Lehrstuhl für Theoretische Informatik
Fakultät für Informatik
Technische Universität München

Esolution
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 / Retake	**Datum:**	Dienstag, 4. Oktober 2022
Prüfer:	Prof. Dr. Dr. h.c. Javier Esparza	**Uhrzeit:**	11:15 – 14:15

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

☐ $|A|^2 = |A^2|$ Gegenbeispiel: $A = \{a, aa\}$

☒ $\{uv : u \in A, v \in B\}^2 = \{uv : u, v \in AB\}$ Beides ist $(AB)^2$.

☒ $A \subseteq (A \cup B^*)^2$ Man beachte $\varepsilon \in B^*$

b) Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA. Wir bezeichnen einen Zustand $q \in Q$ als *Fangzustand*, wenn von q aus kein Finalzustand erreicht werden kann, also $\hat{\delta}(q, w) \notin F$ für alle $w \in \Sigma^*$. Für welche der folgenden regulären Ausdrücke r gibt es einen DFA M mit $L(M) = L(r)$, sodass M *keinen* Fangzustand hat?

☐ a^*b^* $\hat{\delta}(q_0, ba)$ ist immer ein Fangzustand.

☐ $bab^*(a \mid b)^*$ $\hat{\delta}(q_0, a)$ ist immer ein Fangzustand.

☒ $(a \mid b)^*bab^*$ Aus jedem Zustand wird durch Einlesen von ba akzeptiert.

c) Sei $M = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$ ein PDA mit n Zuständen (also $n = |Q|$) und k Kellersymbolen (also $k = |\Gamma|$). Wir konvertieren M zu einer kontextfreien Grammatik G, indem wir das aus der Vorlesung bekannte Verfahren verwenden. Wie viele Nichtterminalsymbole hat G höchstens? Genau eine Antwort ist richtig.

☐ $k2^n + 1$

☐ $nk^2 + 1$

☐ $(k + n)^2 + 1$

☐ 1

☒ $n^2k + 1$ $V = Q \times \Gamma \times Q \cup \{S\}$ (s. Folie 200)

d) Welche der folgenden Sprachen sind kontextfrei?

☒ $\bigcup_{n=0}^{\infty}\{aa, ab\}^n\{ab, bb\}^n$ $S \to ASB \mid \varepsilon$, $A \to aa \mid ab$, $B \to ab \mid bb$

☐ $\{a^nb^na^n : n \in \mathbb{N}\}$ Gleiches Argument wie für $\{a^nb^nc^n : n \in \mathbb{N}\}$

☒ $\{a^n : n \text{ gerade}\}$ $S \to aaS \mid \varepsilon$; die Sprache ist sogar regulär.

e) Für einen NFA oder DFA $M = (Q, \Sigma, \delta, q_0, F)$ und Zustand $q \in Q$ schreiben wir $L_M(q)$ für die Wörter, die M startend in Zustand q akzeptiert. Formal gilt also $L_M(q) = L(M')$, mit $M' := (Q, \Sigma, \delta, q, F)$. Außerdem nennen wir M *verbunden*, wenn alle Zustände von M erreichbar sind. Welche der folgenden Aussagen sind wahr?

☐ Es gibt einen verbundenen NFA M mit Zustand q und $L(M) = \{aa, ba\}^*$ und $L_M(q) = \{ab\}^*$.
Dann würde M ein Wort, das auf b endet, akzeptieren.

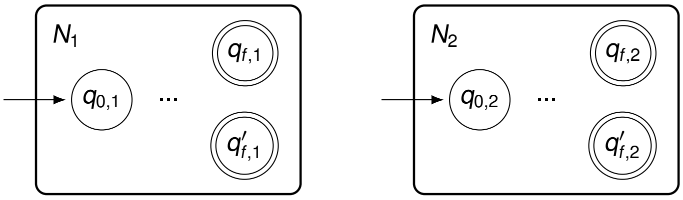
☒ Es gibt einen verbundenen NFA M mit Zustand q und $L(M) = \{aa, ba\}^*$ und $L_M(q) = \{aa\}^*$.
Da $\{aa\}^* \subseteq L(M)$ kann man immer einen solchen Zustand zu einem NFA für $L(M)$ hinzufügen.

☒ Es gibt einen verbundenen DFA M mit Zustand q und $L(M) = \{aa, ba\}^*$ und $L_M(q) = \{a\}\{aa, ba\}^*$.
$\{a\}\{aa, ba\}^*$ ist eine Residualsprache, die Aussage gilt sogar für alle M.

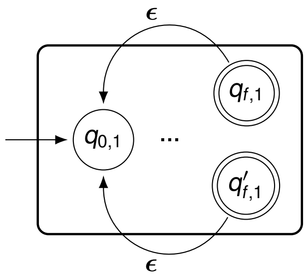
☐ Es gibt einen verbundenen DFA M mit Zustand q und $L(M) = \{aa, ba\}^*$ und $L_M(q) = \{a\}^*$.
$\{a\}^*$ ist keine Residualsprache von $L(M)$.

Aufgabe 2 **REtoNFA (12 Punkte)**

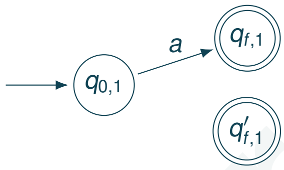
a)* Seien N_1, N_2 NFAs mit jeweils genau zwei Endzuständen. Im Folgenden werden N_1 und N_2 durch Kästen dargestellt, in denen nur der Initial- und die Endzustände gezeichnet sind. Es kann also beliebige weitere Zustände geben, und beliebige aus- und eingehende Transitionen. Ähnlich zur Vorlesung betrachten wir Konstruktionen, die diese Automaten leicht anpassen. Ausschließlich Sichtbares wird hierbei modifiziert, die Transitionen innerhalb eines Kastens bleiben unverändert. Bestimmen Sie, welche der folgenden Konstruktionen korrekt sind. Falls die Konstruktion falsch ist, geben Sie zusätzlich ein Gegenbeispiel an.



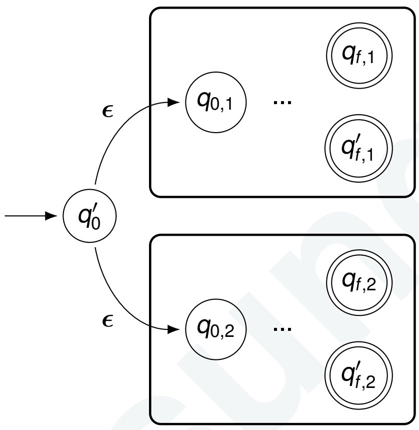
1. Konstruktion für N' mit $L(N') = L(N_1)^*$.



☐ Korrekt ☒ Falsch, Gegenbeispiel:

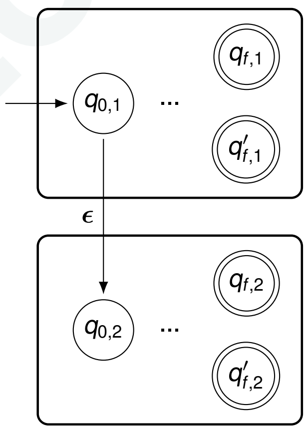


2. Konstruktion für N' mit $L(N') = L(N_1) \cup L(N_2)$.

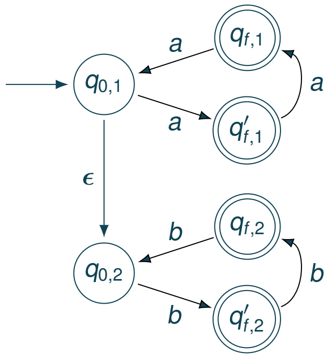


☒ Korrekt ☐ Falsch, Gegenbeispiel:

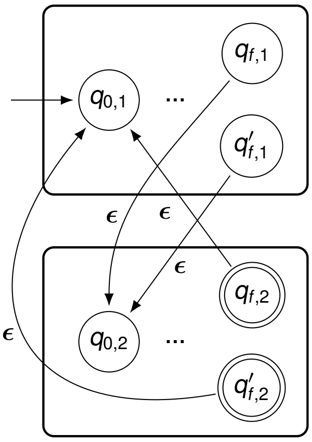
3. Konstruktion für N' mit $L(N') = L(N_1) \cup L(N_2)$.



☐ Korrekt ☒ Falsch, Gegenbeispiel:



4. Konstruktion für N' mit $L(N') = L(N_1)\,(L(N_2)L(N_1))^*L(N_2)$.

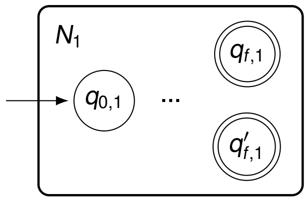


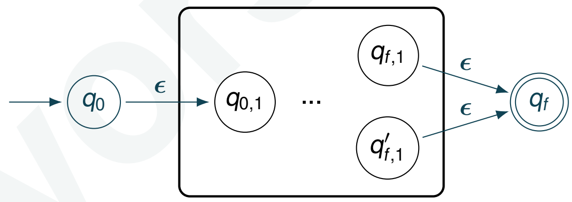
☒ Korrekt ☐ Falsch, Gegenbeispiel:

Sei M ein beliebiger endlicher Automat, also ein ϵ-NFA, NFA, oder DFA. Wir bezeichnen M als *klausurig*, wenn (1) M höchstens einen Finalzustand hat, und (2) der Startzustand von M keine eingehenden Kanten hat.

0 1

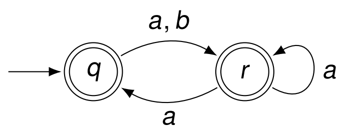
b)* Modifizieren Sie einen ϵ-NFA N_1, der genau zwei Finalzustände hat, sodass er klausurig ist (ohne die akzeptierte Sprache zu verändern).



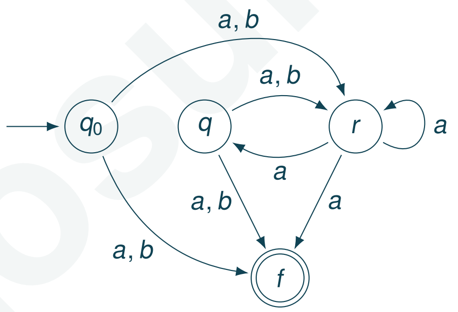


0 1 2 3 4 5

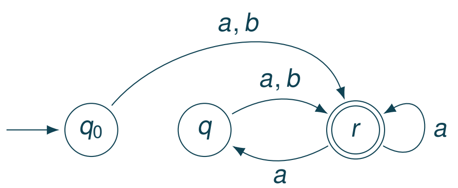
c)* Sei M der rechts abgebildete NFA. Gibt es einen klausurigen NFA / DFA M' mit $L(M') = L(M) \setminus \{\varepsilon\}$? Falls ja, geben Sie einen solchen Automaten mit höchstens 4 Zuständen an, falls nein, beweisen Sie Ihre Aussage.



NFA: ☒ Ja ☐ Nein



Alternative Lösung:



DFA: ☐ Ja ☒ Nein

Angenommen es gäbe einen solchen DFA $M' = (Q, \{a, b\}, \delta, q_0, F)$. Da $a, aa \in L(M)$, muss $\delta(q_0, a), \delta(q_0, aa) \in F$ gelten. Da M' nur einen Finalzustand hat, folgt also $\delta(q_0, a) = \delta(q_0, aa)$. Dann kann aber nicht $ab \notin L(M'), aab \in L(M')$ gelten, ein Widerspruch zu $L(M') = L(M) \setminus \{\varepsilon\}$.

Aufgabe 3 CYK Nützlich (13 Punkte)

a)* Sei G eine Grammatik mit folgenden Produktionen.

$$S \to XX \mid a \mid b \qquad X \to XX \mid YY \mid b \qquad Y \to YY \mid SX$$

Verwenden Sie den aus der Vorlesung bekannten CYK-Algorithmus, um zu entscheiden, ob *bbab* in $L(G)$ enthalten ist. Füllen Sie dazu nachfolgende Tabelle entsprechend aus.

1,4: X, Y			
1,3: —	2,4: —		
1,2: S, X, Y	2,3: —	3,4: Y	
1,1: S, X	2,2: S, X	3,3: S	4,4: S, X
b	*b*	*a*	*b*

b) Bestimmen Sie das längste Präfix von *bbab*, das in $L(G)$ enthalten ist. Begründen Sie Ihre Antwort *kurz* indem Sie auf die Tabelle aus (a) verweisen.

bb, da (1,2) das Symbol S enthält und (1,3) und (1,4) nicht.

c)* Sei H eine Grammatik mit den Produktionen

$$S \to XW \mid a \qquad X \to b \qquad Y \to SZ \mid SX \qquad Z \to WY \qquad W \to XW$$

Bestimmen Sie die erreichbaren, erzeugenden, und nützlichen Nichtterminalsymbole von H.

Erreichbar: $\{S, X, W\}$ Erzeugend: $\{S, X, Y\}$ Nützlich: $\{S\}$

d)* Sei $G = (V, \Sigma, P, S)$ eine beliebige kontextfreie Grammatik in CNF. Ist das Problem, ob $L(G)$ unendlich ist, entscheidbar? Falls ja, beschreiben Sie ein entsprechendes Verfahren, falls nein, widerlegen Sie es.

☒ Ja ☐ Nein Verfahren / Beweis:

Zunächst formen wir G so um, dass G nur nützliche Symbole enthält. Damit $L(G)$ unendlich ist, muss es eine Variable $X \in V$ geben, die sich selbst erzeugt. Formal muss es also $\alpha, \beta \in (V \cup \Sigma)^*$ geben, mit $X \to_G^* \alpha X \beta$ gilt.

Dies bestimmen wir, indem wir jedes Paar an Variablen (X, Y) ermitteln, sodass Y aus X erzeugt werden kann. Dazu starten wir mit $M := \{(X, Y), (X, Z) : (X \to YZ) \in P\}$. Für jedes $(X, Y), (Y, Z) \in M$ fügen wir anschließend (X, Z) hinzu, und wiederholen den Prozess, bis wir nichts weiter hinzufügen können. Falls es dann ein $X \in V$ mit $(X, X) \in M$ gibt, akzeptieren wir, sonst nicht.

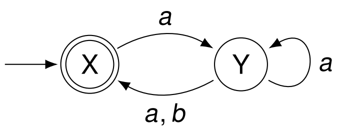
Alternative Lösung: Wir verwenden das Pumping-Lemma für kontextfreie Sprachen. Sei $n = 2^{|V|}$ die PL-Zahl. Wenn es ein Wort in $L(G)$ mit Länge mindestens n gibt, dann muss $L(G)$ unendlich sein. Die Umkehrung gilt offensichtlich auch. Es genügt also $L(G) \cap \Sigma^n\Sigma^* \neq \emptyset$ zu überprüfen, nach H7.2 finden wir eine CFG G' für $L(G) \cap \Sigma^n\Sigma^*$, und $L(G') \neq \emptyset$ gdw. es eine nützliche Variable in G' gibt.

Aufgabe 4 **Ardens Lemma (10 Punkte)**

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

0 1 2 3 4 5 6

a)* Wir betrachten den folgenden NFA. Seien X und Y die Sprachen, die der Automat von den jeweiligen Zuständen aus akzeptiert. Stellen Sie das aus diesem Automaten resultierende Gleichungssystem auf, indem Sie die Lücken mit den entsprechenden regulären Ausdrücken füllen.
Um Ihre Gleichungen anzugeben, müssen Sie sie zunächst nach Variablen gruppieren; z.B. können Sie $X \equiv a(bY \mid a) \mid bbY$ zu $X \equiv (ab \mid bb)Y \mid aa$ umformen und dann als $X \equiv (\emptyset)X \mid (ab \mid bb)Y \mid (aa)$ aufschreiben.
Insbesondere dürfen die regulären Ausdrücke, die Sie in die Lücken schreiben, X und Y nicht enthalten.



$$X \equiv (\ \emptyset\ )X \mid (\ a\ )Y \mid (\ \epsilon\ )$$

$$Y \equiv (\ a \mid b\ )X \mid (\ a\ )Y \mid (\ \emptyset\ )$$

Lösen Sie das Gleichungssystem, indem Sie zunächst die Gleichung für X in die Gleichung für Y einsetzen.

$$Y \equiv (\ aa \mid ba \mid a\ )Y \mid (\ a \mid b\ )$$

Bestimmen Sie nun reguläre Ausdrücke für Y und X.

$$Y \equiv (\ (aa \mid ba \mid a)^*(a \mid b)\ )$$

$$X \equiv (\ a(aa \mid ba \mid a)^*(a \mid b) \mid \epsilon\ )$$

0 1 2 3 4

b)* Geben Sie für die folgenden Gleichungen an, ob sie 0, 1 oder mindestens 2 Lösungen haben. Lösungen können reguläre, oder auch nicht reguläre Sprachen sein. Falls 1, geben Sie eine Lösung an, falls mindestens 2, geben Sie zwei Lösungen an. Falls 0, begründen Sie dies kurz.

1. $X \equiv babX \mid a$
2. $X \equiv a^*X \mid bab$
3. $X \equiv aXb \mid \epsilon$
4. $aX \equiv XX \mid b$

1: ☐ 0 ☒ 1 ☐ ≥ 2
Lösung(en) / Begründung: $L((bab)^*a)$
Ardens Lemma ist anwendbar.

2: ☐ 0 ☐ 1 ☒ ≥ 2
Lösung(en) / Begründung: $L(a^*bab)$, Σ^*
Anm.: Die Gleichung ist äquivalent zu $\{a\}^+X \cup \{bab\} \subseteq X$. Die kleinste Lösung kann man über Ardens Lemma finden, jede größere Sprache ist auch eine Lösung.

3: ☐ 0 ☒ 1 ☐ ≥ 2
Lösung(en) / Begründung: $\{a^n b^n : n \in \mathbb{N}\}$

4: ☒ 0 ☐ 1 ☐ ≥ 2
Lösung(en) / Begründung: Alle Wörter in aX fangen mit einem a an, aber auf der rechten Seite ist b enthalten, was nicht mit a anfängt. Also kann es keine Lösung geben.

Aufgabe 5 **Präfixissimo (10 Punkte)**

Sei Σ ein Alphabet. Für eine Sprache $L \subseteq \Sigma^*$ bezeichnet $L^{\prec}$ die Sprache, die man erhält, wenn man von den Wörtern aus L ein beliebiges Präfix bildet. Formal gilt also $L^{\prec} = \{u : uv \in L,\ u, v \in \Sigma^*\}$. Wir erhalten z.B. $\{babb\}^{\prec} = \{\varepsilon, b, ba, bab, babb\}$ und $(\{ab\}^*)^{\prec} = L((ab)^*(a \mid \epsilon))$.

a)* Sei $G = (V, \Sigma, P, S)$ eine kontextfreie Grammatik (CFG) in Chomsky-Normalform, mit $\Sigma = \{a, b\}$. (Es hat also jede Produktion von G die Form $X \to YZ$ oder $X \to c$, für $X, Y, Z \in V$ und $c \in \Sigma$.)
Wir wollen nun eine CFG $G^{\prec} = (V^{\prec}, \Sigma, P^{\prec}, S^{\prec})$ mit $L(G^{\prec}) = L(G)^{\prec}$ konstruieren. Dazu führen wir eine neue Variable X' für jedes $X \in V$ ein. Wir wollen also die Variablen $V^{\prec} = V \cup \{X' : X \in V\}$ verwenden.
Die Variablen in V sollen ihre ursprüngliche Bedeutung beibehalten, es soll also $L_{G^{\prec}}(X) = L_G(X)$ gelten, für $X \in V$. Was soll für $L_{G^{\prec}}(X')$ gelten, also die Sprache der Wörter, die sich in $G^{\prec}$ aus X' ableiten lassen?

$L_{G^{\prec}}(X') = L_G(X)^{\prec}$

Nun konstruieren wir die Produktionen $P^{\prec}$ von $G^{\prec}$. Beachten Sie, dass $G^{\prec}$ nicht in CNF sein muss, insbesondere ist es auch erlaubt, dass $G^{\prec}$ ε-Produktionen enthält.
Für jede Produktion der Form $(X \to YZ) \in P$, mit $X, Y, Z \in V$, fügen wir folgende Produktion(en) hinzu:

$X' \to YZ' \mid Y'$

Für jede Produktion der Form $(X \to c) \in P$, mit $X \in V, c \in \Sigma$ fügen wir folgende Produktion(en) hinzu:

$X' \to c \mid \varepsilon$

Muss $P^{\prec}$ noch weitere Produktionen, außer den oben von Ihnen angegebenen, enthalten? Falls ja, können Sie die hier notieren.

Wir fügen alle Produktionen in P hinzu.

(*Anm.:* Es ist auch möglich, diese Produktionen bereits oben anzugeben.)

Geben Sie schließlich das Startsymbol von $G^{\prec}$ an.

$S^{\prec} := S'$

b)* Sei G eine CFG. Gibt es eine CFG G', die genau die Wörter aus $L(G)$ enthält, in denen gleich viele a und b vorkommen? Falls ja, begründen Sie dies kurz, falls nein, geben Sie ein Gegenbeispiel an.

☐ ja ☒ nein

Begründung / Gegenbeispiel: Wir wählen das Alphabet $\Sigma = \{a, b, c\}$ und eine beliebige CFG G mit $L(G) = \{a^n b^m c^m : n, m \in \mathbb{N}\}$. (Etwa $S \to aS \mid T$, $T \to bTc \mid \varepsilon$.) Dann gilt $L(G') = \{a^n b^n c^n : n \in \mathbb{N}\}$, aber von dieser Sprache wissen wir, dass sie nicht kontextfrei ist (Beispiel 4.30).

Aufgabe 6 **Irregularität (12 Punkte)**

0 1 2 3 4 5

a)* Sei $L_1 := \{a^n b^m : n, m \in \mathbb{N}, n > m\}$. Zeigen Sie, dass L_1 nicht regulär ist, indem Sie das Pumping-Lemma für reguläre Sprachen verwenden.

Wir führen einen Widerspruchsbeweis und nehmen an, dass L_1 regulär wäre. Dann gilt die Aussage des Pumping-Lemma (PL) für reguläre Sprachen. Sei $n \in \mathbb{N}$ die PL-Zahl. Nun wählen wir das Wort $z := a^{n+1} b^n \in L_1$. Nach PL existiert eine Zerlegung $z = uvw$. Es gilt $|uv| \leq n$ und $v \neq \varepsilon$, wir können also $v = a^i$ schreiben, für ein $i > 0$. Nun gilt auch $uv^0 w \in L_1$, aber $uw = a^{n+1-i} b^n \notin L_1$, da $n + 1 - i \leq n$. Dies ist ein Widerspruch, L_1 kann somit nicht regulär sein.

0 1 2 3 4

b)* Sei $\Sigma := \{a, b\}$ und $L_2 := \{a^n b^m : n, m \in \mathbb{N}, n < m\}$. Geben Sie Wörter $w_0, w_1, w_2, ... \in \Sigma^*$ an, sodass $L_2^{w_i} \neq L_2^{w_j}$ gilt, für alle $i, j \in \mathbb{N}$ mit $i \neq j$.

$w_n := a^n b$ für $n \in \mathbb{N}$

Bestimmen Sie $L_2^{w_n}$, für ein beliebiges $n \in \mathbb{N}$:

$L_2^{w_n} := \{b^{n+i} : i \in \mathbb{N}\}$

Seien nun $i, j \in \mathbb{N}$ beliebig, mit $i < j$. Zeigen Sie $L_2^{w_i} \neq L_2^{w_j}$, indem Sie ein Wort $u \in \Sigma^*$ angeben, das in genau einer dieser Sprachen enthalten ist.

$u := b^i$

0 1 2 3

c)* Zeigen oder widerlegen Sie: Es gibt eine Sprache L, sodass L unendlich viele verschiedene Residualsprachen hat, und jede Residualsprache von L unendlich viele Wörter enthält.

☒ Wahr ☐ Falsch

Beweis: Eine solche Sprache existiert. Sei $L := \{ww : w \in \Sigma^*\}$ mit $\Sigma := \{a, b\}$. Da L nicht regulär ist, hat L unendlich viele verschiedene Residualsprachen.
Sei nun $w \in \Sigma^*$ beliebig. Dann gilt $\{w^{2i+1} : i \in \mathbb{N}\} \subseteq L^w$, also ist L^w unendlich.

Anmerkung: Alle nicht-regulären Sprachen L, sodass jedes Wort $w \in \Sigma^*$ ein Präfix eines Wortes in L ist, funktionieren. Dazu gehören etwa $\{w \in \{a, b\}^* : |w|_a = |w|_b\}$ oder $\{ww^R : w \in \Sigma^*\}$. Hingegen sind $\{a^n b^n : n \in \mathbb{N}\}$ und die Sprache der balancierten Klammerausdrücke keine Beispiele, da $L^b = \emptyset$.

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?

- ☒ $\{w \in \Sigma^* : L(M_w) \text{ ist semi-entscheidbar}\}$ $L(M_w)$ ist immer semi-entscheidbar.
- ☐ $\{w \in \Sigma^* : L(M_w) = \emptyset\}$ Satz von Rice.
- ☐ $\{w \in \Sigma^* : L(M_w) \text{ ist entscheidbar}\}$ Satz von Rice.

b) Welche der folgenden Aussagen sind wahr?

- ☐ Seien $L_1 \subseteq \Sigma^*$ entscheidbar, und $L_2 \subseteq \Sigma^*$ unentscheidbar. Dann ist $L_1 \cup L_2$ unentscheidbar. Gegenbeispiel: $L_1 = \Sigma^*$, $L_2 = \mathcal{H}_0$.
- ☒ Seien $L_1, L_2 \subseteq \Sigma^*$ entscheidbare Sprachen. Dann ist $L_1 \setminus L_2$ entscheidbar. Abschlusseigenschaft.
- ☐ Seien $L_1 \subseteq \Sigma^*$ entscheidbar, und $L_2 \subseteq \Sigma^*$ semi-entscheidbar. Dann gibt es ein $w \in \Sigma^*$, sodass M_w die Sprache $L_1 \cap L_2$ entscheidet. Gegenbeispiel: $L_1 = \Sigma^*$, $L_2 = \mathcal{H}_0$.

c) Welche der folgenden Aussagen sind wahr?

- ☒ Für jedes $v \in \Sigma^*$ ist $L_v := \{w \in \Sigma^* : ww \in L(M_v)\}$ semi-entscheidbar. M_v auf ww ausführen.
- ☐ Für jedes $v \in \Sigma^*$ ist $L_v := \{w \in \Sigma^* : L(M_w) \neq L(M_v)\}$ semi-entscheidbar. Mit $L(M_v) = \Sigma^*$ sind es die nicht-terminierenden TMs, s. Korollar 5.51.
- ☐ Für jedes $v \in \Sigma^*$ ist $L_v := \{w \in \Sigma^* : w \notin L(M_v)\}$ semi-entscheidbar. $L_V = \overline{L(M_v)}$.

d) 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 $(x_1 \wedge x_2) \vee \neg(x_2 \vee \neg x_3)$.
Als *Literal* bezeichnet man eine Variable oder ihre Negation, z.B. sind x_2, x_7 und $\neg x_2$ Literale.
Welche der folgenden Varianten von SAT sind NP-vollständig, unter der Annahme $P \neq NP$?

- ☒ Die Formel F ist in konjunktiver Normalform, sie hat also die Form $F_1 \wedge ... \wedge F_m$, wobei jedes F_i die Form $y_1 \vee ... \vee y_l$ hat und $y_1, ..., y_l$ Literale sind. Ein Beispiel für F ist $(x_1 \vee \neg x_2) \wedge (x_2 \vee x_3)$ Satz 6.22
- ☒ Die Formel F enhält jede Variable x_i höchstens 2022 Mal. Für eine Variable x, die häufiger vorkommt, wird jede zweite Verwendung durch die neue Variable x' ersetzt und wir betrachten die neue Formel $F \wedge (\neg x \vee x') \wedge (\neg x' \vee x)$. Dies wiederholen wir, bis die Bedingung erfüllt ist.
- ☐ Die Formel F ist in disjunktiver Normalform, sie hat also die Form $F_1 \vee ... \vee F_m$, wobei jedes F_i die Form $y_1 \wedge ... \wedge y_l$ hat und $y_1, ..., y_l$ Literale sind. Ein Beispiel für F ist $(x_1 \wedge \neg x_2) \vee (x_2 \wedge x_3)$ F ist erfüllbar, wenn eines der F_i es ist. Letzteres kann man leicht überprüfen.

e) Sei $M := \{L \subseteq \Sigma^* : L \leq_p \text{SAT}\}$ die Menge der Sprachen, die in polynomieller Zeit auf SAT reduziert werden können. Welche der folgenden Aussagen sind wahr? $M = \text{NP}$

- ☒ Wenn $P \neq NP$, dann gibt es eine Sprache $L \in M$, die von keiner Turingmaschine in polynomieller Zeit entschieden wird.
- ☒ Jedes Problem in M kann von einer nichtdeterministischen Turingmaschine in polynomieller Zeit entschieden werden.
- ☒ Alle NP-vollständigen Probleme sind in M enthalten.

Aufgabe 8 **Reduktion (12 Punkte)**

Sei $\Sigma := \{a, b\}$. Für ein Wort $w \in \Sigma^*$ bezeichnen wir mit w^T das Wort, das man erhält, wenn man in w die ersten zwei Buchstaben vertauscht. Falls $|w| < 2$, definieren wir $w^T := w$. Z.B. gilt also $\varepsilon^T = \varepsilon$, $a^T = a$, $(baa)^T = aba$, $(aab)^T = aab$, und $(abbab)^T = babab$. Für eine Sprache $L \subseteq \Sigma^*$ definieren wir dann $L^T := \{w^T : w \in L\}$.

0 1 2

a)* Sei H die Grammatik über dem Alphabet Σ mit den Produktionen $S \rightarrow aSa \mid bSb \mid \varepsilon$. Geben Sie eine kontextfreie Grammatik (CFG) H' an, sodass $L(H') = L(H)^T$ und H' höchstens 10 Produktionen hat. Eine Begründung ist nicht erforderlich.

$S \rightarrow aaTaa \mid abTab \mid baTba \mid bbTbb \mid aa \mid bb \mid \varepsilon$
$T \rightarrow aTa \mid bTb \mid \varepsilon$

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 $L(G) = L(G)^T$?

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.

0 1 2 3 4 5 6

b)* 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 $L(G') = L(G')^T$ gilt.

Wir wählen G' so, dass $L(G') = \{ab\}L(G) \cup \{ba\}\Sigma^*$ gilt. Genauer wählen wir Produktionen $S' \rightarrow abS \mid baR$ und $R \rightarrow aR \mid bR \mid \varepsilon$, sowie die Produktionen aus P.

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

$S' \rightarrow abS \mid baR$
$R \rightarrow aR \mid bR \mid \varepsilon$
$S \rightarrow aSbS \mid ba$

c) Beweisen Sie, dass ihr Verfahren aus Teilaufgabe b) korrekt ist, dass also $L(G) = \Sigma^* \Leftrightarrow L(G') = L(G')^T$ gilt.

0
1
2
3
4

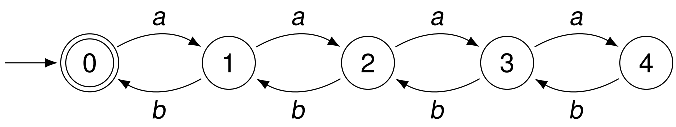
„$\Rightarrow$“: Wenn $L(G) = \Sigma^*$ gilt, dann folgt $L(G') = \{ab\}L(G) \cup \{ba\}\Sigma^* = \{ab, ba\}\Sigma^* = L(G')^T$.

„$\Leftarrow$“: Wir zeigen die Implikation über Kontraposition. Wir nehmen also $L(G) \neq \Sigma^*$ an, somit existiert ein $w \in \Sigma^*$ mit $w \notin L(G)$. Es folgt $abw \notin \{ab\}L(G)$, und offensichtlich gilt $abw \notin \{ba\}\Sigma^*$. Insgesamt erhalten wir $abw \notin L(G')$.
Weiterhin gilt $L(G')^T = \{ba\}L(G)^T \cup \{ab\}\Sigma^*$, und somit $abw \in \{ab\}\Sigma^* \subseteq L(G')^T$. Da das Wort aw in genau einer der Sprachen $L(G'), L(G')^T$ enthalten ist, folgt $L(G') \neq L(G')^T$.

Alternative Lösung: $L(G') = L(G')^T \Rightarrow \{ab\}L(G) \cup \{ba\}\Sigma^* = \{ba\}L(G)^T \cup \{ab\}\Sigma^* \Rightarrow L(G) = \Sigma^*$.

Aufgabe 9 Überwachung (11 Punkte)

Sei $\Sigma := \{a, b\}$, sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA und sei $L \subseteq Q^*$ eine Sprache, d.h. die Wörter von L sind Sequenzen von Zuständen von M. Wir definieren die Sprache $C_L(M)$ als die Menge der Wörter, die M mit einem Lauf in L akzeptiert. Sei z.B. M der folgende DFA:



(Nicht gezeichnete Kanten führen zu einem impliziten Fangzustand.) Für $L = \{0, 1\}^*$ gelten dann $ab \in C_L(M)$ (ab wird mit Lauf $010 \in \{0, 1\}^*$ akzeptiert), $aabb \notin C_L(M)$ ($aabb$ wird mit Lauf $01210 \notin \{0, 1\}^*$ akzeptiert) und $ba \notin C_L(M)$ (ba wird gar nicht akzeptiert).

0 1 2

a)* Sei M der obige DFA. Geben Sie einen regulären Ausdruck für eine Sprache $L \subseteq \{0, 1, 2, 3, 4\}^*$ an, sodass $C_L(M)$ die Menge aller Wörter $w \in L(M)$ ist, die aaa enthalten. Formal: $w \in C_L(M)$ gdw. $w \in L(M)$ und es Wörter $u, v \in \{a, b\}^*$ gibt, mit $w = u\,aaa\,v$.

$(0 \mid 1 \mid 2 \mid 3 \mid 4)^*\ (0123 \mid 1234)\ (0 \mid 1 \mid 2 \mid 3 \mid 4)^*$

0 1 2 3

b)* Sei M der obige DFA. Geben Sie einen regulären Ausdruck für $C_L(M)$ an, wobei $L := \{0, 1\}^*\{2, 3\}^*\{0, 1\}^*$.

$(ab)^*(\epsilon \mid aa(ab)^*bb)(ab)^*$

0 1 2 3 4 5 6

c)* Sei $N = (Q, \Sigma, \delta, q_0, F)$ nun ein beliebiger DFA und sei $N_L = (Q_L, Q, \delta_L, q_{oL}, F_L)$ ein beliebiger DFA mit Alphabet Q für eine Sprache $L \subseteq Q^*$. Konstruieren Sie einen DFA $N' = (Q', \Sigma, \delta', q_0', F')$ mit $L(N') = C_L(N)$.

Geben Sie die Menge der Zustände von N' an:

$Q' := Q \times Q_L$

Geben Sie den Anfangszustand und die Menge der Endzustände von N' an:

$q_0' := (q_0, \delta(q_{0L}, q_0))$ $\qquad$ $F' := F \times F_L$

Beschreiben Sie die Menge δ' der Transitionen von N':

$$\delta'((q, q_L), c) := (\delta(q, c), \delta(q_L, \delta(q, c))) \qquad \text{für } (q, q_L) \in Q' \text{ und } c \in \Sigma$$

Hier können Sie die Idee hinter Ihrer Konstruktion erklären:

Unser DFA simuliert N und N_L gleichzeitig, ähnlich zur Produktkonstruktion. Am Anfang und immer, wenn N eine Transition ausführt, liest N_L den neuen Zustand von N ein. Der DFA akzeptiert, wenn sowohl N als auch N_L akzeptieren.

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