

Hauptklausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2011/2012

Hier Aufkleber mit Name und Matrikelnr. anbringen

Vorname: _____

Nachname: _____

Matrikelnummer: _____

Beachten Sie:

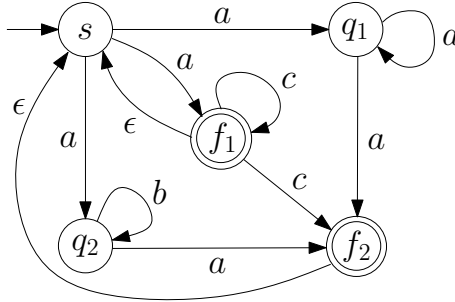
- Bringen Sie den Aufkleber mit Ihrem Namen und Ihrer Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihrem Namen und Ihrer Matrikelnummer.
- Schreiben Sie die Lösungen auf die Aufgabenblätter und Rückseiten. Zusätzliches Papier erhalten Sie bei Bedarf von der Aufsicht.
- Zum Bestehen der Klausur sind **20** der möglichen **60** Punkte hinreichend.
- Es sind keine Hilfsmittel zugelassen.

Aufgabe	Mögliche Punkte					Erreichte Punkte				
	a	b	c	d	Σ	a	b	c	d	Σ
1	1	4	-	-	5			-	-	
2	1	3	-	-	4			-	-	
3	3	4	-	-	7			-	-	
4	5	-	-	-	5		-	-	-	
5	4	1	-	-	5			-	-	
6	4	3	-	-	7			-	-	
7	3	-	-	-	3	-		-	-	
8	1	2	1	1	5					
9	2	1	3	3	9					
10	10x1				10					
Σ					60					

Aufgabe 1:

(1 + 4 Punkte)

Gegeben sei der folgende endliche Automat \mathcal{A} über dem Alphabet $\Sigma = \{a, b, c\}$.



- (a) Geben Sie die von \mathcal{A} akzeptierte Sprache $L(\mathcal{A})$ als regulären Ausdruck an. Es ist nicht verlangt, dass Sie hierzu das Verfahren aus der Vorlesung benutzen.

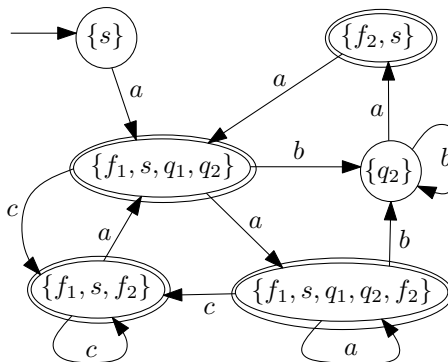
Lösung:

$$(a(a^+ \cup c^* \cup b^*a))^+$$

- (b) Konstruieren Sie mit Hilfe der Potenzmengenkonstruktion einen deterministischen endlichen Automaten \mathcal{A}' , der dieselbe Sprache akzeptiert wie \mathcal{A} . Geben Sie das Ergebnis als Zustandsübergangsdiagramm an und bezeichnen Sie die Zustände mit den Teilmengen der Zustände in \mathcal{A} , die sie repräsentieren.

Lösung:

	a	b	c
$\{s\}$	$\{f_1, s, q_1, q_2\}$	\emptyset	\emptyset
$\{f_1, s, q_1, q_2\}$	$\{f_1, s, q_1, q_2, f_2\}$	$\{q_2\}$	$\{f_1, s, f_2\}$
$\{f_1, s, q_1, q_2, f_2\}$	$\{f_1, s, q_1, q_2, f_2\}$	$\{q_2\}$	$\{f_1, s, f_2\}$
$\{q_2\}$	$\{f_2, s\}$	$\{q_2\}$	\emptyset
$\{f_1, s, f_2\}$	$\{f_1, s, q_1, q_2\}$	\emptyset	$\{f_1, s, f_2\}$
$\{f_2, s\}$	$\{f_1, s, q_1, q_2\}$	\emptyset	\emptyset



Aufgabe 2:

(1+3 Punkte)

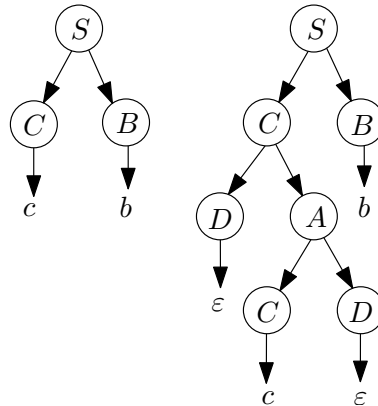
Gegeben sei die Grammatik $G = (\Sigma, V, S, R)$ mit Terminalen $\Sigma = \{a, b, c, d\}$, Nichtterminalen $V = \{S, A, B, C, D\}$, Startsymbol S und Produktionen

$$\begin{aligned} R = \{ & S \rightarrow CB \\ & A \rightarrow a \mid CD \\ & B \rightarrow b \\ & C \rightarrow DA \mid AaDC \mid c \\ & D \rightarrow C \mid d \mid \epsilon \}. \end{aligned}$$

- (a) Zeigen oder widerlegen Sie die folgende Aussage: Die Grammatik G ist eindeutig.

Lösung:

Die Grammatik G ist nicht eindeutig, denn für cb gibt es verschiedene Syntaxbäume:



- (b) Geben Sie eine Grammatik G' in Chomsky-Normalform an, die dieselbe Sprache wie G erzeugt. Gehen Sie dabei systematisch vor, und geben Sie die einzelnen Umformungsschritte in nachvollziehbarer Form an.

Lösung:

(I) Terminale rechts nur einzeln

$$\begin{aligned} R = \{ & S \rightarrow CB \\ & A \rightarrow a \mid CD \\ & B \rightarrow b \\ & C \rightarrow DA \mid AY_aDC \mid c \\ & D \rightarrow C \mid d \mid \epsilon \\ & Y_a \rightarrow a \}. \end{aligned}$$

(II) Rechte Seite hat Länge 2

$$\begin{aligned}
 R = \{ & S \rightarrow CB \\
 & A \rightarrow a \mid CD \\
 & B \rightarrow b \\
 & C \rightarrow DA \mid AX_1 \mid c \\
 & D \rightarrow C \mid d \mid \epsilon \\
 & Y_a \rightarrow a \\
 & X_1 \rightarrow Y_a X_2 \\
 & X_2 \rightarrow DC \}.
 \end{aligned}$$

(III) ϵ entfernen

$$\begin{aligned}
 R = \{ & S \rightarrow CB \\
 & A \rightarrow a \mid CD \mid C \\
 & B \rightarrow b \\
 & C \rightarrow DA \mid A \mid AX_1 \mid c \\
 & D \rightarrow C \mid d \\
 & Y_a \rightarrow a \\
 & X_1 \rightarrow Y_a X_2 \\
 & X_2 \rightarrow DC \mid C \}.
 \end{aligned}$$

(IV) Kettenregeln entfernen. Einziger Kreis ist $A \rightarrow C \rightarrow A$

$$\begin{aligned}
 R = \{ & S \rightarrow AB \\
 & A \rightarrow a \mid AD \mid DA \mid AX_1 \mid c \\
 & B \rightarrow b \\
 & D \rightarrow A \mid d \\
 & Y_a \rightarrow a \\
 & X_1 \rightarrow Y_a X_2 \\
 & X_2 \rightarrow DA \mid A \}.
 \end{aligned}$$

Topologische Sortierung: D, X_2, A , rückwärts einsetzen

$$\begin{aligned}
 R = \{ & S \rightarrow AB \\
 & A \rightarrow a \mid AD \mid DA \mid AX_1 \mid c \\
 & B \rightarrow b \\
 & D \rightarrow a \mid AD \mid DA \mid AX_1 \mid c \mid d \\
 & Y_a \rightarrow a \\
 & X_1 \rightarrow Y_a X_2 \\
 & X_2 \rightarrow DA \mid a \mid AD \mid DA \mid AX_1 \mid c \}.
 \end{aligned}$$

Aufgabe 3:

(3+4 Punkte)

- (a) Geben Sie eine kontextfreie Grammatik an, die die Sprache $L := \{w \in \{a, b\}^* : |w|_a \geq 3|w|_b\}$ erzeugt. Dabei sei $|w|_x$ die Häufigkeit des Zeichens $x \in \{a, b\}$ im Wort w .

Lösung:
$$S \rightarrow SS \mid a \mid \epsilon \mid bSaSaSa \mid aSbSaSa \mid aSaSbSa \mid aSaSaSb$$

- (b) Zeigen Sie, dass die Sprache L aus Teilaufgabe (a) maximal vom Chomsky-Typ 2 ist, d.h., dass L nicht regulär ist.

Lösung:

Annahme: L ist regulär, d.h., das Pumpinglemma für reguläre Sprachen gilt für L . Sei also n die untere Schranke der Wortlänge aus dem Pumpinglemma. Betrachte das Wort $w = b^n a^{3n} \in L$ mit $|w| \geq n$. Dann gilt für jede Zerlegung $w = uvx$ mit $|uv| \leq n$ und $v \neq \epsilon$, dass $v = b^k$, $k \geq 1$. Damit ist $uv^i x = b^{n-k} b^{2k} a^{3n} = b^{n+k} a^{3n} \notin L$ für $i = 2$ (da $3(n+k) = 3n + 3k > 3n$), im Widerspruch zum Pumpinglemma.

Aufgabe 4:

(5 Punkte)

Ein *einfacher Pfad* in einem gerichteten, gewichteten Graphen $G = (V, E, c)$ mit einer natürlichen Kostenfunktion $c : E \rightarrow \mathbb{N}_0$ ist eine Folge $P = (v_1, \dots, v_r)$ von Knoten, die jeden Knoten höchstens einmal enthält und für die $(v_i, v_{i+1}) \in E$ für $1 \leq i \leq r - 1$ gilt. Die Länge ℓ eines Pfades $P = (v_1, \dots, v_r)$ ist definiert als $\ell := \sum_{i=1}^{r-1} c((v_i, v_{i+1}))$.

Problem ℓ -PFAD

Gegeben: Gerichteter, gewichteter Graph $G = (V, E, c)$ mit $c : E \rightarrow \mathbb{N}_0$, Parameter $\ell \in \mathbb{N}_0$.

Frage: Gibt es einen einfachen Pfad der Länge ℓ in G ?

Zeigen Sie, dass das Problem ℓ -PFAD \mathcal{NP} -vollständig ist. Sie dürfen dazu benutzen, dass das Problem SUBSETSUM \mathcal{NP} -vollständig ist:

Problem SUBSETSUM

Gegeben: Endliche Menge M , Gewichtsfunktion $w : M \rightarrow \mathbb{N}_0$, Parameter $k \in \mathbb{N}_0$.

Frage: Gibt es eine Teilmenge $M' \subseteq M$ mit $\sum_{m \in M'} w(m) = k$?

Lösung:

ℓ -PFAD ist in \mathcal{NP} da für eine gegebene Knotenfolge (v_1, \dots, v_r) in polynomieller Zeit geprüft werden kann, ob zwischen jeweils aufeinanderfolgenden Knoten eine Kante existiert (also ob die Folge einen Pfad bildet). Gleichzeitig können bei dieser Prüfung die existierenden Kantengewichte aufsummiert werden und somit geprüft werden, ob diese Summe ℓ entspricht.

Polynomielle Transformation von SUBSETSUM auf ℓ -PFAD: Sei $I = (M, \{w(m_1), \dots, w(m_n)\}, k)$ eine Instanz von SUBSETSUM mit $|M| = n$. Konstruiere in polynomieller Zeit folgende Instanz von ℓ -PFAD:

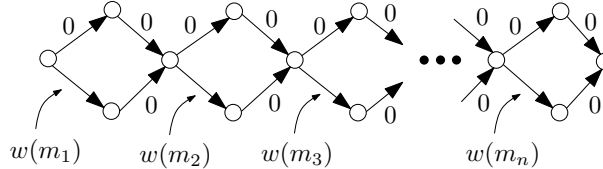


Abbildung 1: Graph G mit $3|M| + 1$ und Kostenfunktion wie an Kanten angegeben, $\ell := k$.

Ist P ein einfacher Pfad der Länge $\ell = k$ in G , so induzieren die Kanten mit echt positiven Kosten eine Teilmenge $M' \subseteq M$ mit $\sum_{m \in M'} w(m) = k$. Existiert eine Teilmenge $M' \subseteq M$ mit $\sum_{m \in M'} w(m) = k$, so induzieren die Gewichte dieser Teilmenge eine Kantenmenge E' in G , die durch Kanten mit Kosten Null zu einem einfachen Pfad der Länge $\ell = k$ verbunden werden können.

Aufgabe 5:

(4 + 1 Punkte)

Gegeben sei folgende Sprache

$$L := \{w \in \{a, b\}^* : |w|_a = |w|_b \text{ und f\u00fcr jede Zerlegung } w = uv \text{ mit } |u| \geq 1 \text{ gilt } |u|_a \geq |u|_b\}.$$

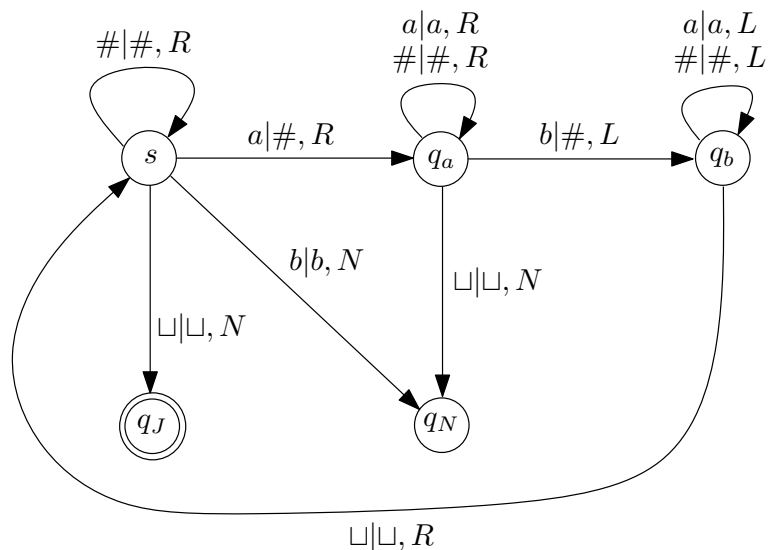
Dabei sei $|w|_x$ die H\u00e4ufigkeit des Zeichens $x \in \{a, b\}$ im Wort w .

- (a) Zeigen Sie, dass L entscheidbar ist, indem Sie eine deterministische Turingmaschine $\mathcal{M} = (Q, \Sigma = \{a, b\}, \Gamma, \delta, s, F = \{q_J\})$ mit Zust\u00e4nden $q_N, q_J \in Q$ angeben, die f\u00fcr jede Eingabe $w \in \Sigma^*$ entweder in q_J oder q_N h\u00e4lt und L durch Halten in q_J akzeptiert. Stellen Sie die \u00dcbergangsfunktion δ von \mathcal{M} graphisch durch ein Zustands\u00fcbergangendiagramm dar und bezeichnen Sie den Startzustand mit s . Beschreiben Sie kurz in Worten, wie Ihre Turingmaschine arbeitet.

L\u00f6sung:

Beschreibung:

TM liest Eingabe von Links nach Rechts, sobald ein a gelesen wird, wird dieses mit $\#$ \u00fcberschrieben und TM wechselt in neuen Zustand, der anzeigt, dass nun ein zugeh\u00f6riges b gesucht wird (bzw. dass a gelesen wurde). Falls zugeh\u00f6riges b nicht gefunden wird, h\u00e4lt die TM in q_N . Falls Eingabe in L , so gibt es dieses b immer. Gefundenes b wird ebenfalls mit $\#$ \u00fcberschrieben und TM geht zum Wortanfang zur\u00fcck. \u00dcbriges Wort auf dem Band (ohne $\#$'s) ist wieder in L . Ablauf wird so oft wiederholt bis nur noch $\#$'s auf dem Band stehen. Dann wird akzeptiert. Falls Eingabe nicht in L ist, so h\u00e4lt TM vorher in q_N .

Bemerkung: L ist die Sprache der korrekten Klammerausdr\u00fccke.\u00dcbergangendiagramm mit $\Gamma = \Sigma \cup \{\#, \sqcup\}$ 

- (b) Dokumentieren Sie die Berechnung des Wortes $aababb \in L$ sowie die Berechnung des Wortes $abbaab \notin L$ durch die von Ihnen angegebene Turingmaschine \mathcal{M} . Geben Sie dazu für jeden Schritt die aktuelle Konfiguration an.

Lösung:

$abbaab \notin L$:

$\sqcup(s)abbaab, \sqcup\#(q_a)bbaab, \sqcup(q_b)\#\#baab, \sqcup(q_b)\sqcup\#\#baab,$
 $\sqcup(s)\#\#baab, \sqcup\#(s)\#baab, \sqcup\#\#(s)baab, \sqcup\#\#(q_N)baab$

$aababb \in L$:

$\sqcup\#(s)aababb, \sqcup\#(q_a)ababb, \sqcup\#a(q_a)babb, \sqcup\#(q_b)a\#abb, \sqcup(q_b)\#a\#abb, (q_b)\sqcup\#a\#abb$
 $\sqcup\#(s)\#a\#abb, \sqcup\#(s)a\#abb, \sqcup\#\#(q_a)\#abb, \sqcup\#\#\#(q_a)abb, \sqcup\#\#\#a(q_a)bb, \sqcup\#\#\#(q_b)a\#b$
 $\sqcup\#\#\#(q_b)\#a\#b, \sqcup\#(q_b)\#\#a\#b, \sqcup(q_b)\#\#\#a\#b, (q_b)\sqcup\#\#\#a\#b, \sqcup(s)\#\#\#a\#b, \sqcup\#(s)\#\#a\#b$
 $\sqcup\#\#\#(s)\#a\#b, \sqcup\#\#\#\#(s)a\#b, \sqcup\#\#\#\#\#(q_a)\#b, \sqcup\#\#\#\#\#\#(q_a)b, \sqcup\#\#\#\#\#(q_b)\#\#, \dots$
 $(q_b)\sqcup\#\#\#\#\#\#, \sqcup(s)\#\#\#\#\#\#, \sqcup\#(s)\#\#\#\#\#, \dots, \sqcup\#\#\#\#\#\#(s)\sqcup, \sqcup\#\#\#\#\#\#(q_J)\sqcup$

Aufgabe 6:

(4+3 Punkte)

- (a) Sei
- L
- eine nicht entscheidbare Sprache.

Zeigen Sie: $L^R := \{w^R \mid w \in L \text{ und } w^R \text{ ist das Spiegelwort zu } w\}$ ist nicht entscheidbar.**Lösung:**

Annahme: L^R sei entscheidbar, und \mathcal{M}^R sei eine TM, die L^R entscheidet. Konstruiere eine neue TM M' , die zu einer Eingabe w das Spiegelwort w^R berechnet und dann die TM \mathcal{M}^R auf w^R simuliert. Die TM M' hält immer, da \mathcal{M}^R immer hält. Die TM M' akzeptiert w genau dann, wenn \mathcal{M}^R w^R akzeptiert, also genau dann wenn $w \in L$. Damit entscheidet M' die Sprache L . Widerspruch!
Somit kann es eine solche TM \mathcal{M}^R nicht geben und L^R ist nicht entscheidbar.

- (b) Sei
- $w \in \{0,1\}^*$
- und
- \mathcal{M}_w
- die Turingmaschine, die durch die Gödelnummer
- w
- beschrieben wird. Zeigen Sie, dass die Diagonalsprache
- $L_d := \{w \in \{0,1\}^* \mid \mathcal{M}_w \text{ akzeptiert } w \text{ nicht}\}$
- nicht semi-entscheidbar ist.

Lösung:

Annahme: L_d sei semi-entscheidbar. Dann gibt es eine Turingmaschine \mathcal{M}' , die sich undefiniert verhält (und insbesondere die Eingabe nicht akzeptiert), falls die Eingabe $w \notin L_d$, und die die Eingabe w akzeptiert, falls $w \in L_d$. Sei nun w' die Gödelnummer von \mathcal{M}' .

Falls $w' \in L_d$, so akzeptiert \mathcal{M}' die Eingabe w' . Damit ist aber $w' \notin L_d$ nach Definition von L_d . Widerspruch!

Falls $w' \notin L_d$, so akzeptiert \mathcal{M}' die Eingabe w' nicht. Damit ist aber $w' \in L_d$ nach Definition von L_d . Widerspruch!

Eine solche TM \mathcal{M}' kann es also nicht geben. Damit ist L_d nicht semi-entscheidbar.

Aufgabe 7:

(3 Punkte)

Sei L_1 eine kontextfreie Sprache und L_2 eine reguläre Sprache. Zeigen Sie, dass $L_1 \cap L_2$ kontextfrei ist.

Lösung:

Sei $\mathcal{K} = (Q_{\mathcal{K}}, \Sigma_{\mathcal{K}}, \Gamma, \delta_{\mathcal{K}}, s_{\mathcal{K}}, Z, F_{\mathcal{K}})$ ein PDA, der L_1 akzeptiert und $\mathcal{A} = (Q_{\mathcal{A}}, \Sigma_{\mathcal{A}}, \delta_{\mathcal{A}}, s_{\mathcal{A}}, F_{\mathcal{A}})$ ein DEA, der L_2 akzeptiert.

Falls $\Sigma_{\mathcal{A}} \cap \Sigma_{\mathcal{K}} = \emptyset$, so gilt $L_1 \cap L_2 = \emptyset$ oder $L_1 \cap L_2 = \{\varepsilon\}$. In beiden Fällen ist $L_1 \cap L_2$ regulär, also auch kontextfrei. Falls $\Sigma_{\mathcal{A}} \cap \Sigma_{\mathcal{K}} \neq \emptyset$, konstruiere einen Produktautomaten $\mathcal{A} \times \mathcal{K} = (Q_{\mathcal{A}} \times Q_{\mathcal{K}}, \Sigma_{\mathcal{A}} \cap \Sigma_{\mathcal{K}}, \Gamma, (s_{\mathcal{A}}, s_{\mathcal{K}}), Z, \delta, F_{\mathcal{A}} \times F_{\mathcal{K}})$

mit $\delta : (Q_{\mathcal{A}} \times Q_{\mathcal{K}}) \times ((\Sigma_{\mathcal{A}} \cap \Sigma_{\mathcal{K}}) \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{(Q_{\mathcal{A}} \times Q_{\mathcal{K}}) \times \Gamma^*}$ wie folgt:

$$\delta((q_{\mathcal{A}}, q_{\mathcal{K}}), \varepsilon, X) := \{((q_{\mathcal{A}}, p_{\mathcal{K}}), \gamma) \mid (p_{\mathcal{K}}, \gamma) \in \delta_{\mathcal{K}}(q_{\mathcal{K}}, \varepsilon, X)\}$$

$$\delta((q_{\mathcal{A}}, q_{\mathcal{K}}), a, X) := \{((p_{\mathcal{A}}, p_{\mathcal{K}}), \gamma) \mid p_{\mathcal{A}} = \delta_{\mathcal{A}}(q_{\mathcal{A}}, a) \text{ und } (p_{\mathcal{K}}, \gamma) \in \delta_{\mathcal{K}}(q_{\mathcal{K}}, a, X)\}$$

Der Produktautomat $\mathcal{A} \times \mathcal{K}$ ist ein PDA, der ein Wort w genau dann akzeptiert, wenn \mathcal{A} und \mathcal{K} nach Abarbeitung von w in einem Endzustand sind. Damit ist $L(\mathcal{A} \times \mathcal{K}) = L_1 \cap L_2$ und $L_1 \cap L_2$ ist kontextfrei.

Aufgabe 8:

(1+2+1+1 Punkte)

Sei $\mathcal{A} = (Q, \Sigma, \Gamma, s, Z, \delta, \{f\})$ der Kellerautomat mit Zustandsmenge $Q = \{s, f\}$, Eingabealphabet $\Sigma = \{0, 1\}$, Stack-Alphabet $\Gamma = \{Y, Z\}$, Anfangszustand s , Stack-Initialisierung Z , einzigem Endzustand f und der folgenden Übergangsrelation δ :

$$(s, 1, Z) \mapsto (s, ZY)$$

$$(s, \varepsilon, Z) \mapsto (f, \varepsilon)$$

$$(f, 0, Y) \mapsto (f, \varepsilon)$$

- (a) Ist \mathcal{A} deterministisch? Begründen Sie Ihre Antwort.

Lösung:

Nein, \mathcal{A} ist nicht deterministisch, da $|\delta(s, 1, Z)| + |\delta(s, \varepsilon, Z)| > 1$.

- (b) Dokumentieren Sie eine durch Endzustand akzeptierende Berechnung des Wortes 1110. Geben Sie dazu für jeden Schritt die aktuelle Konfiguration an.

Lösung:

- $(s, 1110, Z)$
- $(s, 110, ZY)$
- $(s, 10, ZYY)$
- $(s, 0, ZYYY)$
- $(f, 0, YYY)$
- (f, ε, YY)

- (c) Welche Sprache akzeptiert \mathcal{A} durch akzeptierenden Endzustand?

Lösung:

$$L = \{1^i 0^j \mid i, j \in \mathbb{N}_0, j \leq i\}$$

- (d) Welche Sprache akzeptiert \mathcal{A} durch leeren Stack?

Lösung:

$$L = \{1^i 0^i \mid i \in \mathbb{N}_0\}$$

Aufgabe 9:

(2+1+3+3 Punkte)

Gegeben sei eine endliche Menge M mit $|M| \geq 2$ und eine Gewichtsfunktion $w : M \rightarrow \mathbb{N}$.

Für $M' \subseteq M$ sei $w(M') := \sum_{m \in M'} w(m)$.

Das Optimierungsproblem BALANCED PARTITION besteht darin, M so in zwei Teilmengen $A \subseteq M$ und $B := M \setminus A$ aufzuteilen, dass $\min\{w(A), w(B)\}$ maximal ist.

Betrachten Sie folgenden Algorithmus:

Algorithmus 1: BALANCED PARTITION APPROXIMATION

Eingabe : Endliche Menge M mit $|M| \geq 2$ und Gewichtsfunktion $w : M \rightarrow \mathbb{N}$

Ausgabe : Teilmengen $A \subseteq M$ und $B := M \setminus A$

Sortiere M nicht-aufsteigend nach Gewichten. Danach sei $M[i]$ das Element an i -ter Stelle,

d.h. $w(M[i]) \geq w(M[i+1])$ für $1 \leq i \leq n-1$;

$A \leftarrow \{M[i] \mid i \text{ ungerade}\}$;

$B \leftarrow \{M[i] \mid i \text{ gerade}\}$;

- (a) Gegeben sei die folgende BALANCED PARTITION Instanz $I = (M, w)$ mit $M = \{m_1, m_2, m_3, m_4\}$ und Gewichtsfunktion w gegeben durch

$$w(m_1) = 8, \quad w(m_2) = 8, \quad w(m_3) = 10, \quad w(m_4) = 5.$$

Geben Sie eine optimale Lösung für die Instanz I an. Welchen Wert hat jede optimale Lösung?

Lösung:

- $A = \{m_1, m_2\}$, $B = \{m_3, m_4\}$
- Wert: 15

Geben Sie zusätzlich die Lösung an, die Algorithmus 1 bei Eingabe von I berechnet. Welchen Wert hat diese Lösung?

Lösung:

- Sortierung: 10, 8, 8, 5
- $A = \{m_3, m_1\}$, $B = \{m_2, m_4\}$
- Wert: 13

- (b) Zeigen Sie, dass für jede Lösung, die Algorithmus 1 für eine beliebige Instanz zurück gibt, gilt:

$$w(A) \geq w(B) \geq w(A) - w_{\max}$$

wobei $w_{\max} := \max\{w(m) \mid m \in M\}$.

Lösung:

Bemerkung: $w_{\max} = w(M[1])$.

Es gilt $M[1] \in A$ für alle n und $M[n] \in A$ falls n ungerade und $M[n] \in B$ falls n gerade. Damit ist

- $w(M[i]) \in A \geq w(M[i+1]) \in B$ für $i \leq n-1$ ungerade, und insgesamt $w(A) \geq w(B)$ falls n gerade, und sogar $w(A) - w(M[n]) \geq w(B)$ falls n ungerade.
- $w(M[i]) \in B \geq w(M[i+1]) \in A$ für $i \leq n-1$ gerade, und insgesamt $w(B) \geq w(A) - w_{\max}$ falls n ungerade, und sogar $w(B) - w(M[n]) \geq w(A) - w_{\max}$ falls n gerade.

- (c) Zeigen Sie, dass Algorithmus 1 1-approximativ ist, d.h. dass Algorithmus 1 eine relative Gütegarantie von $1 + 1 = 2$ hat.

Lösung:

Mit b) folgt $2w(B) \geq w(B) + w(A) - w_{\max} = W - w_{\max}$

und $\mathcal{A}(I) = w(B) \geq (W - w_{\max})/2$.

Für den optimalen Wert $OPT(I)$ gilt: $OPT(I) \leq W/2$.

Fall 1: $w_{\max} \geq W/2 \Rightarrow OPT(I) = W - w_{\max}$. Damit ist $\mathcal{A}(I) = w(B) \geq OPT(I)/2$.

Fall 2: $w_{\max} < W/2 \Rightarrow w(B) \geq (W - W/2)/2 = W/4$. Dann gilt $OPT(I)/\mathcal{A}(I) \leq (W/2)/(W/4) = 2$.

- (d) Zeigen Sie: Falls $\mathcal{P} \neq \mathcal{NP}$, gibt es keinen absoluten Approximationsalgorithmus für BALANCED PARTITION. Sie dürfen benutzen, dass BALANCED PARTITION \mathcal{NP} -schwer ist.

Lösung:

Annahme: Es gibt einen absoluten Approximationsalgorithmus \mathcal{A}' mit $|OPT(I) - \mathcal{A}'(I)| \leq K$ für alle I . Betrachte Instanz I' , die aus einer beliebigen Instanz I hervorgeht, indem alle Gewichte mit $K + 1$ multipliziert werden. Dann gilt $OPT(I') = (K + 1)OPT(I)$.

Die Lösung $\mathcal{L}_{\mathcal{A}'}(I')$ ist ebenfalls eine Lösung für I , allerdings hat diese Lösung bezüglich I den Wert $a := \frac{\mathcal{A}'(I')}{K+1}$.

Damit gilt $|OPT(I') - \mathcal{A}'(I')| = |(K+1)OPT(I) - (K+1)a| \leq K$, und so ist $|OPT(I) - a| \leq \frac{K}{K+1} < 1$ und a entspräche dem Optimalwert, \mathcal{L} wäre somit eine optimale Lösung für I , die polynomiell berechnet werden kann, da \mathcal{A}' sowie die Umformung zwischen I und I' polynomiell sind. Dies ist ein Widerspruch zu $\mathcal{P} \neq \mathcal{NP}$, da Balanced Partition \mathcal{NP} -schwer ist.

Aufgabe 10:

(10 Punkte)

Jeder der folgenden Aussagenblöcke umfasst drei Einzelaussagen. Die sich unterscheidenden Aussagenbausteine sind durch Kästchen gekennzeichnet. Kreuzen Sie genau jene Bausteine an, die in einer wahren Einzelaussage enthalten sind. Jeder Aussagenblock enthält mindestens eine wahre Einzelaussage. Unvollständig oder falsch angekreuzte Aussagenblöcke werden mit null Punkten bewertet, Sie erhalten einen Punkt für jeden Aussagenblock, für den Sie genau die richtige Menge an Aussagenbausteinen angekreuzt haben.

Die Sprache $L = \{(yx^i z)(zxz)^j | i, j \in \mathbb{N}, i < 19, j \leq 17\}$

- ist maximal vom Chomsky-Typ 0.
- wird von keiner Grammatik erzeugt.
- ist regulär.

Lösung: Sprache ist endlich, also regulär: c)

Ist L eine

- reguläre
- nichtentscheidbare
- kontextsensitive

Sprache so gilt: Für jedes Wort $w \in L$ gibt es einen DEA \mathcal{A}_w , der w akzeptiert.

Lösung: Dies gilt für alle Sprachen: a), b), c)

Jede Sprache, die von einer Grammatik erzeugt wird, ist

- endlich.
- entscheidbar.
- semi-entscheidbar.

Lösung: c)

Für die regulären Ausdrücke (a^*b^+) und $(a^*b)^*$ gilt:

- $(a^*b^+) = (a^*b)^*$.
- $(a^*b^+) \subseteq (a^*b)^*$.
- $(a^*b^+) \supset (a^*b)^*$.

Lösung: b)

Ist Σ ein endliches Alphabet, so ist Σ^*

- kontextfrei.
- NP-vollständig.
- entscheidbar.

Lösung: a),c)

Das Entscheidungsproblem SAT ist

- unentscheidbar.
- semi-entscheidbar.
- in \mathcal{NP} .

Lösung: SAT ist in NP und damit entscheidbar: b),c)

Falls es einen Approximationsalgorithmus für CLIQUE mit absoluter Gütegarantie gibt, so gilt:

- $\mathcal{P} = \mathcal{NP}$.
- $\mathcal{NPC} = \mathcal{P} \setminus \{\Sigma^*, \emptyset\}$.
- $\mathcal{NPI} \neq \emptyset$.

Lösung: a),b)

Für nicht entscheidbare Sprachen L gilt:

- Die zugehörige Nerode-Relation hat unendlichen Index.
- Die Komplementsprache L^c ist nicht semi-entscheidbar.
- $L \notin \mathcal{NPC}$.

Lösung: a) denn reguläre Sprachen sind entscheidbar, c) denn Sprachen in \mathcal{NP} sind entscheidbar, b) nicht, siehe L_u und L_u^c

Für das Optimierungsproblem KNAPSACK gibt es

- einen pseudopolynomiellen optimalen Algorithmus.
- ein *FPAS*.
- ein *PAS*.

Lösung: a),b),c) siehe Skript Satz 4.47

Zu jeder Sprache L , die durch einen nichtdeterministischen Kellerautomaten mit leerem Stack akzeptiert wird, gibt es

- einen deterministischen Kellerautomaten, der L mit leerem Stack akzeptiert.
- einen nichtdeterministischen Kellerautomaten, der L durch akzeptierenden Endzustand akzeptiert.
- eine Grammatik vom Chomsky-Typ 1, die L erzeugt.

Lösung: b), c), denn L ist kontextfrei