Lehrstuhl für Logik und Verifikation
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 / Endterm **Datum:** Freitag, 9. August 2019
Prüfer: Prof. Dr. Tobias Nipkow **Uhrzeit:** 16:00 – 19:00

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

Bearbeitungshinweise

- Diese Klausur umfasst **24 Seiten** mit insgesamt **8 Aufgaben**.
  Bitte kontrollieren Sie jetzt, dass Sie eine vollständige Angabe erhalten haben.
- Die Gesamtpunktzahl in dieser Prüfung beträgt 60 Punkte.
- Das Heraustrennen von Seiten aus der Prüfung ist untersagt.
- Als Hilfsmittel sind zugelassen:
  - ein beidseitig handbeschriebenes DIN A4 Blatt
  - ein **analoges Wörterbuch** Deutsch ↔ Muttersprache **ohne Anmerkungen**
- Mit * gekennzeichnete Teilaufgaben sind ohne Kenntnis der Ergebnisse vorheriger Teilaufgaben lösbar.
- 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.

Hörsaal verlassen von ________ bis ________ / Vorzeitige Abgabe um ________

Aufgabe 1 Quiz 1: Reguläre Sprachen, kontextfreie Sprachen, Sprachen allgemein (9 Punkte)

0 1 2 3 4 5 6 7 8

a)* Beantworten Sie die folgenden Ja/Nein-Fragen. Begründen Sie jeweils *kurz*. Antworten ohne Begründung werden nicht gewertet. Falls eine Aussage nicht gilt, geben Sie (soweit angebracht) entsprechende Gegenbeispiele an. Antworten wie „Ja, siehe Vorlesung“ sind ohne weiteren Kontext *nicht* ausreichend.

1. Falls $|A| = |A^*|$ ist, ist entweder $|A| = \infty$ oder $A = \{\varepsilon\}$.
2. Sei $w \in \Sigma^*$ und $A \subseteq \Sigma^*$ eine beliebige Sprache. Dann ist $B = \{u \mid u \in A \wedge u \text{ ist Präfix von } w\}$ regulär. (Dabei ist u ein Präfix von w gdw. es ein v gibt mit $w = uv$.)
3. Sei $A \subseteq \Sigma^*$ eine reguläre Sprache und $a \in \Sigma$. Dann ist die Sprache aller Wörter aus A, die das Zeichen a nicht enthalten, ebenfalls regulär.
4. Die Äquivalenz zweier regulärer Ausdrücke ist entscheidbar.
5. Sei M ein minimaler DFA, in dem jeder Zustand ein Endzustand ist. Dann hat M genau einen Zustand.
6. Zwei Sprachen $A, B \subseteq \Sigma^*$ mit den gleichen Myhill–Nerode-Klassen sind immer gleich.
7. Sei CFL die Menge der kontextfreien Sprachen über dem Alphabet $\{0, 1\}$. Dann ist $\mathsf{CFL} \subseteq \mathsf{P}$.
8. Wir nennen eine Produktion *linear*, falls sie die Form $A \to a$, $A \to aB$, oder $A \to Ba$ hat für irgendwelche Nichtterminale A, B und/oder irgendein Terminal a. Wir nennen eine kontextfreie Grammatik linear falls alle ihre Produktionen linear sind. Dann erzeugt jede lineare Grammatik eine reguläre Sprache.

1. Korrekt. Wenn A endlich wäre mit $A \neq \{\varepsilon\}$, dann wäre entweder $A = \emptyset$, sodass $|A| = 0 < 1 \leq |A|^*$, oder es gäbe ein $w \in A$ mit $|w| > 0$, sodass $w^* \subseteq A^*$ und damit $|A| < \infty = |A^*|$.
2. Korrekt, denn B ist endlich.
3. Korrekt. Die gesuchte Sprache ist $A \setminus \Sigma^*\{a\}\Sigma^*$, was regulär ist.
4. Korrekt, Konvertierung in NFA, dann determinisieren und auf Äquivalenz von DFAs testen.
5. Korrekt. Wenn jeder Zustand ein Endzustand ist, ist die akzeptierte Sprache Σ^*. Der minimale DFA für Σ^* hat offensichtlich nur einen Zustand.
6. Falsch. Zum Beispiel haben $A = \{w \mid |w| \text{ gerade}\}$ und $B = \{w \mid |w| \text{ ungerade}\}$ die gleichen Äquivalenzklassen (oder auch $\emptyset$ und Σ^*).
7. Korrekt. Das Wortproblem für CFL ist entscheidbar in $O(n^3)$ durch CYK.
8. Falsch. Die Grammatik $S \to a|aA$, $A \to Sb$ erzeugt die nicht reguläre Sprache $\{a^{n+1}b^n \mid n \in \mathbb{N}\}$.

b)* Gegeben sei die CFG $G = (\{S, T, U\}, \{a, b, c\}, \{S \to aT, T \to Sb, U \to cc\}, S)$. Geben Sie eine geschlossene Form für $\mathsf{L}(G)$ an. Begründen Sie Ihre Antwort *kurz*.

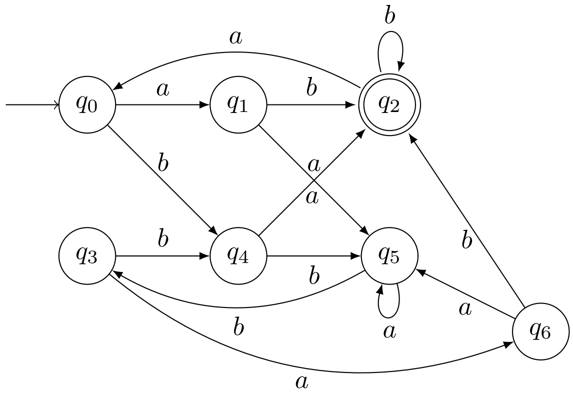
0

1

$\mathsf{L}(G) = \emptyset$, weil kein Nichtterminal nützlich ist (S und T sind nicht erzeugend; U ist nicht erreichbar).

Aufgabe 2 Myhill-Nerode Äquivalenzklassen (8 Punkte)

Sei $\Sigma = \{a, b\}$. Wir betrachten den folgenden DFA M:

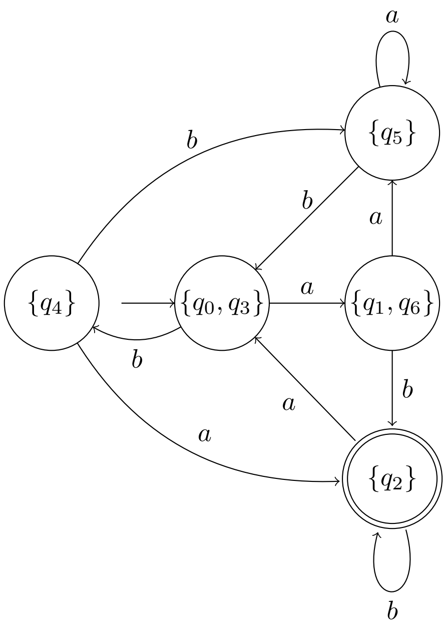


a)* Zeigen Sie, dass $q_1 \not\equiv_M q_3$ gilt.

$\delta(q_1, b) = q_2 \in F$, aber $\delta(q_3, b) = q_4 \notin F$. Damit ist die Definition von Äquivalenz nicht erfüllt. Verweis auf die Ergebnisse der Minimierung wird auch akzeptiert, falls diese richtig ist.

b)* Minimieren Sie den gegebenen DFA M. Geben Sie dazu die vollständige Minimierungstabelle nach Vorlesungsalgorithmus an und zeichnen Sie den resultierenden minimalen DFA.

	q_0	q_1	q_2	q_3	q_4	q_5	q_6
q_0	-	-	-	-	-	-	-
q_1	b	-	-	-	-	-	-
q_2	ε	ε	-	-	-	-	-
q_3	=	b	ε	-	-	-	-
q_4	a	a	ε	a	-	-	-
q_5	ab	b	ε	ba	a	-	-
q_6	b	=	ε	b	a	b	-



c) Geben Sie für jeden Zustand des minimalen Automaten das lexikographisch kleinste kürzeste Wort in der entsprechenden Myhill-Nerode Äquivalenzklasse an.

0

1

$\{q_0, q_3\}$: $[\varepsilon]$
$\{q_1, q_6\}$: $[a]$
$\{q_2\}$: $[ab]$
$\{q_4\}$: $[b]$
$\{q_5\}$: $[aa]$

Aufgabe 3 Rekursion auf regulären Ausdrücken (7 Punkte)

Für ein Wort w und eine Position $1 \leq i \leq |w|$ definieren wir das Wort, das man erhält, indem man das Zeichen an Position i löscht:

$$\mathrm{del}_i(w) = w_1 \dots w_{i-1} w_{i+1} \dots w_{|w|}$$

Sei nun $A \subseteq \Sigma^*$ eine Sprache. Für ein Zeichen $a \in \Sigma$ definieren wir $\mathrm{Del}_a(A)$ als die Menge aller Wörter w, die man erhält, indem man ein Wort aus A nimmt und **genau ein** Vorkommen des Zeichens a daraus löscht.

Formal:

$$\mathrm{Del}_a(A) = \{\mathrm{del}_i(w) \mid w \in A, 1 \leq i \leq |w|, w_i = a\}$$

Beispiel:

$$\mathrm{del}_3(acbdc) = acdc \qquad \mathrm{Del}_a(\{aa, acab, abba\}) = \{a, cab, acb, abb, bba\} \qquad \mathrm{Del}_a(\{bc\}) = \emptyset$$

a)* Finden Sie für $A, B \subseteq \Sigma^*$ einen einfachen Ausdruck für $\mathrm{Del}_a(AB)$ in Abhängigkeit von A, B, $\mathrm{Del}_a(A)$ und $\mathrm{Del}_a(B)$. Beweisen Sie Ihre Behauptung. Falls bei einer Fallunterscheidung zwei Fälle komplett analog sind, dürfen Sie einen davon mit einem entsprechenden Hinweis weglassen.

$\mathrm{Del}_a(AB) = \mathrm{Del}_a(A) \cdot B \cup A \cdot \mathrm{Del}_a(B)$.

Beweis:

Richtung $\subseteq$: Sei $w \in \mathrm{Del}_a(AB)$. Per Definition gibt es dann $u \in A$, $v \in B$, $i \leq |u| + |v|$ mit $w = \mathrm{del}_i(uv)$ und $(uv)_i = a$. Falls $i \leq |u|$ ist $\mathrm{del}_i(uv) = \mathrm{del}_i(u) \cdot v$ und damit $w \in \mathrm{Del}_a(A) \cdot B$. Analog ist für $i > |u|$ dann $w \in A \cdot \mathrm{Del}_a(B)$.

Richtung $\supseteq$: Sei $w \in A \cdot \mathrm{Del}_a(B) \cup \mathrm{Del}_a(A) \cdot B$. Wir betrachten nur den Fall $w \in A \cdot \mathrm{Del}_a(B)$, da der zweite analog ist. Dann gibt es $u \in A$, $v \in B$, $i \leq |v|$, sodass $w = u \cdot \mathrm{del}_i(v)$ ist. Dann ist aber auch $w = \mathrm{del}_{i+|u|}(uv)$ und damit $w \in \mathrm{Del}_a(AB)$.

b)* Geben Sie eine rekursive Prozedur $f_a(r)$ an, die für einen gegebenen regulären Ausdruck r einen regulären Ausdruck zurückgibt mit $\mathsf{L}(f_a(r)) = \mathrm{Del}_a(\mathsf{L}(r))$.

0 1 2

$$
\begin{aligned}
f_a(\emptyset) &= \emptyset \\
f_a(\varepsilon) &= \emptyset \\
f_a(x) &= \begin{cases} \varepsilon & \text{falls } x = a \\ \emptyset & \text{sonst} \end{cases} \\
f_a(r \mid s) &= f_a(r) \mid f_a(s) \\
f_a(r \cdot s) &= f_a(r) \cdot s \mid r \cdot f_a(s) \\
f_a(r^*) &= r^* f_a(r) r^*
\end{aligned}
$$

c) Beweisen Sie die Korrektheit Ihrer Konstruktion durch strukturelle Induktion über r. Sie müssen dabei *nur* die Fälle $r = a$ (einzelnes Zeichen $a \in \Sigma$) und $r = r_1 r_2$ (Konkatenation) behandeln und dürfen die anderen 4 Fälle weglassen. Außerdem dürfen Sie Ihr Ergebnis aus Aufgabe a) verwenden, selbst wenn Sie es nicht beweisen konnten.

Geben Sie in beiden Fällen klar an, was jeweils zu zeigen ist und ggf. was die Induktionshypothesen sind und wo Sie sie verwenden!

Zu zeigen: $\mathsf{L}(f_a(r)) = \mathrm{Del}_a(\mathsf{L}(r))$
Beweis per struktureller Induktion über r:

- Fall $x \in \Sigma$:
  Zu zeigen: $\mathsf{L}(f_a(x)) = \mathrm{Del}_a(\{x\})$

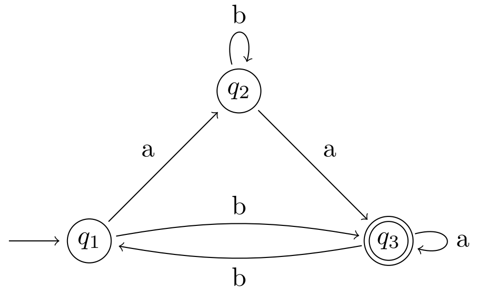
$$\mathrm{Del}_a(\{x\}) = \{\mathrm{del}_i(x) \mid 1 \leq i \leq 1, x = a\} = \begin{cases} \{\varepsilon\} & \text{falls } x = a \\ \emptyset & \text{sonst} \end{cases} = \mathsf{L}(f_a(x))$$

- Fall $r_1 \cdot r_2$:
  Induktionshypothesen: $\mathsf{L}(f_a(r_i)) = \mathrm{Del}_a(\mathsf{L}(r_i))$ für $i \in \{1, 2\}$
  Zu zeigen: $\mathsf{L}(f_a(r_1) \cdot r_2 \mid r_1 \cdot f_a(r_2)) = \mathrm{Del}_a(\mathsf{L}(r_1) \cdot \mathsf{L}(r_2))$

$$\begin{aligned} \mathsf{L}(f_a(r_1) \cdot r_2 \mid r_1 \cdot f_a(r_2)) &= \mathsf{L}(f_a(r_1)) \cdot \mathsf{L}(r_2) \cup \mathsf{L}(r_1) \cdot \mathsf{L}(f_a(r_2)) \\ &\overset{\text{IH}}{=} \mathrm{Del}_a(\mathsf{L}(r_1)) \cdot \mathsf{L}(r_2) \cup \mathsf{L}(r_1) \cdot \mathrm{Del}_a(\mathsf{L}(r_2)) \\ &\overset{\text{a)}}{=} \mathrm{Del}_a(\mathsf{L}(r_1) \cdot \mathsf{L}(r_2)) \\ &= \mathrm{Del}_a(\mathsf{L}(r_1 \cdot r_2)) \end{aligned}$$

Aufgabe 4 **DFA → RegEx (7 Punkte)**

Gegeben sei folgender Automat $M = (\{q_1, q_2, q_3\}, \{a, b\}, \delta, q_1, \{q_3\})$:



a)* Berechnen Sie einen regulären Ausdruck α mit $L(\alpha) = L(M)$. Verwenden Sie dafür eines der beiden aus der Vorlesung bekannten Verfahren (Tabellarisches Verfahren oder Lösen eines Gleichungssystems mithilfe von Ardens Lemma).

Gleichungssystem:

$$X_1 \equiv aX_2 \mid bX_3 \tag{1}$$
$$X_2 \equiv aX_3 \mid bX_2 \tag{2}$$
$$X_3 \equiv aX_3 \mid bX_1 \mid \varepsilon \tag{3}$$

Gleichung (2) nach X_2 auflösen und in (1) einsetzen:

$$X_2 \equiv b^*aX_3 \tag{4}$$
$$X_1 \equiv ab^*aX_3 \mid bX_3 \equiv (ab^*a \mid b)X_3 \tag{5}$$

Gleichung (5) in (3) einsetzen und auflösen:

$$X_3 \equiv aX_3 \mid b(ab^*a \mid b)X_3 \mid \varepsilon \tag{6}$$
$$\equiv (a \mid b(ab^*a \mid b))^* \qquad \equiv (a \mid bb \mid bab^*a)^* \tag{7}$$

Einsetzen in (5):

$$X_1 \equiv \alpha \equiv (ab^*a \mid b)(a \mid bb \mid bab^*a)^* \tag{8}$$

Alternatives Verfahren:
Für die Ausdrücke ohne Zwischenzustand (Schritt 0) erhalten wir (leere Ausdrücke werden nicht dargestellt):

$$\alpha^0_{11} = \varepsilon \quad \alpha^0_{12} = a \quad \alpha^0_{13} = b \quad \alpha^0_{22} = b \mid \varepsilon \quad \alpha^0_{23} = a \quad \alpha^0_{33} = a \mid \varepsilon \quad \alpha^0_{31} = b$$

Mit q_1 als Zwischenzustand (Schritt 1) ergeben sich zwei neue Ausdrücke:

$$\alpha^1_{32} \equiv \emptyset \mid b(\varepsilon)^*a \equiv ba\alpha^1_{33} \equiv a \mid \varepsilon \mid b(\varepsilon)^*b \equiv a \mid \varepsilon \mid bb$$

Mit q_2 als Zwischenzustand erhalten wir zwei neue Ausdrücke (Schritt 2):

$$\alpha^2_{13} \equiv b \mid a(b \mid \varepsilon)^*a \equiv b \mid ab^*a \quad \alpha^2_{33} \equiv a \mid \varepsilon \mid bb \mid ba(b \mid \varepsilon)^*a \equiv \varepsilon \mid a \mid bb \mid bab^*a$$

Der Schritt, in dem wir q_3 als Zwischenzustand betrachten ist der letzte Schritt. Deswegen interessieren wir uns nur noch für Ausdrücke, die vom Startzustand in den Endzustand führen:

$$\alpha^3_{13} \equiv b \mid ab^*a \mid ab^*a \mid (\varepsilon \mid a \mid bb \mid bab^*a)^*(\varepsilon \mid a \mid bb \mid bab^*a) \equiv (ab^*a \mid b)(a \mid bb \mid bab^*a)^*$$

Damit ergibt sich für den Gesamtausdruck: $\alpha \equiv (ab^*a \mid b)(a \mid bb \mid bab^*a)^*$.

0
1
2

b)* Geben Sie eine kontextfreie Grammatik für $L(M)$ mit **höchstens 3** Nichtterminalen an.

Die Grammatik ist fast identisch mit dem Gleichungssystem.
$G = (\{X_1, X_2, X_3\}, \{a, b\}, P, X_1)$ mit den Produktionen

$$X_1 \to aX_2 \mid bX_3$$
$$X_2 \to aX_3 \mid bX_2$$
$$X_3 \to aX_3 \mid bX_1 \mid \varepsilon$$

Alternativ kann man auch eine Grammatik aus dem regulären Ausdruck ablesen.

Aufgabe 5 **Kontextfreie Sprachen (7 Punkte)**

Wir betrachten die Sprache $\mathsf{L} = \{\mathsf{a}^n\mathsf{b}^n \mid n \in \mathbb{N} \wedge n > 0\} \cup \{\mathsf{a}^{2n}\mathsf{b}^n \mid n \in \mathbb{N} \wedge n > 0\}$.

a)* Geben Sie einen Kellerautomaten (PDA) $K = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, \emptyset)$ mit einelementigem $\Gamma = \{Z_0\}$ an, der L mit leerem Keller akzeptiert. Sie dürfen dabei **höchstens 6 Zustände** verwenden. Zeichnen Sie den Automaten.

0 1 2 3 4 5

Eine mögliche Lösung arbeitet nach dem Prinzip zunächst entweder für jedes a ein Z_0 auf den Keller zu legen oder nur für jedes zweite a. Danach wird für jedes b ein Z_0 entfernt. Beim Übergang von der ersten in die zweite Phase löschen wir ein Z_0, damit am Ende der Keller leer ist. Sei $Q = \{q_0, q_0', q_1, q_2, q_2', q_3\}$. Wir definieren:

$$\begin{aligned}
\delta(q_0, a, Z_0) &= \{(q_1, Z_0Z_0), (q_0', Z_0)\} \\
\delta(q_0', a, Z_0) &= \{(q_2, Z_0Z_0)\} \\
\delta(q_1, a, Z_0) &= \{(q_1, Z_0Z_0)\} \\
\delta(q_1, \varepsilon, Z_0) &= \{(q_3, \varepsilon)\} \\
\delta(q_2, a, Z_0) &= \{(q_2', Z_0)\} \\
\delta(q_2', a, Z_0) &= \{(q_2, Z_0Z_0)\} \\
\delta(q_2, \varepsilon, Z_0) &= \{(q_3, \varepsilon)\} \\
\delta(q_3, b, Z_0) &= \{(q_3, \varepsilon)\}
\end{aligned}$$

Eine andere Lösung legt zunächst für jedes a ein oder zwei Z_0 auf den Keller, und löscht anschließend für jedes b zwei Z_0. Dabei muss allerdings sichergestellt werden dass für den Fall $\mathsf{a}^{2n}\mathsf{b}^n$ eine gerade Anzahl von a's gelesen wird.

0
1
2

b)* Geben Sie eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$ mit **höchstens 3 Nichtterminalen** an, die L erzeugt.

Sei $V = \{S, X, Y\}$ und P bestehe aus den folgenden Produktionen:

$$S \to X \mid Y$$
$$X \to \mathsf{ab} \mid \mathsf{a}X\mathsf{b}$$
$$Y \to \mathsf{aab} \mid \mathsf{aa}Y\mathsf{b}$$

Aufgabe 6 Kontextfreie Sprachen (2) (5 Punkte)

Wir betrachten die Sprache $\mathsf{L} = \{w \mid w \in \{\mathsf{a},\mathsf{b}\}^* \wedge |w|_\mathsf{a} \leq |w|_\mathsf{b}\}$ und die Grammatik $G = (\{S, X\}, \{\mathsf{a},\mathsf{b}\}, P, S)$ mit den folgenden Produktionen P:

$$S \to XS\,\mathsf{b} \mid \mathsf{b}\,SX \mid \varepsilon \qquad X \to \mathsf{a} \mid \varepsilon$$

Zeigen Sie $\mathsf{L}(G) \subseteq \mathsf{L}$.

Wir halten zunächst fest, dass $\mathsf{L}(X) = \{\varepsilon, \mathsf{a}\}$. Sei $w \in \mathsf{L}(G)$. Damit $S \to^* w$. Wir zeigen $|w|_\mathsf{a} \leq |w|_\mathsf{b}$ mittels Induktion über die Ableitung von w.

Die einzige Induktionsbasis ist das leere Wort. Dieser Fall is offensichtlich.

Für den Induktionsschritt nehmen wir an $S \to^{n+1} w$. Die Induktionshypothese gilt für alle $m \leq n$, d.h. $\forall m \leq n. \forall w'. S \to^m w' \longrightarrow |w'|_\mathsf{a} \leq |w'|_\mathsf{b}$. Fallunterscheidung über den ersten Schritt der Ableitung:

- $S \to Xu\mathsf{b} \to^n uv\mathsf{b}$ mit $w = uv\mathsf{b}$, $u \in \mathsf{L}(X)$ und $S \to^m u$ für $m \leq n$. Dann gilt

$$\begin{aligned}
& |w|_\mathsf{a} \\
= & |u|_\mathsf{a} + |v|_\mathsf{a} \\
\leq & 1 + |v|_\mathsf{a} \\
\leq & 1 + |v|_\mathsf{b} && \text{(IH)} \\
= & 1 + |u|_\mathsf{b} + |v|_\mathsf{b} \\
= & |w|_\mathsf{b}
\end{aligned}$$

- Der Fall für die zweite Ableitung ist analog.

Damit haben wir $w \in \mathsf{L}$ gezeigt.

0
1
2
3
4
5

Aufgabe 7 **Seminarproblem (8 Punkte)**

Wir wollen ein Seminar über m Wochen abhalten und dafür jede Woche eine(n) Vortragenden einladen. Dafür stehen uns l potenzielle Vortragende zur Verfügung. Jede(r) Vortragende ist nur in bestimmten Wochen verfügbar. Zudem deckt jede(r) Vortragende eine gewisse Menge von Themengebieten ab. Am Ende des Seminars wollen wir insgesamt eine bestimmte Menge von Themengebieten abgedeckt haben.

Wir betrachten das Entscheidungsproblem SEMINAR:

Gegeben: Eine Menge S und Listen von Mengen $L = (L_1, \dots, L_l)$ und $T = (T_1, \dots, T_l)$, wobei L_i angibt, in welchen Wochen der/die Vortragende i verfügbar ist. Die Menge T_i gibt an, welche Themengebiete der/die Vortragende i abdeckt. Die Menge S definiert die abzudeckenden Themengebiete.

Zu entscheiden: Gibt es eine Reihung von Vortragenden $v_1, \dots, v_m$, so dass $i \in L_{v_i}$ für alle $1 \leq i \leq m$ und $S \subseteq \bigcup_{1 \leq i \leq m} T_{v_i}$?

a)* Zeigen Sie, dass SEMINAR $\in$ NP.

Als Zertifikat dient die Reihung von Vortragenden $v_1, \dots, v_m$. Wir können in polynomieller Zeit überprüfen, ob die folgenden Eigenschaften gelten:

- $1 \leq v_i \leq l$ für alle $1 \leq i \leq m$
- $i \in L_{v_i}$ für alle $1 \leq i \leq m$
- $S \subseteq \bigcup_{1 \leq i \leq m} T_{v_i}$

b)* Zeigen Sie, dass SEMINAR NP-schwer ist, indem Sie eine polynomielle Reduktion von 3KNF-SAT auf SEMINAR angeben und deren Korrektheit beweisen. Führen Sie Ihre Reduktion auch explizit für die Formel $(x_1 \vee \neg x_3 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_3)$ durch.

Gegeben seien k Klauseln $c_1, \dots, c_k$. Die Literale von c_i seien mit $l_{i,j}$ für $j \in \{1, 2, 3\}$ bezeichnet (nicht alle davon müssen für jede Klausel existieren). Die Variablen bezeichnen wir mit $x_1, \dots, x_n$.

Wir definieren die Reduktion f, die eine Probleminstanz für SEMINAR folgendermaßen konstruiert. Setze $m := n$ und $l := 2n$. Die Literale entsprechen also den Vortragenden und die Variablen den Seminarwochen. Wir benutzen dabei eine Abbildung g, mit $g(x_i) = 2i$ und $g(\neg x_i) = 2i + 1$. Wir konstruieren L so, dass $L_{g(x_i)} = \{i\}$ und $L_{g(\neg x_i)} = \{i\}$. Setze $S := \{1, \dots, k\}$. T enthält für jedes Literal t die Menge, die aus allen i mit $t \in c_i$ besteht. Formal: $T_t = \{i \mid t \in c_i \wedge 1 \leq i \leq k\}$.

Die Reduktion f ist offensichtlich in polynomieller Zeit berechenbar.

Beispiel: Wir erhalten:

- $l = 8$
- $m = 4$
- $L = (\{1\}, \{1\}, \{2\}, \{2\}, \{3\}, \{3\}, \{4\}, \{4\})$
- $T = (\{1\}, \{2\}, \{2\}, \{\}, \{2\}, \{1\}, \{1\}, \{\})$

Korrektheit: Sei σ eine erfüllende Belegung der Variablen in $c_1, \ldots, c_k$. Dann gibt es mindestens ein Literal $l_{i,j}$ mit $\sigma(l_{i,j}) = 1$ für alle $1 \leq i \leq k$. Sei $v_1, \ldots, v_n$ eine Reihung mit $v_j = g(x_j)$ falls $\sigma(x_j) = 1$ und $v_j = g(\neg x_j)$ falls $\sigma(x_j) = 0$. Offensichtlich gilt $j \in L_{v_j}$ für alle $1 \leq j \leq n$. Für jede Klausel c_i gibt es mindestens ein Literal $t \in c_i$ mit $\sigma(t) = 1$. Damit $i \in T_{g(t)}$ und $i \in \bigcup_{1 \leq j \leq m} T_{v_j}$. Also $S \subseteq \bigcup_{1 \leq i \leq m} T_{v_i}$.

Sei umgekehrt $v_1, \ldots, v_n$ eine valide Reihung. Konstruiere σ so, dass $\sigma(x_i) = 1$ falls $v_i = g(x_i)$ und $\sigma(x_i) = 0$ falls $v_i = g(\neg x_i)$. Diese Belegung ist erfüllend, da es für jede Klausel c_i mit $i \in S$, ein j gibt, so dass $i \in T_{v_j}$. Also $\sigma(t) = 1$ für ein t mit $g(t) = v_j$ und damit ist Klausel c_i erfüllt.

Aufgabe 8 **Quiz 2: Berechenbarkeit und Komplexität (9 Punkte)**

0 1 2 3 4 5 6

a)* Beantworten Sie die folgenden Ja/Nein-Fragen. Begründen Sie jeweils *kurz*. Antworten ohne Begründung werden nicht gewertet. Falls eine Aussage nicht gilt, geben Sie (soweit angebracht) entsprechende Gegenbeispiele an. Antworten wie „Ja, siehe Vorlesung“ sind ohne weiteren Kontext *nicht* ausreichend.

1. Es gibt eine Turingmaschine M, deren Halteproblem $\{w \mid M[w]{\downarrow}\}$ entscheidbar ist.
2. Es gibt eine Turingmaschine M, deren Halteproblem $\{w \mid M[w]{\downarrow}\}$ unentscheidbar ist.
3. Die Menge $\{w \mid \varphi_w$ ist LOOP-berechenbar$\}$ ist unentscheidbar.
4. Die Menge $\{w \mid \varphi_w$ ist WHILE-berechenbar$\}$ ist unentscheidbar.
5. Seien $A, B \subseteq \Sigma^*$. Wenn $A \leq B$ ist und $B \leq A$, dann ist $A = B$.
6. Sei $f : \mathbb{N} \to \mathbb{N}$ und f berechenbar. Dann sind alle Sprachen in $\mathrm{NTIME}(f(n))$ entscheidbar.

1. Korrekt, z.B. die Maschine, die nie hält.
2. Korrekt, z.B. die universelle TM oder die TM, die das PCP (semi-)entscheidet. Allgemein funktioniert hier jede TM, die ein semi-entscheidbares Problem „semi-entscheidet“.
3. Korrekt. Nach Satz von Rice, da es berechenbare Funktionen gibt, die LOOP-berechenbar sind (z.B. konstante Funktionen), aber auch welche, die es nicht sind (z.B. die Ackermann-Funktion oder jede nicht-totale Funktion).
4. Falsch. φ_w ist immer eine berechenbare Funktion, und alle berechenbaren Funktionen sind auch WHILE-berechenbar.
5. Falsch. $A \leq B$ und $B \leq A$ gilt u.a. für alle nicht-trivialen entscheidbaren Sprachen.
6. Korrekt. Wenn $A \in \mathrm{NTIME}(f(n))$ ist, gibt es eine nichtdeterministische Turingmaschine M mit $\mathsf{L}(M) = A$ und $\forall x{\in}\Sigma^*.\ \mathrm{ntime}_M(x) \leq f(|x|)$. Wir können also A entscheiden, indem wir M für $f(n)$ Schritte simulieren und dann *false* zurückgeben.

b)* Beantworten Sie für die folgenden „Beweise“ jeweils folgende Fragen:

- Was ist der Fehler an dem „Beweis“? (je 1 Punkt)
- Gilt die behauptete Aussage trotzdem? (je 0,5 Punkte)

0
1
2
3

Achtung: „Beweis ist falsch, weil die Aussage nicht gilt“ reicht *nicht* als Begründung. Erklären Sie, welcher Schritt in dem Beweis genau falsch ist und warum.

1. Das PCP (Post'sches Korrespondenzproblem) ist in NP, denn eine Lösung $i_1 \dots i_n$ stellt ein in polynomieller Zeit verifizierbares Zertifikat dar.
2. Wir nennen eine Turingmaschine *kreisfrei*, wenn ihr Zustandsübergangsgraph (in der aus der Vorlesung bekannten graphischen Notation für Turingmaschinen) keine Zyklen enthält. Dann ist die Menge $\{w \mid M_w \text{ kreisfrei} \wedge \varphi_w(0) = 0\}$ nach Satz von Rice unentscheidbar, da es offensichtlich sowohl Turingmaschinen gibt, die darin enthalten sind als auch welche, die nicht darin enthalten sind.

1. Die Lösungen sind nicht notwendigerweise polynomiell beschränkt. Sie können zwar in polynomieller Zeit bzgl. ihrer Länge geprüft werden, aber nicht notwendigerweise in polynomieller Zeit bzgl. der Länge der Eingabe (also der Größe der PCP-Probleminstanz selbst). Die Aussage gilt nicht, da das PCP noch nicht einmal entscheidbar ist.
2. „Kreisfrei“ ist keine Eigenschaft von φ_w, sondern eine syntaktische Eigenschaft der Turingmaschine. Die Aussage gilt nicht, denn wir können für eine gegebene Turingmaschine entscheiden, ob sie kreisfrei ist, und wenn sie kreisfrei ist, können wir auch $\varphi_w(0) = 0$ in endlicher Zeit überprüfen, da sie dann auf jeder Eingabe terminiert.

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

Lösungsvorschlag