Universität Karlsruhe
Fakultät für Informatik
ILKD Prof. Dr. D. Wagner

Theoretische Informatik
WS 2004/05
5. April 2005

2. Klausur zur Vorlesung
Informatik III
Wintersemester 2004/2005

Lösung!

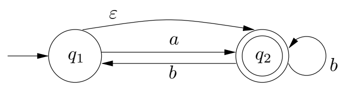
Beachten Sie:

- Bringen Sie Ihren *Aufkleber* auf diesem Deckblatt an, und beschriften Sie *jedes weitere Blatt* mit Ihrem Namen und Ihrer Matrikelnummer.
- Schreiben Sie Ihre Lösungen auf die Aufgabenblätter und Rückseiten. Zusätzliches Papier erhalten Sie bei Bedarf von der Aufsicht.
- Zum Bestehen der Klausur sind **20** der möglichen **60** Punkte hinreichend.
- Es sind keine Hilfsmittel zugelassen.

Aufgabe	Mögliche Punkte					Erreichte Punkte				
	a	b	c	d	Σ	a	b	c	d	Σ
1	3	5	4	-	12				-	
2	2	3	4	3	12					
3	4	4	4	-	12				-	
4	3	4	5	-	12				-	
5	12 × 1				12					
Σ					60					

Aufgabe 1:
$(3 + 5 + 4 = 12$ Punkte)

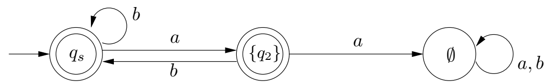
(a) Bestimmen Sie mit Hilfe einer systematischen Konstruktion einen zu folgendem NEA $\mathcal{A}$ äquivalenten DEA, und geben Sie diesen als Übergangsdiagramm an.



Lösung:

	a	b
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_1, q_2\}$
$\{q_2\}$	$\emptyset$	$\{q_1, q_2\}$
$\emptyset$	$\emptyset$	$\emptyset$

Im folgenden Diagramm stehe q_s für $\{q_1, q_2\}$.



(b) Bestimmen Sie mit Hilfe einer systematischen Konstruktion die von $\mathcal{A}$ akzeptierte Sprache, und geben Sie diese als möglichst einfachen regulären Ausdruck an.

Lösung: Um besserer Lesbarkeit willen stehe im Folgenden $L_{m,x,n}$ für L_{q_m,x,q_n} $(m, n \in \{1, 2\}, x \in \{0, 1, 2\})$.

$$\begin{aligned}
L_{1,0,1} &= \varepsilon \\
L_{1,0,2} &= \varepsilon \cup a \\
L_{2,0,1} &= b \\
L_{2,0,2} &= \varepsilon \cup b \\
L_{1,1,1} &= \varepsilon \cup \varepsilon\varepsilon^*\varepsilon = \varepsilon \\
L_{1,1,2} &= \varepsilon \cup a \cup \varepsilon\varepsilon^*(\varepsilon \cup a) = \varepsilon \cup a \\
L_{2,1,1} &= b \cup b\varepsilon^*\varepsilon = b \\
L_{2,1,2} &= \varepsilon \cup b \cup b\varepsilon^*(\varepsilon \cup a) = \varepsilon \cup b(\varepsilon \cup a) \\
L_{1,2,2} &= \varepsilon \cup a \cup (\varepsilon \cup a)(\varepsilon \cup b(\varepsilon \cup a))^*(\varepsilon \cup b(\varepsilon \cup a)) = \varepsilon \cup a \cup (\varepsilon \cup a)(\varepsilon \cup b(\varepsilon \cup a))^+ \\
&= (\varepsilon \cup a)(b(\varepsilon \cup a))^* = L(\mathcal{A})
\end{aligned}$$

(c) Zeigen Sie, dass die Sprache $L = \{0^{2^n} \mid n \geq 1\}$ nicht regulär ist.

Lösung: Wählen zu $n \in \mathbb{N}$ das Wort $w = 0^{2^n} \in L$ ($|w| \geq n$ ✓). Bei allen Zerlegungen $w = uvx$ mit $|uv| \leq n$ und $v \neq \varepsilon$ enthält v mindestens eine und höchstens n Nullen.

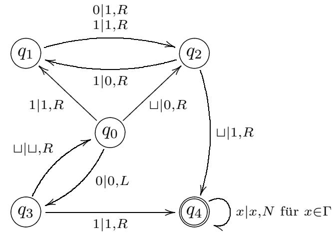
Wählen $i := 2$. Dann hat $w' = uv^ix = 0^{2^n+|v|}$ Länge kleiner 2^{n+1}:

$$2^n + |v| < 2^{n+1} \iff |v| < 2^n \quad \checkmark$$

Somit ist w' nicht in L, also L nicht regulär.

Aufgabe 2: (2 + 3 + 4 + 3 = 12 Punkte)

Betrachten Sie die Turingmaschine $\mathcal{M}$ mit Zustandsmenge $Q = \{q_0, \dots, q_4\}$, Eingabealphabet $\Sigma = \{0, 1\}$, Bandalphabet $\Gamma = \Sigma \cup \{\sqcup\}$, Startzustand q_0, Endzustandsmenge $F = \{q_4\}$ und folgender partieller Übergangsfunktion δ:



(a) Geben Sie die von $\mathcal{M}$ durchlaufenen Konfigurationen bei Abarbeitung des Wortes 1011 an.

Geben Sie ein Wort an, bei dessen Eingabe $\mathcal{M}$ nicht stoppt.

Lösung: $\sqcup(q_0)1011$, $1(q_1)011$, $11(q_2)11$, $110(q_1)1$, $1101(q_2)\sqcup$, $11011(q_4)\sqcup$.

Jedes Wort, das mit 0 beginnt.

(b) Zeigen Sie, dass die von $\mathcal{M}$ akzeptierte Sprache $L(\mathcal{M})$ kontextfrei ist.

Lösung: Keine akzeptierende Berechnung benutzt den Zustand q_3. $\mathcal{M}$ ohne q_3 ist aber eine Rechtsturingmaschine. Da Rechtsturingmaschinen genau die Menge der regulären Sprachen akzeptieren, ist $L(\mathcal{M})$ regulär und insbesondere kontextfrei.

Alternative: Per Induktion lässt sich zeigen, dass

$$L(\mathcal{M}) = \varepsilon \cup 1(0 \cup 1)[1(0 \cup 1)]^* = (1(0 \cup 1))^*$$

gilt. Da sich $L(\mathcal{M})$ durch einen regulären Ausdruck beschreiben lässt, ist $L(\mathcal{M})$ regulär und damit auch kontextfrei.

(c) Modifizieren Sie $\mathcal{M}$ zu $\mathcal{M}'$ so, dass folgende Eigenschaften erfüllt sind:

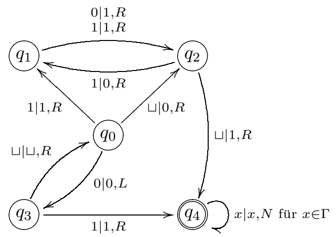
(1) $\mathcal{M}'$ hält stets,

(2) $L(\mathcal{M}) = L(\mathcal{M}')$,

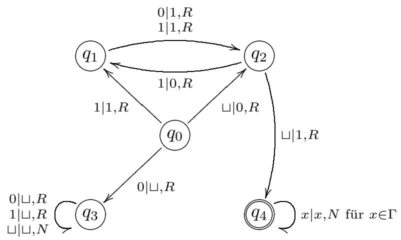
(3) die Ausgaben von $\mathcal{M}$ und $\mathcal{M}'$ sind für die Eingaben aus $L(\mathcal{M})$ identisch und

(4) sei $L' := \{w \in \Sigma^* : \mathcal{M} \text{ hält bei Eingabe } w \text{ nicht}\}$, dann soll die Ausgabe von $\mathcal{M}'$ bei Eingaben aus L' in $L(\mathcal{M})$ sein.

Hinweis: Zur besseren Übersicht sei nachfolgend noch einmal das Übergangsdiagramm von $\mathcal{M}$ gegeben.



Lösung:



Eigenschaften:

(1) $\mathcal{M}'$ terminiert stets, da $\mathcal{M}$ nur bei Wörtern, die mit 0 beginnen, nicht terminiert und diese nun gelöscht werden.

(2) $L(\mathcal{M}) = L(\mathcal{M}')$, da sich die Endzustandsmenge nicht geändert hat und q_4 nicht mit 'neuen' Wörtern erreichbar ist; für Eingaben aus $L(\mathcal{M})$ endet $\mathcal{M}'$ immer noch in q_4.

(3) Die Ausgaben von $\mathcal{M}$ und $\mathcal{M}'$ sind für Eingaben aus $L(\mathcal{M})$ identisch, da der Übergang $\delta(q_0, 0)$ für die Akzeptanz von $\mathcal{M}$ unerheblich ist und die Übergänge bzgl. q_1, q_2 und q_4 sich nicht geändert haben.

(4) Sei $L' := \{w \in \Sigma^* : \mathcal{M}$ hält bei Eingabe w nicht$\}$, dann soll die Ausgabe von $\mathcal{M}'$ bei Eingaben aus L' in $L(\mathcal{M})$ sein. Klar, da $L' = 0(0 \cup 1)^*$; für diese Wörter wird die Eingabe gelöscht, und es gilt: $\varepsilon \in L(\mathcal{M})$.

(d) Seien $L_1, \ldots, L_k$ semientscheidbare Sprachen über einem Alphabet Σ so, dass folgende Eigenschaften erfüllt sind:

(i) für alle $i \neq j$ ist $L_i \cap L_j = \emptyset$ und

(ii) $\bigcup_{i=1}^{k} L_i = \Sigma^*$.

Zeigen Sie, dass die Sprachen L_i (für $i \in \{1, \ldots, k\}$) entscheidbar sind.

Lösung: Da L_i $(i = 1, \ldots, k)$ semientscheidbar ist, existiert eine Turingmaschine $\mathcal{M}_i$, die bei Eingabe $w \in \Sigma^*$ (akzeptierend) hält, falls $w \in L_i$.

Wir können eine Sprache L_i nun wie folgt entscheiden: Sei $w \in \Sigma^*$. Konstruieren eine Turingmaschine $\mathcal{M}^*$, die ‚parallel' alle $\mathcal{M}_i$ mit Eingabe w simuliert, genau dann hält, wenn eine der Maschinen akzeptiert, und genau dann akzeptiert, wenn $\mathcal{M}_i$ akzeptiert.

Falls $w \in L_i$, so wird w nur von $\mathcal{M}_i$ akzeptiert (wegen (i)); falls $w \notin L_i$, so akzeptiert $\mathcal{M}_i$ nicht und es gibt wegen (ii) eine Maschine $\mathcal{M}_j$ $(j \neq i)$, die w akzeptiert. Insbesondere hält $\mathcal{M}^*$ stets.

Aufgabe 3: (4 + 4 + 4 = 12 Punkte)

(a) **Problem HAMILTONKREIS (HC)**

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

Frage: Existiert ein einfacher Kreis in G, der alle Knoten in V enthält, also eine Folge $(v_1, \ldots, v_n)$ von paarweise verschiedenen Knoten $v_i \in V$ $(i = 1, \ldots, n)$ mit $n := |V|$ und $\{v_j, v_{j+1}\}, \{v_n, v_1\} \in E$ $(j = 1, \ldots, n-1)$?

Hinweis: HC ist $\mathcal{NP}$-vollständig.

Problem RURAL POSTMAN (RP)

Gegeben: Graph $G = (V, E)$, Teilmenge $\tilde{E} \subseteq E$ und Parameter $K \in \mathbb{N}$.

Frage: Existiert ein (nicht notwendigerweise einfacher) Kreis der Länge höchstens K in G, der alle Kanten aus $\tilde{E}$ enthält, also eine Folge $(v_1, \ldots, v_\ell)$ mit $v_i \in V$ $(i = 1, \ldots, \ell)$, $\ell \leq K$ und $\tilde{E} \subseteq \big\{\{v_j, v_{j+1}\} \mid j = 1, \ldots, \ell - 1\big\} \cup \big\{\{v_\ell, v_1\}\big\} \subseteq E$?

Hinweis: Der Graph G ist nicht notwendig schleifenfrei, d. h. er kann auch Kanten $\{v, v\}$ enthalten.

Zeigen Sie die $\mathcal{NP}$-Vollständigkeit von RP.

Lösung:

- RP $\in \mathcal{NP}$:

  Eine zu einer Instanz $I = (G = (V, E), \tilde{E}, K)$ gegebene Folge von Knoten aus V kann mit Aufwand $\mathcal{O}(K \cdot |E| \cdot |\tilde{E}|)$, also in polynomieller Zeit bzgl. der Eingabelänge von I, dahingehend überprüft werden, ob sie einen Kreis in G induziert, Länge höchstens K besitzt und ob alle Kanten in $\tilde{E}$ darin enthalten sind.

- HC $\propto$ RP:

  Konstruieren zu einer Instanz $I = G$ von HC mit $G = (V, E)$ eine Instanz $I' = (G', \tilde{E}, 2|V|)$ von RP mit $G' = (V, E \cup \tilde{E})$ und $\tilde{E} = \{\{v, v\} : v \in V\}$. Diese Konstruktion ist mit Aufwand $\mathcal{O}(|V|)$, also in polynomieller Zeit, durchführbar.

  Falls I eine Ja-Instanz ist, so induziert ein Hamiltonkreis $(v_1, \ldots, v_n)$ in G eine Lösung von I', die aus dem Hamiltonkreis ‚mit dazwischengeschobenen Schleifen' besteht, also:

  $$(v_1, v_1, v_2, v_2, \ldots, v_n, v_n)\,.$$

  Diese Folge von Knoten induziert offenbar einen Kreis der Länge $2|V|$ in G' und enthält alle Kanten aus $\tilde{E}$.

  Umgekehrt induziert eine Lösung von I' direkt einen Hamiltonkreis, da jeder Kreis in G', der Länge höchstens $2|V|$ besitzt und alle Schleifen beinhaltet, darüber hinaus nur Kanten enthalten kann, die alle Knoten in einer gewissen Reihenfolge miteinander verbinden.

(b) **Problem SUBSET PRODUCT**

Gegeben: Zahlen $a_1, \ldots, a_n \in \mathbb{N} \setminus \{1\}$ und Parameter $K \in \mathbb{N}$.

Frage: Existiert eine Indexmenge $J \subseteq \{1, \ldots, n\}$ mit $\prod_{j \in J} a_j = K$?

Beschreiben Sie einen pseudopolynomiellen Algorithmus für SUBSET PRODUCT, der auf dem Prinzip der dynamischen Programmierung beruht.

Lösung: Berechne mittels dynamischer Programmierung die folgende Tabelle:

$$W(i, p) := \exists J \subseteq \{1, \ldots, i\} \colon \prod_{j \in J} a_j = p$$

Diese kann wie folgt aufgebaut werden:

- setze $W(i, 1)$ auf `wahr` für $1 \leq i \leq n$
- setze $W(1, p) = (p = 1 \vee a_1 = p)$
- setze $W(i+1, p)$ wie folgt:

$$W(i+1, p) = W(i, p) \vee \begin{cases} W(i, \frac{p}{a_{i+1}}) & \text{, falls } \frac{p}{a_{i+1}} \in \mathbb{N} \\ \texttt{false} & \text{, sonst} \end{cases}$$

Diese Tabelle kann in $\mathcal{O}(n \cdot \prod_i a_i) \subseteq \mathcal{O}(n^2 \cdot \max_i a_i)$ berechnet werden. Dies ist polynomial in der Eingabelänge bei unärer Kodierung.

Der Algorithmus berechnet zuerst die Tabelle und prüft anschließend, ob $W(n, K)$ wahr ist. Ist dies der Fall, so wird die Instanz als JA-Instanz akzeptiert, andernfalls wird sie als NEIN-Instanz verworfen.

Hinweis zur Korrektheit: Wenn es eine Teilmenge gibt, so findet sich diese auch mit Hilfe der Tabelle. (Betrachte die Elemente von J aufsteigend sortiert, etwa $j_1 < j_2 < \cdots < j_k$. Dann müssen folgende Einträge $W(j_i, a_{j_1} \cdot \ldots \cdot a_{j_i})$ wahr sein. Damit ergibt sich, dass $W(j_k, K)$ wahr ist, also auch $W(n, K)$.) Andernfalls kann keine Teilmenge J die Eigenschaft haben.

(c) Zeigen Sie, dass co-INDEPENDENT SET $\in \mathcal{DTAPE}(n)$, wobei n die Eingabelänge einer co-INDEPENDENT SET-Instanz bezeichne.

Lösung: Idee: Aufzählen aller k-elementigen Knotenteilmengen.

Sei $I := (G := (V := \{v_1, \ldots, v_k\}, E), K)$ eine beliebige Instanz von co-INDEPENDENT SET. Falls G keine unabhängige Menge mit mindestens K Elementen enthält, ist I eine JA-Instanz.

Auf linearem Platz kann ein beliebiger k-Vektor w über $\{0,1\}$ dargestellt werden, wobei $w \in \{0,1\}^k$ einer Knotenteilmenge entsprechen soll:

$$w(i) = 1 \iff v_i \text{ ist in der Teilmenge enthalten}$$

Die Verifikation, ob w für I eine unabhängige Menge mit mindestens K Knoten ist, kann mit $\log K$ Band-Speicherplatz durchgeführt werden (zusätzliche Hilfsvariablen und deren Zustände können direkt in die TM eincodiert werden):

- initialisiere w mit $(0, 0, \ldots, 0)$
- überprüfe, ob w eine unabhängige Menge der Größe K (oder größer) repräsentiert
- falls ja, dann ist w eine unabhängige Menge mit mindestens K Knoten und damit ist I eine NEIN-Instanz
- falls nein, erhöhe w um eins (als Bitvektor) und teste erneut, ob w eine unabhängige Menge der Größe K oder größer repräsentiert
- falls alle Bitvektoren keine geeignete unabhängige Menge ergeben haben, dann ist I eine JA-Instanz

Es ist nun leicht einzusehen, dass dieses Verfahren I richtig entscheidet. Falls es eine unabhängige Menge der Größe K (oder größer) gibt, wird diese beim Aufzählen gefunden und I kann keine JA-Instanz sein (da es ein Gegenbeispiel gibt). Andernfalls gibt es keine solche unabhängige Menge und I ist eine JA-Instanz.

Aufgabe 4: (3 + 4 + 5 = 12 Punkte)

Gegeben sei eine Grammatik $G = (\Sigma := \{0,1\}, V := \{S, A, B\}, S, R)$ mit

$$R := \{\ S \to 0A \mid 1B, \quad A \to 1 \mid 1S, \quad B \to 0 \mid 0S\ \}\ .$$

(a) Ist die Grammatik G in Greibach-Normalform? (Begründen Sie Ihre Antwort.)

Geben Sie mit Hilfe des Verfahrens aus der Vorlesung einen Kellerautomaten als vollständiges 7-Tupel $(Q, \Sigma', \Gamma, \delta, q_0, Z_0, F)$ an, der genau $L(G)$ mit leerem Stack akzeptiert.

Lösung: Ja, G ist in Greibach-Normalform, da alle Ableitungsregeln die Form $A \to a\alpha$ mit $A \in V$, $a \in \Sigma$ und $\alpha \in V^*$ haben.

$\mathcal{A} = (\{q\}, \Sigma, V, \delta, q, S)$ mit $\delta(q, a, A) := \{(q, \alpha) : A \to a\alpha\}$ genügt.

(b) Bringen Sie die Grammatik G in Chomsky-Normalform und entscheiden Sie mit Hilfe des Cocke-Younger-Kasami-Algorithmus, ob das Wort 0110 in $L(G)$ enthalten ist.

Lösung: Chomsky-Normalform: $G' = (\Sigma, V \cup \{Y_0, Y_1\}, S, R')$ mit

$$R' := \{\ S \to Y_0A \mid Y_1B, \quad A \to 1 \mid Y_1S, \quad B \to 0 \mid Y_0S, \quad Y_x \to x \text{ für } x \in \Sigma\ \}$$

0	1	1	0
$\{Y_0, B\}$	$\{Y_1, A\}$	$\{Y_1, A\}$	$\{Y_0, B\}$
S	$\emptyset$	S	
$\emptyset$	A		
S			

Damit ist $0110 \in L(G)$.

(c) Geben Sie die Sprache $L(G)$ an und zeigen Sie anschließend für $L(G)$ *explizit*, dass die im Pumping-Lemma (für kontextfreie Sprachen) formulierte notwendige Bedingung für die Kontextfreiheit erfüllt ist. (Geben Sie also eine konkrete 'Belegung' für die existenzquantifizierten Ausdrücke an, und begründen Sie.)

Lösung: $L(G) = (01 \cup 10)^+$. (Beweis per Induktion.)

Setze $n := 4$. Sei $z \in L$ mit $|z| \geq n$ und $z = z_1 \ldots z_k$ mit $z_i \in \Sigma$. Dann benutze folgende Zerlegung:

- $u = v = w = \varepsilon$,
- $x = z_1 z_2$ und
- $y = z_3 \ldots z_k$.

Da $|z| \geq 4$, ist y nicht das leere Wort und damit $y \in L(G)$. Ebenso ist $x \in L$ und $x^i \in L$ für $i \geq 1$. Damit ist $uv^iwx^iy = x^iy \in L$ für $i \geq 0$.

Aufgabe 5: (12 × 1 = 12 Punkte)

Kreuzen Sie für folgende Aussagen an, ob diese wahr oder falsch sind.
Hinweis: Für jede richtige Antwort gibt es einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Es wird keine negative Gesamtpunktzahl für diese Aufgabe gegeben.

Sei $\mathcal{A}$ ein NEA über einem Alphabet Σ. Dann akzeptiert $\mathcal{A}$ ein Wort $w \in \Sigma^*$ genau dann, wenn $\mathcal{A}$ bei der Abarbeitung von w mindestens einen Endzustand benutzt. ☐ wahr ☐ falsch

Lösung: ☐ wahr ☒ falsch

Es existiert ein endlicher Automat, der die durch die Grammatik $(\{a, b, c\}, \{S, A, B, C\}, S, R)$ mit

$$R = \{S \longrightarrow ABC,\ A \longrightarrow C,\ A \longrightarrow AA \mid a,\ B \longrightarrow BB \mid b,\ C \longrightarrow CC \mid c\}$$

erzeugte Sprache erkennt. ☐ wahr ☐ falsch

Lösung: ☒ wahr ☐ falsch

Die Sprache $L = \bigcup_{n\in\mathbb{N}} \{0^{pn} \mid p \text{ prim}\} \subseteq \{0\}^*$ hat endlichen Nerodeindex. ☐ wahr ☐ falsch

Lösung: ☒ wahr ☐ falsch

Seien f_1 und f_2 turingberechenbare Funktionen und $\mathcal{M}$ eine deterministische Turingmaschine. Dann existiert keine nicht-deterministische Turingmaschine, die entscheidet, ob $\mathcal{M}$ Funktion f_1 oder f_2 berechnet. ☐ wahr ☐ falsch

Lösung: ☒ wahr ☐ falsch

Sei L eine semientscheidbare Sprache über dem Alphabet $\{0, 1\}$. Dann ist $\chi^*_{L\cap L_u}$ berechenbar, wobei L_u die universelle Sprache bezeichne. ☐ wahr ☐ falsch

Lösung: ☒ wahr ☐ falsch

Sei L eine Sprache über dem Alphabet $\{0, 1\}$ und $w \in \{0, 1\}^*$. Dann gibt es eine Gödelnummer g so, dass die universelle Turingmaschine die Eingabe (g, w) genau dann akzeptiert, falls $w \in L$. ☐ wahr ☐ falsch

Lösung: ☐ wahr ☒ falsch

Es gilt: co-2SAT $\notin \mathcal{NP}$. ☐ wahr ☐ falsch

Lösung: ☐ wahr ☒ falsch

Falls $\mathcal{P} \neq \mathcal{NP}$, dann kann mit Hilfe eines Algorithmus für SET COVER eine Instanz von INDEPENDENT SET in polynomieller Gesamtzeit gelöst werden. ☐ wahr ☐ falsch

Lösung: ☐ wahr ☒ falsch

Falls es ein PAS für COLOR gibt, dann ist $\mathcal{P} = \mathcal{NP}$. ☐ wahr ☐ falsch

Lösung: ☒ wahr ☐ falsch

Jede Sprache, die durch einen nicht-deterministischen endlichen Automaten erkannt wird, kann auch durch einen deterministischen Kellerautomaten erkannt werden. ☐ wahr ☐ falsch

Lösung: ☒ wahr ☐ falsch

Die Sprache

$$\Big[\big(\{a^n cb^n cd^k : k, n \in \mathbb{N}\} \cap a^*cb^*cd^*\big) \cup a^*cb^*\Big] \cap \Big[a^*c \cdot \{b^n cd^n : n \in \mathbb{N}\}\Big]$$

ist kontextfrei. ☐ wahr ☐ falsch

Lösung: ☐ wahr ☒ falsch

Sei G eine kontextfreie Grammatik und α ein regulärer Ausdruck. Es ist nicht entscheidbar, ob $L(G) \cap L(\alpha) = \emptyset$ ist. ☐ wahr ☐ falsch

Lösung: ☐ wahr ☒ falsch