

Masterarbeit

**Effiziente Berechnung von K_5 -Minoren
in Graphen**

Julian Sauer

20. September 2019

Betreuer:

Prof. Dr. Petra Mutzel

Prof. Dr. Jens Schmidt

Fakultät für Informatik

Algorithm Engineering (LS 11)

Technische Universität Dortmund

<http://ls11-www.cs.tu-dortmund.de>

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation und Hintergrund	1
1.2	Aufbau der Arbeit	1
2	Definitionen	3
3	Algorithmus von Kezdy und McGuinness	9
3.1	Behandlung von $K_{3,3}$ -Minoren	9
3.2	Sequenzieller Algorithmus zum Finden von K_5 -Minoren	17
4	Wagner Struktur	19
4.0.1	2-Zusammenhang	19
4.0.2	3-Zusammenhang	21
4.0.3	(3, 3)-Separatoren	26
4.0.4	Wagner-Struktur	28
4.1	Algorithmus von Kezdy und McGuinness als Wagner-Struktur	28
4.1.1	Block-Trees	28
4.1.2	Zertifizierender Algorithmus	30
5	Implementierung	33
5.1	Open Graph Drawing Framework	33
6	Experimentelle Analyse	35
7	Zusammenfassung und Ausblick	37
A	Weitere Informationen	39
	Abbildungsverzeichnis	42
	Algorithmenverzeichnis	43
	Symbolverzeichnis	45

Literaturverzeichnis	48
Eidesstattliche Versicherung	48

Kapitel 1

Einleitung

1.1 Motivation und Hintergrund

Zunächst kann durch die Berechnung von K_5 -Minoren entschieden werden, ob ein Graph K_5 -Minor frei ist. Das ist insofern interessant, als das einige Algorithmen effizienter auf solchen Graphen ausgeführt werden können. So ist die Berechnung eines *maximalen Schnittes* (*Max-Cut Problem*) - also das Aufteilen der Knoten eines Graphen in zwei Mengen, sodass die Kanten zwischen diesen beiden Mengen in Summe ein maximales Gewicht besitzen - NP-schwer [9]. Allerdings wird etwa in [2] gezeigt, dass es für Graphen ohne K_5 -Minor in Polynomialzeit gelöst werden kann. Darüber hinaus können viele kombinatorische Optimierungsprobleme wie quadratische 0-1 Probleme als Max-Cut Probleme formuliert werden [5]. In [8] transformieren Jünger et al. Ising Spin Glass Probleme zu Max-Cut Problemen, um mit Hilfe von einem Branch-and-Cut Algorithmus eine optimale Lösung zu finden. Neben diesem Ansatz können Ising Spin Glass Probleme ebenfalls auf Quantencomputern wie dem D-Wave 2000Q heuristisch, aber dafür schneller gelöst werden. Zur Bewertung ihrer Qualität können sie mit denen von Jünger et al. verglichen werden. Innerhalb von diesem exakten Verfahren stellen die K_5 -Minoren die Schwierigkeit des Problems dar. Von daher ist es einerseits interessant für die Berechnung, wenn kein K_5 -Minor enthalten ist, sodass das Problem effizient gelöst werden kann. Andererseits können einer oder mehrere gefundene K_5 -Minoren benutzt werden, um innerhalb des linearen Programms den Suchraum zu beschränken und die Berechnung zu beschleunigen.

1.2 Aufbau der Arbeit

Kapitel 2

Definitionen

Vorab werden einige Definitionen und Notationen festgelegt, die im Verlauf der Arbeit verwendet werden. Der Algorithmus arbeitet mit einem ungerichteten Graph $G = (V, E)$ ohne Mehrfachkanten, wobei E die Menge der Kanten und V die Knotenmenge sei.

Eine Kante $e \in E$, die zwei Knoten u und v verbindet, wird durch $e = (u, v)$ angegeben.

Ein Pfad $P(u, v)$ verbindet zwei Knoten u und v über eine Folge von Knoten, die adjazent zueinander sind. Bei der Kontraktion einer Kante $e = (u, v)$ wird diese mit ihren beiden End-

punkten aus dem Graph entfernt und einen neuen Knoten w ersetzt. Die Nachbarknoten von w werden auf die Menge der adjazenten Knoten von u und v gesetzt. In Abbildung 2.1 ist das Vorgehen skizziert. Analog kann, wie in Abbil-

dung 2.2 gezeigt, ein Pfad kontrahiert werden, wobei der neu eingefügte Knoten w Kanten zu der Menge der adjazenten Knoten aller Knoten des Pfades besitzt.

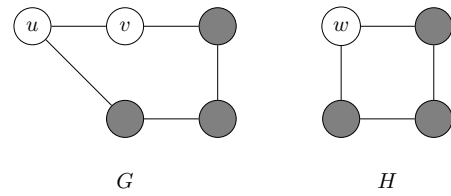


Abbildung 2.1: Die Kante, die u und v in G verbindet, wird kontrahiert, sodass sie in H durch den neuen Knoten w ersetzt wird.

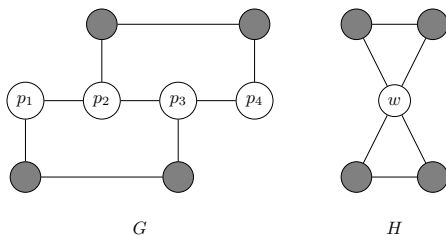


Abbildung 2.2: Der Pfad von p_1 bis p_4 wird kontrahiert. Der neue Knoten w in H enthält alle Nachbarn der Pfadknoten in G .

Ein Minor H eines Graphen G bezeichnet einen Graph, der isomorph zu G ist, nachdem eine beliebige Menge an Operationen von Kantenkontraktionen, Kantenentfernungen und Knotenentfernungen durchgeführt wurde. Ein Beispiel dazu findet sich in Abbildung 2.3. Jeder Graph ist sein eigener Minor, genauso ist jeder Teilgraph ein gültiger Minor. Dass H ein Minor von G ist, wird dargestellt durch $H \prec_M G$. Das Branch-Set eines Knotens w aus einem Minor von G bezeichnet die Menge an Knoten, die durch Kontraktionen zu w verschmolzen wurden. In

Abbildung 2.3 besteht beispielsweise das Branch-Set von g aus $\{a, b, c\}$, zu f in H_2 gehört die Knotenmenge $\{f\}$ in G . Für die Knotenmenge $U \in V$ bezeichnet $G - U$ den Teilgraph, der entsteht, wenn alle Knoten aus U mit ihren inzidenten Kanten aus G entfernt werden. Ein Subdivision eines Graphen G enthält alle Knoten und Kanten aus G , zusätzlich können aber Kanten als gegenteilige Operation zur Kontraktion unterteilt werden. Für eine Kante $e = (u, v)$ bedeute das, dass e entfernt wird, dafür ein neuer Knoten w und zwei neue Kanten (u, w) und (w, v) eingefügt werden.

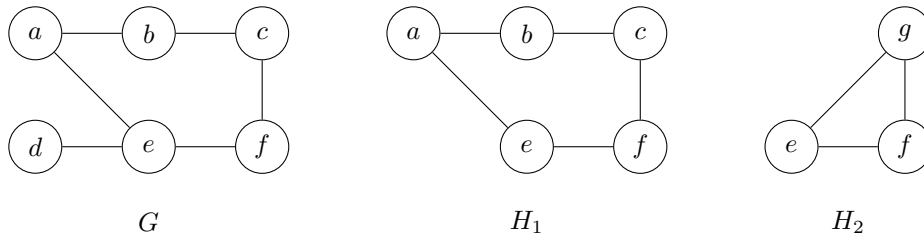


Abbildung 2.3: Ein Graph G mit seinen Minoren H_1 und H_2 . Um H_1 zu erhalten, wurde in G die Kante (d, e) und anschließend der Knoten d entfernt. Für H_2 wurden außerdem der Pfad $P(a, c)$ kontrahiert.

Ein Graph wird als planar bezeichnet, wenn er sich so in der Ebene einbetten lässt, dass sich keine Kanten kreuzen. Ein K_5 (s. Abb. 2.4) ist ein spezieller Graph, der aus fünf Knoten besteht, die alle zueinander adjazent sind. Ein $K_{3,3}$ (s. Abb. 2.5) ist ein vollständig bipartiter Graph mit sechs Knoten. Er lässt sich also in zwei Knotenmengen unterteilen (im Folgenden als rote und blaue Menge bezeichnet), sodass alle Knoten der einen Menge zu allen Knoten der anderen Menge benachbart sind. Nach dem Satz von Kuratowski ist ein Graph planar, wenn er kein K_5 - oder $K_{3,3}$ -Subdivision als Teilgraph beinhaltet. Eine alternative Formulierung von Wagner sagt aus, dass ein Graph planar ist, wenn er keinen K_5 -Minor oder $K_{3,3}$ -Minor enthält. [4]

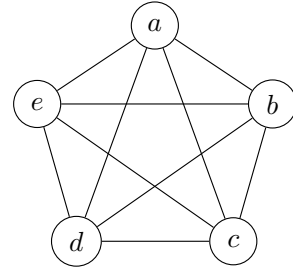


Abbildung 2.4: Der Graph K_5 .

Als nächstes wird ein $K_{3,3}$ genauer betrachtet. Sei dessen rote Knotenmenge $R = \{a, b, c\}$ und blaue $B = \{x, y, z\}$. Diese sechs Knoten werden in einem $K_{3,3}$ -Subdivision H als Branch-Ends genannt und zeichnen sich dadurch aus, dass sie als einzige Knoten in H den Grad 3 haben. Ein Branch-Path in H ist ein Pfad, der zwei Branch-Ends verbindet, beispielsweise $P(a, x)$. Ein Branch-Fan bezieht sich immer auf einen der Branch-Ends und wird z.B. für a als $F(a)$ geschrieben. Bezeichnet werden

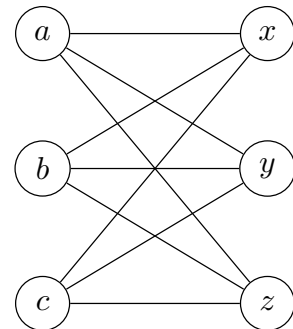


Abbildung 2.5: Der Graph $K_{3,3}$.

dadurch alle Pfade, die zu einem anderen Branch-End führen - für a also die Pfade $P(a, x)$, $P(a, y)$ und $P(a, z)$.

Der Graph W bezeichnet einen speziellen Graph, der aus acht Knoten besteht. Seine äußeren Kanten bilden einen Kreis, außerdem sind die Knoten jeweils adjazent zu den gegenüberliegenden. Eine Darstellung findet sich links in Abbildung 2.6. Er enthält einen $K_{3,3}$ als Minor (in der Abbildung rechts angedeutet), jedoch keinen K_5 . Als M wird ein Graph bezeichnet, der einen $K_{3,3}$ als Teilgraph enthält, jedoch einen zusätzlichen Knoten und zwei zusätzliche Kanten enthält. Er ist insofern interessant, als dass er, wie in Abbildung 2.7 zu sehen, neben einem $K_{3,3}$ -Minor auch einen K_5 -Minor enthält.

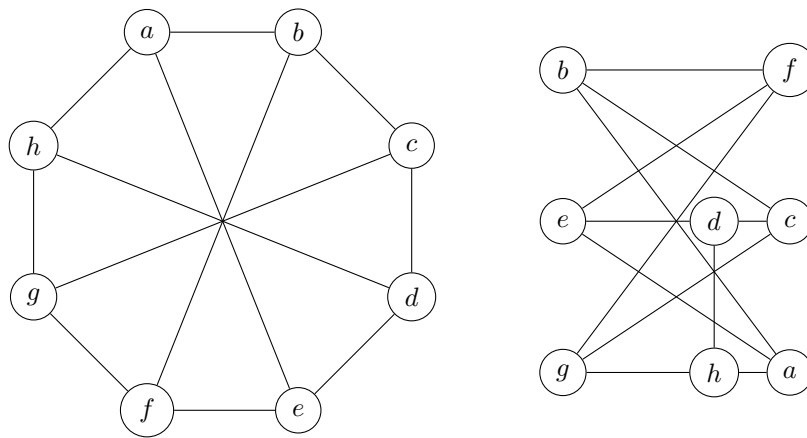


Abbildung 2.6: Der Graph W , links in seiner üblichen Darstellung, rechts mit angedeutetem $K_{3,3}$ -Minor.

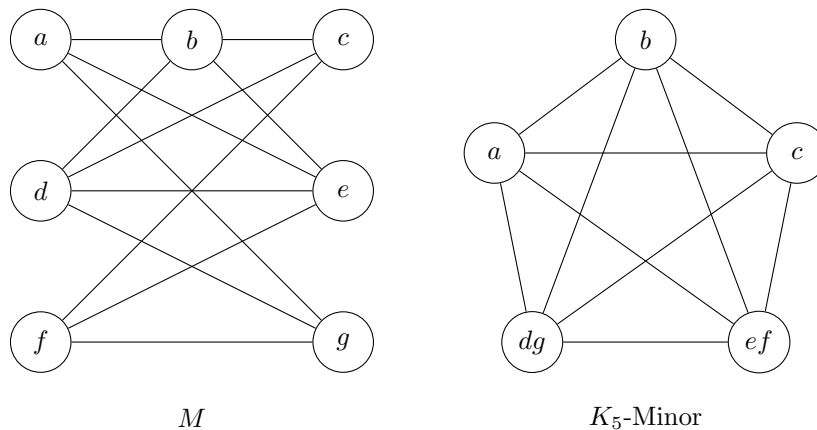


Abbildung 2.7: Der Graph M sowie ein K_5 -Minor aus M .

Als (i) -Separator wird eine Menge S bestehend aus i Knoten in einem zusammenhängenden Graph G bezeichnet, sodass $G - S$ nicht mehr zusammenhängend ist. Ein (i, j) -Separator ist ein i -Separator, sodass $G - S$ aus mindestens j Zusammenhangskomponenten

besteht. In Abbildung 2.8 wird ein $(3, 3)$ -Separator im linken Graph gezeigt. Durch einen Separator können augmentierte Komponenten definiert werden: Sind $C \subseteq V$ die i Knoten, die zu einem (i, j) -Separator in G gehören, wird der Graph durch $G - C$ zunächst in j Zusammenhangskomponenten zerlegt. Anschließend werden Kopien der Knoten aus C zu jeder Komponente hinzugefügt. Sie bilden dabei in jeder Zusammenhangskomponente eine Clique. Außerdem besitzen sie Kanten zu Knoten in der Zusammenhangskomponente, falls eine Kante $e = (c, z)$ in G existierte, $c \in C$ und z ein Knoten der Zusammenhangskomponente ist. Jeder der resultierenden Graphen ist eine augmentierte Komponente des Ursprungsgraphen G und wie in Theorem 3.1.1 bewiesen wird, evtl. ein Minor zu G . Als gegenteilige Operation kann eine Cliques-Summe verwendet werden, um zwei Graphen zu verschmelzen. Dazu müssen zwei Graphen H_1 und H_2 als Teilgraph eine gleich große Clique enthalten. H_1 und H_2 können zu einem Graph zusammengefügt werden, indem je ein Knoten von der Clique in H_1 und in H_2 zu einem zusammengefügt werden. Im resultierenden Graph dürfen außerdem Kanten zwischen den Knoten der Clique entfernt werden. Dadurch ist es möglich, einen Graph G in augmentierte Komponente zu zerlegen und anschließend durch eine Cliques-Summe wieder G zu erhalten. Dieses Vorgehen wird in Abbildung 2.8 gezeigt, dabei wird der linke Graph rechts in augmentierte Komponenten zerlegt bzw. von rechts nach links werden drei Graphen durch eine Cliques-Summe zusammengefügt.

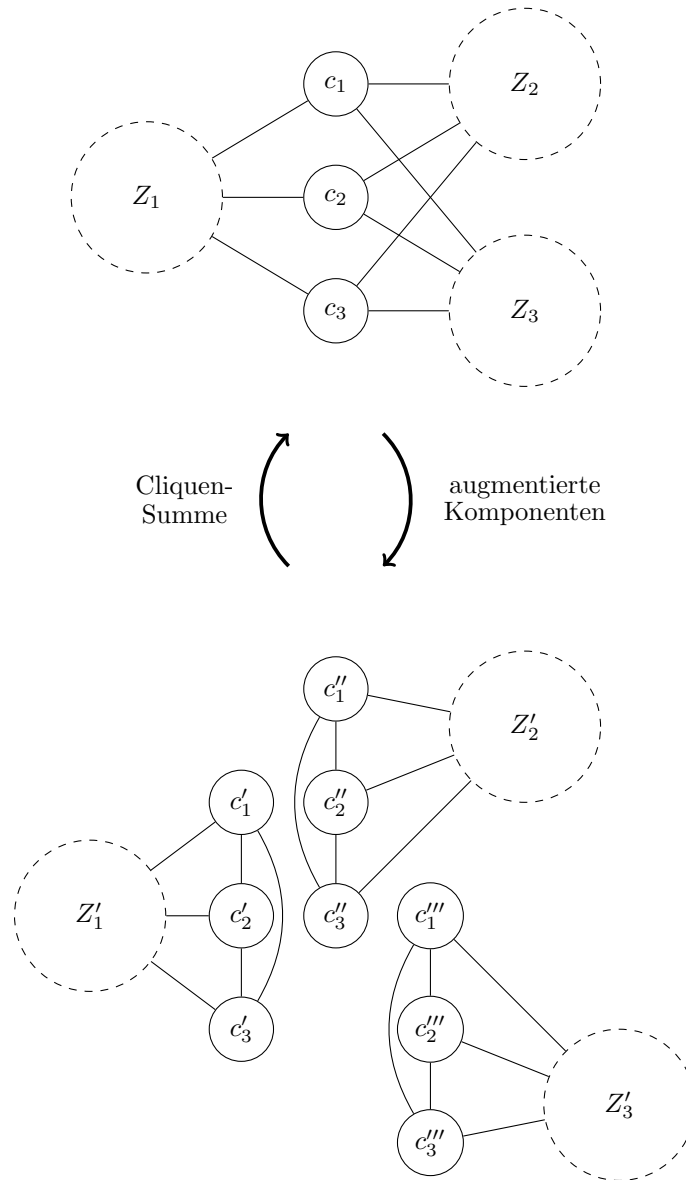


Abbildung 2.8: Der linke Graph wird in drei augmentierte Komponenten durch den $(3, 3)$ -Separator $C = \{c_1, c_2, c_3\}$ geteilt. Alle Z_i und Z'_i stellen Teilgraphen dar, die zur Übersicht zu einem Knoten zusammengefügt wurden. Die drei rechten Graphen können durch die Cliques-Summe der Cliques $\{c'_1, c'_2, c'_3\}$ sowie $\{c''_1, c''_2, c''_3\}$ und $\{c'''_1, c'''_2, c'''_3\}$ den rechten Graph erzeugen. Während der Cliques-Summen Operation dürfen die Kanten, die die Knoten in den Cliques verbinden, gelöscht werden.

Kapitel 3

Algorithmus von Kezdy und McGuinness

Da die Arbeit auf dem sequenziellen Algorithmus von Kezdy und McGuinness, den sie in [1] vorstellen, beruht, wird er im Folgenden erklärt. Als Eingabe wird ein ungerichteter Graph ohne Mehrfachkanten erwartet, ausgegeben wird, ob ein K_5 -Minor enthalten ist oder nicht. Für den Fall, dass einer gefunden wurde, kann zusätzlich ausgegeben werden, welche Knoten den Minor formen. Die Laufzeit liegt in $\mathcal{O}(n^2)$.

Planaritätstests können bereits in linearer Laufzeit entscheiden, ob ein Graph einen K_5 - oder $K_{3,3}$ -Minor enthält. Es muss folglich der Fall behandelt werden, in dem der Test stoppt, weil er einen $K_{3,3}$ -Minor gefunden hat. Denn es kann nicht garantiert werden, ob zusätzlich ein K_5 -Minor enthalten ist. Als Lösung testet der Algorithmus von Kezdy und McGuinness, ob drei der Knoten eines gefundenen $K_{3,3}$ -Minor einen $(3,3)$ -Separator bilden und teilt den Graph in augmentierte Komponenten auf. Anschließend kann der Planaritätstest auf die einzelnen Komponenten rekursiv angewendet werden.

3.1 Behandlung von $K_{3,3}$ -Minoren

Die folgenden Theoreme und ihre Beweise sind [1] entnommen.

Um das zentrale Theorem aus [1], welches den $K_{3,3}$ -Minor untersucht, zu erklären, wird zunächst die Gültigkeit augmentierter Komponenten behandelt:

3.1.1 Theorem. *Für $k \geq 3$: Sei G ein k -zusammenhängender Graph und C ein k -Separator in G . Jede durch C definierte augmentierte Komponente ist ein Minor von G genau dann, wenn es entweder mindestens k Komponenten sind oder mindestens zwei der Komponenten jeweils aus mehr als einem Knoten bestehen.*

Beweis. Seien c_1, c_2, \dots, c_k die Knoten von C und $Z = \{Z_1, Z_2, \dots, Z_k\}$ beziehungsweise $Z = \{Z_1, Z_2, \dots, Z_{k-1}\}$ die Zusammenhangskomponenten, die durch $G - C$ entstehen. Die

zugehörigen augmentierten Komponenten seien A_1, A_2, \dots, A_k bzw. A_1, A_2, \dots, A_{k-1} . Betrachtet wird eine beliebige dieser augmentierten Komponenten A_i . Der Definition der augmentierten Komponenten nach finden sich bereits alle Knoten von A_i in G wieder. Weiterhin enthält G mindestens alle Kanten in $A_i - C$ sowie die verbindenden Kanten zwischen A_i und C . Jedoch bilden in A_i die Knoten von C eine Clique, es existieren also ggf. Kanten zwischen den Knoten von C in A_i , die es nicht in G gibt. Es bleibt zu zeigen, dass die Kanten, die für diese Clique in A_i nötig sind, durch Kantenkontraktionen in G erzeugt werden können. Dadurch, dass G k -zusammenhängend ist, besitzt jede Zusammenhangskomponente von $G - C$ Kanten zu c_1, c_2, \dots, c_k . Würde eine Kante zu einem Knoten c_j mit $1 \leq j \leq k$ fehlen, wäre ein $k - 1$ -Separator bestehend aus $C - c_j$ möglich, was im Widerspruch zu dem k -Zusammenhang stehen würde. Das Theorem unterscheidet nun zwei Fälle, um die fehlenden Kanten bereitstellen zu können:

1. Es existieren k Zusammenhangskomponenten. Wird A_i betrachtet, kommen die Knoten in $Z - Z_i$ in Frage, um durch Kantenkontraktionen die fehlenden Kanten für die Clique von C in A_i zu erzeugen. Um die Kanten von C in A_i in G zu erzeugen, kann zunächst der Pfad, der c_1 mit Z_1 verbindet, kontrahiert werden. Anschließend ist c_1 mit allen Knoten in C verbunden. Dies kann analog für alle Knoten in C und den entsprechenden Zusammenhangskomponenten durchgeführt werden außer für c_i , da A_i der gesuchte Minor ist. Allerdings ist c_i aufgrund des k -Zusammenhangs mit allen anderen Zusammenhangskomponenten verbunden und nach den beschriebenen Kontraktionen bildet C eine Clique.
2. Es existieren $k - 1$ Komponenten, aber mindestens zwei bestehen aus mehr als einem Knoten. Analog zum vorherigen Fall können die Pfade zwischen den Knoten von C und den Zusammenhangskomponenten A kontrahiert werden. Es fehlt jedoch ein Pfad, da eine Zusammenhangskomponente weniger vorliegt. Es gibt mindestens eine Zusammenhangskomponente aus $Z - Z_i$, die aus zwei oder mehr Knoten besteht. Da der Graph k -zusammenhängend ist, sind mindestens zwei dieser Knoten mit allen in C verbunden, sodass sie durch Kontraktionen mit zwei unterschiedlichen Knoten aus C genutzt werden können um die gesuchte Clique zu erzeugen. \square

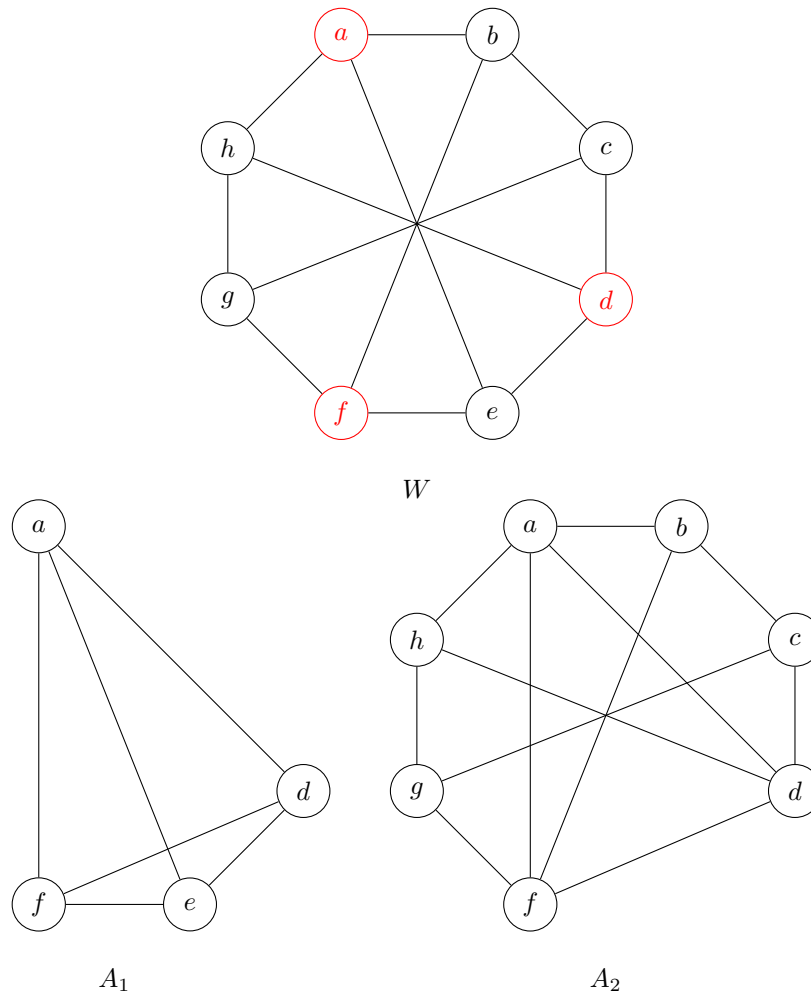


Abbildung 3.1: Gegenbeispiel zu Theorem 3.1.1 für $k = 3$ mit $C = \{a, d, f\}$ in W , wodurch die $k - 1$ Komponenten A_1 und A_2 entstehen, von denen nur eine aus mehr als einem Knoten besteht. Die Komponente A_1 ist zwar ein gültiger Minor, da sie etwa durch die Kontraktionen der Kanten (a, b) , (a, h) , (f, g) und (c, d) aus W erzeugt werden kann. A_2 dagegen kann nicht durch Kontraktionen aus W erzeugt werden - wird beispielsweise die Kante (d, e) in W kontrahiert, fehlt die Kante (a, f) in A_2 . Analog kann e mit keiner seiner inzidenten Kanten kontrahiert werden, um einen zu A_2 isomorphen Graph zu erhalten.

Als nächstes stellen Kezdy und McGuinness fest, dass im Fall eines $(3, 3)$ -Separators der Graph in augmentierte Komponenten zerlegt werden kann:

3.1.2 Theorem. *Sei G ein 3-zusammenhängender Graph mit einem $(3, 3)$ -Separator C . G hat einen K_5 -Minor genau dann, wenn eine der durch C definierten augmentierten Komponenten einen K_5 -Minor enthält.*

Beweis. Zunächst kann festgestellt werden, dass falls eine der augmentierten Komponenten einen K_5 -Minor enthält, dieser laut Theorem 3.1.1 auch ein Minor von G ist. Es bleibt zu zeigen, dass sich ein K_5 -Minor nicht auf zwei augmentierte Komponenten er-

streckt, sondern sich ausschließlich in einer befindet. Angenommen es gilt $K_5 \prec_M G$ und zwei der Branch-Sets, die den K_5 -Minor bilden, befinden sich jeweils vollständig in unterschiedlichen Zusammenhangskomponenten. In diesem Fall wäre C ein 3-Separator in dem gefundenen Minor, was im Widerspruch zu dem 4-Zusammenhang des K_5 steht. \square

Das zentrale Theorem ist darauf zurückzuführen, dass jeder Graph ohne K_5 -Minor durch Cliques-Summen von Teilgraphen, die planar oder isomorph zu W sind, gebildet werden kann. [11] Eine genauere Betrachtung findet in Kapitel 4 statt.

3.1.3 Theorem. *Sei G ein 3-zusammenhängender Graph, der ein $K_{3,3}$ -Subdivision S enthält, dessen Branch-Ends in die rote Knotenmenge $R = \{a, b, c\}$ und blaue $B = \{x, y, z\}$ unterteilt sind. Eine der folgenden Bedingungen trifft auf G zu:*

1. G enthält einen K_5 -Minor.
2. G ist isomorph zu W .
3. $\{a, b, c\}$ bilden einen 3-Separator, sodass $\{x, y, z\}$ in separaten Komponenten liegen.
4. $\{x, y, z\}$ bilden einen 3-Separator, sodass $\{a, b, c\}$ in separaten Komponenten liegen.

Durch die Theoreme 3.1.1 und 3.1.2 wurde gezeigt, dass der Graph in den Fällen 3 und 4 in augmentierte Komponenten zerlegt und darauf ein Planaritätstest ausgeführt werden kann. Anschließend stellen die Autoren einige Lemmata auf, mit denen untersucht wird, ob S einen K_5 -Minor enthält - also ob Bedingung 1 zutrifft.

3.1.4 Lemma. *Sei G ein 3-zusammenhängender Graph und S ein $K_{3,3}$ -Subdivision in G . Enthält G drei Pfade von einem Knoten $w \in G - S$ zu drei Knoten in S , die nicht alle im selben Branch-Fan liegen, enthält G einen K_5 -Minor.*

Beweis. Seien t, u, v die drei Endpunkte der Pfade in S . Mindestens einer von ihnen ist ein innerer Knoten, da sonst alle im selben Branch-Fan liegen würden. Sei o. B. d. A. t ein solcher innerer Knoten auf dem Pfad $P(a, x)$. Folglich können u und v nicht beide in $F(a)$ oder $F(x)$ liegen, sonst lägen alle drei im gleichen Branch-Fan.

1. u und v sind nicht im gleichen Branch-Fan wie t . Dann müssen u und v ebenfalls innere Knoten sein, im Beispiel auf den Pfaden $P(y, b)$ bzw. $P(z, c)$. Es kann ein M -Minor durch folgende Kontraktionen erzeugt werden: u mit einem der roten und v mit einem der blauen Knoten (analog u mit blau und v mit rot) sowie $P(w, t)$.
2. u oder v liegen auf $P(a, x)$. Sei o. B. d. A. $u \in P(a, x)$. Da t ebenfalls in diesem Pfad liegt, gilt $\{t, u\} \in F(a) \cup F(x)$, sodass v nicht in diesen beiden Branch-Fans liegen kann. Es können t und v getauscht werden, sodass eine Reduktion auf Fall 1 erreicht wird.

3. Entweder u oder v liegen im gleichen Branch-Fan wie t . Sei o.B.d.A. $u \in F(x) - P(a, x)$, im Beispiel auf dem Pfad $P(b, x)$. Es gilt $\{t, u\} \in F(x)$, weshalb v in einem anderen Branch Fan sein muss. Da alle roten Knoten in $F(x)$ liegen, gilt konkreter $v \in (F(y) \cup F(z)) - \{a, b, c\}$. Es können $P(b, u)$ kontrahiert werden sowie je nach Fall entweder $P(v, y)$ oder $P(v, z)$. Wird $P(w, t)$ ebenfalls kontrahiert, entsteht erneut ein M -Minor. \square

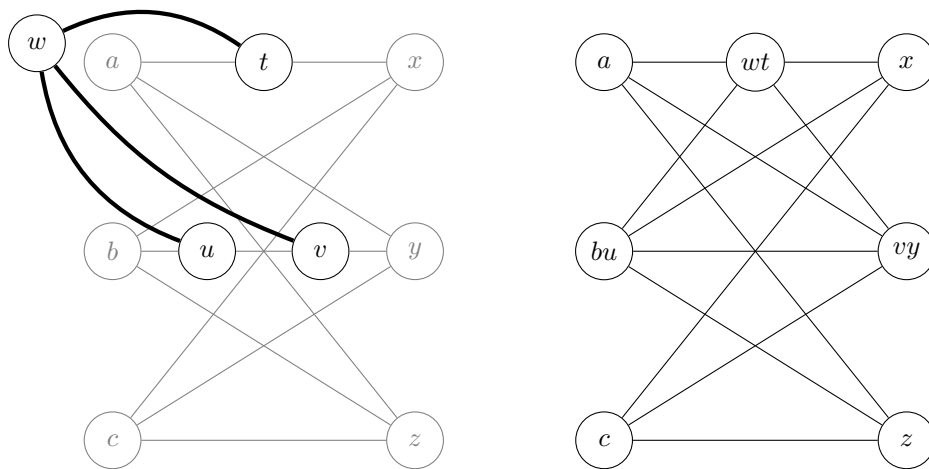


Abbildung 3.2: Beispiel zum ersten Fall von Lemma 3.1.4.

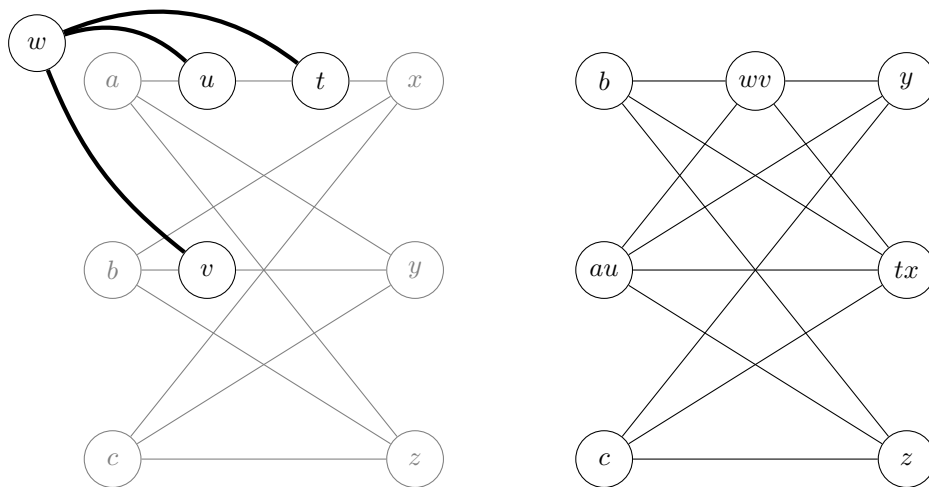


Abbildung 3.3: Beispiel zum zweiten Fall von Lemma 3.1.4.

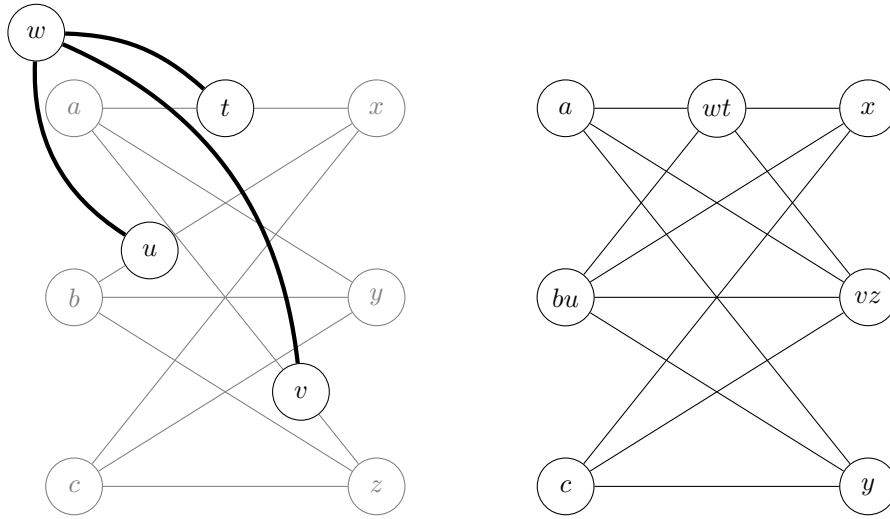


Abbildung 3.4: Beispiel zum dritten Fall von Lemma 3.1.4.

3.1.5 Lemma. Sei G ein 3-zusammenhängender Graph und S ein $K_{3,3}$ -Subdivision in G . Betrachtet wird ein Pfad außerhalb von S , der zwei Knoten in einem roten Branch-Fan verbindet, welche jedoch nicht beide auf dem gleichen Pfad in S liegen. Analog dazu wird ein Pfad außerhalb von S gesucht, der zwei Knoten in einem blauen Branch-Fan verbindet, ohne dass diese beide auf dem gleichen Pfad in S liegen. Existieren diese beiden Pfade in G , dann enthält G einen K_5 -Minor.

Beweis. Sei P_1 der Pfad, der zwei Knoten in einem roten Branch-Fan verbindet und P_2 der, der zwei in einem blauen Branch-Fan verbindet. O.B.d.A. hat P_1 Endpunkte in $F(a)$ und P_2 in $F(x)$. Da laut Bedingung die Endpunkte nicht in einem einzelnen Pfad von S liegen, kann a kein Endpunkt von P_1 und x kein Endpunkt von P_2 sein. Es ergeben sich zwei Fälle:

1. Die beiden Pfade haben keine gemeinsamen Knoten. Da P_1 beide Endpunkte in $F(a)$ hat, liegen diese beiden Endpunkte in zwei unterschiedlichen blauen Branch-Fans. Entsprechend sind die Endpunkte von P_2 in unterschiedlichen roten Branch-Fans. Werden die Endpunkte von P_1 je mit den beiden blauen und die von P_2 mit den beiden roten Knoten von S kontrahiert, entsteht ein K_5 -Minor. Abbildung 3.5 skizziert den Fall beispielhaft.
2. Die beiden Pfade haben einen gemeinsamen Knoten w . Liegt dieser gemeinsame Knoten außerhalb von S , kann Lemma 3.1.4 angewendet werden, da die Endpunkte der Pfade nicht alle im gleichen Branch-Fan liegen. Liegt w innerhalb von S , ist er ein Endpunkt von P_1 und P_2 und muss auf dem Pfad $P(a, x)$ liegen, da dieser der einzige gemeinsame Pfad ist, siehe links in 3.6. Sei $P_1 = P(w, u)$ und $P_2 = P(w, v)$. Da u nicht in $F(x)$ liegt und v nicht in $F(a)$, gibt es einen Pfad von u zu einem

blauen Knoten und von v zu einem roten Knoten, die sich nicht kreuzen und daher kontrahiert werden können. Durch die Kontraktion dieser beiden Pfade entsteht, wie in Abbildung 3.6 rechts zu sehen, ein M -Minor. \square

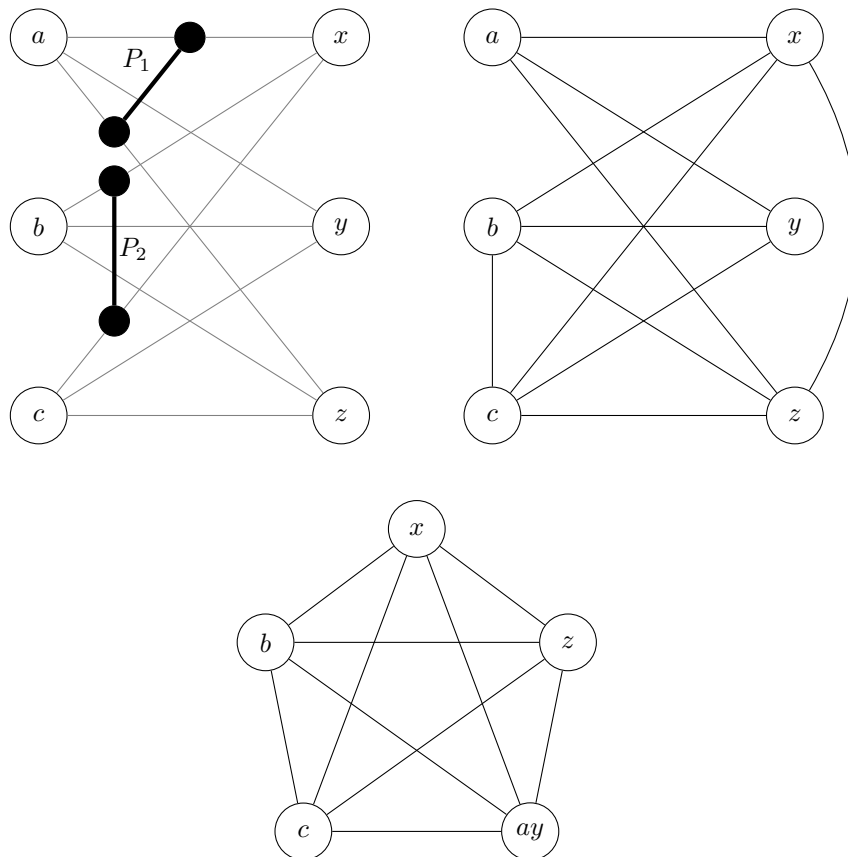


Abbildung 3.5: Beispiel zum ersten Fall von Lemma 3.1.5. Links oben ist S mit zwei zusätzlichen Pfaden aus G abgebildet. P_1 hat beide Endpunkte in $F(a)$ sowie den blauen Branch-Fans $F(x)$ und $F(y)$, mit denen die Endpunkte kontrahiert werden. P_2 hat seine Endpunkte in den roten Branch-Fans $F(b)$ und $F(c)$, mit denen sie kontrahiert werden. Rechts oben ist der dadurch entstehende Minor abgebildet und unten wird der enthaltene K_5 -Minor gezeigt.

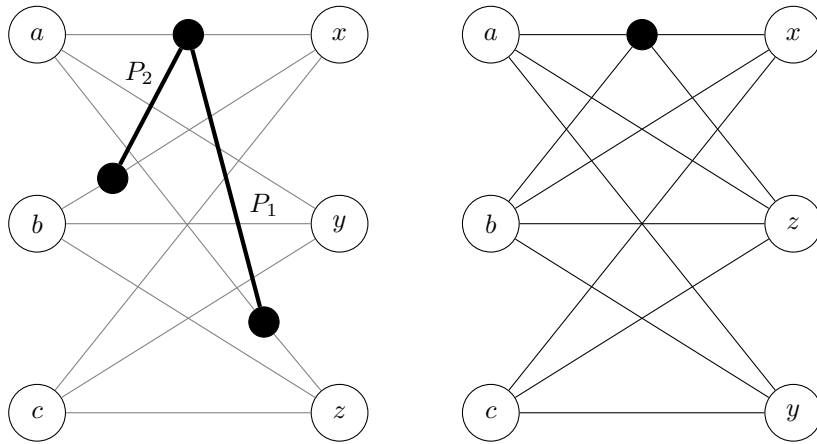


Abbildung 3.6: Beispiel zum zweiten Fall von Lemma 3.1.5. Rechts ist der enthaltene M -Minor abgebildet.

3.1.6 Lemma. Sei G ein 3-zusammenhängender Graph und S ein $K_{3,3}$ -Subdivision in G . Betrachtet wird ein Pfad außerhalb von S , der zwei innere Knoten paralleler Pfade in S verbindet sowie ein Pfad außerhalb von S , dessen Endpunkte nicht beide im gleichen Pfad von S liegen. Bestehen die Endpunkte der beiden Pfade aus mindestens drei unterschiedlichen Knoten in S , enthält G einen K_5 -Minor.

3.1.7 Lemma. Sei G ein 3-zusammenhängender Graph und S ein $K_{3,3}$ -Subdivision in G mit den roten Knoten $R = \{a, b, c\}$ und den blauen Knoten $B = \{x, y, z\}$. Bilden weder R , noch B einen $(3, 3)$ -Separator, enthält G einen K_5 -Minor.

Beweis. Falls R und B keinen $(3, 3)$ -Separator bilden, ist sowohl der Graph $G - R$ als auch $G - B$ zusammenhängend. Beide liegen außerhalb von S . Die Endpunkte von P_1 seien u_1 und v_1 , die von P_2 seien u_2 und v_2 . u_1 und v_1 besitzen jeweils einen Pfad in S zu einem der roten Knoten. Folglich gibt es einen dritten roten Knoten, der keinen solchen Pfad besitzt - u_2 wird so gewählt, dass er in dem Branch-Fan dieses Knotens liegt. Demnach sind u_1 , v_1 und u_2 unterschiedliche Knoten. Anschließend kann je nach vorliegendem Fall die Aussage auf eines der vorherigen Lemmata reduziert werden:

1. P_1 oder P_2 verbindet zwei parallele Pfade in S . In dem Fall kann Lemma 3.1.6 angewendet werden und G enthält einen K_5 -Minor.
2. Die Endpunkte von P_1 liegen in einem einzelnen roten Branch-Fan - analog liegen die von P_2 in einem blauen. Nach Lemma 3.1.5 enthält G einen K_5 -Minor. \square

Als nächstes folgt der Beweis zu 3.1.3.

Beweis. Gezeigt wird, dass falls S keinen $(3, 3)$ -Separator bildet, G entweder einen K_5 -Minor enthält oder isomorph zu W ist. Falls kein K_5 -Minor enthalten ist, gilt nach Lemma

3.1.7, dass $G - R$ oder $G - B$ nicht zusammenhängend ist. Demnach ist B ein 3-Separator, der den Graph teilt, aber die Knoten aus R liegen nicht alle in unterschiedlichen Zusammenhangskomponenten. Deshalb muss es außerhalb von S mindestens einen Pfad P_1 geben, der zwei der roten Knoten in $G - B$ verbindet. Analog gibt es einen Pfad P_2 , der zwei blaue Knoten in $G - R$ verbindet. Da P_1 zwei rote Branch-Fans verbindet, liegen seine Endpunkte in zwei verschiedenen Pfaden von S . Gleiches gilt für die Endpunkte von P_2 . Liegen die Endpunkte von P_1 beide in einem einzelnen blauen Branch-Fan und die von P_2 in einem einzelnen roten, dann enthält G laut Lemma 3.1.5 einen K_5 -Minor. Liegen die Endpunkte von P_1 in parallelen Pfaden von S , enthält G laut Lemma 3.1.6 einen K_5 -Minor, da die Endpunkte von P_2 nicht auf einem Pfad von S liegen (analog falls P_2 auf parallelen Pfaden liegt). Übrig bleibt die Möglichkeit, dass die Endpunkte der beiden Pfade paarweise identisch sind, siehe rechts in Abbildung 2.6. Dann ist G ein Subdivision zu W und enthält laut Lemma 3.1.3 keinen K_5 -Minor bei Isomorphie zu W . \square

3.2 Sequenzieller Algorithmus zum Finden von K_5 -Minoren

Da die Theoreme größtenteils auf 3-zusammenhängenden Graphen arbeiten, muss der Eingabegraph ggf. zunächst angepasst werden, bevor der Planaritätstests angewendet werden kann. Ist der Graph 1-zusammenhängend, gibt es einen Knoten, der einen $(1, j)$ -Separator für $j \geq 2$ bildet. Genauso müssen zwei Knoten existieren, die einen $(2, j)$ -Separator bilden, falls der Graph 2-zusammenhängend ist. In beiden Fällen kann der Separator benutzt werden, um den Graph in j augmentierte Komponenten zu zerlegen. Anschließend kann der 3-Zusammenhang der einzelnen Komponenten rekursiv geprüft werden. Sind die Komponenten alle 3-zusammenhängend, kann auf jede ein Planaritätstest angewendet werden. Kezdy und McGuinness verwenden den Williamson-Algorithmus, [12] welcher in Linearzeit für einen Graph einen K_5 - bzw. $K_{3,3}$ -Subdivision ausgibt oder feststellt, dass der Graph planar ist. In der Implementierung wird stattdessen der Planaritätstest von Boyer und Myrvold [7] verwendet, der bereits in OGDF existiert. Ergibt der Planaritätstest, dass eine Komponente planar ist, wird sie nicht weiter beachtet. Enthält sie einen K_5 -Minor, kann der Algorithmus stoppen und diesen ausgeben. Wird ein $K_{3,3}$ -Minor gefunden, wird geprüft, welcher der vier Fälle aus Theorem 3.1.3 zutrifft. Bei Isomorphie zu W wird die Komponente nicht weiter beachtet. Ist der $K_{3,3}$ -Minor ein $(3, 3)$ -Separator in der untersuchten Komponente, kann sie in weitere augmentierte Komponenten zerlegt und der Algorithmus rekursiv darauf angewendet werden. Andernfalls müssen genügend Pfade in der Komponente existieren, sodass der $K_{3,3}$ -Minor auch einen K_5 -Minor bildet und der Algorithmus ihn ausgeben und anhalten kann. Diese Schritte werden solange wiederholt, bis alle augmentierten Komponenten planar, isomorph zu W sind oder ein K_5 -Minor gefunden wurde.

Für einen schlichten Graph G sind die einzelnen Schritte zusammengefasst:

1. Solange es 1-Separatoren gibt: Erzeuge augmentierte Komponenten mit diesen Separatoren.
2. Solange es 2-Separatoren gibt: Erzeuge augmentierte Komponenten mit diesen Separatoren.

Für alle j Zusammenhangskomponenten Z_i von G :

3. Planaritätstest:
 - (a) Z_i ist planar: Falls $i < j$ handle Z_{i+1} in Schritt 3, sonst stopp.
 - (b) Z_i enthält ein K_5 -Subdivision: Stopp
 - (c) Z_i enthält ein $K_{3,3}$ -Subdivision: Weiter zu Schritt 4
4. Z_i ist isomorph zu W : Falls $i < j$ handle Z_{i+1} in Schritt 3, sonst stopp. Z_i ist nicht isomorph zu W : Weiter zu Schritt 5
5. Seien $R = \{a, b, c\}$ die roten und $B = \{x, y, z\}$ die blauen Branch-Ends des $K_{3,3}$ -Subdivision. Bildet R einen $(3, 3)$ -Separator, sodass alle Knoten aus B in unterschiedlichen Zusammenhangskomponenten sind? Dann erzeuge die augmentierten Komponenten mit R als Separator und gehe zu Schritt 3. (Analog für B als Separator)
6. Z_i enthält nach Theorem 3.1.3 einen K_5 -Minor: Stopp.

Kapitel 4

Wagner Struktur

In diesem Kapitel soll Theorem 3.1.3 genauer betrachtet werden. Zunächst werden einige Baumstrukturen eingeführt, die den 3-Zusammenhang herstellen können. Anschließend werden $(3,3)$ -Separatoren behandelt und das Theorem von Wagner, das in Theorem 3.1.3 benutzt wurde, genauer betrachtet.

4.0.1 2-Zusammenhang

Block-Cut-Tree: Als *Block* eines Graphen G wird jeder seiner maximalen Teilgraphen bezeichnet, der keinen 1-Separator enthält - als 2-zusammenhängend ist. Je zwei Blöcke haben maximal einen gemeinsamen Knoten, der einen 1-Separator in G bildet. [6] Ein Block-Cut Tree ist ein Baum, der für jeden Block von G einen Knoten besitzt und für jeden Separator eine Kante. Als Beispiel wird in Abbildung 4.1 ein Graph mit dazugehörigem Block-Cut Tree in Abbildung 4.2 gezeigt. Die Knoten des Block-Cut Tree sind als durchgezogene Linien angegeben, in jedem dieser Knoten findet sich ein 2-zusammenhängender Teilgraph aus G_1 .

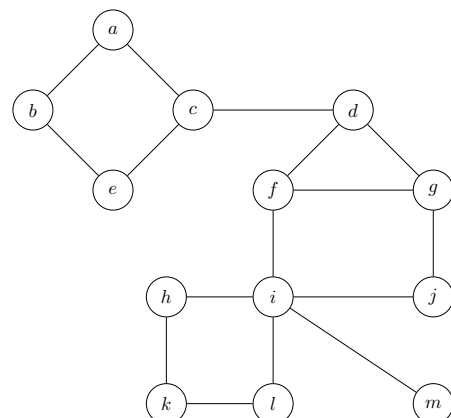


Abbildung 4.1: Ein nicht 2-zusammenhängender Graph G_1 .

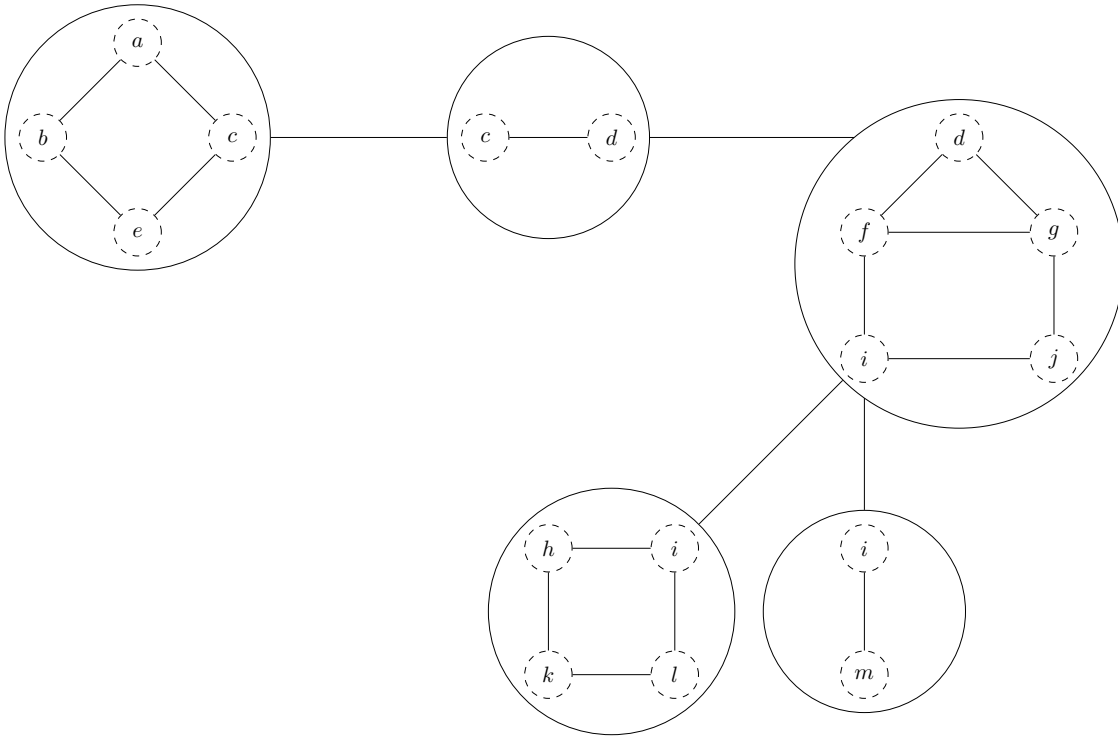


Abbildung 4.2: Der Block-Cut Tree zu G aus Abbildung 4.1.

In [10] stellt Li eine Erweiterung zum Block-Cut Tree vor, die als *1-Block-Tree* bezeichnet wird, da neben den Blöcken auch 1-Separatoren enthalten sind.

1-Block-Tree: Seien für einen Graph G die Knoten seines Block-Cut Trees T_C als Blockknoten bezeichnet. Dann enthält der 1-Block-Tree T_1 von G statt jeder Kante, die zwei Blockknoten u und v in T_C verbindet, die Kanten (u, c) und (c, v) , wobei c als Cliquenknoten bezeichnet wird und den 1-Separator beinhaltet, den die zu u und v zugehörigen Teilgraphen in G als gemeinsamen Knoten besitzen. Ein Beispiel findet sich in Abbildung 4.3. Es kann beobachtet werden, dass die in Blockknoten enthaltenen Teilgraphen identisch zu den augmentierten Komponenten sind, die im Algorithmus von Kezdy und McGuinness in Schritt 1 durch 1-Separatoren gebildet werden.

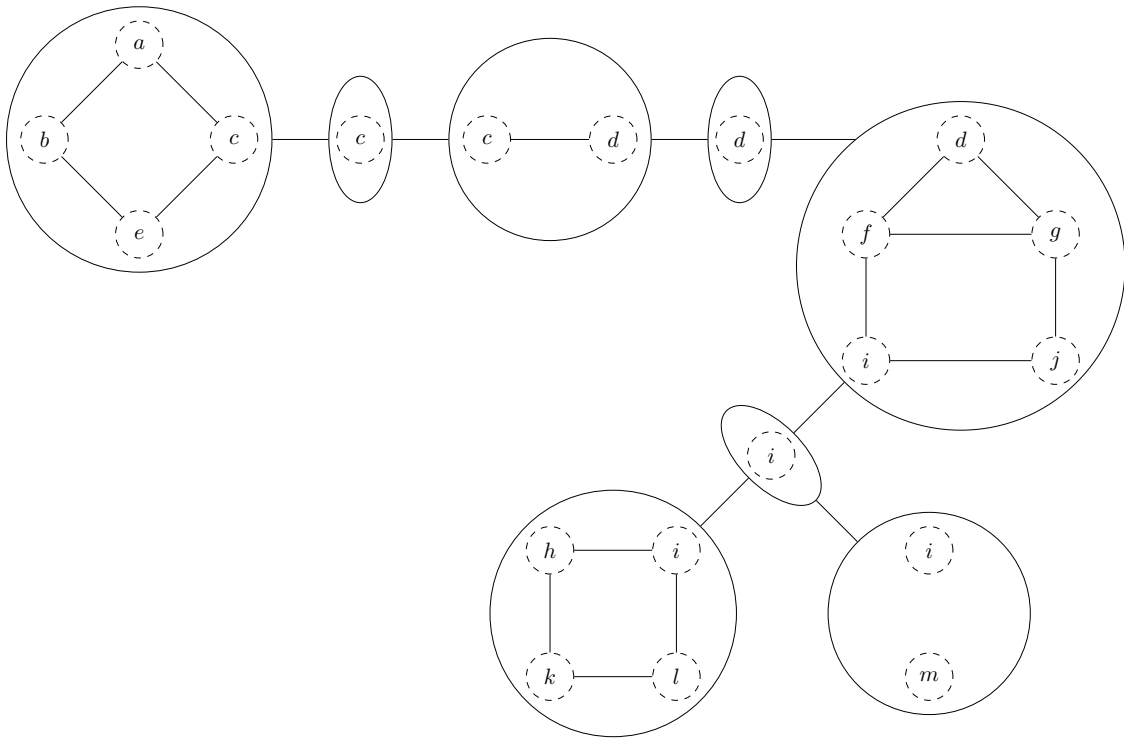


Abbildung 4.3: Der 1-Block-Tree zu G aus Abbildung 4.2. Die Cliquenknoten sind ovalförmig dargestellt.

4.0.2 3-Zusammenhang

Nachdem durch Block-Cut Trees bzw. 1-Block-Trees der 2-Zusammenhang in allen Blockknoten hergestellt wurde, können die darin enthaltenen Teilgraphen jeweils als Eingabe für einen SPQR-Baum genutzt werden, um 3-zusammenhängende Komponenten zu erzeugen. Die folgende Definition dazu ist [3] entnommen.

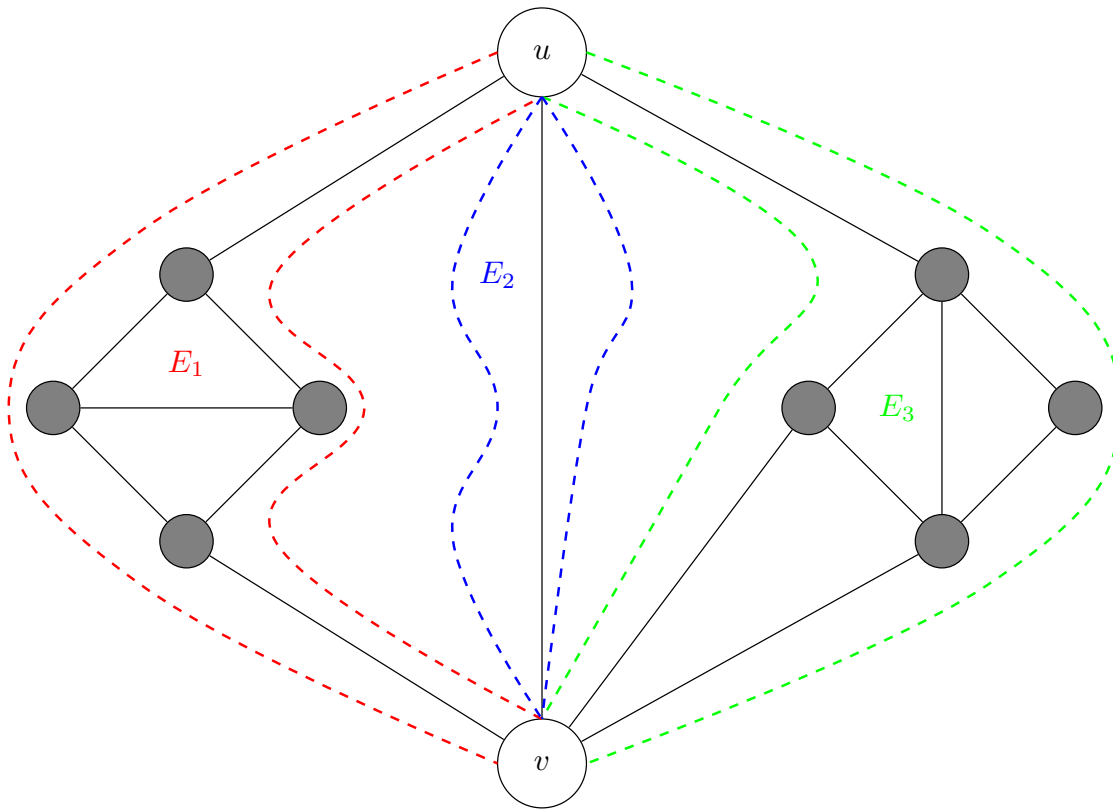


Abbildung 4.4: Unterteilung der Kanten in drei Mengen E_1 , E_2 und E_3 anhand des Knotenpaares $\{u, v\}$.

SPQR-Baum: Sei $G = (V, E)$ ein 2-zusammenhängender Graph, betrachtet werden alle Knotenpaare $\{u, v\}$ in G , die entweder benachbart sind oder einen 2-Separatoren bilden. Dieses Knotenpaar teilt die Kanten in die Mengen E_1, \dots, E_k , sodass eine Menge E_i alle Pfade enthält, die u oder v höchstens als Endpunkte haben. In Abb. 4.4 ist ein 2-zusammenhängender Graph skizziert, in dem $\{u, v\}$ eingezeichnet ist, sowie farblich hervorgehoben die dadurch entstehende Unterteilung in Kantenmengen. Jeder Knoten des SPQR-Baums von G wird durch diese Kantenmengen festgelegt, und je nach Eigenschaft entweder mit S , P , Q oder R markiert:

- **Q:** Tritt der Randfall auf, dass G nur eine einzige Kante (u, v) besitzt, dann enthält der SPQR-Baum einen einzelnen Q-Knoten, der auf G verweist.
- **P:** Entstehen durch das Knotenpaar die Kantenmengen E_1, \dots, E_k mit $k \geq 3$, dann wird ein P-Knoten hinzugefügt, der das Knotenpaar enthält sowie k viele parallele Kanten zwischen diesem.
- **S:** Andernfalls teilen u und v die Kanten in zwei Mengen, sodass es zwei unterschiedliche Pfade gibt, die das Knotenpaar verbinden. Der hinzugefügte S-Knoten enthält das Knotenpaar sowie diese beiden Pfade.

- **R:** Tritt keiner der obigen Fälle ein, dann wird in je einem R-Knoten der maximale 3-zusammenhängende Teilgraph gespeichert, der $\{u, v\}$ enthält. Zusätzlich wird die Kante (u, v) hinzugefügt, falls sie noch nicht vorhanden ist.

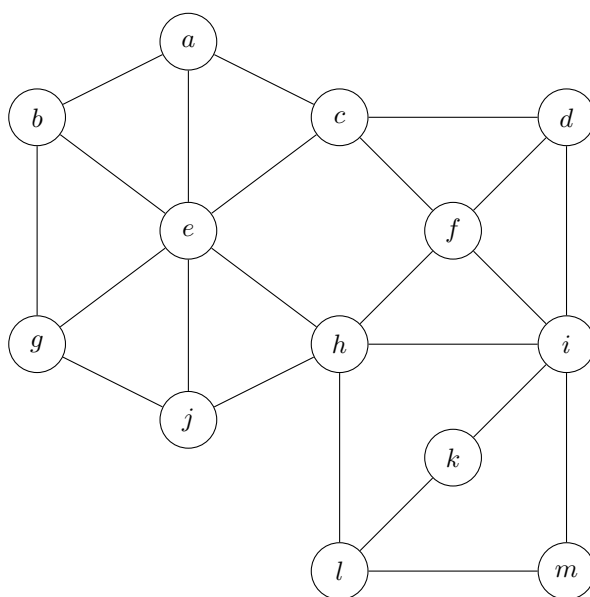


Abbildung 4.5: Ein 2-zusammenhängender Graph G_2 .

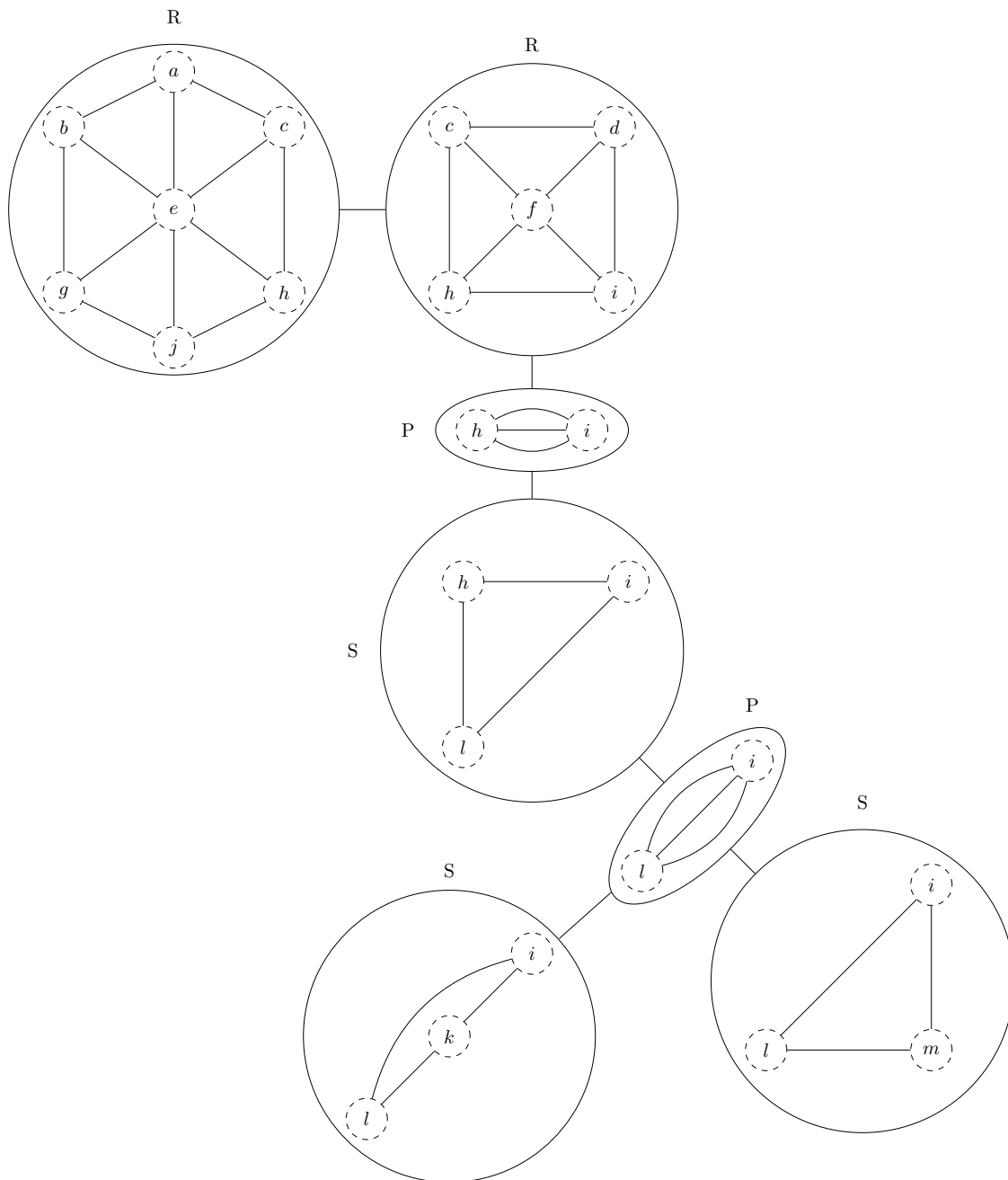


Abbildung 4.6: SPQR Baum zum Graph G_2 aus Abb. 4.5.

Zu dem Graph in Abb. 4.5 ist in Abb. 4.6 der zugehörige SPQR Baum skizziert. Besonders interessant sind die R-Knoten, die 3-zusammenhängende Minoren von G_2 sind. Zwar gilt das auch für die drei S-Knoten, allerdings sind Kreise generell nicht 3-zusammenhängend, aber immer planar.

In [10] stellt Li analog zum 1-Block-Tree den 2-Block-Tree vor, der aus 3-zusammenhängende Komponenten (Blockknoten) und 2-Separatoren (Cliquesknoten) besteht.

2-Block-Tree: Sei $G = (V_G, E_G)$ ein 2-zusammenhängender, K_5 -Minor freier Graph und $T_2 = (V_T, E_T)$ der 2-Block-Tree. Bilden die Knoten $u, v \in V_G$ einen $(2, j)$ -Separator in G für $j \geq 2$, dann seien erneut Z_1, \dots, Z_j die Zusammenhangskomponenten von $G - \{u, v\}$. Analog zum 1-Block-Tree wird ein Cliquenknoten für den Separator angelegt, der alle Blockknoten bestehend aus $Z_i \cup \{u, v\}$ als Nachbarn besitzt. Außerdem wird die Kante (u, v) sowohl in den Block- als auch in dem Cliquenknoten hinzugefügt, falls sie nicht vorhanden ist. Ist G 3-zusammenhängend, besitzt T_2 einen einzelnen Blockknoten, der G vollständig enthält. Ein 2-Block-Tree zu G_2 aus Abb. 4.5 ist in Abb. 4.7 zu sehen. Es ist zu beobachten, dass Knoten wie etwa h oder i Teil mehrerer Separatoren sein können und somit nicht nur in mehreren Graph-, sondern auch in mehreren Cliquenknoten enthalten sind. Allerdings sind keine zwei Knoten im 2-Block-Tree identisch.

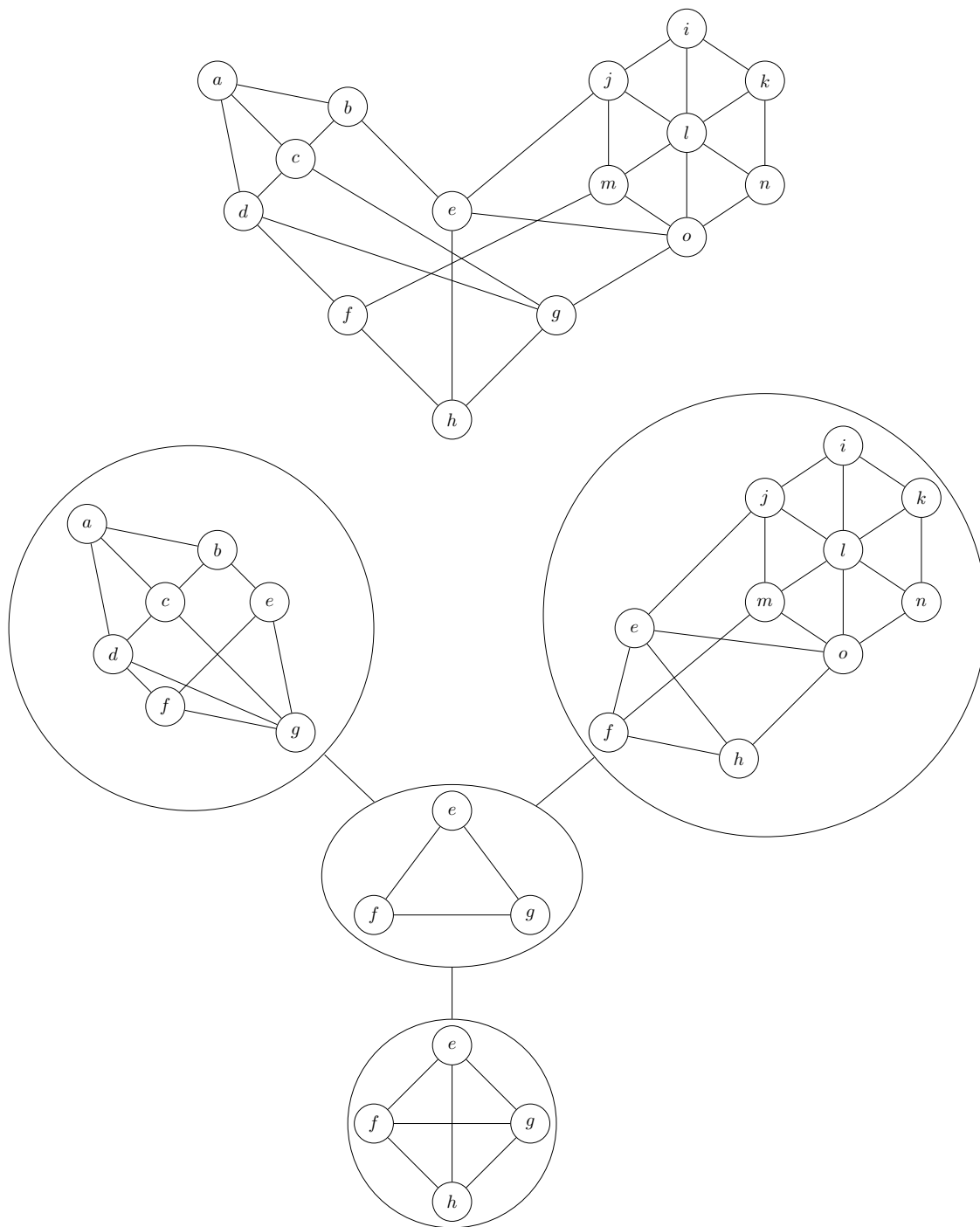


Abbildung 4.8: Ein 3-zusammenhängender Graph mit dazugehörigem (3,3)-Block-Tree. Der Separator wird durch $\{e, f, g\}$ gebildet. Es ist zu sehen, dass der Eingabegraph nicht planar ist, da ein $K_{3,3}$ -Minor enthalten ist. Die Blockknoten enthalten jedoch alle planare Minoren des ursprünglichen Graphs.

4.0.4 Wagner-Struktur

Sei G ein schlichter Graph ohne Mehrfachkanten, dessen Zusammenhangskomponenten 3-zusammenhängend sind. Dann ist die *Wagner-Struktur* von G ein Wald von Bäumen, wobei jeder Baum der $(3, 3)$ -Block-Tree einer Zusammenhangskomponente ist.

Um nun aus einem schlichten nicht zusammenhängenden Graph G ohne K_5 -Minor eine Wagner-Struktur zu erzeugen, wird zunächst für jede Zusammenhangskomponente ein 1-Block-Tree T_1 angelegt. Jeder Blockknoten $t \in T_1$ beinhaltet einen 2-zusammenhängenden Teilgraph G' von G , welcher als Eingabe für einen 2-Block-Tree T_2 dient. Ist G bereits 2- bzw. 3-zusammenhängend, kann der entsprechende Block-Tree übersprungen und G direkt als Eingabe für den 2- bzw. $(3, 3)$ -Block-Tree verwendet werden. Anschließend kann jeder 3-zusammenhängende Teilgraph G'' , der mit einem Blockknoten von T_2 assoziiert wird, als Eingabe für einen $(3, 3)$ -Block-Tree verwendet werden. Dadurch entsteht ein Baum, der zeigt, wie nach Theorem ?? der Graph G aus planaren Graphen und Kopien von W (Blockknoten) durch Cliques-Summen (Cliquen-knoten) erzeugt werden kann. Zwar ist der Eingabegraph nicht immer eindeutig aus der Wagner-Struktur wiederherstellbar, allerdings ist die relevante Eigenschaft einer solchen Struktur vielmehr, dass sie nur für K_5 -Minor freie Graphen existiert. Scheitert der Aufbau einer Wagner-Struktur, ist der Graph nicht K_5 -Minor frei.

4.1 Algorithmus von Kezdy und McGuinness als Wagner-Struktur

Im Folgenden wird die Wagner-Struktur von Reed und Li mit dem Algorithmus von Kezdy und McGuinness verglichen und dieser so modifiziert, dass er sie erzeugen kann. Anschließend werden augmentierte Komponenten für 1- und 2-Separatoren durch Block-Cut Trees und SPQR-Bäume erzeugt sowie ein zertifizierender Algorithmus beschrieben.

4.1.1 Block-Trees

Behauptung: Wird der Graph in augmentierte Komponenten aufgeteilt, sind diese genau dann isomorph zu mindestens einem Blockknoten der verschiedenen Block-Trees, wenn alle Separatoren gleich gewählt wurden. Im Algorithmus von Kezdy und McGuinness wird der Graph in augmentierte Komponenten aufgeteilt, wenn ein $(1, 2)$ -, $(2, 2)$ - oder $(3, 3)$ -Separator gefunden wird.

Sei C ein solcher (i, j) -Separator, der einen zusammenhängenden Graph G in genau j Zusammenhangskomponenten Z_1, \dots, Z_j aufteilt, wenn die i Knoten von C aus G entfernt werden.

Angenommen, es gäbe eine augmentierte Komponenten G_a , zu der es im zugehörigen $(3, 3)$ -Block-Tree keinen Blockknoten t mit assoziiertem Graph G_t gibt, sodass G_a und G_t

isomorph sind. Dann gibt es einen Separator C_a in G , aus dem G_a entstanden ist. Da es kein t mit einem zu G_a isomorphen G_t gibt, wurden keine Knoten im Block-Tree mit C_a erzeugt, sodass es ebenfalls keinen Cliquenknoten mit C_a gibt. Dann wurde mindestens ein Separator C'_a für den 1-Block-Tree gewählt, wodurch die Knoten von C_a so in Blockknoten enthalten sind, dass sie dort keinen Separator bilden, was der gleichen Wahl von Separatoren widerspricht. Oder C_a ist vollständig in einem Blockknoten enthalten und bildet dort einen Separator, was entweder dem 2-Zusammenhang oder der Planarität bzw. Isomorphie zu W des Blockknotens widerspricht.

Der Algorithmus von Kezdy und McGuinness erzeugt also zu Beginn alle Blockknoten eines 1-Block-Trees. Zusätzlich können die Knoten der gefundenen Separatoren gespeichert werden, um ebenfalls die Cliquenknoten zu erhalten, sodass der Algorithmus einen 1-Block-Tree erzeugt. Im zweiten Schritt werden die augmentierten Komponenten bzw. Blockknoten des 1-Block-Tree in 3-zusammenhängende augmentierte Komponenten bzw. einen 2-Block-Tree überführt. Analog zum vorherigen Fall gibt es für jede augmentierte Komponente, die aus einem Blockknoten des 1-Block-Trees mit einem $(2, 2)$ -Separator gebildet wird, einen isomorphen Blockknoten im 2-Block-Tree, falls alle Separatoren gleich gewählt wurden. Erneut können aus den Separatorknoten Cliquen gebildet und in zusätzlichen Knoten gespeichert werden, um als Ausgabe des zweiten Schrittes einen 2-Block-Tree zu erzeugen.

Um die Blockknoten des 2-Block-Trees in je einen $(3, 3)$ -Block-Tree zu verwandeln, müssen nicht nur entsprechende Separatoren gefunden und augmentierte Komponente gebildet, sondern auch garantiert werden, dass die Blockknoten der $(3, 3)$ -Block-Trees entweder planar oder isomorph zu W sind und in keinem ein K_5 -Minor enthalten ist. Hier kann das Theorem 3.1.3 in Kombination mit dem zuvor angewendeten Planaritätstest genutzt werden, um die Struktur aufzubauen. Sei dazu G_t der Graph, der zu einem beliebigen Blockknoten t im 2-Block-Tree gehört. G_t muss 3-zusammenhängend sein, da er als Graph vollständig in t vorkommt. Im Algorithmus von Kezdy und McGuinness wird G_t nun im dritten Schritt durch einen Planaritätstest überprüft. Wird ein K_5 -Minor gefunden, ist die Wagner-Struktur ungültig und der Algorithmus terminiert. Andernfalls kann G_t planar sein - dann wird im $(3, 3)$ -Block-Tree ein einzelner Blockknoten angelegt, der G_t enthält und als planar markiert ist. Letztlich kann der Planaritätstest ein $K_{3,3}$ -Subdivision S finden, sodass die Voraussetzungen für Theorem 3.1.3 gegeben sind. Im ersten Fall von Theorem 3.1.3 enthält G_t jedoch einen K_5 -Minor, da keine Knoten von S einen $(3, 3)$ -Separator bilden - der Beweis dazu findet sich in Lemma 3.1.7. Damit wäre die Wagner-Struktur erneut ungültig. Im zweiten Fall ist G_t isomorph zu W . Analog zum planaren Fall wird ein neuer Blockknoten im $(3, 3)$ -Block-Tree angelegt, der G_t enthält und entsprechend gekennzeichnet ist. Der dritte und vierte Fall treten ein, falls drei Knoten aus S einen $(3, 3)$ -Separator bilden. Kezdy und McGuinness erzeugen hier augmentierte Komponenten, die als Blockknoten des $(3, 3)$ -Block-Trees genutzt werden können. Zusätz-

lich müssen die drei Knoten des Separators als neuer Cliquenknoten eingefügt und mit den entstandenen Blockknoten verbunden werden. Der Algorithmus wird rekursiv auf jeden der Blockknoten angewendet, bis alle zugehörigen Teilgraphen planar oder isomorph zu W sind bzw. ein K_5 -Minor gefunden wurde. Somit ist das Ergebnis des modifizierten Algorithmus entweder eine Wagner-Struktur, die aus einem Wald von 1-Block-Trees mit Blockknoten, die aus 2-Block-Trees mit Blockknoten aus $(3, 3)$ -Block-Trees bestehen und die die planaren Minoren und Kopien von W im Eingabegraph zeigen. Oder es wird ein K_5 -Minor als Teilgraph des Eingabegraphen gefunden, sodass die Wagner-Struktur ungültig ist, die Frage, ob ein K_5 -Minor enthalten ist, jedoch in beiden Fällen eindeutig beantwortet werden kann.

4.1.2 Zertifizierender Algorithmus

Wie zuvor erwähnt erzeugen Block-Cut Trees alle 2-zusammenhängenden augmentierten Komponenten für 1-Separatoren. Anschließend kann für jede Komponente ein SPQR-Baum aufgebaut werden, sodass genau die 3-zusammenhängenden augmentierten Komponenten für 2-Separatoren gebildet werden, die aus mehr als drei Knoten bestehen. Letztlich kann der Algorithmus von Kezdy und McGuinness den $(3, 3)$ -Block-Tree mit Hilfe eines Planaritätstests aufbauen. Es ist zu beachten, dass in dem Fall eine Wagner-Struktur erzeugt wird, bei der alle Cliquenknoten mit einem oder zwei Knoten fehlen, da nur $(3, 3)$ -Separatoren durch den Algorithmus selbst behandelt werden. Das Ziel ist jedoch, einen K_5 -Minor zu finden oder ein Zertifikat zu liefern, dass keiner enthalten ist. Dafür sind folgende Strukturen in einem Graph relevant: Planare Teilgraphen, W -Subdivisionen, K_5 -Minoren und $(3, 3)$ -Separatoren - andere Separatoren werden lediglich benötigt, um den 3-Zusammenhang zu garantieren. Beispielsweise sind zwei planare Teilgraphen, die durch einen 2-Separator verbunden sind, ebenfalls planar. Das Zertifikat für einen Graph G besteht also entweder aus einem gefundenen K_5 -Minor in Form einer Liste von Kanten E_{K_5} - dann ist zu prüfen, ob dadurch im ursprünglichen Graph ebenfalls einer geformt wird. Jeder Knoten, der inzident zu vier unterschiedlichen Kanten aus E_{K_5} ist, fungiert als einer der fünf Knoten des K_5 . Genau diese fünf Knoten sind paarweise durch Pfade in G verbunden, die ebenfalls in E_{K_5} enthalten sein müssen und können beispielsweise durch eine Tiefensuche gefunden werden. Oder es wird eine Wagner-Struktur erzeugt, es muss jeder Knoten der Struktur wie folgt geprüft werden:

1. **Planar:** Sei G' der Teilgraph von G , der alle Knoten des planaren Minoren enthält. Dann ist G' ebenfalls planar, was z.B. durch einen Planaritätstest geprüft werden kann.
2. W : Analog zum Test eines K_5 -Minor kann die Kantenliste E_W genutzt werden, um acht Knoten zu finden, die inzident zu je drei Kanten aus E_W sind. Durch die

übrigen Kanten formen sie ein W -Subdivision. Der für den Algorithmus von Kezdy und McGuinness implementierte Isomorphietest kann wiederverwendet werden.

3. **(3,3)-Separator:** Jeder gefundene Separator bildet auch im ursprünglichen Graph einen Separator. Sei a die Anzahl der Zusammenhangskomponenten von G . Beispielsweise kann eine Tiefensuche genutzt werden, die anschließend ein zweites Mal die Zusammenhangskomponenten zählt, aber die drei Knoten des Separators nicht beachtet. Sei b diese Anzahl an Zusammenhangskomponenten ohne den (3,3)-Separator. Dann muss $b - a \geq 2$ gelten.

Kapitel 5

Implementierung

Im Folgenden werden Details zur Implementierung im *Open Graph Drawing Framework* erklärt. Dazu wird zunächst kurz das Framework vorgestellt, anschließend werden wichtige Klassen und Algorithmen behandelt und letztlich zusammengesetzt, um einen zertifizierenden Algorithmus zum Finden von K_5 -Minoren zu bilden.

5.1 Open Graph Drawing Framework

Das *Open Graph Drawing Framework (OGDF)* ist ein in C++ geschriebenes Framework

Kapitel 6

Experimentelle Analyse

Kapitel 7

Zusammenfassung und Ausblick

Anhang A

Weitere Informationen

Abbildungsverzeichnis

2.1	Die Kante, die u und v in G verbindet, wird kontrahiert, sodass sie in H durch den neuen Knoten w ersetzt wird.	3
2.2	Der Pfad von p_1 bis p_4 wird kontrahiert. Der neue Knoten w in H enthält alle Nachbarn der Pfadknoten in G	3
2.3	Ein Graph G mit seinen Minoren H_1 und H_2 . Um H_1 zu erhalten, wurde in G die Kante (d, e) und anschließend der Knoten d entfernt. Für H_2 wurden außerdem der Pfad $P(a, c)$ kontrahiert.	4
2.4	Der Graph K_5	4
2.5	Der Graph $K_{3,3}$	4
2.6	Der Graph W , links in seiner üblichen Darstellung, rechts mit angedeutetem $K_{3,3}$ -Minor.	5
2.7	Der Graph M sowie ein K_5 -Minor aus M	5
2.8	Der linke Graph wird in drei augmentierte Komponenten durch den $(3, 3)$ -Separator $C = \{c_1, c_2, c_3\}$ geteilt. Alle Z_i und Z'_i stellen Teilgraphen dar, die zur Übersicht zu einem Knoten zusammengefügt wurden. Die drei rechten Graphen können durch die Cliques-Summe der Cliques $\{c'_1, c'_2, c'_3\}$ sowie $\{c''_1, c''_2, c''_3\}$ und $\{c'''_1, c'''_2, c'''_3\}$ den rechten Graph erzeugen. Während der Cliques-Summen Operation dürfen die Kanten, die die Knoten in den Cliques verbinden, gelöscht werden.	7
3.1	Gegenbeispiel zu Theorem 3.1.1 für $k = 3$ mit $C = \{a, d, f\}$ in W , wodurch die $k - 1$ Komponenten A_1 und A_2 entstehen, von denen nur eine aus mehr als einem Knoten besteht. Die Komponente A_1 ist zwar ein gültiger Minor, da sie etwa durch die Kontraktionen der Kanten (a, b) , (a, h) , (f, g) und (c, d) aus W erzeugt werden kann. A_2 dagegen kann nicht durch Kontraktionen aus W erzeugt werden - wird beispielsweise die Kante (d, e) in W kontrahiert, fehlt die Kante (a, f) in A_2 . Analog kann e mit keiner seiner inzidenten Kanten kontrahiert werden, um einen zu A_2 isomorphen Graph zu erhalten.	11
3.2	Beispiel zum ersten Fall von Lemma 3.1.4.	13

3.3	Beispiel zum zweiten Fall von Lemma 3.1.4.	13
3.4	Beispiel zum dritten Fall von Lemma 3.1.4.	14
3.5	Beispiel zum ersten Fall von Lemma 3.1.5. Links oben ist S mit zwei zusätzlichen Pfaden aus G abgebildet. P_1 hat beide Endpunkte in $F(a)$ sowie den blauen Branch-Fans $F(x)$ und $F(y)$, mit denen die Endpunkte kontrahiert werden. P_2 hat seine Endpunkte in den roten Branch-Fans $F(b)$ und $F(c)$, mit denen sie kontrahiert werden. Rechts oben ist der dadurch entstehende Minor abgebildet und unten wird der enthaltene K_5 -Minor gezeigt.	15
3.6	Beispiel zum zweiten Fall von Lemma 3.1.5. Rechts ist der enthaltene M -Minor abgebildet.	16
4.1	Ein nicht 2-zusammenhängender Graph G_1	19
4.2	Der Block-Cut Tree zu G aus Abbildung 4.1.	20
4.3	Der 1-Block-Tree zu G aus Abbildung 4.2. Die Cliquenknoten sind ovalförmig dargestellt.	21
4.4	Unterteilung der Kanten in drei Mengen E_1 , E_2 und E_3 anhand des Knotenpaares $\{u, v\}$	22
4.5	Ein 2-zusammenhängender Graph G_2	23
4.6	SPQR Baum zum Graph G_2 aus Abb. 4.5.	24
4.7	2-Block-Tree zum Graph aus Abb. 4.5. Die Cliquenknoten sind ovalförmig dargestellt und enthalten immer genau zwei Knoten, alle übrigen sind Blockknoten.	26
4.8	Ein 3-zusammenhängender Graph mit dazugehörigem $(3, 3)$ -Block-Tree. Der Separator wird durch $\{e, f, g\}$ gebildet. Es ist zu sehen, dass der Eingabegraph nicht planar ist, da ein $K_{3,3}$ -Minor enthalten ist. Die Blockknoten enthalten jedoch alle planare Minoren des ursprünglichen Graphs.	27

Algorithmenverzeichnis

Symbolverzeichnis

Literaturverzeichnis

- [1] ANDRÉ E. KÉZDY, PATRICK MCGUINNESS: *Sequential and Parallel Algorithms to Find a K_5 Minor*. In: *Proceedings of the Third Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 27-29 January 1992, Orlando, Florida, USA.*, Seiten 345–356, 1992.
- [2] BARAHONA, FRANCISCO: *The Max-cut Problem on Graphs Not Contractible to K_5* . *Oper. Res. Lett.*, 2(3):107–111, 1983.
- [3] CARSTEN GUTWENGER, PETRA MUTZEL: *A Linear Time Implementation of SPQR-Trees*. In: *Graph Drawing, 8th International Symposium, GD 2000, Colonial Williamsburg, VA, USA, September 20-23, 2000, Proceedings*, Seiten 77–90, 2000.
- [4] DIESTEL, REINHARD: *Graph Theory, 4th Edition*, Band 173 der Reihe *Graduate texts in mathematics*. Springer, 2012.
- [5] FRANCISCO BARAHONA, MICHAEL JÜNGER, GERHARD REINELT: *Experiments in quadratic 0-1 programming*. *Math. Program.*, 44(1-3):127–137, 1989.
- [6] JOHN A. BONDY, UPPALURI S. R. MURTY: *Graph Theory*. Springer Publishing Company, Incorporated, 1st Auflage, 2008.
- [7] JOHN M. BOYER, WENDY J. MYRVOLD: *On the Cutting Edge: Simplified $O(n)$ Planarity by Edge Addition*. *J. Graph Algorithms Appl.*, 8(3):241–273, 2004.
- [8] JÜNGER, MICHAEL, ELISABETH LOBE, PETRA MUTZEL, GERHARD REINELT, FRANZ RENDL, GIOVANNI RINALDI und TOBIAS STOLLENWERK: *Performance of a Quantum Annealer for Ising Ground State Computations on Chimera Graphs*. *CoRR*, abs/1904.11965, 2019.
- [9] KARP, RICHARD M.: *Reducibility Among Combinatorial Problems*. In: *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, Seiten 85–103, 1972.

- [10] LI, ZHENTAO: *Tree decompositions and linear time algorithms*. Doktorarbeit, School of Computer Science, McGill University, Montreal, Quebec, 12 2011.
- [11] WAGNER, KLAUS: *Über eine Eigenschaft der ebenen Komplexe*. Mathematische Annalen, 114:570–590, 1937.
- [12] WILLIAMSON, S. G.: *Depth-First Search and Kuratowski Subgraphs*. J. ACM, 31(4):681–693, 1984.

Eidesstattliche Versicherung

Sauer, Julian

197859

Name, Vorname

Matr.-nr.

Ich versichere hiermit an Eides statt, dass ich die vorliegende Masterarbeit mit dem Titel

Effiziente Berechnung von K_5 -Minoren in Graphen

selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Dortmund, den 20. September 2019

Ort, Datum

Unterschrift

Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/ die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -)

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird gfs. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Dortmund, den 20. September 2019

Ort, Datum

Unterschrift

