1. Klausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2018/2019

Lösung!

Beachten Sie:

- Bringen Sie den Aufkleber mit Ihrem Namen und Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihrem Namen und Matrikelnummer.
- Schreiben Sie die Lösungen auf die Aufgabenblätter und Rückseiten. Zusätzliches Papier erhalten Sie bei Bedarf von der Aufsicht.
- Es sind keine Hilfsmittel zugelassen.

	Mögliche Punkte					Erreichte Punkte				
	a	b	c	d	Σ	a	b	c	d	Σ
Aufg. 1	3	2	2	2	9					
Aufg. 2	4	1	4	–	9				–	
Aufg. 3	6				6					
Aufg. 4	2	2	2	3	9					
Aufg. 5	1	1	9	–	11				–	
Aufg. 6	5	1	2	–	8				–	
Aufg. 7	1	3	1	3	8					
Σ					60					

Problem 1: Unendliche Automaten $3 + 2 + 2 + 2 = 9$ Punkte

In der Vorlesung und in der Übung haben wir uns intensiv mit endlichen Automaten beschäftigt.

In dieser Aufgabe geht es um deterministische *unendliche* Automaten. Ein deterministischer unendlicher Automat ist ein Tupel $\mathcal{U} = (Q, \Sigma, \delta, s, F)$, wobei Q eine nicht notwendigerweise endliche Zustandsmenge ist, Σ ein endliches Eingabealphabet, $s \in Q$ der Anfangszustand, $F \subseteq Q$ eine nicht notwendigerweise endliche Menge von akzeptierenden Zuständen und $\delta : Q \times \Sigma \to Q$ die Zustandsüberführungsfunktion.

Im Gegensatz zu den Ihnen vertrauten deterministischen endlichen Automaten hat sich also nur verändert, dass Q und F jetzt unendlich groß sein dürfen.

Wir sagen, dass ein Automat $\mathcal{A}$ in *Baumnormalform* ist, wenn der Zustandsübergangsgraph ein gerichteter Baum ist, wenn also der Eingangsgrad aller Zustandsknoten höchstens Eins ist.

(a) Sei $L \subseteq \Sigma^\star$ eine beliebige Sprache. Geben Sie einen deterministischen unendlichen Automaten $\mathcal{U}_L$ in Baumnormalform an, der L entscheidet.

(b) Begründen Sie, dass $\mathcal{U}_L$ in Baumnormalform ist.

(c) Zeigen Sie, dass L von $\mathcal{U}_L$ entschieden wird.

(d) Aus den vorigen Teilaufgaben folgt, dass deterministische unendliche Automaten auch Sprachen entscheiden können, die von Turingmaschinen nicht entschieden werden können. Es ist also nicht möglich, den Beweis, dass die Diagonalsprache unentscheidbar ist, auf unendliche Automaten zu übertragen. Erklären Sie, woran das liegt.

Lösung:

(a) $\mathcal{U}_L = (\Sigma^\star, \Sigma, \delta, \varepsilon, L)$ mit $\delta(w, a) = wa$ für alle $w \in Q$ und $a \in \Sigma$.

(b) Weil ein Zustand wa nur vom Zustand w aus erreichbar ist, ist $\mathcal{U}_L$ in Baumnormalform.

(c) Sei $w \in \Sigma^\star$ ein beliebiges Wort. Nach Konstruktion befindet sich $\mathcal{U}_L$ nach Abarbeitung von w im Zustand w. Der Zustand w ist genau dann akzeptierend, wenn $w \in L$ gilt, also akzeptiert $\mathcal{U}_L$ genau L.

(d) Die Diagonalsprache ist definiert als $L_d := \{w_i \mid \mathcal{M}_i \text{ akzeptiert } w_i \text{ nicht}\}$. Diese Definition setzt voraus, dass es eine eindeutige Zuordnung zwischen Wörtern w_i und Turingmaschinen $\mathcal{M}_i$ gibt. Bei unendlichen Automaten kann es so eine Zuordnung nicht geben, weil es überabzählbar viele Sprachen (und somit überabzählbar viele unendliche Automaten) gibt, aber nur abzählbar viele Wörter.

Angenommen, es gäbe eine solche Zuordnung, d.h. wir könnten für jeden deterministischen unendlichen Automaten eine eindeutige Gödelnummer angeben. Nach Aufgabenteil (a) gibt es einen deterministischen unendlichen Automaten $\mathcal{U}$, der L_d entscheidet. Für die Gödelnummer $\langle\mathcal{U}\rangle$ gilt dann: Wenn $\langle\mathcal{U}\rangle$ von $\mathcal{U}$ akzeptiert wird, liegt $\langle\mathcal{U}\rangle$ in L_d. Nach Definition von L_d muss $\mathcal{U}$ dann aber $\langle\mathcal{U}\rangle$ ablehnen. Wenn $\mathcal{U}$ aber $\langle\mathcal{U}\rangle$ ablehnt, dann liegt $\langle\mathcal{U}\rangle$ nicht in L_d, muss also von $\mathcal{U}$ akzeptiert werden. In beiden Fällen ergibt sich ein Widerspruch.

Als Begründung soll hier ohne weiteren Beweis ausreichen, dass die Menge der unendlichen Automaten nicht abzählbar ist.

Problem 2: Abschlusseigenschaften von regulären Sprachen 4 + 1 + 4 = 9 Punkte

Gegeben seien zwei endliche Alphabete Σ, Σ', eine reguläre Sprache L und eine surjektive Abbildung $f\colon \Sigma \to \Sigma'$. Wir erweitern f auf naheliegende Weise auf ganze Wörter: $f(w_1 \cdot w_2 \cdot \ \dots \ \cdot w_n) := f(w_1) \cdot f(w_2) \cdot \ \dots \ \cdot f(w_n)$. Damit definieren wir die Sprache $f(L)$ über Σ' als $f(L) := \{f(w) \mid w \in L\}$.

Hinweis: Surjektiv bedeutet, dass es für jedes Symbol $a \in \Sigma'$ ein $b \in \Sigma$ mit $f(b) = a$ gibt.

(a) Zeigen Sie, dass $f(L)$ regulär ist. Sei dazu ein DEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ gegeben, der L erkennt. Konstruieren Sie einen NEA $\mathcal{A}' = (Q, \Sigma', \delta', s, F)$, der $f(L)$ erkennt, d.h. ändern Sie nur das Terminalalphabet und die Überführungsfunktion. Zeigen Sie, dass $\mathcal{A}'$ tatsächlich $f(L)$ erkennt.

(b) Welche Bedingungen müssen für f gelten, damit $\mathcal{A}'$ deterministisch ist?

(c) Betrachten Sie nun die Abbildung $g\colon \Sigma \to \Sigma'^\star$ und analog die Sprache $g(L)$. Symbole aus Σ werden nun also nicht mehr auf einzelne Symbole aus Σ', sondern auf ganze Wörter abgebildet. Zeigen Sie, dass auch $g(L)$ regulär ist. Wie muss Ihre Konstruktion aus Aufgabenteil (a) angepasst werden? Diesmal dürfen Sie auch zusätzliche Zustände hinzufügen.

Lösung:

(a) Der NEA $\mathcal{A}' = (Q, \Sigma', \delta', s, F)$ mit $\delta'(q, a) = \{\delta(q, x) \mid x \in \Sigma \wedge f(x) = a\}$ erkennt $f(L)$. Wir beweisen dies, indem wir für ein beliebiges Wort $w \in \Sigma'^\star$ zeigen: $w \in f(L) \Leftrightarrow \delta'(s, w) \cap F \neq \emptyset$.

$w \in f(L) \Leftrightarrow \exists u \in L\colon w = f(u) \Leftrightarrow \exists u \in \Sigma^\star\colon w = f(u) \wedge \delta(s, u) \in F \Leftrightarrow \delta'(s, w) \cap F \neq \emptyset$

(b) $\mathcal{A}'$ ist genau dann deterministisch, wenn $|\delta'(q, a)| = 1$ für alle $a \in \Sigma'$ gilt. Das ist wiederum genau der Fall, wenn $|\{x \in \Sigma \mid f(x) = a\}| = 1$ gilt, also wenn f zusätzlich injektiv (also bijektiv) ist.

(c) Für jeden Zustand $q \in Q$ und jedes Symbol $a \in \Sigma$ machen wir Folgendes: Falls $|g(a)| \leq 1$, fügen wir wie in Aufgabenteil a) einen $g(a)$-Übergang von q nach $\delta(q, a)$ zu $\mathcal{A}'$ hinzu. Falls $|g(a)| = w_1 \dots w_n$ mit $n > 1$, ersetzen wir den einzelnen Übergang in $\mathcal{A}$ durch eine Kette von $|g(a)|$ Übergängen in $\mathcal{A}'$. Dafür führen wir $n - 1$ zusätzliche Zustände $q_{a,1}, \dots, q_{a,n-1}$ ein und setzen $q_{a,0} = q$ und $q_{a,n} = \delta(q, a)$. Für $0 \leq i < n$ fügen wir dann einen w_{i+1}-Übergang von $q_{a,i}$ nach $q_{a,i+1}$ hinzu.

Problem 3: Turingmaschinen 6 Punkte

Für $k \in \mathbb{N}_+$ ist die Komplexitätsklasse $\mathsf{NTAPE}(kn)$ definiert als die Menge der Sprachen, die von einer nichtdeterministischen Turingmaschine mit Platzbedarf höchstens kn entschieden werden können.

Zeigen Sie, dass für feste $k \in \mathbb{N}_+$ gilt, dass $\mathsf{NTAPE}(kn) = \mathsf{NTAPE}(n)$.

Lösung: $\mathsf{NTAPE}(n) \subseteq \mathsf{NTAPE}(kn)$ ist trivialerweise erfüllt.

$\mathsf{NTAPE}(kn) \subseteq \mathsf{NTAPE}(n)$: Sei $\mathcal{M}$ eine Turingmaschine mit Platzbedarf höchstens kn. Konstruiere eine Turingmaschine $\mathcal{M}'$ mit Platzbedarf höchstens n und $L(\mathcal{M}) = L(\mathcal{M}')$. Komprimiere jeweils k Bandzeichen von $\mathcal{M}$ in ein Zeichen von $\mathcal{M}'$. Zusätzlich muss in einem Zeichen von $\mathcal{M}'$ kodiert werden, auf welchem der k Zeichen der Kopf von $\mathcal{M}$ steht. Das Bandalphabet von $\mathcal{M}'$ enthält also Zeichen der Form $(Q \times \{\text{Kopf}, \text{kein Kopf}\})^k$.

Die Turingmaschine $\mathcal{M}'$ arbeitet folgendermaßen. Zunächst wird die Eingabe der Länge n mit dem eben beschriebenen Verfahren auf eine Länge von $\frac{n}{k}$ komprimiert. Dann wird $\mathcal{M}$ auf dieser komprimierten Eingabe simuliert. Bewegt sich der Kopf von $\mathcal{M}$ nur innerhalb eines Bereiches des Bandes, der zu einem Symbol komprimiert wurde, so bewegt sich der Kopf von $\mathcal{M}'$ nicht und es wird nur das aktuelle Bandsymbol entsprechend angepasst. Andernfalls bewegt sich auch der Kopf von $\mathcal{M}$. Dadurch entscheidet $\mathcal{M}$ dieselbe Sprache wie $\mathcal{M}'$. Da $\mathcal{M}$ einen Platzbedarf von höchstens kn hat und $\mathcal{M}'$ auf einem komprimierten Band arbeitet, ergibt sich ein maximaler Platzbedarf von $kn/k = n$.

Problem 4: Entscheidbarkeit $2 + 2 + 2 + 3 = 9$ Punkte

In dieser Aufgabe geht es um einige kleine Erkenntnisse bezüglich Entscheidbarkeit.

(a) Nur Sprachen können (semi-)entscheidbar sein; es gibt nicht so etwas wie ein „unentscheidbares Wort“. Sei Σ ein endliches Alphabet. Zeigen Sie, dass es kein Wort $w \in \Sigma^\star$ gibt, so dass jede Sprache, die w enthält, nicht entscheidbar ist.

(b) Zeigen Sie, dass zu jeder Sprache L Turingmaschinen $\mathcal{M}^+, \mathcal{M}^-$ existieren, die auf jeder Eingabe halten, so dass $\mathcal{M}^+$ jedes Wort in L akzeptiert und $\mathcal{M}^-$ jedes Wort, das nicht zu L gehört, ablehnt.

(c) Dass eine Turingmaschine $\mathcal{M}$ die von ihr akzeptierte Sprache $L(\mathcal{M})$ nicht entscheidet, bedeutet nicht, dass $L(\mathcal{M})$ nicht entscheidbar ist. Beschreiben Sie eine Turingmaschine $\mathcal{M}$, so dass $L(\mathcal{M})$ eine entscheidbare Sprache ist, $\mathcal{M}$ aber $L(\mathcal{M})$ nicht entscheidet.

(d) Sei L eine semi-entscheidbare Sprache. Vom 3. Übungsblatt wissen Sie, dass dann eine Turingmaschine $\mathcal{M}$ existiert, die die Wörter von L aufzählt, d.h. nacheinander und eindeutig voneinander getrennt auf das Band schreibt. Bezeichne mit $w_1, w_2, \ldots$ die Wörter in der Reihenfolge, in der sie auf das Band geschrieben werden. Definiere K als die Sprache, die w_1 enthält und jedes w_i, für das alle Wörter w_j mit $j < i$ echt kürzer als w_i sind. Zeigen Sie, dass K entscheidbar ist.

Anmerkung: Damit haben Sie gezeigt, dass jede unendliche semi-entscheidbare Sprache eine unendliche entscheidbare Teilmenge hat.

Lösung:

(a) Die Sprachen $\{w\}$ und $\Sigma^\star$ sind sogar regulär.

(b) Wähle $\mathcal{M}^+$ als die Turingmaschine, die jede Eingabe akzeptiert, und $\mathcal{M}^-$ als die Turingmaschine, die jede Eingabe ablehnt.

(c) Sei L eine beliebige entscheidbare Sprache und $\mathcal{M}'$ eine Turingmaschine, die L entscheidet. Konstruiere nun folgendermaßen $\mathcal{M}$ aus $\mathcal{M}'$. Die Turingmaschine $\mathcal{M}'$ lehnt eine Eingabe ab, indem sie in einem nicht-akzeptierenden Zustand das aktuelle Bandsymbol nicht verändert und den Kopf nicht bewegt. Die Turingmaschine $\mathcal{M}$ soll aus solchen Zuständen stattdessen in eine Endlosschleife laufen, wodurch die entsprechende Eingabe nicht mehr abgelehnt wird. Die Turingmaschine $\mathcal{M}$ akzeptiert also nach wie vor alle Wörter der entscheidbaren Sprache L, entscheidet L aber nicht, da sie nicht immer terminiert.

(d) Falls K endlich ist, ist K trivialerweise entscheidbar. Sei K also unendlich. Konstruiere eine Turingmaschine $\mathcal{M}'$, die K wie folgt entscheidet. Bei Eingabe von w simuliert $\mathcal{M}'$ die Turingmaschine M. Falls $w \in K$ ist, existiert ein w_i mit $w_i = w$. Dann wird $\mathcal{M}$ das Wort w_i nach endlicher Zeit auf das Band schreiben und somit wird $\mathcal{M}'$ das Wort w nach endlicher Zeit akzeptieren. Oder es ist $w \notin K$. Da es nur endlich viele Wörter gibt, die kürzer als w sind, wird $\mathcal{M}$ nach endlicher Zeit ein Wort auf das Band schreiben, das länger als w ist. Nach Konstruktion von K bezeugt dieses Wort die Tatsache $w \notin K$. Somit kann $\mathcal{M}'$ das Wort w nach endlicher Zeit ablehnen.

Problem 5: NP-Vollständigkeit 1 + 1 + 9 = 11 Punkte

Das Entscheidungsproblem KNOTENÜBERDECKUNG ist wie folgt definiert:

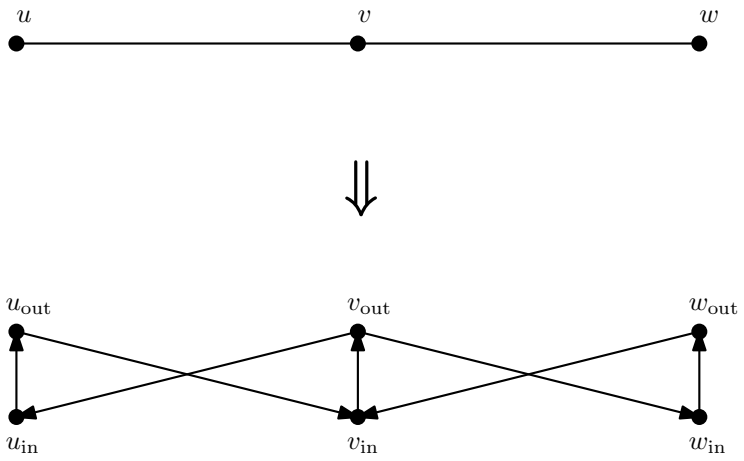
Gegeben: Ungerichteter Graph $G = (V, E)$ und Zahl $k \in \mathbb{N}$.
Frage: Gibt es eine Teilmenge $V' \subseteq V$ mit $|V'| \leq k$, sodass für alle $\{u, v\} \in E$ gilt: $u \in V'$ oder $v \in V'$?

Das Entscheidungsproblem FEEDBACK ARC SET ist wie folgt definiert:

Gegeben: Gerichteter Graph $\overrightarrow{G} = (\overrightarrow{V}, \overrightarrow{E})$ und Zahl $k \in \mathbb{N}$.
Frage: Gibt es eine Teilmenge $\overrightarrow{E}' \subseteq \overrightarrow{E}$ mit $|\overrightarrow{E}'| \leq k$, sodass $(\overrightarrow{V}, \overrightarrow{E} - \overrightarrow{E}')$ azyklisch ist?

Die NP-Vollständigkeit von FEEDBACK ARC SET kann über eine Reduktion von KNOTENÜBERDECKUNG gezeigt werden. Dabei wird die folgende Konstruktion genutzt, um den ungerichteten Graph $G = (V, E)$ aus der KNOTENÜBERDECKUNG-Instanz in einen gerichteten Graphen $\overrightarrow{G} = (\overrightarrow{V}, \overrightarrow{E})$ zu transformieren: Für jeden Knoten $v \in V$ enthält $\overrightarrow{V}$ die zwei Knoten v_{in} und v_{out}, die durch die Kante $(v_{\text{in}}, v_{\text{out}})$ verbunden sind. Jede Kante $e = \{u, v\} \in E$ wird außerdem auf die zwei Kanten $(u_{\text{out}}, v_{\text{in}})$ und $(v_{\text{out}}, u_{\text{in}})$ abgebildet.

Beispiel:



(a) Zeigen Sie: Jeder Zyklus in $\overrightarrow{G}$, der eine Kante der Form $(u_{\text{out}}, v_{\text{in}})$ enthält, enthält außerdem die Kanten $(u_{\text{in}}, u_{\text{out}})$ und $(v_{\text{in}}, v_{\text{out}})$.

(b) Zeigen Sie: Wenn $\overrightarrow{G}$ ein Feedback Arc Set der Größe k enthält, dann existiert auch ein Feedback Arc Set mit Größe $\leq k$, das nur Kanten der Form $(v_{\text{in}}, v_{\text{out}})$ enthält.

(c) Benutzen Sie die angegebene Transformation und die Aussagen aus den vorigen Aufgabenteilen, um zu zeigen, dass FEEDBACK ARC SET NP-vollständig ist.

Lösung:

(a) Die einzige eingehende Kante von u_{out} ist $(u_{\text{in}}, u_{\text{out}})$, also muss sie im Zyklus enthalten sein. Ebenso ist $(v_{\text{in}}, v_{\text{out}})$ die einzige ausgehende Kanten von v_{in} und muss deshalb im Zyklus enthalten sein.

(b) Sei $\overrightarrow{E}'$ ein Feedback Arc Set der Größe k, das eine Kante der Form $(u_{\text{out}}, v_{\text{in}})$ enthält. Nach Aufgabenteil a) enthält jeder Zyklus, der $(u_{\text{out}}, v_{\text{in}})$ enthält, auch $(v_{\text{in}}, v_{\text{out}})$. Also bleibt $\overrightarrow{E}'$ auch dann ein Feedback Arc Set, wenn man $(u_{\text{out}}, v_{\text{in}})$ durch $(v_{\text{in}}, v_{\text{out}})$ ersetzt.

(c) Wir zeigen zunächst, dass FEEDBACK ARC SET in NP liegt. Gegeben ein gerichteter Graph $\overrightarrow{G} = (\overrightarrow{V}, \overrightarrow{E})$ und eine Kantenmenge $\overrightarrow{E}' \subseteq \overrightarrow{E}$, kann man in Linearzeit überprüfen, ob $(\overrightarrow{V}, \overrightarrow{E} - \overrightarrow{E}')$ azyklisch ist, indem man eine topologische Sortierung durchführt.

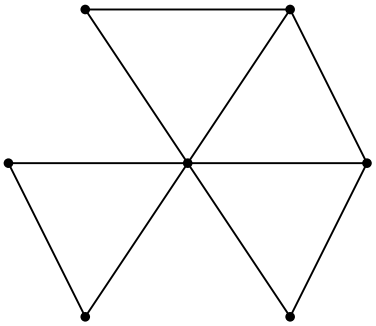
Wir zeigen, dass FEEDBACK ARC SET NP-schwer ist, indem wir von KNOTENÜBERDECKUNG auf FEEDBACK ARC SET reduzieren. Sei eine KNOTENÜBERDECKUNG-Instanz mit $G = (V, E)$ und $k \in \mathbb{N}$ gegeben. Wir konstruieren einen gerichteten Graphen $\overrightarrow{G} = (\overrightarrow{V}, \overrightarrow{E})$ wie in der Aufgabenstellung beschrieben. Dies geht offensichtlich in polynomieller Zeit, da $\overrightarrow{G}$ $2|V|$ Knoten und $2|E|+|V|$ Kanten hat. Zusammen mit demselben k bildet $\overrightarrow{G}$ die FEEDBACK ARC SET-Instanz.

Wir zeigen: Die KNOTENÜBERDECKUNG-Instanz ist genau dann lösbar, wenn die FEEDBACK ARC SET-Instanz lösbar ist.

- $\Rightarrow$: Sei $V' \subseteq V$ mit $|V'| \leq k$ eine Knotenüberdeckung. Wir wählen $\overrightarrow{E}' = \{(v_{\text{in}}, v_{\text{out}}) \mid v \in V'\}$ als Feedback Arc Set. Offensichtlich gilt $|\overrightarrow{E}'| \leq k$. Jeder Zyklus in $\overrightarrow{G}$ muss eine Kante der Form $(u_{\text{out}}, v_{\text{in}})$ enthalten. Nach Aufgabenteil a) enthält der Zyklus dann auch $(u_{\text{in}}, u_{\text{out}})$ und $(v_{\text{in}}, v_{\text{out}})$. Da G die Kante $\{u, v\}$ enthält, muss die Knotenüberdeckung V' entweder u oder v enthalten. Dann ist entweder $(u_{\text{in}}, u_{\text{out}})$ oder $(v_{\text{in}}, v_{\text{out}})$ in $\overrightarrow{E}'$ enthalten und der Zyklus ist in $(\overrightarrow{V}, \overrightarrow{E} - \overrightarrow{E}')$ nicht mehr vorhanden.
- $\Leftarrow$: Sei $\overrightarrow{E}' \subseteq \overrightarrow{E}$ mit $|\overrightarrow{E}'| \leq k$ ein Feedback Arc Set. Nach Aufgabenteil b) können wir o.B.d.A. annehmen, dass $\overrightarrow{E}'$ nur Kanten der Form $(v_{\text{in}}, v_{\text{out}})$ enthält. Dann wählen wir als Knotenüberdeckung $V' = \{v \in V \mid (v_{\text{in}}, v_{\text{out}}) \in \overrightarrow{E}'\}$. Offensichtlich gilt $|V'| \leq k$. Angenommen, V' wäre keine Knotenüberdeckung. Dann gäbe es eine Kante $\{u, v\} \in E$, sodass V' weder u noch v enthält. Dann sind $(u_{\text{in}}, u_{\text{out}})$ und $(v_{\text{in}}, v_{\text{out}})$ nicht in E'. Nach Annahme sind $(u_{\text{out}}, v_{\text{in}})$ und $(v_{\text{out}}, u_{\text{in}})$ auch nicht in E', also enthält $(\overrightarrow{V}, \overrightarrow{E} - \overrightarrow{E}')$ einen Zyklus.

Problem 6: Approximationsalgorithmen $5 + 1 + 2 = 8$ Punkte

Ein ungerichteter Graph $G = (V, E)$ lässt sich so in die Ebene zeichnen, dass jeder Knoten durch einen Punkt repräsentiert wird und jede Kante durch ein Geradensegment, das ihre beiden Endpunkte verbindet. Wir nennen G *planar*, wenn es eine solche Zeichnung gibt, in der sich keine Kanten kreuzen. Jeder Teilgraph eines planaren Graphen ist selbst wieder planar. Jeder planare Graph enthält einen Knoten mit höchstens fünf benachbarten Knoten.



Ein planarer Graph mit entsprechender Zeichnung in der Ebene.

Beim Minimierungsproblem COLOR ist ein ungerichteter Graph $G = (V, E)$ mit $V \neq \emptyset$ gegeben. Ziel ist es, G mit möglichst wenigen Farben zu färben, so dass benachbarte Knoten nicht dieselbe Farbe erhalten. Formal gilt es, eine Menge C und eine Abbildung $f : V \to C$ mit $f(u) \neq f(v)$ für alle $\{u, v\} \in E$ zu finden, und dabei $|C|$ zu minimieren.

(a) Geben Sie einen Algorithmus für COLOR für planare Graphen an, der konstant viele Farben benutzt.

(b) Beweisen Sie, dass Ihr Algorithmus aus Teilaufgabe (a) in polynomieller Zeit läuft.

(c) Ihr Algorithmus aus Teilaufgabe (a) kann als Approximationsalgorithmus aufgefasst werden. Welche absolute Gütegarantie hat Ihr Algorithmus? Begründen Sie.

Lösung:

(a) Jeder planare Graph lässt sich wie folgt mit sechs Farben färben. Färbe einen Graphen, der nur einen Knoten enthält, indem man dem Knoten eine beliebige Farbe zuweist. Färbe einen beliebigen planar Graphen G, indem man einen Knoten v mit höchstens fünf benachbarten Knoten aus G entfernt, diesen Graph rekursiv mit sechs Farben färbt und dann den Knoten v wieder einfügt und ihm eine Farbe zuweist, die keinem seiner Nachbarn zugewiesen wurde. Weil v höchstens fünf Nachbarn hat, existiert eine solche Farbe auf jeden Fall.

(b) Innerhalb linearer Zeit lässt sich ein Knoten v mit höchstens fünf Nachbarn finden. Rekursiv ergibt sich dadurch quadratische Laufzeit.

(c) Jeder Graph benötigt mindestens eine Farbe, um gefärbt zu werden. Dadurch ergibt sich eine absolute Gütegarantie von 5.

Problem 7: Kellerautomaten $1 + 3 + 1 + 3 = 8$ Punkte

(a) Betrachten Sie die Grammatik $G = (\{\mathtt{0}, \mathtt{1}\}, \{S\}, S, R)$ mit

$$R = \{S \to \mathtt{1}S\mathtt{0} \mid \mathtt{0}S\mathtt{1} \mid \varepsilon\} \ .$$

Geben Sie die Sprache $L(G)$ in Mengenschreibweise an.

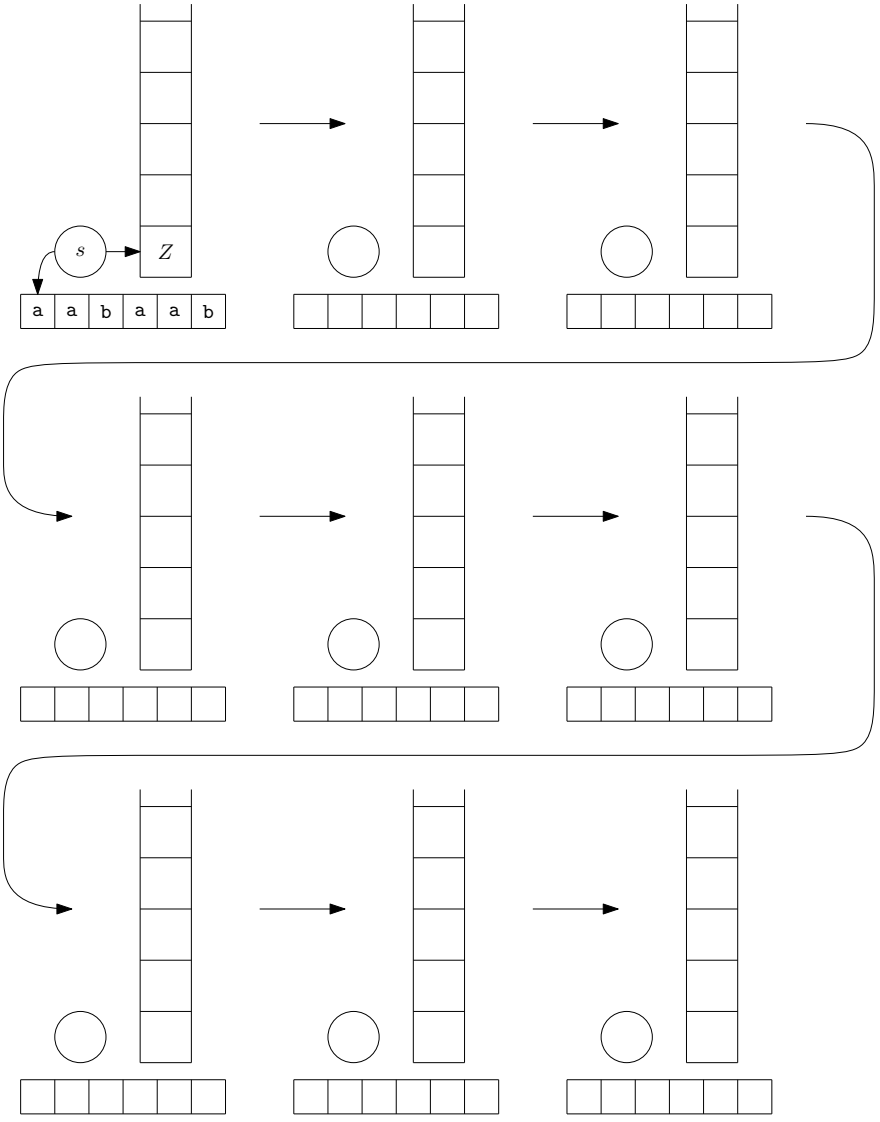
(b) Sei $\Sigma = \{\mathtt{0}, \mathtt{1}\}$ und $L = \{w \in \Sigma^\star \setminus \{\varepsilon\} \mid |w|_{\mathtt{0}} = |w|_{\mathtt{1}}\}$. Geben Sie eine Grammatik G in Greibach-Normalform an, so dass $L(G) = L$ gilt.

Sei $\mathcal{A} = (\{s, q\}, \{\mathtt{a}, \mathtt{b}\}, \{Y, Z\}, \delta, s, Z, \emptyset)$ der Kellerautomat mit der folgenden Übergangsrelation δ:

$$\begin{array}{llll} (s,\mathtt{a},Z) & \mapsto & (s,YZ), & (s,\varepsilon,Z) \mapsto (s,\varepsilon) \\ (s,\mathtt{a},Y) & \mapsto & (s,YY), & (q,\mathtt{a},Y) \mapsto (q,\varepsilon) \\ (s,\mathtt{b},Y) & \mapsto & (q,\ Y\ ), & (q,\mathtt{b},Z) \mapsto (s,Z) \end{array}$$

(c) Ist $\mathcal{A}$ deterministisch?

(d) Dokumentieren Sie eine akzeptierende Berechnung des Wortes `aabaab`. Geben Sie dazu für jeden Schritt die Zustandsübergänge und den Zustand des Stacks an. Der Automat akzeptiert durch leeren Stack.



Lösung:

(a) $L(G) = \{w \in \{\mathtt{0}, \mathtt{1}\}^\star \mid w$ hat gerade Länge und für $i = 1, 2, \ldots, |w|$ gilt $w_i \neq w_{|w|-i+1}\}$

(b)

$$S \to \mathtt{0}A \mid \mathtt{0}AS \mid \mathtt{1}B \mid \mathtt{1}BS$$
$$A \to \mathtt{0}AA \mid \mathtt{1}$$
$$B \to \mathtt{1}BB \mid \mathtt{0}$$

(c) Nein, denn es gilt $|\delta(s, \mathtt{a}, Z)| + |\delta(s, \varepsilon, Z)| > 1$.

(d) s, aabaab, Z

s, abaab, YZ

s, baab, YYZ

q, aab, YYZ

q, ab, YZ

q, b, Z

s, ε, Z

s, ε, ε (akzeptiert durch leeren Stack)