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

SoSe 2024
05.08.2024

Erstklausur zur Vorlesung

Formale Sprachen und Komplexität

Die Bearbeitungszeit beträgt **120 Minuten**. Hilfsmittel sind nicht erlaubt, auch das Mitführen ausgeschalteter elektronischer Geräte wird als Betrug 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 1–12**. 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. 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:

Vorname:

Matrikelnummer:

Studiengang:

☐ **Bitte *nur* ankreuzen, wenn die Klausur entwertet und nicht korrigiert werden soll.**

Please check with an X *only* if the exam should be voided and not graded.

Hiermit erkläre ich die Richtigkeit der obigen Angaben:

____
Unterschrift

Die folgende Tabelle nicht ausfüllen:

Aufgabe	1	2	3	4	5	$\sum$
Punkte	26	18	24	16	16	100
Erreicht						

Aufgabe 1 (Reguläre Sprachen): **(26 Punkte)**

a) Sei L_1 die Sprache über dem Alphabet $\{a, b, c\}$, die vom regulären Ausdruck

$$(ab|ba)(a|b|c)^*(ab|bc)$$

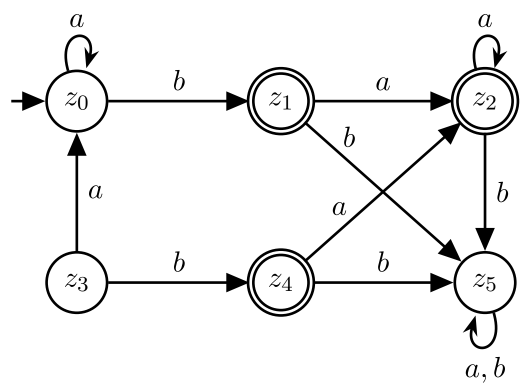
erzeugt wird. Geben Sie den Zustandsgraphen eines nichtdeterministischen endlichen Automaten (ohne ε-Übergänge) an, der L_1 akzeptiert.

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

$$bb(a|b)^*aa$$

erzeugt wird. Geben Sie den Zustandsgraphen eines deterministischen endlichen Automaten an, der L_2 akzeptiert. Bitte zeichnen Sie alle Übergänge und Zustände sorgfältig, inklusive des Müllzustands (falls vorhanden).

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 Zustandsgraph an.



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

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

$$L_3 = \{ab^icd^i \mid i \in \mathbb{N}\}$$

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.

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

$$L_4 = \overline{\{a^i b^j c d^j \mid i, j \in \mathbb{N}\}}$$

über dem Alphabet $\{a, b, c, d\}$ nicht regulär ist. $\overline{L}$ bezeichnet dabei das Komplement einer Sprache L. Sie dürfen annehmen, dass die Sprache $L_3 = \{ab^i cd^i \mid i \in \mathbb{N}\}$ 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): (24 Punkte)

a) Die Sprache L_5 über dem Alphabet $\Sigma_1 = \{a, b, c, d, e\}$ sei definiert als

$$L_5 = \{ab^i c^j d^i e \mid i, j \in \mathbb{N}_{>0}\}$$

Geben Sie eine kontextfreie Grammatik G_1, die L_5 erzeugt, als 4-Tupel an. Die Grammatik darf keine ε-Produktionen enthalten. Erläutern Sie, warum G_1 die Sprache L_5 erzeugt. Beschreiben Sie beispielsweise, welche „Aufgabe“ die einzelnen Nichtterminale bei der Erzeugung übernehmen.

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

b) Gegeben sei die kontextfreie Grammatik $G_2 = (V_2, \Sigma_2, P_2, S)$ mit $V_2 = \{S, T, U\}$, $\Sigma_2 = \{a, b, c, d, e\}$ und

$$P_2 = \{S \to TU, T \to aTb, T \to c, U \to dUe, U \to de\}$$

Geben Sie eine mathematische Beschreibung der Sprache an, die G_2 erzeugt.

c) Sei $G_3 = (V_3, \Sigma_3, P_3, S)$ eine kontextsensitive Grammatik mit $V_3 = \{S, B, C\}$, $\Sigma_3 = \{a, b, c\}$ und

$$P_3 = \{S \to aSBC,\, S \to aBC,\, CB \to BC,\, aB \to ab,\, bB \to bb,\, bC \to bc,\, cC \to cc\}$$

Um das Wortproblem für G_3 zu entscheiden, berechnen wir Mengen L_i^n mit $i \in \mathbb{N}$, $n \in \mathbb{N}_{>0}$, wobei

$$L_i^n = \{w \in (V \cup \Sigma)^* \mid |w| \leq n \text{ und } S \Rightarrow_{G_3}^k w \text{ mit } k \leq i\}$$

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

Geben Sie anhand Ihrer Berechnung ein Wort über Σ_3 an, das in $L(G_3)$ enthalten ist.

Aufgabe 4 (Berechenbarkeit und Entscheidbarkeit): (16 Punkte)

a) Geben Sie eine μ-rekursive Funktion f an, die $n \mapsto \sum_{i=1}^{n} i$ berechnet. Zum Beispiel soll $f(4) = \sum_{i=1}^{4} i = 10$ gelten. Sie können die Tatsachen nutzen, dass

$$\sum_{i=1}^{0} i = 0 \qquad \text{und} \qquad \sum_{i=1}^{n} i = \left(\sum_{i=1}^{n-1} i\right) + n \quad \text{für } n > 0$$

gelten. Erläutern Sie, warum f μ-rekursiv ist. Sie dürfen dabei annehmen, dass Addition (+) μ-rekursiv ist.

> **Zur Erinnerung:**
>
> Eine Funktion $f : \mathbb{N}^k \to \mathbb{N}$ ist *μ-rekursiv*, wenn sie der folgenden Definition genügt:
>
> - Jede *konstante Funktion* $f(x_1, \ldots, x_k) = c$ mit $c \in \mathbb{N}$ ist μ-rekursiv.
> - Die *Projektionsfunktionen* $\pi_i^k(x_1, \ldots, x_k) = x_i$ sind μ-rekursiv.
> - Die *Nachfolgerfunktion* $succ(x) = x + 1$ ist μ-rekursiv.
> - *Komposition*: Wenn die Funktionen $g : \mathbb{N}^m \to \mathbb{N}$ und $h_i : \mathbb{N}^k \to \mathbb{N}$ für $i = 1, \ldots, m$ μ-rekursiv sind, dann ist auch f mit $f(x_1, \ldots, x_k) = g(h_1(x_1, \ldots, x_k), \ldots, h_m(x_1, \ldots, x_k))$ μ-rekursiv.
> - *Rekursion*: Wenn $g : \mathbb{N}^{k-1} \to \mathbb{N}$ und $h : \mathbb{N}^{k+1} \to \mathbb{N}$ μ-rekursiv sind, dann ist auch folgende Funktion μ-rekursiv:
>
> $$f(x_1, \ldots, x_k) = \begin{cases} g(x_2, \ldots, x_k) & \text{falls } x_1 = 0 \\ h(f(x_1 - 1, x_2, \ldots, x_k), x_1 - 1, x_2, \ldots, x_k) & \text{sonst} \end{cases}$$
>
> - *μ-Operator*: Wenn $h : \mathbb{N}^{k+1} \to \mathbb{N}$ μ-rekursiv ist, dann ist auch $f = \mu h$ μ-rekursiv.

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 nach höchstens 1000 Schritten anhält.

(ii) Es ist unentscheidbar, ob eine gegebene deterministische Turingmaschine für mindestens eine Eingabe $i \in \mathbb{N}$ die Zahl $i + 1$ berechnet.

Aufgabe 5 (Komplexität): (16 Punkte)

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

gegeben:	ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$
gefragt:	Gibt es eine Färbung der Knoten in V mit höchstens k Farben, sodass keine zwei benachbarten Knoten in G die gleiche Farbe erhalten?

In der Vorlesung wurde bewiesen, dass GRAPH-COLORING $\mathcal{NP}$-schwer ist. Zur Erinnerung bestand der Beweis aus folgenden Schritten:

- Wir zeigen 3-CNF-SAT $\leq_{\text{p}}$ GRAPH-COLORING.
- Sei $F = K_1 \wedge \cdots \wedge K_m$ eine 3-CNF, sodass jede Klausel K_i genau 3 Literale enthält.
- Wir erzeugen ein GRAPH-COLORING-Problem mit $k = 3$ Farben.
- Wir konstruieren die „Palette“ und für jede Klausel ein „Oder-Gatter“.
- Der Graph ist k-färbbar g.d.w. F erfüllbar ist.
- Die Übersetzung von F in einen Graphen ist total und polynomiell.

Ein verwandtes Problem ist das GRAPH-3-COLORING-Problem:

gegeben:	ein ungerichteter Graph $G = (V, E)$
gefragt:	Gibt es eine Färbung der Knoten in V mit höchstens **drei** Farben, sodass keine zwei benachbarten Knoten in G die gleiche Farbe erhalten?

Ist GRAPH-3-COLORING $\mathcal{NP}$-schwer? Begründen Sie kurz Ihre Antwort.

b) In der Vorlesung wurden die Probleme CLIQUE, INDEPENDENT-SET und VERTEX-COVER vorgestellt. Das CLIQUE-Problem lässt sich so definieren:

gegeben:	ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$
gefragt:	Besitzt G eine Clique der Größe mindestens k, d.h. eine Menge $V' \subseteq V$, sodass $\|V'\| \geq k$ und für alle $u, v \in V'$ mit $u \neq v$ gilt $\{u, v\} \in E$?

Das INDEPENDENT-SET-Problem lässt sich so definieren:

gegeben:	ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$
gefragt:	Besitzt G eine unabhängige Knotenmenge der Größe mindestens k, d.h., eine Menge $V' \subseteq V$ mit $\|V'\| \geq k$, sodass keine zwei Knoten aus V' über eine Kante aus E verbunden sind?

Das VERTEX-COVER-Problem lässt sich so definieren:

gegeben:	ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$
gefragt:	Besitzt G eine überdeckende Knotenmenge der Größe höchstens k, d.h. eine Menge $V' \subseteq V$ mit $\|V'\| \leq k$, sodass jede Kante aus E mindestens einen ihrer beiden Knoten in der Menge hat?

Sie dürfen als bekannt annehmen, dass diese drei Probleme $\mathcal{NP}$-vollständig sind.

Nun führen wir folgendes STRANGERS-Problem ein:

gegeben:	eine Menge von Personen P, eine binäre Relation $K \subseteq P \times P$, wobei $x\,K\,y$ intuitiv „x kennt y“ heißt, und eine Zahl $n \in \mathbb{N}$
gefragt:	Gibt es eine Teilmenge $S \subseteq P$ mit $\|S\| \geq n$, sodass alle Elemente von S „Fremde“ sind (formal: $\neg\, s\,K\,t$ für alle $s, t \in S$)?

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