Lehrstuhl für Theoretische Informatik
Fakultät für Informatik
Technische Universität München

Eexam
Sticker mit SRID hier einkleben

Hinweise zur Personalisierung:

- Ihre Prüfung wird bei der Anwesenheitskontrolle durch Aufkleben eines Codes personalisiert.
- Dieser enthält lediglich eine fortlaufende Nummer, welche auch auf der Anwesenheitsliste neben dem Unterschriftenfeld vermerkt ist.
- Diese wird als Pseudonym verwendet, um eine eindeutige Zuordnung Ihrer Prüfung zu ermöglichen.

Einführung in die Theoretische Informatik

Klausur:	IN0011 / Endterm	**Datum:**	Mittwoch, 3. August 2022
Prüfer:	Prof. Dr. Dr. h.c. Javier Esparza	**Uhrzeit:**	08:15 – 11:15

	A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A 9
I									

Bearbeitungshinweise

- Diese Klausur umfasst **16 Seiten** mit insgesamt **9 Aufgaben**.
  Bitte kontrollieren Sie jetzt, dass Sie eine vollständige Angabe erhalten haben.
- Die Gesamtpunktzahl in dieser Klausur beträgt 100 Punkte.
- Das Heraustrennen von Seiten aus der Prüfung ist untersagt.
- Als Hilfsmittel sind zugelassen:
  - ein **beidseitig handschriftlich beschriebenes DIN A4 Blatt**
  - ein **analoges Wörterbuch** Deutsch $\leftrightarrow$ Muttersprache **ohne Anmerkungen**
- Mit * gekennzeichnete Teilaufgaben sind ohne Kenntnis der Ergebnisse vorheriger Teilaufgaben lösbar.
- **Es werden nur solche Ergebnisse gewertet, bei denen der Lösungsweg erkennbar ist.**
- Schreiben Sie weder mit roter / grüner Farbe noch mit Bleistift.
- Schalten Sie alle mitgeführten elektronischen Geräte vollständig aus, verstauen Sie diese in Ihrer Tasche und verschließen Sie diese.
- $0 \in \mathbb{N}$

Hörsaal verlassen von __________ bis __________ / Vorzeitige Abgabe um __________

Aufgabe 1 **Reguläre und kontextfreie Sprachen (10 Punkte)**

Für die folgenden Fragen ist eine Begründung nicht gefordert. Sie erhalten die Punkte auf eine Teilaufgabe genau dann, wenn Sie alle Antwortmöglichkeiten korrekt angekreuzt haben. Es ist immer mindestens eine Antwortmöglichkeit richtig. Jede Teilaufgabe bringt 2 Punkte.

Kreuzen Sie richtige Antworten an ☒
Kreuze können durch vollständiges Ausfüllen gestrichen werden ■
Gestrichene Antworten können durch nebenstehende Markierung erneut angekreuzt werden ×■

In dieser Aufgabe verwenden wir durchgehend das Alphabet $\Sigma := \{a, b\}$.

a) Seien $A, B, C \subseteq \Sigma^*$ Sprachen. Welche der folgenden Aussagen sind wahr?

☐ $A \neq B \Rightarrow A^* \neq B^*$

☐ $A(B \cup C) = AB \cup AC$

☐ $|A||B| \leq |AB|$

b) Sei r ein regulärer Ausdruck und M ein NFA mit n Zuständen und $L(M) = L(r)$. Welche Aussagen sind wahr?

☐ Es gibt einen NFA für $L(ar)$ mit höchstens $n + 1$ Zuständen.

☐ Es gibt einen NFA für $L(rar)$ mit höchstens $n + 1$ Zuständen.

☐ Es gibt einen NFA für $L(ra)$ mit höchstens $n + 1$ Zuständen.

c) Wieviele Zustände hat der minimale DFA für die Sprache $L((a^*b)^+)$? (Beachten Sie $r^+ = rr^*$.)

☐ 1

☐ 2

☐ 3

☐ ≥ 4

d) Sei G die Grammatik mit Produktionen $S \to aSb \mid Sb \mid b$. Wie viele Zustände hat der kleinste PDA M mit $L_\epsilon(M) = L(G)$? (Beachten Sie, dass $L_\epsilon(M)$ die Wörter sind, die M über leeren Keller akzeptiert.)

☐ 1

☐ 2

☐ 3

☐ ≥ 4

e) Sei G eine kontextfreie Grammatik mit 7 Produktionen. Welche Aussagen sind wahr?

☐ G hat höchstens 7 erreichbare Nichtterminale.

☐ G hat höchstens 7 nützliche Nichtterminale.

☐ G hat höchstens 7 erzeugende Nichtterminale.

Aufgabe 2 Rekursive Prozeduren (12 Punkte)

Sei $\Sigma = \{a, b\}$ das Alphabet.

a)* Geben Sie eine rekursive Prozedur $\mathrm{start}_{\mathrm{a}}(r)$ an, die für einen regulären Ausdruck r ausgibt, ob es ein Wort $w \in L(r)$ gibt, das mit a anfängt. Formal gilt also $\mathrm{start}_{\mathrm{a}}(r) := \big(L(r) \cap \{aw \mid w \in \Sigma^*\} \neq \emptyset\big)$.
Sie dürfen hierfür die Hilfsprozeduren $\mathrm{empty}(r) := \big(L(r) = \emptyset\big)$ und $\text{contains-}\varepsilon(r) := (\varepsilon \in L(r))$ verwenden.

0 1 2 3 4

b)* Für einen regulären Ausdruck r sei $\mathrm{contains}_{\mathrm{ab}}(r)$ die Aussage, dass ab ein Teilwort von **jedem** Wort $w \in L(r)$ ist. Formal soll also $\mathrm{contains}_{\mathrm{ab}}(r) := \big(L(r) \subseteq \Sigma^*\{ab\}\Sigma^*\big)$ gelten. Anhand folgender Äquivalenzen könnte man eine rekursive Prozedur für $\mathrm{contains}_{\mathrm{ab}}(r)$ konstruieren – allerdings sind zwei der Fälle falsch. Sei $x \in \Sigma$, und seien α, β reguläre Ausdrücke:

0 1 2 3 4

(1) $\mathrm{contains}_{\mathrm{ab}}(\emptyset) \Leftrightarrow \mathrm{true}$

(2) $\mathrm{contains}_{\mathrm{ab}}(\epsilon) \Leftrightarrow \mathrm{false}$

(3) $\mathrm{contains}_{\mathrm{ab}}(x) \Leftrightarrow \mathrm{false}$

(4) $\mathrm{contains}_{\mathrm{ab}}(\alpha^*) \Leftrightarrow \mathrm{contains}_{\mathrm{ab}}(\alpha)$

(5) $\mathrm{contains}_{\mathrm{ab}}(\alpha\beta) \Leftrightarrow \mathrm{contains}_{\mathrm{ab}}(\alpha) \vee \mathrm{contains}_{\mathrm{ab}}(\beta)$

(6) $\mathrm{contains}_{\mathrm{ab}}(\alpha \mid \beta) \Leftrightarrow \mathrm{contains}_{\mathrm{ab}}(\alpha) \wedge \mathrm{contains}_{\mathrm{ab}}(\beta)$

Welche zwei Fälle sind falsch? Begründen Sie dies jeweils anhand eines passenden Gegenbeispiels.

c)* Seien $\mathrm{AG}(r)$ und $\mathrm{AU}(r)$ die Aussagen, dass alle Wörter eines regulären Ausdrucks r gerade bzw. ungerade Länge haben. Formal gilt $\mathrm{AG}(r) := (L(r) \subseteq (\Sigma\Sigma)^*)$ und $\mathrm{AU}(r) := (L(r) \subseteq \Sigma(\Sigma\Sigma)^*)$. Beweisen Sie, dass $\mathrm{AG}(rs) \Leftrightarrow (\mathrm{AG}(r) \wedge \mathrm{AG}(s)) \vee (\mathrm{AU}(r) \wedge \mathrm{AU}(s))$ gilt, für reguläre Ausdrücke r, s, die das Symbol $\emptyset$ **nicht enthalten**.

0 1 2 3 4

„$\Leftarrow$“:

„$\Rightarrow$“:

Aufgabe 3 Chomsky-Normalform (10 Punkte)

In dieser Aufgabe geht es darum, eine kontextfreie Grammatik (CFG) in Chomsky-Normalform (CNF) zu konvertieren. Wir führen allerdings jeden Schritt einzeln aus, und jeweils auf einer anderen Grammatik – Sie können also die Aufgabenteile unabhängig voneinander bearbeiten.
Eine Grammatik $G = (V, \Sigma, P, S)$ ist in CNF, wenn jede Produktion $(X \to \alpha) \in P$, mit $X \in V$ und $\alpha \in (\Sigma \cup V)^*$, folgende Bedingungen erfüllt:

(1) $\alpha \in \Sigma \cup V^*$; Terminale dürfen nur in Produktionen der Länge 1 erzeugt werden.

(2) $|\alpha| \leq 2$; jede Produktion hat höchstens Länge 2.

(3) $\alpha \neq \varepsilon$; es gibt keine ε-Produktionen.

(4) $\alpha \notin V$; es gibt keine Kettenproduktionen.

Achtung: Die Teilaufgaben fragen jeweils spezifisch nach dem Ergebnis, das sich durch die Ausführung des Algorithmus aus der Vorlesung ergibt, nicht nach einer beliebigen äquivalenten CFG, die den Bedingungen genügt. Details, wie etwa die Namen der Variablen oder die Reihenfolge, in der Produktionen betrachtet werden, können Sie frei wählen.

0
1
2

a)* *Entfernen von Terminalen in langen Produktionen.* Die CFG G_a ist gegeben durch folgende Produktionen:

$$S \to c \mid X \mid aSX \mid SdS \mid dd \qquad X \to a \mid aX$$

Führen Sie den ersten Schritt des Algorithmus zur Überführung in CNF aus und geben Sie die Produktionen einer CFG G'_a an, so dass $L(G_a) = L(G'_a)$ gilt und G'_a Bedingung (1) erfüllt.

(Platz für Zwischenschritte / Erklärungen)

Fertige Grammatik G'_a:

0
1
2

b)* *Entfernen langer Produktionen.* Die CFG G_b ist gegeben durch die folgenden Produktionen:

$$S \to c \mid ABSA \qquad A \to a \mid AAA \qquad B \to d$$

Führen Sie den zweiten Schritt des Algorithmus zur Überführung in CNF aus und geben Sie die Produktionen einer CFG G'_b an, so dass $L(G_b) = L(G'_b)$ und G'_b Bedingung (1) und (2) erfüllt.

(Platz für Zwischenschritte / Erklärungen)

Fertige Grammatik G'_b:

c)* *Entfernen von ε-Produktionen.* Die CFG G_c ist gegeben durch folgende Produktionen:

0
1
2
3

$$\begin{array}{ll} S \to AD \mid D & C \to BA \\ A \to a \mid BB & D \to d \mid SS \mid DB \\ B \to b \mid \varepsilon & E \to e \mid CA \end{array}$$

Führen Sie den dritten Schritt des Algorithmus zur Überführung in CNF aus. Geben Sie die Produktionen einer CFG G'_c an, so dass $L(G_c) = L(G'_c)$ und G'_c Bedingungen (1), (2) und (3) erfüllt.

(Platz für Zwischenschritte / Erklärungen)

Fertige Grammatik G'_c:

d)* *Entfernen von Kettenproduktionen.* Die CFG G_d ist gegeben durch die Produktionen:

0
1
2
3

$$\begin{array}{ll} S \to A \mid D & C \to c \mid B \mid D \\ A \to a \mid S \mid AA & D \to d \\ B \to b \mid B \mid BC & \end{array}$$

Führen Sie den vierten Schritt des Algorithmus zur Überführung in CNF aus und geben Sie die Produktionen einer CFG G'_d in CNF an, so dass $L(G_d) = L(G'_d)$ gilt.

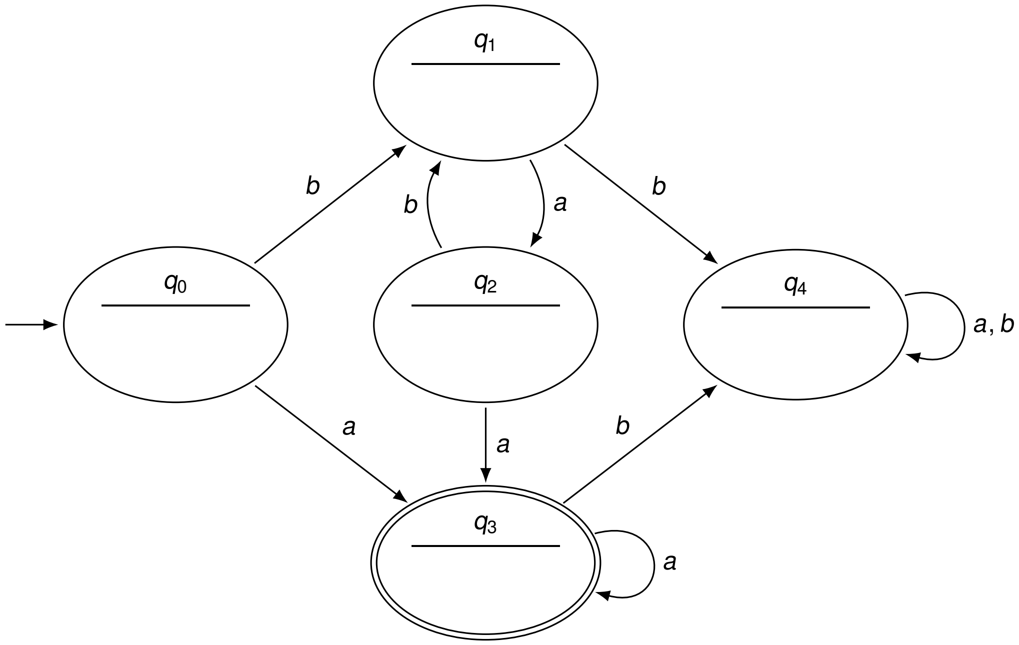
(Platz für Zwischenschritte / Erklärungen)

Fertige Grammatik G'_d:

Aufgabe 4 **Residualsprachen (10 Punkte)**

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA und $q \in Q$ ein Zustand von M. Die Menge $L_M(q) := \{w \in \Sigma^* : \hat{\delta}(q, w) \in F\}$ sind genau die Wörter, die M akzeptiert, wenn M in Zustand $q \in Q$ startet. Es gilt, dass $L_M(q)$ immer eine Residualsprache von $L(M)$ ist.

a)* Der folgende DFA M akzeptiert die Sprache $L((ba)^*aa^*)$. Beschriften Sie jeden Zustand q mit einem regulären Ausdruck für die Sprache $L_M(q)$.



b)* Sei N ein DFA mit Zuständen $\{p_1, p_2, p_3, p_4\}$. Es gilt $L_N(p_i) = L(r_i)$, für $i \in \{1, 2, 3, 4\}$ und reguläre Ausdrücke $r_1 = a(ba)^*$, $r_2 = (ba)^*$, $r_3 = (ab)^*a$, $r_4 = \emptyset$. Alle Zustände von N sind erreichbar. Ist N minimal? Begründen Sie Ihre Antwort.

☐ ja ☐ nein
Begründung:

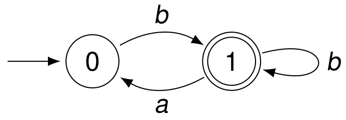
c)* Ist der DFA M aus der Teilaufgabe a) minimal? Begründen Sie ihre Antwort.

☐ ja ☐ nein
Begründung:

Aufgabe 5 Ein a geht noch (10 Punkte)

Sei $\Sigma := \{a, b\}$. Für eine Sprache $L \subseteq \Sigma^*$ bezeichnet L^{Υ} die Sprache, die man erhält, wenn man den Wörtern aus L an einer beliebigen Stelle ein a einfügt. Formal gilt also $L^{\Upsilon} = \{uav : uv \in L,\ u, v \in \Sigma^*\}$. Wir erhalten z.B. $\{a, ab\}^{\Upsilon} = \{aa, aab, aba\}$ und $(\{ab\}^*)^{\Upsilon} = L((ab)^*(a \mid aab)(ab)^*)$. Beachten Sie insbesondere, dass das Einfügen nicht optional ist.

a)* Sei M der folgende NFA. Geben Sie einen NFA M' mit $L(M') = L(M)^{\Upsilon}$ und höchstens 4 Zuständen an.



b)* Sei $M = (Q, \Sigma, \delta, q_0, F)$ nun ein beliebiger NFA. Konstruieren Sie einen NFA $M' = (Q', \Sigma, \delta', q_0', F')$ mit $L(M') = L(M)^{\Upsilon}$ und $|Q'| \leq 2|Q|$. Schildern Sie insbesondere Ihre Idee in natürlicher Sprache und geben Sie Q', δ', q_0', und F' präzise an.

c)* Geben Sie $L_1, L_2 \subseteq \Sigma^*$ mit $L_1 \neq L_2$ und $L_1^{\Upsilon} = L_2^{\Upsilon}$ an. **Hinweis:** Es gibt solche L_1, L_2 mit $|L_1|, |L_2| \leq 3$.

Aufgabe 6 **Sensitivität (18 Punkte)**

Sei $\Sigma = \{a, b\}$. Gegeben sind die folgenden vier Sprachen:

$$\begin{aligned} L_1 =& \{a^n b^n a^m b^m \mid n, m \in \mathbb{N}\} \\ L_2 =& \{a^n b^m a^m b^n \mid n, m \in \mathbb{N}\} \\ L_3 =& \{a^n b^m a^n b^m \mid n, m \in \mathbb{N}\} \\ L_4 =& \{a^n b^m b^n a^m \mid n, m \in \mathbb{N}\} \end{aligned}$$

0 1 2

a)* Genau eine dieser Sprachen ist nicht kontextfrei. Welche? (Ohne Begründung.)

Die nicht kontextfreie Sprache ist ____.

0 1 2 3 4 5 6

b)* Geben Sie für die drei anderen Sprachen jeweils eine kontextfreie Grammatik (CFG) mit höchstens 5 Produktionen an.

Die Produktionen der CFG für ____ sind:	Die Produktionen der CFG für ____ sind:	Die Produktionen der CFG für ____ sind:

0 1 2

c)* Welche der Sprachen L_1, L_2, L_3, L_4 sind regulär? (Ohne Begründung.)

d)* Zeigen Sie von einer der Sprachen L_1, L_2, L_3, L_4, dass sie nicht regulär ist, indem Sie das Pumping-Lemma für reguläre Sprachen verwenden.

0
1
2
3
4

e)* Wählen Sie eine Sprache $L \in \{L_1, L_2, L_3, L_4\}$, die nicht regulär ist, und die Sie in Teilaufgabe d) nicht gewählt haben. Zeigen Sie, dass L unendlich viele Residualsprachen hat und somit nicht regulär ist. Geben Sie insbesondere Wörter $w_1, w_2, ... \in \Sigma^*$ an, sodass $L^{w_i} \neq L^{w_j}$ gilt, für alle $i, j \in \mathbb{N}$ mit $i \neq j$.

0
1
2
3
4

Aufgabe 7 **Berechenbarkeit und Komplexität (10 Punkte)**

Für die folgenden Fragen ist eine Begründung nicht gefordert. Sie erhalten die Punkte auf eine Teilaufgabe genau dann, wenn Sie alle Antwortmöglichkeiten korrekt angekreuzt haben. Es ist immer mindestens eine Antwortmöglichkeit richtig. Jede Teilaufgabe bringt 2 Punkte.

Kreuzen Sie richtige Antworten an ☒

Kreuze können durch vollständiges Ausfüllen gestrichen werden ■

Gestrichene Antworten können durch nebenstehende Markierung erneut angekreuzt werden ×■

In dieser Aufgabe verwenden wir durchgehend das Alphabet $\Sigma := \{0, 1\}$.

a) Welche der folgenden Sprachen sind entscheidbar?

☐ $\{w \in \Sigma^* : L(M_w) = \{vv : v \in \Sigma^*\}\}$

☐ $\{ww : w \in \Sigma^*\}$

b) Für welche der folgenden Sprachen gibt es ein $v \in \Sigma^*$, sodass sie entscheidbar sind?
Beachten Sie, dass $\varphi_u : \Sigma^* \to \Sigma^*$ die Funktion ist, die von M_u berechnet wird.

☐ $\{w \in \Sigma^* : \varphi_v(v) = 0\}$

☐ $\{w \in \Sigma^* : \varphi_v(w) = 0\}$

☐ $\{w \in \Sigma^* : \varphi_w(v) = 0\}$

c) Sei $L \subseteq \Sigma^*$ endlich. Welche der folgenden Sprachen sind semi-entscheidbar?

☐ $\{w \in \Sigma^* : L(M_w) = \Sigma^*\} \cap L$

☐ $\{w \in \Sigma^* : L(M_w) \cap L = L\}$

☐ $\{w \in \Sigma^* : L(M_w) \cap L = \Sigma^*\}$

d) Gibt es eine reguläre Sprache $L \in \mathrm{NP}$?

☐ Ja

☐ Nein

☐ Unbekannt (hängt von $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ ab)

e) Bei dem NP-vollständigen Problem SAT geht es darum, von einer aussagenlogische Formel F zu überprüfen, ob sie erfüllbar ist. Die Formel F besteht dabei aus Variablen $x_1, ..., x_k$, die beliebig mit $\wedge, \vee, \neg$ (also logischer Konjunktion, Disjunktion und Negierung) verknüpft werden. Ein Beispiel für eine solche Formel ist $(x_1 \wedge x_2) \vee \neg(x_2 \vee \neg x_3)$.
Welche der folgenden Varianten von SAT sind NP-vollständig, unter der Annahme $\mathrm{P} \neq \mathrm{NP}$?

☐ Die Formel F enhält $\neg$ nicht.

☐ Die Formel F enthält $\wedge$ nicht.

☐ Jede Variable x_i kommt nur einmal in F vor.

Aufgabe 8 **Reduktion (12 Punkte)**

Sei $\Sigma := \{a, b\}$. Für ein Wort $w \in \Sigma^*$ bezeichnen wir mit w^S das Wort, das man erhält, wenn man in w jedes a durch ein b, und jedes b durch ein a ersetzt. Z.B. gilt also $\varepsilon^S = \varepsilon$, $(baa)^S = abb$ und $(abbab)^S = baaba$. Für eine Sprache $L \subseteq \Sigma^*$ definieren wir dann $L^S := \{w^S : w \in L\}$.

a)* Sei H die Grammatik über dem Alphabet Σ mit den Produktionen $S \to aSbb \mid SS \mid ba$. Geben Sie eine kontextfreie Grammatik (CFG) H' an, sodass $L(H') = L(H)^S$ und H' höchstens 3 Produktionen hat. Eine Begründung ist nicht erforderlich.

0
1

Wir betrachten nun die beiden folgenden Probleme, jeweils für eine CFG G über Σ.

$\langle 1 \rangle$ Ist $L(G) = \Sigma^*$?

$\langle 2 \rangle$ Ist $L(G) = L(G)^S$?

b)* Ist $\overline{\langle 2 \rangle}$ semi-entscheidbar (also $\langle 2 \rangle$ co-semi-entscheidbar)? Begründen Sie Ihre Antwort.

0
1
2

☐ ja ☐ nein

Begründung:

Wir wissen bereits, dass $\langle 1 \rangle$ unentscheidbar ist. Unser Ziel ist es, zu zeigen, dass auch $\langle 2 \rangle$ unentscheidbar ist, indem wir $\langle 1 \rangle$ auf $\langle 2 \rangle$ reduzieren, also $\langle 1 \rangle \leq \langle 2 \rangle$ zeigen.

c)* Geben Sie eine Reduktionsfunktion für $\langle 1 \rangle \leq \langle 2 \rangle$ an. Beschreiben Sie also ein Verfahren, das eine CFG $G = (V, \Sigma, P, S)$ zu einer CFG $G' = (V', \Sigma, P', S')$ konvertiert, sodass $L(G) = \Sigma^*$ gilt, genau dann wenn $L(G') = L(G')^S$ gilt.

0
1
2
3
4
5

Führen Sie Ihr Verfahren zur Veranschaulichung auf der Grammatik G mit Produktionen $S \to aSbS \mid ba$ aus, und geben Sie die resultierende Grammatik G' an.

0
1
2
3
4

d) Beweisen Sie, dass ihr Verfahren aus Teilaufgabe c) korrekt ist, dass also $L(G) = \Sigma^* \Leftrightarrow L(G') = L(G')^S$ gilt.

„$\Rightarrow$“:

„$\Leftarrow$“:

Aufgabe 9 Sequenz (8 Punkte)

Sei $\Sigma := \{0, 1\}$ und $L \subseteq \Sigma^*$ eine Sprache. Wir definieren eine Folge an Wörtern $w_0, w_1, w_2, ... \in \Sigma^*$ wie folgt:

$$w_0 := \varepsilon \qquad w_{n+1} := \begin{cases} w_n 1 & \text{wenn } w_n \in L \\ w_n 0 & \text{wenn } w_n \notin L \end{cases} \qquad \text{für } n \in \mathbb{N}$$

Wir schreiben $\sigma(L) := \{w_i : i \in \mathbb{N}\}$ für die Sprache, die genau die Wörter der Folge enthält.

a)* Geben Sie einen regulären Ausdruck für die Sprache $\sigma(\Sigma^* 0)$ an. (Ohne Begründung.)

0
1
2

b)* Zeigen Sie: Wenn L regulär ist, dann ist auch $\sigma(L)$ regulär.
Hinweis: Für eine Aussage A können Sie die Notation $I(A)$ verwenden; es sei $I(A) := 1$, falls A gilt, und $I(A) := 0$ sonst. Obige Definition lässt sich z.B. als $w_{n+1} = w_n I(w_n \in L)$ schreiben.

0
1
2
3
4
5
6

Zusätzlicher Platz für Lösungen. Markieren Sie deutlich die Zuordnung zur jeweiligen Teilaufgabe. Vergessen Sie nicht, ungültige Lösungen zu streichen.