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

SoSe 2024
30.09.2024

Zweitklausur 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	31	18	24	15	12	100
Erreicht						

Aufgabe 1 (Reguläre Sprachen): (31 Punkte)

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

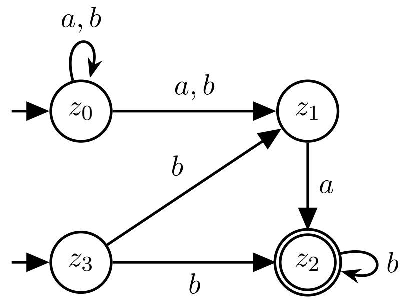
Geben Sie einen regulären Ausdruck an, der L_1 erzeugt.

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

$$(a|b|c)^*(abc \mid ca(ba)^* \mid \varepsilon)$$

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

c) Der nichtdeterministische endliche Automat M_1 über $\{a, b\}$ ist durch den folgenden Zustandsgraphen gegeben:



Berechnen Sie einen deterministischen endlichen Automaten M_1' aus M_1 mit der Potenzmengenkonstruktion. Es ist ausreichend, die erreichbaren Zustände von M_1' anzugeben. Sie müssen keine Begründung angeben, nur das Ergebnis.

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

a) Zeigen Sie mit dem Satz von Myhill und Nerode, dass die Sprache

$$L_3 = \{ab^ic^i \mid i \in \mathbb{N}_{>0}\}$$

über dem Alphabet $\{a, b, c\}$ nicht regulär ist. Listen Sie die Äquivalenzklassen $[u_1]_{\sim_{L_3}}, [u_2]_{\sim_{L_3}}, \ldots$ von $\sim_{L_3}$ auf, die Sie für den Beweis nutzen. Geben Sie für jede Äquivalenzklasse $[u_i]_{\sim_{L_3}}$ ein Suffix w_i an, sodass $u_i w_i \in L_3$ aber $u_j w_i \notin L_3$ für jedes $j \neq i$.

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

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

über dem Alphabet $\{a, b\}$ nicht regulär ist, wobei $\overline{L}$ das Komplement einer Sprache L bezeichnet. Sie dürfen annehmen, dass die Sprache $\{a^i b^i \mid i \in \mathbb{N}\}$ nicht regulär ist.

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

Aufgabe 3 (Kontextfreie Sprachen): (24 Punkte)

a) Die Sprache L_5 über dem Alphabet $\{a, b\}$ sei definiert als

$$L_5 = \{a^i b a^{2i} b^j a b^j \mid i, j \in \mathbb{N}\}$$

Geben Sie eine kontextfreie Grammatik G_1 als 4-Tupel an, die L_5 erzeugt. 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.

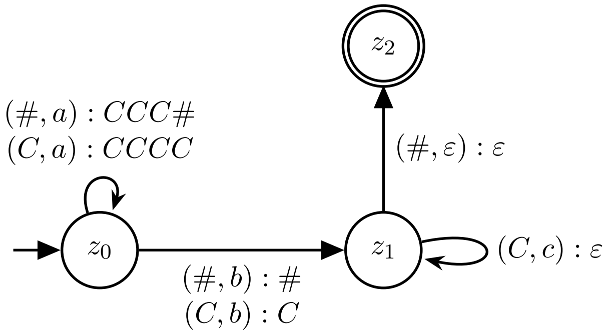
Geben Sie zusätzlich eine Linksableitung für das Wort *aabaaaabab* für Ihre Grammatik an.

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

$$P_2 = \{S \to TUT,\ T \to aTa,\ T \to c,\ U \to bUb,\ U \to c\}$$

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

c) Gegeben sei der folgende Zustandsgraph eines Kellerautomaten M_2 mit Endzuständen. Geben Sie eine mathematische Beschreibung der Sprache an, die M_2 akzeptiert.



Aufgabe 4 (Berechenbarkeit und Entscheidbarkeit): (15 Punkte)

a) 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.

(i) Es ist unentscheidbar, ob eine gegebene deterministische Turingmaschine für die Eingabe 42 die Zahl 43 berechnet.

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

b) Das Postsche Korrespondenzproblem lässt sich so definieren:

Sei Σ ein Alphabet mit $|\Sigma| > 1$. Eine Instanz des *Postschen Korrespondenzproblems* (*PCP*) besteht aus einer endlichen Folge $(x_1, y_1), \ldots, (x_k, y_k)$ von Wortpaaren mit $x_i, y_i \in \Sigma^+$. Das Entscheidungsproblem ist die Frage, ob es eine Folge von Indizes $i_1, \ldots, i_m$ mit $i_j \in \{1, \ldots, k\}$ und $m > 0$ gibt, sodass $x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$ gilt.

Eine weniger bekannte Variante des Problems lässt sich so definieren:

Sei Σ ein Alphabet mit $|\Sigma| > 1$. Eine Instanz des *umgedrehten Postschen Korrespondenzproblems* (*UPCP*) besteht aus einer endlichen Folge $(x_1, y_1), \ldots, (x_k, y_k)$ von Wortpaaren mit $x_i, y_i \in \Sigma^+$. Das Entscheidungsproblem ist die Frage, ob es eine Folge von Indizes $i_1, \ldots, i_m$ mit $i_j \in \{1, \ldots, k\}$ und $m > 0$ gibt, sodass $x_{i_1} \cdots x_{i_m} = \overline{y_{i_1}} \cdot \ldots \cdot \overline{y_{i_m}}$ gilt, wobei $\overline{a_1 \cdots a_n}$ als $a_n \cdots a_1$ definiert ist.

Beweisen Sie mithilfe einer Reduktion, dass UPCP unentscheidbar ist.

Aufgabe 5 (Komplexität): (12 Punkte)

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

gegeben:	k Gegenstände mit Gewichten $w_1, \dots, w_k \in \mathbb{N}$ und Nutzenwerten $n_1, \dots, n_k \in \mathbb{N}$, sowie zwei Schwellenwerte $s_w, s_n \in \mathbb{N}$
gefragt:	Gibt es eine Teilmenge $I \subseteq \{1, \dots, k\}$, sodass $\sum_{i \in I} w_i \leq s_w$ und $\sum_{i \in I} n_i \geq s_n$?

In der Vorlesung wurde bewiesen, dass KNAPSACK $\mathcal{NP}$-vollständig ist.

Eine Variante namens LIGHTWEIGHT-KNAPSACK sei definiert wie folgt:

gegeben:	k Gegenstände mit Nutzenwerten $n_1, \dots, n_k \in \mathbb{N}$, sowie zwei Schwellenwerte $s_c, s_n \in \mathbb{N}$ mit $s_c \leq k$
gefragt:	Gibt es eine Teilmenge $I \subseteq \{1, \dots, k\}$, sodass $\lvert I \rvert \leq s_c$ und $\sum_{i \in I} n_i \geq s_n$?

Nehmen Sie an, dass $\mathcal{P} \neq \mathcal{NP}$. Ist LIGHTWEIGHT-KNAPSACK $\mathcal{NP}$-vollständig? Begründen Sie Ihre Antwort (z.B. mit einer Reduktion oder einer Skizze eines Algorithmus).

b) Wir erinnern 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?

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

Ein weniger bekanntes Problem ist ASSIGN-GIFTS:

gegeben:	eine endliche Menge P von Personen, eine Menge $M = \{K_1, \ldots, K_n\}$ von Mengen $K_i \subseteq P$, sodass K_i einen Freundeskreis darstellt, und eine Zahl $t \in \mathbb{N}_{>0}$, die angibt, wie viele Geschenktypen es gibt
gefragt:	Gibt es eine Zuweisung $g : P \to \{0, 1, \ldots, t-1\}$ von Personen nach Geschenktypen, sodass keine zwei Personen im selben Freundeskreis denselben Geschenktypen bekommen?

Ein Beispiel für eine ASSIGN-GIFTS-Instanz ist

$$\begin{aligned} P &= \{anita, bashar, cindy\} \\ M &= \{\{anita, bashar\}, \{anita, cindy\}\} \\ t &= 2 \end{aligned}$$

Diese Instanz ist lösbar, wie die Zuweisung $g(anita) = 0$, $g(bashar) = 1$ und $g(cindy) = 1$ bezeugt.

Zeigen Sie mithilfe einer Polynomialzeit-Reduktion, dass ASSIGN-GIFTS $\mathcal{NP}$-schwer ist.