Institut für Informatik der Ludwig-Maximilians-Universität München
Prof. Dr. David Sabel
Stephan Barth
Sarah Vaupel

SoSe 2022
17.08.2022

Lösung zur Klausur zur Vorlesung

Theoretische Informatik für Medieninformatiker

Die Bearbeitungszeit beträgt **90 Minuten**. Hilfsmittel sind nicht erlaubt, auch der Besitz 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 1–8**. Sie dürfen die Rückseiten für Nebenrechnungen nutzen. Falls Sie die Rückseiten für Antworten nutzen, so markieren Sie klar, was zu welcher Aufgabe gehört und schreiben Sie in der entsprechenden Aufgabe, 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 4 unterschiedlich gewichtete Aufgaben zu insgesamt 90 Punkten. Mit 45 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 gewertet werden: ☐ **JA** ☐ **NEIN**

Hinweis: Ist keines der beiden Felder angekreuzt, so wird **JA** angenommen.

Hiermit erkläre ich die Richtigkeit der obigen Angaben: ____________________

(Unterschrift)

Die folgende Tabelle **NICHT** ausfüllen:

Aufgabe	1	2	3	4	$\sum$	Bonus	Note
Punkte	34	28	16	12	**90**		
erreicht							

Lösung Aufgabe 1 (Reguläre Sprachen): **(34 Punkte)**

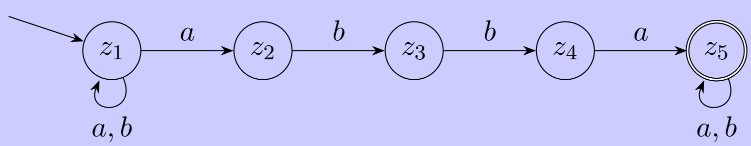
a) Die Sprache $L_1 \subseteq \{a, b\}^*$ sei definiert als:

$$L_1 \quad := \quad \{uabbav \mid u, v \in \{a, b\}^*\}$$

L_1 enthält daher genau alle Wörter, die *abba* als Teilwort haben.

Geben Sie den Zustandsgraphen eines nichtdeterministischen endlichen Automaten (NFA) 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

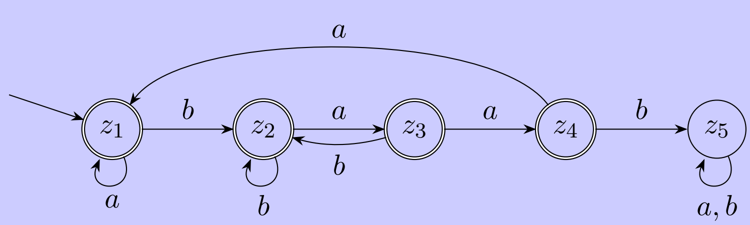
b) Die Sprache $L_2 \subseteq \{a, b\}^*$ sei definiert als:

$$L_2 \quad := \quad \{w \mid w \in \{a, b\}^*, \forall u, v \in \{a, b\}^* : w \neq ubaabv\}$$

L_2 enthält daher genau alle Wörter, die *baab* **nicht** als Teilwort besitzen.

Geben Sie den Zustandsgraphen eines deterministischen endlichen Automaten (DFA) A_2 an, für den gilt: $L(A_2) = L_2$. (7 Punkte)

LÖSUNG: A_2:



7 Punkte, wenn alles stimmt
1 Punkt Abzug bei fehlendem Startzustand
1 Punkt Abzug bei fehlenden Endzuständen

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 1:

c) Die Sprache $L_3 \subseteq \{a, b\}^*$ sei definiert als:

$$\begin{array}{lcl} L_3 & := & L_4 \cup L_5 \\ L_4 & := & \{w \mid \#_a(w) \geq 2\} \\ L_5 & := & \{w \mid \#_b(w) \leq 3\} \end{array}$$

Die Sprache L_3 enthält daher alle Wörter, die mindestens zwei a's oder höchstens drei b's enthalten.

Geben Sie einen regulären Ausdruck α mit $L(\alpha) = L_3$ an. (6 Punkte)

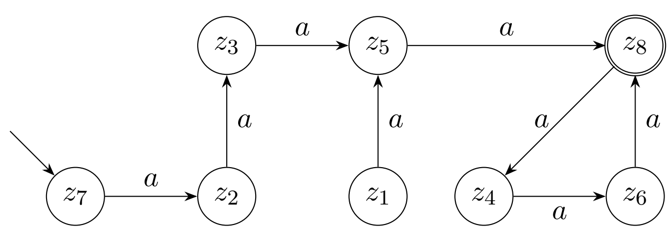
LÖSUNG:

$\alpha = ((a|b)^*a(a|b)^*a(a|b)^*) \mid (a^*(b|\varepsilon)a^*(b|\varepsilon)a^*(b|\varepsilon)a^*)$

6 Punkte = jeweils 3 Punkte für mindestens und höchstens

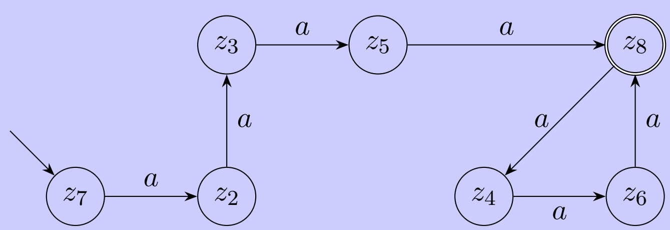
d) Berechnen Sie für den folgenden deterministischen endlichen Automaten A_3 (der Worte über dem Alphabet $\Sigma = \{a\}$ verarbeitet) den **Minimalautomaten**, d. h. einen deterministischen endlichen Automaten, der die gleiche Sprache akzeptiert und eine minimale Anzahl an Zuständen benutzt.

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

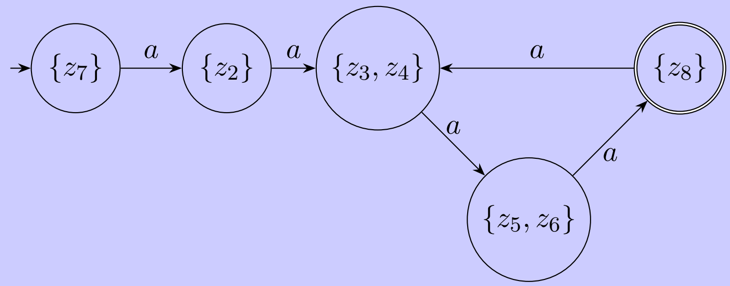


(14 Punkte)

LÖSUNG: Der DFA nach dem Entfernen nicht erreichbarer Zustände:



Minimaler DFA



Berechnung:

	z_2	z_3	z_4	z_5	z_6	z_7
z_3	X_2					
z_4	X_2					
z_5	X_1	X_1	X_1			
z_6	X_1	X_1	X_1			
z_7	X_3	X_2	X_2	X_1	X_1	
z_8	X_0	X_0	X_0	X_0	X_0	X_0

Notation: X_i : Als verschieden erkannt in Iteration i.

Die folgende Auflistung des Ablaufs ist nicht notwendig für eine vollständige Lösung

Initialisierung:

- (z_2, z_8) verschieden, da z_2 nicht akzeptierend aber z_8 akzeptierend.
- (z_3, z_8) verschieden, da z_3 nicht akzeptierend aber z_8 akzeptierend.
- (z_4, z_8) verschieden, da z_4 nicht akzeptierend aber z_8 akzeptierend.
- (z_5, z_8) verschieden, da z_5 nicht akzeptierend aber z_8 akzeptierend.
- (z_6, z_8) verschieden, da z_6 nicht akzeptierend aber z_8 akzeptierend.
- (z_7, z_8) verschieden, da z_7 nicht akzeptierend aber z_8 akzeptierend.

Iteration 1:

- (z_2, z_5) verschieden, da $(\delta(z_2, a), \delta(z_5, a)) = (z_3, z_8)$ und (z_3, z_8) bereits verschieden.
- (z_2, z_6) verschieden, da $(\delta(z_2, a), \delta(z_6, a)) = (z_3, z_8)$ und (z_3, z_8) bereits verschieden.
- (z_3, z_5) verschieden, da $(\delta(z_3, a), \delta(z_5, a)) = (z_5, z_8)$ und (z_5, z_8) bereits verschieden.
- (z_3, z_6) verschieden, da $(\delta(z_3, a), \delta(z_6, a)) = (z_5, z_8)$ und (z_5, z_8) bereits verschieden.
- (z_4, z_5) verschieden, da $(\delta(z_4, a), \delta(z_5, a)) = (z_6, z_8)$ und (z_6, z_8) bereits verschieden.
- (z_4, z_6) verschieden, da $(\delta(z_4, a), \delta(z_6, a)) = (z_6, z_8)$ und (z_6, z_8) bereits verschieden.
- (z_5, z_7) verschieden, da $(\delta(z_5, a), \delta(z_7, a)) = (z_8, z_2)$ und (z_8, z_2) bereits verschieden.
- (z_6, z_7) verschieden, da $(\delta(z_6, a), \delta(z_7, a)) = (z_8, z_2)$ und (z_8, z_2) bereits verschieden.

Iteration 2:

- (z_2, z_3) verschieden, da $(\delta(z_2, a), \delta(z_3, a)) = (z_3, z_5)$ und (z_3, z_5) bereits verschieden.
- (z_2, z_4) verschieden, da $(\delta(z_2, a), \delta(z_4, a)) = (z_3, z_6)$ und (z_3, z_6) bereits verschieden.
- (z_3, z_7) verschieden, da $(\delta(z_3, a), \delta(z_7, a)) = (z_5, z_2)$ und (z_5, z_2) bereits verschieden.
- (z_4, z_7) verschieden, da $(\delta(z_4, a), \delta(z_7, a)) = (z_6, z_2)$ und (z_6, z_2) bereits verschieden.

Iteration 3:

- (z_2, z_7) verschieden, da $(\delta(z_2, a), \delta(z_7, a)) = (z_3, z_2)$ und (z_3, z_2) bereits verschieden.

Bewertung:

- 2 Punkte für Entfernen von z_1
- 7 Punkte für die Tabelle:
  4 Punkte für Anfangsmarkierungen, 5 Punkte für Markierungen der Iterationen
- 5 Punkte für den Minimalautomaten:
  1 Punkt Abzug wenn Start- und/oder Endzustand nicht gekennzeichnet sind.

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

a) Die Sprache L_6 sei definiert als

$$L_6 := \{w \mid w = a^m w', w' \in \{a,b\}^*, m = \#_a(w'), m \geq 0\}$$

Informell: Alle Worte aus L_6 lassen sich in zwei Teilworte $w_1 w_2$ aufteilen, so dass w_1 nur aus a's besteht und w_2 genauso viele a's wie w_1 enthält.

(i) Geben Sie eine **kontextfreie Grammatik** G an, die L_6 erzeugt (d.h. $L(G) = L_6$) und erläutern Sie, warum G die Sprache L_6 erzeugt: Beschreiben Sie beispielsweise, welche „Aufgabe" die einzelnen Nichtterminale bei der Erzeugung übernehmen. (6 Punkte)
Geben Sie zusätzlich eine **Linksableitung** für das Wort *aabbbbaa* für Ihre Grammatik an. (2 Punkte)

LÖSUNG:

$$\{S \to aSR \mid B, B \to \varepsilon \mid BB \mid b, R \to bR \mid Rb \mid a\}$$

Erläuterung: S erzeugt zunächst $a^m B R^m$ für $m \in \mathbb{N}$ aus B lassen sich beliebige b's erzeugen, die z.B. für den Fall $m = 0$ nötig sind, aus R lassen sich beliebige Wörter über $\{a,b\}^*$ erzeugen, die genau ein a enthalten (rekursiv werden b's links oder rechts erzeugt und im Basisfall das a).
Linksableitung:

$$S \Rightarrow aSR \Rightarrow aaSRR \Rightarrow aaRR \Rightarrow aabRR \Rightarrow aabbRR \Rightarrow aabbbRR \Rightarrow aabbbbRR \Rightarrow aabbbbaR \Rightarrow aabbbbaa$$

6 Punkte für die Grammatik: Dabei 4 Punkte für die Grammatik und 2 Punkte für die Erläuterung
Ableitungen: 2 Punkte, 1 Punkt Abzug, wenn keine Links-Ableitung.

(ii) Zeigen Sie mit dem Pumpinglemma für reguläre Sprachen, dass

$$L_6 = \{w \mid w = a^m w', w' \in \{a,b\}^*, m = \#_a(w'), m \geq 0\}$$

nicht regulär ist.

(10 Punkte)

Zur Erinnerung:

> **Pumpinglemma 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, so dass 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.
Geeignete Wahlen für z sind z.B.
$z = a^n(ba)^n$ oder $z = a^n b^n a^n$ oder $z = a^n b a^n$
Beweis mit $z = a^n b a^n$:
Sei $z = uvw$ eine beliebige Zerlegung von z mit $|uv| \leq n$ und $|v| \geq 1$.

Dann ist $u = a^{n_1}$, $v = a^{n_2}$, $w = a^{n_3}ba^n$ mit $n = n_1 + n_2 + n_3$ und $n_2 > 0$. Dann ist $uv^0w \notin L_6$, denn $uv^0w = a^{n_1}a^{n_3}ba^n = a^{n-n_2}ba^n$. Damit erfüllt L_6 die Pumpingeigenschaft nicht und das Pumpinglemma zeigt, dass L_6 nicht regulär ist.

- 1 Punkt: $z \in L_6$ gewählt
- 1 Punkt: $|z| \geq n$
- 2 Punkte: z ist geeignet
- 3 Punkte: Über alle Zerlegungen u, v, w argumentiert
- 3 Punkte: Fall gefunden, sodass $uv^kw \notin M$

b) Für eine kontextfreie Grammatik G mit Startsymbol S und Variablen $\{S, T, A, B, C, D, E, F\}$ wurde der CYK-Algorithmus für das Wort *abaaca* ausgeführt. Der Algorithmus hat dabei die folgende Tabelle für die $V(i, j)$-Mengen berechnet:

Wort	a	b	a	a	c	a
$j\backslash i$	1	2	3	4	5	6
1	T, A	B, D, F	T, A	T, A	C	T, A
2	S, B, C, D, E, F			S, B, C		
3	T		S, B, C	T		
4			T			
5	S, T					
6	T					

(i) Geben Sie anhand obiger Tabelle fünf verschiedene Wörter an, die **nicht** in $L(G)$ liegen und begründen Sie Ihre Antwort. (4 Punkte)

(ii) Geben Sie anhand obiger Tabelle drei verschiedene Wörter an, die in $L(G)$ liegen und begründen Sie Ihre Antwort. (6 Punkte)

LÖSUNG: *ab* $\in L$, da $S \in V_{1,2}$ *aca* $\in L$, da $S \in V_{4,2}$ *aaca* $\in L$ da $S \in V_{3,3}$ *abaac* $\in L$, da $S \in V_{1,5}$ alle ande

Lösung Aufgabe 3 (Berechenbarkeit und PCP): **(16 Punkte)**

Wir erinnern zunächst an die Definition des 01-Postschen Korrespondenzproblems:

> Sei $\Sigma = \{0,1\}$ und sei K eine Folge von Wortpaaren $K = ((x_1, y_1), \ldots, (x_k, y_k))$ mit $x_i, y_i \in \Sigma^+$. Das 01-Postsche Korrespondenzproblem (01-PCP) ist die Frage, ob es für die gegebene Folge K eine Folge von Indizes $i_1, \ldots, i_m$ mit $i_j \in \{1, \ldots, k\}$ gibt, sodass $x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$ gilt.

In der Vorlesung wurde die Unentscheidbarkeit des 01-PCP gezeigt.

Für ein Wort $a_1 \cdots a_n$ mit $a_i \in \{0,1\}$ und $n \geq 0$ ist das Komplement $C(a_1 \cdots a_n)$ definiert als $C(a_1) \cdots C(a_n)$ wobei

$$C(x) := \begin{cases} 1, & \text{falls } x = 0 \\ 0, & \text{falls } x = 1 \end{cases}$$

Das Komplement-Postsche Korrespondenzpoblem (CPCP) ist definiert als

> Sei $\Sigma = \{0,1\}$ und sei K eine Folge von Wortpaaren $K = ((x_1, y_1), \ldots, (x_k, y_k))$ mit $x_i, y_i \in \Sigma^+$. Das Komplement-Postsche Korrespondenzproblem (CPCP) ist die Frage, ob es für die gegebene Folge K eine Folge von Indizes $i_1, \ldots, i_m$ mit $i_j \in \{1, \ldots, k\}$ gibt, sodass $C(x_{i_1} \cdots x_{i_m}) = y_{i_1} \cdots y_{i_m}$ gilt.

a) Es sei $K = ((x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4))$ mit

$$\begin{aligned} (x_1, y_1) &= (00, 111) \\ (x_2, y_2) &= (00, 0) \\ (x_3, y_3) &= (11, 0) \\ (x_4, y_4) &= (1, 100) \end{aligned}$$

Alternativ in der Spielstein-Schreibweise

$$K = \left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}, \begin{bmatrix} x_4 \\ y_4 \end{bmatrix} \right) = \left(\begin{bmatrix} 00 \\ 111 \end{bmatrix}, \begin{bmatrix} 00 \\ 0 \end{bmatrix}, \begin{bmatrix} 11 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 100 \end{bmatrix} \right)$$

Zeigen Sie, dass $K \in \text{CPCP}$ gilt.

(Hinweis: Es gibt Lösungen, die nicht mehr als vier Indizes verwenden).

(6 Punkte)

b) Zeigen Sie die Unentscheidbarkeit von CPCP, indem Sie 01-PCP auf CPCP reduzieren. (10 Punkte)

LÖSUNG:

a) (i) Für CPCP können Lösungen nicht mit 2 oder 4 beginnen (da diese mit derselben Zahl beginnen). Mit 1,1,2,3 ist eine Lösung:

$$\begin{bmatrix} 00 \\ 111 \end{bmatrix} \begin{bmatrix} 00 \\ 111 \end{bmatrix} \begin{bmatrix} 00 \\ 0 \end{bmatrix} \begin{bmatrix} 11 \\ 0 \end{bmatrix} = \begin{matrix} 00000011 \\ 11111100 \end{matrix}$$

b) Sei $f((x_1, y_1), \ldots, (x_k, y_k)) = ((C(x_1), y_i), \ldots, (C(x_k), y_k))$ und $f(w) = w$ für ungültige Eingaben.

f ist damit total. f ist berechenbar: Eine 2-Band TM liest und dekodiert die Eingabe $(x_1, y_1), \ldots, (x_k, y_k))$ und schreibt die Ausgabe durch Komplementbildung der x_i (vertausche 0en und 1en) auf das 2. Band.

f ist korrekt, denn:

$$\begin{aligned} K \in 01-PCP \quad & \text{gdw.} \quad \text{es gibt } i_1,\ldots,i_n : x_{i_1}\cdots x_{i_n} = y_{i_1}\cdots y_{i_n} \\ & \text{gdw.} \quad \text{es gibt } i_1,\ldots,i_n : C(C(x_{i_1}))\cdots C(C(x_{i_n})) = y_{i_1}\cdots y_{i_n} \\ & \text{gdw.} \quad \text{es gibt } i_1,\ldots,i_n : C(f(x_{i_1}))\cdots C(f(x_{i_n})) = f(y_{i_1})\cdots f(y_{i_n}) \\ & \text{gdw.} \quad f(K) \in CPCP \end{aligned}$$

Bewertung:

- a) 6 Punkte
- b) 10 Punkte:
  Kodierungsfunktion 6 Punkte
  Korrektheit der Kodierung 2 Punkte
  Funktion ist total (1 Punkt)
  Funktion ist berechenbar (1 Punkt)

Lösung Aufgabe 4 (Gemischte Fragen): **(12 Punkte)**

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

Punkte in dieser Aufgabe werden nach folgender Tabelle vergeben. Eine Antwort, bei der 0 oder mindestens 2 Kästchen angekreuzt sind, zählt als falsch.

Richtige Antworten	0	1	2	3	4	5	6
Erzielte Punkte	0	1	3	6	8	10	12

Seien L_7 und L_8 formale Sprachen.

- [X] Wenn beide Sprachen L_7 und L_8 jeweils regulär sind, dann ist $\overline{L_7} \cap L_7\overline{L_8}$ ebenfalls regulär.
- [ ] Wenn eine der beiden Sprachen L_7 oder L_8 nicht regulär ist, dann ist $L_7 \cup L_8$ ebenfalls nicht regulär.
- [ ] Wenn eine der beiden Sprachen L_7 oder L_8 regulär ist, dann ist $L_7 \cup L_8$ ebenfalls regulär.
- [ ] Wenn beide Sprachen L_7 und L_8 jeweils nicht regulär sind, dann ist $L_7 \cap L_8$ ebenfalls nicht regulär.

Das Präfixfreie Postsche Korrespondenzproblem (PPCP) ist eine Variante des PCP-Problems, welche nur solche Eingaben $((x_1, y_1), \dots, (x_k, y_k))$ verwendet, sodass

für alle $i \in \{1, \dots, k\}$ gilt: x_i ist kein Präfix von y_i und y_i ist kein Präfix von x_i.

Eingaben, die diese Bedingung verletzen, liegen nicht in PPCP. Andere Eingaben liegen in PPCP genau dann, wenn sie in PCP liegen. Es gilt:

- [ ] PPCP ist unentscheidbar, aber semi-entscheidbar
- [ ] PPCP ist entscheidbar, aber nicht semi-entscheidbar
- [ ] PPCP ist unentscheidbar und nicht semi-entscheidbar
- [X] PPCP $\in \mathcal{P}$

Wenn $L_9 \in \mathcal{NP}$, dann

- [X] wird L_9 durch eine deterministische Turingmaschine akzeptiert
- [ ] gilt $L_9 \notin \mathcal{P}$
- [ ] führt jede nichtdeterministische Turingmaschine M mit $L(M) = L_9$ mehr als polynomiell viele Schritte aus, d.h. für jedes Polynom $p(n)$ gilt $ntime_M(|w|) > p(|w|)$ für alle $w \in \Sigma^*$
- [ ] führt jede deterministische Turingmaschine M mit $L(M) = L_9$ höchstens polynomiell viele Schritte aus, d.h. es gibt ein Polynom $p(n)$, sodass $ntime_M(|w|) \leq p(|w|)$ für alle $w \in \Sigma^*$ gilt

Die Aufgabe wird auf dem nächsten Blatt fortgesetzt.

Fortsetzung von Aufgabe 4:

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

$$\begin{aligned} P = \{ & A \to a, \\ & A \to BC, \\ & B \to b, \\ & B \to C, \\ & BC \to CD, \\ & D \to b, \\ & D \to B\} \end{aligned}$$

ist

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

Die Sprache $L_{10} := \{w \mid \text{Turingmaschine } M_w \text{ entscheidet das Halteproblem}\}$ ist

- [X] eine reguläre Sprache
- [ ] unentscheidbar, aber semi-entscheidbar
- [ ] unentscheidbar, aber das Komplement $\overline{L_{10}}$ ist semi-entscheidbar
- [ ] vom Typ 0, aber nicht vom Typ 1

Typ 0-Sprachen

- [ ] sind **nicht** unter Schnitt **abgeschlossen**
- [ ] sind unter Komplement **abgeschlossen**
- [X] sind unter Vereinigung **abgeschlossen**
- [ ] sind **nicht** rekursiv-aufzählbar