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

SoSe 2023
25.09.2023

Lösung zur Nachholklausur 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 Farben **rot noch grün**.

Kontrollieren Sie, ob Sie alle Aufgabenblätter erhalten haben. Aufgabenstellungen befinden sich auf den **Seiten 1–13**. 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.

Lesen Sie die Aufgabenstellungen vollständig durch, bevor Sie mit der Bearbeitung der Aufgaben beginnen. Es gibt 5 unterschiedlich gewichtete Aufgaben zu insgesamt 100 Punkten. Mit 50 Punkten haben Sie sicherlich 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:

Vorname:

Matrikelnummer:

Studiengang:

☐ **Die Klausur soll nicht korrigiert werden**

Hinweis: Ist das Kästchen **nicht** angekreuzt, wird die Klausur **gewertet**.

Hiermit erkläre ich die Richtigkeit der obigen Angaben: ______________________
(Unterschrift)

Die folgende Tabelle **nicht** ausfüllen:

Aufgabe	1	2	3	4	5	$\sum$	Bonus	Note
Punkte	33	28	11	16	12	**100**		
Erreicht								

Lösung Aufgabe 1 (Reguläre Sprachen): (33 Punkte)

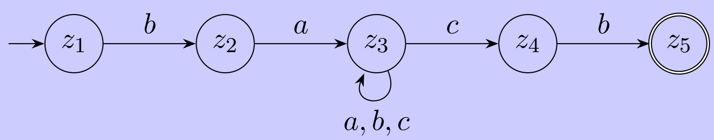
a) Die Sprache $L_1 \subseteq \{a, b, c\}^*$ sei definiert als

$$L_1 := \{baucb \mid u \in \{a, b, c\}^*\}$$

L_1 enthält daher genau alle Wörter über $\{a, b, c\}$, die mit ba anfangen und cb enden.

Geben Sie den Zustandsgraphen eines nichtdeterministischen endlichen Automaten (ohne ε-Übergänge) A_1 an, für den gilt: $L(A_1) = L_1$. (7 Punkte)

LÖSUNG: A_1:



7 Punkte, wenn alles stimmt
1 Punkt Abzug bei fehlendem Startzustand
1 Punkt Abzug bei fehlendem Endzustand
1 Punkt Abzug bei falschem Übergang

b) Die Sprache L_2 sei definiert als

$$L_2 := \{w \in \{a, b, c, d\}^* \mid \#_b(w) \geq 2\}$$

Die Sprache L_2 enthält daher alle Wörter über $\{a, b, c, d\}$, die mindestens zwei b's enthalten. Geben Sie einen regulären Ausdruck α mit $L(\alpha) = L_2$ an. (5 Punkte)

LÖSUNG: Zum Beispiel:

$\alpha = (a|c|d)^*b(a|c|d)^*b(a|b|c|d)^*$

oder

$\alpha = (a|b|c|d)^*b(a|b|c|d)^*b(a|b|c|d)^*$

2 Punkte Abzug falls ≤ 2 oder $= 2$ statt ≥ 2
1 Punkt Abzug für jeden fehlenden/zusätzlichen $(a|b|c|d)^*$ Block
1 Punkt Abzug für jeden fehlenden/zusätzlichen b Block
2 Punkte Abzug für Sprachen-Ausdruck statt Regex (z.B. $\{a, c, d\}^*$ statt $(a|c|d)^*$)

c) Bestimmen Sie den Index der Nerode-Relation für die Sprache $L_3 = \{a^i b^i \mid i \in \mathbb{N}\}$ über $\Sigma = \{a, b\}$ und begründen Sie kurz Ihre Antwort. (4 Punkte)

LÖSUNG: Es ist bekannt, dass L_3 nicht regulär ist. Daher gilt Index$(\sim_{L_3}) = \infty$.

Alternativ kann man die Äquivalenzklassen $[a^i b]$ für jedes i angeben. Es gibt unendlich viele davon und man kann zeigen, dass sie paarweise disjunkt sind, indem man die Suffixe b^{i-1} betrachtet.

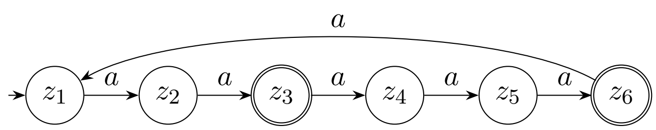
3 Punkte für richtigen Index
1 Punkt für Begründung
(als Begründung reicht eine Auflistung der Äquivalenzklassen ohne weitere Erklärung)

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 1:

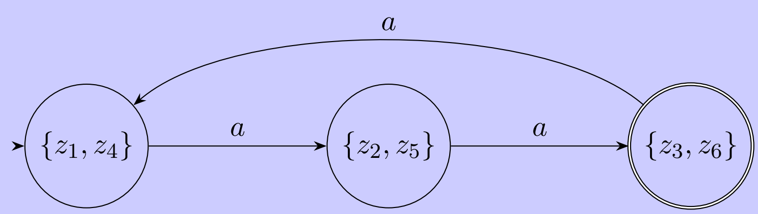
d) Berechnen Sie für den folgenden deterministischen endlichen Automaten A_2 (der Wörter über dem Alphabet $\Sigma = \{a\}$ verarbeitet) den **Minimalautomaten**, d.h. einen deterministischen endlichen Automaten, der die gleiche Sprache akzeptiert und eine minimale Anzahl an Zuständen benutzt.

Erläutern Sie Ihre Berechnung, indem Sie die Minimierungstabelle angeben, und geben Sie den Minimalautomaten als Zustandsgraphen an.



(11 Punkte)

LÖSUNG: Minimaler DFA



Berechnung:

	z_1	z_2	z_3	z_4	z_5
z_2	X_1				
z_3	X	X			
z_4		X_1	X		
z_5	X_1		X	X_1	
z_6	X	X		X	X

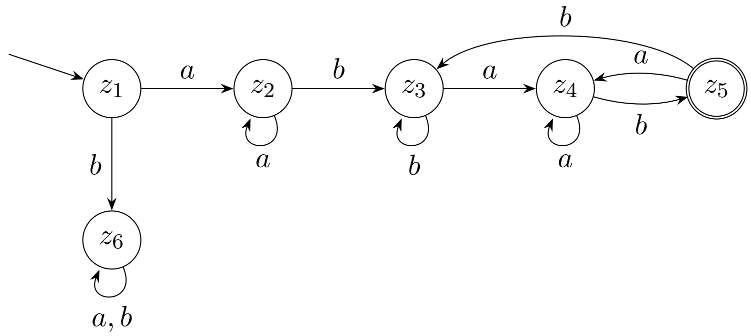
Notation: X: Als verschieden erkannt in der Initialisierung.
X_1: Als verschieden erkannt in der ersten Iteration.

- 7 Punkte für die Tabelle:
  1 Punkt Abzug per falschem Tabelleneintrag
- 4 Punkte für den Minimalautomaten:
  1 Punkt Abzug, wenn Start- und/oder Endzustand nicht gekennzeichnet sind
  1 Punkt Abzug für jeden falschen Übergang (bezüglich der Tabelle)

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 1:

e) Der folgende Automat A_3



ist ein DFA mit Eingabealphabet $\Sigma = \{a, b\}$.

Begründen Sie für jedes der Zustandspaare

(i) (z_4, z_5)

(ii) (z_1, z_2)

(iii) (z_3, z_6)

warum die beiden Zustände des Paares **nicht äquivalent** sind. (6 Punkte)

LÖSUNG: Zustände z, z' sind äquivalent g.d.w. $\forall w \in \Sigma^* : \widehat{\delta}(z, w) \in E \iff \widehat{\delta}(z', w) \in E$.

(i) (z_4, z_5): $\widehat{\delta}(z_4, \varepsilon) = z_4 \notin E$ aber $\widehat{\delta}(z_5, \varepsilon) = z_5 \in E$

(ii) (z_1, z_2): $\widehat{\delta}(z_1, bab) = z_6 \notin E$ aber $\widehat{\delta}(z_2, bab) = z_5 \in E$

(iii) (z_3, z_6): $\widehat{\delta}(z_1, ab) = z_5 \in E$ aber $\widehat{\delta}(z_6, ab) = z_6 \notin E$ oder
$\widehat{\delta}(z_2, b) = z_4 \notin E$ aber $\widehat{\delta}(z_4, b) = z_3 \in E$

2 Punkte pro Antwort

Lösung Aufgabe 2 (Kontextfreiheit): **(28 Punkte)**

a) Die Sprache L_4 über $\{a, b, c, d\}$ sei definiert als

$$L_4 := \{a^i b^j c^j d^k \mid i, j, k \in \mathbb{N}_{>0}\}$$

Informell: Alle Wörter aus L_4 sind von der Form $a^i b^j c^j d^k$, wobei i, j, k positive Zahlen sind.

Geben Sie eine **kontextfreie Grammatik** G_1 an, die L_4 erzeugt (d.h. $L(G_1) = L_4$) und erläutern Sie kurz, warum G_1 die Sprache L_4 erzeugt: Beschreiben Sie beispielsweise, welche „Aufgabe" die einzelnen Nichtterminale bei der Erzeugung übernehmen. (5 Punkte)

Geben Sie zusätzlich eine **Rechtsableitung** für das Wort *abbccdd* für Ihre Grammatik an. (3 Punkte)

LÖSUNG: $G_1 = (V, \{a, b, c, d\}, P, S)$ mit $V = \{S, X, Y, Z\}$ und $P = \{S \to XYZ, X \to aX, X \to a, Y \to bYc, Y \to bc, Z \to dZ, Z \to d\}$. Erläuterung:

- S erzeugt zunächst XYZ. Aus X lassen sich beliebig viele a's erzeugen (jedoch mindestens 1). Aus Y lassen sich beliebig viele b/c-Paare erzeugen (jedoch mindestens 1). Aus Z lassen sich beliebig viele d's erzeugen (jedoch mindestens 1).

Rechtsableitung: $S \Rightarrow XYZ \Rightarrow XYdZ \Rightarrow XYdd \Rightarrow XbYcdd \Rightarrow Xbbccdd \Rightarrow abbccdd$

5 Punkte für die Grammatik: 3 Punkte für die Grammatik (als Tupel oder Text angegeben), 2 Punkte für die Erläuterung
Maximal 1 Punkt falls Grammatik ganz falsch.
Maximal 1 Punkt für Erläuterung bei falscher Grammatik.
1 Punkt Abzug falls das Startsymbol nicht angegeben ist.

3 Punkte für die Rechtsableitung: 1 Punkt falls keine Rechtsableitung oder falls Syntaxbaum.

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 2:

b) Zeigen Sie mit dem Pumping-Lemma für reguläre Sprachen, dass

$$L_4 = \{a^i b^j c^j d^k \mid i, j, k \in \mathbb{N}_{>0}\}$$

nicht regulär ist.

(10 Punkte)

Zur Erinnerung:

Pumping-Lemma für reguläre Sprachen.
Sei L eine reguläre Sprache. Dann gibt es eine Zahl $n \in \mathbb{N}_{>0}$, sodass jedes Wort $z \in L$, das Mindestlänge n hat (d.h. $|z| \geq n$), als $z = uvw$ geschrieben werden kann, sodass gilt:
i) $|uv| \leq n$,
ii) $|v| \geq 1$ und
iii) für alle $i \geq 0$: $uv^i w \in L$.

LÖSUNG: Sei $n > 0$ beliebig.
Eine geeignete Wahl für z ist z.B.
$z = ab^n c^n d$.

Beweis mit $z = ab^n c^n d$:

Sei $z = uvw$ eine beliebige Zerlegung von z mit $|uv| \leq n$ und $|v| \geq 1$.

Wenn $u = \varepsilon$, dann $v = ab^{n_1}$ und $w = b^{n_2} c^n d$ mit $n = n_1 + n_2$. Dann ist $uv^0 w \notin L_4$, denn $uv^0 w = b^{n_2} c^n d$ und es gibt keine a's.

Wenn $u \neq \varepsilon$, dann ist $u = ab^{n_1}$, $v = b^{n_2}$ und $w = b^{n_3} c^n d$ mit $n = n_1 + n_2 + n_3$. Dann ist $uv^0 w \notin L_4$, denn $uv^0 w = ab^{n_1+n_3} b^n cd$ und es gibt mehr c's als b's.

Damit erfüllt L_4 die Pumping-Eigenschaft nicht und das Pumping-Lemma zeigt, dass L_4 nicht regulär ist.

- 1 Punkt: $z \in L_4$ gewählt
- 1 Punkt: $|z| \geq n$
- 2 Punkte: z ist geeignet
- 3 Punkte: Über alle Zerlegungen u, v, w argumentiert
  1 Punkt Abzug, falls der Fall $u = \varepsilon$ vergessen wurde
- 3 Punkte: Fall gefunden, sodass $uv^i w \notin L_4$
- 2 Punkte Abzug, wenn komplett unnötige Fälle betrachtet wurden

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 2:

c) Sei $G_2 = (\{S, T, U\}, \{a, b\}, P, S)$ mit

$$\begin{aligned} P = \{ & S \to SS \mid TT \mid UU \\ & T \to ST \mid a \\ & U \to TU \mid b \} \end{aligned}$$

eine kontextfreie Grammatik.

Entscheiden Sie, ob $baabaa \in L(G_2)$, indem Sie den Algorithmus von Cocke, Younger und Kasami (CYK-Algorithmus) ausführen. Geben Sie die dabei entstehende Tabelle und die Ausgabe des Algorithmus an und dokumentieren Sie Ihren Rechenweg. (10 Punkte)

LÖSUNG:

Der Algorithmus hat die folgende Tabelle für die $V(i, j)$-Mengen berechnet:

Wort	b	a	a	b	a	a
$j \backslash i$	1	2	3	4	5	6
1	U	T	T	U	T	T
2		S	U		S	
3		U				
4	S					
5	T					
6	S					

Nachdem $S \in V(1, 6)$ liegt, ist $baabaa \in L(G_2)$.

2 Punkte für richtige Antwort
8 Punkte für Tabelle
1 Punkt Abzug per falsche Zelle

Lösung Aufgabe 3 (Berechenbarkeitstheorie): **(11 Punkte)**

a) Welche Funktion $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ berechnet das folgende LOOP-Programm? Begründen Sie kurz Ihre Antwort.

Beachten Sie insbesondere, dass die Eingaben über die Variablen x_1 und x_2 empfangen werden und die Ausgabe in der Variable x_0 erfolgt. (5 Punkte)

LOOP x_2 **DO**
  $x_2 := x_2 + 1$
END;
LOOP x_2 **DO**
  $x_0 := x_0 + 1$
END;
LOOP x_1 **DO**
  $x_0 := x_0 + 3$
END

LÖSUNG: $f(x, y) = 3x + 2y$: Die Eingaben werden über x_1, x_2 empfangen und die Ausgabe ist in x_0. Die erste Schleife verdoppelt x_2. Die zweite Schleife addiert x_2 zu x_0. Da x_0 noch 0 ist, entspricht sie dem Wert von x_2. Schließlich wird zu x_0 der Wert von $3x_1$ addiert.

4 Punkte für richtige Antwort
1 Punkt für kurze Begründung
1 Punkt Abzug falls x und y vertauscht sind
2 Punkte Abzug falls $3x$ bzw. $2y$ falsch ist

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 3:

b) Geben Sie ein LOOP-Programm an, welches die Funktion $f(x, y, z) = x + 2y + 3z$ berechnet. (6 Punkte)

LÖSUNG:

LOOP x_1 **DO**
$x_0 := x_0 + 1$
END;
LOOP x_2 **DO**
$x_0 := x_0 + 2$
END;
LOOP x_3 **DO**
$x_0 := x_0 + 3$
END

2 Punkte pro Schleife
1 Punkt Abzug für jeden Syntaxfehler (erweiterte Syntax ist erlaubt)

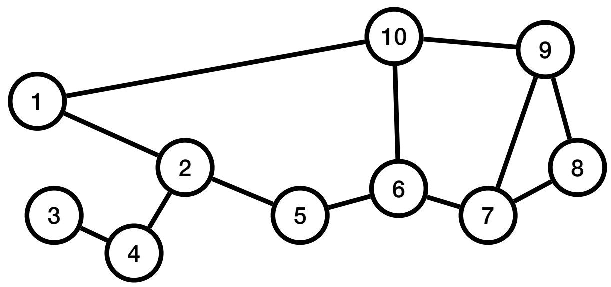
Lösung Aufgabe 4 (Komplexitätstheorie): **(16 Punkte)**

Wir erinnern zunächst an die Definition des INDEPENDENT-SET-Problems:

gegeben:	Ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$
gefragt:	Besitzt G eine unabhängige Knotenmenge der Größe mindestens 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. $u, v \in V' \implies \{u, v\} \notin E$.

a) Es sei G_1 der folgende Graph und $k = 4$:



Zeigen Sie, dass $(G_1, k) \in$ INDEPENDENT-SET gilt.

(5 Punkte)

LÖSUNG: Es gibt viele mögliche Lösungen, z.B. $V' = \{1, 3, 5, 7\}$.

5 Punkte für V'
2 Punkte Abzug für jeden Fehler (zwei benachbarte Knoten in V')
3 Punkte Abzug für weniger als 4 Elemente in V'

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 4:

b) Es sei G_1 derselbe Graph wie in Frage a) und $k = 10$. Gilt $(G_1, k) \in$ INDEPENDENT-SET? Begründen Sie kurz Ihre Antwort. (3 Punkte)

LÖSUNG: Nein. Dann ist $V' = V$, und es gibt z.B. $1, 2 \in V'$ mit $\{1, 2\} \in E$.

2 Punkte für richtige Antwort
1 Punkt für kurze Begründung

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 4:

c) Der folgende Beweis ist falsch. Lokalisieren Sie den Fehler und begründen Sie, wieso der Beweis falsch ist. (8 Punkte)

Satz: INDEPENDENT-SET ist in $\mathcal{NP}$.

Beweis: Wir zeigen CLIQUE $\leq_{\mathrm{p}}$ INDEPENDENT-SET. Da CLIQUE in $\mathcal{NP}$ liegt, liegt auch INDEPENDENT-SET in $\mathcal{NP}$.

Sei $f((V, E, m)) = (V, \overline{E}, m)$ wobei $\overline{E} = \{\{u, v\} \mid u, v \in V, \{u, v\} \notin E\}$. Dann gilt: (V, E) hat eine Clique der Größe m g.d.w. $(V, \overline{E})$ eine unabhängige Knotenmenge der Größe m hat. Da die Funktion f in Polynomialzeit berechnet werden kann, gilt CLIQUE $\leq_{\mathrm{p}}$ INDEPENDENT-SET.

LÖSUNG: Der Fehler liegt im ersten Teil: „Wir zeigen CLIQUE $\leq_{\mathrm{p}}$ INDEPENDENT-SET. Da CLIQUE in $\mathcal{NP}$ liegt, liegt auch INDEPENDENT-SET in $\mathcal{NP}$.“ Um $\mathcal{NP}$ zu beweisen, geht die Reduktion andersherum.

4 Punkte für Ort des Fehlers korrekt identifiziert
4 Punkte fur Begründung

Lösung Aufgabe 5 (Gemischte Fragen): **(12 Punkte)**

Pro Frage gibt es **genau eine richtige** Antwort.

Jede Antwort ist 2 Punkte wert. Eine Antwort, bei der 0 oder mindestens 2 Kästchen angekreuzt sind, zählt als falsch.

Das CLIQUE-Problem ist in $\mathcal{NP}$. Zudem ist Folgendes bekannt:

- [ ] CLIQUE $\in \mathcal{P}$
- [ ] CLIQUE $\notin \mathcal{P}$
- [X] CLIQUE ist $\mathcal{NP}$-schwer
- [ ] Wenn $\mathcal{P} \neq \mathcal{NP}$, dann CLIQUE $\in \mathcal{P}$

Turing-berechenbare Funktionen sind manchmal

- [ ] nicht μ-rekursiv
- [ ] nicht WHILE-berechenbar
- [ ] nicht GOTO-berechenbar
- [X] nicht primitiv rekursiv

Die Grammatik $G = (\{A, B\}, \{a, b\}, P, A)$ mit

$$\begin{aligned} P = \{&a \to B, \\ &bA \to A\} \end{aligned}$$

ist

- [X] vom Typ 0, aber nicht vom Typ 1
- [ ] vom Typ 1, aber nicht vom Typ 2
- [ ] vom Typ 2, aber nicht vom Typ 3
- [ ] vom Typ 3

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 5:

Das WHILE-Programm

$$\begin{array}{l} x_0 := x_1 + 2; \\ x_1 := x_2 + 0; \\ \textbf{WHILE } x_1 \neq 0 \textbf{ DO} \\ \quad x_1 := x_1 - 1; \\ \quad x_0 := x_0 + 5 \\ \textbf{END} \end{array}$$

berechnet die Funktion f mit

- [ ] $f(x) = 2 \cdot x$
- [X] $f(x, y) = x + 5 \cdot y + 2$
- [ ] $f(x, y) = 5 \cdot y + 2$
- [ ] $f(x, y, z) = x \cdot y + 5 \cdot z$

Sei L_5 eine kontextsensitive Sprache.

- [ ] Dann gibt es einen DFA, der L_5 akzeptiert.
- [ ] Dann gibt es einen NFA, der L_5 akzeptiert, aber keinen DFA.
- [ ] Dann gibt es einen Kellerautomaten, der L_5 akzeptiert, aber keinen NFA oder DFA.
- [X] Dann gibt es eine Turingmaschine, die L_5 akzeptiert, aber keinen Kellerautomaten, NFA oder DFA.

Die Grammatik $G = (\{A, B, C, D\}, \{a, b\}, P, A)$ mit

$$\begin{aligned} P = \{ & A \to a, \\ & A \to BC, \\ & B \to b, \\ & bB \to CC, \\ & D \to b, \\ & aD \to BC\} \end{aligned}$$

ist

- [ ] vom Typ 0, aber nicht vom Typ 1
- [X] vom Typ 1, aber nicht vom Typ 2
- [ ] vom Typ 2, aber nicht vom Typ 3
- [ ] vom Typ 3