Grundbegriffe der Informatik
Wintersemester 2024/25 – Nachklausur

INF (6 ECTS)

Hier Aufkleber mit Matrikelnummer anbringen

LÖSUNG!

- Prüfen Sie, ob Sie die richtige Version der Klausur haben:
  Physik/Geo: PH/GEO (4 ECTS), alle anderen: INF (6 ECTS)
- 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

	Mögliche Punkte							Erreichte Punkte						
	a	b	c	d	e	f	Σ	a	b	c	d	e	f	Σ
Aufg. 1	4	1	1	2	2	–	10						–	
Aufg. 2	2	2	3	3	–	–	10					–	–	
Aufg. 3	3	1	6	–	–	–	10				–	–	–	
Aufg. 4	4	6	–	–	–	–	10			–	–	–	–	
Aufg. 5	2	1	2	5	–	–	10					–	–	
Aufg. 6	2	2	1	1	2	2	10							
Σ							60	Bestehensgrenze:						26

Aufgabe 1

$4 + 1 + 1 + 2 + 2 = 10$ Punkte

(a) Gegeben sei die Relation $R = \{(a, b) \in \mathbb{N}_+ \times \mathbb{N}_+ \mid b - a = 3\}$. Geben Sie an, welche der Eigenschaften *linkstotal*, *rechtstotal*, *linkseindeutig* und *rechtseindeutig* R erfüllt und welche nicht. Beweisen Sie jeweils Ihre Antwort.

☐ R ist linkstotal ☐ R ist **nicht** linkstotal
☐ R ist rechtstotal ☐ R ist **nicht** rechtstotal
☐ R ist linkseindeutig ☐ R ist **nicht** linkseindeutig
☐ R ist rechtseindeutig ☐ R ist **nicht** rechtseindeutig

(b) Gegeben sei die Funktion $f : A \to B$ mit $f(x) = x - 1$ für alle $x \in A$. Geben Sie **unendliche** Mengen A und B an, sodass f eine surjektive Funktion ist.

$A = \boxed{}$

$B = \boxed{}$

(c) Sei $\mathcal{A}_n$ eine Aussage, die abhängig von einer natürlichen Zahl $n \in \mathbb{N}_0$ ist. Es gelte $\mathcal{A}_k$ für ein $k \in \mathbb{N}_0$ und für alle $n \geq 1$ folge aus $\mathcal{A}_{n-1}$ die Aussage $\mathcal{A}_n$. Geben Sie die Menge an Zahlen an, für die $\mathcal{A}_n$ auf jeden Fall gilt.

(d) Geben Sie eine Funktion $f : \mathbb{N}_0^4 \to \mathbb{N}_0$ an, sodass die Gleichung

$$f(\mathrm{Num}_2(w), \mathrm{Num}_2(w'), |w|, |w'|) = \mathrm{Num}_2(w \cdot w')$$

für alle $w, w' \in \{\texttt{0}, \texttt{1}\}^*$ erfüllt ist. Sie dürfen dabei Num, Repr, Trans **nicht** benutzen.

$f(n, n', \ell, \ell') = \boxed{}$

(e) Gegeben sei die folgende induktive Definition für die Sprachen $L_i \subseteq \{\texttt{a}, \texttt{b}\}^*$ für alle $i \in \mathbb{N}_0$:

- $L_0 = \{\varepsilon\}$
- Für alle $i > 0$ gilt: $L_i = \{\texttt{a}\} \cdot L_{i-1} \cdot \{\texttt{b}\} \cup \{\texttt{b}\} \cdot L_{i-1} \cdot \{\texttt{a}\} \cup L_{i-1} \cdot L_{i-1}$

Sei $L = \bigcup_{i \in \mathbb{N}_0} L_i$ und $n \in \mathbb{N}_0$. Geben Sie in Abhängigkeit von n alle möglichen Tupel (x, y) an, sodass es ein Wort der Länge n in L gibt, das genau x mal $\texttt{a}$ und genau y mal $\texttt{b}$ enthält.

Lösung

(a) **Linkstotal:** Ja, sei $a \in \mathbb{N}_+$. Dann gilt $(a + 3) - a = 3$. Also ist $(a, a + 3) \in R$ für alle $a \in \mathbb{N}_+$. Somit ist R linkstotal.

Rechtstotal: Nein, es gibt kein $a \in \mathbb{N}_+$, sodass $1 - a = 3$, denn dann wäre $a = 1 - 3 = -2 \notin \mathbb{N}_+$. Also ist $(a, 1) \notin R$ für alle $a \in \mathbb{N}_+$. Somit ist R nicht rechtstotal.

Linkseindeutig Ja, seien $(a_1, b), (a_2, b) \in R$. Dann gilt $b - a_1 = b - a_2$. Also gilt $a_1 = a_2$. Somit ist R linkseindeutig.

Rechtseindeutig Ja, seien $(a, b_1), (a, b_2) \in R$. Dann gilt $b_1 - a = b_2 - a$. Also gilt $b_1 = b_2$. Somit ist R rechtseindeutig.

(b) $A = \mathbb{N}_+$ und $B = \mathbb{N}_0$ oder $A = B = \mathbb{Z}$

(c) Die Aussage gilt für alle natürlichen Zahlen größergleich k.

(d) $f(n, n', \ell, \ell') = 2^{\ell'} \cdot n + n'$

(e) L enthält nur Wörter gerader Länge. Wenn n ungerade ist, gibt es also keinen solchen Tupel. Alle Wörter in L enthalten gleich viele `a` und `b`. Wenn n gerade ist, ist also der einzige mögliche Tupel $(\frac{n}{2}, \frac{n}{2})$.

Aufgabe 2

$2 + 2 + 3 + 3 = 10$ Punkte

(a) Seien A, B und C aussagenlogische Variablen. Geben Sie eine Interpretation I an, die ein Modell für die folgende aussagenlogische Formel ist.

$$((A \lor \neg B) \land (\neg A \lor C)) \leftrightarrow \neg((B \land C) \lor \neg A)$$

$I(A) = \boxed{}$ $\quad I(B) = \boxed{}$ $\quad I(C) = \boxed{}$

(b) Seien P, Q und R aussagenlogische Variablen. Zeigen Sie durch logische Umformung, dass die folgenden beiden Formeln äquivalent sind.

$$P \to (Q \to R)$$

$$(P \land Q) \to R$$

(c) Zeigen Sie, dass die Implikation „$\to$" weder kommutativ noch assoziativ ist.
Hinweis: Ein logischer Operator $*\colon \mathbb{B}^2 \to \mathbb{B}$ *ist* kommutativ, *wenn für alle* $x, y \in \mathbb{B}$ *gilt, dass* $x * y = y * x$.
Hinweis: Ein logischer Operator $*\colon \mathbb{B}^2 \to \mathbb{B}$ *ist* assoziativ, *wenn für alle* $x, y, z \in \mathbb{B}$ *gilt, dass* $x * (y * z) = (x * y) * z$.

(d) Gegeben sei die folgende prädikatenlogische Signatur $\mathcal{S} = (\mathrm{Var}_{\mathrm{PL}}, \mathrm{Fun}_{\mathrm{PL}}, \mathrm{Rel}_{\mathrm{PL}})$ mit Stelligkeitsfunktion ar und

$$\mathrm{Var}_{\mathrm{PL}} = \{x, y, z\}, \quad \mathrm{Fun}_{\mathrm{PL}} = \emptyset, \quad \mathrm{Rel}_{\mathrm{PL}} = \{\mathtt{W}, \mathtt{N}\}, \quad \mathrm{ar}(\mathtt{W}) = 2, \quad \mathrm{ar}(\mathtt{N}) = 1.$$

Wir betrachten die Interpretation (D, I), für die Folgendes gilt:

- $D = \mathbb{R}$
- $(x, y) \in I(\mathtt{W})$ genau dann, wenn y die Wurzel von x ist.
- $(x) \in I(\mathtt{N})$ genau dann, wenn x negativ ist.

Geben Sie jeweils eine Formel der Prädikatenlogik über $\mathcal{S}$ mit Gleichheit an, die folgende Sachverhalte darstellt.

(i) Jede nicht-negative reelle Zahl hat genau eine reelle Wurzel.
(ii) Wenn eine reelle Zahl keine Wurzel hat, ist sie negativ.
(iii) Es gibt eine reelle Zahl, die ihre eigene Wurzel ist.

Lösung

(a) Die Interpretationen

$$I_1(A) = \mathbf{f}, \qquad I_1(B) = \mathbf{w}, \qquad I_1(C) = \mathbf{f}$$
$$I_2(A) = \mathbf{f}, \qquad I_2(B) = \mathbf{w}, \qquad I_2(C) = \mathbf{w}$$
$$I_3(A) = \mathbf{w}, \qquad I_3(B) = \mathbf{f}, \qquad I_3(C) = \mathbf{w}$$

sind (die einzigen) Modelle der Formel.

(b)

$$\begin{aligned}
& P \to (Q \to R) \\
\equiv\ & \neg P \vee (Q \to R) \\
\equiv\ & \neg P \vee (\neg Q \vee R) \\
\equiv\ & (\neg P \vee \neg Q) \vee R \\
\equiv\ & \neg(P \wedge Q) \vee R \\
\equiv\ & (P \wedge Q) \to R
\end{aligned}$$

(c) **Kommutativität** Betrachte zwei aussagenlogische Variablen P und Q und eine Interpretation I mit $I(P) = \mathbf{w}$ und $I(Q) = \mathbf{f}$. Dann gilt

$$val_I(P \to Q) = \mathbf{f} \neq \mathbf{w} = val_I(Q \to P).$$

Somit ist $\to$ nicht kommutativ.

Assoziativität Betrachte drei aussagenlogische Variablen P, Q und R. Sei I eine Interpretation mit $I(P) = I(Q) = I(R) = \mathbf{f}$. Dann gilt

$$val_I((P \to Q) \to R) = val_I(\neg(P \to Q) \vee \mathbf{f}) = val_I(\neg(\neg P \vee \mathbf{f})) = val_I(P) = \mathbf{f},$$

aber

$$val_I(P \to (Q \to R)) = val_I(\mathbf{f} \to (Q \to R)) = \mathbf{w}.$$

Somit ist $\to$ nicht assoziativ.

(d) (i) $\forall x\ (\neg \mathtt{N}(x) \to \exists y\ (\mathtt{W}(x,y) \wedge \forall z\ (\mathtt{W}(x,z) \to y \doteq z)))$ **(1,5 Punkte)**

(ii) $\forall x\ (\forall y\ \neg \mathtt{W}(x,y)) \to \mathtt{N}(x)$ **(1 Punkt)**

(iii) $\exists x\ \mathtt{W}(x,x)$ **(0,5 Punkte)**

Aufgabe 3 **$3 + 1 + 6 = 10$ Punkte**

(a) Geben Sie für die Sprache $L = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w \text{ endet mit } \mathtt{abb}\}$ einen endlichen Automaten A mit $L = L(A)$ an. Der Automat A darf höchstens 6 Zustände haben.

Wir betrachten nun die Grammatik G mit Terminalsymbolen $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, Variablen $V = \{S\}$, Startsymbol S und den folgenden Regeln R:

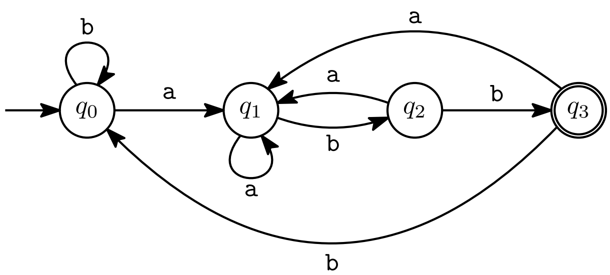
$$S \to \mathtt{a} \mid S\mathtt{a}S \mid S\mathtt{b}S$$

(b) Geben Sie einen Ableitungsbaum für das Wort `ababaaa` an.

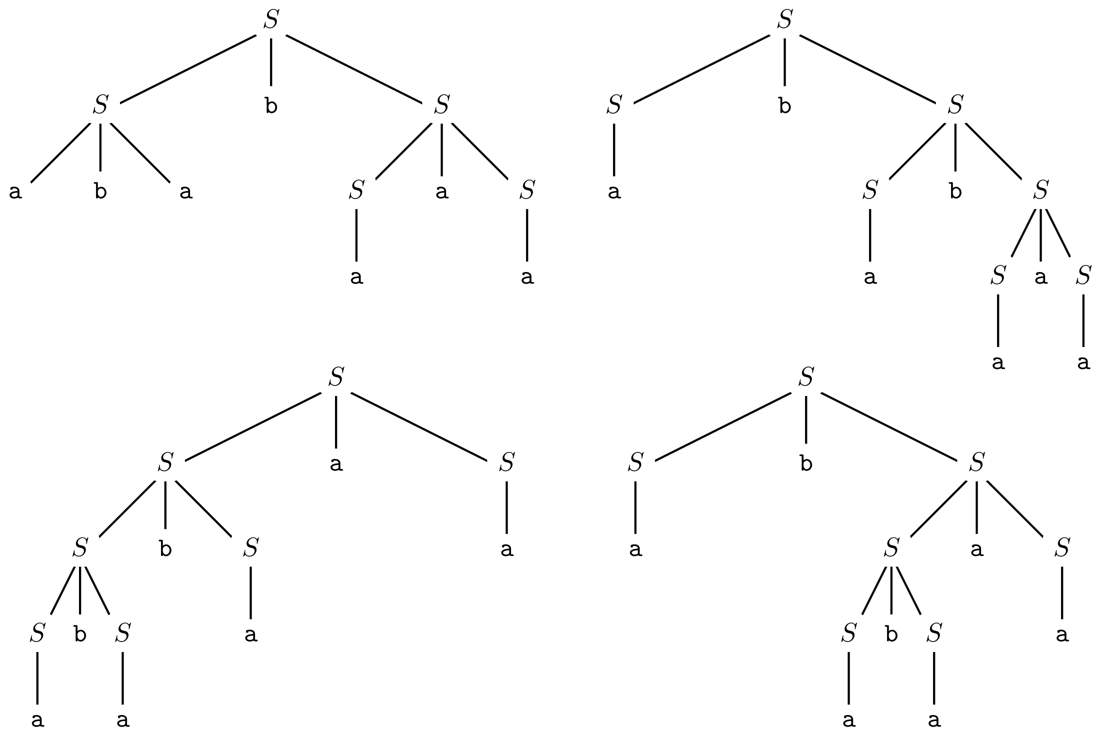
(c) Zeigen Sie mit vollständiger Induktion, dass für jedes von G erzeugte Wort w gilt: vor und nach jedem `b` kommt ein `a`.
Hinweis: Betrachten Sie den ersten Schritt der Ableitung.

Lösung

(a) Der folgende Automat erkennt die Sprache L:



(b) Alle möglichen Ableitungsbäume für das Wort `ababaaa`:



(c) Wir zeigen die Aussage mittels vollständiger Induktion über die Wortlänge $n = |w|$.

Induktionsanfang: Für $n \leq 1$ muss das Startsymbol S direkt zu $\mathtt{a}$ abgeleitet werden. Also enthält w kein $\mathtt{b}$ und die Aussage ist trivialerweise erfüllt.

Induktionsschritt: Sei $n > 1$. Wir betrachten einen beliebigen Ableitungsbaum von w. Da $|w| > 1$ kann die erste Ableitungsregel nicht $S \to \mathtt{a}$ sein. Die erste Ableitung hat somit die Form $S \to SxS$ mit $x \in \{\mathtt{a}, \mathtt{b}\}$. Es gilt also $w = w_1 x w_2$ für $w_1, w_2 \in L(G)$ mit $|w_1|, |w_2| < |w| = n$. Wir wissen durch Induktion, dass vor und nach allen $\mathtt{b}$ in w_1 und w_2 ein $\mathtt{a}$ kommt. Da jede Ableitung von G ein Terminal auf der rechten Seite hat, wissen wir außerdem, dass S nicht zum leeren Wort abgeleitet werden kann. Also wissen wir auch, dass w_1 und w_2 nicht gleich dem leeren Wort sind und somit mit $\mathtt{a}$ beginnen und aufhören. Also kommt vor und nach x ein $\mathtt{a}$. Damit ist vor und nach jedem $\mathtt{b}$ ein $\mathtt{a}$ und wir haben die Aussage gezeigt.

Aufgabe 4

$4 + 6 = 10$ Punkte

(a) Entscheiden Sie für die folgenden drei Funktionen jeweils, ob der dadurch induzierte Homomorphismus eine Huffman-Kodierung für das Wort $w = \texttt{erdbeere}$ ist. Wenn ja, geben Sie den dazugehörigen Huffman-Baum an. Wenn nein, begründen Sie.
Erinnerung: Eine Huffman-Kodierung φ eines Wortes w hat die Eigenschaft, dass $|\varphi(w)| \leq |\psi(w)|$ für alle präfixfreien Kodierungen ψ.

- $f\colon \{\texttt{e}, \texttt{r}, \texttt{d}, \texttt{b}\}^* \to \{\texttt{0}, \texttt{1}\}^*$ mit $f(\texttt{e}) = \texttt{0},\ f(\texttt{r}) = \texttt{10},\ f(\texttt{d}) = \texttt{110},\ f(\texttt{b}) = \texttt{111}$
  ☐ Ja, f^{**} ist eine Huffman-Kodierung für $w = \texttt{erdbeere}$.
  ☐ Nein, f^{**} ist keine Huffman-Kodierung für $w = \texttt{erdbeere}$.
- $g\colon \{\texttt{e}, \texttt{r}, \texttt{d}, \texttt{b}\}^* \to \{\texttt{0}, \texttt{1}\}^*$ mit $g(\texttt{e}) = \texttt{0},\ g(\texttt{r}) = \texttt{01},\ g(\texttt{d}) = \texttt{011},\ g(\texttt{b}) = \texttt{111}$
  ☐ Ja, g^{**} ist eine Huffman-Kodierung für $w = \texttt{erdbeere}$.
  ☐ Nein, g^{**} ist keine Huffman-Kodierung für $w = \texttt{erdbeere}$.
- $h\colon \{\texttt{e}, \texttt{r}, \texttt{d}, \texttt{b}\}^* \to \{\texttt{0}, \texttt{1}\}^*$ mit $h(\texttt{e}) = \texttt{00},\ h(\texttt{r}) = \texttt{01},\ h(\texttt{d}) = \texttt{10},\ h(\texttt{b}) = \texttt{11}$
  ☐ Ja, h^{**} ist eine Huffman-Kodierung für $w = \texttt{erdbeere}$.
  ☐ Nein, h^{**} ist keine Huffman-Kodierung für $w = \texttt{erdbeere}$.

(b) Zeigen Sie, dass für jeden Homomorphismus $p\colon \{\texttt{a}, \texttt{b}, \texttt{c}\}^* \to \{\texttt{a}, \texttt{b}, \texttt{c}\}^*$ mit

$$p(\texttt{ab}) = p(\texttt{bc}) = p(\texttt{ca}) \qquad (\star)$$

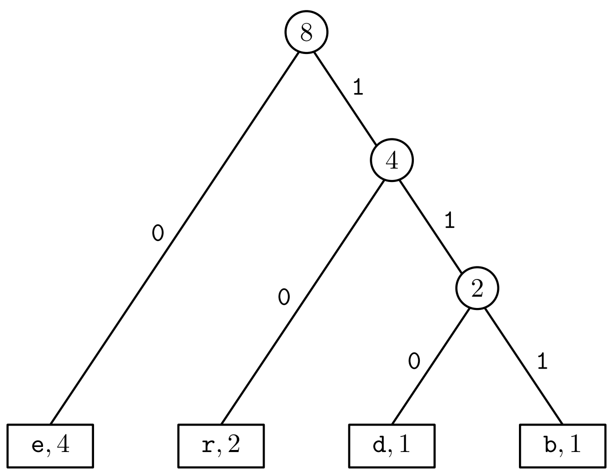
gilt, dass

(i) $|p(\texttt{a})| = |p(\texttt{b})| = |p(\texttt{c})|$
Hinweis: Sie dürfen (ii) ***nicht*** *verwenden.*

(ii) $p(\texttt{a}) = p(\texttt{b}) = p(\texttt{c})$
Hinweis: Sie dürfen (i) verwenden, um (ii) zu zeigen, auch wenn Sie (i) nicht bearbeitet haben.

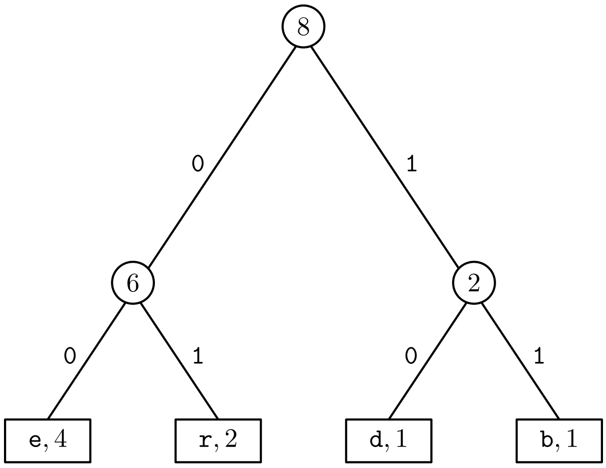
Lösung

(a)
- Ja, f^{**} ist eine Huffman-Kodierung für w.



- Nein, g^{**} ist keine Huffman-Kodierung für w, weil g nicht präfixfrei ist: $g(\texttt{e}) = \texttt{0}$ ist Präfix von $g(\texttt{r}) = \texttt{01}$.
- Nein, h^{**} ist keine Huffman-Kodierung des Wortes `erdbeere`, da $|h(\texttt{erdbeere})| = |\texttt{00 01 10 11 00 00 01 00}| = 16 > 14 = |\texttt{0 01 110 111 0 0 10 0}| = |f(\texttt{erdbeere})|$ und f präfixfrei ist. Also ist g keine kürzestmögliche präfixfreie Kodierung.

  Alternativ kann man sehen, dass der folgende Baum zu der Kodierung h^{**} kein legitimer Huffman-Baum ist.



  Dieser Baum ist kein Huffman-Baum, da nach dem ersten Schritt, in dem `d` und `b` zu einem Knoten vereinigt wurden, `e` und `r` vereinigt wurden, die höhere Häufigkeiten haben als `r` und der Elternknoten von `d` und `b`.

(b) (i) Da p ein Homomorphismus ist, gilt $p(\texttt{ab}) = p(\texttt{a}) \cdot p(\texttt{b}), p(\texttt{bc}) = p(\texttt{b}) \cdot p(\texttt{c})$ und $p(\texttt{ca}) = p(\texttt{c}) \cdot p(\texttt{a})$. Mit $x = |p(\texttt{a})|, y = |p(\texttt{b})|, z = |p(\texttt{c})|$ gilt also $|p(\texttt{ab})| = x + y, |p(\texttt{bc})| = y + z$ und $|p(\texttt{ca})| = z + x$. Mit $(\star)$ folgt $x + y = y + z = z + x$. Nach der erst Gleichheit gilt $x = z$ und nach der zweiten $y = x$, insgesamt also $x = y = z$ und somit $|p(\texttt{a})| = |p(\texttt{b})| = |p(\texttt{c})|$.

(ii) Zwei gleichlange Wörter sind genau dann gleich, wenn für jedes i von 1 bis Wortlänge das i-te Zeichen gleich ist. Da $p(\texttt{a}) \cdot p(\texttt{b}) \overset{\text{Hom}}{=} p(\texttt{ab}) \overset{(\star)}{=} p(\texttt{bc}) \overset{\text{Hom}}{=} p(\texttt{b}) \cdot p(\texttt{c})$ und nach (i) $|p(\texttt{a})| = |p(\texttt{b})|$, sind die ersten $|p(\texttt{a})| = |p(\texttt{b})|$ Zeichen gleich, also ist $p(\texttt{a}) = p(\texttt{b})$. Genauso sind die letzten $|p(\texttt{b})| = |p(\texttt{c})|$ Zeichen gleich, also ist $p(\texttt{b}) = p(\texttt{c})$.

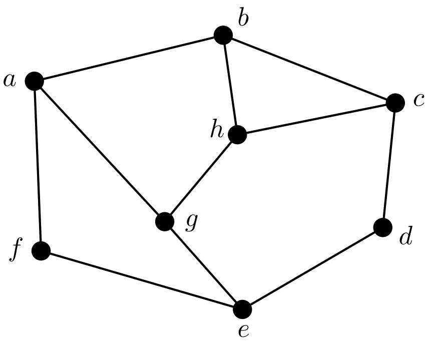
Aufgabe 5 **$2 + 1 + 2 + 5 = 10$ Punkte**

(a) Geben Sie für die folgenden Graphen an, wie viele Kanten sie in Abhängigkeit von der Anzahl Knoten $n \in \mathbb{N}_+$ haben.

- vollständiger Graph K_n:
- Baum auf n Knoten:
- Graph $G = (V, E)$ mit $V = [n]$ und $E = \{uv \in \binom{V}{2} \mid (u + v) \bmod 2 = 1\}$

Wir definieren ein neues Konzept: Sei $G = (V, E)$ ein Graph und $D \subseteq V$ eine Knotenmenge. Wir nennen D genau dann ein *Dominating Set* von G, wenn jeder Knoten von G einen Nachbarn in D hat oder selbst in D ist.

(b) Geben Sie ein Dominating Set D der Größe 2 für den folgenden Graphen an.



$D =$

(c) Geben Sie für die folgenden Graphen jeweils an, wie groß ein kleinstes Dominating Set in Abhängigkeit von der Anzahl Knoten $n \geq 3$ ist.

- vollständiger Graph K_n
- Kreis C_n

(d) Im Folgenden sind drei **falsche** Aussagen. Geben Sie jeweils ein Gegenbeispiel an und begründen Sie, warum es ein Gegenbeispiel ist.
Erinnerung: $N(v)$ bezeichnet die Nachbarschaft eines Knotens v.

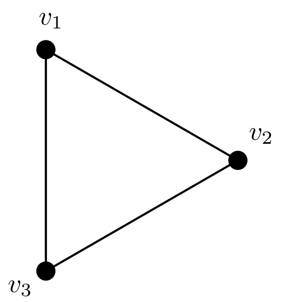
- $D \subseteq V$ ist genau dann ein Dominating Set eines Graphen $G = (V, E)$, wenn
$$\forall uv \in E \colon u \in D \lor v \in D. \tag{1}$$
Gegenbeispiel und Begründung:
- $D \subseteq V$ ist genau dann ein Dominating Set eines Graphen $G = (V, E)$, wenn
$$\forall u \in D \, \exists v \in N(u) \colon v \notin D \tag{2}$$
Gegenbeispiel und Begründung:
- $D \subseteq V$ ist genau dann ein Dominating Set eines Graphen $G = (V, E)$, wenn
$$\sum_{v \in D} |N(v) \cup \{v\}| \geq |V| \tag{3}$$
Gegenbeispiel und Begründung:

Lösung

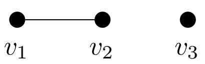
(a)
- $\binom{n}{2} = \frac{n\cdot(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2}$ **(0,5 Punkte)**
- $n - 1$ **(0,5 Punkte)**
- $\lfloor \frac{n}{2} \rfloor \cdot \lceil \frac{n}{2} \rceil$ **(1 Punkt)**

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

(c)
- Das kleinste Dominating Set eines vollständigen Graphen K_n hat Größe 1.
- Das kleinste Dominating Set eines Kreises C_n hat Größe $\lceil \frac{n}{3} \rceil$.

(d)
- 

  Sei G der abgebildete Graph und $D = \{v_1\}$. Dann ist jeder Knoten entweder in D oder adjazent zu einem Knoten in D. Also ist D ein Dominating Set. Aber für die Kante v_2v_3 ist weder v_2 noch v_3 in D. Also ist (1) nicht erfüllt. **(2 Punkte)**
- Sei $G = K_1$ und $D = V(G)$. D ist ein Dominating Set, weil es alle Knoten enthält, aber es erfüllt nicht (2), weil der einzige Knoten keine Nachbarn hat. **(1 Punkt)**
- G besteht aus einer Kante v_1v_2 und einem isolierten Knoten v_3 wie im Bild.

  $D = \{v_1, v_2\}$ ist kein Dominating Set, weil v_3 nicht dominiert wird, aber $\sum_{v\in D} |N(v) \cup \{v\}| = |N(v_1) \cup \{v_1\}| + |N(v_2) \cup \{v_2\}| = |\{v_2, v_1\}| + |\{v_1, v_2\}| = 4 \geq 3 = |V|$. **(2 Punkte)**

Aufgabe 6

$2 + 2 + 1 + 1 + 2 + 2 = 10$ Punkte

(a) Gegeben seien die folgenden Mengen von Funktionen:

$$A = \mathcal{O}(\sqrt{n}) \qquad B = \mathcal{O}(n^2) \qquad C = \mathcal{O}(\sqrt[3]{n})$$
$$D = \mathcal{O}(2^n) \qquad E = \mathcal{O}(n^3 \log(n)) \qquad F = \mathcal{O}(4^n).$$

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 $X \subseteq Y$.

$$2\log(n) + 2 \in \square \qquad 2^{2025} \in \square \qquad (3n^2 - 9n) \cdot n \in \square \qquad n^4 \in \square$$

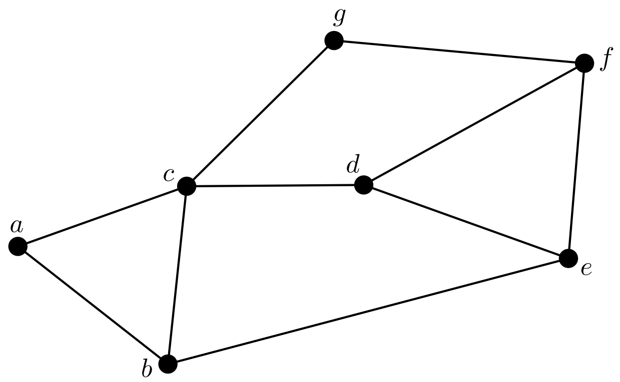
Gegeben sei der folgende Algorithmus $\mathcal{A}$, der als Eingabe einen Graphen $G = (V, E)$ bekommt und eine Menge D ausgibt.

Algorithmus $\mathcal{A}$

Eingabe: Graph $G = (V, E)$
Ausgabe: Menge D

$D \leftarrow \emptyset$
for all $uv \in E$ **do**
  for all $x \in V \setminus \{u, v\}$ **do**
    if $ux \in E$ und $vx \in E$ **then**
      $D \leftarrow D \cup \{\{u, v, x\}\}$

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



$D =$

(c) Zeichnen Sie einen Graphen mit mindestens 6 Kanten, für den die Ausgabe von $\mathcal{A}$ die leere Menge ist.

(d) Geben Sie für jedes $n \geq 3$ einen Graphen $G_n = (V_n, E_n)$ mit n Knoten an, bei dem die Ausgabe von $\mathcal{A}$ jede 3-elementige Teilmenge von V_n enthält.

(e) Geben Sie die Laufzeit von $\mathcal{A}$ mit Eingabe $G = (V, E)$ in Abhängigkeit von $n = |V|$ möglichst genau im $\mathcal{O}$-Kalkül an. Gehen Sie davon aus, dass die Ausführung jedes einzelnen Befehls (d. h. jeder einzelnen Zeile) $\mathcal{O}(1)$ Zeit benötigt. Vereinfachen Sie so weit wie möglich.

(f) Sei $n \in \mathbb{N}$ und G ein zusammenhängender Graph mit n Knoten, sodass Algorithmus $\mathcal{A}$ bei Eingabe G die Zeile 4 (if ...) insgesamt $n^{5/2}$ mal ausführt. Geben Sie die Laufzeit einer Breitensuche (BFS) auf G in Abhängigkeit von n möglichst genau im $\mathcal{O}$-Kalkül an. Vereinfachen Sie so weit wie möglich.

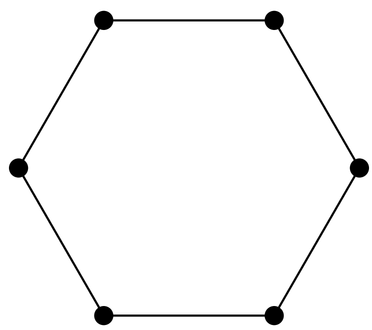
Lösung

(a)

$$(\log(n) + 2)^2 \in C \qquad 2^{2025} \in C \qquad 3n^3 - 9n^2 \in E \qquad n^4 \in D$$

(b) Der Algorithmus gibt die Menge $D = \{\{a, b, c\}, \{d, e, f\}\}$ aus.

(c) Für jeden dreiecksfreien Graphen ist die Ausgabe des Algorithmus die leere Menge. Also zum Beispiel für den folgenden Graphen:



(d) Für jede dreielementige Knotenmenge $V' \subseteq V_n$, muss jede Kante aus $\binom{V'}{2}$ in E_n sein. Insbesondere ist also zwischen jedem Paar von Knoten eine Kante. Also gilt $G_n = K_n$.

(e) Der Algorithmus betrachtet für jede Kante jeden Knoten. Ein Graph hat höchstens $\binom{n}{2} \in \mathcal{O}(n^2)$ Kanten. Also hat der Algorithmus Laufzeit $\mathcal{O}(n^3)$.

(f) Für jede Kante in G wird Zeile 4 von Algorithmus $\mathcal{A}$ insgesamt $n-2$ mal ausgeführt. G hat also $n^{5/2}/(n-2) \in \Theta(n^{3/2})$ Kanten. Da eine BFS Laufzeit $\mathcal{O}(m)$ hat, wobei m die Anzahl Kanten ist, hat die BFS mit Eingabe G Laufzeit $\mathcal{O}(n^{3/2})$.