Institut für Informatik der Ludwig-Maximilians-Universität München
Prof. Dr. Jasmin Blanchette
Elisabeth Lempa
Luca Maio

SoSe 2025
30.09.2025

Wiederholungsklausur zur Vorlesung **Formale Sprachen und Komplexität**

Die Bearbeitungszeit beträgt **120 Minuten**. Hilfsmittel sind nicht erlaubt, auch das Mitführen ausgeschalteter elektronischer Geräte (inklusive Smartwatches aber exklusive normaler Uhren) wird als Täuschungsversuch gewertet. Schreiben Sie Ihren vollständigen Namen und Ihre Matrikelnummer deutlich lesbar auf dieses Deckblatt, sowie Ihren Namen in die Kopfzeile auf jedem Blatt der Klausurangabe. Geben Sie alle Blätter ab. Lassen Sie diese zusammengeheftet. Verwenden Sie nur **dokumentenechte Stifte** und **weder** die Farbe **rot noch grün**.

Kontrollieren Sie, ob Sie alle Aufgabenblätter erhalten haben. Aufgabenstellungen befinden sich auf den **Seiten 1–11**. Sie dürfen die Rückseiten für Nebenrechnungen nutzen. Falls Sie die Rückseiten für Antworten nutzen, so markieren Sie klar, was zu welcher Aufgabe gehört und geben Sie in der entsprechenden Aufgabe an, wo alle Teile Ihrer Antwort zu finden sind. Streichen Sie alles durch, was nicht korrigiert werden soll.

Es gibt 5 unterschiedlich gewichtete Aufgaben zu insgesamt 100 Punkten. Mit 50 Punkten haben Sie sicher bestanden. Die Teilaufgaben können unabhängig voneinander bearbeitet werden.

Mit Ihrer Unterschrift bestätigen Sie, dass Sie zu Beginn der Klausur in ausreichend guter gesundheitlicher Verfassung sind und diese Klausurprüfung verbindlich annehmen.

Nachname (in GROSSBUCHSTABEN):

Vorname (in GROSSBUCHSTABEN):

Matrikelnummer:

Studiengang:

Hiermit erkläre ich die Richtigkeit der obigen Angaben:

Unterschrift

Die folgende Tabelle nicht ausfüllen:

Aufgabe	1	2	3	4	5	$\sum$
Punkte	28	18	20	18	16	100
Erreicht						

Aufgabe 1 (Reguläre Sprachen): **(28 Punkte)**

a) Die Sprache L_1 sei definiert als die Menge aller Wörter über dem Alphabet $\{a, b, c\}$, die genau ein a und zwei b's enthalten.

Geben Sie einen regulären Ausdruck an, der L_1 erzeugt.

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 1:

b) Sei L_2 die Sprache über dem Alphabet $\{a, b, c\}$, die vom regulären Ausdruck

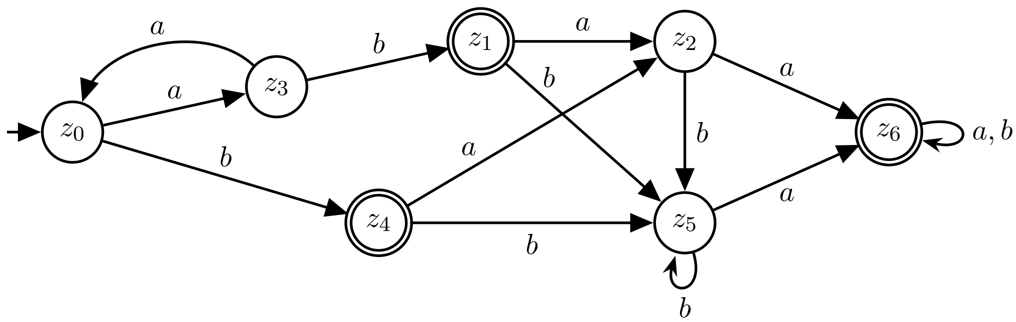
$$(abc)^*(cba^* \mid \varepsilon)$$

erzeugt wird. Geben Sie den Zustandsgraphen eines nichtdeterministischen endlichen Automaten (ohne ε-Übergänge) an, der L_2 akzeptiert.

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 1:

c) Minimieren Sie den folgenden deterministischen endlichen Automaten (d. h. konstruieren Sie einen deterministischen endlichen Automaten, der die gleiche Sprache akzeptiert und eine minimale Anzahl an Zuständen benutzt). Nutzen Sie dazu das Verfahren aus der Vorlesung. Geben Sie eine Partitionstabelle und den minimierten Automaten als Zustandsgraphen an.



Aufgabe 2 (Nicht reguläre Sprachen): **(18 Punkte)**

a) Zeigen Sie mit dem Pumping-Lemma für reguläre Sprachen, dass die Sprache

$$L_1 = \{a^i b^i c^i d^i \mid i \in \mathbb{N}\}$$

über dem Alphabet $\{a, b, c, d\}$ nicht regulär ist.

Zur Erinnerung:

Eine Sprache L hat die Pumping-Eigenschaft, wenn es eine Zahl $n \in \mathbb{N}_{>0}$ gibt, sodass jedes Wort $z \in L$, welches Mindestlänge n hat, als $z = uvw$ geschrieben werden kann, sodass $|uv| \leq n$, $|v| \geq 1$ und für alle $i \in \mathbb{N}$: $uv^iw \in L$.

Das Pumping-Lemma für reguläre Sprachen besagt, dass jede reguläre Sprache die Pumping-Eigenschaft hat.

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 2:

b) Beweisen Sie mithilfe der Abschlusseigenschaften der regulären Sprachen, dass die Sprache

$$L_2 = \overline{\{e^j a^i b^i c^i d^i e^j \mid i, j \in \mathbb{N}\}}$$

über dem Alphabet $\{a, b, c, d, e\}$ nicht regulär ist. $\overline{L}$ bezeichnet dabei das Komplement einer Sprache L. Sie dürfen annehmen, dass die Sprache $L_1 = \{a^i b^i c^i d^i \mid i \in \mathbb{N}\}$ aus Teilfrage a) nicht regulär ist.

Zur Erinnerung: Die regulären Sprachen sind unter Vereinigung, Schnitt, Komplement, Produkt und Kleeneschem Abschluss abgeschlossen.

Aufgabe 3 (Kontextfreie und kontextsensitive Sprachen): (20 Punkte)

a) Die Sprache L_1 über dem Alphabet $\{a, b\}$ sei definiert als

$$L_1 = \{ab^iab^ja \mid i, j \in \mathbb{N}, i < j\}$$

Geben Sie eine kontextfreie Grammatik G_1 als 4-Tupel an, die L_1 erzeugt. Die Grammatik darf keine ε-Produktionen enthalten. Erläutern Sie, warum G_1 die Sprache L_1 erzeugt. Beschreiben Sie beispielsweise, welche „Aufgabe" die einzelnen Nichtterminale bei der Erzeugung übernehmen.

Geben Sie zusätzlich eine Linksableitung für das Wort *ababbba* für Ihre Grammatik an.

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 3:

b) Sei $G_2 = (V_2, \Sigma_2, P_2, S)$ eine kontextsensitive Grammatik mit $V_2 = \{S, B, A\}$, $\Sigma_2 = \{a, b, c\}$ und

$$P_2 = \{S \to ABSc,\ S \to ABc,\ BA \to AB,\ Bc \to bc,\ Bb \to bb,\ Ab \to ab,\ Aa \to aa\}$$

Um das Wortproblem für G_2 zu entscheiden, berechnen wir Mengen L_i^n mit $i \in \mathbb{N}$, $n \in \mathbb{N}_{>0}$, wobei

$$L_i^n = \{w \in (V \cup \Sigma)^* \mid |w| \le n \text{ und } S \Rightarrow_{G_2}^k w \text{ mit } k \le i\}$$

Berechnen Sie L_i^5 für alle $i \in \mathbb{N}$. Sie müssen keine Begründung angeben.

Geben Sie zusätzlich anhand Ihrer Berechnung ein Wort über Σ_2 an, das in $L(G_2)$ enthalten ist.

Aufgabe 4 (Berechenbarkeit und Entscheidbarkeit): **(18 Punkte)**

a) Für eine Funktion f steht μf für die Anwendung des μ-Operators auf f. Berechnen Sie μg_i für folgende Funktionen g_i.

(i) $g_1 : \mathbb{N} \to \mathbb{N},\ g_1(x) = 50$

(ii) $g_2 : \mathbb{N} \to \mathbb{N},\ g_2(x) = 3x$

(iii) $g_3 : \mathbb{N} \to \mathbb{N},\ g_3(x) = 5x + 1$

(iv) $g_4 : \mathbb{N} \to \mathbb{N},\ g_4(x) = x^2$

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 4:

b) Der Satz von Rice besagt:

> Sei $\mathcal{R}$ die Klasse aller turingberechenbaren Funktionen. Sei $\mathcal{S}$ eine nichtleere echte Teilmenge von $\mathcal{R}$. Dann ist folgende Sprache unentscheidbar:
>
> $$C(\mathcal{S}) = \{w \mid \text{die von der deterministischen Turingmaschine } M_w \text{ berechnete Funktion liegt in } \mathcal{S}\}$$
>
> wobei M_w die Turingmaschine mit der Gödelnummer w bezeichnet.

Wenden Sie für jede der beiden Aussagen unten den Satz von Rice an, um sie zu beweisen, oder erklären Sie, warum der Satz nicht anwendbar ist.

(i) Es ist unentscheidbar, ob eine gegebene deterministische Turingmaschine für jede Eingabe die Zahl 36 berechnet.

(ii) Es ist unentscheidbar, ob eine gegebene deterministische Turingmaschine für jede Eingabe $i \in \mathbb{N}$ eine Zahl $j > i$ berechnet.

Aufgabe 5 (Komplexität): **(16 Punkte)**

a) Wir erinnern zunächst an die Definition des CLIQUE-Problems:

gegeben:	ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$
gefragt:	Besitzt G eine Clique der Größe mindestens k?

Ein anderes, weniger bekanntes Problem ist FRIENDS:

gegeben:	eine endliche Menge von Personen P, eine endliche Menge von Mengen $\mathbf{K} \subseteq \mathcal{P}(P)$, sodass jede Menge $K \in \mathbf{K}$ einen Freundeskreis darstellt und eine Zahl $m \in \mathbb{N}$
gefragt:	Existiert einen Freundeskreis $K \in \mathbf{K}$, sodass $\lvert K\rvert \geq m$?

Der folgende Beweis ist falsch. Finden Sie und erklären Sie den Fehler.

Satz: $\mathsf{P} = \mathsf{NP}$.

Beweis: Das FRIENDS-Problem ist in $\mathcal{P}$, denn ein Polynomialzeitalgorithmus könnte einfach $\mathbf{K}$ durchlaufen und für jede Menge $K \in \mathbf{K}$ prüfen, ob $\lvert K\rvert \geq m$.

Dennoch ist FRIENDS auch $\mathcal{NP}$-schwer. Der Beweis erfolgt mithilfe einer Polynomialzeit-Reduktion vom $\mathcal{NP}$-schweren Problem CLIQUE auf FRIENDS. Die Reduktionsfunktion ist definiert wie folgt:

$$f((V,E),k) = \big(\underbrace{V}_{P :=}, \underbrace{\{V' \mid V' \text{ ist eine Clique von } (V,E)\}}_{\mathbf{K} :=}, \underbrace{k}_{m :=}\big)$$

Diese Funktion ist offensichtlich total und berechenbar. Sie ist auch korrekt:

$((V,E),k) \in$ CLIQUE
g.d.w. (V,E) besitzt eine Clique der Größe k
g.d.w. es existiert $K \in \{V' \mid V' \text{ ist eine Clique von } (V,E)\}$, sodass $\lvert K\rvert \geq m$
g.d.w. es existiert $K \in \mathbf{K}$, sodass $\lvert K\rvert \geq m$
g.d.w. $(P, \mathbf{K}, m) \in$ FRIENDS.

FRIENDS ist sowohl NP-schwer als auch in P. Dies ist nur dann möglich, wenn $\mathsf{P} = \mathsf{NP}$.

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 5:

b) In der Vorlesung wurde das INDEPENDENT-SET-Problem vorgestellt:

gegeben:	ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$
gefragt:	Besitzt G eine unabhängige Knotenmenge der Größe k?

Für einen ungerichteten Graphen $G = (V, E)$ ist $V' \subseteq V$ eine *unabhängige Knotenmenge*, wenn keine zwei Knoten aus V' über eine Kante verbunden sind, d. h. $u, v \in V'$ impliziert $\{u, v\} \notin E$.

Sie dürfen als bekannt annehmen, dass INDEPENDENT-SET $\mathcal{NP}$-vollständig ist.

Auch für gerichtete Graphen $G = (V, E)$ lassen sich unabhängige Knotenmengen definieren. Für einen gerichteten Graphen $G = (V, E)$ ist $V' \subseteq V$ eine *unabhängige Knotenmenge*, wenn keine zwei Knoten aus V' über eine Kante verbunden sind, d. h. $u, v \in V'$ impliziert $(u, v) \notin E$ und $(v, u) \notin E$.

Nun führen wir folgendes DIRECTED-INDEPENDENT-SET-Problem ein:

gegeben:	ein gerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$
gefragt:	Besitzt G eine unabhängige Knotenmenge der Größe k?

Beweisen Sie mithilfe einer Polynomialzeit-Reduktion, dass DIRECTED-INDEPENDENT-SET $\mathcal{NP}$-schwer ist.