

**2. Klausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2024/2025**

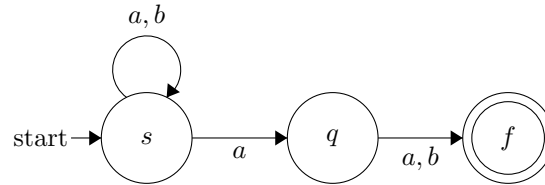
Lösung!

- Schreiben Sie Ihren Namen und Matrikelnummer auf das Deckblatt. Beschriften Sie jedes Aufgabenblatt mit Ihrer Matrikelnummer.
- Schreiben Sie die Lösungen auf die Aufgabenblätter und Rückseiten. Am Ende der Klausur sind zusätzliche Leerseiten. Fordern Sie zusätzliches Papier bitte nur an, wenn nötig.
- Die Tackernadel darf nicht gelöst werden, Sie dürfen allerdings die NP-vollständigen Probleme im Anhang abtrennen.
- Begründen/Beweisen Sie Ihre Antworten ausreichend, wenn "Zeigen/Beweisen Sie, dass" gefordert wird.
- \mathbb{N}_0 bezeichnet die natürlichen Zahlen inklusive 0, \mathbb{N} die natürlichen Zahlen ohne 0.
- Als Hilfsmittel ist ein handbeschriebenes A4-Papier erlaubt.
- Einlesezeit: 15 min
Bearbeitungszeit: 2 h

	Mögliche Punkte					Erreichte Punkte					
	a	b	c	d	Σ	a	b	c	d	Σ	
Aufg. 1	2	4	4	–	10						
Aufg. 2	3	2	3	1	9						
Aufg. 3	1	2	4	2	9						
Aufg. 4	2	2	6	–	10						
Aufg. 5	2	2	4	4	12						
Aufg. 6	1	2	5	2	10						
Σ						60					

Problem 1: Warmup

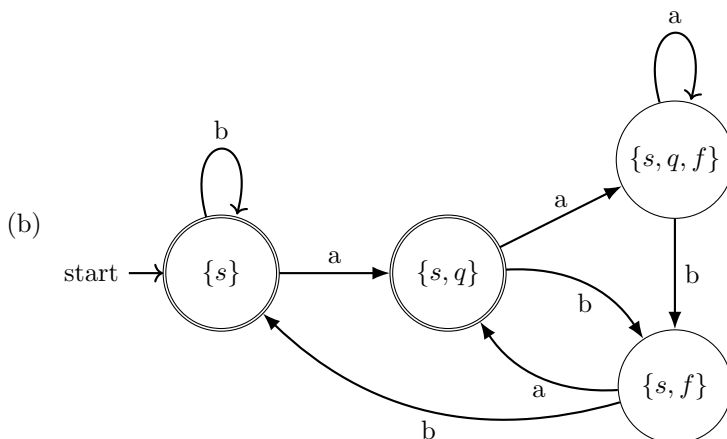
2 + 4 + 4 = 10 Punkte

Gegeben sei der folgende NEA \mathcal{A} über $\Sigma = \{a, b\}$:

- (a) Geben Sie einen regulären Ausdruck an, der $L(\mathcal{A})$ beschreibt.
- (b) Geben Sie einen DEA an, der die Sprache $(L(\mathcal{A}))^C$ akzeptiert. Die Übergangsfunktion des DEAs $\delta : Q \times \Sigma \rightarrow Q$ soll total definiert sein.
- (c) Seien r_1, r_2 beliebige reguläre Ausdrücke. Geben Sie einen Algorithmus an, der entscheidet, ob es ein Wort $w \in \Sigma^*$ gibt, sodass $w \in L(r_1)^C \cap L(r_2)^C$.

Lösung:

- (a) $(a \cup b)^* a (a \cup b)$



- (c) Seien A_1, A_2 jeweils DEAs die $L(r_1)$ und $L(r_2)$ akzeptieren. Wir konstruieren nun die Komplementautomaten \hat{A}_1, \hat{A}_2 und bilden den Produktautomaten $A = \hat{A}_1 \times \hat{A}_1$. Dieser erkennt offensichtlich die Sprache $L(r_1)^C \cap L(r_2)^C$.

Um zu entscheiden ob es ein Wort $w \in L(\mathcal{A})$ gibt, lösen wir das Leerheitsproblem (i.e. ob gilt dass $L(\mathcal{A}) = \emptyset$). Dazu führen wir eine BFS beginnend von jedem Startzustand aus. Gibt es einen erreichbaren Endzustand, so gilt $L(\mathcal{A}) \neq \emptyset$ und somit existiert ein Wort $w \in \Sigma$, sodass $w \in L(r_1)^C \cap L(r_2)^C$. Anderenfalls ist die Sprache des Automaten leer und es existiert kein solches w .

Problem 2: Kontextfreie Sprachen

3+2+3+1 = 9 Punkte

- (a) Zeigen Sie, dass die Sprache $L_1 := \{c^{2n}b^{3n} \mid n \in \mathbb{N}_0\}$ nicht regulär ist.
- (b) Zeigen Sie, dass die Sprache L_1 kontextfrei ist, indem Sie eine kontextfreie Grammatik G angeben, die L_1 akzeptiert.
- (c) Betrachten Sie die folgende kontextfreie Grammatik $G = (\{S, T, U, A, B, C\}, \{a, b, c\}, S, R)$ in Chomsky-Normalform mit den folgenden Regeln R :

$$S \rightarrow TS \mid CT \mid a$$

$$T \rightarrow AU \mid TT \mid c$$

$$U \rightarrow SB \mid AB$$

$$A \rightarrow a$$

$$B \rightarrow b$$

$$C \rightarrow c$$

Geben Sie an, ob das Wort $ccaab$ zur Sprache $\mathcal{L}(G)$ gehört. Wenden Sie hierzu den CYK-Algorithmus an. Benutzen Sie die unten gegebene Tabelle.

V_{i5}					
V_{i4}					
V_{i3}					
V_{i2}					
V_{i1}					
	c	c	a	a	b

- (d) Welche Bedingung muss für ein Nichtterminal T gelten, damit $T \in V_{ij}$ im CYK-Algorithmus? Hier bezeichnet V_{ij} wie in der Vorlesung die Zelle in der i -ten Spalte und der j -ten Zeile, nummeriert wie in der Tabelle.

Lösung:

- (a) Wir widerlegen das Pumping-Lemma für reguläre Sprachen:

Sei $w = c^{2n}b^{3n} \in L_1$ der Länge $|w| = 5n > n$ für ein beliebiges n .

Für alle Zerlegungen der Form $w = uvx$ sodass $|uv| \leq n$ mit $v \neq \varepsilon$ gilt, dass $uv = c^{|uv|}$ und $x = c^{2n-|uv|}b^{3n}$.

Wir wählen $i = 0$ und es gilt $uv^0x = c^{|u|}c^{2n-|uv|}b^{3n} = c^{2n-|v|}b^{3n} \notin L_1$ da $v \neq \varepsilon$ und somit $|v| > 0$.

- (b) Wir geben die folgende Kontextfreie Grammatik an. Sei $G = (\{S, B, C\}, \{b, c\}, S, R)$ mit den folgenden Regeln:

$$S \rightarrow CCSBBB \mid \varepsilon$$

$$C \rightarrow c$$

$$B \rightarrow b$$

(c)

V_{i5}	$\{T, S\}$				
V_{i4}	\emptyset	$\{T, S\}$			
V_{i3}	$\{S\}$	\emptyset	$\{T\}$		
V_{i2}	$\{T, C\}$	$\{S\}$	\emptyset	$\{U\}$	
V_{i1}	$\{T, C\}$	$\{T, C\}$	$\{S, A\}$	$\{S, A\}$	$\{B\}$
	c	c	a	a	b

- (d) Ein Nichtterminal steht T in Zelle V_{ij} bei Anwendung des CYK-Algorithmus auf ein Wort w genau dann wenn $T \xrightarrow{*} w_i, \dots, w_{i+j-1}$.

Problem 3: Approximationsalgorithmen

1 + 2 + 4 + 2 = 9 Punkte

In dieser Aufgabe betrachten wir das Problem 3-HITTING SET.

3-HITTING SET**Gegeben:** Eine Grundmenge M und Teilmengen $A_1, \dots, A_n \subseteq M$, mit $|A_i| \leq 3$ für alle $i \in \{1, \dots, n\}$.**Problem:** Berechne eine Menge $Z \subseteq M$ mit minimaler Kardinalität, sodass für alle $i \in \{1, \dots, n\}$ gilt $Z \cap A_i \neq \emptyset$.

- (a) Gegeben der folgenden Instanz von 3-Hitting Set über der Grundmenge $\{1, \dots, 8\}$. Geben Sie eine optimale Lösung an.

$$A_1 = \{2, 4, 5\}$$

$$A_2 = \{1, 2, 5\}$$

$$A_3 = \{1, 3, 8\}$$

$$A_4 = \{1, 4, 7\}$$

$$A_5 = \{4, 6, 7\}$$

$$A_6 = \{4, 5, 8\}$$

Wir geben den folgenden Algorithmus an:

Algorithmus \mathcal{A} : GREEDY 3-HITTING SET**Input:** (A_1, A_2, \dots, A_n) mit $|A_i| \leq 3$.1 $Z \leftarrow \emptyset$ 2 **while** $\exists i : Z \cap A_i = \emptyset$ **do**3 $Z \leftarrow Z \cup A_i$ 4 **return** Z

- (b) Zeigen Sie, dass Algorithmus \mathcal{A} ein korrekter Approximationsalgorithmus für 3-Hitting Set ist.
- (c) Zeigen Sie, dass Algorithmus \mathcal{A} eine Approximation mit relativer Gütegarantie 3 erreicht.
- (d) Geben Sie für jedes $n \in \mathbb{N}$ eine Instanz $I_n = (M, A_1, \dots, A_n)$ von 3-Hitting Set an, sodass ein minimales 3-Hitting Set Größe OPT hat und $\mathcal{A}(I_n) = 3 \cdot OPT$ gilt.

Lösung:

- (a) Die (einzig) optimale Lösung ist $\{1, 4\}$.
- (b) Wenn der Algorithmus terminiert, gilt klar dass $Z \cap A_i \neq \emptyset$ für all $i \in \{1, \dots, n\}$, somit müssen wir nur argumentieren, dass er terminiert. Da in jedem Durchlauf der Schleife mindestens ein weiteres Set A_i mit $A_i \cap Z = \emptyset$ zu Z hinzugefügt wird, terminiert der Algorithmus nach maximal n Schritten.
- (c) Sei $OPT(I)$ eine optimale Lösung und $Z := \mathcal{A}(I)$ die Lösung des Greedy Algorithmus für eine gegebenen Instanz I von 3-Hitting Set.

Betrachte die A_i die von \mathcal{A} zur Lösung Z hinzugefügt wurden. Es gilt $A_i \cap A_j = \emptyset$ for $i \neq j \in [n]$. Sei nun also $J = \{i \mid A_i \text{ wurde von } \mathcal{A} \text{ zu } Z \text{ hinzugefügt}\}$.

Dann gilt $OPT \geq |J|$, da jedes A_i mit mindestens einem eigenen Element abgedeckt werden muss, da die Mengen disjunkt sind. Ferner gilt $Z = 3 \cdot |J|$, da für alle $i \in [n]$ gilt, dass $|A_i| = 3$.

Somit folgt

$$\frac{|\mathcal{A}(I)|}{|OPT(I)|} \leq \frac{3|J|}{|J|} = 3$$

(d) Wir definieren eine Instanz $I_n = (M, \{A_1, \dots, A_n\})$ wie folgt:

$$M = \{1, \dots, 3n\}$$
$$A_i = \{3i, 3i + 1, 3i + 2\} \quad \forall i \in \{0, \dots, n - 1\}$$

Nun gilt dass $Z = M = 3n$, da \mathcal{A} jedes Set A_i zu Z hinzufügt, da alle A_i paarweise disjunkt sind. Die optimale Lösung beinhaltet notwendigerweise genau ein Element aus jedem A_i . Somit gilt $3 \cdot |OPT| = 3 \cdot n = \mathcal{A}(I_n)$

Problem 4: NP-Vollständigkeit

2 + 2 + 6 = 10 Punkte

In dieser Aufgabe betrachten wir *gewichtete Multigraphen*.

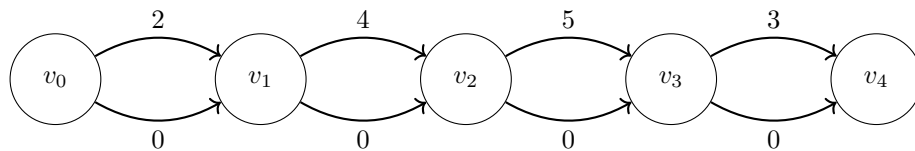
Ein gewichteter Multigraph ist ein Graph $G = (V, E, c)$, bei dem zwei Knoten u und v mit mehr als einer Kante verbunden werden können, wobei jede Kante $e \in E$ dabei ein Gewicht $c(e) \in \mathbb{N}_0$ hat. Das Gewicht eines Pfades $P = e_1, \dots, e_n$ ist gegeben durch $c(P) = \sum_{e \in P} c(e)$.

GEWICHTETE MULTIGRAPH ERREICHBARKEIT

Gegeben: Ein gerichteter, gewichteter Mutigraph $G = (V, E, c)$ und eine natürliche Zahl $t \in \mathbb{N}_0$ sowie zwei Knoten $u, v \in V$.

Problem: Gibt es einen Pfad von u nach v in G mit Gewicht genau t ?

(a) Betrachten Sie den folgenden gewichteten Multigraphen G :



Geben Sie für die folgenden Eingaben an, ob es sich um eine Ja oder Nein-Instanz handelt:

(u, v, t)	Ja/Nein-Instanz
$(v_0, v_4, 7)$	
$(v_2, v_3, 2)$	
$(v_0, v_3, 2)$	
$(v_4, v_1, 12)$	

(b) Zeigen Sie, dass GEWICHTETE MULTIGRAPH ERREICHBARKEIT \in NP gilt.

(c) Zeigen Sie, dass GEWICHTETE MULTIGRAPH ERREICHBARKEIT NP-schwer ist. Reduzieren Sie hierzu von einem der NP-vollständigen Probleme im Anhang.

Tipp: Betrachten Sie den Graphen aus Aufgabenteil a)

Lösung:

(w, t)	Ja/Nein-Instanz
$(v_0, v_4, 7)$	Ja
$(v_2, v_3, 2)$	Nein
$(v_0, v_3, 2)$	Ja
$(v_4, v_1, 12)$	Nein

(a) Gegeben einer Instanz (G, u, v, t) konstruieren wir einen polynomielle Orakel-TM auf die folgende Art und Weise: Das Orakel schreibt uns als Zertifikat eine Pfad e_1, \dots, e_p auf das Band. Wir prüfen nun folgendes:

- Es handelt sich um einen korrekten Pfad in G wobei $e_1 = u$ und $e_p = v$. Überprüfung geschieht in $\mathcal{O}(p)$
- Die Summer der Gewichte des Pfades beträgt genau t . Hierzu prüfen wir genau p Kanten und summieren ihre Gewichte auf, das geschieht in $\mathcal{O}(p)$

Sind diese Eigenschaften erfüllt akzeptieren wir, ansonsten lehnen wir ab. Die Orakel-Turingmaschine läuft somit in polynomieller Gesamtzeit $\mathcal{O}(p) = \mathcal{O}(m)$.

(c) Wir reduzieren von SUBSETSUM. Gegeben einer Instanz von SubsetSum (a_1, \dots, a_n, t) konstruieren wir folgenden Graphen $G = (V, E, c)$

$$\begin{aligned} V &= \{v_0\} \cup \{v_i \mid a_i \in \{a_1, \dots, a_n\}\} \\ E_w &= \{e_i \mid e_i := (v_{i-1}, v_i), i \in [n]\} \\ E_0 &= \{e'_i \mid e'_i := (v_{i-1}, v_i), i \in [n]\} \\ E &= E_w \cup E_0 \end{aligned}$$

mit

$$\begin{aligned} c(e_i) &= a_i \quad e_i \in E_w \\ c(e'_i) &= 0 \quad e'_i \in E_0 \end{aligned}$$

Schließlich wählen wir $t' = t$ und setzen $u = v_0$ sowie $v = v_n$ um die Instanz (G, v_0, v_n, t') zu erhalten. **Laufzeit:** Der Graph G hat $n + 1$ Knoten so wie $2n$ Kanten, somit lässt sich dieser in Zeit $\mathcal{O}(n)$ konstruiert. Wir können t' in Zeit $\mathcal{O}(\log(t))$ kopieren, sowie u, v in konstanter Zeit.

Korrektheit:

\Rightarrow : Sei (a_1, \dots, a_n, t) eine Ja-Instanz für SUBSETSUM. Es existiert also ein $I \subseteq \{1, \dots, n\}$, sodass $\sum_{i \in I} a_i = t$.

Wir konstruieren nun eine Pfad der Länge n von v_0 zu v_n in G auf die folgende Art und Weise:

$$P = e_1, \dots, e_n$$

mit $e_i = (v_{i-1}, v_i)$. Welche Multikanten wir wählen hängt nun von A ab: Wir wählen $e_i \in E_w$ wenn $i \in I$ andernfalls wählen wir $e_i \in E_0$.

P hat ein Gewicht von

$$c(P) = \sum_{i=1}^n c(e_i) = \sum_{i \in I} c(e_i) + \sum_{i \notin I} c(e_i) = \sum_{i \in I} a_i + \sum_{i \notin I} 0 = t = t'.$$

und somit ist (G, v_0, v_n, t') eine Ja-Instanz für GEWICHTETE MULTIGRAPH ERREICHBARKEIT.

\Leftarrow : Gegeben einer Ja-Instanz (G, v_0, v_n, t') für GEWICHTETE MULTIGRAPH ERREICHBARKEIT, konstruieren wir eine SubsetSum Lösung auf die folgende Art:

Sei P der Pfad von v_0 nach v_n mit Gewicht t . Aufgrund der Wahl von G hat dieser genau n Kanten, da G gerichtet und azyklisch ist sowie Kanten lediglich zwischen v_{i-1} und v_i existieren für alle $i \in [n]$. Wir schreiben $P = e_1, \dots, e_n$ und wählen nun $I = \{i \mid c(e_i) \neq 0\}$. Damit gilt folglich:

$$w = \sum_{i \in I} a_i = \sum_{e_i \in P, c(e_i) \neq 0} c(e_i) = \sum_{e_i \in P} c(e_i) = t' = t.$$

Somit hat unsere Lösung I für SUBSETSUM Gewicht t und ist somit eine Ja-Instanz.

Problem 5: Zeige oder Widerlege

2 + 2 + 4 + 4 = 12 Punkte

- (a) Jeder reguläre Sprache ist in NP.

Kreuzen Sie an: Ich zeige die Aussage. Ich widerlege die Aussage.

- (b) Wenn
- $L_1 \cup L_2$
- kontextfrei ist, dann ist mindestens eine der Sprachen
- L_1
- oder
- L_2
- bereits kontextfrei.

Kreuzen Sie an: Ich zeige die Aussage. Ich widerlege die Aussage.

- (c) Sei
- $L \subseteq \{0, 1\}^*$
- eine Sprache, sodass
- L
- semi-entscheidbar und
- L^C
- nicht semi-entscheidbar ist. Die Sprache

$$L' = \{0w \mid w \in L\} \cup \{1w \mid w \notin L\}$$

ist semi-entscheidbar.

Kreuzen Sie an: Ich zeige die Aussage. Ich widerlege die Aussage.

- (d) Seien
- $L_1, \dots, L_k \subseteq \Sigma^*$
- semi-entscheidbar. Außerdem gelte, dass die Sprachen
- L_i
- eine Partition von
- Σ^*
- bilden. Das heißt, die Sprachen
- L_i
- sind paarweise disjunkt, also für
- $i < j$
- gilt
- $L_i \cap L_j = \emptyset$
- und
- $\bigcup_{i=1}^k L_i = \Sigma^*$
- .

Es gilt, dass jede Sprache L_i für $i \in \{1, \dots, k\}$ entscheidbar ist.Kreuzen Sie an: Ich zeige die Aussage. Ich widerlege die Aussage.**Lösung:**

- (a)
- Zeige:**

Sei L regulär. Dann gibt es einen DEA \mathcal{A} mit $\mathcal{L}(\mathcal{A}) = L$, der diese Sprache in polynomial Zeit akzeptiert, somit gilt $L \in P \subseteq NP$.

- (b)
- Widerlege:**

Wähle eine Sprache L , so dass weder L noch L^C kontextfrei sind (z.B. Halteproblem). Dann gilt $L \cup L^C = \Sigma^*$, was jedoch regulär und somit kontextfrei ist.

- (c)
- Widerlege:**

Angenommen L' sei semi-entscheidbar. Dann existiert eine TM T' mit $\mathcal{L}(T') = L'$.Wir konstruieren nun eine TM T wie folgt: Schreibe eine 1 vor die Eingabe w und simuliere nun T' . Akzeptiere genau dann wenn T' akzeptiert, andernfalls gehe in eine Endloschleife.Nun gilt allerdings, dass T die Sprache L^C semi-entscheidet.

$$\begin{aligned} w \in \mathcal{L}(T) &\Leftrightarrow 1w \in \mathcal{L}(T') \\ &\Leftrightarrow w \notin L \\ &\Leftrightarrow w \in L^C \end{aligned}$$

Somit war die Annahme falsch und L' ist nicht semi-entscheidbar.

- (d)
- Zeige:**

- (e) Wie beweisen die Aussage per Widerspruch. Angenommen es gäbe ein j , sodass L_j nicht entscheidbar ist, dann ist das Komplement L_j^C nicht semi-entscheidbar. Nach Voraussetzung wissen wir aber

$$\begin{aligned} \bigcup_{i=1}^k L_i &= \Sigma^* \\ \Leftrightarrow \bigcup_{i=1, i \neq j}^k L_i &= \Sigma^* \setminus L_j \\ \Leftrightarrow \bigcup_{i=1, i \neq j}^k L_i &= L_j^C \end{aligned}$$

Da semi-entscheidbare Sprache unter Vereinigung abgeschlossen sind, ist L_j^C auch semi-entscheidbar. Somit gilt die Annahme nicht und alle L_i für $i \in [k]$ sind entscheidbar.

Problem 6: Abgeschlossenheit

1+2+5+2 = 10 Punkte

Sei Σ ein festes endliches Alphabet. Ein Wort $w = a_1 \dots a_n$ ist eine Teilfolge von $v = b_1 \dots b_m$, falls es Indizes $1 \leq i_1 < i_2 < \dots < i_n \leq m$ gibt, sodass $a_j = b_{i_j}$ für jedes $j \in \{1, \dots, n\}$. Zum Beispiel ist $abac$ eine Teilfolge von $adbdadc$. Wir schreiben $w \preceq v$ falls w eine Teilfolge von v ist.

- (a) Gegeben seien folgende Wörter über dem Alphabet $\Sigma = \{a, b, c\}$. Geben Sie an, welche Wörter keine Teilfolge eines anderen gegebenen Wortes sind. In anderen Worten: geben Sie alle Wörter w an, sodass es kein $w' \neq w$ gibt mit $w \preceq w'$.

acbacb bcccbbb ccacbba cab cbabac

- (b) Der Aufwärtsabschluss von einer Sprache L ist definiert als $\uparrow L := \{w \in \Sigma^* \mid \exists v \in L : v \preceq w\}$. Der Abwärtsabschluss von einer Sprache L ist definiert als $\downarrow L := \{w \in \Sigma^* \mid \exists v \in L : w \preceq v\}$.

Geben Sie für die Sprache $\{a^n b^n \mid n \in \mathbb{N}\}$ jeweils den Aufwärts- und Abwärtsabschluss an.

- (c) Eine Sprache L über dem Alphabet Σ heißt *aufwärts abgeschlossen*, wenn

$$\forall v \in L, \forall w \in \Sigma^* : v \preceq w \implies w \in L.$$

Zeigen Sie, dass jede aufwärts abgeschlossene Sprache regulär ist.

Sie dürfen folgendes Resultat ohne Beweis benutzen:

Higman's Lemma: Für jede unendliche Folge an Wörtern $(w_i)_{i \in \mathbb{N}}$ über einem endlichen Alphabet Σ , gibt es je zwei Indizes $i < j$, sodass $w_i \preceq w_j$ gilt.

- (d) Eine Sprache L über dem Alphabet Σ heißt *abwärts abgeschlossen*, wenn

$$\forall v \in L, \forall w \in \Sigma^* : w \preceq v \implies w \in L.$$

Zeigen Sie, dass jede abwärts abgeschlossene Sprache regulär ist.

Lösung:

- (a) Worte die keine Teilfolge eines anderen Wortes sind, sind $\{acbacb, bcccbbb, ccacbba, cbabac\}$.

- (b) Der Aufwärtsabschluss der Sprache ist gegeben durch $L((a \cup b)^* ab (a \cup b)^*)$.

Der Abwärtsabschluss der Sprache ist gegeben durch $L(a^* b^*)$

- (c) Gegeben einem Wort $w \in \Sigma$ können wir einen regulären Ausdruck konstruieren, der alle Worte akzeptiert, von denen w ein Teilwort ist. Hierzu sei $\alpha(w) = \Sigma^* w_1 \Sigma^* \dots \Sigma^* w_n \Sigma^*$.

Mit Higman's Lemma wissen wir, dass L nur endlich viele Worte enthalten kann, die keine Teilwort von einander sind. Wir nennen diese endliche Menge an Wörtern W . Wir konstruieren nun den regulären Ausdruck $A = \cup_{w \in W} \alpha(w)$. Dieser erkennt L , da jedes Wort $w \in L$ entweder in W enthalten ist oder es gibt ein $w' \in W$, so dass $w' \preceq w$. Somit ist L regulär.

- (d) Wir betrachten das Komplement L^C . Für jedes Wort $w \in L$ gilt, dass jedes Teilwort v von w ebenfalls in der Sprache L ist. Betrachten wir nun ein Wort $w' \in L^C$. Gäbe es v , sodass $w' \preceq v$ wobei $v \in L$, dann gilt nach Abwärtsabgeschlossenheit, dass $w' \in L$, ein Widerspruch. Folglich gilt für jedes v mit $w' \preceq v$, dass $v \notin L \iff v \in L^C$. Also ist das Komplement einer Abwärtsabgeschlossenen Sprache Aufwärtsabgeschlossen.

Da reguläre Sprachen unter Komplementbildung abgeschlossen sind, ist mit Aufgabenteil (c) also auch L regulär.

Die folgenden Probleme können Sie als NP-vollständig annehmen und für Reduktionen verwenden.

CLIQUE

Gegeben: Ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k < |V|$.

Problem: Gibt es eine Clique der Größe mindestens k ?

SUBSET SUM

Gegeben: Zahlen $a_1 \dots a_n \in \mathbb{N}$, und Wert $t \in \mathbb{N}$.

Problem: Gibt es $I \subseteq \{1, \dots, n\}$, sodass $\sum_{i \in I} a_i = t$?

HAMILTON-KREIS

Gegeben: Ein ungerichteter Graph $G = (V, E)$.

Problem: Gibt es einen Kreis in G , der jeden Knoten genau einmal besucht?

3-COLOR

Gegeben: Ein ungerichteter Graph $G = (V, E)$.

Problem: Gibt es eine Funktion $c : V \rightarrow \{1, 2, 3\}$, sodass für jede Kante $\{u, v\}$ gilt $c(u) \neq c(v)$?

PARTITION

Gegeben: Zahlen $a_1 \dots a_n \in \mathbb{N}$.

Problem: Gibt es $I \subseteq \{1, \dots, n\}$, sodass $\sum_{i \in I} a_i = \sum_{j \notin I} a_j$?

3-SAT

Gegeben: Menge U an aussagenlogischen Variablen und eine Menge C von Klauseln mit genau drei Literalen.

Problem: Gibt es eine Wahrheitsbelegung der Variablen aus U , sodass alle Klauseln aus C erfüllt sind?

SAT

Gegeben: Menge U an aussagenlogischen Variablen und eine Menge C von Klauseln.

Problem: Gibt es eine Wahrheitsbelegung der Variablen aus U , sodass alle Klauseln aus C erfüllt sind?

SET COVER

Gegeben: Eine Grundmenge M , Teilmengen $U_1, \dots, U_m \subseteq M$ mit $\bigcup_{i=1}^m U_i = M$, sowie eine Zahl c .

Problem: Gibt es eine Indexmenge $I \subseteq \{1, \dots, m\}$ mit $|I| \leq c$ und $\bigcup_{i \in I} U_i = M$?

DOMINATING SET

Gegeben: Ein Graph $G = (V, E)$ und eine Zahl k .

Gesucht: Eine Teilmenge $D \subseteq V$, sodass $|D| \leq k$ und für alle Knoten $v \in V$ gilt $v \in D$ oder v hat einen Nachbarn $u \in D$, also $\{u, v\} \in E$.