

[ beforekip=-.7afterskip=1sp ]paragraph [ beforekip=.7afterskip=-1em ]subpara-  
graph

# **Algorithmen für planaren Graphen**

**Mitschrieb der Vorlesung von Prof. Dr. Dorothea Wagner am KIT  
im Sommersemester 2016**

Joshua Gleitze

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
<b>2</b>	<b>Grundlegende Eigenschaften planarer Graphen</b>	<b>4</b>
2.1	Der Dualgraph . . . . .	4
2.2	Knoten- und Kantenzahlen . . . . .	6
<b>3</b>	<b>Der Satz von Kuratowski</b>	<b>9</b>
<b>4</b>	<b>Färbung planarer Graphen</b>	<b>16</b>
4.1	Einfache Färbung . . . . .	16
4.2	Listenfärbung . . . . .	17
<b>5</b>	<b>Separatoren von planaren Graphen</b>	<b>24</b>
5.1	Planar Separator Theorem . . . . .	24
5.2	Anwendung des Planar Separator Theorems . . . . .	29
5.2.1	Matching . . . . .	29
5.2.2	Mixed Max Cut . . . . .	32
<b>6</b>	<b>Das Menger-Problem</b>	<b>36</b>
6.1	Kantendisjunktes Menger-Problem . . . . .	37
6.2	Knotendisjunktes Menger-Problem . . . . .	42
<b>7</b>	<b>Das Problem von Okamura und Seymour</b>	<b>44</b>

# Verzeichnis der Sätze

$K_5$ ist nicht planar . . . . .	1
$K_{3,3}$ ist nicht planar . . . . .	3
Satz von Euler . . . . .	7
Satz von Kuratowski . . . . .	11
Vierfarbensatz . . . . .	16
Fünffarbensatz . . . . .	16
Vierlistenfärbbarkeit . . . . .	18
Fünflistenfärbbarkeit . . . . .	21
Planar Separator Theorem . . . . .	27
Satz von Menger . . . . .	37
Satz von Okamura und Seymour . . . . .	45

# 1 Einführung

Diese Vorlesung handelt meist von einem ungerichteten, einfachen Graph  $G = (V, E)$  ohne Schlingen. Kann man  $G$  so zeichnen, dass sich keine seiner Kanten kreuzen, so bezeichnet man  $G$  als planar. Dies wird im Folgenden formal definiert.

Prinzipiell kann ein Graph  $G = (V, E)$  dargestellt werden indem man seine Knoten  $v \in V$  auf Punkte in  $\mathbb{R}^2$  und seine Kanten  $e \in E$  auf Jordan-Kurven in  $\mathbb{R}^2$ , die die entsprechenden Punkte verbinden, abbildet.

---

**Definition 1.** Eine stetige, sich nicht selbst kreuzende Kurve heißt JORDAN-KURVE.

---

*Notation.*  $K_n = (V, E)$  bezeichnet den vollständigen Graph mit  $|V| = n$  und

$$E = \{\{u, v\} \mid u, v \in V \wedge u \neq v\}$$

$K_{n,m} = (V, E)$  bezeichnet den vollständig bipartiten Graph mit  $V = V_1 \oplus V_2$ ,  $|V_1| = n$ ,  $|V_2| = m$  und

$$E = \{\{v_1, v_2\} \mid v_1 \in V_1 \wedge v_2 \in V_2\}$$

---

**Definition 2.** Ein Graph  $G = (V, E)$  heißt PLANAR, falls es eine Darstellung von  $G$  in  $\mathbb{R}^2$  gibt, bei der sich keine zwei Kanten kreuzen. Das heißt, dass die Jordan-Kurven, die die Kanten darstellen, sich nur in Endpunkten berühren.

---

| **Satz 1.**  $K_5$  ist nicht planar.

*Beweis.* Sei o.B.d.A  $V := \{1, 2, 3, 4, 5\}$ . Bette die Knoten und alle zu 1 inzidenten Kanten beliebig in die Ebene ein. Dann existieren zwei Knoten (hier o.B.d.A. 2 und 4), sodass der Kreis  $C := \{\{4, 1\}, \{1, 2\}, \{2, 4\}\}$  die Ebene in zwei Teile teilt: Äußeres( $C$ ) und Inneres( $C$ ), wobei  $3 \in \text{Inneres}(C)$  und  $5 \in \text{Äußeres}(C)$ . Jede Kante zwischen Innerem und Äußerem von  $C$  muss den Rand von  $C$  kreuzen. Die Kante  $\{3, 5\}$  kann also nicht kreuzungsfrei gezogen werden (Jordan'scher Kurvensatz). ■

## 1 Einführung

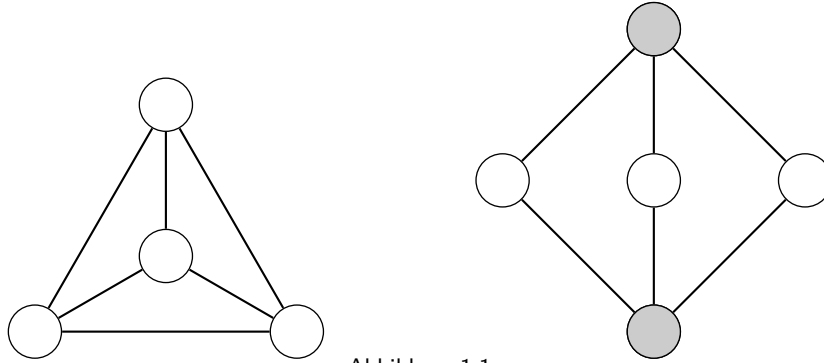


Abbildung 1.1  
 $K_4$  und  $K_{2,3}$  sind planar.

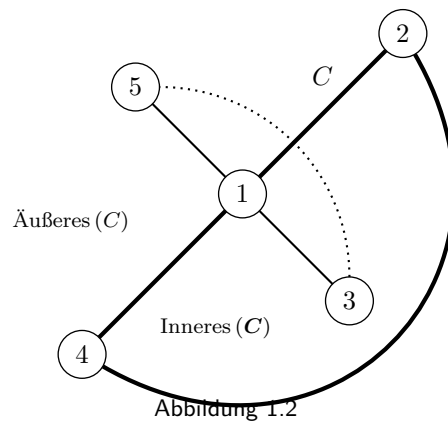


Abbildung 1.2

In Satz 1 teilt  $C$  die Ebene in zwei Gebiete. Die Kante  $\{3, 5\}$  kann nicht kreuzungsfrei gezogen werden.

**Satz 2.**  $K_{3,3}$  ist nicht planar.

*Beweis.* Seien o.B.d.A.  $V = \{1, 2, 3, a, b, c\}$ . Bette den Kreis  $C$  der Knoten  $\{1, a, 2, b, 3, c\}$  kreuzungsfrei in die Ebene ein. Wird o.B.d.A.  $\{1, b\}$  durch das Innere von  $C$  gezogen, so muss  $\{2, c\}$  durch das Äußere von  $C$  gezogen werden. Sei  $D := \{\{1, c\}, \{c, 2\}, \{2, b\}, \{b, 1\}\}$ . Dann ist  $3 \in \text{Inneres}(D)$  und  $a \in \text{Äußeres}(D)$ . Die verbleibende Kante  $\{3, a\}$  kann also nicht kreuzungsfrei gezogen werden.

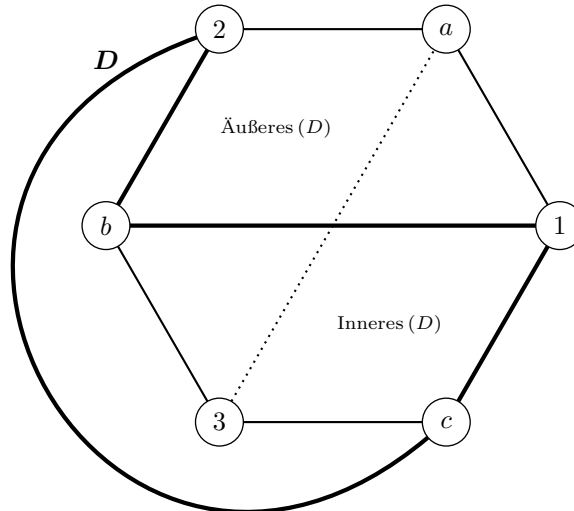


Abbildung 1.3

In Satz 2 kann die Kante  $\{3, a\}$  nicht kreuzungsfrei gezogen werden, da sie  $D$  überqueren müsste.

■

**Korollar.** Es gibt Graphen, die nicht planar sind.

## 2 Grundlegende Eigenschaften planarer Graphen

### 2.1 Der Dualgraph

---

**Definition 3.** Die planare Darstellung oder EINBETTUNG von  $G$  zerlegt die Ebene in FACETTEN (auch GEBIETE, FLÄCHEN). Eine planare Einbettung, die durch Facetten bzw. die Reihenfolge der Adjazenzlisten der Knoten induziert wird, heißt KOMBINATORISCHE EINBETTUNG. Ist die Einbettung mit Koordinaten der Knoten und Kanten gegeben, so heißt sie GEOMETRISCHE EINBETTUNG.

---

Ein planarer Graph  $G = (V, E)$  kann durch die Facettenmenge  $\mathcal{F}$  bezeichnet werden. Hierbei ist es ausreichend,  $\mathcal{F}$  durch die Auflistung der Anzahl der an jede Facette grenzenden Kanten anzugeben.

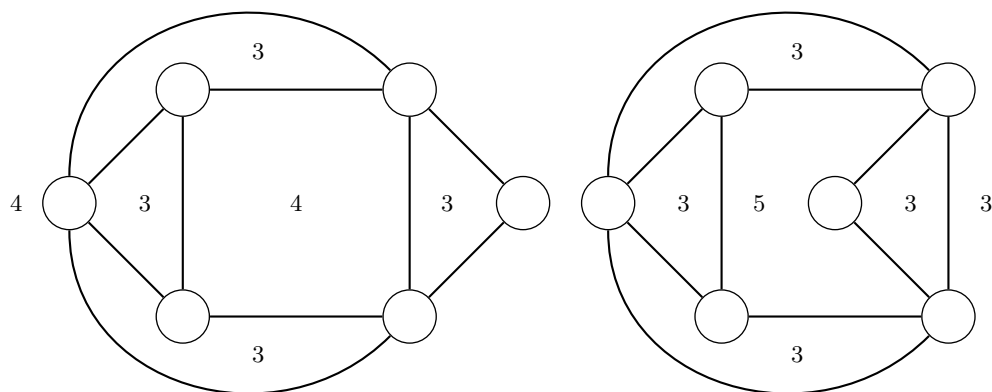


Abbildung 2.1

Der selbe Graph mit unterschiedlichen kombinatorischen Einbettungen. Kanten pro Facette links: 3,3,3,3,4,4; rechts: 3,3,3,3,3,5.



---

**Definition 4.** Zu einem Graphen  $G = (V, E)$  mit einer planaren Einbettung und einer Menge  $\mathcal{F}$  der entsprechenden Facetten ist der DUALGRAPH  $G^* = (V^*, E^*)$  definiert durch

- Für jede Facette  $f \in \mathcal{F}$  existiere  $v_f \in V^*$
  - Zu jeder Kante  $e = \{u, v\} \in E$  existiere die DUALKANTE  $e^* \in E^*$  die die beiden/den Knoten aus  $V^*$  verbindet an deren/dessen Facetten aus  $\mathcal{F}$   $e$  angrenzt.
- 

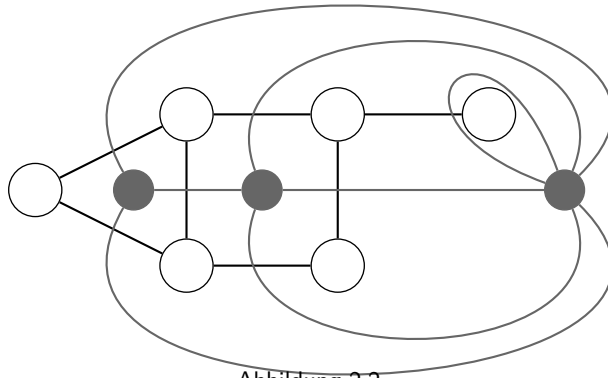


Abbildung 2.2

Ein Graph und sein Dualgraph

*Bemerkung* (Eigenschaften des Dualgraphen).

- Der Dualgraph eines einfachen Graphen ist nicht notwendigerweise einfach.
- Der Dualgraph zu  $G$  hängt von der Einbettung von  $G$  ab. Dualgraphen des selben Graphens basierend auf unterschiedlichen Einbettungen können verschieden (also nicht isomorph) sein.
- Der Dualgraph zu einem planaren Graph  $G$  (mit planarer Einbettung  $\mathcal{F}$ ) ist wieder planar.
- Der Dualgraph  $(G^*)^*$  zu  $G^*$  (bezüglich „kanonischer“ Einbettung von  $G^*$ ) ist wieder  $G$  selbst.

Im Folgenden wird der Satz von Kuratowski eine Charakterisierung planarer Graphen liefern. Aus ihm lässt sich (allerdings nicht auf triviale Weise) ein Algorithmus für den Test auf Planarität ableiten. Es wird sich zeigen, dass manche Algorithmen, die auf allgemeinen Graphen  $\mathcal{NP}$ -schwer sind, auf planaren Graphen in polynomieller Zeit lösbar sind (z.B. MAX CUT bzw. MIXED MAX CUT). Andere, polynomielle Algorithmen sind für planare Graphen einfacher bzw. effizienter lösbar (z.B. MATCHING).

Planare Graphen haben Eigenschaften, die allgemeine Graphen nicht haben, wie beispielsweise Färbbarkeit oder balancierte Zerlegbarkeit.

---

**Definition 5.** Sei  $(V_1, V_2)$  ein Partition der Knotenmenge eines Graphen  $G = (V, E)$ . Sie induziert eine Menge  $E' = \{\{v_1, v_2\} \in E \mid v_1 \in V_1, v_2 \in V_2\}$ , die **SCHNITT** in  $G$  genannt wird.

---

**Lemma 1** (Dualität von Schnitten und Kreisen). *Für einen planaren Graph  $G$  mit fester planarer Einbettung und zugehörigem Dualgraph  $G^*$  gilt:*  
 $S \subseteq E$  ist ein Schnitt in  $G \iff S^* \subseteq E^*$  ist eine Menge von Kreisen in  $G^*$ .

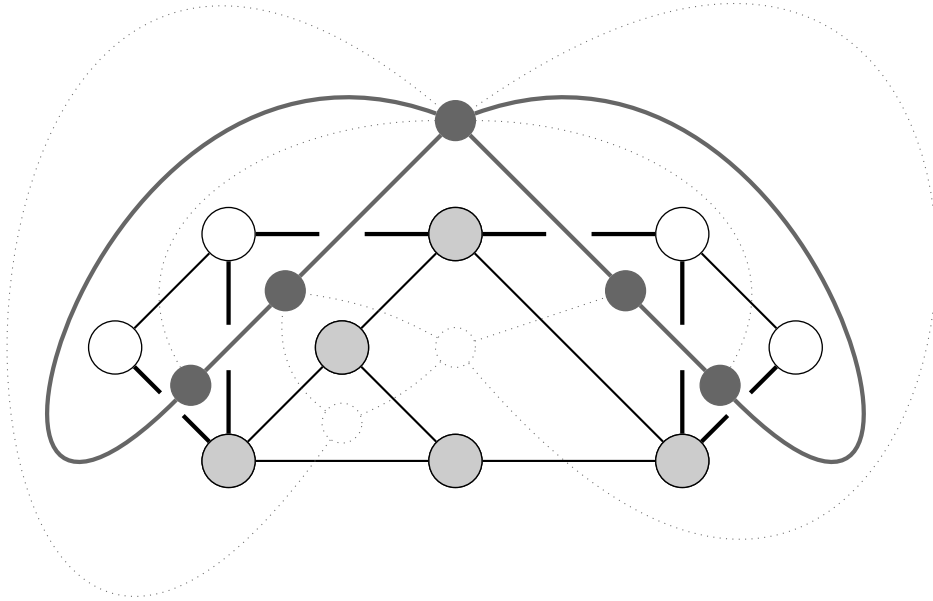


Abbildung 2.3

Illustration von Lemma 1: Die Dualkanten zu einem Schnitt in  $G$  bilden eine Menge von Kreisen in  $G^*$ .

Hierbei bezeichnet  $S^*$  die Dualkanten der Kanten in  $S$ . Da  $(G^*)^* = G$  gilt das Lemma entsprechend für Schnitte in  $G^*$  und Kreise in  $G$ .

## 2.2 Knoten- und Kantenzahlen

*Notation.* Im Folgenden gelte für einen Graphen  $G = (V, E)$  und seiner planaren Einbettung  $\mathcal{F}$   $n := |V|$ ;  $m := |E|$ ;  $f = |\mathcal{F}|$ .

## 2 Grundlegende Eigenschaften planarer Graphen

*Notation.* Für einen Graphen  $G = (V, E)$ , einen Knoten  $v \in V$  und eine Kante  $e \in E$  sei

$$G - v := (V \setminus \{v\}, \{\{x, y\} \in E \mid x \neq v \wedge y \neq v\})$$

sowie

$$G - e := (V, E \setminus \{e\})$$

der Graph, der entsteht, indem man  $v$  bzw.  $e$  aus  $G$  „löscht“

**Satz 3** (Satz von Euler). *In einem zusammenhängenden, planaren Graph  $G = (V, E)$  mit  $|V| > 0$  gilt für jede planare Einbettung*

$$n - m + f = 2$$

*Beweis.* Vollständige Induktion über  $m$ :

**Induktionsanfang** Wenn  $m = 0$  ist  $n = 1$  und  $f = 1$ . Es folgt  $n - m + f = 1 - 0 + 1 = 2$ .

**Induktionsschritt** Sei  $G = (V, E)$  mit  $m \geq 2$ .

*Fall 1.*  $G$  enthält einen Kreis

Sei  $e \in E$  eine Kante, sodass  $G' := G - e = (V', E')$  zusammenhängend ist und für eine beliebige aber feste Einbettung von  $G$  die beiden Facetten, die an  $e$  angrenzen, in  $G'$  zu einer Facette verschmelzen. Dann gilt  $|E'| = m - 1$  und  $|F'| = f - 1$ . Die Induktionsvoraussetzung gilt und impliziert  $n - (m - 1) + (f - 1) = 2$ . Es folgt die Behauptung.

*Fall 2.*  $G$  ist kreisfrei.

Das heißt  $G$  ist ein Baum und  $f = 1$  und es existiert eine Kante  $e$  sodass  $G - e$  in zwei zusammenhängenden Graphen  $G_1 = (V_1, E_1)$  und  $G_2 = (V_2, E_2)$  zerfällt. Nach Induktionsvoraussetzung gilt  $n_1 - m_1 + f_1 = n_1 - m_1 + 1 = 2$  und  $n_2 - m_2 + f_2 = n_2 - m_2 + 1 = 2$ . Da  $n_1 + n_2 = n$  und  $m_1 + m_2 = m - 1$  folgt

$$\begin{aligned} n - m + f &= n - m + 1 \\ &= n_1 + n_2 - m_1 - m_2 - 1 + 1 \\ &= (n_1 - m_1 + 1) + (n_2 - m_2 + 1) - 2 \\ &= 2 + 2 - 2 \\ &= 2 \end{aligned}$$

■

**Korollar 1.** *Für jede planare Einbettung eines Graphen  $G$  ist die Anzahl der Facetten gleich.*

## 2 Grundlegende Eigenschaften planarer Graphen

**Korollar 2.** *Ein Baum mit  $n$  Knoten hat  $n - 1$  Kanten und eine Facette.*

**Lemma 2** (Maximale Kantenzahl in planaren Graphen). *Ein planarer Graph mit  $n$  Knoten,  $n \geq 3$ , hat höchstens  $3n - 6$  Kanten.*

*Beweis.* Sei  $G$  o.B.d.A maximal planar, das heißt, dass das Hinzufügen einer beliebigen Kante zu  $G$  zu einem nicht planaren Graphen führt. Dann sind bezüglich jeder planaren Einbettung von  $G$  alle Facetten Dreiecke und jede Kante trennt zwei Facetten.  $G$  ist insbesondere zusammenhängend.

$$\begin{aligned} \Rightarrow \quad & 2m = 3f \\ & = 3 \cdot (2 - n + m) \\ & = 6 - 3n + 3m \\ \Rightarrow \quad & -m = 6 - 3n \\ \Rightarrow \quad & m = 3n - 6 \end{aligned}$$

■

**Lemma 3** (Summenformel für Knotengrade in planaren Graphen). *Sei  $G = (V, E)$  ein planarer Graph mit mindestens 3 Knoten. Es bezeichne  $d_{\max}(E)$  den maximalen Knotengrad in  $G$  und für  $0 \leq i \leq d_{\max}(E)$  bezeichnen  $n_i$  die Anzahl der Knoten von Grad  $i$  in  $G$ . Dann gilt:*

$$6n_0 + 5n_1 + 4n_2 + 3n_3 + 2n_4 + n_5 \geq n_7 + 2n_8 + 3n_9 + \dots + (d_{\max}(E) - 6)n_{d_{\max}(E)} + 12$$

*Beweis.* Es gilt  $\sum_{i=0}^{d_{\max}(E)} n_i = n$  und  $\sum_{i=0}^{d_{\max}(E)} i \cdot n_i = 2m$ . Damit:

$$\begin{aligned} \Rightarrow \quad & m \leq 3n - 6 \\ \Rightarrow \quad & 6n \geq 2m + 12 \\ \Rightarrow \quad & 6 \cdot \sum_{i=0}^{d_{\max}(E)} n_i \geq \sum_{i=0}^{d_{\max}(E)} i \cdot n_i + 12 \end{aligned}$$

Es folgt die Behauptung. ■

**Korollar 3** (Knoten mit Grad kleiner 5). *In jedem planaren Graphen  $G = (V, E)$  existiert  $v \in V$  mit  $d(v) \leq 5$ .*

### 3 Der Satz von Kuratowski

---

**Definition 6.** Sei  $G = (V, E)$  ein Graph.

- Ein Graph  $G' = (V', E')$  heißt SUBGRAPH von  $G$ , falls  $V' \subseteq V \wedge E' \subseteq E$ .
  - Ein Graph  $G'$  heißt UNTERTEILUNG von  $G$ , falls  $G'$  aus  $G$  entsteht, indem Kanten durch einfache Wege ersetzt werden.
  - Ein Graph  $H$  heißt MINOR von  $G$  falls,  $H$  aus  $G$  durch Löschen von Knoten und/oder Kanten und/oder Kontraktion von Kanten entsteht. Hierbei meint die Kontraktion zweier Kanten  $e = \{u, v\}$  und  $f = \{v, w\}$ , wobei  $v$  den Grad 2 hat, das Löschen von  $v$ ,  $e$  und  $f$  sowie das Einfügen der Kante  $\{u, w\}$  („Gegenteil von Unterteilung“)
- 

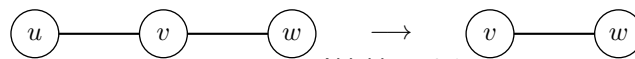


Abbildung 3.1

Kontraktion der Kanten  $\{u, v\}$  und  $\{v, w\}$

*Bemerkung 1.*

- $H$  ist Minor von  $G$  wenn  $G$  eine Unterteilung von  $H$  als Subgraph enthält.
- Ein planarer Graph kann keinen nicht-planaren Graphen als Subgraph enthalten.
- Ein planarer Graph kann nicht eine Unterteilung eines nicht-planaren Graphen sein.

**Korollar 4.** *Ein planarer Graph kann keinen nicht-planaren Graphen als Minor enthalten.*

---

**Definition 7.** Ein Graph, der  $K_{3,2}$  als Minor oder Unterteilung enthält, heißt  $\Theta$ -GRAPH.

---

**Lemma 4.** *Wenn  $G$  ein planarer Graph ist, der einen einfachen Kreis  $C$  enthält und keine zwei Knoten auf  $C$  in  $G - E(C)$  verbunden sind, so kann  $G$  so planar eingebettet werden, dass  $C$  eine Facette begrenzt.*

### 3 Der Satz von Kuratowski

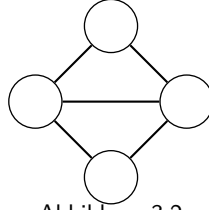


Abbildung 3.2

Der kleinste  $\Theta$ -Graph: Ein Kreis mit einer Querkante.

**Lemma 5.** *Facettenränder enthalten keinen  $\Theta$ -Graph. Ist also  $G$  planar mit fester Einbettung,  $f$  eine Facette und  $F$  der Subgraph von  $G$ , der aus allen zu  $f$  inzidenten Knoten und Kanten besteht,  $F$  keinen  $\Theta$ -Graphen enthält.*

---

**Definition.** Ein Graph  $G$  ist ein MINOR-MINIMALER, NICHT-PLANARER Graph wenn

- $G$  nicht planar ist, aber
  - jeder Minor von  $G$  planar ist
- 

**Beispiel.**  $K_5$  und  $K_{3,3}$  sind minor-minimal nicht-planar.

---

**Definition 8.** Auf der Kantenmenge  $E$  eines Graphen  $G = (V, E)$  sei die Äquivalenzrelation  $\sim$  definiert durch:

$$\forall l_1, l_2 \in E : l_1 \sim l_2 \iff l_1 = l_2 \vee \exists \text{ einfacher Kreis, der } l_1 \text{ und } l_2 \text{ enthält}$$

In diesem Kontext wird ein Subgraph, der aus einer Äquivalenzklasse besteht, BLOCK und die Zerlegung von  $G$  in seine Blöcke BLOCKZERLEGUNG genannt. In der Blockzerlegung ist jede Kante  $e \in E$  genau einem Block zugeordnet. Knoten, die in mehreren Blöcken enthalten sind, heißen SEPARATORKNOTEN.

---

*Notation.* In einem Graphen  $G = (V, E)$  bezeichne für ein beliebiges  $v \in V$   $N(v)$  die NACHBARSCHAFT von  $v$ :

$$N(v) := \{u \in V \mid \{v, u\} \in E\}$$

### 3 Der Satz von Kuratowski

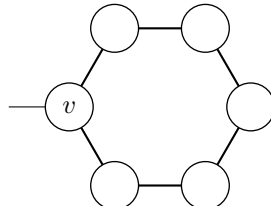


Abbildung 3.3

Ein Kreis mit Separator-Knoten  $v$ .

**Satz 4** (Satz von Kuratowski). *Ein Graph  $G = (V, E)$  ist genau dann planar, wenn er weder  $K_5$  noch  $K_{3,3}$  als Minor enthält.*

*Beweis.*

„  $\Rightarrow$  “ Klar, denn  $K_5$  und  $K_{3,3}$  sind nicht planar und ein planarer Graph kann keinen nicht-planaren Graphen als Minor enthalten.

„  $\Leftarrow$  “ Falls ein Graph nicht planar ist, so enthält er einen minor-minimalen, nicht-planaren Graphen als Minor. Zum Beweis wird nun ein solcher minor-minimaler nicht-planarer Graph  $G = (V, E)$  sowie  $x, y \in V$  mit  $\{x, y\} \in E$  betrachtet. In  $G$  hat jeder Knoten mindestens Grad 3 (da er ansonsten nicht minor-minimal wäre). Sei  $G'$  der Graph, der aus  $G$  entsteht, wenn  $x, y, \{x, y\}$  und alle zu  $x$  und  $y$  inzidenten Kanten entfernt werden ( $G' = G - x - y = G - \{x, y\}$ ). Nun wird nacheinander bewiesen:

1.  $G'$  enthält keinen  $\Theta$ -Graph
2.  $G'$  enthält höchstens einen Knoten mit Grad 1
3.  $G'$  ist ein einfacher Kreis

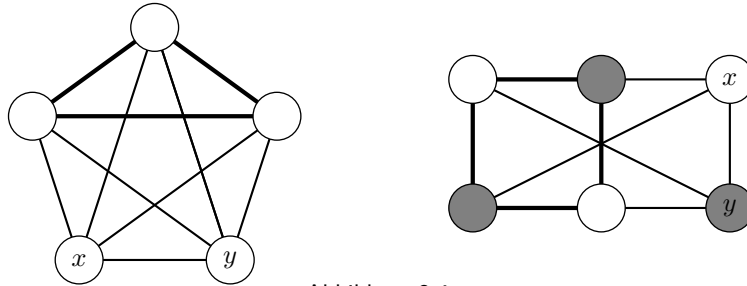


Abbildung 3.4

Das Vorgehen im Beweis von Satz 4 ist sinnvoll, da sowohl  $K_5$  also auch  $K_{3,3}$  entstehen, wenn man zwei Knoten (im Beweis  $x$  und  $y$ ) zu einem einfachen Kreis hinzufügt.

**1.:  $G'$  enthält keinen  $\Theta$ -Graph**

*Beweis.* Angenommen,  $G'$  enthält den  $\Theta$ -Graphen.  $G'$  kann so eingebettet werden, dass  $N(x) \cup N(y)$  auf dem Rand einer Facette liegt. Sei  $F$  der planare Subgraph von  $G'$ , der durch den Rand dieser Facette induziert wird.  $F$  enthält einen einfachen Kreis  $C$ , aber nach Lemma 5 keinen  $\Theta$ -Graph.

Da  $G'$  jedoch den  $\Theta$ -Graphen enthält, muss eine Kante  $e \in E(G) \setminus E(F)$  existieren. O.B.d.A. sei diese Kante im Äußeren von  $C$ . Äußeres( $C$ ) ist also nicht leer. Da  $G$  minor-minimal ist, ist somit  $\tilde{G} := G - \text{Äußeres}(C)$  planar. Nach Lemma 4 kann  $\tilde{G}$  so planar eingebettet werden, dass  $C$  die äußere Facette begrenzt. Zu dieser Einbettung kann Äußeres( $C$ ) allerdings wieder hinzugefügt werden, ohne die Planarität zu zerstören.  $G$  ist somit planar.  $\nexists$  □

**2.:  $G'$  enthält höchstens einen Knoten mit Grad 1**

*Beweis.* Angenommen,  $G'$  enthalte zwei Knoten  $u$  und  $v$ , die beide Grad 1 haben. Da in  $G$  der minimale Grad jedes Knoten 3 beträgt, sind  $u$  und  $v$  in  $G$  sowohl mit  $x$  als auch mit  $y$  durch Kanten verbunden. Damit enthält  $G$  den  $\Theta$ -Graphen. Deshalb kann keine weitere Kante  $\{i, j\}$  mit  $i, j \notin \{u, v, x, y\}$  in  $G$  existieren, da  $G - i - j$  sonst den  $\Theta$ -Graphen enthielte. Das stünde im Widerspruch zu 1..

$G$  kann außer  $\{u, v, x, y\}$  nicht mehr als zwei weitere Knoten besitzen. Ein dritter Knoten müsste mindestens Grad 3 haben, könnte aber nur mit  $x$  und  $y$  verbunden werden, da  $u$  und  $v$  durch die anderen beiden Knoten schon Grad 3 hätten.  $G$  kann also nur wie folgt aussehen:

- es existiert außer  $\{u, v, x, y\}$  kein weiterer Knoten.
- es existiert außer  $\{u, v, x, y\}$  ein weiterer Knoten.
- es existieren außer  $\{u, v, x, y\}$  zwei weitere Knoten  $z, z'$  mit  $\{z, z'\} \notin E$ .

In allen Fällen wäre jedoch  $G$  planar.  $\nexists$  □

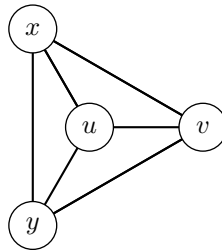
**3.:  $G'$  ist ein einfacher Kreis**

*Beweis.* Betrachte die Blockzerlegung von  $G'$ . Wegen 1. besteht jeder Block entweder aus nur einer Kante oder einem einfachen Kreis. Wegen 2. besteht  $G'$  nicht nur aus einer Kante und enthält auch keine zwei „Endblöcke“, die jeweils nur aus einer Kante bestehen. Es folgt, dass mindestens ein Block in  $G$  aus einem einfachen Kreis besteht.

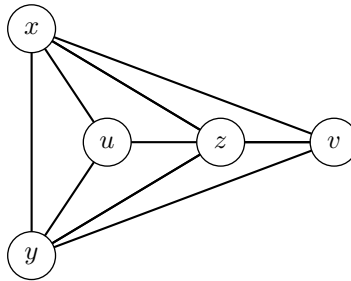
Angenommen,  $G'$  enthalte außer einem Kreis  $C$  noch einen weiteren Knoten. Sei  $v$  ein Separatorknoten auf  $C$ . Da jeder Knoten in  $G$  mindestens Grad 3 hat, sind alle  $i \in C$  (außer  $v$ ) mit  $x$  oder  $y$  verbunden. Außerdem enthält  $C$  mindestens 3 Knoten. Somit existieren mindestens zwei Knoten in  $C \setminus \{v\}$ , die mit  $\{x, y\}$  einen  $\Theta$ -Graph bilden. Wegen 1. gibt es also keine zu  $C$  und  $\{x, y\}$  unabhängige Kante und natürlich auch keine isolierten Knoten.  $G - u - v$  kann wegen 1. keinen  $\Theta$ -Graphen enthalten.  $C$  hat also nur drei Knoten und insgesamt ist  $G$  der Prisma-Graph, welcher planar ist.  $\nexists$  □



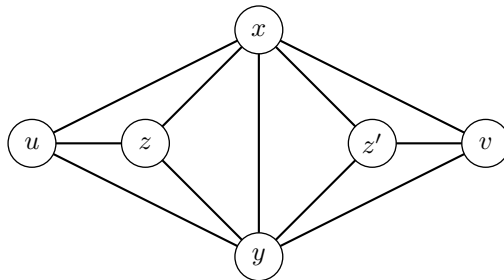
### 3 Der Satz von Kuratowski



(a) Kein weiterer Knoten



(b) Ein weiterer Knoten.



(c) Zwei weitere, nicht verbundene Knoten

Abbildung 3.5

Zu 2. im Beweis von Satz 4:  $G$  wäre in allen Fällen planar, falls es in  $G' = G - x - y$  zwei Knoten  $\{u, v\}$  mit Grad 1 gäbe.

### 3 Der Satz von Kuratowski

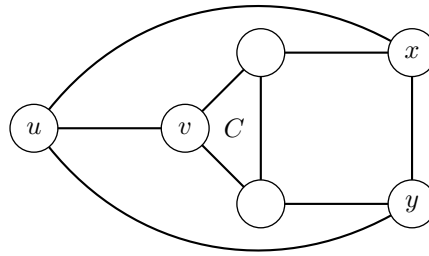


Abbildung 3.6

Falls es in 3. im Beweis von Satz 4 außer dem Kreis in  $C$  in  $G'$  noch einen weiteren Knoten  $u$  gäbe, müsste  $G$  der Prisma-Graph sein. Dieser ist jedoch planar.

**Beweis des Satzes** Ein minor-minimaler, nicht-planarer Graph  $G$  besteht aus einer Kante  $\{x, y\}$ , einem einfachen Kreis  $C$  und Kanten zwischen Knoten auf  $C$  und  $x$  und/oder  $y$ .

Es gilt nun für alle  $u, v \in V$  mit  $\{u, v\} \in E$ :

$$\{u, x\} \in E \wedge \{u, y\} \notin E \implies \{v, x\} \notin E$$

Denn angenommen,  $\{v, x\} \in E$ .  $G - v - x$  ist planar. Zur planaren Einbettung von  $G - v - x$  kann  $\{v, x\}$  kreuzungsfrei eingefügt werden. Somit ist  $G$  auch planar.  $\nexists$

Also gilt für die Knoten auf  $C$ : entweder sind *alle* Knoten mit  $x$  und  $y$  verbunden, oder jeweils abwechselnd. Hierfür gibt es aber nur zwei minor-minimale, nicht-planare Möglichkeiten:  $K_5$  und  $K_{3,3}$ . Alle weiteren Möglichkeiten hätten  $K_5$  oder  $K_{3,3}$  als Minor. ■

### 3 Der Satz von Kuratowski

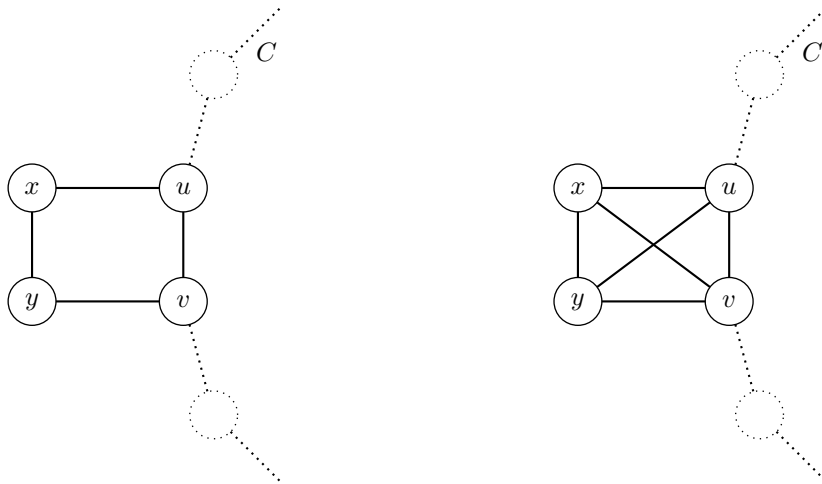


Abbildung 3.7

Im minor-minimalen, nicht-planaren Graph  $G$  sind zwei Knoten  $\{u, v\}$  auf dem Kreis  $C$  entweder jeweils *nur* mit  $x$  oder  $y$  oder *beide* mit  $x$  und  $y$  verbunden. Nur eine „Kreuzkante“ kann nicht vorkommen.

## 4 Färbung planarer Graphen

### 4.1 Einfache Färbung

**Problem** (Landkartenfärbungsproblem). Gegeben eine Landkarte, wie viele Farben werden mindestens benötigt, um jedem Land eine Farbe zuzuweisen, so, dass keine aneinandergrenzenden Länder die selbe Farbe haben?

Eine Landkarte einzufärben entspricht hierbei dem Problem, einen Graphen einzufärben, dessen Knoten mit den Ländern der Karte identifiziert werden und zwei Knoten genau dann eine Kante haben, wenn die entsprechenden Länder aneinander grenzen.

**Problem 1** (Färbungsproblem). Gegeben ein Graph  $G = (V, E)$  und ein  $k \in \mathbb{N}$ , kann  $G$  mit  $k$  Farben so gefärbt werden, dass für jede Kante  $\{u, v\} \in E$   $u$  und  $v$  verschiedene Farben haben?

| **Satz 5.** *Jeder planare Graph kann mit vier Farben gefärbt werden.*

Ohne Beweis.

| **Satz 6.** *Jeder planare Graph ist fünffärbbar*

*Beweis.* Induktion über die Anzahl der Knoten  $n$ .

**Induktionsanfang**  $n \leq 5$ : trivial

**Induktionsvoraussetzung** Sei jeder planarer Graph  $G' = (V, E)$  mit  $|V| \leq n - 1$  fünffärbbar.

**Induktionsschritt** Sei  $G := (V, E)$  mit  $|V| = n$ . Sei  $v \in G$  sodass  $\forall u \in V : d(v) \leq d(u)$ . Dann gilt:  $d(v) \leq 5$ . Sei  $\bar{G} := G - v$ .

*Fall 1.*  $d(v) < 5$ :

Färbe  $\bar{G}$  mit fünf Farben (nach IV möglich). Für die Knoten, die in  $G$  an  $v$  grenzen, werden maximal 4 Farben benötigt, da es höchstens 4 sind. Verwende nicht verwendete Farbe für  $v$ .

## 4 Färbung planarer Graphen

*Fall 2.*  $d(v) = 5$

Beweite  $\tilde{G}$  planar ein. O.B.d.A. seien die Nachbarn von  $v$   $N(v) := \{1, 2, 3, 4, 5\}$ . Diese liegen in  $\tilde{G}$  auf einer Facette (da sie in  $G$  „um  $v$  herum“ liegen). Nach Lemma 2 können nicht gleichzeitig alle Paare in  $N(v)$  verbunden sein: Dies wären 10 Kanten, allerdings können es nicht mehr als  $3 \cdot 5 - 6 = 9$  sein. Es gibt also ein Paar nicht verbundener Knoten. Dies seien o.B.d.A. 2 und 5. Verschmelze 2 und 5 in  $\tilde{G}'$  zu einem Knoten  $u$ . Laut Induktionsvoraussetzung lässt sich  $\tilde{G}'$  fünffärben. Somit kann auch  $\tilde{G}$  mit fünf Farben gefärbt werden, wobei 2 und 5 die selbe Farbe erhalten. Insgesamt werden also für 1, 2, 3, 4 und 5 nur vier Farben benötigt. Die verbleibende kann für  $v$  verwendet werden.

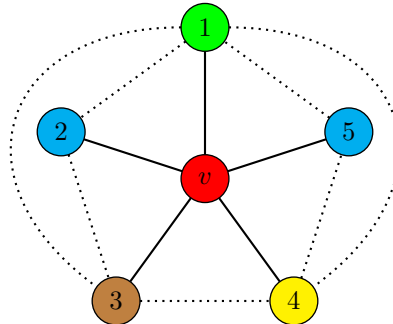


Abbildung 4.1

Ordnet man im zweiten Fall von Satz 6 verbundene Knoten im Kreis an, können nicht alle Paare „sich gegenüberliegender“ Knoten verbunden sein, da  $G$  sonst nicht planar wäre. Somit bleibt eine Farbe für  $v$  übrig.

■

### 4.2 Listenfärbung

**Problem 2** (Listenfärbungsproblem). Gegeben ein Graph  $G = (V, E)$  und ein  $k \in \mathbb{N}$ , gibt es für jede beliebige Zuordnung  $s_v$ ,  $v \in V$  mit  $|s_v| = k$  ( $s_v$  ist eine „Liste“ von Farben) eine „korrekte“ Knotenfärbung, bei der die Farbe jedes Knoten  $v$  aus seiner Liste  $s_v$  ist?

---

**Definition 9.** Existiert zu einem Graph  $G = (V, E)$  für jede beliebige Zuordnung von Listen  $s_v$ ,  $|s_v| = k$  zu jedem  $v \in V$  eine korrekte Listenfärbung, so heißt  $G$   $k$ -LISTENFÄRBBAR.

---

#### 4 Färbung planarer Graphen

**Satz 7.** *Es gibt planare Graphen, die nicht 4-listenfärbbar sind.*

*Beweis.* Es wird ein Gegenbeispiel konstruiert. Basis des Beispiels eine (unvollständige) Listenfärbungsinstanz  $I$  mit fünf Knoten. Diese hat die „vis-à-vis“-Eigenschaft: In jeder korrekten Färbung haben zwei gegenüberliegende Knoten die selbe Farbe (vgl. Abbildung 4.1).

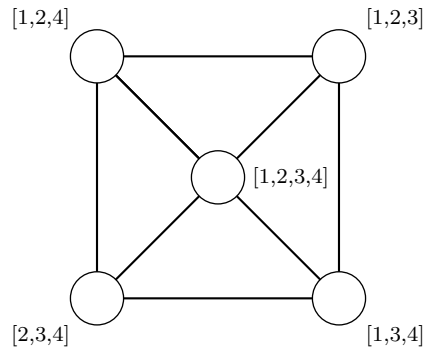


Abbildung 4.1

Die Instanz  $I$  im Beispiel von Satz 7. Eine korrekte Färbung weist mindestens zwei gegenüberliegenden Knoten die selbe Farbe zu. Andernfalls würden vier Farben für die äußeren Knoten verwendet und der mittlere Knoten könnte nicht korrekt gefärbt werden.

Aus fünf  $I$ -Instanzen wird ein Graph  $H$  konstruiert. Hieraus folgt, dass wenn den 3-elementigen Listen der äußeren Knoten eine weitere Farbe hinzugefügt wird, diese Farbe in einer korrekten Färbung von  $H$  auch verwendet wird.

Nun kann das Gegenbeispiel konstruiert werden. Es besteht aus vier Instanzen von  $H$ ,  $H_1, H_2, H_3, H_4$ . Den Listen aller äußeren Knoten einer Instanz  $H_i$  wird eine neue Farbe  $f_i$  mit  $f_i \neq f_j \iff i \neq j$  zugewiesen. Die äußeren Knoten aller Instanzen werden nun mit einem einzigen Knoten  $x$  verbunden (was planar möglich ist). Dieser Knoten erhält als Liste  $[f_1, f_2, f_3, f_4]$ . Der so konstruierte Graph kann keine korrekte Färbung haben. Denn erhält  $x$  die Farbe  $f_k$ , so kann  $H_k$  nicht mehr korrekt gefärbt werden.

#### 4 Färbung planarer Graphen

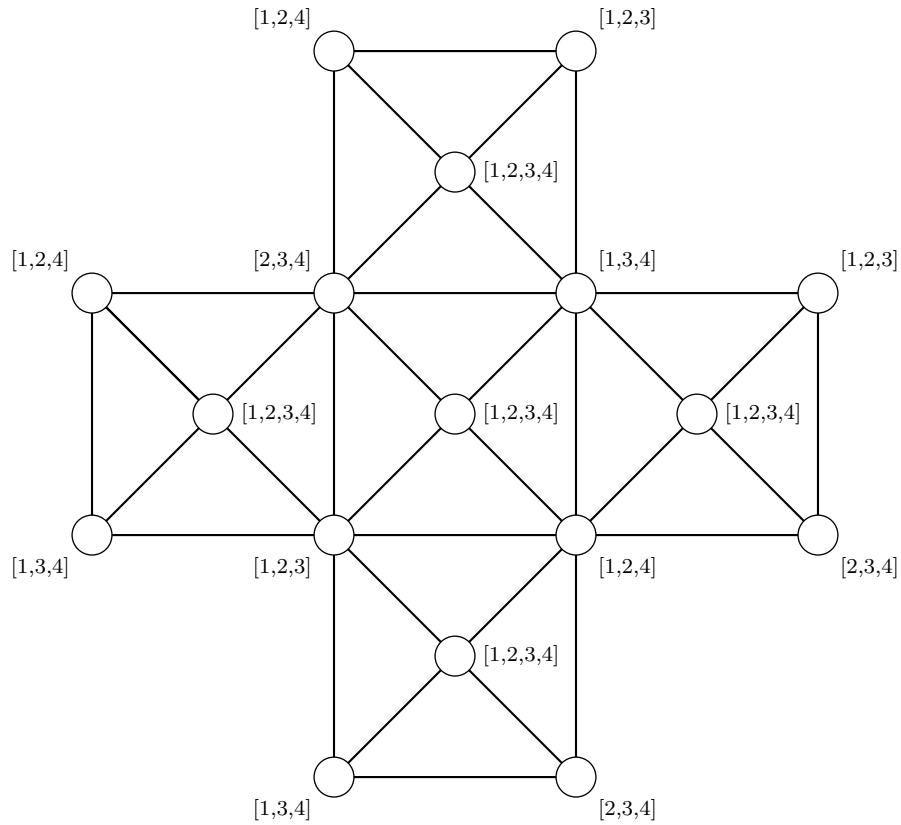
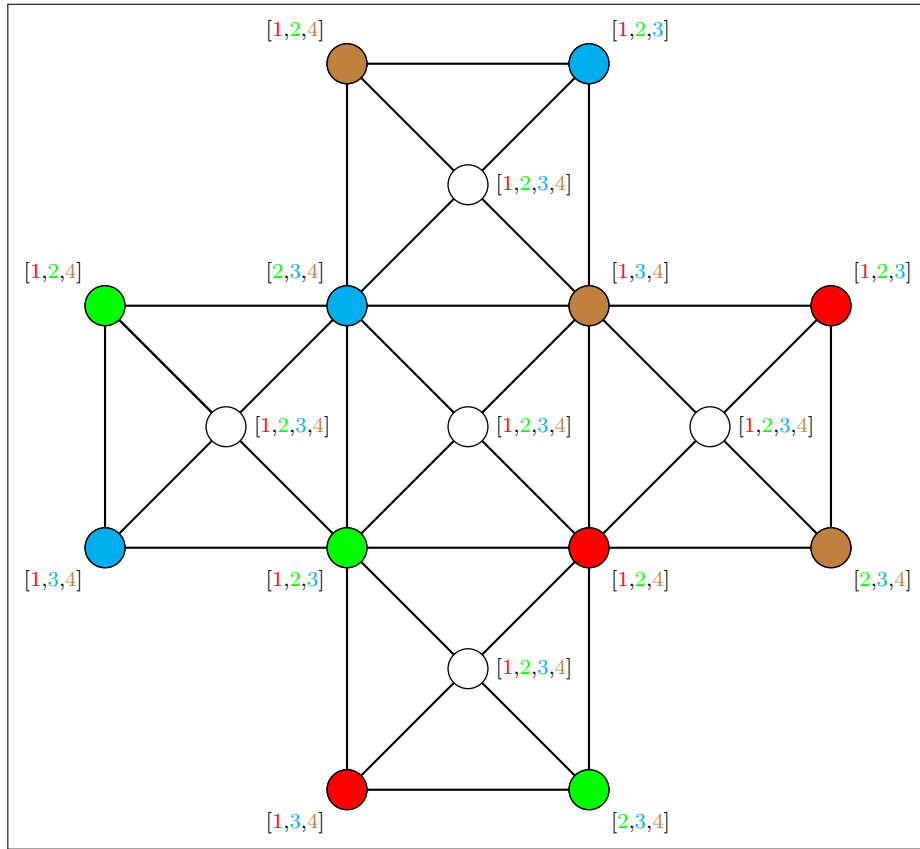


Abbildung 4.2

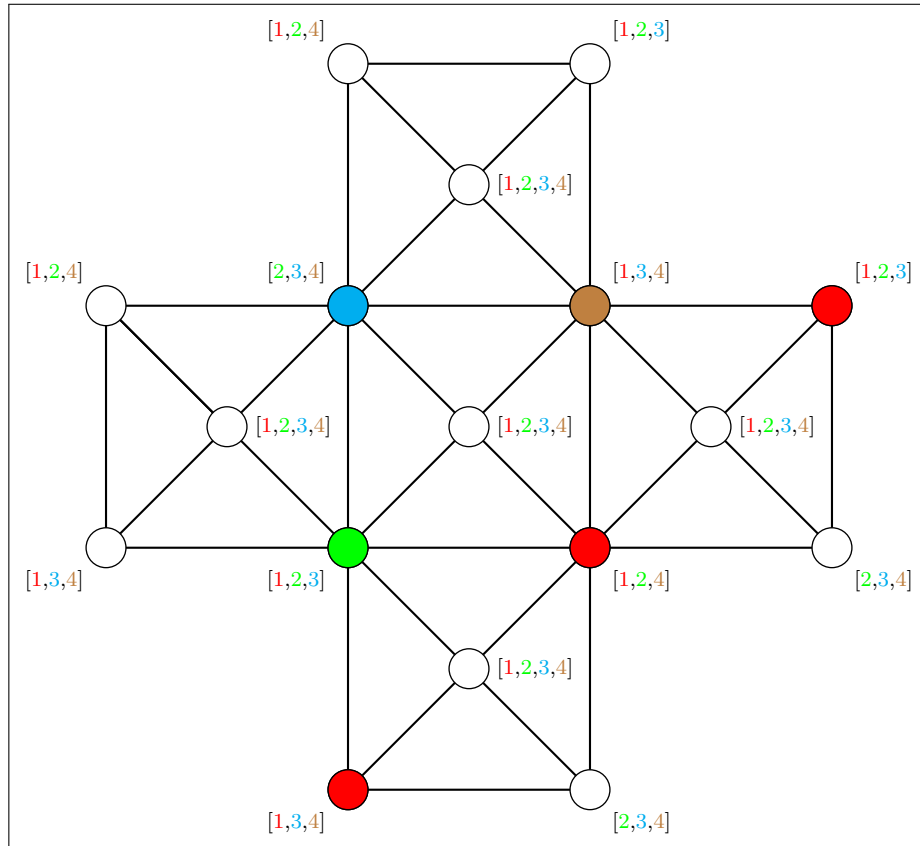
Der Graph  $H$  im Beispiel von Satz 7. Er ist nicht korrekt färbbar, solange den Listen der äußeren Knoten nicht eine weitere Farbe hinzugefügt wird. Denn jede Färbung von  $H$ , die die angegebenen Listen respektiert, bricht die vis-à-vis-Eigenschaft einer der  $I$ -Instanzen in  $H$ .

#### 4 Färbung planarer Graphen





#### 4 Färbung planarer Graphen



**Satz 8.** Jeder planare Graph ist 5-Listenfärbbar

*Beweis.* Vollständige Induktion über  $|V| = n$ . Es wird eine stärkere Behauptung bewiesen.

*Behauptung (\*).* Falls  $G$  planar ist und

- jede innere Facette ein Dreieck (bezüglich einer beliebigen planaren Einbettung) bildet
- die äußere Facette durch einen Kreis  $C = v_1, v_2, \dots, v_k, v_1$  begrenzt ist
- $v_1$  mit der Farbe 1 und  $v_2$  mit der Farbe 2 gefärbt ist
- jedem Knoten eine Liste mit mindestens drei Farben zugeordnet ist
- jedem Knoten in  $G - C$  eine Liste mit mindestens fünf Farben zugeordnet ist

#### 4 Färbung planarer Graphen

dann ist  $G$  unter Berücksichtigung der Listen und Farben von  $v_1$  und  $v_2$  korrekt färbbar.

Offensichtlich folgt aus (\*) die 5-Listenfärbbarkeit.

**Induktionsschritt** Sei  $|V| \geq 4$ ,  $G$  planar und  $C$  der Kreis, der die äußere Facette begrenzt.

*Fall 1.*  $C$  enthält Sehne  $\{v, w\}$

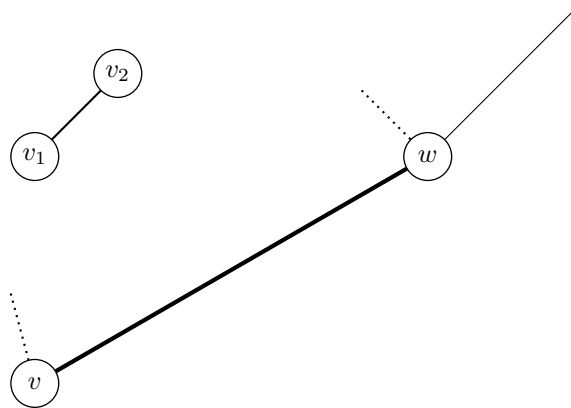


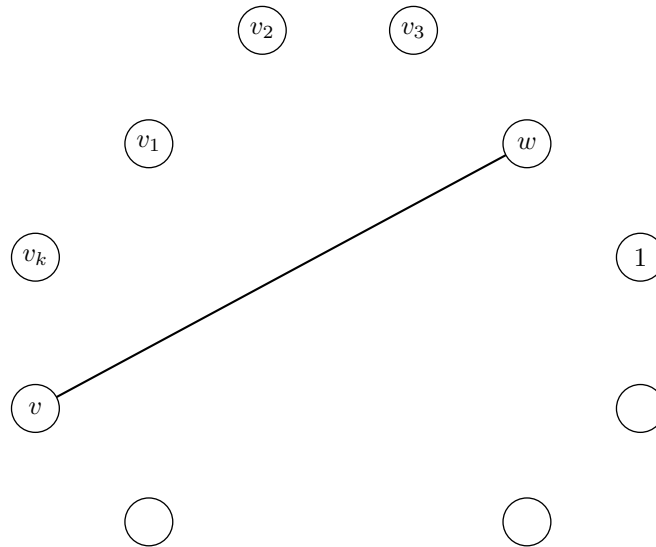
Abbildung 4.3

Fall 1 im Beweis von Satz 8:  $G$  enthält eine Sehne  $\{v, w\}$ . Sie trennt  $G$  eindeutig in  $G_1$  und  $G_2$ .

$\{v, w\}$  liegt auf zwei eindeutig bestimmten Kreisen  $C_1$  und  $C_2$  welche eindeutig zwei Graphen  $G_1, G_2$  induzieren.  $v_1$  und  $v_2$  liegen, da sie benachbart sind, im selben Teilgraphen (oder sind selbst  $v$  oder  $w$ ). O.B.d.A seien  $v_1$  und  $v_2$  in  $G_1$ . Die Induktionsvoraussetzung kann auf  $G_1$  angewendet werden. Danach kann die Induktionsvoraussetzung auf  $G_2$  mit  $v$  und  $w$  in den Rollen von  $v_1$  und  $v_2$  und den Farben aus der Färbung von  $G_1$  angewandt werden.

*Fall 2.*  $G$  enthält keine Sehne

#### 4 Färbung planarer Graphen



Die Nachbarn  $v_{k-1}, u_1, \dots, u_l, v_1$  von  $v_k$  bilden einen Weg  $P$ , da all inneren Facetten von  $G$  Dreiecke sind.  $(G - v_k) \cup P =: C'$  ist wiederum ein Kreis. Reserviere nun zwei Farben (die nicht die Farbe von  $v_1$  sind) aus der Liste von  $v_k$  und streiche sie ggf. aus den Listen von  $u_1, \dots, u_l$ . Sei  $G'$  der durch  $C'$  induzierte Subgraph von  $G$ .  $G'$  erfüllt die Induktionsvoraussetzung. Höchstens eine der reservierten Farben wird für  $v_{k-1}$  verbraucht. Also lässt sich  $v_k$  auch korrekt färben.

■

# 5 Separatoren von planaren Graphen

## 5.1 Planar Separator Theorem

---

**Definition 10.** Sei  $G = (V, E)$  ein Graph.  $S \subset V$  heißt **SEPARATOR** von  $G$  wenn der durch  $V \setminus S$  induzierte Subgraph  $G_S$  von  $G$  nicht zusammenhängend ist.  $S$  **TRENNT** zwei Knoten  $u, v \in V \setminus S$ , falls sich  $u$  und  $v$  in  $G_S$  in verschiedenen Zusammenhangskomponenten befinden.

---

**Problem 3** (Min Bounded Separator Problem). Sei  $G = (V, E)$  ein Graph und  $\frac{1}{2} \leq \alpha < 1$ . Finde eine Partition  $V_1, V_2, S$  von  $V$  wobei  $S$  ein Separator minimaler Kardinalität und  $|V_1| \leq \alpha |V|, |V_2| \leq \alpha |V|$  ist. Idealerweise ist dabei  $\alpha = \frac{1}{2}$ .

---

**Definition 11.** Zu einem Graphen  $G = (V, E)$  heißt der durch eine von  $w \in V$  ausgehende Breitensuche in  $G$  induzierte Baum  $T(G) = (V, E(T))$  der **AUFGE-SPANNT** BAUM beziehungsweise **BFS-BAUM** von  $G$  mit Wurzel  $w$ . Die Länge des Weges von  $v \in V$  zu  $w$  in  $T$  wird **HÖHE** oder **LEVEL** von  $v$  genannt. Zu einem Level  $i$  bezeichne  $s_i$  die Menge der Knoten mit Level  $i$ .

---

*Bemerkung.* Jedes  $s_i, i > 0$  ist ein Separator in  $G$ , da in einem BFS-Baum nur Kanten zwischen Knoten aus demselben oder aus aufeinander folgenden Leveln existiert.

**Lemma 6.** Sei  $G = (V, E)$  ein planarer, zusammenhängender Graph mit  $|V| \geq 5$  und  $T = (V, E(T))$  ein aufspannender Baum von  $G$  mit Wurzel  $w$  und Höhe  $h$ . Dann kann  $V$  so in drei Mengen  $V_1, V_2, S$  partitioniert werden, dass

1.  $|V_1|, |V_2| \leq \frac{2}{3}n$
2.  $S$  ein Separator ist, der  $V_1$  und  $V_2$  trennt
3.  $|S| \leq 2h + 1$

*Beweis.* Trianguliere  $G$ . Eine Nichtbaumkante  $\{x, y\} \in E \setminus E(T)$  induziert einen Kreis  $K_{xy}$ , der von  $\{x, y\}$  aus über Baumkanten zum gemeinsamen Elternknoten von  $x$  und  $y$  maximalen Levels in  $T$  geschlossen wird. Jedes  $K_{xy}$  enthält maximal  $2h + 1$  Knoten.

Wähle also eine beliebige Nichtbaumkante  $\{x, y\}$ . Sei o.B.d.A.  $|\text{Inneres}(K_{xy})| > \frac{2}{3}n$ . Inneres( $K_{xy}$ ) wird nun systematisch verkleinert (Falls  $|\text{Äußeres}(K_{xy})| > \frac{2}{3}n$  wird das Äußere verkleinert).

## 5 Separatoren von planaren Graphen

$\{x, y\}$  begrenzt ein Dreieck in  $\text{Inneres}(K_{xy})$ , da  $G$  trianguliert ist. Sei  $xy$  dieses Dreieck. Sei o.B.d.A.  $\{x, t\}$  Nichtbaumkante.  $K_{xy}$  soll nun durch  $K_{xt}$  ersetzt werden.

*Fall 1.*  $\{y, t\}$  ist Baumkante

*Fall i.*  $t \notin K_{xy}$

$t$  ist in  $T$  also Kind von  $y$ .  $y$  bleibt somit in  $K_{xt}$ :

$$\begin{aligned} |\text{Äußeres}(K_{xt})| &= |\text{Äußeres}(K_{xy})| \\ |\text{Inneres}(K_{xt})| &= |\text{Inneres}(K_{xy})| - 1 \end{aligned}$$

*Fall ii.*  $t \in K_{xy}$ :

$y$  ist in  $T$  also Kind von  $t$ .  $y$  wird somit Äußeres:

$$\begin{aligned} |\text{Äußeres}(K_{xt})| &= |\text{Äußeres}(K_{xy})| + 1 \\ |\text{Inneres}(K_{xt})| &= |\text{Inneres}(K_{xy})| \end{aligned}$$

*Fall 2.*  $\{y, t\}$  ist Nichtbaumkante

o.B.d.A. ist  $|\text{Inneres}(K_{xt})| \geq |\text{Inneres}(K_{ty})|$  und somit  $|\text{Inneres}(K_{xt})| \geq \frac{1}{2} |\text{Inneres}(K_{xy})|$ :

$$\begin{aligned} |\text{Äußeres}(K_{xt})| &\leq n - \frac{1}{2} |\text{Inneres}(K_{xy})| \leq \frac{2}{3}n \\ |\text{Inneres}(K_{xt})| &\leq |\text{Inneres}(K_{xy})| - 1 \end{aligned}$$

In jedem Fall kann also  $\text{Inneres}(K_{xy})$  verkleinert werden, ohne dass  $\text{Äußeres}(K_{xy})$  zu groß wird. Die Ersetzungen können durchgeführt werden, bis

$$\frac{1}{3}n < |\text{Inneres}(K_{xy})| \leq \frac{2}{3}n$$

■

### Implementierung von Lemma 6 in $\mathcal{O}(n)$

Da  $G$  trianguliert ist, gilt nach dem Satz von Euler (Satz 3)  $f = 2n - 4$ . Das obige Verfahren endet somit nach höchstens  $f = 2n - 4$  Ersetzungen. Der Aufwand eines einzelnen Übergangs ist, abgesehen vom Vergleich  $|\text{Inneres}(K_{xt})| \geq |\text{Inneres}(K_{yt})|$  in Fall 2, in  $\mathcal{O}(1)$ . Nach Bestimmung der Nichtbaumkante  $\{x, y\}$  wird eine einmalige Vorabrechnung vorgenommen:

- Speichere für jeden Knoten  $v$  seine inzidente Baumkante und die Anzahl der Knoten im Unterbaum mit Wurzel  $v$ .
- markiere alle Knoten auf  $K_{xy}$
- berechne damit  $|\text{Äußeres}(K_{xy})|$  und  $|\text{Inneres}(K_{xy})|$

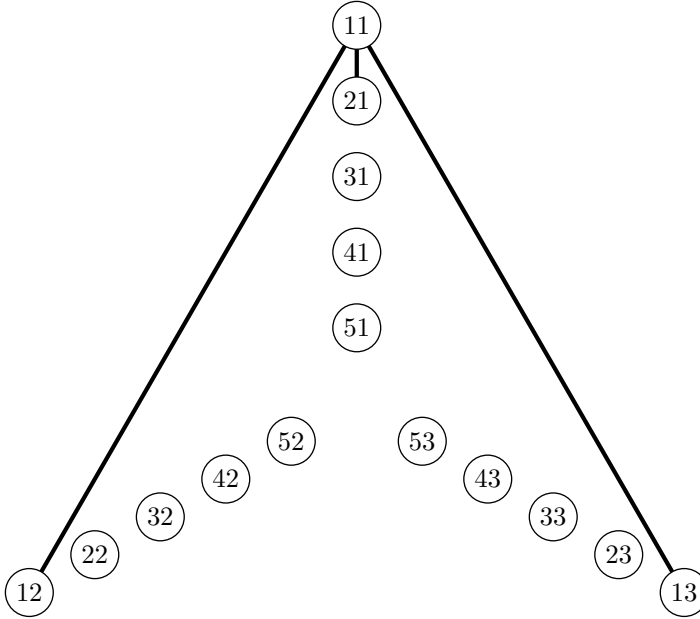


Abbildung 5.1

Um  $|\text{Inneres}(K_{xt})|$  und  $|\text{Inneres}(K_{yt})|$  zu berechnen, sei o.B.d.A.  $x$  links von  $y$ . Verfahre wie folgt:

- Laufe von  $t$  im Baum nach oben bis zum ersten Knoten  $v$  auf  $K_{xy}$  und berechne die Anzahlen  $a_t$  der Knoten *auf* dem Pfad  $\overline{tv}$ ,  $r_t$  der Knoten *rechts* vom Pfad  $\overline{tv}$  und  $l_t$  *links* vom Pfad  $\overline{tv}$ .
- Laufe von  $x$  und  $y$  *abwechselnd* im Baum nach oben bis zum *ersten Mal*  $v$  erreicht wird. Berechne dabei die Anzahl  $I_x$  der Knoten rechts des Pfades ab  $x$  bzw. die Anzahl  $I_y$  der Knoten links des Pfades ab  $y$ .

Sei o.B.d.A.  $v$  zuerst von  $x$  aus erreicht worden. Dann gilt

$$\begin{aligned}\text{Inneres}(K_{xt}) &= I_x + l_t \\ \text{Inneres}(K_{yt}) &= |\text{Inneres}(K_{xy})| - I_x - l_t - a_t\end{aligned}$$

*Bemerkung.* Es kann keine Baumkante geben, die aus dem Inneren von  $K_{xy}$  ins Äußere führt. Daher ist jeder Unterbaum, dessen Wurzel im Inneren von  $K_{xy}$  liegt, auch vollständig innerhalb von  $K_{xy}$ .

Da der Aufwand der Berechnung linear in der Größe des Teils von  $K_{xy}$  ist, der *nicht* weiter betrachtet wird, wird jeder Knoten höchstens einmal in die Berechnung aufgenommen. Der Aufwand bleibt also amortisiert in  $\mathcal{O}(n)$ .

## 5 Separatoren von planaren Graphen

**Satz 9** (Planar Separator Theorem). *Die Knotenmenge eines planaren, zusammenhängenden Graphen  $G = (V, E)$ ,  $|V| \geq 5$  kann so in  $V_1, V_2, S$  partitioniert werden, dass*

1.  $|V_1|, |V_2| \leq \frac{2}{3}n$
2.  $S$  Separator, der  $V_1$  und  $V_2$  trennt
3.  $|S| \leq 4\sqrt{n}$

*Diese Partitionierung kann in  $\mathcal{O}(n)$  konstruiert werden.*

*Beweis und Algorithmus.*

1. Trianguliere  $G$
2. Wähle Wurzel; berechne BFS-Baum
3. Bestimme Level  $\mu$  sodass  $\sum_{i=0}^{\mu-1} |s_i| \leq \frac{n}{2}$ , aber  $\sum_{i=0}^{\mu} |s_i| > \frac{n}{2}$   
 $m \leftarrow$  größtes Level kleiner als  $\mu$ , sodass  $|s_m| \leq \sqrt{n}$   
 $M \leftarrow$  kleinstes Level größer als  $\mu$ , sodass  $|s_M| \leq \sqrt{n}$   
 $h \leftarrow$  Höhe des BFS-Baums

$$A_1 := \bigcup_{i=0}^{m-1} s_i \qquad A_2 := \bigcup_{i=m+1}^{M-1} s_i \qquad A_3 := \bigcup_{i=M+1}^h s_i$$

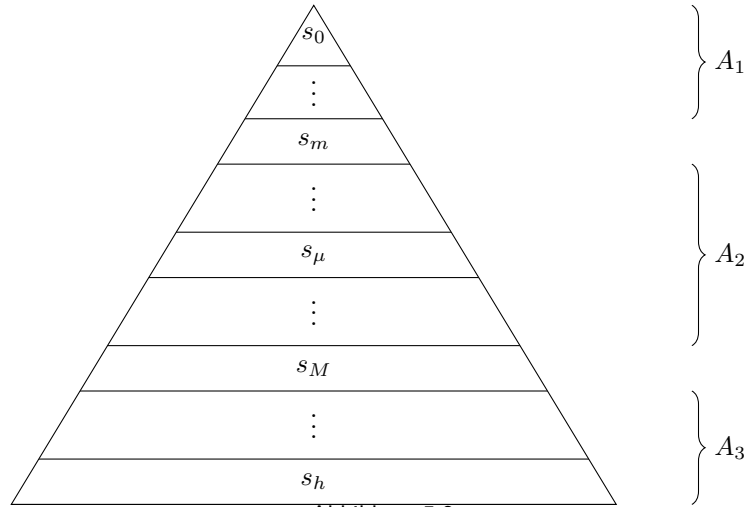


Abbildung 5.2

Schematische Einteilung des BFS-Baums im Algorithmus zu Satz 9

4. *Fall 1.*  $|A_2| \leq \frac{2}{3}n$

## 5 Separatoren von planaren Graphen

$$\begin{aligned} S &\leftarrow S_m \cup S_M \\ V_1 &\leftarrow \operatorname{argmax}_{A \in \{A_1, A_2, A_3\}} |A| \\ V_2 &\leftarrow V \setminus (S \cup V_1) \end{aligned}$$

Aufgrund der Wahl von  $m$  und  $M$  gilt  $|A_1| \leq \frac{n}{2}$  und  $|A_3| < \frac{n}{2}$ . Mit  $|A_2| \leq \frac{2}{3}n$  folgt

$$\begin{aligned} |V_1| &\leq \frac{2}{3}n \\ |V_2| &\leq n - |V_1| \end{aligned}$$

Da  $|V_1| = \max\{|A_1|, |A_2|, |A_3|\} \geq \frac{|V_2|}{2}$  gilt, gilt auch

$$|V_2| \leq n - |V_1| \leq \frac{2}{3}n$$

*Fall 2.*  $|A_2| > \frac{2}{3}n$

Verschmelze  $A_1 \cup s_m$  zu einem Knoten  $W$  und entferne  $s_M \cup A_3$ . Betrachte dadurch induzierten Graphen  $G' = (V', E')$  und entsprechenden BFS-Baum  $T'$ .

Dann gilt:  $V' = A_2 \cup \{W\}$  und  $|V'| = n' = |A_2| + 1$ . Die Höhe von  $T'$  ist höchstens  $\sqrt{n}$ , da  $\forall m < i < M : |s_i| > \sqrt{n}$

Nach Lemma 6 lässt sich in  $G'$  in  $\mathcal{O}(n)$  eine Partitionierung  $S', V'_1, V'_2$  finden, sodass

- a)  $|V'_1|, |V'_2| \leq \frac{2}{3}n'$
- b)  $S'$  ist Separator der  $V'_1$  und  $V'_2$  trennt
- c)  $|S'| \leq 2 \cdot h(T') \leq 2\sqrt{n}$

Setze  $S := (S' \cup S_m \cup S_M) \setminus \{W\}$ . Dann gilt:

- a)  $|S| \leq 4 \cdot \sqrt{n}$
- b)  $S$  ist Separator, der  $A_1, V'_1, V'_2, A_3$  trennt

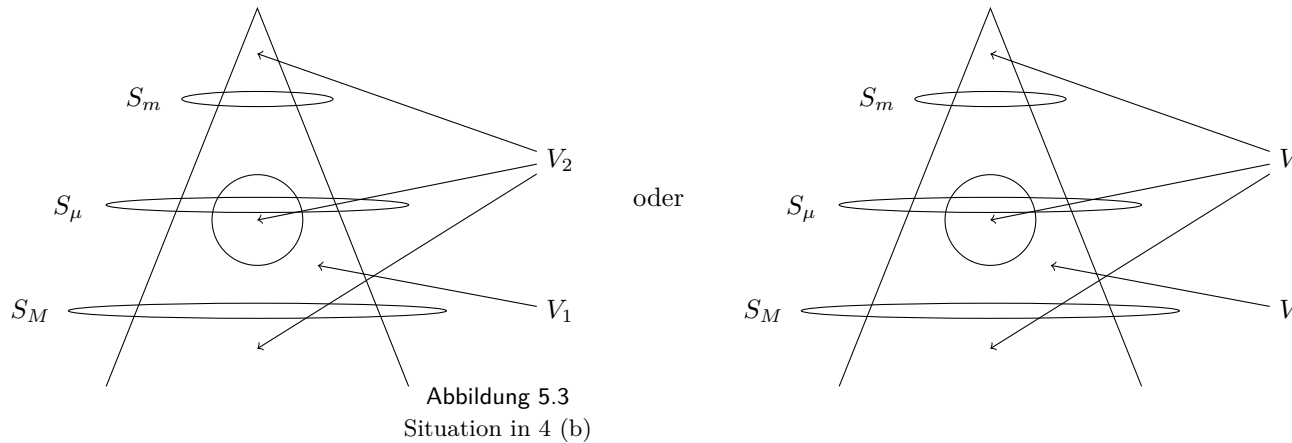
Wähle als  $V_1$  die größere Menge von  $V'_1 \setminus W$  und  $V'_2 \setminus W$  und den Rest von  $V$  als  $V_2$ . Falls  $|V'_1| = |V'_2|$ , wähle als  $V_1$  die Menge, die  $W$  nicht enthält. Da  $|A_2| > \frac{2}{3}n$  gilt:

$$\begin{aligned} |V_1| &\leq \frac{2}{3}n' \leq \frac{2}{3}n \\ |V_2| &\leq n - |V_1| \leq n - \frac{|A_2|}{2} \leq \frac{2}{3}n \end{aligned}$$

■



## 5 Separatoren von planaren Graphen



## 5.2 Anwendung des Planar Separator Theorems

Das Planar Separator Theorem (PST) wird oft in Algorithmen auf planaren Graphen verwendet. Ein generischer Divide-and-Conquer-Algorithmus unter Zuhilfenahme des PST sieht folgendermaßen aus:

**Algorithmus** (Generischer Divide-and-Conquer-Algorithmus mit PST).

- zerlege  $G = (V, E)$  mittels PST in  $G_1, G_2$  und Separator  $S$
- wende den Algorithmus rekursiv auf  $G_1$  und  $G_2$  an,  $G_1 = (V_1, E_1), |V_1| = n_1, G_2 = (V_2, E_2), |V_2| = n_2$
- berechne aus Lösungen zu  $G_1$  und  $G_2$  unter Berücksichtigung von  $S$  Lösung zu  $G$

Rekursionsabschätzung der Laufzeit  $T(n)$  eines solchen Algorithmus:

$$T(n) \leq T(n_1) + T(n_2) + \sqrt{n} \cdot f(n)$$

wobei  $f(n)$  die Laufzeit des eigentlichen Algorithmus ist und  $n_1, n_2 \leq \frac{2}{3}n, n_1 + n_2 \leq n$ .

### 5.2.1 Matching

**Problem 4** (Matching). Zu  $G = (V, E)$  heißt  $M \subseteq E$  MATCHING falls keine zwei Kanten in  $M$  den selben Endknoten haben. Ein Knoten  $v \in V$  heißt UNGEMATCHT falls  $v$  zu keiner Kante aus  $M$  inzident ist. Andernfalls heißt  $v$  GEMATCHT.

Gegeben ein Graph  $G = (V, E)$ , mit Kantengewichten  $w : E \rightarrow \mathbb{R}$ , wird das Matching mit maximalem Gewicht, also jenes Matching  $M$  für das  $w(M) := \sum_{l \in M} w(l)$  unter allen Matchings in  $G$  maximal ist, gesucht.

---

**Definition 12.** Gegeben ein Graph  $G = (V, E)$  und Matching  $M \subseteq E$ , wird ein einfacher Weg (oder einfacher Kreis), dessen Kanten abwechselnd in  $M$  und  $E \setminus M$  liegen, **ALTERNIERENDER WEG** bezüglich  $M$  genannt.

Sei  $P$  ein alternierender Weg bezüglich eines Matchings  $M$ , der entweder ein Kreis gerader Länge ist oder ein Weg ist, dessen erste und letzte Kante in  $M$  liegt oder inzident zu einem ungematchten Knoten, ist.  $P$  heißt **ERHÖHEND** bezüglich  $M$ , falls

$$\sum_{l \text{ auf } P, l \in E \setminus M} w(l) > \sum_{l \text{ auf } P, l \in M} w(l)$$


---

*Bemerkung.* Sei  $M$  ein Matching in  $G = (V, E)$ . Sei  $P$  ein erhöhender Weg bezüglich  $M$ . Dann ist  $M' := (M \setminus P) \cup (P \setminus M)$  ein Matching in  $G$  mit  $w(M') > w(M)$ .

---

**Definition 13.** Für zwei Mengen  $A$  und  $B$  sei die **GEOMETRISCHE DIFFERENZ**  $\Delta$  von  $A$  und  $B$  definiert als

$$A \Delta B := (A \cup B) \setminus (A \cap B)$$


---

**Lemma 7.** Sei  $G = (V, E)$  ein Graph mit Kantengewichten  $w : E \rightarrow \mathbb{R}$  und  $M$  ein Matching in  $G$ .  $w(M)$  ist genau dann maximal, falls es in  $G$  keinen erhöhenden Weg bezüglich  $M$  gibt.

*Beweis.*

„ $\implies$ “ klar.

„ $\impliedby$ “ Angenommen  $M$  sei ein nicht maximales Matching in  $G$ . Dann existiert ein Matching  $M^*$  mit  $w(M^*) > w(M)$ . Sei  $G_{M \Delta M^*}$  der Subgraph von  $G$ , der durch  $M \Delta M^*$  induziert wird.  $G_{M \Delta M^*}$  besteht ausschließlich aus einfachen Kreisen und einfachen Wegen, da der Knotengrad jedes Knotens maximal 2 sein kann. Falls keiner der Kreise erhöhend bezüglich  $M$  ist, existiert ein inklusionsmaximaler, bezüglich  $M$  alternierender Weg  $P$ , für den  $w(P \cap M^*) > w(P \cap M)$ . Entweder gehören beide Endknoten von  $P$  zu  $M$  oder es gehört mindestens ein Endknoten zu  $M^*$  und damit (da wir in  $G_{M \Delta M^*}$  sind) nicht in  $M$ . In beiden Fällen ist  $P$  bezüglich  $M$  erhöhend. ■

## 5 Separatoren von planaren Graphen

**Lemma 8.** Sei  $G = (V, E)$  ein Graph mit Kantengewichten  $w : E \rightarrow \mathbb{R}$  und  $v \in V$  ein Knoten in  $G$ . Sei  $M$  ein Matching maximalen Gewichts in  $G - v$ . Dann gilt:

1. Falls es in  $G$  keinen erhöhenden Weg bezüglich  $M$  mit Endknoten  $v$  gibt, so ist  $M$  auch maximales Matching in  $G$ .
2. Falls es in  $G$  ein bezüglich  $M$  erhöhenden Weg  $P$  mit Endknoten  $v$  gibt und  $w(P \cap E \setminus M) - w(P \cap M)$  maximal unter allen solchen Wegen ist, dann ist  $M \Delta P$  ein Matching maximalen Gewichts in  $G$ .

*Beweis.* Sei  $M$  ein Matching maximalen Gewichts in  $G - v$ .  $M$  ist auch ein Matching in  $G$ . Ein erhöhender Weg bezüglich  $M$  muss  $v$  als Endknoten enthalten. Sei  $M^*$  ein maximales Matching in  $G$ .  $M \Delta M^*$  ist eine Menge von alternierenden einfachen Kreisen und Wegen bezüglich  $M$  und  $M^*$ . Ein erhöhender Weg bezüglich  $M$  in  $G_{M \Delta M^*}$  ist auch ein erhöhender Weg bezüglich  $M$  in  $G$ .  $G_{M \Delta M^*}$  kann höchstens einen erhöhenden Weg  $P^*$  bezüglich  $M$  mit Endknoten  $v$  enthalten, da  $v$  in  $M^*$  sonst zwei inzidente Kanten hätte. Nun gilt  $w(M) - w(P^* \cap M) = w(M^*) - w(P^* \cap M^*)$ . Das Gewicht des Matchings  $M'$ , das durch erhöhenden Weg  $P^*$  induziert wird, ist  $w(M') = w(M) - w(P^* \cap M) + w(P^* \cap E \setminus M) = w(M) - w(P^* \cap M) + w(P^* \cap M^*) = w(M^*)$ . ■

**Algorithmus 1** (Divide-and-Conquer-Algorithmus für Matching in planaren Graphen mit PST).

1. Falls  $G$  höchstens 5 Knoten enthält bestimme das maximale Matching direkt.
2. Ansonsten zerlege  $G$  mit dem Planar Separator Theorem in  $G_1, G_2$  und Separator  $S$ . Wende diesen Algorithmus rekursiv auf  $G_1$  und  $G_2$  an, die entsprechenden maximalen Matchings seien  $M_1$  und  $M_2$ .  $V' := V \setminus S$ .
3. Solange bis  $S \neq \emptyset$ : wähle  $v \in S$ , setze  $S \leftarrow S \setminus \{v\}$  und  $V' \leftarrow V' \cup \{v\}$ . Wende Lemma 8 an.

Die Laufzeit  $T(n)$  von Algorithmus 1 berechnet sich als:

$$\begin{aligned} T(n_0) &= c_0 \\ T(n) &\leq T(c_1 \cdot n) + T(c_2 \cdot n) + c_3 \sqrt{n} \cdot f(n) \end{aligned}$$

mit  $c_1, c_2 < \frac{2}{3}$ ,  $c_1 + c_2 < 1$  und  $f(n)$  die Laufzeit für das Erweitern der Matchings nach Lemma 8. Per Induktion kann gezeigt werden, dass falls

$$\begin{aligned} f(n) \in \Theta(n) &\implies T(n) \in \mathcal{O}\left(n^{\frac{3}{2}}\right) \\ f(n) \in \Theta(n \log n) &\implies T(n) \in \mathcal{O}\left(n^{\frac{3}{2}} \log n\right) \end{aligned}$$

Falls  $G$  gewichtet ist, ist  $f(n) \in \Theta(n \log n)$ , in einem ungewichteten Graphen ist  $f(n) \in \Theta(n)$

### 5.2.2 Mixed Max Cut

---

**Definition 14.** Sei  $G = (V, E)$  ein Graph mit Kantengewichten  $w : E \rightarrow \mathbb{R}$  und  $S \subseteq E$  ein Schnitt in  $G$ . Dann heißt  $\sum_{l \in S} w(l) =: w(S)$  GEWICHT von  $S$ .

---

**Problem 5** (Min Cut). Sei  $G = (V, E)$  ein Graph mit Kantengewichtsfunktion  $w : E \rightarrow \mathbb{R}^+$ . Gesucht wird der Schnitt  $S$  ( $S \neq \emptyset$ ) mit minimalem Gewicht  $w(S)$ . Das Min-Cut-Problem ist auf einem beliebigen Graphen in  $\mathcal{O}(n \cdot m + n^2 \cdot \log n)$  lösbar.

**Problem 6** (Max Cut). Sei  $G = (V, E)$  ein Graph mit Kantengewichtsfunktion  $w : E \rightarrow \mathbb{R}^+$ . Gesucht wird der Schnitt  $S$  ( $E \neq S \neq \emptyset$ ) mit maximalem Gewicht  $w(S)$ . Das Max-Cut-Problem ist auf einem beliebigen Graphen  $\mathcal{NP}$ -schwer.

**Problem 7** (Mixed Max Cut). Sei  $G = (V, E)$  ein Graph mit Kantengewichtsfunktion  $w : E \rightarrow \mathbb{R}$ . Gesucht wird der Schnitt  $S$  ( $E \neq S \neq \emptyset$ ) mit maximalem Gewicht  $w(S)$ . Das Min-Cut-Problem ist äquivalent zum Mixed-Max-Cut-Problem bei dem  $w(l)$  durch  $-w(l)$  ersetzt wird. Da Max-Cut ein Spezialfall von Mixed-Max-Cut ist, ist Mixed-Max-Cut natürlich ebenfalls  $\mathcal{NP}$ -schwer. Auf planaren Graphen existiert ein Algorithmus für Mixed-Max-Cut in planaren Graphen mit Laufzeit  $\mathcal{O}(n^{\frac{3}{2}} \log n)$ .

Der Algorithmus auf planaren Graphen nutzt die Dualität zwischen einem Schnitt in einem planaren Graph  $G = (V, E)$  und einer Menge von Kreisen in seinem Dualgraph  $G^* = (V^*, E^*)$  (siehe Lemma 1). Das Mixed-Max-Cut-Problem in  $G$  ist äquivalent zum Mixed-Max-Kreis-Problem in  $G^*$  mit  $w : E^* \rightarrow \mathbb{R}$ ;  $w(e^*) = w(e)$  wobei  $e^*$  die Dualkante zu  $e \in E$  ist.

---

**Definition 15.** Sei  $G = (V, E)$  ein Graph. Eine Kantenmenge  $E' \subseteq E$  heißt GERADE, falls im durch sie induzierten Subgraph von  $G$  alle Knoten geraden Grad haben.

Ein Graph  $G = (V, E)$  heißt EULERSCH wenn jede Zusammenhangskomponente von  $G$  einen Euler-Kreis enthält, das heißt einen Kreis, der jede Kante genau einmal enthält.

---

*Bemerkung.* Für einen beliebigen Graph  $G = (V, E)$  gilt:

- $G$  ist eulersch
- $\iff E$  ist eine gerade Menge
- $\iff G$  besteht aus einer kantendisjunkten Vereinigung von einfachen Kreisen.

**Problem 8** (Mixed Max Kreis). Gegeben ein Graph  $G = (V, E)$  mit Kantengewichten  $w : E \rightarrow \mathbb{R}$  wird eine nichtleere, gerade Menge  $E' \subseteq E$  mit maximalem  $w(E')$  gesucht.

---

**Definition 16.** Sei  $G = (V, E)$  gerade. Ein Matching  $M$  in  $G$  mit  $|M| = \frac{|V|}{2}$  heißt PERFEKT.

---

**Algorithmus 2** (Mixed-Max-Cut auf planaren Graphen).

1. *trianguliere  $G$  ( $\mathcal{O}(n)$ )*
2. *berechne den Dualgraph  $G^*$  zu  $G$  und eine beliebige planare Einbettung. Definiere  $w : E^* \rightarrow \mathbb{R}$  durch  $w(e^*) := w(e)$  ( $\mathcal{O}(n)$ )*
3. *konstruiere aus  $G^*$  einen Graphen  $G'$  sodass ein perfektes Matching minimalen Gewichts in  $G'$  eine Gerade Menge  $M^*$  maximalen Gewichts in  $G^*$  induziert ( $\mathcal{O}(n)$ )*
4. *berechne perfektes Matching  $M$  minimalen Gewichts in  $G'$  und induzierte gerade Menge  $M^*$  in  $G^*$  ( $\mathcal{O}(n^{\frac{3}{2}} \log n)$ )*
5. *Sonderfall: Falls  $M$  eine leere Menge  $M^*$  induziert, berechne aus  $M$  eine nicht-triviale gerade Menge in  $G^*$  mit maximalem Gewicht.*

**Zu Schritt 1 und 2** Wenn  $G$  trianguliert wird, erhält jede eingefügte Kante das Gewicht 0. Deshalb ändert sich durch das Triangulieren nicht das Ergebnis. Da  $G$  trianguliert ist haben alle Knoten in  $G^*$  Grad 3, d.h.  $G^*$  ist 3-regulär. Jede gerade Menge in  $G^*$  ist somit sogar eine Menge von knotendisjunkten Kreisen.

**Zu Schritt 3**  $G^*$  ist 3-regulär. In  $G^*$  kann somit jeder Knoten  $v^*$  durch ein Dreieck  $\Delta_{v^*}$  ersetzt werden. Sei  $G'$  der resultierende Graph. In  $G'$  kann nun ein perfektes Matching berechnet werden.

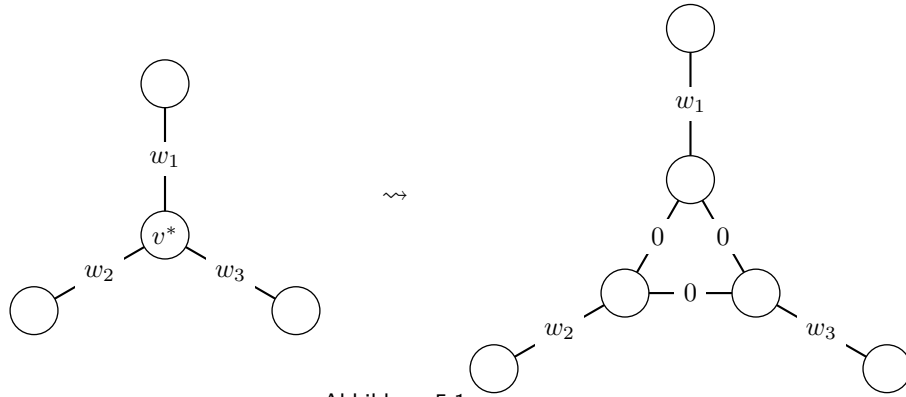


Abbildung 5.1

Ersetzung in Schritt 3 von Algorithmus 2.

*Bemerkung.*  $G'$  hat eine gerade Anzahl an Knoten, da  $G^*$  3-regulär ist und damit  $2m = 3n$  gilt.

**Lemma.**

1. Ist  $M$  ein perfektes Matching in  $G'$ , so induziert  $E' \setminus M$  eine gerade Menge in  $G^*$ .
2. Ist  $E_0^*$  eine gerade Menge in  $G^*$ , so induziert  $M^* = E^* \setminus E_0^*$  ein perfektes Matching in  $G'$ .

*Beweis.* Sei  $M$  ein perfektes Matching in  $G'$ . Dann hat jeder Knoten in  $G'$  eine inzidente Kante in  $M$ . Für jeden Knoten  $v^* \in V^*$  und sein korrespondierende Dreieck  $\Delta_{v^*}$  in  $G'$  gibt es hierfür aber nur zwei Möglichkeiten: entweder liegt *keine* Kanten von  $\Delta_{v^*}$  in  $M$  oder *nur eine*. Im ersten Fall sind alle zu  $v^*$  inzidenten Kanten in  $M$ , in letzterem eine. In beiden Fällen liegen eine gerade Anzahl an zu  $v^*$  inzidenten Kanten in  $E' \setminus M$ : Entweder 0 oder 2.

Ist wiederum  $E_0^* \neq \emptyset$  eine gerade Menge in  $G^*$ , so sind zu jedem  $v^* \in V^*$  zwei oder keine inzidenten Kanten in  $E_0^*$ . Somit sind eine oder drei zu  $v^*$  inzidenten Kanten in  $M^* = E^* \setminus E_0^*$ .  $M^*$  kann zu  $M$  erweitert werden, im ersten Fall die Kante, die die beiden ungematchten Knoten in  $\Delta_{v^*}$  verbindet, im zweiten Fall keine Kante hinzugefügt wird. Dann hat jedes  $v' \in V'$  genau eine inzidente Kante in  $M$ .  $M$  ist somit ein perfektes Matching in  $G'$ . ■

**Zu Schritt 4**

*Bemerkung.* Sei  $G = (V, E)$  ein Graph mit Kantengewichtsfunktion  $w : E \rightarrow \mathbb{R}$ .  $M \subseteq E$  ist genau dann ein perfektes Matching minimalen Gewichts in  $G$  falls  $M$  ein perfektes Matching maximalen Gewichts in  $G$  bezüglich  $\bar{w} : E \rightarrow \mathbb{R}$  mit  $\bar{w}(e) = W - w(e)$  ist.  $W \in \mathbb{R}$  sei hierbei eine geeignete Konstante.

Die Forderung, dass das Matching minimalen Gewichts, das in  $G'$  berechnet wird, perfekt ist, kann durch die Kantengewichtsfunktion kodiert werden. Sei hierzu  $w_{\max} := \max_{e \in E'} w(e)$ ,  $w_{\min} := \min_{l \in E'} w(l)$ . Für ein perfektes Matching  $M$  in  $G'$  gilt

$$\bar{w}(M) = \sum_{e \in M} \bar{w}(e) = \frac{n}{2} \cdot W - \sum_{e \in M} w(e) \geq \frac{n}{2} (W - w_{\max})$$

Für ein nicht perfektes Matching  $M'$  in  $G'$  gilt

$$\bar{w}(M') \leq \left(\frac{n}{2} - 1\right) \cdot (W - w_{\min})$$

Wähle also  $W$  so, dass  $\frac{n}{2} (W - w_{\max}) > \left(\frac{n}{2} - 1\right) (W - w_{\min})$ , also  $W > \frac{n}{2} (w_{\max} - w_{\min}) + w_{\min}$ . Dann gilt für ein perfektes Matching maximalen Gewichts  $M$ , dass  $\bar{w}(M) > \bar{w}(M')$  für jedes nicht perfekte Matching  $M'$ .

**Zu Schritt 5** Induziert das perfekte Matching  $M^*$  minimalen Gewichts in  $G'$  eine leere Menge an Kreisen in  $G^*$ , so gibt es keinen positiven Kreis in  $G^*$ . Nun berechnen wir in  $\mathcal{O}\left(n^{\frac{3}{2}} \log n\right)$  in  $G^*$  einen einfachen negativen Kreis maximalen Gewichts unter Verwendung des Planar Separator Theorems

## 5 Separatoren von planaren Graphen

**Algorithmus 3** (Negativer Kreis maximalen Gewichts).

1. *Berechne entsprechend des Planar Separator Theorems eine Partitionierung  $V_1, V_2, S$  von  $G$ .*
2. *Falls  $G$  nur einen Kreis enthält, gebe diesen aus. Berechne ansonsten rekursiv negative Kreise maximalen Gewichts in den von  $V_1$  und  $V_2$  induzierten Subgraphen  $G_1$  und  $G_2$  von  $G$ .*
3. *Gib den Kreis maximalen Gewichts unter allen konstruierten Kreisen aus.*

**Schritt 2:** Die Graphen  $G_1$  und  $G_2$  sind nicht notwendigerweise 3-regulär. Die 3-Regularität kann jedoch durch rekursives Entfernen von Knoten mit Grad 1 und Kontrahieren von Knoten mit Grad 2 (unter Addition der Kantengewichte) hergestellt werden. Dies beeinflusst nicht das Gewicht des schwersten Kreises.

## 6 Das Menger-Problem

---

**Definition 17.** Gegeben ein Graph  $G = (V, E)$ ,

$$\mathcal{K}_G(u, v) := \begin{cases} \min_{S \subseteq V; S \text{ trennt } u \text{ und } v} |S| & \text{falls } \{u, v\} \notin E \\ |V| - 1 & \text{sonst} \end{cases}$$

wird als KNOTENZUSAMMENHANG der Knoten  $u, v \in V$  bezeichnet. Dies induziert den Knotenzusammenhang  $\mathcal{K}(G)$  des Graphen  $G$ :

$$\mathcal{K}(G) := \min_{u, v \in V} \mathcal{K}_G(u, v) = \min_{S \subseteq V \text{ Separator}} \{|S|, |V| - 1\}$$

Analog wird

$$\lambda_G(u, v) := \min_{S \subseteq E; S \text{ trennt } u \text{ und } v} |S|$$

als KANTENZUSAMMENHANG der Knoten  $u, v \in V$  bezeichnet. Dies induziert ebenfalls den Kantenzusammenhang  $\lambda(G)$  des Graphen  $G$ :

$$\lambda(G) := \min_{u, v \in V} \lambda_G(u, v) = \min_{S \subseteq E \text{ Schnitt}} |S|$$

$G$  heißt  $k$ -FACH KNOTEN- beziehungsweise KANTENZUSAMMENHANG wenn  $k \leq \mathcal{K}(G)$  beziehungsweise  $k \leq \lambda(G)$ .

---



---

**Definition 18.** Zwei Wege in einem Graphen  $G$  heißen KNOTENDISJUNKT, falls sie außer den Endknoten keinen gemeinsamen Knoten enthalten und KANTENDISJUNKT, falls sie keine gemeinsame Kante enthalten.

---



**Satz 10** (Satz von Menger). Sei  $G = (V, E)$  ein Graph und  $s, t \in V$  Knoten in  $G$  im knotendisjunkten Fall sei zusätzlich  $\{s, t\} \notin E$ . Dann gilt:

- $\kappa_G(s, t) \geq k \iff$  es existieren  $k$  knotendisjunkte Wege zwischen  $s$  und  $t$  in  $G$ .
- $\lambda_G(s, t) \geq k \iff$  es existieren  $k$  paarweise kantendisjunkte Wege zwischen  $s$  und  $t$  in  $G$ .

*Beweis.* Folklore. ■

**Problem 9** (Menger-Problem). Sei  $G = (V, E)$  ein Graph. Finde die maximale Anzahl von konten- beziehungsweise kantendisjunkten Wegen zwischen zwei Knoten  $s, t \in V$  von  $G$ .

## 6.1 Kantendisjunkttes Menger-Problem

**Algorithmus 4** (Linearzeitalgorithmus für das kantendisjunkte Menger-Problem auf planaren Graphen). Der Algorithmus basiert auf *Right-First-DFS*.

*Spezialfall:*  $s$  und  $t$  liegen beide auf der äußeren Facette. Hier können die Wege direkt mit *Right-First-DFS* bestimmt werden.

*Im Allgemeinen:* Sei  $G = (V, E)$  ein Graph und  $s, t \in V$  zwei Knoten aus  $G$ . Bette  $G$  so planar ein, dass  $t$  auf der äußeren Facette liegt.

1. Ersetze  $G$  durch gerichteten Graphen  $\vec{G} = (V, \vec{E})$ , bei dem  $\{u, v\} \in E$  durch  $\{(u, v), (v, u)\} \subseteq \vec{E}$  ersetzt wird. ( $\mathcal{O}(n)$ )
2. Berechne in  $\vec{G}$  geeignete kantendisjunkte Kreise  $\vec{C}_1, \dots, \vec{C}_l$  und den Graphen  $\vec{G}_C$  der aus  $\vec{G}$  entsteht, indem alle Kanten, die auf einem  $\vec{C}_i$ ,  $1 \leq i \leq l$  liegen, „umgedreht“ werden. ( $\mathcal{O}(n)$ )
3. Führe eine *Right-First-DFS* auf  $\vec{G}_C$  aus um die maximale Anzahl kantendisjunkter Wege zwischen  $s$  und  $t$  zu berechnen. ( $\mathcal{O}(n)$ )
4. Berechne aus den  $k$  gerichteten kantendisjunkten Wegen in  $\vec{G}_C$   $k$  kantendisjunkte Wege in  $G$ . ( $\mathcal{O}(n)$ )

### Zu Schritt 1

**Lemma 9.** Seien  $\vec{p}_1, \dots, \vec{p}_r$  kantendisjunkte, gerichtete  $s$ - $t$ -Wege in  $\vec{G}$ . Dann enthält

$$P := \{\{u, v\} \in E \mid \text{genau eine der beiden gerichteten Kanten } (u, v) \text{ und } (v, u) \text{ gehören zu einem der Wege } p_i\}$$

$r$  kantendisjunkte  $s$ - $t$ -Wege in  $G$ .

## 6 Das Menger-Problem

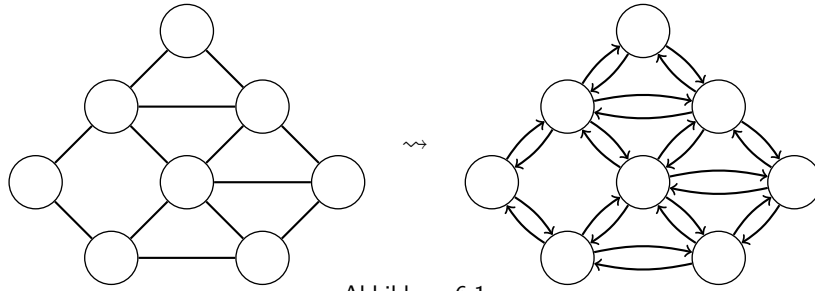


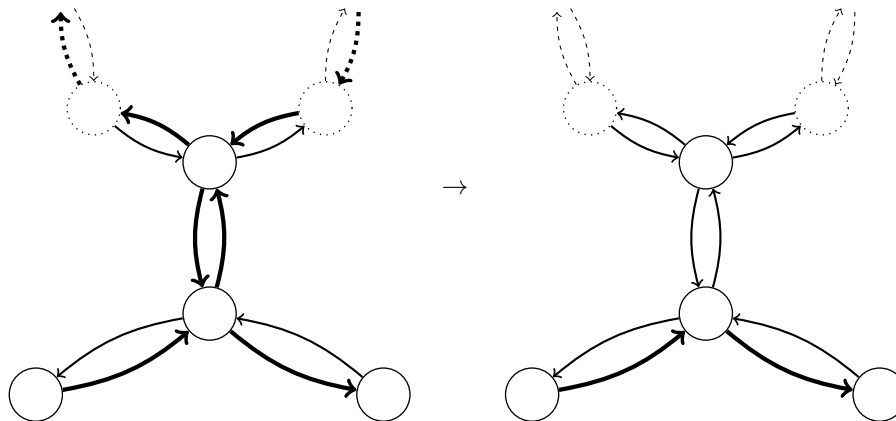
Abbildung 6.1

Der Übergang von  $G$  zu  $G'$  im ersten Schritt von Algorithmus 4.

*Beweis.* Betrachte  $\{u, v\} \in E$  mit  $(u, v), (v, u)$  belegt durch  $p_1, \dots, p_r$ .

*Fall 1.*  $(u, v)$  und  $(v, u)$  sind durch den selben Kreis  $p_i$  belegt. Dann bildet der Teil zwischen den beiden Kanten einen Kreis, der in  $G$  weggelassen werden kann.

*Fall 2.*  $(u, v)$  ist durch  $p_i$  und  $(v, u)$  durch  $p_j$ ,  $i \neq j$  belegt. Dann können im gerichteten Graphen zwei Wege  $p_a$  und  $p_b$  gewonnen werden, indem jeder der Wege  $p_i, p_j$  bis zu  $u$  bzw.  $v$  verfolgt wird, und danach auf den jeweils anderer Weg gewechselt wird. ■



(a) Ein Kreis kann weggelassen werden.

Abbildung 6.2

Beide Fälle von Lemma 9.

*Schlussfolgerung.* Die maximale Anzahl kantendisjunkter  $s$ - $t$ -Wege in  $\vec{G}$  induziert die maximale Anzahl kantendisjunkter Wege in  $G$ .

**Zu Schritt 2**  $\vec{G}_C$  kann wie folgt in Linearzeit konstruiert werden: Berechne einfache, gerichteten Kreise  $\vec{C}_1, \dots, \vec{C}_l$ , sodass der Graph  $\vec{G}_C$ , der aus  $\vec{G}$  entsteht, indem  $\vec{C}_1, \dots, \vec{C}_l$  umgedreht werden, folgenden Eigenschaften hat:

1.  $\vec{G}_C$  enthält keine Rechtskreise (Kreise, deren Inneres rechts liegt).
2. Sei  $\vec{P}_c \subseteq \vec{P}_E$  die Menge kantendisjunkter gerichteter  $s$ - $t$ -Wege in  $\vec{G}_C$ . Definiere  $\vec{P} \subseteq \vec{E}$ :

$$\vec{P} := \left( \vec{P}_C \cap \vec{E} \right) \cup \left\{ (u, v) \in \vec{E} \mid (u, v) \text{ auf einem der } \vec{C}_i \text{ und } (v, u) \notin \vec{P}_C \right\}$$

$\vec{P}$  ist genau dann eine maximale Menge kantendisjunkter gerichteter  $s$ - $t$ -Wege, falls  $\vec{P}_C$  eine maximale Menge kantendisjunkter gerichteter  $s$ - $t$ -Wege in  $\vec{G}_C$  ist.

*Bemerkung.* Zu  $(u, v) \in \vec{E}$ ,  $(u, v)$  auf einem der  $\vec{C}_i$  wird die entsprechende Kante in  $\vec{E}_C$  als  $(v, u)'$  bezeichnet, um sie von der „eigentlichen“ Kante  $(v, u) \in \vec{E}$  zu unterscheiden.

---

**Definition 19.** Sei  $G$  ein planarer Graph und  $\mathcal{F}$  die Facettenmenge einer Einbettung von  $G$ . Hierzu wird der ABSTAND von  $f \in \mathcal{F}$   $\text{dist}(f)$  zur äußeren Facette  $f_0$  als Länge eines kürzesten Weges vom Dualknoten von  $f$  zum Dualknoten von  $f_0$  in  $G^*$  definiert.

---

**Konstruktion von  $C_1, \dots, C_l$  und  $\vec{C}_1, \dots, \vec{C}_l$**

Sei  $\vec{\mathcal{F}}$  die Facettenmenge von  $\vec{G}$  und  $l := \max_{f \in \vec{\mathcal{F}}} \text{dist}(f)$ . Definiere für  $1 \leq i \leq l$   $C_i$  als Vereinigung aller einfachen Kreise  $c$  in  $G$ , für die gilt:

$$\begin{aligned} \forall f \in \text{Inneres}(c) : \text{dist}(f) &\geq i \\ \forall f \in \text{Äußeres}(c) : \text{dist}(f) &< i \end{aligned}$$

Dann gilt:

Abbildung 6.3  
Konstruktion der  $\vec{C}_1, \dots, \vec{C}_l$

1.  $\vec{G}_C$  enthält keine Rechtskreise, da von jedem Rechtskreis in  $\vec{G}$  mindestens eine Kante umgedreht wurde.
2.  $k$  kantendisjunkte, gerichtete  $s$ - $t$ -Wege  $\vec{P}_C$  in  $\vec{G}_C$  induzieren  $k$  kantendisjunkte gerichtete  $s$ - $t$ -Wege  $\vec{P}$  in  $\vec{G}$ .

**Lemma.** Sei  $\vec{H} = (V_{\vec{H}}, E_{\vec{H}})$  ein gerichteter und zusammenhängender Graph,  $s, t \in V_{\vec{H}}$ .  $\vec{H}$  besteht aus  $k$  gerichteten kantendisjunkten  $s$ - $t$ -Wegen genau dann, wenn

1.  $\forall v \in V_{\vec{H}} \setminus \{s, t\} : d^{\leftarrow}(v) = d^{\rightarrow}(v)$
2.  $k = d^{\rightarrow}(s) - d^{\leftarrow}(s) = d^{\leftarrow}(t) - d^{\rightarrow}(t)$

*Beweis.* „klar“ ■

Betrachte  $\vec{P}_C$  und  $\vec{P}$ , sowie  $v$  auf einem  $\vec{C}_i$  mit  $(v, u), (w, v)$  auf  $\vec{C}_i$ . Offensichtlich ändern sich  $d^{\rightarrow}(s)$  und  $d^{\leftarrow}(v)$  nur für sich beim Übergang von  $\vec{P}_C$  nach  $\vec{P}$  und zwar um den gleichen Betrag, denn:

$$\begin{aligned} (u, v)', (v, w)' \in \vec{P}_C &\iff (v, u), (w, v) \notin \vec{P} \\ (u, v)' \in \vec{P}_C, (v, w)' \notin \vec{P}_C &\iff (v, u) \notin \vec{P}, (w, v) \in \vec{P} \\ (u, v)' \notin \vec{P}_C, (v, w)' \in \vec{P}_C &\iff (v, u) \in \vec{P}, (w, v) \notin \vec{P} \\ (u, v)', (v, w)' \notin \vec{P}_C &\iff (v, u), (w, v) \in P \end{aligned}$$

**Zu Schritt 3** Die maximale Anzahl kantendisjunkter gerichteter  $s - t$ -Wege in  $\vec{G}_C$  wird mit einer Right-First-Tiefensuche ermittelt.

```

 $\vec{P}_C \leftarrow \emptyset$ 
forall the  $i \in \{1, \dots, r\}$  do
     $l \leftarrow l_i$ 
     $p_i \leftarrow \{l_i\}$ 
    while weder  $s$  noch  $t$  erreicht do
        setze  $v$  Einlauf Knoten von  $l$ 
        Wähle  $l'$  als rechteste freie aus  $v$  auslaufende Kante bezüglich einer „Referenz-
        kante“
         $p_i \leftarrow p_i \cup \{l'\}$ 
         $l \leftarrow l'$ 
        if  $p_i$  ist  $s - t$ -Weg then
             $\vec{P}_C \leftarrow \vec{P}_C \cup p_i$ 
        end
    end
end

```

Abbildung 6.4

Die Right-First-Tiefensuche im 3. Schritt von Algorithmus 4.  $l_1, \dots, l_r$  sind die aus  $s$  auslaufenden Kanten.

Dabei gibt es zwei Möglichkeiten für die Referenzkante, also jene Kante, die bestimmt, welche Kante am weitesten rechts ist und als nächstes genommen werden soll:

## 6 Das Menger-Problem

- Die erste Kante über die  $v$  jemals erreicht wurde (Version von Coupry)
- Die Kante über die  $v$  aktuell erreicht wurde (Version von Weihe)

### Korrektheitsbeweis

*Behauptung.* Die Right-First-Prozedur berechnet die maximale Anzahl kantendisjunkter gerichteter  $s$ - $t$ -Wege in  $\vec{G}_C$ .

*Beweis.* Es wird die Right-First-Prozedur in der Version von Weihe betrachtet. Da  $\forall v \in \vec{V}_C : d^{\rightarrow}(v) = d^{\leftarrow}(v)$  gilt, endet jede der Right-First-Suchen entweder bei  $t$  oder nach einem Linkskreis bei  $s$ . Gilt  $k = d^{\rightarrow}(s)$  ist die Behauptung offensichtlich.

Sei also  $k < d^{\rightarrow}(s)$ . Zum Beweis wird der Satz von Menger verwendet. Für  $k$  gefundene  $s$ - $t$ -Wege wird gezeigt, dass es in  $\vec{G}_C$  einen  $s$ - $t$ -Schnitt gibt, der genau  $k$  „von  $s$  nach  $t$  gerichtete Kanten“ enthält. Seien  $p_1, \dots, p_r$  die durch die Right-First-Prozedur berechneten Linkskreise und  $s$ - $t$ -Wege. Es wird ein Kreis  $\vec{K}$  in  $\vec{G}_C$  konstruiert, sodass

1.  $s \in \text{Inneres}(\vec{K})$  oder  $s \in \vec{K}$
2.  $t \in \text{Äußeres}(\vec{K})$
3. für  $A := \{(u, v) \in \vec{E}_C \mid u \text{ auf } \vec{K} \text{ und } v \in \text{Äußeres}(\vec{K})\}$  gilt  $|A| = k$

Hierzu sei  $(v, s)$  eine beliebige von  $p_1, \dots, p_r$  besetzte, links einlaufende Kante. Nach Voraussetzung existiert diese. Führe von  $(v, s)$  aus eine rückwärts gerichtete Links-First-Prozedur im durch  $p_1, \dots, p_r$  induzierten Subgraph von  $\vec{G}_C$  aus. Wähle dazu jeweils nach  $(v_i, v_{i-1})$  als nächstes die im Uhrzeigersinn nächste einlaufende Kante in der Adjazenzliste von  $v_i$  als  $(v_{i+1}, v_i)$ .  $\vec{K}$  ist dann die Folge von Kanten  $(v_r, v_{r-1}), \dots, (v_1, v_0)$ . Entweder ist  $v_r = s = v_0$  oder  $v_r = v_0$  und  $s \in \text{Inneres}(\vec{K})$ .

**Lemma.** Jede Kante  $(u, v) \in \vec{E}_C$  mit  $u$  auf  $\vec{K}$  und  $v \in \text{Äußeres}(\vec{K})$  führt zu einem der  $s$ - $t$ -Wege aus  $\vec{P}_C$ .

*Beweis.*  $p_1, \dots, p_r$  seien die  $s$ - $t$ -Wege und Linkskreise. Aufgrund der Konstruktion von  $\vec{K}$  kann es keine Kante geben, die auf einem der  $p_1, \dots, p_r$  liegt und von Außen auf  $\vec{K}$  zeigt. Insofern kann es aber auch keine Kante  $(x, y)$  geben, die auf einem Linkskreis in  $p_1, \dots, p_r$  liegt und aus  $\vec{K}$  heraus führt.

Betrachte  $(u, v)$  mit  $u$  auf  $\vec{K}$  und  $v \in \text{Äußeres}(\vec{K})$  und  $(u, w) \in \vec{K}$ . Die Referenzkante zu  $(u, w)$  liegt entweder auf  $\vec{K}$  oder zeigt aus  $\text{Inneres}(\vec{K})$  nach  $u$ . Bei der Right-First-Prozedur müsste dann  $(u, v)$  vor  $(u, w)$  von der Referenzkante aus gewählt worden sein, also liegt  $(u, v)$  auf einem der  $p_i$ . □

*Schlussfolgerung.*  $|A| = k := \text{Anzahl der } s\text{-}t\text{-Wege in } \vec{P}_C$ . ■

## 6.2 Knotendisjunktes Menger-Problem

Aus dem Satz von Menger ergibt sich, dass die maximale Anzahl knotendisjunkter  $s - t$ -Wege genau dann  $k$  ist, wenn die Kardinalität eines minimalen Separators, der  $s$  und  $t$  trennt,  $k$  ist.

Sei  $G$  planar. Ohne Beschränkung der Allgemeinheit sei  $G$  so planar eingebettet, dass  $t$  auf der äußeren Facette liegt.

**Algorithmus 5** (Knotendisjunktes Menger-Problem auf planaren Graphen).

1. Transformiere  $G$  zu  $\vec{G}$ , indem jede ungerichtete Kante  $\{u, v\}$ ,  $u, v \notin \{s, t\}$  durch die gerichteten Kanten  $(u, v)$  und  $(v, u)$  ersetzt wird. Alle Kanten  $\{s, x\}$  werden nur durch  $(s, x)$  und alle Kanten  $\{t, x\}$  werden nur durch  $(x, t)$  ersetzt. Sind  $\vec{p}_1, \dots, \vec{p}_l$  knotendisjunkte  $s - t$ -Wege in  $\vec{G}$ , so induzieren diese direkt knotendisjunkte  $s - t$ -Wege  $p_1, \dots, p_l$  in  $G$ .
2. Seien  $E_S := (s, v_1), \dots, (s, v_r) \in \vec{E}$  alle aus  $s$  auslaufenden Kanten. Iteriere über  $E_S$ , wobei jeweils von  $(s, v_i)$  aus jeweils eine *Right-First-Tiefensuche* durchgeführt wird. „Konflikte“, das heißt Auftreffen der Suche auf besetzte Knoten werden geeignet aufgelöst und induzieren den „backtrack-remove“-Schritt. Falls keine auslaufende freie Kante existiert, wird ebenfalls der „backtrack-remove“-Schritt ausgeführt.  
 $\rightsquigarrow$  Die *Right-First-Tiefensuche* endet bei  $t$  oder bei  $s$  (nach backtrack-remove).

### Behandlung von Konflikten

**Konflikt von links:** Trifft der aktuelle Suchweg von links auf einen besetzten Knoten  $v$ , so wird ein „backtrack-remove“-Schritt ausgeführt, es wird also die letzte Kante vom Suchweg und aus  $\vec{G}$  entfernt.

**Konflikt von rechts:** Nun trifft der aktuelle Suchweg von rechts auf einen besetzten Knoten  $v$ . Sei  $p$  der  $s - t$ -Weg, der  $v$  besetzt. Sei  $p_1 \subset p$  der Teilweg von  $s$  nach  $v$ , zu dem die besetzte, in  $v$  einlaufende Kante gehört und  $p_2 \subset p$  der Teilweg von  $v$  nach  $t$ , zu dem die besetzte, aus  $v$  auslaufende Kante gehört. Sei  $q$  der aktuelle Suchweg.  $q$  und  $p_2$  wird nun zu einem  $s - t$ -Weg kombiniert, der  $p$  ersetzt.  $p_1$  wird zum aktuellen Suchweg.  $p_1$  hat nun einen Konflikt von links, der wie oben behandelt wird.

Die Auflösung eines Konflikts von rechts ist nur dann sinnvoll, falls die Kanten, die den Konfliktknoten  $v$  besetzen, nicht zum aktuellen Suchweg  $q$  gehören. Dies könnte auftreten, falls der Algorithmus einen Rechtskreis durchläuft. Der Algorithmus ist daher so angelegt, dass er niemals einen Rechtskreis durchläuft. Hierzu werden zwei „Tricks“ verwendet.

**Erster Trick:** Man kann zeigen, dass für eine Kante  $(v, w)$ , nach der der Suchweg einen Rechtskreis mit anschließendem Konflikt von rechts in  $v$  durchläuft, bereits zu einem früheren Zeitpunkt die Kante  $(w, v)$  in einem Linkskreis durchlaufen wurde. Daher wird vom Algorithmus eine Kante  $(v, w)$  entfernt, falls sie als nächste Kante des

## 6 Das Menger-Problem

aktuellen Suchwegs gewählt würde,  $(w, v)$  aber vom aktuellen Suchweg bereits belegt wurde.

Damit dieser Trick ausgeführt werden kann, muss für jede Kante bekannt sein, ob sie auf dem aktuellen Suchweg liegt. Allerdings kann sich die Zugehörigkeit von Kanten zu Wegen im Zuge der Auflösung eines Konflikts von rechts ändern. Alle betroffenen Kanten bei solch einem Konflikt zu aktualisieren würde allerdings zu einer quadratischen Laufzeit des Algorithmus führen.

**Zweiter Trick:** Es wird ein globaler Zähler eingeführt. Dieser wird immer dann erhöht, wenn ein neuer Suchweg begonnen wird oder die Suchwege zur Auflösung eines Konflikts von rechts umorganisiert werden. Wird ein Knoten vom aktuellen Suchweg belegt, so erhält er den aktuellen Stand des Zählers als „Zeitstempel“. Eine Kante  $(w, v)$  ist somit genau dann bereits zuvor vom aktuellen Suchweg gewählt worden, falls der Zeitstempel von  $v$  identisch zu jenem von  $w$  ist.

## 7 Das Problem von Okamura und Seymour

**Problem 10** (Kantendisjunktes Wegpackungsproblem). Gegeben ein Graph  $G = (V, E)$  und eine Menge an  $k$  ausgezeichneten Knotenpaaren  $D := \{\{s_i, t_i\} \subseteq V \mid 1 \leq i \leq k\}$  (wobei  $s_i$  und  $t_i$  nicht notwendigerweise verschieden sind). Finde paarweise kantendisjunkte  $s_i$ - $t_i$ -Wege für  $1 \leq i \leq k$ .  $s_i, t_i$  heißen TERMINALE, Paare  $\{s_i, t_i\}$  heißen auch NETZE. Das Problem ist, auch wenn  $G$  planar ist,  $\mathcal{NP}$ -schwer. Folgende Einschränkungen sind sinnvoll:

- $s_i, t_i$  müssen für  $1 \leq i \leq k$  auf dem Rand derselben Facette (o.B.d.A. die äußere Facette) liegen.
- Sei  $D := \{\{s_1, t_1\}, \dots, \{s_k, t_k\}\}$  ( $\hat{=}$  Menge der „Demand-Kanten“).  $G + D := (V, E \cup D)$  muss planar sein.

---

**Definition 20.** Sei  $G = (V, E)$ ,  $X \subseteq V$ . Dann heißt

$$\text{cap}(X) := |\{\{u, v\} \in E \mid u \in X, v \in V \setminus X\}|$$

KAPAZITÄT von  $X$ .

Sei  $D := \{\{s_i, t_i\} \subseteq V \mid 1 \leq i \leq k\}$  gegeben. Dann heißt

$$\text{dens}(X) := |\{\{s_i, t_i\} \in D \mid |\{s_i, t_i\} \cap X| = 1\}|$$

DICHTE von  $X$  und

$$\text{fcap}(X) := \text{cap}(X) - \text{dens}(X)$$

FREIE KAPAZITÄT von  $X$ .

---

Eine notwendige Bedingung für die Lösbarkeit des kantendisjunktes Wegpackungsproblems ist

$$\forall X \subset V : \text{fcap}(X) \geq 0$$

Im Falle  $\text{fcap}(X) = 0$  heißt  $X$  SATURIERT, im Falle  $\text{fcap}(X) < 0$  ÜBERSATURIERT. Diese Bedingung ist allerdings nicht hinreichend.



## 7 Das Problem von Okamura und Seymour

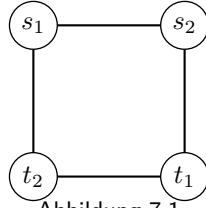


Abbildung 7.1

Es gibt Graphen mit  $\forall X \subset V : \text{fcap}(X) \geq 0$ , für die das kantendisjunkte Wegpackungsproblem nicht lösbar ist.

---

**Definition 21.** Ein Graph  $G = (V, E)$  mit einer Menge an ausgezeichneten Knotenpaaren  $D := \{\{s_i, t_i\} \subseteq V \mid 1 \leq i \leq k\}$  erfüllt die GERADHEITSBEDINGUNG, falls

$$\forall X \subset V : \text{fcap}(X) \text{ ist gerade}$$


---

**Problem 11** (Okamura-Seymour Problem). Gegeben ein Graph  $G = (V, E)$  und eine Menge an  $k$  ausgezeichneten Knotenpaaren  $D := \{\{s_i, t_i\} \subseteq V \mid 1 \leq i \leq k\}$  (wobei  $s_i$  und  $t_i$  nicht notwendigerweise verschieden sind).  $G$  lasse sich so planar einbetten, dass  $s_i, t_i \forall 1 \leq i \leq k$  auf der äußeren Facette liegen. Erfülle  $G$  die Geradheitsbedingung. Finde paarweise kantendisjunkte  $s_i - t_i$ -Wege für  $1 \leq i \leq k$ .

**Satz 11** (Satz von Okamura und Seymour). *Eine Instanz des Okamura-Seymour-Problems ist genau dann lösbar, wenn die Kapazitätsbedingung erfüllt ist.*

Der Beweis wird durch den Korrektheitsbeweis von Algorithmus 6 geführt.

**Lemma.** *Gegeben ein planarer Graph  $G = (V, E)$  und  $D := \{\{s_i, t_i\} \subseteq V \mid 1 \leq i \leq k\}$  gilt:*

$$\forall X \subseteq V : \text{fcap}(X) \text{ gerade} \iff \forall v \in V : \text{fcap}(\{v\}) \text{ gerade}$$

*Beweis.*

„ $\implies$ “ klar.

„ $\impliedby$ “ Sei  $\forall v \in V \text{ fcap}(\{v\})$  gerade. Sei  $X \subseteq V$ . Dann:

$$\begin{aligned} \text{cap}(X) &= \sum_{v \in X} \text{cap}(\{v\}) - 2|\{\{u, v\} \in E \mid u, v \in X\}| \\ \text{dens}(X) &= \sum_{v \in X} \text{dens}(\{x\}) - 2|\{\{s_i, t_i\} \in D \mid s_i, t_i \in X\}| \end{aligned}$$

$$\begin{aligned}
 \implies \text{fcap}(X) &= \sum_{v \in X} \text{cap}(\{v\}) - 2|\{\{u, v\} \in E \mid u, v \in X\}| \\
 &\quad - \sum_{v \in X} \text{dens}(\{x\}) - 2|\{\{s_i, t_i\} \in D \mid s_i, t_i \in X\}| \\
 &= \sum_{v \in X} \text{fcap}(\{v\}) - 2(|\{\{u, v\} \in E \mid u, v \in X\}| + |\{\{s_i, t_i\} \in D \mid s_i, t_i \in X\}|)
 \end{aligned}$$

Da nach Voraussetzung alle  $\text{fcap}(\{v\})$  gerade sind, folgt die Behauptung.  $\blacksquare$

**Algorithmus 6** (Linearzeitalgorithmus für das Okamura-Seymour Problem).

1. Konstruiere aus  $G = (V, E)$  und  $D = \{\{s_1, t_1\}, \dots, \{s_k, t_k\}\}$  „einfachere“ Instanz mit  $D' = \{\{s'_1, t'_1\}, \dots, \{s'_k, t'_k\}\}$ .
2. Berechne in  $\mathcal{O}(n)$  paarweise kantendisjunkte (orientierte)  $s'_i$ - $t'_i$ -Wege, also eine Lösung zu  $D'$  in  $G$ . Dies induziert einen gerichteten Hilfsgraphen  $\vec{G} = (\vec{V}, \vec{E})$ .
3. Berechne in  $\mathcal{O}(n)$  in  $\vec{G}$  eine Lösung zu  $D$  in  $G$ .

### Zu Schritt 1

$D'$  entsteht aus  $D$  indem die  $s_i, t_i$  neu gepaart werden, sodass  $\{s_1, \dots, s_k, t_1, \dots, t_k\} = \{s'_1, \dots, s'_k, t'_1, \dots, t'_k\}$ . Die neuen Paarungen  $\{s'_i, t'_i\}$  haben dabei eine „Klammerstruktur“:

### Zu Schritt 2

Man kann beobachten, dass sich weder zwei  $s'_i - t'_i$ -Wege kreuzen noch sich ein  $s'_i - t'_i$ -Weg mit sich selbst kreuzt.

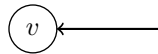


Abbildung 7.2  
Widerspruch zur RF-Suche.

**Lemma.**  $\vec{G}$  enthält keinen Rechtskreis.

*Beweis.* Angenommen,  $\vec{G}$  enthält einen Rechtskreis  $\vec{C}$ . Seine Struktur stünde jedoch im Widerspruch zur Klammerung. ■

### Korrektheitsbeweis des zweiten Schritts

*Behauptung.* Für eine wie in Schritt 1 konstruierte, lösbare Instanz des Okamura-Seymour-Problems ( $G = (V, E)$ ,  $D' = \{\{s'_i, t'_i\} \subseteq V \mid 1 \leq i \leq k\}$ ) findet Schritt 2 Wege  $q_1, \dots, q_k$ , die jeweils  $s'_i$  mit  $t'_i$  verbinden.

*Beweis.* Sei  $q'_1, \dots, q'_k$  eine beliebige kreuzungsfreie Lösung der Instanz  $(G, D')$ . Induktiv über  $i$  gilt für jedes von der Right-First-Prozedur gefundene  $q_i$ , dass die linke Seite von  $q'_i$  ganz in der linken Seite von  $q_i$  enthalten ist. Dies liegt daran, dass per Right-First-Regel ausgewählt wurde und die Klammer-Paarungen der  $s'_i, t'_i$  von innen nach außen abgearbeitet wurden. Deshalb muss  $q_i$  gerade  $s'_i$  mit  $t'_i$  verbinden. ■

Die Linearzeit von Schritt 2 ist offensichtlich.

### Ausführung von Schritt 3

Führe eine Right-First-Suche auf  $\vec{G}$  aus. Dabei werden die  $\{s_i, t_i\}$  so indiziert, dass  $s_i$  vor  $t_i$  und  $t_i$  vor  $t_{i+1}$  als Startterminal.

**Korrektheitsbeweis des dritten Schritts** Da in  $\vec{G}$  alle Knoten gleichen Ein- wie Ausgangsgrad haben, endet eine Right-First-Tiefensuche von einem  $s_i$  immer bei einem  $t_j$ .

*Behauptung.* Eine Right-First-Suche wie in Schritt 3, die bei  $s_i$  beginnt, endet nur dann in  $t_j$   $i \neq j$ , wenn es einen Schnitt  $X$  in  $G$  gibt mit  $\text{fcap}(X) < 0$ .

*Beweis.* Prozedur, die auf Basis von  $p_1, \dots, p_i$ ,  $i \leq k$ , die nach der  $i$ -ten Right-First-Suche in Schritt 3 gefunden wurden, einen Schnitt  $X$  konstruiert, für den gilt:

1. falls  $p_1, \dots, p_i$  jeweils  $s_j - t_j$ -Wege,  $\forall 1 \leq j \leq i$ , so ist  $\text{fcap}(X) = 0$
2. falls  $p_i$   $s_i$  mit  $t_j$ ,  $i \neq j$ , verbindet, so ist  $\text{fcap}(X) < 0$ .

**Algorithmus** (Schnitt-Prozedur). *Rückwärts-Left-First-Suche beginnt bei einem  $t$ -Terminal in dem durch die bisherige Suche in Schritt 3 induzierten Graph. Wurden bisher also  $i$  Suchen durchgeführt, wird in dem durch  $p_1, \dots, p_i$  induzierten Graphen gesucht. Die Suche induziert einen Weg  $P$  vom  $t$ -Terminal, bei dem begonnen wurde, zu einem  $s$ -Terminal.  $P$  und alles links von  $P$  induziert den Schnitt  $X \subset V$ .*

Betrachte den Schnitt  $X$  und die Kanten, die über  $X$  gehen.

$$\begin{aligned} \text{cap}(X) &= |\{\{s_j, t_j\} \mid s_j \in X, t_j \in V \setminus X, 1 \leq j \leq i\}| + |\{\{s'_j, t'_j\} \mid s'_j \in V \setminus X, t'_j \in X, 1 \leq j \leq k, \{s'_j, t'_j\} \not\subseteq \{s_1, \dots, s_i\}\}| \\ &= \text{dens}(X) \end{aligned}$$

■

## 7 *Das Problem von Okamura und Seymour*

**Laufzeit**  $\mathcal{O}(n)$  mit Union-Find wie im kantendisjunkten Menger-Problem in der Version von Weihe.

# Index

- $G - x$ , 6
- $K_{n,m}$ , 1
- $K_n$ , 1
- $\text{cap}(X)$ , 44
- $\text{dens}(X)$ , 44
- $\text{dist}(f)$ , 39
- $\text{fcap}(X)$ , 44
- $k$ -Listenfärbbarkeit, 17
- übersättigt, 44
  
- Abstand, 39
  
- BFS-Baum, 24
  - Level, 24
- Blockzerlegung, 10
  
- Dichte, 44
- Dualgraph, 5
  
- Einbettung
  - geometrische, 4
  - kombinatorische, 4
  - planare, 4
- eulersch, 32
  
- $f$ , 6
- Färbungsproblem, 16
- Facette, 4
  
- gematcht, 29
- Geradenbedingung, 45
  
- Jordan-Kurve, 1
  
- kantendisjunkt, 36
- Kantendisjunktes Wegpackungsproblem, 44
  
- Kapazität, 44
  - freie, 44
- kontendisjunkt, 36
  
- Listenfärbbarkeit, 17
- Listenfärbungsproblem, 17
  
- $m$ , 6
- Matching, 29
  - perfektes, 33
- Max Cut, 32
- Menge
  - gerade, 32
- Menger-Problem, 37
- Min Bounded Separator Problem, 24
- Min Cut, 32
- Minor, 9
- minor-minimal, nicht-planar, 10
- Mixed Max Cut, 32
- Mixed Max Kreis, 32
  
- $n$ , 6
- Nachbarschaft, 10
  
- Okamura-Seymour Problem, 45
  
- planar, 1
  
- sättigt, 44
- Schnitt, 6
  - Gewicht, 32
- Separator, 24
- Separatorknoten, 10
- Subgraph, 9
  
- $\Theta$ -Graph, 9

## *Index*

Unterteilung, 9

Weg

alternierender, 30

erhöhender, 30

Zusammenhang

-Kanten, 36

-Knoten, 36