solution:

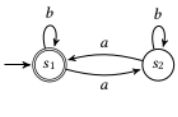
3. Eine Menge A heißt gleichmächtig zu einer Menge B, wenn es eine Bijektion von A nach B gibt. Zeigen Sie:
   (a) Die Menge der natürlichen Zahlen $\mathbb{N}$ ist nicht gleichmächtig zur Menge der reellen Zahlen $\mathbb{R}$.

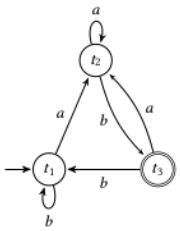
   Solution:

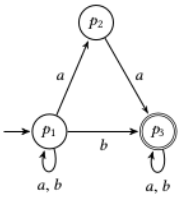
   (b) Keine Menge ist gleichmächtig zu ihrer Potenzmenge. (Satz von Cantor)

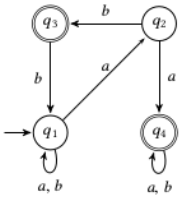
   Solution:

4. 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.

Solution:

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

Solution:

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

Solution:

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

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

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 $\sum^* \setminus L_i$ aus und überprüfen Sie, ob M_i auf diesen korrekt arbeitet.

Solution:

7. Sei $\sum = \{a, b, c\}$. Unter den folgenden 16 Sprachen über $\sum$ 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.

$$L_1 = \{w \in \sum^* | \quad |w|_a = |w|_b = |w|_c\}$$

$$L_2 = \{w \in \sum^* | \quad |w|_a = |w|_b\}$$

$$L_3 = \{w \in \sum^* | \quad |w|_a = 0\}$$

$$L_4 = \{w \in \sum^* | \quad |w|_a = 2\}$$

$$L_5 = \{w \in \sum^* | \quad |w|_a = 4\}$$

$$L_6 = \{b, c\}^*\{a\}\{b, c\}^*\{a\}\{b, c\}^*$$

$$L_7 = \{a\}\{ba\}^*\{b\}$$

$$L_8 = \{a^n b^n | n \in \mathbb{N}\}$$

$$L_9 = L_2 \cap \{a\}^*\{b\}^*$$

$$L_{10} = L_2 \cap \{w \in \sum^* | \quad |w|_b = |w|_c\}$$

$$L_{11} = (L_3 L_4)^2$$

$$L_{12} = \sum^* \setminus L_3$$

$$L_{13} = L_2^3$$

$L_{14} = \{ab\}^+$
$L_{15} = \{b, c\}^*$
$L_{16} = \sum^* \{a\} \sum^*$

Solution:

8. 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 $|\sum^{\leq n}| = 2^{n+1} - 1$.

Solution:

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

$$S \to A|C \qquad A \to Aa|a \qquad Bb \to bb \qquad Bc \to bbc \qquad C \to BCc|c.$$

   (a) Geben Sie eine Ableitung von bbbccc an.

   Solution:

   (b) Geben Sie eine möglichst kurze aber präzise Beschreibung von L(G) an. Begründen Sie Ihre Antwort.

   Solution:

10. 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 $\sum = \{a, b, c\}$

   Solution:

   (b) $L(G_2) = \{ww^R | w \in \{a, b\}^*$: w startet mit einem b$\}$ Hinweis: Für $w = w_1 w_2 ... w_{n-1} w_n$ sei $w^R = w_n w_{n-1} ... w_2 w_1$ das umgekehrte Wort.

   Solution:

   (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 $(f * g)$.

   Solution:

11. 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) $L_a = \{w \in \{a, b, c\}^* |$ entweder kommen a und b in w vor oder weder a noch b$\}$

    Solution:

    (b) $L_b = \{w \in \{a, b, c\}^* |$w enthält nicht das Infix bc$\}$

    Solution:

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

    Solution:

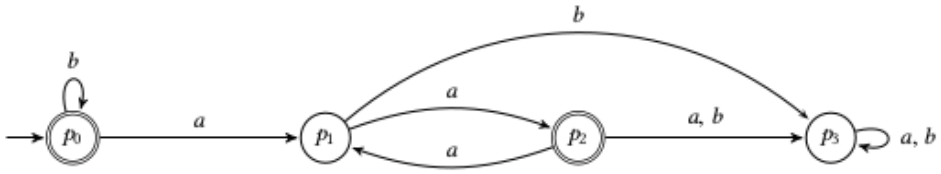
13. Sei $L \subseteq \sum^*$ eine Sprache. Unter der Verdopplung von L verstehen wir die Sprache $2 * L := \{ww | w \in L\}$. Überprüfen Sie, ob die Klasse der regulären Sprachen unter Verdopplung abgeschlossen ist. Beweisen Sie Ihre Behauptung!

    Solution:

14. Sei $\sum$ ein Alphabet (eine endliche Menge). Zeigen Sie, dass $\sum^*$ abzählbar ist.

    Solution:

15. Bearbeiten Sie folgende Teilaufgaben:
    (a) Beschreiben Sie die Sprache des folgenden Automaten kurz und präzise

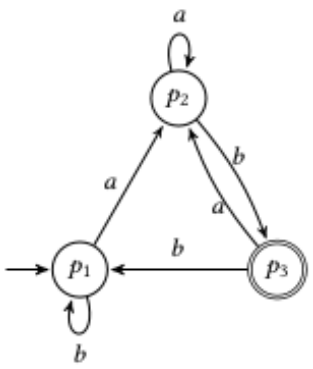


    Solution:

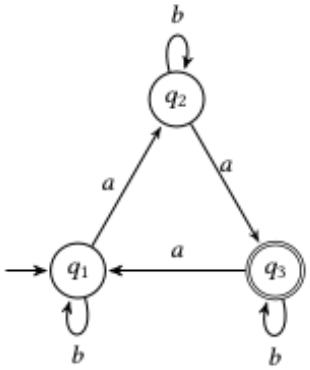
    (b) Sei $\sum = \{a, b, c\}$. Geben Sie einen DFA an, der die Sprache $L = \{w \in \sum^* | |w|_a \leq 1$ und $|w|_b = 0\}$ akzeptiert. Dabei steht für $x \in \sum, w \in \sum^*$ der Ausdruck $|w|_x$ für die Anzahl der x in w.

Solution:

16. Gegeben seien die folgenden DFAs M_1 und M_2.



M_1



M_2

Konstruieren Sie folgende Automaten:

(a) einen DFA $M_\cap$ mit $L(M_\cap) = L(M_1) \cap L(M_2)$,

Solution:

(b) einen NFA $M_\cdot$ mit $L(M_\cdot) = L(M_1) * L(M_2)$ und

Solution:

(c) einen NFA $M_* mit L(M_*) = L(M_1)^*$.

Solution:

17. Zeigen Sie die folgenden Aussagen:

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

Solution:

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

Solution:

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

Solution:

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

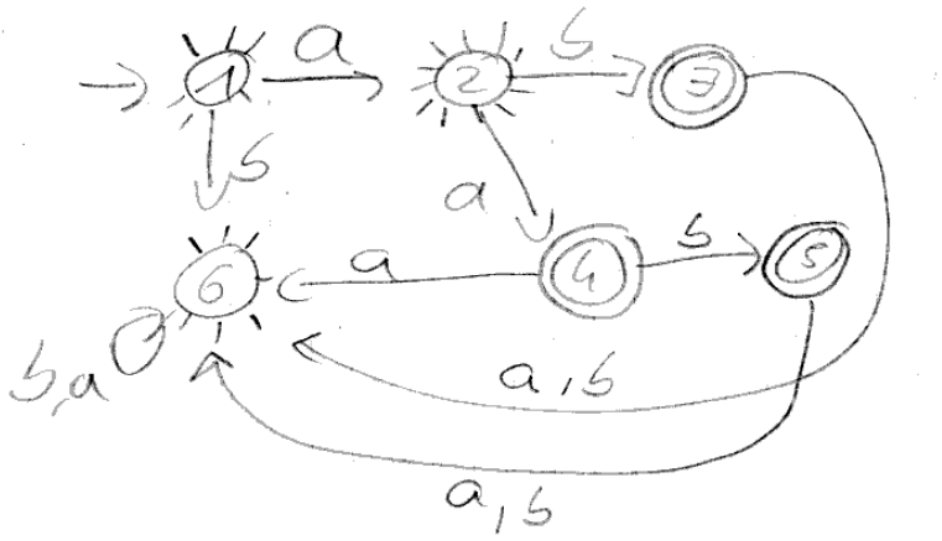
- $L_1 = \{w \in \sum^* |$der vorletzte Buchstabe von w ist ein a$\}$
- $L_1 = \sum^* \setminus \{aa, ab, aab\}$
- $L_3 = \{w \in \sum^* \mid$ in w kommt ein Buchstabe zweimal direkt hintereinander vor$\}$
- $L_4 = \sum^* \setminus L_3$

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

Solution:

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

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



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)

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

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

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

Solution: Behauptung: Die Sprache L ist nicht regulär.

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

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

   i $x = uvw$
   ii $\mid uv \mid \leq n$
   iii $\mid v \mid \geq 1$
   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 $\mid uv \mid <= n$ mit $\mid v \mid \geq k$. Es gilt $v \in \{a\}^+$. Nun sei $i = 0$, damit ist $x = a^{n-k} b^j$, da $\mid v \mid \geq 1$ ist, und da nun $j = n - k$ gilt, ist $n = j$, was allerdings der Bedingung $n > j$ widerspricht. Wählen wir $\mid v \mid = k$ mit $k \in \mathbb{N}$. so gilt: $uw = a^{(n-k)} b^j$
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

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

Solution:

1. Runde: G wählt eine Zahl $n \geq 1$
2. Runde: B wählt $x \in L$ mit $\mid x \mid \geq n$. Sei $x = a^{(2^n)}$.
3. Runde: G wählt u, v, w mit i) $x = u, v, w$ ii) $\mid uv \mid \leq n$ iii) $\mid v \mid \geq 1$
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 $\mid x \mid \geq n$.
   Alle möglichen Stückelungen des Worts sind gemäß der Form: $u = a^p \quad v = a^q \quad w = a^{2^n} - a^q - a^p$ 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 $uv^iw \notin L$ q.e.d

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

Solution: Sei $L \subseteq \Sigma^*$ eine reguläre Sprache. Dann exisitiert 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 $\mid x_1 \mid \geq n$ Wörter $u, v, w \in \Sigma^*$ existieren mit:

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

Beweis: Sei $n = \mid Z \mid$, 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 $\mid x_1 \mid \geq \mid v \mid +1 \geq n$ 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 $z_0, z_1, \ldots, z_m \in Z$ 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 $u = a_1 \ldots a_j, v = a_{j+1} \ldots a_k, w = a_{k+1} \ldots a_m$. Dann gilt:

i $x_1 = a_1 \ldots a_j a_{j+1} \ldots a_k a_{k+1} \ldots a_m = uvw$ und $x = x_0x_1x_2 = x_0a_1 \ldots a_j a_{j+1} \ldots a_k a_{k+1} \ldots a_m x_2 = x_0uvwx_2$

ii $\mid uv \mid = \mid a_1 \ldots a_k = k \leq n$

iii $\mid v \mid = k - (j+1) + 1 = k - j > 0$, 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 $uv^iw \in L(x_1)$ ist. Und dies in Kombi mit den weiteren Übergängen $= L$ ist.

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

    (a) Bestimmen Sie die Myhill-Nerode Äquivalenzklassen von L.

    Solution:

    (b) Geben Sie den Automaten M_L an

    Solution:

24. 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 ∞).

    Solution:

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

    Solution:

26. Seien $u_1, u_2 \in \sum$ zwei Wörter und $L \subseteq \sum$ eine Sprache. Ein trennendes Wort für die Myhill-Nerode Äquivalenz-klassen $[u_1]_L, [u_2]_L$ ist ein Wort $w \in \sum^*$, so dass $u_1 w \in L, u_2 w < L$ oder umgekehrt. Bearbeiten Sie die folgenden Teilaufgaben:

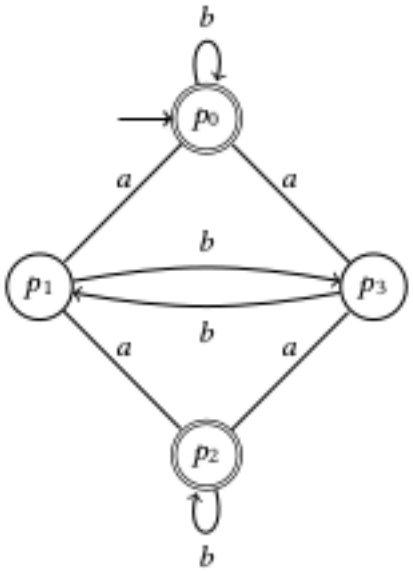
    (a) Wir betrachten die paarweise verschiedenen Myhill-Nerode Äquivalenzklassen $[\epsilon], [a], [c]$ der Sprache $L_a = \{w \in \{a, b, c\}^* || w|_a$ ist gerade oder $|w|c \geq 1\}$. Geben Sie für jedes Paar von unterschiedlichen Äquivalenzklassen ein trennendes Wort an.

    Solution:

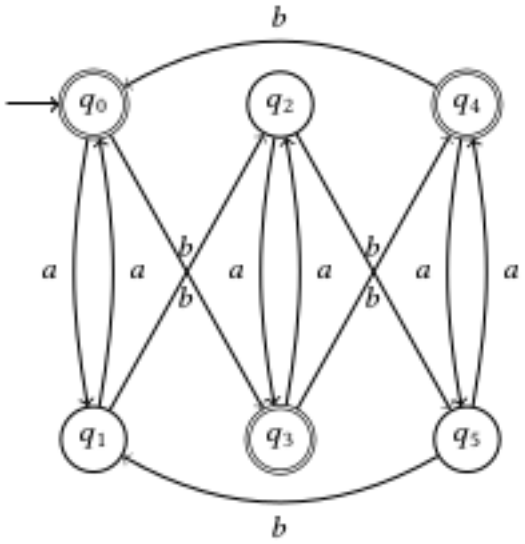
    (b) Wir betrachten die Myhill-Nerode Äquivalenzklassen der Sprache $L_b = \{0^l 10^m 10^{l+m} | l, m \in \mathbb{N}\}$. Geben Sie für $l \in \mathbb{N}, m \neq m'$ ein trennendes Wort für die Äquivalenzklassen $[0^l 10^m]$ und $[0^l 10^m]$ an.

    Solution:

27. 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.



M_1



M_2

Solution:

28. Wir betrachten das Universalitätsproblem:

    Eingabe NFA $M = (Z, \sum, S, \delta, E)$.
    Frage Gilt $L(M) = \sum^*$?

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

    Solution:

29. 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.

    Solution:

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

Solution:

30. 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 $K_n = \{w \in \{a,b\}^* || w| \geq n$ und der n-letzte Buchstabe von w ist ein a.

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

    Solution:

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

    Solution:

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

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

    Solution:

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

    Solution:

    (c) $L_c = \sum^* \setminus \{ww | w \in \sum^*\}$

    Solution:

32. Entscheiden Sie für jede der folgenden Sprachen, ob sie regulär oder kontextfrei und nicht regulär ist. Geben sie dafür eine rechtslineare Grammatik an oder geben Sie eine kontextfreie Grammatik an und zeigen Sie, dass die Sprache nicht regulär ist.

    (a) $L_a = \{w \in \{a,b,c\}^* || w|_a$ ist gerade oder $|w|_c \geq 1\}$

    Solution:

    (b) $L_b = \{uv | u,v \in \{a,b,c\}^*$ und $|u|_a > |v|_b\}$

    Solution:

    (c) $L_c = \{a^l b a^m b a^n | l = m$ oder $l = n\}$

    Solution:

    (d) $L_d = \{r \in \{a,b,\lambda,\varnothing,+,,*,(,)\}^* |$ r ist ein regulärer Ausdruck über $\sum\}$

Solution:

(e) $L_e = \{r \in L_d | \epsilon \in L(r)\}$

Solution:

33. Konstruieren Sie zu zwei kontextfreien Grammatiken $G_1 = (V_1, \sum, P_1, S_1)$ und $G_2 = (V_2, \sum, P_2, S_2)$

(a) eine kontextfreie Grammatik $G_\cup$ mit $L(G_\cup) = L(G_1) \cup L(G_2)$.

Solution:

(b) eine kontextfreie Grammatik $G_\circ$ mit $L(G_\circ) = L(G_1) * L(G_2)$.

Solution:

(c) eine kontextfreie Grammatik G_* mit $L(G_*) = L(G_1)^*$

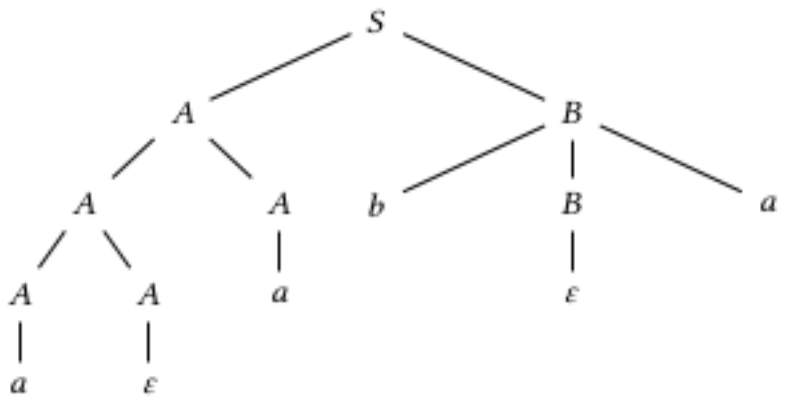
Solution:

Hinweis: Sie müssen die Korrektheit Ihrer Konstruktionen nicht beweisen.

34. Wir betrachten die Spiegelung einer Sprache. Zeigen Sie, dass die Klasse der kontext-freien Sprachen unter Spiegelung abgeschlossen ist.

Solution:

35. Betrachten Sie diejenige kontextfreie Grammatik G über $\sum = \{a, b\}$ mit Startvariable S, die folgenden Ableitungsbaum T ermöglicht und nicht mehr Produktionen enthält, als für T not-



wendig sind.

(a) Geben Sie das Blattwort $\alpha(T)$ von T an und ermitteln Sie weiterhin die Variablen und Produktionen der Grammatik G.

Solution:

(b) Konstruieren Sie die zu T gehörige Links- und Rechtsableitung. Geben Sie eine weitere zu T gehörige Ableitung an, die weder Links- noch Rechtsableitung ist.

Solution:

(c) Geben Sie einen von T verschiedenen S-Ableitungsbaum für das Wort $\alpha(T)$ an. Ist die Grammatik G mehrdeutig?

Solution:

(d) Beschreiben Sie die von G erzeugte Sprache und geben Sie eine eindeutige Grammatik G_0 mit $L(G_0) = L(G)$ an.

Solution:

36. Betrachten Sie die nachstehende Grammatik G mit Startsymbol $S : S \to BA|a$, $A \to BS|\epsilon$, $B \to bBaB|b$

(a) Überführen Sie G in eine äquivalente Grammatik G_0 in Chomsky-Normalform.

Solution:

(b) Entscheiden Sie mithilfe des CYK-Algorithmus, welche der Wörter $w_1 = bbbaba$ und $w_2 = bbaab$ von Ihrer in (a) berechneten Grammatik erzeugt werden.

Solution:

(c) Geben Sie für diejenigen Wörter aus Aufgabe (b), die von der Grammatik G erzeugt werden, jeweils einen Ableitungsbaum und eine Linksableitung an.

Solution:

37. Wir betrachten die kontextfreie Grammatik G mit Startvariable S und den folgenden Produktionen: $S \to ABC$, $A \to aA|\epsilon$, $B \to aDb|D$, $C \to bC|aC|\epsilon$, $D \to bDa|ba$
Bearbeiten Sie die folgenden Teilaufgaben:

(a) Geben Sie eine kurze Beschreibung von $L(G)$ an.

Solution:

(b) Geben Sie je eine Linksableitung für abab, babaa und abbaab an.

Solution:

(c) Zeigen Sie, dass G eine mehrdeutige Grammatik ist.

Solution:

(d) Zeigen Sie nun, dass $L(G)$ nicht inhärent mehrdeutig ist. Geben Sie also eine eindeutige kontextfreie Grammatik G_0 mit $L(G_0) = L(G)$ an. Sie müssen nicht zeigen, dass G_0 eindeutig ist.

Solution:

38. Betrachten Sie die kontextfreie Grammatik G mit Startsymbol S und den nachstehenden Produktionen:
$S \to Z|(S+S)|(S*S)$, $Z \to Q|PY$, $Y \to Q|YY|epsilon$, $Q \to 0|P$, $P \to 1$
Bearbeiten Sie folgende Teilaufgaben:

(a) Geben Sie eine Ableitung des Wortes $w = (100 + 1)$ in G an und geben Sie eine kurze, aber präzise Beschreibung von $L(G)$ an.

Solution:

(b) Überführen Sie G mit dem Verfahren aus der Vorlesung in eine äquivalente Grammatik G_0 in Chomsky-Normalform.

Solution:

(c) Geben Sie eine Ableitung des Wortes w in G_0 an.

Solution:

39. Gegeben sei die kontextfreie Grammatik G in Chomsky-Normalform mit dem Startsymbol S und den Regeln
$S \to AB|CC$, $A \to BA|a$, $B \to AC|b$, $C \to CC|c$
Überprüfen Sie mithilfe des CYK-Algorithmus, folgende Wörter. Geben Sie für jedes dieser beiden Wörter, welches in $L(G)$ enthalten ist, je eine Ableitung und einen Ableitungsbaum des Wortes an.

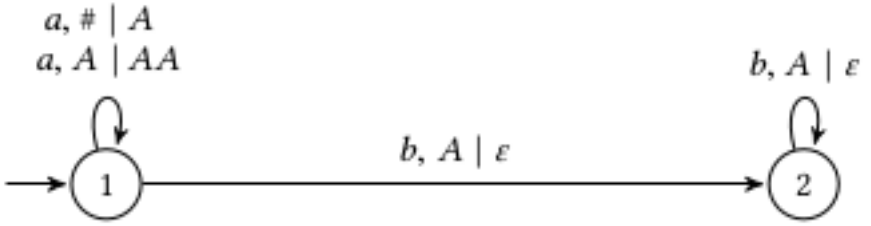
(a) $aacc \in L(G)$.

Solution:

(b) $bacca \in L(G)$.

Solution:

40. Wir betrachten den PDA M mit folgender grafischen Darstellung mit Kellerinitialisierungszei-



chen #:

(a) Gilt $aabb \in L(M)$? Gilt $aabbbb \in L(M)$?

Solution:

(b) Geben Sie eine einfache, aber präzise Beschreibung von $L(M)$ an.

Solution:

41. Sei $L = \{a^n b a^{2n} | n \in \mathbb{N}\}$.

(a) Geben Sie einen PDA M_1 an mit $L(M_1) = L$.

Solution:

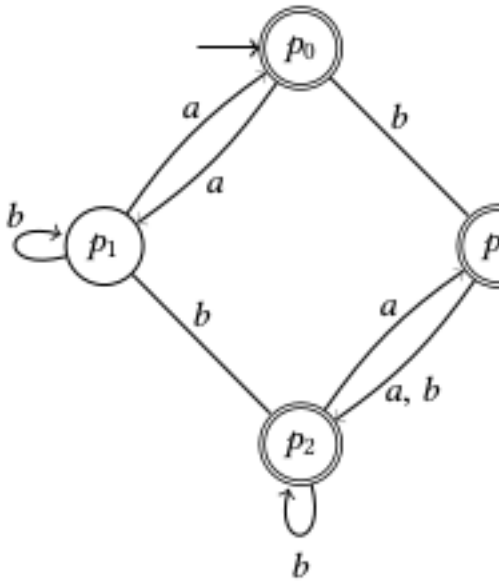
(b) Geben Sie einen PDA M_2 mit genau einem Zustand an mit $L(M_2) = L$.

Solution:

42. Sei G die kontextfreie Grammatik mit Startvariable A 1 und den folgenden Produktionen. Konstruieren Sie mithilfe des Verfahrens aus der Vorlesung eine Grammatik G_0 in Greibach-Normalform mit $L(G_0) = L(G)$.
$A_1 \to 0|A_2A_2$, $A_2 \to 1|A_1A_1$

Solution:

43. Wenden Sie das in der Vorlesung vorgestellte Verfahren an, um zu entscheiden, ob die beiden



M_1

dargestellten DFAs M_1 und M_2 die gleiche Sprache akzeptieren.

Solution:

44. Die Syntax von Programmiersprachen wird in der Regel in der Erweiterten Backus-Naur-Form 1 (kurz: EBNF) angegeben. Wir wollen in dieser Aufgabe exemplarisch zeigen, dass solche Programmiersprachen kontextfrei sind. Betrachten Sie also die folgende (funktionale) Programmiersprache: $< Nicht - Null >::=' 1'|'2'|...|'9'$
$< Zahl >::=' 0'| < Nicht - Null > '0'| < Nicht - Null >$
$< Variable >::=' x' < Zahl >$
$< Wert >::=< Variable > |['-'] < Zahl >$
$< Verzweigung >::=' if' < Wert >'='< Wert >' then' < Programm > ['else' < Programm >]$
$< Programm >::=< Wert > |'('< Programm > ('+'|'*'|' -' |' :') < Programm >')'| < Verzweigung >$
Geben Sie eine kontextfreie Grammatik an, die alle möglichen Werte von $< Programm >$ erzeugt.

Solution:

45. Wir betrachten arithmetische Ausdrücke in Präfixnotation über den Konstanten 0,1,2 und mit den Operatoren + (Addition) und * (Multiplikation). Bei der Präfixnotation stehen der Operator vor den Operanden und es gibt keine Klammern. Die Notation ist dennoch

eindeutig, so entspricht zum Beispiel +21 dem Ausdruck $(2 + 1)$ und $2 + +210$ entspricht $(2((2 + 1) + 0))$.

(a) Geben Sie eine Regelmenge P an, sodass $G = (\{S, M_0, M_1, M_2\}, \{0, 1, 2, +, *\}, P, S)$ eine kontextfreie Grammatik ist, wobei von S alle arithmetischen Ausdrücke in Präfixnotation erzeugt werden und von M_i alle, die modolu 3 zu i ausgewertet werden.

Solution:

(b) Geben Sie einen Kellerautomaten an, der genau die arithmetischen Ausdrücke in Postfixnotation akzeptiert, die modulo 3 zu 0 ausgewertet werden.

Solution:

46. Sei $1 \leq k \in \mathbb{N}$. Ein k-PDA $M = (Z, \sum, \Gamma, \delta, z_0, \#)$ ist ein PDA mit der Eigenschaft, dass der Keller höchstens k Elemente aufnehmen kann. Eine Konfiguration ist also ein Tripel $c \in Z \times \sum^* \times \Gamma^k$ und die Konfigurationsüberführung ist wie folgt definiert: Zunächst wird die Transition wie in den klassischen PDAs ausgeführt und im Falle eines Kellerüberlaufs wird anschließend der Inhalt auf die obersten k Symbole gekürzt. Zeigen Sie, dass die von einem k-PDA M akzeptierte Sprache $L(M)$ regulär ist.

Solution:

47. Wir betrachten arithmetische Ausdrücke in Postfixnotation über den Konstanten 0,1,2 und mit den Operatoren + (Addition) und * (Multiplikation). Diese Ausdrücke werden von der Grammatik $G = (\{S\}, \{0, 1, 2, +, *\}, P, S)$ erzeugt, wobei P gegeben ist durch: $S \vdash 0|1|2|SS+|SS*$
Hierbei werden zuerst die Operanden und dann der Operator notiert. Zum Beispiel entspricht 12+ dem Ausdruck $(1 + 2)$ und $012 + +2*$ entspricht $((0 + (1 + 2)) * 2)$. Geben Sie einen Kellerautomaten an, der genau die arithmetischen Ausdrücke in Postfixnotation akzeptiert, die modulo 3 zu 0 ausgewertet werden. Hinweis: Es gibt so einen Kellerautomaten mit Kelleralphabet $\Gamma = \{\#, 0, 1, 2\}$.

Solution:

48. Sei $\sum = \{a, b\}$. Entscheiden Sie für jede der folgenden Sprachen, ob sie kontextfrei oder nicht kontextfrei ist. Beweisen Sie Ihre Aussagen.

(a) $L_a = \{a^n ba^n ba^n | n \in \mathbb{N}\}$

Solution:

(b) $L_b = \sum^* \setminus L_a$

Solution:

49. Ziel dieser Aufgabe ist es, zu zeigen, dass die Klasse der deterministisch kontextfreien Sprachen nicht unter Vereinigung abgeschlossen ist. Bearbeiten Sie dazu folgende Teilaufgaben:

(a) Zeigen Sie, dass die Sprache $\{a^k b^l c^m | k, l, m \in \mathbb{N}, k \neq l\}$ deterministisch kontextfrei ist.

Solution:

(b) Folgern Sie aus (a), dass $L = \{a^k b^l c^m | k, l, m \in \mathbb{N}, k \neq l \text{ oder } k \neq m \text{ oder } l \neq m\}$ kontextfrei ist.

Solution:

(c) Angenommen, L wäre deterministisch kontextfrei. Zeigen Sie, dass unter dieser Annahme auch die Sprache $K = \{a^m b^m c^m | m \in \mathbb{N}\}$ kontextfrei wäre.

Solution:

(d) Folgern Sie unter Verwendung aus (a) und (c), dass die Klasse der deterministisch kontextfreien Sprachen nicht unter Vereinigung abgeschlossen ist. Hinweis: Die Sprache K ist nicht kontextfrei.

Solution:

50. Zeigen Sie, dass folgende Sprachen nicht kontextfrei sind:

(a) $L_a = \{a^k b^m a^{k*m} | k, m \in \mathbb{N}\}$

Solution:

(b) $L_b = \{0^p | p \text{ Primzahl}\}$

Solution:

(c) $L_c = \{s\#t | s, t \in \{a, b\}^* \text{ und s ist ein Infix von t }\}$

Solution:

51. In dieser Aufgabe zeigen wir, dass die Klasse der deterministisch kontextfreien Sprachen nicht unter Konkatenation abgeschlossen ist.

(a) Zeigen Sie, dass $L_2 = \{b^i c^j d^k | i \neq j\} \cup \{a b^i c^j d^k | j \neq k\}$ deterministisch kontextfrei ist.

Solution:

(b) Geben Sie eine deterministisch kontextfreie Sprache L_1 an so, dass $L_1 * L_2$ nicht deterministisch kontextfrei ist.

Solution:

(c) Zeigen Sie, dass $L_1 * L_2$ nicht deterministisch kontextfrei ist.

Solution:

52. Geben Sie einen Algorithmus an, der folgende Funktion berechnet:

Eingabe kontextfreie Grammatik G

Ausgabe: $|L(G)| \in \mathbb{N} \cup \{\infty\}$

Solution:

53. Beweisen Sie das doppelte Pumping-Lemma für reguläre Sprachen, das wie folgt lautet: Wenn L eine reguläre Sprache ist, dann gibt es $n \geq 1$ derart, dass für alle $z \in L$ mit $|z| \geq n$ gilt: Es gibt Wörter $u, v, w, x, y \in \sum^*$ mit

- (i) $z = uvwxy$
- (ii) $|uvwx| \leq n$
- (iii) $|v|, |x| \geq 1$
- (iv) $uv^iwx^jy \in L$ für alle $i, j \in \mathbb{N}$.

Hinweis: Orientieren Sie sich am Beweis des Pumping-Lemmas für reguläre Sprachen aus der Vorlesung.

Solution:

54. Wir betrachten das vereinfachte doppelte Pumping-Lemma für kontextfreie Sprachen (welches nicht gilt): Wenn L eine kontextfreie Sprache ist, dann gibt es $n \geq 1$ derart, dass für alle $z \in L$ mit $|z| \geq n$ gilt: Es gibt Wörter $q, r, s, t, u, v, w, x, y \in \sum^*$ mit

- (i) $z = qrstuvwxy$
- (ii) $|rt|, |vx| \geq 1$
- (iii) $qr^ist^iuv^jwx^jy \in L$ für alle $i, j \in \mathbb{N}$.

(a) Zeigen Sie, dass die Sprache $\{a^nb^n | n \in \mathbb{N}\}$ ein Gegenbeispiel für das Lemma ist.

Solution:

(b) Formulieren Sie ein gültiges doppeltes Pumping-Lemma für kontextfreie Sprachen. Ein Korrektheitsbeweis ist nicht nötig. Hinweis: Orientieren Sie sich für die Formulierung an Ihrem Beweis aus Aufgabe 1 und an dem Beweis für das Pumping-Lemma für kontextfreie Sprachen aus der Vorlesung.

Solution:

55. Sei $\sum$ ein Alphabet und $K, L \subseteq \sum^*$. Beweisen Sie die folgenden Aussagen:

(a) Ist K deterministisch kontextfrei und L regulär, so ist $K \cap L$ deterministisch kontextfrei.

Solution:

(b) Ist L regulär beziehungsweise kontextfrei, so gilt dies auch für den Abschluss von L unter Präfixen, d.h. für die Sprache $\{u \in \sum^* | \exists v \in \sum^* : uv \in L\}$.

Solution:

56. Geben Sie einen Algorithmus an, der bei Eingabe eines PDAs M und eines NFAs N entscheidet, ob $L(M)$ Teilmenge von $L(N)$ ist. Anmerkung: Später in der Vorlesung werden wir zeigen, dass es keinen Algorithmus geben kann, der die umgekehrte Teilmengenbeziehung entscheidet.

Solution:

57. Geben Sie für folgende Funktionen je eine primitiv rekursive Definition und ein Loop-Programm an.

(a) $f : \mathbb{N} \to \mathbb{N} : n \to n^2$

Solution:

(b) $f : \mathbb{N} \to \mathbb{N} : n \to n^n$

Solution:

58. Zeigen Sie, dass die Funktion $c : \mathbb{N}^2 \to \mathbb{N} : (m, n) \to m + \binom{m+n+1}{2}$ eine Bijektion ist.

Solution:

59. Geben Sie ein Loop-Programm an, das für zwei gegebene Zahlen $m, n \in \mathbb{N}$ den größten gemeinsamen Teiler berechnet.

Solution:

60. Geben Sie für folgende Funktionen je eine primitiv rekursive Definition und ein Loop-Programm an:

(a) $f : \mathbb{N} \to \mathbb{N}, n \to 2^n$

Solution:

(b) $f : \mathbb{N} \to \mathbb{N}, n \to n!$

Solution:

61. Sei $a(k,n) := ack(k,n) mod 2$, d.h., $a(k,n)$ ist die Parität des Wertes $ack(k,n)$. Zeigen oder widerlegen Sie, dass die Funktion a Loop-berechenbar ist.

Solution:

62. Sei $(f_k)_{k\in\mathbb{N}}$ eine Folge von Loop-berechenbaren Funktionen $f_k : \mathbb{N} \to \mathbb{N}$, so dass jedes f_k durch ein Loop-Programm mit k Loop-Schleifen berechnet werden kann, nicht aber durch ein Programm mit $k-1$ Loop-Schleifen. Zeigen oder widerlegen Sie: Die Funktion g mit $g(k,n) = f_k(n)$ ist Loop-berechenbar.

Solution:

63. Jedes Jahr, kurz vor Heiligabend, startet der Weihnachtsmann auf seinem Keller-Rechner ein Loop-Programm, welches die optimale Route für das Verteilen der Geschenke berechnet. Dieses Jahr geschieht jedoch eine Katastrophe. Schon bei der Eingabe gerät eine Zuckerstange in das Getriebe, wodurch der unendliche Kellerspeicher in sich zusammen fällt. „Oh nein, das war der letzte Rechner mit Kellerspeicher! Woher soll ich denn jetzt wissen, wo ich lang fliegen soll?", fragt Rudolf. „Ich weiß auch nicht weiter", sagt der Weihnachtsmann, „Wir haben zwar noch andere Rechner auf denen Loop-Programme laufen, aber der Kellerspeicher ist nötig um den Geschenkstapel auf dem Schlitten zu simulieren." „Könnten wir nicht den Stack in einem Loop-Programm simulieren?" schlägt ein Wichtel vor. „Das könnte funktionieren. . . ", sagt der Weihnachtsmann.
Helfen Sie dem Weihnachtsmann, indem Sie zeigen, dass jede Funktion, die durch ein Loop-Programm mit Stack berechnet wird, auch durch ein Loop-Programm ohne Stack berechnet werden kann. Erklärung: Loop-Programme mit Stack verfügen über alle Befehle, die Loop-Programme besitzen. Zusätzlich besitzen sie einen Stack auf den mit dem Befehl $push(x_i)$ der Wert von x_i gelegt werden kann und von dem mit dem Befehl $x_i = pop$ der oberste Wert ausgelesen werden kann, welcher dann gleichzeitig vom Stack entfernt wird. Falls der Stack leer ist gibt *pop* den Wert 0 zurück.
Zeigen Sie, dass für jedes Loop-Programm mit Stack, das eine Funktion berechnet, ein Loop-Programm (ohne Stack) existiert, welches die gleiche Funktion berechnet.

Solution:

64. Welche Funktion berechnet die folgende Turingmaschine M? $M = (\{z_0, z_1, z_2, z_3, z_e\}, \{0,1\}, \{0,1,\square\}, \delta, z_0, \square, \{z_e\})$

δ	0	1	$\square$
z_0	$(z_0, 0, R)$	$(z_1, 1, R)$	$(z_3, 0, L)$
z_1	$(z_1, 0, R)$	$(z_1, 1, R)$	$(z_2, \square, L)$
z_2	$(z_2, 1, L)$	$(z_3, 0, L)$	$(z_e, 0, N)$
z_3	$(z_3, 0, L)$	$(z_3, 1, L)$	$(z_e, \square, R)$
z_e	$(z_e, 0, N)$	$(z_e, 1, N)$	$(z_e, \square, N)$

Solution:

65. Geben Sie formal je eine Turingmaschine über dem Alphabet $\sum = \{0, 1\}$ an, die als Eingabe ein Wort $w \in \sum^+$ erhält und

    (a) den Wert $\lceil w/2 \rceil$ berechnet, wobei w als Binärzahl mit dem höchstwertigsten Bit links betrachtet wird.

    Solution:

    (b) 1 ausgibt, falls w ein Palindrom ist, und 0 sonst.

    Solution:

66. Geben Sie für jede der folgenden Sprachen je einen PDA an, der die Sprache akzeptiert.

    (a) $L_1 = \{a^n b^{3n} | n \in \mathbb{N}\}$

    Solution:

    (b) $L_2 = \{a^n b^m | n \leq m \leq 2n\}$

    Solution:

    (c) $L_3 = \{w \in \{a, b\}^* | 2 * |w|_a = 3 * |w|_b\}$

    Solution: