

Dr. Arne Schmidt

Klausur
Parametrisierte Algorithmen
12.03.2026

Nachname:

Vorname:

Matr.-Nr.:

Studiengang:

☐ Bachelor ☐ Master ☐ Andere:

Hinweise:

- Bitte das Deckblatt in Druckschrift vollständig ausfüllen.
 - Die Klausur besteht aus 12 Blättern, bitte auf Vollständigkeit überprüfen.
 - Die Bearbeitungszeit für die Klausur beträgt 120 Minuten.

 - Erlaubte Hilfsmittel: keine
 - Eigenes Papier ist nicht erlaubt.
 - Die Rückseiten der Blätter dürfen beschrieben werden.
 - Antworten, die *nicht* gewertet werden sollen, bitte deutlich durchstreichen.
 - Kein Tippex verwenden.
 - Werden mehrere Antworten gegeben, werten wir die mit der geringsten Punktzahl.
 - Mit *Bleistift* oder in *Rot* geschriebene Klausurteile können nicht gewertet werden.
-
-

Aufgabe	1	2	3	4	5	6	Σ
Max	20	25	15	15	10	15	100
Erreicht							

Aufgabe 1: Verschiedenes

(5+5+10 Punkte)

a) Wie sind die Klassen FPT und XP für parametrisierte Probleme definiert?

b) Betrachte folgendes generisches Problem:

Gegeben Eine Menge $U = \{u_1, \dots, u_n\}$ und eine Zahl k .

Parameter k

Frage Existiert $S \subseteq U$ mit $|S| \leq k$, sodass S Eigenschaft $P(S)$ besitzt?

Zeige: Ist die Eigenschaft $P(S)$ für jedes S in polynomieller Zeit überprüfbar, dann ist das Problem in XP.

c) Gehe im folgenden von der üblichen Annahme $P \neq NP$ aus. Kreuze an, ob die Aussagen wahr oder falsch sind.

(i) FPT und XP

Wahr	Falsch	
☐	☐	Jedes parametrisierte Problem ist in FPT oder XP.
☐	☐	Zwischen FPT und XP gibt es unendlich viele Zwischenklassen.
☐	☐	Es ist bekannt, dass $FPT \neq XP$ gilt.

(ii) W-Hierarchie

Wahr	Falsch	
☐	☐	Es ist ein $t \geq 1$ bekannt, für welches $W[t] \neq W[t+1]$ gilt.
☐	☐	Ist ein Problem $W[2]$ -schwer, dann ist es auch $W[1]$ -schwer.
☐	☐	Für jedes $t \geq 1$ ist die Vermutung $W[t] \subseteq FPT$

(iii) Kernelisation

Wahr	Falsch	
☐	☐	Besitzt ein parametrisiertes Problem einen Kernel, dann liegt das Problem in FPT.
☐	☐	Liegt ein parametrisiertes Problem in FPT, dann existiert für das Problem ein Kernel.
☐	☐	Ein Kernel hat immer polynomielle Größe im Parameter.

(iv) Bounded Tree Search mit Parameter k

Wahr	Falsch	
☐	☐	Jede Entscheidung reduziert k um mindestens 1.
☐	☐	Man darf höchstens $O(k)$ viele Entscheidungen treffen.
☐	☐	Der resultierende Entscheidungsbaum hat immer eine Tiefe von mindestens k .

(v) Baumweite

Wahr	Falsch	
☐	☐	Baumgraphen (kreisfrei) besitzen Baumweite 1.
☐	☐	Vollständige Graphen auf n Knoten besitzen Baumweite $n - 1$.
☐	☐	Ist G ein Graph und H ein Minor von G , dann gilt $tw(H) \leq tw(G)$.

Aufgabe 2: Vertex Cover — Kernelization**(6+5+7+4+3 Punkte)**

a) Betrachte folgende Reduktionsregel für Vertex Cover:

Sei (u, v, w, z) ein Pfad mit $\delta(v) = \delta(w) = 2$. Kontrahiere v mit u , sowie w mit z und reduziere k um 1 (siehe Abbildung 1).

Zeige oder widerlege: Diese Regel ist sicher.



Abbildung 1: (Links) Originalgraph (Rechts) Nach Kontraktion

b) Was ist eine Kronenzerlegung für einen Graphen $G = (V, E)$?

c) Bestimme eine Kronenzerlegung der Instanz aus Abbildung 2. Beachte dazu folgende Punkte:

- Bestimme ein inklusionsmaximales Matching per Greedy-Algorithmus. Betrachte die Kanten dazu in lexikographischer Ordnung, d.h. Kante $\{v_i, v_j\}$ kommt vor Kante $\{v_k, v_\ell\}$, wenn entweder $i < k$ oder $i = k \wedge j < \ell$ gilt. Das Matching kann direkt in die Abbildung eingezeichnet werden.
- Bestimme die Partition (C, H, R) über das erhaltene Matching.

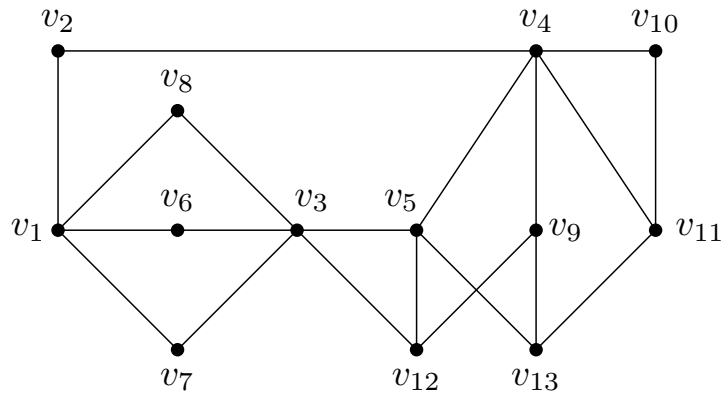


Abbildung 2

d) Benutze die in c) erhaltene Kronenzerlegung, um die Instanz zu reduzieren. Gib dazu den neuen Graphen an.

e) Begründe mit Hilfe der reduzierten Instanz, ob der ursprüngliche Graph ein Vertex Cover der Größe 5 enthält.

Aufgabe 3: Bounded-Tree-Search**(6+9 Punkte)**

- a) Beschreibe, wie Bounded-Tree-Search für Vertex Cover über Knoten benutzt werden kann. Leite die Laufzeit her. Nutze zur Bestimmung einer Nullstelle Tabelle 1.

(i, j)	1	2	3	4	5	6
1	2.0000	1.6181	1.4656	1.3803	1.3248	1.2852
2		1.4143	1.3248	1.2721	1.2366	1.2107
3			1.2560	1.2208	1.1939	1.1740
4				1.1893	1.1674	1.1510
5					1.1487	1.1348
6						1.1225

Tabelle 1: Positive Nullstellen für Polynome der Form $P(n) = \lambda^j - \lambda^{j-i} - 1$

- b) Betrachte den in der nachstehenden Abbildung dargestellten Graphen G . Gib den gesamten Baum der Bounded-Tree-Search auf G an, und bestimme damit, ob G ein Vertex Cover der Größe $k = 5$ besitzt. Branche dabei jeweils über den Knoten mit dem höchsten Grad. Stehen mehrere zur Auswahl, wähle den mit dem kleinsten Index. Bei trivialen Instanzen (Maximalgrad 2 oder $k \leq 0$) muss nicht weiter gebrannt werden. Gib nach jedem Branching die jeweils noch zu betrachtende Instanz an.

(Hinweis: Nutze die Vorlage auf der nächsten Seite für die Instanzen und nutze Nummern, um zwischen dem Bounded-Search-Tree und den Instanzen zu verweisen.)

k = 5 Nr. 1	k = Nr.
k = Nr.	k = Nr.
k = Nr.	k = Nr.
k = Nr.	k = Nr.

Aufgabe 4: Iterative Compression**(8+2+5 Punkte)**

Betrachte das Problem CLUSTER VERTEX DELETION:

Gegeben: Ungerichteter Graph $G = (V, E)$, sowie eine Zahl $k \in \mathbb{N}$.**Parameter:** k **Frage:** Existiert eine Menge von höchstens k Knoten D , so dass $G \setminus D$ ein *Clustergraph* ist, d.h. jede Zusammenhangskomponente ist ein vollständiger Graph?

Das dazu passende Problem DISJOINT CLUSTER VERTEX DELETION (DCVD) ist wie folgt definiert:

Gegeben: Ungerichteter Graph $G' = (V', E')$, eine CVD-Lösung $S \subset V'$, sowie eine Zahl $k' \in \mathbb{N}$.**Parameter:** k' **Rückgabe:** Eine Menge von höchstens k' Knoten D , sodass $D \cap S = \emptyset$ und $G' \setminus D$ ein *Clustergraph* ist, oder die Ausgabe, dass D nicht existiert.

- a) Angenommen, es gibt einen Algorithmus $\text{DISJOINT}(G', S, k')$ mit Laufzeit $O(\alpha^{k'} |G|^c)$ für $\alpha, c \in \mathbb{N}$, welcher DCVD löst. Entwirf einen Algorithmus, der CLUSTER VERTEX DELETION mit Hilfe des Algorithmus löst und analysiere die Laufzeit.

- b) Betrachte weiterhin DCVD. Welche Struktur müssen die Teilgraphen $G[S]$, sowie $G[V \setminus S]$ in einer Ja-Instanz besitzen?
- c) Argumentiere mit einer Methode deiner Wahl, warum DCVD in FPT liegt.
(Hinweis: Zur Erinnerung: Ein Graph ist ein Clustergraph, wenn es keinen induzierten Pfad mit drei Knoten gibt.)

Aufgabe 5: Tree Width**(10 Punkte)**

Bestimme die Baumweite des in Abbildung 3 dargestellten Graphen. Gib dazu eine passende Baumzerlegung an und zeige, dass es keine bessere geben kann.

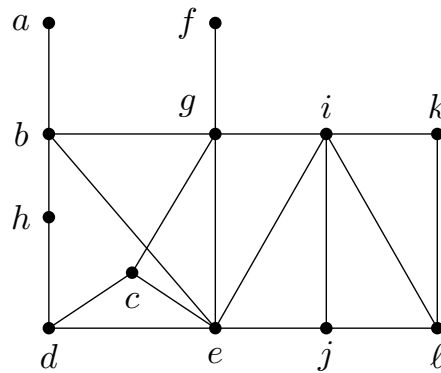


Abbildung 3

Aufgabe 6: Intractability**(6+9 Punkte)**

Betrachte das parametrisierte Problem HITTING SET:

Gegeben Universum $U = \{u_1, \dots, u_n\}$, Familie $\mathcal{F} \subseteq \mathcal{P}(U)$ und $k \in \mathbb{N}$.

Parameter k

Frage: Existiert $H \subseteq U$ mit $|H| \leq k$, sodass für jedes $F \in \mathcal{F}$ gilt $F \cap H \neq \emptyset$?

(Hinweis: Auf der letzten Seite gibt es eine Tabelle mit Problemen und Komplexitätsklassen.)

a) Zeige: Hitting Set liegt in $W[2]$.

b) Zeige: Hitting Set ist $W[2]$ -schwer.

Problem	Parameter	Klasse
VERTEX COVER	k	FPT
VERTEX COVER	Baumweite	FPT
INDEPENDENT SET	Baumweite	FPT
INDEPENDENT SET	k	W[1]-vollständig
CLIQUE	k	W[1]-vollständig
DOMINATING SET	k	W[2]-vollständig
SET COVER	k	W[2]-vollständig
WEIGHT. 2-NORM. SAT.	k	W[2]-vollständig
WEIGHT. MONO. 2-NORM. SAT.	k	W[2]-vollständig

Viel Erfolg ☺