Grundbegriffe der Informatik
Wintersemester 2025/26 – Klausur 1

Hier Aufkleber mit Matrikelnummer anbringen

LÖSUNG!

- Bringen Sie den Aufkleber mit Ihrer Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihrer Matrikelnummer.
- Die Klausur ist **doppelseitig** gedruckt, inklusive der Rückseite des Titelblatts.
- Am Ende der Klausur sind zusätzliche Leerseiten. Wenn Sie diese nutzen, verweisen Sie in der entsprechenden Aufgabe darauf. Fordern Sie zusätzliches Papier bitte nur an, falls Sie den gesamten Platz aufgebraucht haben.
- Verwenden Sie nur dokumentenechte Stifte in blau oder schwarz.
- Die Tackernadel darf nicht gelöst werden.
- Bearbeitungszeit: 2 Stunden

Nötige Punkte	**Note**
44	1.0
41	1.3
38	1.7
35	2.0
32	2.3
29	2.7
26	3.0
23	3.3
20	3.7
17	4.0
0	5.0 = nicht bestanden

Aufgabe 1

$1 + 1 + 1 + 3 + 4 = 10$ Punkte

(a) Sei w ein Wort. Geben Sie eine induktive Definition von w^n für $n \in \mathbb{N}_0$ an.

(b) Geben Sie einen regulären Ausdruck für die folgende Sprache L an.

$$L = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w \text{ enthält } \mathtt{ab} \textbf{ nicht} \text{ als Teilwort}\}$$

(c) Sei $M = \{1, 2, 3\}$. Geben Sie $M' = 2^M \setminus \binom{M}{2}$ an, indem Sie die Elemente aufzählen.

$$M' = \left\{\boxed{}\right\}$$

(d) Sei $A = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}\}$ ein Alphabet und $h\colon A \to \{\mathtt{0}, \mathtt{1}\}^*$ eine Funktion mit:

$$h(\mathtt{a}) = \mathtt{00} \quad h(\mathtt{b}) = \mathtt{01} \quad h(\mathtt{c}) = \mathtt{10} \quad h(\mathtt{d}) = \mathtt{110} \quad h(\mathtt{e}) = \mathtt{111}$$

Sei h^{**} der von h induzierte Homomorphismus und $L = \{w, x, y, z\} \subseteq A^*$ mit:

$$w = \mathtt{a}^4\mathtt{b}^3\mathtt{cde} \quad x = \mathtt{a}^3\mathtt{b}^3\mathtt{c}^2\mathtt{de} \quad y = \mathtt{a}^2\mathtt{b}^2\mathtt{cde} \quad z = \mathtt{a}^3\mathtt{bcde}$$

Geben Sie $L_H = \{w' \in L \mid h^{**} \text{ ist eine Huffman-Kodierung von } w'\}$ an.

$$L_H = \left\{\boxed{}\right\}$$

(e) Sei $A = \{\mathtt{a}, \mathtt{b}\}$ ein Alphabet. Zeigen Sie, dass die folgenden zwei Relationen R und R' **keine** surjektiven Funktionen sind.

- $R = \{(w, L) \mid w \in A^*, L = \{w^2\}\} \ \subseteq \ A^* \times (2^{A^*} \setminus \{\emptyset\})$
- $R' = \{(w, L) \mid w \in A^*, \emptyset \neq L \subseteq A^*, w \in L^2\} \ \subseteq \ A^* \times (2^{A^*} \setminus \{\emptyset\})$

Lösung

(a) $w^0 = \varepsilon$ und $w^n = w^{n-1} \cdot w$ für $n \in \mathbb{N}_+$

(b) $\mathtt{b}^*\mathtt{a}^*$

(c) $M = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2, 3\}\}$

(d) $L_H = \{x, y\}$

(e)
- R nicht surjektiv, da für jedes $(w, L) \in R$ gilt, dass L nur ein Wort enthält. Also gibt es für die Sprache $L' = \{\mathtt{a}, \mathtt{b}\}$ kein Wort $w' \in A^*$, sodass $(w', L') \in R$.
- R' ist zwar rechtstotal und linkstotal, aber nicht rechtseindeutig und deshalb keine Funktion. Um das zu zeigen, betrachten wir $w = \mathtt{a}^2$, $L_1 = \{\mathtt{a}\}$ und $L_2 = \{\mathtt{a}, \mathtt{b}\}$. Dann ist $L_1^2 = \{\mathtt{a}^2\}$, also $w \in L_1^2$ und daher $(w, L_1) \in R$. Genauso ist $L_2^2 = \{\mathtt{a}^2, \mathtt{ab}, \mathtt{ba}, \mathtt{b}^2\}$, also $w \in L_2^2$ und daher $(w, L_2) \in R$.
  Anmerkung: R ist linkstotal, da für jedes $w \in A^*$ für die Sprache $L = \{w, \varepsilon\}$ gilt, dass $w \in \{w^2, w, \varepsilon\} = L^2$, also $(w, L) \in R$. Außerdem ist R rechtstotal, denn es gibt für jede nicht-leere Sprache L ein Wort in $w \in L^2$, für das dann $(w, L) \in R$ gilt.

Aufgabe 2 **3 + 2 + 2 + 3 = 10 Punkte**

(a) Seien P, Q und R aussagenlogische Variablen. Gegeben sei außerdem die aussagenlogische Formel Z mit

$$Z = \neg P \Leftrightarrow (Q \vee R)$$

und eine Menge $M = \{A, B, C, D, E, F\}$ aussagenlogischer Formeln mit

$$\begin{aligned}
A &= \neg P \wedge (Q \vee R)\\
B &= P \Leftrightarrow (\neg Q \wedge \neg R)\\
C &= (\neg P \Leftrightarrow Q) \vee (\neg P \Leftrightarrow R)\\
D &= (\neg P \wedge P) \vee (\neg P \Leftrightarrow (Q \vee R))\\
E &= (\neg P \Rightarrow (Q \vee R)) \wedge ((Q \vee R) \Rightarrow \neg P)\\
F &= (\neg P \wedge Q) \vee (\neg P \wedge R) \vee (P \wedge \neg Q \wedge \neg R)
\end{aligned}$$

Geben Sie $M' = \{X \in M \mid X \equiv Z\}$ an, indem Sie die enthaltenen Formeln nennen.

$$M' = \left\{\boxed{}\right\}$$

(b) Sei $n \in \mathbb{N}_+$ und seien P_i und Q_i aussagenlogische Variablen für alle $i \in [n]$. Geben Sie die Anzahl der Modelle der folgenden Formel in Abhängigkeit von n an.

$$\bigwedge_{i=1}^{n} (P_i \Rightarrow Q_i)$$

(c) Seien P, Q aussagenlogische Variablen. Wir definieren ein neues Konnektiv $P \not\Leftrightarrow Q$ durch

$$P \not\Leftrightarrow Q \equiv \neg(P \Leftrightarrow Q).$$

Geben Sie eine unerfüllbare Formel an, die nur aus $P, Q, \not\Leftrightarrow, \wedge, \vee$ sowie Klammern besteht (insb. **keine** Negationen, Implikationen). Sie dürfen Zeichen mehrfach verwenden, aber insgesamt **höchstens zwei Konnektive**. Setzen Sie alle Klammern.

(d) Seien M_1, M_2, M_3 Mengen. Wir betrachten die aussagenlogischen Variablen E_1, E_2, E_3, $S_{12}, S_{13}, S_{21}, S_{23}, S_{31}, S_{32}$ mit einer Interpretation I, für die Folgendes gilt:

- $I(E_i) = \mathbf{w}$ genau dann, wenn M_i endlich ist für $i \in [3]$
- $I(S_{ij}) = \mathbf{w}$ genau dann, wenn $M_i \subseteq M_j$ für $i, j \in [3]$ mit $i \neq j$

Formalisieren Sie die folgenden Aussagen als aussagenlogische Formeln bezüglich I.

(i) Wenn eine der drei Mengen M_1, M_2, M_3 unendlich ist, dann sind alle drei unendlich.

(ii) Genau eine der beiden Mengen M_1, M_2 ist endlich.

(iii) $M_1 = M_2 \neq M_3$.

Lösung

(a) $M' = \{B, D, E, F\}$

(b) Für jedes i muss $P_i \Rightarrow Q_i$ zu wahr auswerten, dafür gibt es jeweils drei Möglichkeiten, nämlich alle außer $I(P_i) = \mathbf{w}, I(Q_i) = \mathbf{f}$. Die Wahlen für jedes i werden unabhängig getroffen, also gibt es insgesamt 3^n Modelle.

(c) $P \not\Leftrightarrow P$

(d) (i) $(\neg E_1 \vee \neg E_2 \vee \neg E_3) \Rightarrow (\neg E_1 \wedge \neg E_2 \wedge \neg E_3)$

(ii) $(E_1 \vee E_2) \wedge \neg(E_1 \wedge E_2)$. Alternativ z.B. $\neg(E_1 \Leftrightarrow E_2)$

(iii) $(S_{12} \wedge S_{21}) \wedge \neg(S_{23} \wedge S_{32})$

Aufgabe 3

$7 + 1 + 2 = 10$ Punkte

Sei A ein Alphabet mit $|A| \geq 2$, sei $f\colon A \to A^*$ eine Funktion und sei f^{**} der durch f induzierte Homomorphismus.
Hinweis: Für eine Funktion $\varphi\colon A \to A^*$ definieren wir $\varphi(A) = \{\varphi(x) \mid x \in A\}$. Sie dürfen die Notation $\prod_{i=1}^{n} w_i$ für die Konkatenation mehrerer Wörter w_i, $i \in [n]$, $n \in \mathbb{N}_+$ verwenden.

(a) Zeigen Sie, dass f^{**} genau dann surjektiv ist, wenn $f(A) = A$.

Seien außerdem $g, h\colon A \to A^*$ zwei Funktionen, sodass $g(x) \in A$ und $f(x) = g(x) \cdot h(x)$ für alle $x \in A$.

(b) Geben Sie eine Wahl für das Alphabet A und die Funktion $h\colon A \to A^*$ an, sodass $|f^{**}(w)| \neq |w|$ für alle Wörter $w \in A^+$.

(c) Sei nun zusätzlich g injektiv und $A = \{\mathtt{a}, \mathtt{b}\}$. Geben Sie alle Funktionen $h\colon A \to A^*$ an, sodass f^{**} surjektiv ist.

Lösung

(a) „$\Rightarrow$“ Da f^{**} surjektiv ist, gibt es für jedes Wort aus A^*, also insbesondere für jedes $x \in A$ ein $w \in A^*$, sodass $f^{**}(w) = x$. Da f^{**} ein Homorphismus ist, gilt $\prod_{i=1}^{|w|} f^{**}(w(i)) = f^{**}(w) = x \in A$ und es folgt, dass für genau ein i gilt $f(w(i)) = x$, während alle anderen auf ε abbilden. Damit haben wir $A \subseteq f(A)$. Da $f(A)$ höchstens $|A|$ Elemente enthält (und $|A|$ endlich ist), gilt auch Gleichheit.

„$\Leftarrow$“ Sei $w \in A^*$. Wir müssen zeigen, dass es ein $v \in A^*$ mit $f^{**}(v) = w$ gibt. Da $A \subseteq f(A)$, gibt es für jedes $y \in A$ ein $x \in A$, sodass $f(x) = y$. Insbesondere gibt es also für jedes $i \in [|w|]$ ein $v_i \in A$ mit $f(v_i) = w(i)$. Mit $v = \prod_{i=1}^{|w|} v_i$ folgt $f^{**}(v) = \prod_{i=1}^{|w|} f(v_i) = \prod_{i=1}^{|w|} w(i) = w$.

(b) Jedes ε-freie h ist eine korrekte Lösung, z.B. $h(x) = x$ für alle $x \in A$ und ein beliebiges Alphabet, z.B. $A = \{\mathtt{a}, \mathtt{b}\}$

(c) Wir zeigen, dass f^{**} genau dann surjektiv ist, wenn $h(x) = \varepsilon$ für alle $x \in A$. (*Beweis nicht gefordert*)

Zuerst sehen wir ein, dass $g(A) = A$ gilt: Aus $g(x) \in A$ für alle $x \in A$ folgt schon $g(A) \subseteq A$. Es gilt auch Gleichheit, denn wäre $g(A) \subsetneq A$, so hätten wir $|g(A)| < |A|$, und da $|A|$ endlich ist, gibt es dann zwei verschiedene $x, x' \in A$ mit $g(x) = g(x')$, was der Injektivität von g widerspricht.

Wenn $h(x) = \varepsilon$ für alle $x \in A$, dann ist $f(x) = g(x)h(x) = g(x)$ für alle $x \in A$, also ist f^{**} wegen (a) surjektiv.

Wenn andersrum $h(x) \neq \varepsilon$ für ein $x \in A$, dann ist $|f(x)| = |g(x) \cdot h(x)| = |g(x)| + |h(x)| \geq 2$, also ist $f(x) \notin A$. Nach (a) ist f^{**} nicht surjektiv.

Aufgabe 4

$3 + 1 + 6 = 10$ Punkte

(a) Geben Sie einen endlichen Automaten mit höchstens 6 Zuständen graphisch an, der die Sprache $L = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w$ enthält **nicht** das Teilwort $\mathtt{bab}\}$ akzeptiert.

Sei $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ ein Alphabet. Gegeben sei die Sprache

$$L = \{w \in \Sigma^* \colon |w| \bmod 3 = 0 \text{ und für jedes } k \in [|w|/3] \text{ gilt:}$$
$$\text{genau eines der drei Zeichen } w(2k-1), w(2k), w(|w|-k+1) \text{ ist ein } \mathtt{a}\}.$$

(b) Geben Sie zwei Wörter $w, \bar{w} \in \Sigma^*$ der Länge mindestens 6 an, sodass $w \in L$ und $\bar{w} \notin L$.

$w =$

$\bar{w} =$

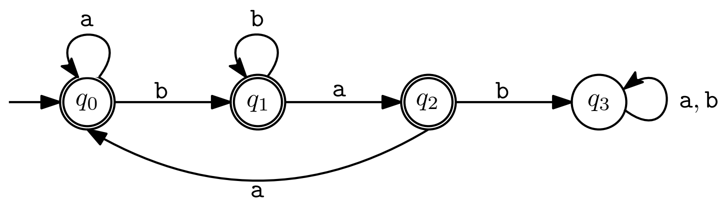
(c) Gegeben sei außerdem die kontextfreie Grammatik

$$G = (\Sigma, \{X\}, X, \{X \to \mathtt{ab}X\mathtt{b} \mid \mathtt{ba}X\mathtt{b} \mid \mathtt{bb}X\mathtt{a} \mid \varepsilon\}).$$

Zeigen Sie mittels Induktion, dass $L \subseteq L(G)$ gilt. Geben Sie dabei explizit die Induktionsvoraussetzung an und wo Sie diese verwenden.
Hinweis: Für $x, y, z \in \Sigma$ und $w \in \Sigma^*$ gilt: Wenn $xywz \in L$, dann ist auch $w \in L$.

Lösung

(a) Der folgende Automat erkennt die Sprache L:



(b) Alle möglichen Lösungen für w der Länge genau 6:
$\mathtt{ababbb}$, $\mathtt{abbabb}$, $\mathtt{abbbab}$, $\mathtt{baabbb}$, $\mathtt{bababb}$, $\mathtt{babbab}$, $\mathtt{bbabba}$, $\mathtt{bbbaba}$, $\mathtt{bbbbaa}$ (längere Lösungen sind auch möglich). Alle anderen Wörter aus Σ^* der Länge 6 sind nicht in L und somit mögliche Lösungen für $\bar{w}$ (also z.B. $\mathtt{aaaaaa}$).

(c) Wir zeigen, dass jedes Wort $w \in L$ auch in $L(G)$ ist. Dazu machen wir eine Induktion über die Wortlänge $|w| = n$.

Induktionsanfang: Sei $n \leq 2$ und $w \in L$ mit $|w| = n$. Da alle Wörter in L eine durch 3 teilbare Länge haben, gilt $|w| = 0$ und somit $w = \varepsilon \in L$. Da ε direkt vom Startsymbol aus ableitbar ist ($X \to \varepsilon$), ist ε auch in $L(G)$.

Induktionsschritt: Sei nun $n \geq 3$ und $w \in L$ ein beliebiges Wort der Länge $|w| = n$. Wir zeigen, dass $w \in L(G)$ ist und verwenden dabei die Induktionsvoraussetzung, dass jedes Wort $w' \in L$ der Länge $|w'| < n$ in $L(G)$ ist.

Da $w \in L$ ist und $|w| \geq 3$, ist mit $k = 1$ genau eines der drei Zeichen $w(2\cdot 1-1) = w(1)$, $w(2 \cdot 1) = w(2)$ und $w(|w| - 1 + 1) = w(|w|)$ ein $\mathtt{a}$. Also hat w die Form $w = \mathtt{ab}w'\mathtt{b}$, $w = \mathtt{ba}w'\mathtt{b}$ oder $w = \mathtt{bb}w'\mathtt{a}$. Außerdem folgt aus dem Hinweis, dass $w' \in L$ ist. Da $|w'| = |w| - 3 = n - 3 < n$ folgt somit nach Induktionsvoraussetzung, dass $w' \in L(G)$ und daher auch $X \xrightarrow{*} w'$. Außerdem gibt es für alle Konstellationen von $w(1), w(2)$ und $w(|w|)$ die Ableitung $X \to w(1)w(2)Xw(|w|)$. Es gilt also $X \to w(1)w(2)Xw(|w|) \xrightarrow{*} w(1)w(2)w'w(|w|) = w$. Somit ist $w \in L(G)$.

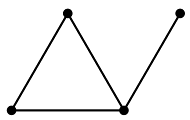
Aufgabe 5 $1 + 2 + 1 + 2 + 2 + 2 = 10$ Punkte

(a) Geben Sie eine Bedingung an, wann ein zusammenhängender Graph $G = (V, E)$ eine geschlossene Euler-Tour hat. Die Bedingung muss hinreichend und notwendig sein.

Wir sagen ein Graph G ist *d-degeneriert*, wenn jeder Teilgraph von G einen Knoten mit Grad höchstens d hat. Wir definieren die *Degeneriertheit* von G als das kleinste d, sodass G d-degeneriert ist.

(b) Sei $n \geq 3$ und $m \geq 1$. Geben Sie die Degeneriertheit der folgenden Graphen an.

- Den im folgenden abgebildeten Graphen: ☐



- Vollständiger Graph K_n: ☐
- Vollständig bipartiter Graph $K_{n,m}$: ☐
- Baum auf n Knoten: ☐

(c) Sei $n \in \mathbb{N}_+$. Geben Sie in Abhängigkeit von n die maximale Anzahl Kanten eines Graphen G mit n Knoten und Degeneriertheit 1 an.

(d) Seien $n, d \in \mathbb{N}_+$ und $2 \leq d \leq n$. Geben Sie in Abhängigkeit von n und d die maximale Anzahl Kanten eines Graphen G mit n Knoten und Degeneriertheit d an.

Für einen Graphen $G = (V, E)$ definieren wir $\Delta(G) := \max\{\deg(v) \mid v \in V\}$.

(e) Zeigen oder widerlegen Sie: Für jedes $d \in \mathbb{N}_0$ und jeden d-degenerierten Graphen G gilt $\Delta(G) \leq 2d$.
☐ Ich zeige die Aussage. ☐ Ich widerlege die Aussage.

(f) Zeigen oder widerlegen Sie: Jeder Graph G ist $\Delta(G)$-degeneriert.
☐ Ich zeige die Aussage. ☐ Ich widerlege die Aussage.

Lösung

(a) Ein zusammenhängender Graph $G = (V, E)$ hat genau dann eine geschlossene Euler-Tour, wenn $\deg(v) \bmod 2 = 0$ für alle $v \in V$.

Alternativ: Eine geschlossene Euler-Tour eines Graphen $G = (V, E)$ ist eine Folge $v_1, v_2, ..., v_k$ von Knoten, $k \in \mathbb{N}_+$, sodass für jedes $i \in [k - 1]$ gilt $v_i v_{i+1} \in E$ und für jede Kante uv gibt es genau ein $i \in [k - 1]$ mit $uv = v_i v_{i+1}$.

(b)
- 2, der Graph enthält einen K_3 als Teilgraph, der Minimalgrad 2 hat. Alle anderen Teilgraphen haben einen Knoten mit Grad höchstens 1.
- $n-1$, jeder Knoten hat Grad $n-1$.
- $\min(n,m)$, jeder Teilgraph entält einen Knoten mit Grad höchstens $\min(n,m)$. $K_{n,m}$ selbst hat keinen Knoten mit Grad kleiner als $\min(n,m)$.
- 1, jeder Teilgraph ist ein Wald und enthält damit ein Blatt.

(c) $n-1$. Es sind genau die Bäume.

(d) $d \cdot (n-d) + \binom{d}{2}$. Um das zu sehen, entfernen wir Schritt für Schritt den Knoten mit dem kleinsten Grad. Die ersten $n-d$ entfernen Knoten haben Grad höchstens d, die verbleibenden d Knoten haben kleineren Grad, nämlich höchstens so groß, wie die Zahl der übrigen Knoten. Hier haben wir maximal eine d-Clique, also $\binom{d}{2}$ Kanten.

Diese Schranke ist für alle n, d bestmöglich. Zum Bespiel wird sie erreicht durch den Graphen, der aus einem K_d besteht mit zusätzlichen $n-d$ Knoten, die jeweils zu allen d Knoten des K_d verbunden sind.

(e) Die Aussage gilt nicht. Wir betrachten $K_{1,3}$. Jeder Teilgraph von $K_{1,3}$ ist ein Baum und hat somit einen Knoten mit Grad höchstens 1. Folglich ist $K_{1,3}$ 1-degeneriert. Allerdings gilt $\Delta(K_{1,3}) = 3 > 2$.

(f) Die Aussage gilt. Sei G ein Graph und $H \subseteq G$ ein beliebiger Teilgraph. Wir betrachten einen beliebigen Knoten $v \in V(H)$. Da $\Delta(G) \geq \deg_G(v) \geq \deg_H(v)$, ist G $\Delta(G)$-degeneriert.

Aufgabe 6

$2+2+2+2+2=10$ Punkte

(a) Gegeben seien die folgenden Mengen von Funktionen:

$$A = \mathcal{O}(\log(n)) \qquad B = \mathcal{O}(2^n) \qquad C = \mathcal{O}(\log(\log(n)))$$
$$D = \mathcal{O}(\sqrt{n}) \qquad E = \mathcal{O}(1) \qquad F = \mathcal{O}(\frac{1}{\sqrt{n}}).$$

Tragen Sie in die Kästen jeweils eine der Mengen $X \in \{A, B, C, D, E, F\}$ ein, sodass eine wahre Aussage entsteht und für jede andere mögliche Wahl Y gilt, dass $X \subseteq Y$.

$$\frac{2^n}{2n^2} \in \square \qquad \log(\sqrt{n}) \in \square \qquad \frac{3n}{2\sqrt{n}} \in \square \qquad \sqrt[10]{n} \in \square$$

Gegeben sei der folgende Algorithmus $\mathcal{A}$, der als Eingabe einen zusammenhängenden Graphen $G = (V, E)$ bekommt, und eine Menge D oder $\perp$ ausgibt. Er benutzt die Funktionen IstZusammenhängend und AnzahlKanten, die als Eingabe einen Graphen G und eine Menge D an Knoten bekommen.

- IstZusammenhängend(G, D) gibt genau dann „wahr“ zurück, wenn $G - D$ zusammenhängend ist.
- AnzahlKanten(G, D) gibt $|E(G - D)|$ zurück.

$G - D$ bezeichnet dabei den Graphen, der durch das Löschen von allen Knoten in D und allen Kanten, die zu Knoten aus D inzident sind, entsteht.

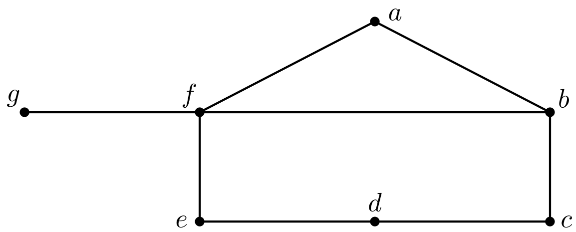
Algorithmus $\mathcal{A}$

Eingabe: Zusammenhängender Graph $G = (V, E)$

Ausgabe: Menge D oder $\perp$

```
for i = 0, ..., 5 do
    for all D ∈ (V choose i) do
        if IstZusammenhängend(G, D) then
            if AnzahlKanten(G, D) = |V| − |D| − 1 then
                return D
return ⊥
```

(b) Geben Sie die Ausgabe D von Algorithmus $\mathcal{A}$ mit dem folgenden Graphen als Eingabe an.



$D =$ $\boxed{}$

(c) Sei $G = (V, E)$ ein Baum. Geben Sie die Ausgabe von Algorithmus $\mathcal{A}$ für die Eingabe G an.

(d) Geben Sie einen zusammenhängenden Graphen mit höchstens 8 Knoten an, für den Algorithmus $\mathcal{A}$ $\perp$ zurückgibt.

(e) Die Funktionen IstZusammenhängend(G, D) hat für Graphen $G = (V, E)$ Laufzeit $\Theta(|E| + |D|)$. Die Funktionen AnzahlKanten(G, D) hat für Graphen $G = (V, E)$ Laufzeit $\Theta(|E|)$. Jeder andere Befehl (d. h. jede einzelne Zeile) benötigt Zeit $\Theta(1)$.

Geben Sie die Laufzeit von $\mathcal{A}$ mit Eingabe G in Abhängigkeit von $n = |V|$ an. Verwenden Sie hierfür die Θ-Notation und vereinfachen Sie so weit wie möglich.

Lösung

(a)

$$\frac{2^n}{2n^2} \in B \qquad \log(\sqrt{n}) \in A \qquad \frac{3n}{2\sqrt{n}} \in D \qquad \sqrt[10]{n} \in D$$

(b) $D = \{b\}$

(c) $\emptyset$. Die beiden Bedingungen in den Zeilen 3 und 4 zusammen ergeben eine Charakterisierung von Bäumen. Der Algorithmus prüft also, ob es eine Menge D von höchstens 5 Knoten gibt, sodass $G - D$ ein Baum ist und gibt ggf. eine kleinste solche Menge aus. Für Bäume muss dafür kein Knoten gelöscht werden, also gibt der Algorithmus schon im ersten Schleifendurchlauf für $i = 0$ die leere Menge aus.

(d) K_8. Beim Löschen von fünf Knoten bleibt ein Dreieck, also kein Baum.

(e) $\Theta(n^7)$