

# **Zusammenfassung - Algorithmen für planare Graphen**

Julian Shen

24. August 2023

# 1 Einführung

**Definition: Graph** ist ein Tupel  $G = (V, E)$  mit endliche Knotenmenge  $V$  und endliche Kantenmenge  $E$

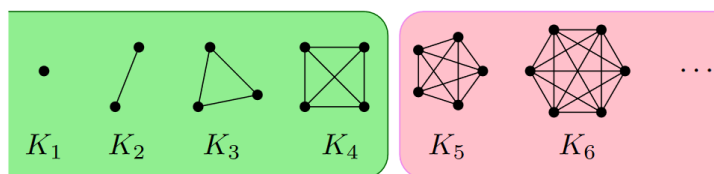
- Kante  $e \in E$  hat Form  $e = uv$  mit  $u, v \in V$ .
- $uv = vu \rightarrow$  Graphen **ungerichtet**
- $e = uu$  ist erlaubt  $\rightarrow$  **Schlinge**
- Auch  $e = uv$  und  $e' = uv$  erlaubt mit  $e \neq e' \rightarrow$  **Mehrfachkante**
- **Einfacher Graph**  $\iff$  ohne Schlingen und Mehrfachkanten
- **Zusammenhängend**  $\iff$  ein Weg zwischen je zwei Knoten

**Definition:** Eine **Zeichnung** von  $G = (V, E)$  bildet diesen so auf  $\mathbb{R}^2$  ab, dass

1. Knoten Punkte in der Ebene sind, d.h.  $V \subset \mathbb{R}^2$
  2. Kante  $e = uv$  ist injektive, stetige Kurve von  $u$  nach  $v$ , d.h.  $\gamma_e: [0, 1] \rightarrow \mathbb{R}^2$  mit
    - $\gamma_e(0) = u$  und  $\gamma_e(1) = v$
    - $\gamma_e(t) \notin V$  für alle  $0 < t < 1$
- Zeichnung heißt **kreuzungsfrei** bzw. **planar** wenn für je zwei Kanten  $e, e'$  und  $0 < t, t' < 1$  gilt:  $\gamma_e(t) \neq \gamma_{e'}(t')$
  - Graph heißt **planar**, wenn er mindestens eine kreuzungsfreie Zeichnung besitzt

**Definition:** Für  $n \in \mathbb{N}$  ist der **vollständige Graph**  $K_n$

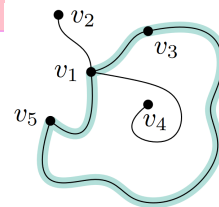
- $V(K_n) = \{v_1, \dots, v_n\}$
- $E(K_n) = \{v_i v_j \mid i, j \in \{1, \dots, n\}, i \neq j\}$



**Lemma:** Graph  $K_5$  ist nicht planar

*Beweis:* Betrachte beliebige Zeichnung von  $K_5$

- Betrachte  $v_1$  und seine 4 ausgehenden Kanten
- O.B.d.A. Kanten kreuzungsfrei zu  $v_2, v_3, v_4, v_5$  in zyklischer Reihenfolge um  $v_1$
- Kanten  $v_1 v_3, v_3 v_5, v_5 v_1$  bilden geschlossene Kurve in  $\mathbb{R}^2$  die  $v_2$  und  $v_4$  trennt  $\implies v_2 v_4$  kann nicht kreuzungsfrei gezeichnet sein



**Definition:** Für  $m, n \in \mathbb{N}$  ist der **vollständig bipartite Graph**  $K_{m,n}$

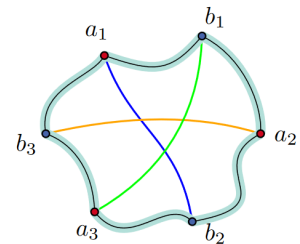
- $V(K_{m,n}) = \{a_1, \dots, a_m\} \cup \{b_1, \dots, b_n\}$
- $E(K_{m,n}) = \{a_i b_j \mid i \in \{1, \dots, m\}, j \in \{1, \dots, n\}\}$



**Lemma:** Graph  $K_{3,3}$  ist nicht planar

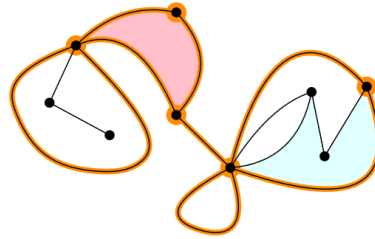
*Beweis:* Betrachte beliebige Zeichnung von  $K_{3,3}$

- Kreis  $a_1 b_1 a_2 b_2 a_3 b_3$  im Graphen bildet eine geschlossene Kurve in  $\mathbb{R}^2$
  - Jede Kante von  $a_1 b_2, a_2 b_3, a_3 b_1$  liegt komplett innerhalb oder komplett außerhalb dieser Kurve
- $\implies$  mindestens zwei liegen auf der gleichen Seite
- $\implies$  diese zwei kreuzen sich



**Definitionen:** Für eine feste planare Zeichnung eines planaren Graphen definiere:

- **Facetten:** Zusammenhangskomponenten von  $\mathbb{R}^2$  nach Entfernen aller Knoten und Kanten  $\implies$  Es gibt genau eine **äußere Facette** und mehrere **innere Facetten**
- **Äußere Knoten** sind die, die inzident zur äußeren Facette sind
- **Innere Knoten** sind die übrigen Knoten
- **Äußere Kanten** sind die, die komplett im Rand der äußeren Facette liegen
- **Innere Kanten** sind die übrigen Kanten



$n = 9$  Knoten (5 äußere, 4 innere)  
 $m = 14$  Kanten (8 äußere, 6 innere)  
 $f = 7$  Facetten (1 äußere, 6 innere)

**Satz von Euler:** Sei  $G$  ein zusammenhängender Graph mit einer planaren Zeichnung mit  $n$  Knoten,  $m$  Kanten und  $f$  Facetten. Dann gilt

$$n - m + f = 2$$

*Beweis:* Beweise  $m - (f - 1) = n - 1$ , woraus die Behauptung folgt. Führe dafür eine Induktion nach  $f - 1$ , der Anzahl der inneren Facetten, durch.

- I.A.:  $f - 1 = 0$ , d.h. keine innere Facette  $\rightarrow G$  ist ein Baum, also kreisfrei und zusammenhängend  $\rightarrow m = n - 1$
- I.S.:  $f - 1 \geq 1$ , d.h. min. eine innere Facette
  - Sei  $e$  eine Kante zwischen äußerer und innerer Facette  $\rightarrow G' = G - e$  ist zusammenhängend  $\rightarrow$  In  $G'$  gilt  $n' = n, m' = m - 1, f' = f - 1$
  - Mit I.V. folgt:  $m' - (f' - 1) = n' - 1 \Leftrightarrow m - 1 - (f - 1 - 1) = n - 1 \Leftrightarrow m - (f - 1) = n - 1$



**Korollar aus Euler-Formel:** Sei  $G$  ein planarer, einfacher Graph mit  $n \geq 3$  Knoten,  $m$  Kanten, und kleinstem vorkommenden Knotengrad  $\delta(G)$ . Dann gilt

$$m \leq 3n - 6 \quad \text{und} \quad \delta(G) \leq 5$$

Beide Ungleichungen sind bestmöglich.

*Beweis:*  $m \leq 3n - 6$

- O.B.d.A.  $G$  ist zusammenhängend, da man Kanten einfügen kann bis er das ist
- Jede Facette ist berandet von min. 3 Kantenseiten, da  $n \geq 3$
- Jede Kantenseite in genau 1 Facette
- Jede Kante hat genau 2 Seiten

$$\implies 3f \leq \text{Anzahl der Seiten-Facetten-Inzidenzen} = 2m$$

$$\implies 3(2 + m - n) \leq 2m \implies m \leq 3n - 6 \text{ (mit Euler-Formel)}$$

*Beweis:*  $\delta(G) \leq 5$

- Jede Kante hat genau 2 inzidente Knoten
- Jeder Knoten  $v$  hat genau  $\deg(v)$  inzidente Kanten
- Für jeden Knoten  $v$  gilt  $\deg(v) \geq \delta(G)$

$$\implies 2m = \text{Anzahl der Knoten-Kanten-Inzidenzen} = \sum_{v \in V(G)} \deg(v) \geq \delta(G) \cdot n$$

$$\implies 2(3n - 6) \geq 2m \geq \delta(G) \cdot n \implies \delta(G) \leq 6 - 12/n$$

## 2 Einbetten und Dualisieren

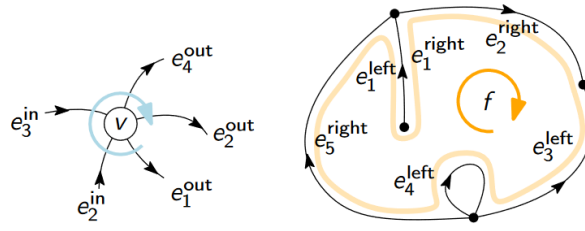
**Einbettung** = Äquivalenzklasse von planaren Zeichnungen



**Definition:** Sei  $G = (V, E)$  ein zusammenhängender Graph mit einer planaren Zeichnung. Die **(kombinatorische) Einbettung** ist

- für jeden Knoten  $v$  die zyklische (cw = „clockwise“) Reihenfolge der inzidenten Halbkanten an  $v$
- für jede Facette  $f$  die zykl. (cw) Reihenfolge der inzidenten Kantenseiten an  $f$

Betrachte dafür beliebige Orientierung der Kanten und man erhält Halbkanten  $e^{\text{in}}$  und  $e^{\text{out}}$  sowie Kantenseiten  $e^{\text{left}}$  und  $e^{\text{right}}$  von  $e$



Alle Zeichnungen mit der gleichen Einbettung sind äquivalent.

**Definition:** Sei  $G = (V, E)$  ein zusammenhängender Graph mit einer festen Einbettung und Facettenmenge  $F$ . Der **Dualgraph**  $G^* = (V^*, E^*)$  ist

- $V^* = F$ , das heißt,  $f \in F \mapsto v_f \in V^*$
- für jede Kante  $e \in E$  läuft die duale Kante  $e^*$  zwischen der Facette an  $e^{\text{left}}$  und der an  $e^{\text{right}}$



Die Einbettung des **Primalgraphen**  $G = (V, E)$  induziert eine Einbettung des **Dualgraphen**  $G^* = (V^*, E^*)$ :

primal	dual
$f \in F$	$v_f = f \in V^* = F$
$e^{\text{left}}, e^{\text{right}}$	$(e^*)^{\text{out}}, (e^*)^{\text{in}}$
$v \in V$	$f_v = v \in F^*$
$e$ Brücke	$e^*$ Schlinge
$e$ Schlinge	$e^*$ Brücke

**Bemerkungen:**

- Der Dualgraph  $G^*$  ist immer zusammenhängend
- Falls  $G$  zusammenhängend ist, gilt  $G = (G^*)^*$
- Für jede Einbettung von  $G$  und jede Facette  $f$  gibt es eine planare Zeichnung mit dieser Einbettung und  $f$  als äußere Facette

### 3 Graphfärbung

**Definition:** Sei  $G = (V, E)$  ein Graph,  $k \in \mathbb{N}$ . Eine  **$k$ -Färbung** von  $G$  ist eine Abbildung  $c: V \rightarrow \{1, 2, \dots, k\}$ , sodass  $c(u) \neq c(v)$  für jede Kante  $uv \in E$

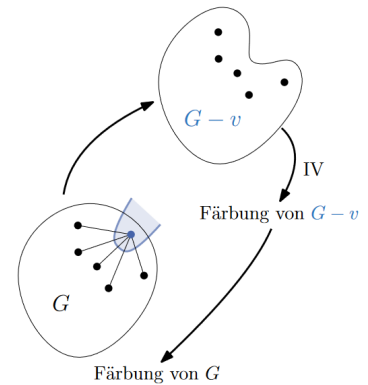
- Kleinstes  $k$ , für das so eine  $k$ -Färbung existiert, heißt **chromatische Zahl**  $\chi(G)$
- Bei Färbungen nehmen wir Graphen implizit als schlingenfrei an

**Frage:** Was ist die größte chromatische Zahl die ein planarer Graph annehmen kann, d.h. was ist  $\chi_{\text{planar}} := \max\{\chi(G) \mid G \text{ planar}\}$ ?

**Lemma:**  $\chi_{\text{planar}} \leq 6$

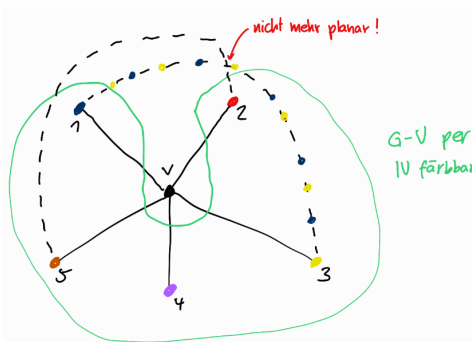
*Beweis:* Führe Induktion über  $|V|$

- I.A.:  $|V| \leq 6$ : Man erhält eine Färbung, indem jeder Knoten eine eigene Farbe bekommt
- I.S.:  $|V| > 6$ 
  - Nach Euler-Formel gibt es  $v \in V$  mit  $\deg(v) \leq 5$
  - Nach I.V. gibt es 6-Färbung von  $G \setminus v$
  - Nachbarn von  $v$  in  $G \setminus v$  decken höchstens fünf Farben ab  $\rightarrow$  Färbe  $v$  in verbleibender Farbe



**Lemma:**  $\chi_{\text{planar}} \leq 5$

*Beweis:* Induktion analog zum oberen Beweis



I.S.:

- Nach Euler-Formel gibt es  $v \in V$  mit  $\deg(v) \leq 5$
- Nach I.V. gibt es 5-Färbung von  $G \setminus v$
- Betrachte Teilgraph, der nur blau-gelbe Knoten enthält:

- Fall 1: 1 und 3 liegen in unterschiedlichen Zusammenhangskomponenten  
 → Tausche in einer Zusammenhangskomponente alle blauen durch gelbe und alle gelben Knoten durch blaue aus → Farbe wird für  $v$  frei
- Fall 2 (siehe Bild): 1 und 3 liegen in der selben Zusammenhangskomponente.  
 Für rot-braun-Teilgraph können 2 und 5 nicht in der selben Zusammenhangskomponente liegen, da Graph sonst nicht mehr planar wäre  
 → Farbe wird für  $v$  frei

**Definition:** Sei  $G = (V, E)$  ein einfacher Graph. Sei  $L: V \rightarrow 2^{\mathbb{N}}$  eine Listenzuweisung, d.h.  $L(v)$  ist Menge von Zahlen / Farben. Eine  **$L$ -Listenfärbung** von  $G$  ist eine Knotenfärbung  $c$  mit

- $c(v) \in L(v)$  für jeden Knoten  $v \in V$
- $c(u) \neq c(v)$  für jede Kante  $uv \in E$

$G$  heißt  **$k$ -listenfärbbar** wenn für jede Listenzuweisung  $L$  mit  $|L(v)| \geq k$  für jeden Knoten  $v \in V$  eine  $L$ -Listenfärbung von  $G$  existiert.

- Kleinstes  $k$ , für das  $G$   $k$ -listenfärbbar ist, heißt **listenchromatische Zahl**  $\chi_{\text{list}}(G)$

**Beweisskizze zu Listenfärbungen:**

- $\chi_{\text{list}}(G) > k: \exists \text{ Listen } L \not\exists L\text{-Listenfärbung}$
- $\chi_{\text{list}}(G) \leq k: \forall \text{ Listen } L \exists L\text{-Listenfärbung}$

**Lemma:** Für jeden planaren Graphen gilt  $\chi_{\text{list}}(G) \leq 6$

*Beweis:* Die gleiche Argumentation wie für  $\chi_{\text{planar}} \leq 6$  funktioniert

**Beobachtung:** Für jeden Graphen  $G$  gilt  $\chi_{\text{list}}(G) \geq \chi(G)$

*Beweis:*

- Setze  $L(v) = \{1, \dots, k\}$  für jeden Knoten  $v$
- Dann sind  $L$ -Listenfärbungen genau  $k$ -Knotenfärbungen  $\implies \chi(G) \leq \chi_{\text{list}}(G)$

**Satz von Voigt:** Es gibt einen planaren Graphen mit  $\chi_{\text{list}}(G) \geq 5$

*Beweis:* Konstruiere einen planaren Graphen  $G$  mit Listenzuweisung  $L$ , sodass

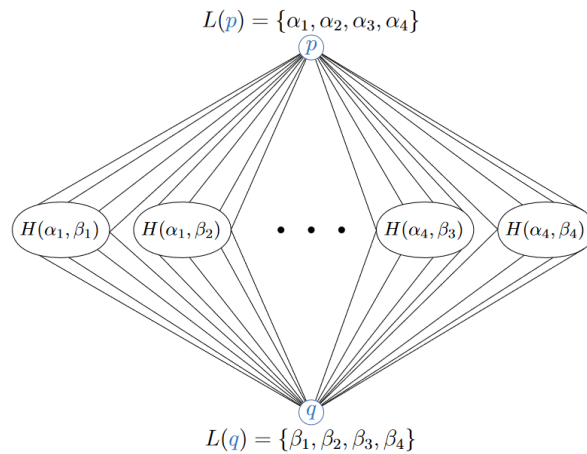
- $|L(v)| = 4$  für jeden Knoten  $v$
- keine  $L$ -Listenfärbung von  $G$  existiert

Betrachte dazu folgendes Gadget  $H(\alpha, \beta)$ :





Dieses Gadget ist nicht färbbar. Konstruiere nun aus 16 Gadgets den folgenden Graphen:



Dieser ist nicht  $L$ -listenfärbbar, denn für jede Färbung  $c$  ist Gadget  $H(c(p), c(q))$  nicht färbbar.

### Weitere Sätze und Beobachtungen:

- Für jeden planaren Graphen gilt  $\chi_{\text{list}}(G) \leq 5$  (**Satz von Thomassen**)  
 $\implies$  Mit obigem Satz folgt  $\max\{\chi_{\text{list}}(G) \mid G \text{ planar}\} = 5$
- Es gibt einen planaren Graphen mit  $\chi(G) \geq 4$
- Für jeden planaren Graphen  $G$  gilt  $\chi(G) \leq 4$  (**4-Farben-Satz**)  
 $\implies \chi_{\text{planar}} = 4$

**Ziel:** Beweise  $\chi_{\text{planar}} \leq 5$  mit einer stärkeren Aussage

**Satz:** Sei  $G = (V, E)$  ein planarer Graph mit:

- jede innere Facette ist ein Dreieck
- äußere Facette ist ein einfacher Kreis  $C$

Seien  $v_1, v_2$  zwei aufeinanderfolgende Knoten auf  $C$  und  $L$  eine Listenzuweisung mit:

- $|L(v)| = 5$  für  $v \in V \setminus C$
- $|L(v)| = 3$  für  $v \in C \setminus \{v_1, v_2\}$
- $L(v_1) = \{\alpha\}, L(v_2) = \{\beta\}$  mit  $\alpha \neq \beta$

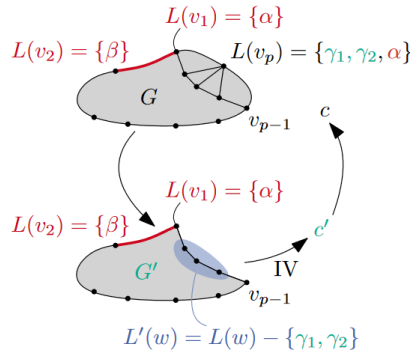
Dann gibt es eine  $L$ -Listenfärbung von  $G$ .

*Beweis:* Führe Induktion über  $|V|$

- I.A.:  $|V| = 3$ . Wähle  $c(v_3) \in L(v_3) \setminus \{\alpha, \beta\}$
- I.S.:  $|V| \geq 4$ . Betrachte nun 2 Fälle:
  - Fall 1:  $C$  hat eine Sehne  $e = v_i v_j$ . Zerteile  $G$  entlang  $e$  in zwei Graphen  $G_1, G_2$ . O.B.d.A liegt  $v_1 v_2$  in  $G_1$ . Nach IV gibt es eine Färbung  $c_1$  von  $G_1$ . Sei  $c_1(v_i) = \alpha'$  und  $c_1(v_j) = \beta'$ . Wende IV auf  $G_2$  an mit Listen  $\{\alpha'\}$  für  $v_i$  und  $\{\beta'\}$  für  $v_j$ .  $\rightarrow$  Färbung  $c_2$  von  $G_2 \rightarrow$  Da  $c_1$  und  $c_2$  an der Sehne  $v_i v_j$  übereinstimmen, erhalten wir eine Färbung von  $G$ .



- Fall 2:  $C$  hat keine Sehne. Betrachte Nachbarn  $v_p \neq v_2$  von  $v_1$  auf  $C$ . Lösche  $v_p$  auf  $G$  und erhalte  $G'$ .  $G'$  hat einfachen Kreis als äußere Facette, da  $v_p$  keine inzidente Sehne hat. Seien  $\gamma_1, \gamma_2$  zwei Farben aus  $L(v_p) \setminus \{\alpha\}$ . Für jeden inneren Nachbarn  $w$  von  $v_p$  definiere  $L'(w) = L(w) \setminus \{\gamma_1, \gamma_2\}$  und  $L'(v) = L(v)$  für jeden anderen Knoten  $v$ . Nach IV gibt es  $L'$ -Listenfärbung von  $G'$ , sodass kein innerer Nachbar von  $v_p$  die Farbe  $\gamma_1$  oder  $\gamma_2$  hat. Wähle  $c(v_p) \in \{\gamma_1, \gamma_2\} \setminus c'(v_{p-1})$  und erhalte somit eine  $L$ -Listenfärbung  $c$  von  $G$ .



**Bemerkung:** Für jeden beliebigen planaren Graphen  $G$  lassen sich Kanten und Knoten hinzufügen sowie Farben aus Listen entfernen, sodass der neue Graph  $G'$  den Anforderungen des obigen Satzes entspricht. Damit wurde die Aussage  $\chi(G) \leq \chi_{\text{list}}(G) \leq 5$  für jeden planaren Graphen  $G$  bewiesen.

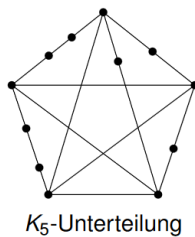
## 4 Unterteilungen und Minoren

**Definition:** Sei  $G = (V, E)$  ein Graph,  $e = uv$  eine Kante. Dann ist die **Unterteilung von  $e$  in  $G$**  der Graph  $G \circ e = (V', E')$  mit

- $V' = V + \{w\}$
- $E' = (E - uv) + \{uw, vw\}$

**Beobachtung:**  $G$  planar  $\iff G \circ e$  planar

**Definition:** Graph  $G$  ist eine **Unterteilung von  $H$**  wenn  $G = ((H \circ e_1) \circ e_2) \cdots \circ e_k$ . Wir sagen auch  $G$  ist  **$H$ -Unterteilung**. Graph  $G$  **enthält eine  $H$ -Unterteilung**, wenn ein Teilgraph  $G' \subseteq G$  eine  $H$ -Unterteilung ist.



**Beobachtung:**

- $K_5$ - und  $K_{3,3}$ -Unterteilungen sind nicht-planar
- Jeder Graph der eine  $K_5$  oder  $K_{3,3}$ -Unterteilung enthält, ist nicht planar

**Satz von Kuratowski:**  $G$  ist planar  $\iff G$  enthält keine  $K_5$ - oder  $K_{3,3}$ -Unterteilung

*Beweis:* „ $\Rightarrow$ “ folgt aus obiger Beobachtung. Die Rückrichtung ist komplizierter und beweisen wir später.

**Definition:** Sei  $G = (V, E)$  ein Graph,  $e = uv$  eine Kante. Der Graph  $G/e = (V', E')$  ist der Graph, der durch **Kontrahieren der Kante  $e$**  entsteht, genauer:

- $V' = V - \{u, v\} + \{w\}$
- $E' = E(G - u - v) \cup \{wa \mid au \in E \text{ oder } av \in E\}$

Diesen Prozess nennt man auch **Kantenkontraktion**. Dabei können Multikanten und Schleifen entstehen.



**Definition:** Graph  $H$  ist **Minor von  $G$** , wenn  $H$  aus  $G$  durch eine Folge von Kantenkontraktionen entsteht, also  $H = ((G/e_1)/e_2 \cdots)/e_k$ . Wir sagen dann auch:  $G$  ist ein  **$H$ -Minor** ( $H$  ist der kleinere Graph,  $G$  der Größere).

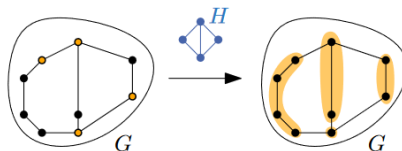
**Beobachtung:**

- $G$  planar  $\Rightarrow G/e$  planar
- $G$  enthält  $K_5$ - oder  $K_{3,3}$ -Minor  $\Rightarrow G$  nicht planar

**Satz von Wagner:**  $G$  planar  $\iff G$  enthält keinen  $K_5$ - oder  $K_{3,3}$ -Minor

**Lemma:**  $G$  enthält  $H$ -Unterteilung  $\Rightarrow G$  enthält  $H$ -Minor

*Beweis:* Kontrahiere durch Unterteilung entstandene Knoten zu ursprünglich adjazenten Knoten.



**Es sind also äquivalent:**

1.  $G$  ist nicht planar
2.  $G$  enthält  $K_5$ - oder  $K_{3,3}$ -Minor
3.  $G$  enthält  $K_5$ - oder  $K_{3,3}$ -Unterteilung

(3)  $\implies$  (2)  $\implies$  (1) wurde schon bewiesen, (1)  $\implies$  (2)  $\implies$  (3) müssen wir noch beweisen. Wir beginnen mit (1)  $\implies$  (2).

*Beweis von Wagner:* Es muss nur noch die Rückrichtung beweisen werden. Sei hierfür  $G$  ein nicht-planarer Graph. Wir müssen einen  $K_5$ - oder  $K_{3,3}$ -Minor in  $G$  finden. O.B.d.A. sei  $G$  sogar **minimal nicht-planar**, d.h.

- $G - v$  ist planar für jeden Knoten  $v \in V$
- $G - e$  ist planar für jede Kante  $e \in E$
- $G/e$  ist planar für jede Kante  $e \in E$

Beweise zunächst folgendes Lemma:

**Lemma:** Sei  $G$  minimal nicht-planar,  $xy \in E(G)$ . Dann ist  $G - x - y$  ein Kreis.

*Beweis:* Da  $G$  minimal nicht-planar ist,

- ist  $G$  zusammenhängend, da ansonsten Knoten aus einer Zusammenhangskomponente gelöscht werden könnte
- ist  $\deg(v) \geq 3$  für jeden Knoten  $v \in V(G)$ , denn Knoten von Grad 0 und 1 tragen nichts zur Nicht-Planarität bei, können also gelöscht werden ohne die Nicht-Planarität zu verlieren. Für einen Knoten  $v$  von Grad 2 mit Kanten  $e, e'$  bleibt  $G/e$  nicht-planar. Wäre  $G/e$  planar, so muss wegen  $G = (G/e) \circ e'$  bereits  $G$  planar sein. Widerspruch.



Das Lemma wird nun anhand von 3 Behauptungen bewiesen.

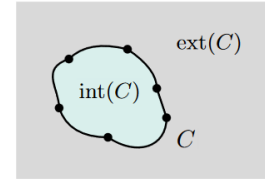
1. Behauptung:  $G - x - y$  enthält kein  $\Theta$ .

**Theta-Graphen** sind Unterteilungen des Graphen mit zwei Knoten und drei parallelen Kanten.



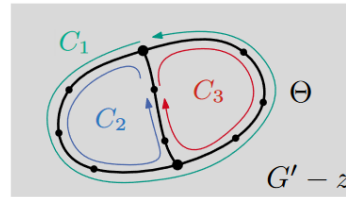
**Notation:** Für einen Kreis  $C$  in einer planaren Zeichnung erhält man eine **geschlossene Jordankurve**, die die Ebene in zwei Komponenten unterteilt:

- $\text{int}(C)$ , das Innere von  $C$
- $\text{ext}(C)$ , das Äußere von  $C$

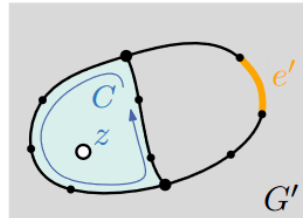


*Beweis von Behauptung 1:*

- Angenommen  $G - x - y$  enthält ein  $\Theta$ .
- $G' := G/xy$  ist planar mit Kante  $xy$  zu Knoten  $z$  kontrahiert.
- $G' - z = G - x - y$  ist ebenfalls planar.
- Zeichnung von  $G'$  enthält ein  $\Theta$  und das  $\Theta$  hat 3 Kreise:

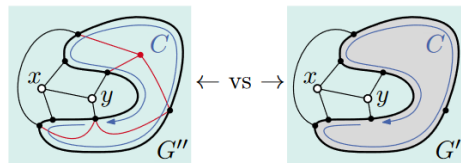


- Betrachte Kreis  $C$  im  $\Theta$ , sodass Knoten  $z$  auf einer Seite von  $C$  und eine Kante  $e' \in E(\Theta)$  auf der anderen Seite von  $C$  liegt.
- Wähle  $\Theta$  und  $C$  so, dass die Seite von  $C$  mit  $z$  inklusionsminimal ist, d.h. es gibt kein anderes  $\Theta$  mit Kreis  $C$ , was  $z$  enthält und ein kleineres Inneres hat.
- O.B.d.A. gilt  $z \in \text{int}(C)$  und  $e' \in \text{ext}(C)$ .



- Betrachte  $G'' = G - \text{ext}(C)$ .
- Da  $e' \notin G''$  ist, wird mindestens eine Kante gelöscht, also ist  $G''$  planar.
- Betrachte eine planare Zeichnung von  $G''$  mit Kreis  $C$ .

Ziel: Zeige, dass  $C$  eine Facette berandet, denn dann kann  $\text{ext}(C)$  in  $C$  eingesetzt werden, was aber eine planare Zeichnung von  $G$  wäre. ⚡



- Betrachte Pfad  $P$  in  $G''$ , der auf verschiedenen Knoten von  $C$  startet und endet und ansonsten zu  $C$  disjunkt ist.
- $P$  entspricht auch einem Pfad  $P'$  in  $G'$ .

Wenn  $z \notin P'$ :

- Dann ist  $C \cup P'$  ein  $\Theta$  in  $G - x - y$ .
- Dieses  $\Theta$  hat einen Kreis der  $z$  enthält, aber ein kleineres Inneres als  $C$  hat.
- Widerspruch zur Wahl von  $\Theta$  und  $C$ .



Also liegt  $z$  auf  $P'$  und  $P$  muss  $x$  oder  $y$  enthalten. Also liegt jeder solche Pfad in der Zeichnung von  $G''$  auf der Seite von  $C$ , die  $xy$  enthält.

$\Rightarrow C$  liegt im Rand einer Facette von  $G''$

Damit ist *Behauptung 1* bewiesen.

2. Behauptung:  $G - x - y$  enthält keine zwei Knoten vom Grad 1.

*Beweis von Behauptung 2:*



- Angenommen  $u, v$  sind zwei Knoten in  $G - x - y$  mit Grad 1.
- Da  $\deg(u), \deg(v) \geq 3$  in  $G$ , sind  $ux, uy, vx, vy \in E(G)$  und  $u, v, x, y$  bilden ein  $\Theta$
- Nach Behauptung 1 hat jede Kante in  $G$  mindestens einen Endpunkt in  $u, v, x, y$ , um das  $\Theta$  bei Kontraktion einer beliebigen Kante zu zerstören.
- Jedes  $w \neq u, v, x, y$  ist zu  $u, v$  oder beiden benachbart, da  $\deg(w) \geq 3$ .
- Höchstens 2 Knoten außerhalb von  $u, v, x, y$ .



$\implies$  In allen Fällen ergibt sich ein Widerspruch zur Nicht-Planarität von  $G$ .

Damit ist *Behauptung 2* bewiesen.

**Definition:** Ein Graph enthält genau dann kein  $\Theta$ , wenn jede Kante auf höchstens einem Kreis liegt. Solche Graphen nennt man **Kakteen**. Kakteen sind kantendisjunkte Vereinigungen von Kreisen und Brücken.



**Definition:** Der **Block-Cutvertex-Tree** eines zusammenhängenden Graphen  $G$  (hier ist  $G$  Kaktus) ist ein Baum  $T$  mit:

- $V(T) = \{\text{Artikulationspunkt in } G\} \cup \{\text{Kreise in } G\} \cup \{\text{Brücken in } G\}$
- $E(T) = \{vb \mid v \text{ Artikulationspunkt, } b \text{ Brücke oder Kreis, } v \text{ Knoten auf } b \text{ in } G\}$



Behauptung 3:  $G - x - y$  ist tatsächlich ein Kreis.

*Beweis von Behauptung 3:*

- Sei  $T$  Block-Cutvertex-Tree von  $G - x - y$ .
- Wenn  $G - x - y$  keinen Artikulationspunkt enthält, ist  $G - x - y$  ein Kreis (Beweis fertig) oder eine Kante. Dann gilt  $|V(G)| \leq 4$ , da  $G$  höchstens nur die Knoten  $x, y$  und die Endpunkte der Kante enthält  $\implies G$  ist planar  $\nexists$
- Also gibt es Artikulationspunkte und  $|T| \geq 2$ .
- Also hat  $T$  mindestens 2 Blätter.
- Blätter im Block-Cutvertex-Tree sind entweder Brücken oder Kreise im ursprünglichen Graphen. Brücken führen immer zu Grad 1 Knoten. Nach Behauptung 2 gibt es ein Blatt in  $T$ , das in  $G$  ein Kreis  $C$  ist.
- Sei  $v$  der Artikulationspunkt in  $C$ , der  $C$  an den Graphen „klebt“.



- Jedes  $u \in V(C) - v$  hat Grad 2 in  $G - x - y$  (da Blatt im Cutvertex-Tree), aber mindestens Grad 3 in  $G$ .
- Also ist jedes  $u$  zu  $x$  oder  $y$  benachbart.  $u$  kann nicht zu beiden benachbart sein, da sonst  $G - v - w$  ein  $\Theta$  enthält.

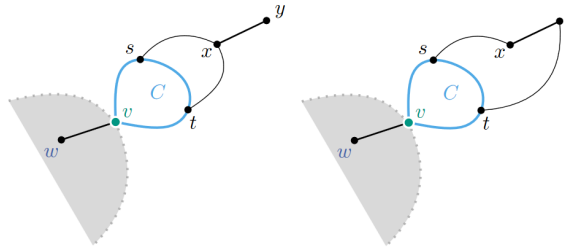


- Ebenso darf  $C$  nur Länge 3 haben, da sonst  $G - v - w$  ebenso wieder ein  $\Theta$  bildet.



$\Rightarrow C$  hat Länge genau 3  
 $V(C) = \{v, s, t\}$

- Dann hat  $C \cup \{x, y\}$  ein  $\Theta$  in  $G$ .



- Nach Behauptung 1 hat jede Kante mindestens einen Endpunkt im  $\Theta$ .
- Jedes  $w \in G - (C \cup \{x, y\})$  hat alle Nachbarn in  $C \cup \{x, y\}$ , sonst gäbe es eine Kante außerhalb des  $\Theta$ .  $w$  muss genau die Nachbarschaft  $\{x, y, v\}$  haben, denn  $w$  kann nicht zu  $s$  oder  $t$  benachbart sein, da diese Grad 2 haben.
- Würden zwei solche  $w, w'$  existieren, so wäre  $w, w', x, y$  ein  $\Theta$  in  $G - C$ .



$\Rightarrow$  Also ist  $w$  der einzige Knoten in  $G - (C \cup \{x, y\})$ .

- O.B.d.A sei  $sx \in E$ . Es gilt entweder  $ty \in E$  oder  $tx \in E$ .



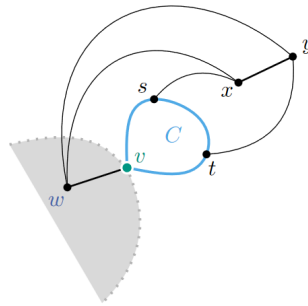
- Wir wissen, dass  $G$  nur die Knoten  $v, s, t, x, y, w$  besitzt.
- Wenn  $vx \in E$  oder  $vy \in E$ , dann gibt es ein  $\Theta$  in  $G - s - t$ .



- Wenn  $tx \in E$ , dann gibt es ein  $\Theta$  in  $G - w - y$ .



- Insgesamt wissen wir  $vx \notin E, vy \notin E, tx \notin E, ty \in E, ws \notin E, wt \notin E$ . Wir kennen also ganz  $G$  und  $G$  ist planar. Widerspruch.



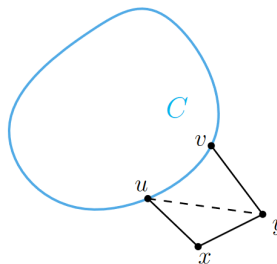
Damit sind *Behauptung 3* und das *Lemma* bewiesen.

*Beweis von Wagner - Abschluss:*

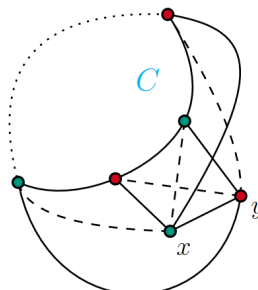
- Sei  $xy \in E$  eine Kante und  $C$  der Kreis  $G - x - y$ .
- Sei  $uv \in E$  eine Kante auf  $C$  mit  $ux \in E$ .

1. Fall:  $uy \notin E$ .

- $G - u - x$  ist ein Kreis, d.h.  $v$  muss Grad 2 haben, also ist  $vy \in E$ .



- Wenn  $vx \in E$  dann hat  $G - x - v$  einen Knoten  $u$  mit Grad 1.  $\nexists$   
 $\implies vx \notin E$
- Analoge Argumente liefern:  $N(x), N(y)$  sind auf  $C$  disjunkt und alternierend.
- $|C| \geq 4$  und wir finden einen  $K_{3,3}$ -Minor.



2. Fall: Jeder Knoten auf  $C$  ist zu  $x$  und  $y$  benachbart.

- $|C| \geq 3$ . Wir finden einen  $K_5$ -Minor.



Damit ist der **Satz von Wagner** bewiesen.

Wir beweisen nun  $(2) \implies (3)$ .

**Lemma:** Seien  $G, H$  Graphen. Maximaler Grad von  $H$  höchstens 3, d.h.  $\Delta(H) \leq 3$ . Dann sind äquivalent:

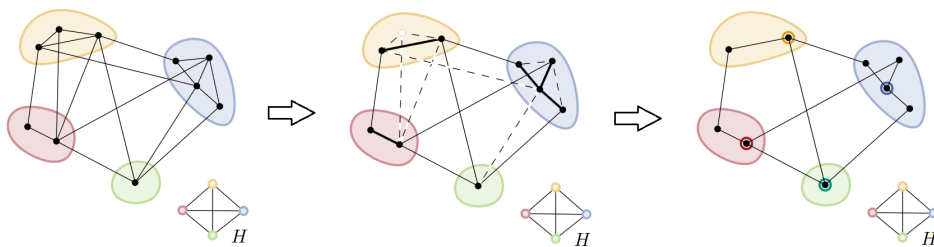
- $G$  enthält  $H$ -Minor
- $G$  enthält  $H$ -Unterteilung

*Beweis:* Die Richtung Unterteilung  $\implies$  Minor wurde bereits gezeigt. Beweise nun also die Rückrichtung. In einem  $H$ -Minor finden wir  $H$ -Unterteilung.

- O.B.d.A ist jede Kontraktionsmenge ein Baum, sodass
  - jedes Blatt hat Nachbarn in anderer Menge,
  - zwischen je zwei Mengen ist maximal eine Kante

Überflüssige Kanten können gelöscht werden.

- Wähle Knoten von maximalem Grad in jeder Menge.
- Dann bilden diese Bäume schon eine  $H$ -Unterteilung, da  $\Delta(H) \leq 3$ .



*Beweis von Kuratowski:* Es muss nur noch die Richtung  $G$  nicht planar  $\implies G$  enthält eine  $K_5$  oder  $K_{3,3}$ -Unterteilung bewiesen werden.

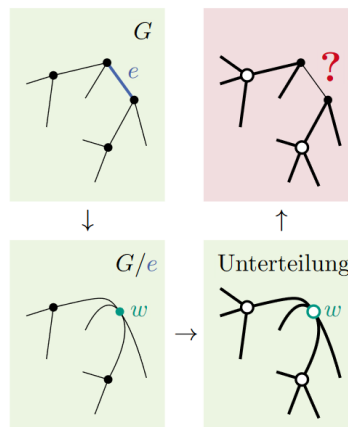
- Sei also  $G$  nicht planar. Wir müssen Unterteilung von  $K_{3,3}$  oder  $K_5$  finden.
- Nach Wagner gibt es einen  $K_{3,3}$ - oder  $K_5$ -Minor in  $G$ .
- Bei  $K_{3,3}$ -Minor, sind wir nach vorigem Lemma fertig.

Sonst: Induktion über Knotenzahl von  $G$ :

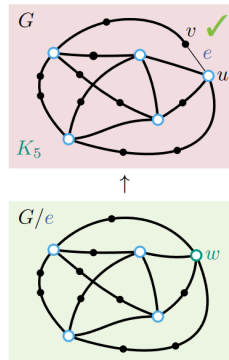
- I.A.:  $G$  muss mindestens 5 Knoten besitzen, um nicht-planar zu sein, und dort kommt auch nur  $K_5$  in Frage.
- I.S.: Wenn es sich beim  $K_5$ -Minor um einen  $K_5$ -Teilgraph handelt, dann sind wir fertig. Andernfalls gibt es  $e = uv$ , sodass  $G/e$  immer noch einen  $K_5$ -Minor enthält.  $G/e$  ist also immer noch nicht-planar. Nach IV existiert eine  $K_{3,3}$ - oder  $K_5$ -Unterteilung in  $G/e$ . Sei  $w$  der Knoten, zu dem  $e$  kontrahiert wird.
  - Wenn  $w$  in der Unterteilung ein Unterteilungspunkt ist (also  $\deg(w) = 2$ ), gibt es auch solch eine Unterteilung in  $G$ .
  - Wenn  $\deg(w) = 3$  in Unterteilung, gibt es auch in  $G$  eine Unterteilung.



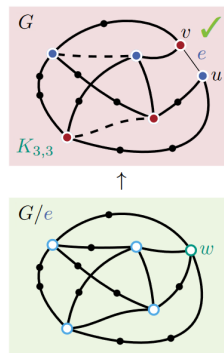
- Also o.B.d.A.  $\deg(w) = 4$  in  $K_5$ -Unterteilung in  $G/e$ .



- Betrachte die vier anderen Knoten von Grad 4.
- Sind mindestens drei davon zu  $u$  verbunden, finden wir wieder eine  $K_5$ -Unterteilung in  $G$ .



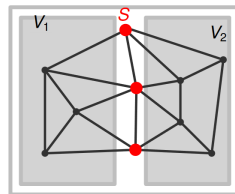
- Andernfalls sind zwei zu  $u$  und zwei zu  $v$  verbunden und wir finden eine  $K_{3,3}$ -Unterteilung in  $G$ .



Damit wurde der **Satz von Kuratowski** bewiesen.

## 5 Separatoren in planaren Graphen

**Definition:** Eine Menge  $S \subset V$  heißt Separator von  $G = (V, E)$ , falls der durch  $V \setminus S$  induzierte Subgraph von  $G$  unzusammenhängend ist.



**Minimum-Balanced-Separator-Problem:** Gegeben sei ein Graph  $G = (V, E)$ . Finde eine Partition von  $V$  in drei Mengen  $V_1, V_2$  und  $S$ , wobei der Separator  $S$  minimale Kardinalität hat und  $V_1$  von  $V_2$  trennt mit  $|V_1|, |V_2| \leq \alpha \cdot |V|$  und  $\frac{1}{2} \leq \alpha < 1$  konstant.

- Separator soll also klein sein
- Separator soll etwa gleich große Teilgraphen erzeugen
- Problem ist  $\mathcal{NP}$ -schwer

**Planar-Separator-Theorem:** Die Knotenmenge eines zusammenhängenden, planaren Graphen  $G = (V, E)$ ,  $n = |V| \geq 5$ , kann so in drei Mengen  $V_1, V_2, S \subseteq V$  partitioniert werden, dass

- $|V_1|, |V_2| \leq \frac{2}{3} \cdot n$
- $S$  ist ein Separator, der  $V_1$  von  $V_2$  trennt
- $|S| \leq 4 \cdot \sqrt{n}$

Eine solche Partition kann in  $\mathcal{O}(n)$  Zeit konstruiert werden.

Für den Beweis dieses Satzes benötigen wir folgendes Lemma.

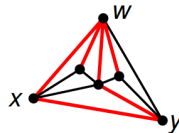
**Lemma:** Sei  $G = (V, E)$  ein planarer, zusammenhängender Graph mit  $n = |V| \geq 5$  und  $T = (V, E(T))$  ein Spannbaum von  $G$  mit Wurzel  $w$  und Höhe  $h$ . Die Knotenmenge von  $G$  kann so in drei Mengen  $V_1, V_2$  und  $S$  partitioniert werden, dass

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

Eine solche Partition kann in  $\mathcal{O}(n)$  Zeit konstruiert werden.

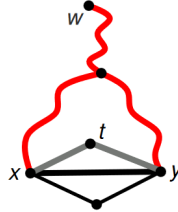
*Beweis:*

- Konstruiere eine Triangulierung von  $G$ . Nach Satz von Euler hat der neue Graph  $3n - 6$  Kanten und  $2n - 4$  Facetten.
- Spannbaum  $T$  von  $G$  ist Spannbaum des triangulierten Graphen
- In  $T$  induziert jede Nichtbaumkante  $\{x, y\}$  einen Kreis  $K_{x,y}$  mit  $\leq 2 \cdot h + 1$  Knoten (maximal  $h$  Knoten in beide Richtungen + Wurzel)



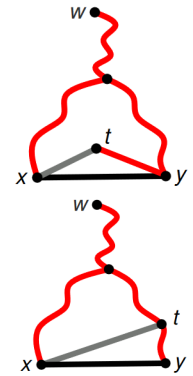
- Sei  $\text{Inneres}(K_{x,y})$  die Knoten, die innerhalb des Kreises, aber nicht auf dem Rand des Kreises liegen. Definiere  $\text{Äußeres}(K_{x,y})$  dementsprechend.
- Wähle Nichtbaumkante  $\{x, y\}$  aus, wobei  $|\text{Inneres}(K_{x,y})| \geq |\text{Äußeres}(K_{x,y})|$
- Wenn  $|\text{Inneres}(K_{x,y})| \leq \frac{2}{3}n$ , dann gilt das Lemma und wir sind fertig

- Sei also  $|\text{Inneres}(K_{x,y})| > \frac{2}{3}n$ , dann ist  $|\text{Äußeres}(K_{x,y})| < \frac{1}{3}n$
- Ziel: Ersetze  $\{x, y\}$  durch eine andere Nichtbaumkante, sodass das Innere kleiner wird und das Äußere nicht über  $\frac{2}{3}n$  wächst
- Da Graph trianguliert, begrenzt Kante  $\{x, y\}$  zwei Dreiecke, von denen eins im Inneren( $K_{x,y}$ ) liegt  $\implies$  Dreieck  $x y t$



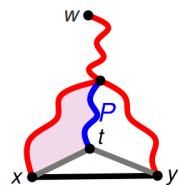
Fall 1:  $\{x, t\}$  oder  $\{t, y\}$  ist eine Baumkante, o.B.d.A sei  $\{t, y\}$  eine Baumkante. Ersetze  $\{x, y\}$  durch  $\{x, t\}$ .

- Falls  $t \notin K_{x,y}$ :
  - $|\text{Äußeres}(K_{x,t})| = |\text{Äußeres}(K_{x,y})|$
  - $|\text{Inneres}(K_{x,t})| = |\text{Inneres}(K_{x,y})| - 1$
- Falls  $t \in K_{x,y}$ :
  - $|\text{Äußeres}(K_{x,t})| = |\text{Äußeres}(K_{x,y})| + 1$
  - $|\text{Inneres}(K_{x,t})| = |\text{Inneres}(K_{x,y})|$



Fall 2:  $\{x, t\}$  und  $\{t, y\}$  sind beides Nichtbaumkanten.

- Sei  $|\text{Inneres}(K_{x,t})| \geq |\text{Inneres}(K_{t,y})|$ . Ersetze  $\{x, y\}$  durch  $\{x, t\}$ .



- $|\text{Äußeres}(K_{x,t})| \leq n - (|\text{Inneres}(K_{x,t})| + P) \leq n - \frac{1}{2}|\text{Inneres}(K_{x,y})| < n - \frac{1}{2} \cdot \frac{2}{3}n = \frac{2}{3}n$
- $|\text{Inneres}(K_{x,t})| \leq |\text{Inneres}(K_{x,y})| - 1$

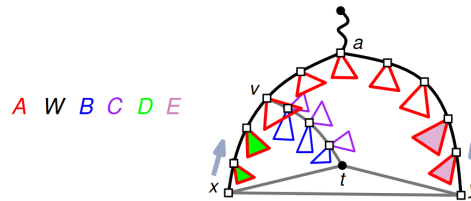
In beiden Fällen verkleinern wir  $|\text{Inneres}(K_{x,y})|$  und lassen  $|\text{Äußeres}(K_{x,y})|$  klein genug. Dies kann nun so lange wiederholt werden, bis auch  $|\text{Inneres}(K_{x,y})| \leq \frac{2}{3}n$  gilt.

$\implies$  Partition mit den gewünschten Eigenschaften lässt sich konstruieren. Wir müssen nun noch deren Implementation in linearer Laufzeit sicherstellen.

- Triangulierung des Graphen in  $\mathcal{O}(n)$  möglich  $\rightarrow$  Übung



- Ersetzung einer Nichtbaumkante durch eine andere, welche die Anzahl der Dreiecke im Inneren reduziert  $\implies$  Höchstens  $2n - 4$  Schritte
- In Fall 1 können wir  $|\text{Inneres}(K_{x,y})|$  und  $|\text{Äußeres}(K_{x,y})|$  in  $\mathcal{O}(1)$  berechnen
- Für Fall 2 muss entschieden werden, ob  $|\text{Inneres}(K_{x,t})|$  oder  $|\text{Inneres}(K_{t,y})|$  größer ist. Zeige, dass auch dieser Fall nur konstante Zeit benötigt mithilfe einer amortisierten Analyse.
- Führe dazu folgende Vorberechnung durch:
  - Durchlaufe  $T$  von den Blättern zur Wurzel
  - Speichere für jeden Knoten und inzidente Baumkanten die Anzahl Knoten im Unterbaum links bzw. rechts der Kante und markiere den Knoten
  - Dies kann einmalig in Linearzeit durchgeführt werden
- Laufe von  $t$  nach oben bis zum ersten markierten Knoten  $v$  und berechne die Anzahl der Knoten rechts und links des Weges
- Laufe von  $x$  und  $y$  abwechselnd in Richtung Wurzel bis erstmals  $v$ , d.h. Weg von  $t$  zur Wurzel, erreicht wird.



- $|\text{Inneres}(K_{x,t})| = D + B$
- $|\text{Inneres}(K_{t,y})| = A - D - B - W$

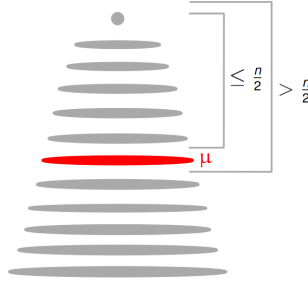
Die Anzahl der Operationen in einem Schritt ist proportional zu der Anzahl der Knoten in dem Teil von  $K_{x,y}$ , der nicht weiter betrachtet wird. Also ist auch Fall 2 in amortisiert konstanter Zeit implementierbar.

Damit ist auch die Laufzeit und somit das gesamte Lemma bewiesen.

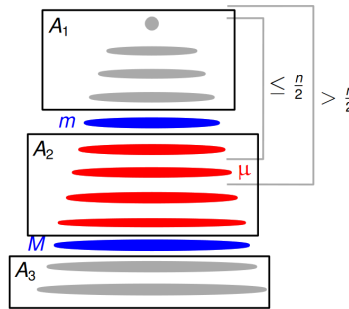
**BFS-Lemma:** Sei  $T = (V, E(T))$  ein BFS-Baum von  $G = (V, E)$ . Für eine Nichtbaumkante  $\{u, v\}$  gilt  $|\text{level}(u) - \text{level}(v)| \leq 1$ .

*Beweis des Planar-Separator-Theorem:*

- Konstruiere eine Triangulierung von  $G$  und ein BFS-Baum  $T$  mit beliebiger Wurzel
- Sei  $\mu$  das Level mit der Eigenschaft:



- Wenn  $|\text{level } \mu| \leq 4\sqrt{n}$ , dann ist  $\mu$  ein geeigneter Separator und wir sind fertig.
- Sei also  $|\text{level } \mu| > 4\sqrt{n}$ .
- Sei  $m$  das unterste Level oberhalb von  $\mu$  und  $M$  das oberste Level unterhalb von  $\mu$  mit  $|\text{level } m| < \sqrt{n}$  und  $|\text{level } M| < \sqrt{n}$ .



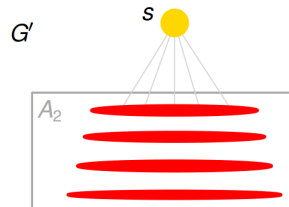
- Offensichtlich gilt  $|A_1| \leq \frac{n}{2}$  und auch  $|A_3| \leq \frac{n}{2}$ , da schon  $> \frac{n}{2}$  Knoten über  $\mu$

Fall 1:  $|A_2| \leq \frac{2}{3}n$

- $S = \text{level } m \cup \text{level } M$  ist Separator
- $V_1 = \max\{A_1, A_2, A_3\}, |V_1| \leq \frac{2}{3}n$
- $V_2 = V \setminus (S \cup V_1), |V_2| \leq n - |V_1| \leq n - \frac{|V_2|}{2}$ , da  $|V_1| \geq \frac{|V_2|}{2}$ , sonst wäre  $|V_1|$  nicht maximal, da  $V_2$  ein größeres  $A_i$  beinhaltet  $\Rightarrow |V_2| \leq \frac{2}{3}n$
- Damit wurde ein geeigneter Separator gefunden und wir sind fertig

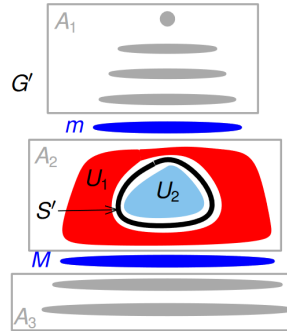
Fall 2:  $|A_2| > \frac{2}{3}n$

- Verschmelze die Knoten in  $A_1 \cup \text{level } m$  zu einem Knoten  $s$  und entferne alle Knoten aus  $\text{level } M \cup A_3$ . Dadurch entsteht ein neuer Graph  $G' = (V', E')$ .

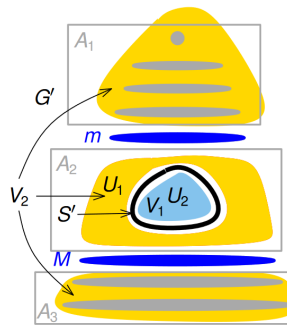


- BFS-Baum  $T$  induziert BFS-Baum  $T'$  in  $G'$

- $T'$  hat maximal Höhe  $\sqrt{n}$ , da  $|V'| \leq n$  und durch die Wahl von  $m$  und  $M$  für jede Schicht  $S_i$  zwischen  $m$  und  $M$   $|S_i| \geq \sqrt{n}$  gilt
- Wende obiges Lemma auf  $G'$  und  $T'$  an und erhalte  $S', U_1, U_2$



- Sei  $S = S' \cup \text{level } m \cup \text{level } M$
- Nach dem Lemma folgt  $|S'| \leq 2\sqrt{n} + 1$ , also  $|S| \leq 4\sqrt{n}$
- Sei  $V_1 = \max\{U_1, U_2\}$ . Nach dem Lemma gilt  $|V_1| \leq \frac{2}{3}n$
- Weiterhin gilt  $|V_1| + |S| > |V_1| + |S'| > \frac{1}{2} \cdot |A_2|$ . Setzt man also  $V_2 = V \setminus (S \cup V_1)$ , dann gilt  $|V_2| = n - |V_1| - |S| < n - \frac{1}{2} \cdot |A_2| < \frac{2}{3}n$



Auch hier findet man also einen geeigneten Separator, womit das **Planar-Separator-Theorem** bewiesen ist.

## 6 Matchings und Maximum Independent Set

**MAXIMUM INDEPENDENT SET:** Für  $G = (V, E)$ , finde eine **größte unabhängige Menge**. Also Knotenmenge  $I \subseteq V$  mit  $|I|$  maximal, sodass jede Kante in  $E$  höchstens einen Endpunkt in  $I$  hat.

**Approximationsalgorithmus für MAXIMUM INDEPENDENT SET:**

1. Zerkleinere den Graphen mit Planar Separator, bis Komponenten nur noch  $\mathcal{O}(\log \log n)$  Knoten haben

2. Löse Komponenten mit Brute-Force in  $\mathcal{O}(2^{\log \log n}) = \mathcal{O}(\log n)$  Zeit pro Komponente  $\rightarrow \mathcal{O}(n \log n)$  Gesamtlaufzeit
3. Zusammenfügen ist disjunkte Vereinigung der Teillösungen (Kein Problem an Schnittpunkten, da Separator dazwischen)

### Güte der Approximation:

- **Satz:** Wiederholtes Anwenden des planar Separators gibt Komponenten der Größe  $\mathcal{O}(r)$  bei Separator-Gesamtgröße  $\mathcal{O}(n/\sqrt{r})$

*ohne Beweis.*

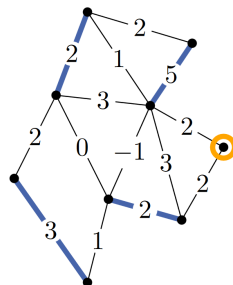
- Mit obigem Satz folgt Separator-Gesamtgröße  $|S| \leq \mathcal{O}(n/\sqrt{\log \log n})$
- Für optimale Lösung  $OPT(G)$  gilt  $OPT(G) \geq n/4$  nach Vier-Farben-Satz
- Damit folgt:

$$\begin{aligned}
 OPT(G) - A(G) &\leq |S| \leq \mathcal{O}\left(\frac{n}{\sqrt{\log \log n}}\right) \\
 \Rightarrow A(G) &\geq OPT(G) - \mathcal{O}\left(\frac{n}{\sqrt{\log \log n}}\right) \\
 &\geq OPT(G) - \mathcal{O}\left(\frac{OPT(G)}{\sqrt{\log \log n}}\right) \\
 &= OPT(G) \cdot \left(1 - \mathcal{O}\left(\frac{1}{\sqrt{\log \log n}}\right)\right)
 \end{aligned}$$

wobei  $OPT(G) - A(G) \leq |S|$  gilt, da  $A$  für die Separatoren keine Lösungen berechnet und somit die Abweichung von der optimalen Lösung max. so groß ist wie die Kardinalität von  $S$ . Diese Approximation ist bestmöglich wenn  $\mathcal{P} \neq \mathcal{NP}$ .

### Definition:

- Kantenmenge  $M$  ist ein **Matching** wenn jeder Knoten zu höchstens einer Kante in  $M$  inzident ist.
- Wenn ein Knoten  $v$  zu einer Kante in  $M$  inzident ist, heißt  $v$  **gematcht**, ansonsten **ungematcht**.

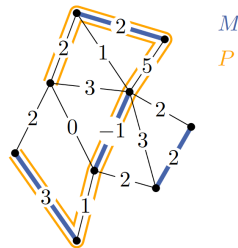


# GEWICHTSMAXIMALES MATCHING:

- **Gegeben:** Graph  $G = (V, E)$  und Gewichtsfunktion  $w: E \rightarrow \mathbb{R}$
- **Gesucht:** Matching  $M \subseteq E$  mit  $w(M) = \sum_{e \in M} w(e)$  maximal

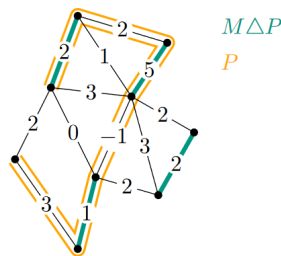
**Definition:** Sei  $M \subseteq E$  Matching in  $(G = (V, E), w)$ . Ein **M-alternierender Weg** ist ein einfacher Pfad oder Kreis  $P$  in  $G$ , sodass

- sich Kanten von  $M$  und  $E - M$  auf  $P$  abwechseln
- wenn  $P$  ein Pfad mit Endpunkt  $v$  und Kante  $e$  an  $v$  in  $P$  ist, dann ist  $e \in M$  oder  $v$  ungematcht (verhindert, dass der Pfad verlängert werden kann)



Für Matching  $M$  und alternierenden Weg  $P$  ist auch  $M \Delta P := (M - P) \cup (P - M)$  (**symmetrische Differenz**) ein Matching.

Dabei gilt  $w(M \Delta P) - w(M) = w(P - M) - w(P \cap M)$ , denn sowohl neues als auch altes Matching enthalten die Kanten, die nicht auf dem Pfad liegen. Dann ist die Differenz zwischen neuem und altem Matching nur die Kanten von  $M \Delta P$  auf dem Pfad, was  $P - M$  ist, und die Kanten von  $M$  auf dem Pfad, was  $P \cap M$  ist.



$$2 + 5 + 1 > 2 - 1 + 3$$

$$w(P - M) > w(P \cap M)$$

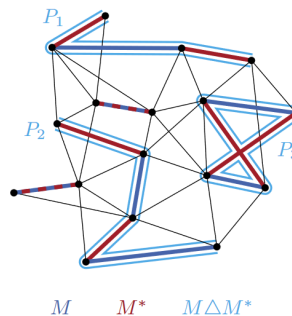
**Defintion:** Ein alternierender Weg heißt **erhöhend** wenn  $w(M \Delta P) > w(M)$ , also  $w(P - M) > w(P \cap M)$ .

**Lemma:** Sei  $M$  ein Matching in  $(G, w)$ . Es sind äquivalent:

- $M$  ist gewichtsmaximal
- Es gibt keinen erhöhenden alternierenden Weg bezüglich  $M$

*Beweis:*

- „ $\Rightarrow$ “: Falls es zu  $M$  einen erhöhenden Weg gibt, so kann  $M$  natürlich nicht maximales Gewicht haben
- „ $\Leftarrow$ “:
  - Sei  $M$  nicht gewichtsmaximal, also gibt es Matching  $M^*$  mit  $w(M^*) > w(M)$
  - Betrachte  $M \Delta M^* = (M \cup M^*) \setminus (M \cap M^*)$



- $M \Delta M^*$  hat nur Knoten vom Grad 1 oder 2, besteht also aus einfachen Kreisen und Wegen  $P_1, \dots, P_t$
- Jedes  $P_i$  ist  $M$ -alternierender Weg
- Es gilt  $w(M^*) - w(M) = \sum_{i=1}^t (w(M^* \cap P_i) - w(M \cap P_i))$
- Ein Summand ist positiv, da  $w(M^*) - w(M) > 0$
- Einer der  $P_i$  ist also erhöhend, mit  $w(M^* \cap P_i) > w(P_i \cap M) \implies$  Es gibt also einen erhöhenden Weg. Widerspruch.

**Algorithmus für gewichtsmaximales Matching in planaren Graphen:**

1. Falls  $|V(G)| \leq 5$ , finde gewichtsmaximales Matching durch Brute-Force
2. Falls  $|V(G)| > 5$ :
  - Finde  $\frac{2}{3}$ -balancierten Separator  $S$  mit  $|S| = \mathcal{O}(\sqrt{n})$
  - Berechne optimale Matchings auf allen Komponenten von  $G' := G - S$
  - Sei  $M'$  die Vereinigung dieser optimalen Matchings.  $M'$  ist optimal für  $G'$

3. Solange  $S \neq \emptyset$ :

- Wähle  $v \in S$ . Finde alternierenden Weg  $P$  in  $G' + v$  mit Endpunkt  $v$  mit  $w(P - M') - w(P \cap M')$  maximal
- Falls  $P$  erhöhend, ersetze  $M'$  durch  $M' \Delta P$
- Lösche  $v$  aus  $S$
- Ersetze  $G'$  durch  $G' + v$

Im Folgenden wollen wir die Korrektheit des Algorithmus beweisen und dessen Laufzeit bestimmen. Mit folgendem Lemma folgt die Korrektheit.

**Lemma:** Sei  $G = (V, E)$  ein Graph,  $v \in V$  ein Knoten,  $M$  ein gewichtsmaximales Matching in  $G - v$ .

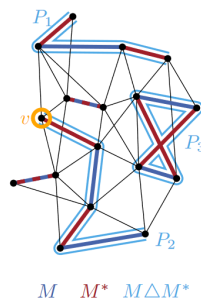
- $M$  ist gewichtsmaximal in  $G \iff$  Es ex. kein erhöhender Weg mit Endpunkt  $v$

Wenn ein erhöhender Pfad  $P$  mit Endpunkt  $v$  und  $M' = M \Delta P$  existiert, dann gilt:

- $M'$  ist gewichtsmaximal in  $G \iff$  Differenz  $w(P - M) - w(P \cap M)$  ist maximal unter all solchen Pfaden mit Endpunkt  $v$

*Beweis - Teil 1:*

- „ $\Rightarrow$ “: Wenn es einen solchen erhöhenden Weg gäbe, dann kann  $M$  nicht gewichtsmaximal in  $G$  sein.
- „ $\Leftarrow$ “:
  - Sei  $M$  nicht gewichtsmaximal in  $G$ , dann gibt es ein Matching  $M^*$  mit  $w(M^*) > w(M)$
  - Betrachte Pfade und Kreise  $P_1, \dots, P_t$  in  $M \Delta M^*$
  - Analog zum letzten Beweis gibt es  $P_i$  mit  $w(M^* \cap P_i) > w(M \cap P_i)$
  - Wenn  $v \notin P_i$ , dann ist  $P_i$  erhöhend für  $M$  in  $G - v \implies$  Widerspruch, da angenommen wurde, dass  $M$  optimales Matching für  $G - v$  ist
  - Also ist  $v \in P$ . Da  $v \notin M$ , weil  $M$  Matching für  $G - v$  ist, ist  $v$  ein Endpunkt von  $P_i$



*Beweis - Teil 2:*

- „ $\Rightarrow$ “: Klar. Wenn  $M'$  maximal ist, dann kann es keinen besseren Pfad geben.
- „ $\Leftarrow$ “:
  - Sei nicht  $M'$ , sondern  $M^*$  gewichtsmaximal in  $G$
  - Analog zu oben gibt es erhöhenden Pfad  $P$  in  $M \Delta M^*$  mit  $v$  als Endpunkt (da nur Komponente mit  $v$  zu Verbesserung führen kann, weil andere Komponenten auch von  $M$  betrachtet wurden) und  $w(M^*) - w(M) = w(P - M) - w(P \cap M)$
  - Da  $w(M^*) - w(M) > w(M') - w(M)$  war Pfad für  $M'$  nicht maximal

**Satz:** Ein gewichtsmaximales Matching eines planaren Graphen mit  $n$  Knoten kann in  $\mathcal{O}(n^{\frac{3}{2}} \log n)$  berechnet werden.

*Beweis:* Siehe Algorithmus für gewichtsmaximales Matching in planaren Graphen.

- 1. geht in  $\mathcal{O}(1)$
- Finden eines  $\frac{2}{3}$ -balancierten Separator in  $\beta \cdot n$  Schritten
- Finden von  $|S|$  erhöhenden Wegen mit Endpunkt  $v$  in  $|S| \cdot \mathcal{O}(n \log n) \leq \gamma \cdot n^{\frac{3}{2}} \log n$  Schritten
- Sei  $T(n)$  die worst-case Gesamtlaufzeit, dann ist

$$T(n) \leq T(n_1) + T(n_2) + \beta \cdot n + \gamma \cdot n^{\frac{3}{2}} \log n$$

wobei  $n_1, n_2$  die Anzahl der Knoten der Teilgraphen nach dem Separieren ist

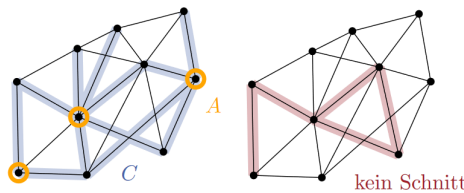
- Der Rest vom Beweis ist viel Mathe und nicht relevant für die Klausur

## 7 Mixed Max Cut

**Definition:** Ein **Schnitt** in  $G = (V, E)$  ist eine Kantenmenge  $C \subseteq E$ , die von einer Knotenmenge  $A \subseteq V$  folgendermaßen induziert wird:

$$C = \{uv \in E \mid |A \cap \{u, v\}| = 1\}$$

$C$  enthält also genau die Kanten, die genau einen Endpunkt in  $A$  haben.





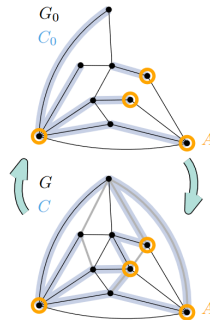
MIXED-MAX-CUT:

- **Gegeben:** Graph  $G = (V, E)$  und Gewichtsfunktion  $w: E \rightarrow \mathbb{R}$
- **Gesucht:** Schnitt  $C \subseteq E$  mit  $w(C) = \sum_{e \in C} w(e)$  maximal und  $C \neq \emptyset$

**Satz:** MIXED-MAX-CUT ist auf planaren Graphen polynomiell lösbar.

*Beweis:* Gegeben einen Graphen  $G_0 = (V, E_0)$  und  $w: E_0 \rightarrow \mathbb{R}$ .

1. Trianguliere  $G_0$  zu  $G = (V, E)$  und setze  $w(e) = 0$  für jede Kante  $e \in E \setminus E_0$



**Beobachtung:** Für Schnitt  $C$  in  $G$  und  $C_0 = C \cap E_0$  in  $G_0$  sind äquivalent:

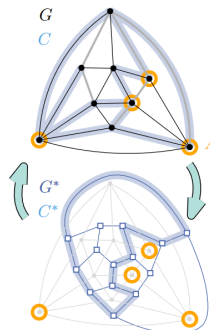
- $A \subseteq V$  induziert  $C_0 \subseteq E_0$  in  $G_0$
- $A \subseteq V$  induziert  $C \subseteq E$  in  $G$

Außerdem gilt  $w(C_0) = w(C)$ , also reicht es im Folgenden den triangulierten Graphen anzuschauen.

**Achtung:**  $C_0 = \emptyset$  könnte gelten! Das wird später behoben.

2. Betrachte Dualgraph  $G^* = (F, E^*)$  von  $G = (V, E)$ .

- Setze  $w(e^*) = w(e)$  für alle  $e \in E$
- $G^*$  ist 3-regulär, d.h. jeder Knoten hat Grad 3
- Für jede Kantenmenge  $C^* \subseteq E^*$  hat jeder Dualknoten 0, 1, 2 oder 3 inzidente Kanten in  $C^*$



**Definition:** Kantenmenge  $X \subseteq E^*$  heißt **gerade**, wenn jeder Knoten zu gerade vielen Kanten in  $X$  inzident ist.

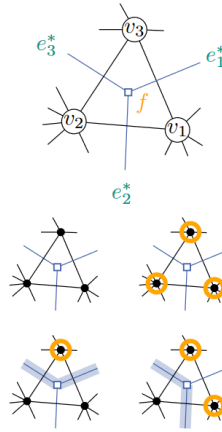
**Lemma:** Sei  $G = (V, E)$  trianguliert,  $G^*$  zu  $G$  dual. Dann gilt:

$$C \subseteq E \text{ ist Schnitt} \iff C^* \subseteq E^* \text{ ist eine gerade Kantenmenge}$$

Außerdem ist  $w(C) = w(C^*)$ .

*Beweis:*

- „ $\Rightarrow$ “: Sei  $C \subseteq E$  Schnitt in  $G$  induziert von  $A \subseteq V$ . Sei  $f \in V(G^*)$  und  $e_1^*, e_2^*, e_3^*$  seine drei inzidenten Kanten. Betrachte das zu  $f$  zugehöriges Dreieck  $v_1, v_2, v_3$  in  $G$ .



- Ist  $|A \cap \{v_1, v_2, v_3\}| = 0, 3$ , dann  $|C^* \cap \{e_1^*, e_2^*, e_3^*\}| = 0$
- Ist  $|A \cap \{v_1, v_2, v_3\}| = 1, 2$ , dann  $|C^* \cap \{e_1^*, e_2^*, e_3^*\}| = 2$

Also ist  $C^*$  gerade.

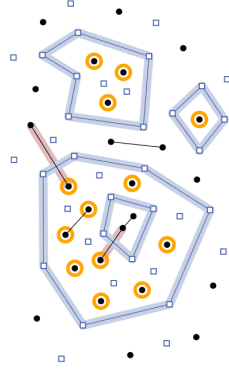
- „ $\Leftarrow$ “: Sei  $C^* \subseteq E^*$  eine gerade Kantenmenge in  $G^*$ . Dann hat jeder Dualknoten 0 oder 2 inzidente Kanten in  $C^*$ , also ist  $C^*$  eine disjunkte Vereinigung von Kreisen und isolierten Punkten  $C_1, \dots, C_k$ . Sei nun

$$A = \{v \in V \mid v \text{ ist im Inneren von ungerade vielen Kreisen}\}$$

Dann gilt:

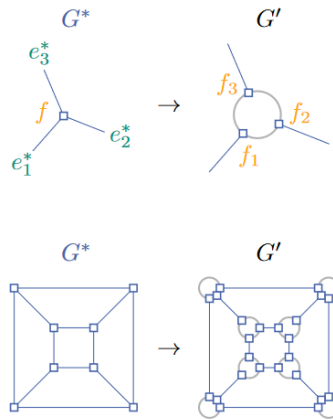
$$\begin{aligned} e \in E \text{ ist in } C & \\ \iff e^* \in C^* & \\ \iff e^* \in C_i \text{ für ein } i \in \{1, \dots, k\} & \\ \iff \text{Endpunkte von } e \text{ liegen auf verschiedenen Seiten von } C_i & \\ \iff \text{Genau einer der Endpunkte von } e \text{ ist in } A & \end{aligned}$$

Also ist  $C$  ein Schnitt und wird von  $A$  induziert.



Wir suchen jetzt also eine gewichtsmaximale gerade Kantenmenge  $C^*$  in  $G^*$ . Das heißt, jeder Knoten hat Grad 0 oder 2 in  $C^*$ .

3. Modifiziere  $G^* = (F, E^*)$  zu  $G' = (V', E')$  wie folgt:



Ursprüngliche Kanten behalten ihr Gewicht und neue Kanten erhalten Gewicht 0.  $G'$  ist wieder planar und 3-regulär.

**Definition:** Sei  $k \in \mathbb{N}$  eine Zahl. Eine Kantenmenge  $X \subseteq E$  heißt **k-Faktor**, wenn jeder Knoten zu genau  $k$  Kanten in  $X$  inzident ist.

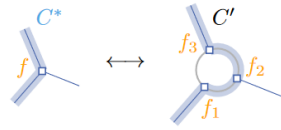
- 1-Faktoren heißen auch **perfekte Matchings**

**Lemma:**

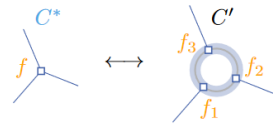
- Für jede gerade Menge  $C^* \subseteq E^*$  ex. ein 2-Faktor  $C' \subseteq E'$  mit  $C' \cap E^* = C^*$
- Für jeden 2-Faktor  $C' \subseteq E'$  ist  $C^* = C' \cap E^*$  eine gerade Menge
- Es gilt  $w(C') = w(C^*)$

*Beweis:*

1. Fall  $f \in F$  hat zwei inzidente Kanten in  $C^*$ :



2. Fall  $f \in F$  hat keine inzidente Kante in  $C^*$ :



Wir suchen jetzt also einen gewichtsmaximalen 2-Faktor  $C'$  in  $G'$ .

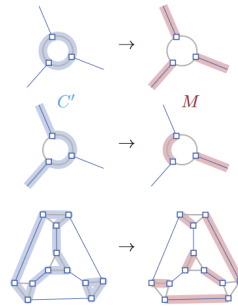
4. Betrachte 1-Faktoren (perfekte Matchings) statt 2-Faktoren.

- Da  $G'$  3-regulär ist, ist das Komplement eines 2-Faktors  $C'$  in  $G'$  ein perfektes Matching  $M$ .

$$M = E' - C'$$

- 2-Faktor  $C'$  ist gewichtsmaximal genau dann, wenn das komplementäre perfekte Matching  $M$  gewichtsminimal ist

$$w(M) = w(E') - w(C')$$



Damit haben wir einen Algorithmus angegeben, der das Problem löst. Im Folgenden betrachten wir die Laufzeit.

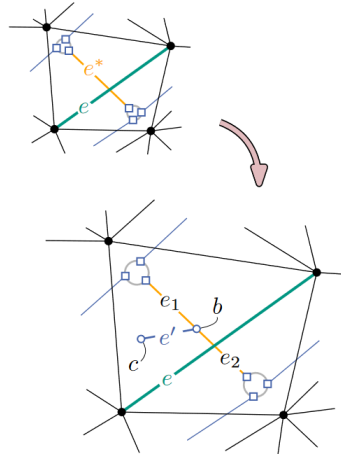
**Satz:** In planaren Graphen können gewichtsm minimale perfekte Matchings in  $\mathcal{O}(n^{\frac{3}{2}} \log n)$  berechnet werden.

*Beweis:* Reduziere auf gewichtsmaximales Matching.

- $w'(e) := -w(e)$ , d.h. maximal bezüglich  $w' \iff$  minimal bezüglich  $w$
- $w''(e) := W + w'(e)$  für großes  $W > |V| \cdot \max_{e \in E'}(|w'(e)|)$ , also hat max. Matching bzgl.  $w''$  die größtmögliche Anzahl Kanten. Damit sind gewichtsmaximale Matchings bzgl.  $w''$  perfekt.
- Max. Matchings bzgl.  $w''$  entsprechen also genau den min. perfekten Matchings bezüglich  $w$ .
- Da MAX MATCHING in  $\mathcal{O}(n^{\frac{3}{2}} \log n)$  ist, ist der Satz bewiesen.

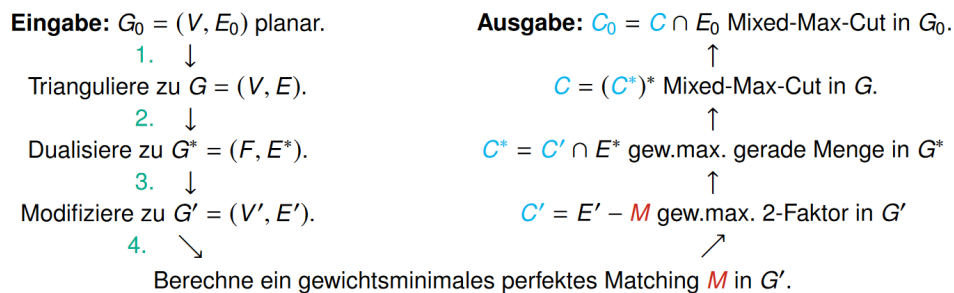
Nun muss nur noch  $C_0 \neq \emptyset$  sichergestellt werden.

- Für eine Kante  $e \in E_0$  wollen wir erzwingen, dass  $e \in C_0$
- Also soll  $e^*$  nicht in  $M$  sein:
  - Unterteile dafür  $e^*$  mit Knoten  $b$  und setze für die neu entstandenen Kanten  $e_1, e_2$  die Gewichte  $w(e_1) = w(e^*)$  und  $w(e_2) = 0$
  - Füge Kante  $e' = bc$  mit neuem Knoten  $c$  hinzu und setze  $w(e') = 0$
  - Jedes perfekte Matching muss dann  $e'$  enthalten

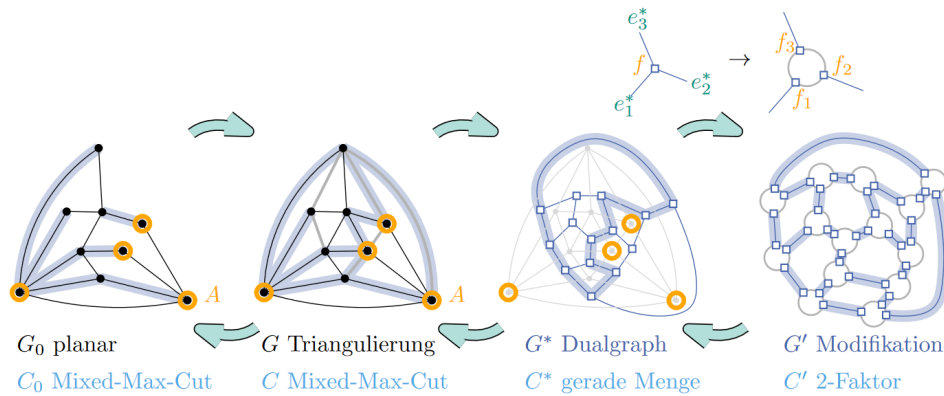


- Um besten Schnitt zu erhalten, wiederholt man den Vorgang für jede Kante  $e \in E_0$
- Wir erhalten Laufzeit  $\mathcal{O}(n^{\frac{3}{2}} \log n) \cdot \mathcal{O}(n) = \mathcal{O}(n^{\frac{5}{2}} \log n)$ .  $\mathcal{O}(n^{\frac{3}{2}} \log n)$  ist aber möglich!

Damit ist der Beweis abgeschlossen. Es folgt eine Übersicht über den Algorithmus.



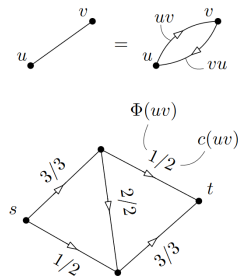
Da alle Schritte bis auf Berechnung von  $M$  nur  $\mathcal{O}(n)$  Zeit brauchen, ist MIXED-MAX-CUT in  $\mathcal{O}(n^{\frac{3}{2}} \log n)$ .



## 8 Flussnetzwerke und Maximum-Flow

Wir betrachten im Folgenden gerichtete Graphen  $D = (V, A)$

- Kante von  $u$  nach  $v$  heißt  $uv$  und es gilt  $uv \neq vu$
- Wir nehmen an, dass  $uv \in A \iff vu \in A$



**Definition:** Ein **Flussnetzwerk** ist ein 4-Tupel

$$(D = (V, A), c: A \rightarrow \mathbb{R}_{>0}, s \in V, t \in V)$$

wobei  $D$  wie oben,  $c$  jeder Kante ihre Kapazität zuordnet,  $s$  die Quelle und  $t$  die Senke darstellen. Die Kapazität einer Kante ist in beide Richtungen gleich, also

$$c(uv) = c(vu) \quad \forall uv \in A$$

**Definition:** Ein **s-t-Fluss** bezüglich eines Flussnetzwerkes ist eine Funktion  $\Phi: A \rightarrow \mathbb{R}$ , die jeder Kante  $uv$  ihren Fluss von  $u$  nach  $v$  zuordnet und Folgendes einhält:

- Flusserhaltung:  $\sum_{uv \in A} \Phi(uv) = 0$  für jeden Knoten  $u \neq s, t$   
(Netto-Ausfluss aus  $u$  muss 0 sein, da Einfluss negativ gezählt wird)
- Zulässigkeit:  $\Phi(uv) \leq c(uv)$  für alle  $uv \in A$
- Antisymmetrie:  $\Phi(uv) = -\Phi(vu)$  für alle  $uv \in A$

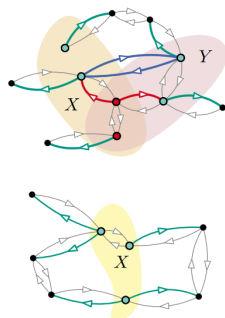
Der **Wert** von  $\Phi$  ist der Netto-Ausfluss bei  $s$ :  $\Phi(s) = \sum_{sv \in A} \Phi(sv) = -\Phi(t)$

**MAXIMUM-FLOW:** Gegeben ein Flussnetzwerk, finde einen maximalen s-t-Fluss. Im Allgemeinen in  $\mathcal{O}(n^2)$ , aber für planare Graphen geht es besser.

**Notation:** Für  $X \subseteq V$  sei  $\Phi(X) := \sum_{\substack{uv \in A \\ u \in X, v \notin X}} \Phi(uv)$  der Netto-Ausfluss aus  $X$ .

Also ist  $\Phi(\{s\}) = \Phi(s)$  und  $\Phi(\{v\}) = 0$  für alle  $v \neq s, t$ .

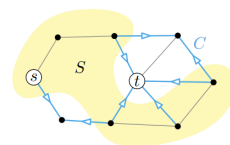
**Beobachtung:** Sind  $X, Y \subseteq V$  disjunkt, so gilt  $\Phi(X \cup Y) = \Phi(X) + \Phi(Y)$ . Zerlegt man  $X$  in einelementige Mengen, so folgt  $\Phi(X) = \sum_{u \in X} \Phi(u)$ .



**Definition:** Ein **s-t-Schnitt** ist ein Schnitt  $C \subseteq A$ , induziert von einer Knotenmenge  $S \subseteq V$  mit  $s \in S, t \notin S$ :

$$C := \{uv \in A \mid u \in S, v \notin S\}$$

Die **Kapazität** eines solchen Schnitts ist  $c(C) := \sum_{e \in C} c(e)$ .



**Max-Flow-Min-Cut-Lemma:** Für jeden s-t-Schnitt  $C$  und s-t-Fluss  $\Phi$  gilt

$$\Phi(s) \leq c(C)$$

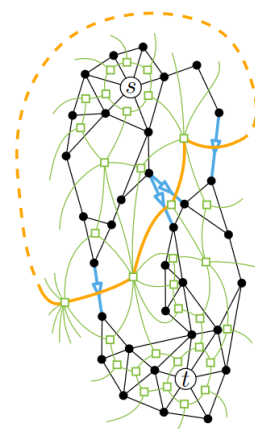
*Beweis:*  $\Phi(s) = \sum_{v \in S} \Phi(\{v\}) = \Phi(S) = \sum_{e \in C} \Phi(e) \leq \sum_{e \in C} c(e) = c(C)$

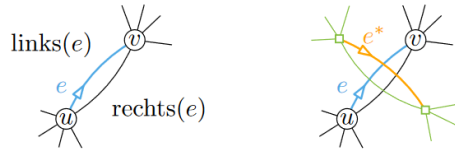
wobei die erste Gleichheit gilt, da  $\Phi(\{v\}) = 0$  für alle  $v \neq s$  ist.

**Max-Flow-Min-Cut-Theorem:**  $\max \Phi(s) = \min c(C)$  *ohne Beweis.*

Im Folgenden reicht es also nach einen s-t-Schnitt mit minimaler Kapazität zu suchen. Da  $c(e) > 0$  für jede Kante  $e \in A$ , reicht es inklusionsminimale s-t-Schnitte  $C \subseteq A$  zu betrachten, also solche, die keinen s-t-Schnitt enthalten. Schnitte in  $D$  entsprechen Kreise im Dualgraphen.

**Definition:** Der **gerichtete Dualgraph**  $D^* = (V^*, A^*)$  zu  $D$ : Für  $e = uv$  in  $D$  sei  $\text{links}(e)$  und  $\text{rechts}(e)$  die links bzw. rechts von  $e$  liegende Facette, wenn man über  $e$  von  $u$  nach  $v$  geht. In  $D^*$  sei die Dualkante  $e^*$  von  $\text{links}(e)$  nach  $\text{rechts}(e)$  orientiert.





**Definition:** Ein  **$s$ - $t$ -Kreis** ist ein einfacher gerichteter Kreis in  $D^*$  mit  $s$  auf der rechten und  $t$  auf der linken Seite.

**Lemma:** Sei  $C \subseteq A$  eine Kantenmenge und  $C^* \subseteq A^*$  die dazu duale Kantenmenge. Dann gilt:

$$C \text{ ist ein } s\text{-}t\text{-Schnitt} \iff C^* \text{ ist ein } s\text{-}t\text{-Kreis}$$

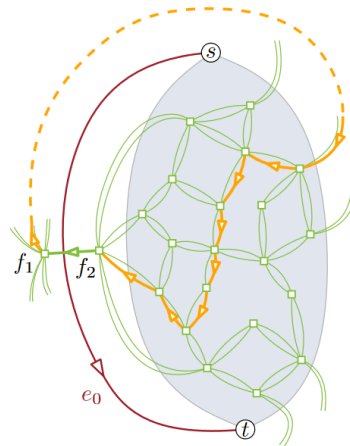
Wir setzen  $l(e^*) := c(e)$  für alle  $e \in A$  und interpretieren das als Länge der Dualkante  $e^*$ .

Es reicht also einen  $s$ - $t$ -Kreis mit minimaler Länge zu finden.

**Satz:** Für planare Graphen mit  $s$  und  $t$  an einer gemeinsamen Facette, kann ein Max-Flow in Linearzeit gefunden werden.

*Beweis:*

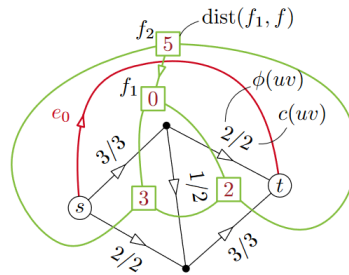
- O.B.d.A. liegen  $s$  und  $t$  an der äußeren Facette
- Jeder  $s$ - $t$ -Kreis muss die äußere Facette  $f_0$  als Dualknoten enthalten
- Füge neue Kante  $e_0 = st$  mit Kapazität  $c(e_0) = 0$  in äußere Facette ein
- Dies spaltet die äußere Facette  $f_0$  in  $f_1 = \text{rechts}(e_0)$  und  $f_2 = \text{links}(e_0)$
- Das Resultat ist  $D_+ = D + e_0$ . Berechne Dual  $D_+^*$  mit  $l(e^*) := c(e)$
- Berechne kürzesten Weg von  $f_1$  nach  $f_2$ :  $\text{dist}(f_1, f_2) = \min c(C) = \max \Phi(s)$



- Berechne daraus einen maximalen Fluss  $\Phi$ :

$$\Phi(e) := \text{dist}(f_1, \text{rechts}(e)) - \text{dist}(f_1, \text{links}(e))$$



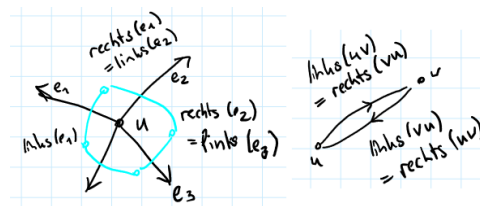


- Überprüfe Eigenschaften eines Flusses:

- Flusserhaltung: Für einen Knoten rechnet man im Kreis rechts – links, sodass sich rechts und links jeweils rauskürzen
- Zulässigkeit: Eine Möglichkeit von  $f_1$  nach  $\text{rechts}(uv)$  zu gehen, ist, zuerst von  $f_1$  nach  $\text{links}(uv)$  und dann über die Kante  $e^* = (uv)^*$  nach  $\text{rechts}(uv)$  zu gehen. Also ist

$$\begin{aligned} \text{dist}(f_1, \text{rechts}(uv)) &\leq \text{dist}(f_1, \text{links}(uv)) + c(uv) \\ \iff \text{dist}(f_1, \text{rechts}(uv)) - \text{dist}(f_1, \text{links}(uv)) &\leq c(uv). \end{aligned}$$

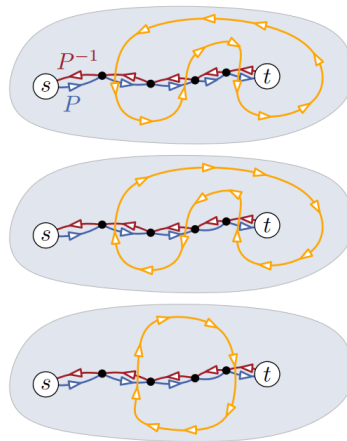
- Asymmetrie: Dies folgt aus  $\text{links}(uv) = \text{rechts}(vu)$  bzw.  $\text{links}(vu) = \text{rechts}(uv)$



- Damit ist der Satz bewiesen.

Betrachte nun den allgemeinen Fall, dass  $s$  und  $t$  an beliebigen Facetten liegen:

- Wähle einen gerichteten Pfad  $P$  von  $s$  nach  $t$
- Sei  $C^* \subseteq A^*$  ein gerichteter Kreis im Dualen und  $C \subseteq A$  der entsprechende Schnitt im Primalen
- Ist  $C^*$  ein  $s$ - $t$ -Kreis, dann gilt  $|C^* \cap P^*| = |C^* \cap P^{*-1}| + 1$ , d.h. der Kreis geht einmal mehr von links nach rechts als von rechts nach links
- Ist  $C^*$  ein  $t$ - $s$ -Kreis, dann gilt  $|C^* \cap P^*| = |C^* \cap P^{*-1}| - 1$
- Ansonsten gilt:  $|C^* \cap P^*| = |C^* \cap P^{*-1}|$



- Verringert man alle Kapazitäten der Kanten auf  $P$  um  $\alpha$  und erhöht alle Kapazitäten der Kanten auf  $P^{-1}$  um  $\alpha$ , so wird
  - jeder  $s$ - $t$ -Kreis um genau  $\alpha$  kürzer,
  - jeder  $t$ - $s$ -Kreis um genau  $\alpha$  länger,
  - jeder andere Kreis weder länger noch kürzer
- Anfangs waren alle Kreislängen positiv. Wählt man  $\alpha > 0$  groß genug, werden Kreise negative Länge bekommen, aber nur  $s$ - $t$ -Kreise!
- Ein Kreis, der bei kleinstem  $\alpha$  negative Länge bekommt, ist ein kürzester  $s$ - $t$ -Kreis

Finde jetzt also maximales  $\alpha$  so, dass noch keine negative Kreise entstehen.

**Satz:** Dieses maximale  $\alpha$  kann in  $\mathcal{O}(n \log n)$  bestimmt werden. *ohne Beweis.*

**Korollar:** MAX-FLOW in planaren Graphen kann in  $\mathcal{O}(n \log n)$  berechnet werden.

## 9 Menger-Problem

MENGER PROBLEM:

- **Gegeben:** Graph  $G = (V, E)$  und  $s, t \in V$
- **Gesucht:** Maximale Anzahl **kantendisjunkter**  $s$ - $t$ -Pfade in  $G$

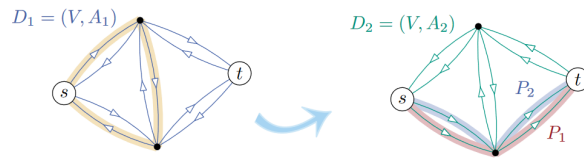
Kann im Allgemeinen mit MAX FLOW gelöst werden, indem jede Kante Kapazität 1 erhält. Wenn Fluss  $k$  ist, dann gibt es  $k$  kantendisjunkte Wege. Laufzeit für planare Graphen liegt dann in  $\mathcal{O}(n \log n)$ .

Wir werden aber eine Lösung in Linearzeit kennenlernen.

### Algorithmus für das MENDER PROBLEM:

Sei o.B.d.A.  $t$  an der äußeren Facette von  $G$ .

1. Konstruiere gerichteten Graphen  $D_1 = (V, A_1)$  mit  $uv \in A_1 \iff uv \in E$
2. Konstruiere  $D_2 = (V, A_2)$  aus  $D_1$ , sodass  $D_2$  keine gerichteten Kreise im Uhrzeigersinn (Rechtskreise) enthält
3. Finde maximale Anzahl gerichteter, kantendisjunkter  $s$ - $t$ -Pfade  $P_1, \dots, P_k$  in  $D_2$



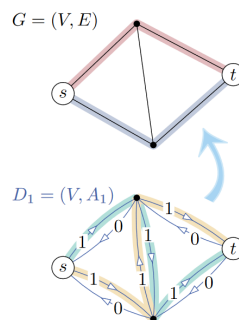
4. Berechne aus  $P_1, \dots, P_k$  ungerichtete  $s$ - $t$ -Pfade  $Q_1, \dots, Q_k$  in  $G$

Zeige **Korrektheit von Schritt 1**. Laufzeit und Durchführung sind klar.

**Lemma:**  $G$  hat  $k$  kantendisjunkte  $s$ - $t$ -Pfade  $\iff D_1$  hat  $k$  kantendisjunkte gerichtete  $s$ - $t$ -Pfade

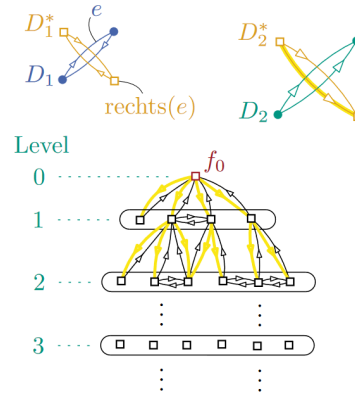
*Beweis:*

- Gegeben  $k$  kantendisjunkte gerichtete  $s$ - $t$ -Pfade in  $D_1$ , interpretiere diesen als Fluss  $\Phi$  von Wert  $k$  auf  $D_1$  mit  $\Phi(uv) \in \{0, 1\}$  für jede gerichtete Kante
- Definiere  $\Phi'$  als
  - $\Phi'(uv) = \Phi'(vu) = 0$ , wenn  $\Phi(uv) = \Phi(vu) = 1$
  - Ansonsten  $\Phi'(uv) = \Phi(uv)$
  - Es wurde der Fluss also dort auf 0 gesetzt, wo hin und Rückkante benutzt werden
- $\Phi'$  hat Wert  $k$  und es gibt  $k$  kantendisjunkte Wege in  $G$



Nun wird die **Konstruktion in Schritt 2** beschrieben:

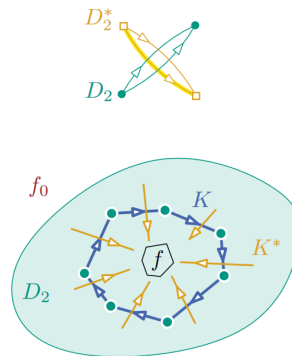
- Betrachte Breitensuche von der äußeren Facette  $f_0$  im gerichteten Dualgraphen  $D_1^*$  von  $D_1$
- Auf Level  $i$  sind alle Facetten mit Abstand  $i$  zu  $f_0$ . Drehe für jedes  $i$  die Richtung aller gerichteten Kanten von Level  $i$  nach Level  $i + 1$  um.
- Ergebnis:  $D_2^*$  mit primalen Graph  $D_2$
- Da Breitensuche in Linearzeit möglich ist, ist auch dieses Verfahren in Linearzeit möglich.



**Lemma:**  $D_2$  enthält keine Rechtskreise

*Beweis:*

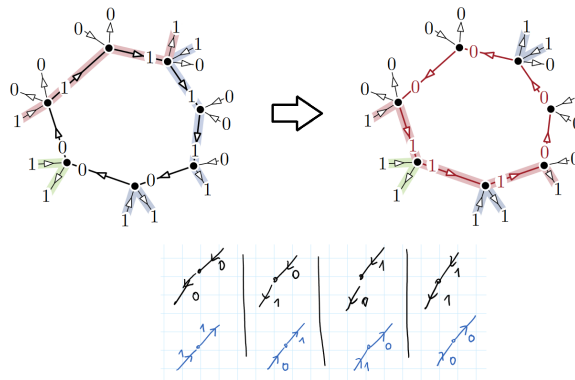
- Angenommen  $K$  wäre ein Rechtskreis in  $D_2$
- Dann ist  $K^*$  ein gerichteter Schnitt in  $D_2^*$
- Sei  $f$  eine Facette innerhalb von  $K$
- Ein kürzester Weg von  $f_0$  zu  $f$  muss den Schnitt passieren, aber diese Kanten wurden in  $D_2^*$  umgedreht.
- Widerspruch dazu, dass  $K$  ein Rechtskreis ist



**Lemma:**  $D_2$  enthält  $k$  kantendisjunkte  $s$ - $t$ -Wege  $\iff D_1$  enthält  $k$  kantendisjunkte  $s$ - $t$ -Wege.

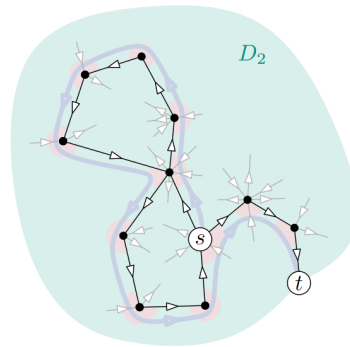
*Beweis:*

- In Schritt 2 wurden im Dualen immer Kantenmengen zwischen zwei Levels umgedreht
- Das sind im Dualen  $D_1^*$  knoteninduzierte  $s$ - $t$ -Schnitte
- Im Primalen  $D_1$  sind das für ein festes Level eine disjunkte Vereinigung von Kreisen
- Es reicht also zu zeigen, dass  $k$  kantendisjunkte Wege, erhalten bleiben wenn ein gerichteter Kreis umgedreht wird
- Sei also ein gerichteter Kreis umgedreht
- Interpretiere wieder  $k$  Wege als Fluss  $\Phi$  mit Wert  $k$
- Drehe Orientierung der Kanten und gleichzeitig den Flusswert der Kante
- Dies erhält die Flusseigenschaften und wir erhalten wieder ein Fluss  $\Phi'$  mit Wert  $k \implies$  Wir erhalten auch wieder  $k$  Wege.



Nun wird das **Vorgehen in Schritt 3** beschrieben:

- Jeder Knoten in  $D_2$  hat gleich viele eingehende wie ausgehende Kanten, da das in  $D_1$  der Fall war und in Schritt 2 nur Kreise umgedreht wurden, was die Eigenschaft weiterhin erhält
- Starte Tiefensuche bei  $s$  mit ausgehender Kante
- Nimm immer die rechteste ausgehende Kante bzgl. der gerade genommenen eingehenden Kante und gehe über jede Kante nur einmal
- Wenn  $t$  erreicht wird, ist ein  $s$ - $t$ -Pfad gefunden und man fängt wieder bei  $s$  mit neuer Kante an
- Aufhören wenn keine ausgehende Kante verfügbar ist (das passiert bei  $s$ )
- Wenn Knoten  $t$  insgesamt  $k$ -mal erreicht wurde, wurden  $k$  kantendisjunkte  $s$ - $t$ -Pfade gefunden.

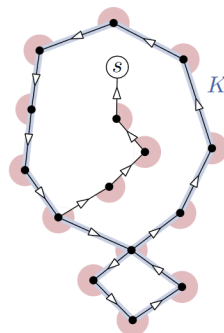


### Laufzeit von Schritt 3:

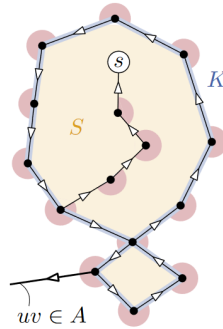
- Rechteste Kante kann mit Union-Find-Datenstruktur gefunden werden
- Laufzeit von Union-Find ist allgemein  $\mathcal{O}(n \cdot \alpha(n))$ , also nur fast linear
- Hier Spezialfall, in dem  $n$  Operationen vom Typ Union und Find in Linearzeit ausgeführt werden können  $\rightarrow$  Insgesamt Linearzeit

### Korrektheit von Schritt 3:

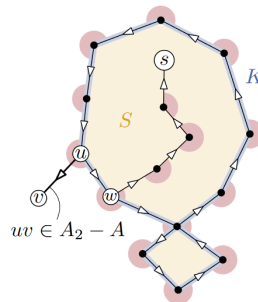
- **Ziel:** Finde einen  $s$ - $t$ -Schnitt dessen Kapazität der Anzahl  $k$  der gefundenen  $s$ - $t$ -Pfade entspricht. Dann kann es nicht mehr als  $k$  kantendisjunkte Pfade von  $s$  nach  $t$  geben, da diese den Schnitt passieren müssen
- Betrachte Graphen  $D = (V, A)$  mit allen Kanten, die im Algorithmus genommen wurden
- Starte Tiefensuche in  $D$  bei  $s$  aber gehe Kanten rückwärts und nehme die linkeste mögliche Fortsetzung
- Baue damit rückwärts einen gerichteten Weg  $W$  zu  $s$  auf und stoppe, wenn wir eine Kante aus  $W$  als Fortsetzung nehmen würden oder wenn wir wieder auf  $s$  treffen
- Wir treffen nicht auf  $t$ , da  $t$  in  $D$  keine ausgehenden Kanten hat. Außerdem bleiben wir nicht hängen, da alle anderen Knoten so viele eingehende wie ausgehende Kanten haben



- $W$  umschließt ein Gebiet, dessen Rand  $K$  eine Menge gerichteter Kreise ist
- Entlang von  $K$  sind auf der rechten Seite Winkel ohne eingehenden Kanten
- Wegen Schritt 2 hat  $K$  keinen Rechtskreis  $\rightarrow$  Winkel sind außen an den Kreisen
- Also muss  $s$  entweder auf  $K$  oder  $\text{int}(K)$  liegen, da (Rückwärts-)Wege von  $s$  zu den Knoten in  $K$  existieren und diese nicht in die Winkel laufen können
- Knoten  $t$  liegt nicht auf  $K$ , da er ohne ausgehende Kante nicht von der Rückwärts-Tiefensuche gefunden werden kann.  $t \in \text{ext}(K)$ , da  $t$  nach Annahme auf der äußeren Facette von  $D$  liegt
- Definiere nun  $S \subseteq V$  als Menge aller Knoten auf  $K$  und in  $\text{int}(K)$ .  $S$  induziert unseren  $s$ - $t$ -Schnitt  $C \subseteq A_2$
- **Zu zeigen:** Kapazität  $c(C) \leq k$
- **Fall 1:** Sei  $uv \in A$  mit  $u \in V(K)$  und  $v \in V - S$  eine Kante, die vom Algo benutzt wurde



- Dann ist die Tiefensuche nach  $uv$  nicht wieder nach  $S$  gekommen, endete also in  $t$ , denn Winkel haben keine eingehenden Kanten aus  $A$ !
- Es gibt also höchstens  $k$  solcher Kanten!
- **Fall 2:** Sei  $uv \in A_2 - A$  Kante mit  $u \in V(K)$  und  $v \in V - S$ , die vom Algo nicht benutzt wurde:



- Betrachte  $uw \in E(K)$ , die benutzt wurde und deren benutzte Vorgängerkante  $xu$ . Dann ist  $x \in S$ , da keine benutzten eingehenden Kanten in Winkeln liegen

- Dann ist aber  $uv$  eine bessere (weiter rechts) Fortsetzung für  $xu$  als  $uw \rightarrow$  Widerspruch!
- Also gibt es keine solche Kanten!
- In beiden Fällen hat der Schnitt also Kapazität höchstens  $k$  und die Optimalität des Algorithmus ist bewiesen.

Damit wurde bewiesen, dass das **MENGER PROBLEM** in Linearzeit gelöst werden kann. Für Schritt 4 muss lediglich das Vorgehen aus Schritt 1 und 2 invertiert werden.

## 10 Problem von Okamura und Seymour

### KANTENDISJUNKTES WEGPACKUNGSPROBLEM

- **Gegeben:** Graph  $G = (V, E)$  und Paare von Knoten  $\{s_1, t_1\}, \dots, \{s_k, t_k\}, s_i, t_i \in V$
- **Gesucht:** Paarweise kantendisjunkter  $s_i$ - $t_i$ -Wege  $p_i$  in  $G, 1 \leq i \leq k$

$s_i, t_i$  werden **Terminale** genannt, die Mengen  $\{s_i, t_i\}$  heißen **Netze**.

Problem ist  $\mathcal{NP}$ -vollständig, auch falls  $G$  planar ist. Wir werden das Problem deshalb später einschränken.

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

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

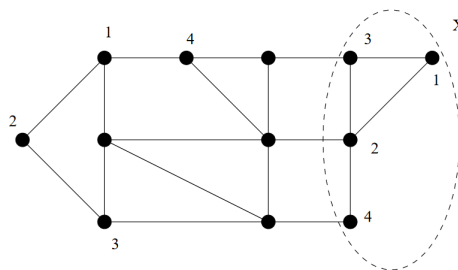
die **Kapazität** von  $X$ . Zu  $G$  sei  $D = \{\{s_i, t_i\} \mid s_i, t_i \in V, 1 \leq i \leq k\}$  gegeben. Dann heißt

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

die **Dichte** von  $X$ . Weiterhin bezeichnet

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

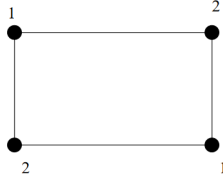
die **freie Kapazität** von  $X$ .  $X$  heißt **saturiert**, falls  $\text{fcap}(X) = 0$  und **übersaturiert**, falls  $\text{fcap}(X) < 0$ .





**Definition:**

- **Kapazitätsbedingung:**  $\text{fcap}(X) \geq 0$  für alle  $X \subseteq V$ . Jedes lösbares KANTENDISJUNKTE WEGPACKUNGSPROBLEM erfüllt die Bedingung, aber nicht jedes Problem, das die Bedingung erfüllt, ist lösbar.



$\text{cap}(X) \geq 2$  für alle  $X \neq \emptyset$  und  $\text{dens}(X) \leq 2$ , aber nicht lösbar

- **Geradheitsbedingung:**  $\text{fcap}(X)$  ist gerade für alle  $X \subseteq V$

**Lemma:** Sei  $G = (V, E)$  und  $D = \{\{s_i, t_i\} \mid s_i, t_i \in V, 1 \leq i \leq k\}$ . Es gilt  $\text{fcap}(X)$  gerade für alle  $X \subseteq V \iff \text{fcap}(v) := \text{fcap}(\{v\})$  gerade für alle  $v \in V$

*Beweis:*

- „ $\Rightarrow$ “: Trivial.
- „ $\Leftarrow$ “: Sei  $\text{fcap}(v)$  gerade für alle  $v \in V$ . Für  $X \subseteq V$  ist
  - $\text{cap}(X) = \sum_{v \in X} \text{cap}(v) - 2 \cdot |\{uv \in E \mid u, v \in X\}|$
  - $\text{dens}(X) = \sum_{v \in X} \text{dens}(v) - 2 \cdot |\{\{s_i, t_i\} \in D \mid s_i, t_i \in X\}|$

Dann ist:

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

Also ist  $\text{fcap}(X)$  gerade, falls alle  $\text{fcap}(v)$  gerade sind.

**OKAMURA & SEYMOUR-PROBLEM:**

- **Gegeben:** Graph  $G = (V, E)$  und Paare von Knoten  $\{s_1, t_1\}, \dots, \{s_k, t_k\}, s_i, t_i \in V$ , wobei alle  $s_i, t_i$  auf dem Rand der selben Facette, o.B.d.A der äußeren, liegen. Außerdem sei die Geradheitsbedingung erfüllt.
- **Gesucht:** Paarweise kantendisjunkter  $s_i$ - $t_i$ -Wege  $p_i$  in  $G, 1 \leq i \leq k$

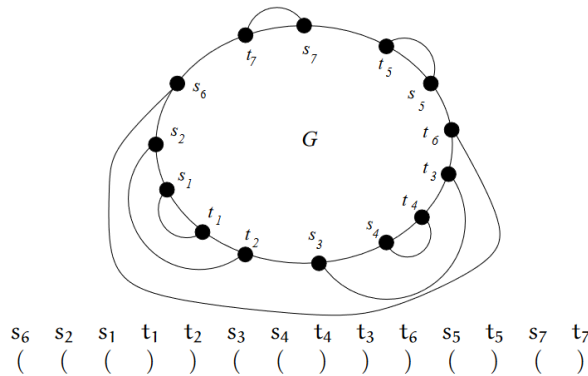
**Satz:** Gegeben sei ein planar eingebetteter Graph  $G = (V, E)$  und  $D = \{\{s_1, t_1\}, \dots, \{s_k, t_k\}\}$ , wobei  $s_1, \dots, s_k, t_1, \dots, t_k$  auf dem Rand der äußeren Facette von  $G$  liegen. Die Kapazitätsbedingung und die Geradheitsbedingung seien erfüllt. Dann existieren paarweise kantendisjunkte  $s_i$ - $t_i$ -Wege in  $G$ . *ohne Beweis.*

**Definition:**  $G = (V, E)$  mit  $D = \{\{s_1, t_1\}, \dots, \{s_k, t_k\}\}$  hat **Klammerstruktur**, falls  $G + D$  so planar eingebettet werden kann, dass die Kanten aus  $D$  kreuzungsfrei in die äußere Facette der entsprechenden Einbettung von  $G$  eingebettet sind

#### Linearzeitalgorithmus für das OKAMURA & SEYMOUR-PROBLEM:

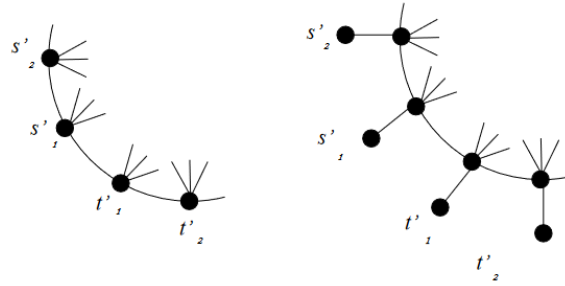
**Schritt 1:** Konstruiere aus  $G = (V, E)$  mit  $D = \{\{s_1, t_1\}, \dots, \{s_k, t_k\}\}$  ein Problem bestehend aus  $G = (V, E)$  mit  $D' = \{\{s'_1, t'_1\}, \dots, \{s'_k, t'_k\}\}$  so, dass  $\{s_1, \dots, s_k, t_1, \dots, t_k\} = \{s'_1, \dots, s'_k, t'_1, \dots, t'_k\}$  und die  $\{s'_1, t'_1\}, \dots, \{s'_k, t'_k\}$  Klammerstruktur haben

- Klammerstruktur leicht in  $\mathcal{O}(n)$  konstruiert werden
- Wähle beliebiges Terminal als Startterminal  $s$  und gehe im Gegenuhrzeigersinn um die äußere Facette
- Dem jeweils ersten Terminal eines  $\{s_i, t_i\}$  ordne eine öffnende Klammer zu und dem jeweils zweiten eine schließende



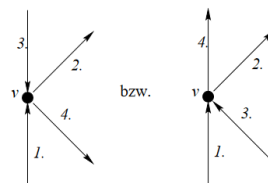
**Schritt 2:** Berechne in  $\mathcal{O}(n)$  kantendisjunkte  $s'_i$ - $t'_i$ -Wege  $q_1, \dots, q_k$  mittels Right-First-Tiefensuche und eine Orientierung der entsprechenden Wege von  $s'_i$  nach  $t'_i$

- Die  $s'_i, t'_i$  seien, beginnend beim Startterminal  $s$  im Gegenuhrzeigersinn so angeordnet, dass jeweils  $s'_i$  vor  $t'_i$  und  $t'_i$  vor  $t'_{i+1}$
- Füge zusätzlich aus technischen Gründen (s. Algorithmus S. 7) Dummy-Kanten hinzu



- Wegen der Geradheitsbedingung endet jeder Durchlauf der Right-First-Tiefensuche bei einem Terminalknoten

**Beobachtung:** Wegen der Right-First-Auswahlregel können an einem Knoten  $v$  folgende Reihenfolgen von Kanten nicht vorkommen:



Daher gilt für die Wege  $q_i$ :

- Keine zwei Wege  $q_i$  und  $q_j$  kreuzen sich
- Kein Weg  $q_i$  kreuzt sich selbst

Sei  $\vec{G} = (\vec{V}, \vec{E})$ ,  $\vec{V} \subseteq V$  der Graph, der durch die von  $q_1, \dots, q_k$  belegten Kanten zusammen mit der Orientierung induziert wird.

**Korollar:**  $\vec{G} = (\vec{V}, \vec{E})$  enthält keinen Rechtskreis.

*Beweis:*

- Wenn  $\vec{C}$  Kanten eines Rechtskreises in  $\vec{G}$  sind, so gehören nicht alle Kanten aus  $\vec{C}$  zum selben Weg  $q_i$ , da sich Wege nicht kreuzen
- Seien  $q_i, q_j$ , Wege, die Kanten aus  $\vec{C}$  besetzen. Da  $q_i$  und  $q_j$  sich nicht kreuzen müssen die Terminale  $s'_i, t'_i$  und  $s'_j, t'_j$  in der Reihenfolge  $s'_i, t'_j, s'_j, t'_i$  im Gegenuhrzeigersinn auf der äußeren Facette liegen.
- Widerspruch zur Anordnung der Terminale.

**Korrektheit von Schritt 2:** Die Right-First-Tiefensuche berechnet für einen Graphen mit Struktur nach Schritt 1 kantendisjunkte  $s'_i-t'_i$ -Wege  $q_1, \dots, q_k$ .

- Betrachte eine beliebige kreuzungsfreie Lösung  $q'_1, \dots, q'_k$ . Eine solche Lösung existiert immer

- Wenn  $q'_1, \dots, q'_k$  kreuzungsfrei ist, so gilt für jedes  $q'_i$ , dass es vollständig zur linken Seite aller  $q'_1, \dots, q'_{i-1}$  gehört
- Für jedes  $q_i$  gilt, da es mittels Right-First-Auswahlregel und entsprechend der Klammerstruktur von  $D'$  bestimmt wurde, dass die linke Seite von  $q'_i$  ganz enthalten ist in der linken Seite von  $q_i$
- Daraus folgt, dass  $q_i$  die Terminale  $s'_i$  und  $t'_i$  verbindet

**Laufzeit von Schritt 2:** Amortisiert in  $\mathcal{O}(n)$  realisierbar, da die rechteste freie Kante bzgl. der führenden Kante immer die nächste Kante nach der führenden Kante in der Adjazenzliste ist, und damit in konstanter Zeit gefunden werden kann

**Schritt 3:** Berechne in  $\mathcal{O}(n)$  kantendisjunkte  $s_i$ - $t_i$ -Wege  $p_1, \dots, p_k$  in  $\vec{G}$  mittels gerichteter Right-First-Tiefensuche, und zwar der Reihe nach entsprechend der Reihenfolge der  $t'_i$ .

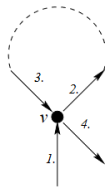
**Korrektheit von Schritt 3:**

- Konstruiere Weg  $p$  mit der Eigenschaft:
  1. Falls  $p_i$  korrekt  $s_i$  mit  $t_i$  verbindet, so induz.  $p$  einen saturierten Schnitt in  $G$
  2. Falls  $p_i$  Terminal  $s_i$  nicht mit  $t_i$  verbindet, aber jedes  $p_j, 1 \leq j < i$  korrekt  $s_j$  mit  $t_j$  verbindet, so induziert  $p$  einen übersaturierten Schnitt in  $G$
- Mit 2. ist die Kapazitätsbedingung und damit eine notwendige Bedingung für Lösbarkeit, verletzt, also existiert in dem Fall sowieso keine Lösung
- Berechne  $p$ , der einen Schnitt mit den gewünschten Eigenschaften induziert, mit einer Left-First-Tiefensuche rückwärts vom Endknoten  $t$  des  $i$ -ten Weges  $p_i$ . Das ist offensichtlich in  $\mathcal{O}(n)$  möglich.

**Lemma:** Weg  $p$ , der wie oben berechnet wird, enthält keine Kreuzung. Insbesondere sind seine linke und rechte Seite wohldefiniert.

*Beweis:*

- Angenommen  $p$  würde sich selbst in einem Knoten  $v$  kreuzen
- Da  $\vec{G}$  keinen Rechtskreis enthält, gibt es in  $v$  einen einfachen Linkskreis, wobei die entsprechenden Kanten von  $p$ , die zu  $v$  inzident sind, folgende Konstellation bilden:



- Widerspruch zur Right-First-Tiefensuche aus Schritt 2.

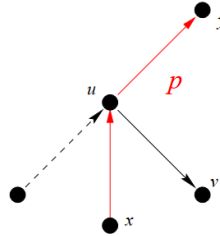
**Lemma:** Sei  $A$  die Menge der Kanten  $uv$  aus  $G$  mit  $u$  auf  $p$  und  $v$  rechts von  $p$ . Jede Kante  $uv \in A$  mit  $u$  auf  $p$  gehört zu  $\vec{G}$ , und zwar in der Orientierung  $(u, v)$  genau dann, wenn sie von einem der  $p_j, 1 \leq j < i$ , besetzt ist.

*Beweis:*

- „ $\Leftarrow$ “: Sei  $uv \in A$  mit  $u$  auf  $p$ . Falls  $uv$  von einem der  $p_j, 1 \leq j < i$  besetzt ist, so ist deren Orientierung  $(u, v)$  in  $\vec{G}$ , nach Konstruktion von  $p$
- „ $\Rightarrow$ “:

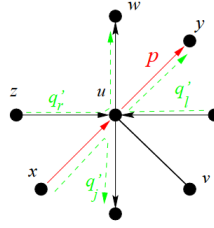
**Fall 1:** Angenommen  $uv \in A$  mit  $u$  auf  $p$  habe Orientierung  $(u, v)$  in  $\vec{G}$  und sei nicht durch eines der  $p_i$  besetzt

- Betrachte die zu  $p$  gehörenden Kanten  $(x, u), (u, y)$  inzident zu  $p$ , für die  $uv$  rechts von  $(x, u)$  und  $(u, y)$  liegt
- Kante, die bei der Berechnung der  $p_1, \dots, p_i$  unmittelbar vor  $(u, y)$  gewählt wurde, ist dann  $(x, u)$  oder liegt links von  $(x, u)$  und  $(u, y)$  wegen der Konstruktion von  $p$
- Dann ist aber die Wahl von  $(u, y)$  ein Widerspruch zur Right-First-Auswahlregel



**Fall 2:** Kante  $uv \in A$  mit  $u$  auf  $p$  gehört nicht zu  $\vec{G}$

- Wegen der Right-First Auswahlregel können die Kanten  $(x, u), (u, y)$  von  $p$ , für die  $uv$  rechts liegt, nicht beide zu dem selben Weg  $q'_i$  aus  $\vec{G}$  gehören
- Gehöre  $(x, u)$  zu  $q'_j$  und  $(u, y)$  zu  $q'_l$
- Dann liegt Vorgängerkante von  $(u, y)$  auf  $q'_l$  rechts von  $uv$  und  $(u, y)$  und kann daher von keinem der  $p_1, \dots, p_i$  besetzt sein, da sonst  $p$  wegen Left-First-Tiefensuche über  $q'_l$  verlaufen würde
- Dann muss es eine weitere Kante  $(z, u)$  in  $\vec{G}$  geben, die von einem der  $p_1, \dots, p_i$  besetzt ist und links von  $(x, u)$  und  $(u, y)$  liegt, und eine weitere Kante  $(u, w)$ , die Nachfolgerkante von  $(z, u)$  auf dem entsprechenden Weg  $q'_r$  in  $\vec{G}$  ist, und auch links von  $(x, u)$  und  $(u, y)$  liegt
- Dann liegt aber die Kante  $uv$  rechts von  $q'_r$  im Widerspruch zur Right-First-Auswahlregel



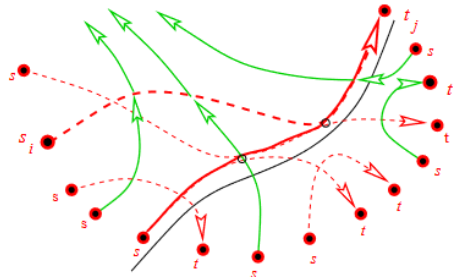
**Lemma:** Sei  $X \subseteq V$  Menge der Knoten rechts von  $p$ . Falls  $p_i$  ein  $s_i$ - $t_i$ -Weg ist, ist  $X$  saturiert, ansonsten ist  $X$  übersaturiert

*Beweis:* Alle Kanten  $\{u, v\}$  mit  $u$  auf  $p$  und  $v \in X$  gehören entweder zu einem Weg  $p_j$ ,  $1 \leq j < i$ , mit  $s_j \in V \setminus X$  und  $t_j \in X$ , oder zu einem Weg  $q'_j$ , mit  $s'_j \in X$  und  $t'_j \in V \setminus X$ . Wenn  $p_i$  ein  $s_i$ - $t_i$ -Weg ist, dann ist damit

$$\begin{aligned} \text{cap}(X) &= |\{\{s_j, t_j\} : s_j \in V \setminus X, t_j \in X, 1 \leq j \leq i\}| \\ &\quad + |\{\{s'_j, t'_j\} : s'_j \in X \text{ und } t'_j \in V \setminus X, \{s'_j, t'_j\} \notin \{\{s_1, t_1\}, \dots, \{s_i, t_i\}\}\}| \\ &= \text{dens}(X). \end{aligned}$$

Wenn  $p_i$  kein  $s_i$ - $t_i$ -Weg ist, so verbindet  $p_i$  das Terminal  $s_i$  mit einem Terminal  $t \in \{t_j; i < j \leq k\}$ . Da  $s_i \in V \setminus X$  und  $t_i \in X$  ist, dann gilt

$$\text{cap}(X) \leq \text{dens}(X) - 1$$



**Laufzeit von Schritt 3:** Mit Union-Find kann die Right-First-Tiefensuche zu Schritt 3 in  $\mathcal{O}(n)$  realisiert werden

*Visualisierung des Algorithmus ab S.13 im Skript*

## 11 Planarität testen

**Problem:** Gegeben ein Graph  $G$  als Adjazenzliste, entscheide ob  $G$  planar ist

**Möglichkeit:**

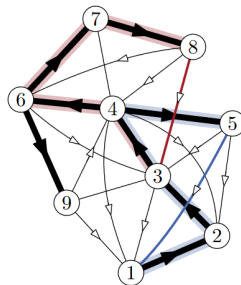
- Teste auf  $K_5$ - oder  $K_{3,3}$ -Unterteilung
- Es existiert ein theoretisch effizienter, aber nicht praktikabler Algorithmus dafür, da die Laufzeit sehr groß werden kann. Im Folgenden: **LR-Planarität**

### Annahmen:

- $G$  hat keine Mehrfachkante
- $G$  hat keine Schlingen
- $G$  hat keine Brücken, also liegt jede Kante auf einem Kreis
- $G$  ist zusammenhängend

### Vorbereitendes Vorgehen:

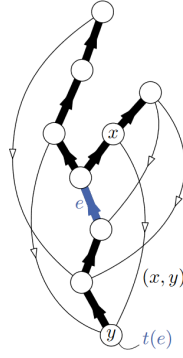
- Tiefensuche von beliebigem Startknoten
- Nummeriere Knoten in Explorierungsreihenfolge
- Orientiere Baumkanten in Explorierungsreihenfolge und Nichtbaumkanten vom Knoten mit größerer DFS-Zahl zum Knoten mit kleinerer DFS-Zahl



### Erste Beobachtungen:

- Nichtbaumkanten verbinden immer zwei Knoten auf einem Wurzel-Blatt-Pfad.
- Jede Nichtbaumkante  $e$  schließt einen eindeutigen Kreis mit den Baumkanten (**Fundamentalkreis** zu  $e$ ).
- Jede planare Zeichnung von  $G$  liefert auch eine planare Zeichnung des DFS-Baums. Betrachte Zeichnung mit der Wurzel an der äußeren Facette.
- Jede Nichtbaumkante  $e$  führt entweder links (**L-Kante**) oder rechts (**R-Kante**) am Baum zurück.
- Um herauszufinden, ob  $G$  planar ist, müssen wir entweder eine Einteilung der Nichtbaumkanten in  $L$ - und  $R$ -Kanten finden bei der keine Überschneidungen entstehen (**LR-Zerlegung**) oder zeigen, dass eine solche Einteilung nicht existiert.

**Definition:** Sei  $e$  eine Kante. Eine Nichtbaumkante  $(x, y)$  heißt **Rückkante** von  $e$ , wenn  $e$  auf dem Fundamentalkreis von  $(x, y)$  liegt. Dann nennt man  $y$  einen **Rückkehrpunkt** von  $e$ . Ist  $e$  eine Nichtbaumkante, so ist  $e$  seine einzige und eigene Rückkante mit eindeutigem Rückkehrpunkt.



**Definition:** Sei  $e$  eine Kante. Der tiefste Rückkehrpunkt von  $e$

$$t(e) := \text{kleinste DFS-Zahl eines Rückkehrpunktes}$$

nennt man den **Tiefpunkt** von  $e$ .

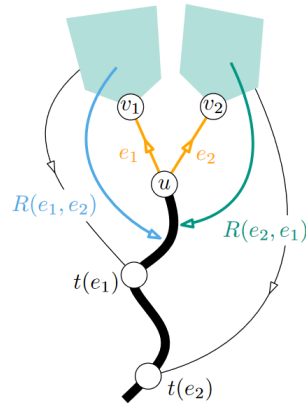
**Definition:** Eine **Gabel** sind zwei Kanten mit dem gleichen Startpunkt, also  $e_1 = (u, v_1)$  und  $e_2 = (u, v_2)$ .

**Definition:** Für eine Gabel  $e_1 = uv_1$ ,  $e_2 = uv_2$  definiere die Mengen

- $R(e_1, e_2) := \{e \text{ Rückkante von } e_1 \text{ mit } t(e_2) < t(e) < u\}$
- $R(e_2, e_1) := \{e \text{ Rückkante von } e_2 \text{ mit } t(e_1) < t(e) < u\}$

Zwei Kanten  $f_1, f_2$  haben einen **Konflikt** bezüglich  $e_1, e_2$ , wenn

- $f_1, f_2 \in R(e_1, e_2)$  oder  $f_1, f_2 \in R(e_2, e_1)$  (**Gleichheitskonflikt**)
- $f_1 \in R(e_1, e_2)$  und  $f_2 \in R(e_2, e_1)$  oder umgekehrt (**Ungleichheitskonflikt**)



**Definition:** Eine **LR-Zerlegung** ist eine Partition der Nichtbaumkanten in  $L$  und  $R$ , sodass je zwei Kanten  $f_1, f_2$

- mit Gleichheitskonflikt in der gleichen Menge sind.
- mit Ungleichheitskonflikt in unterschiedlichen Mengen sind.



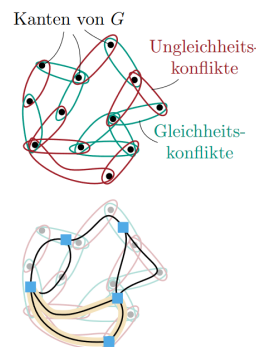
**Satz:** Folgende Aussagen sind äquivalent:

- $G$  ist planar.
- Es gibt eine LR-Zerlegung bezüglich einer Tiefensuche.
- Es gibt zu jeder Tiefensuche eine LR-Zerlegung

*Der Beweis des Satzes wird später skizziert.*

### Planaritätstest basierend auf LR-Zerlegung

1. Tiefensuche auf Graph in  $\mathcal{O}(n) = \mathcal{O}(|V(G)|)$
2. Bestimme naiv alle Kantenkonflikte in  $\mathcal{O}(n^2)$ , indem man sich z.B. alle Kantenpaare anschaut
3. Baue Hilfsgraph  $H$  mit
  - Knoten in  $H$  sind Kanten von  $G$ .
  - Kanten in  $H$  sind Ungleichheitskonflikte in  $G$ .
  - Knoten in  $H$  werden verschmolzen, wenn sie einen Gleichheitskonflikt haben.



4. LR-Zerlegung von  $G$  existiert  $\iff H$  ist bipartit. Teste also den Graphen auf Bipartitheit. Das geht in  $\mathcal{O}(n^2)$

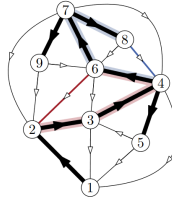
$\implies$  Damit haben wir einen Planaritätstest in  $\mathcal{O}(n^2)$ . Eine Beschleunigung auf  $\mathcal{O}(n)$  ist aber möglich.

*Korrektheitsbeweis vom obigen Satz:*

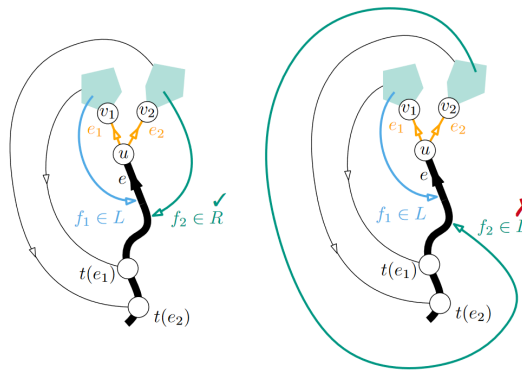
Zeige zunächst  $G$  planar  $\implies G$  hat LR-Zerlegung:

- Einbettung von  $G$  mit Knoten 1 an der äußeren Facette induziert eine Einbettung des gerichteten DFS-Baumes.
- Nichtbaumkante  $e$  ist in  $L$ , wenn der Fundamentalkreis zu  $e$  gegen den Uhrzeigersinn orientiert ist

- Nichtbaumkante  $e$  ist in  $R$ , wenn der Fundamentalkreis zu  $e$  im Uhrzeigersinn orientiert ist



- Zeige, dass diese Einteilung in  $L$  und  $R$  alle Konflikte erfüllt
- Betrachte eine Gabel  $e_1, e_2$  an Gabelknoten  $u$  mit eingehender Baumkante  $e$
- Seien o.B.d.A.  $e, e_1, e_2$  in dieser cw-Reihenfolge um  $u$  und  $t(e_1) \geq t(e_2)$
- Betrachte nun  $f_1 \in R(e_1, e_2)$  und  $f_2 \in R(e_2, e_1)$ :

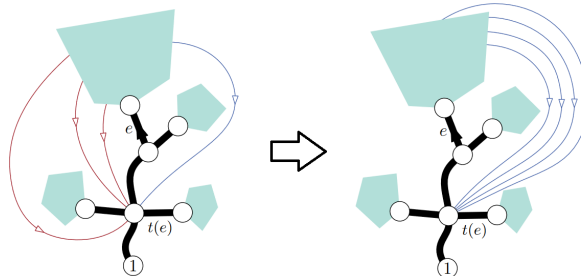


Zeige nun  $G$  hat LR-Zerlegung  $\implies G$  planar:

- Betrachte eine LR-Zerlegung, konstruiere eine planare Einbettung von  $G$  mit Knoten 1 an der äußeren Facette, d.h. gib für jeden Knoten die zyklische Reihenfolge der inzidenten Kanten an

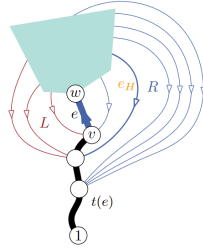
**Schritt 1:** LR-Zerlegung bündeln

- Eine LR-Zerlegung heißt **gebündelt** wenn zu jeder Kante  $e$  alle Rückkanten von  $e$  mit Rückkehrpunkt  $t(e)$  in der gleichen Menge ( $L$  oder  $R$ ) liegen
- **Lemma:** Jede LR-Zerlegung kann gebündelt werden. *ohne Beweis.*



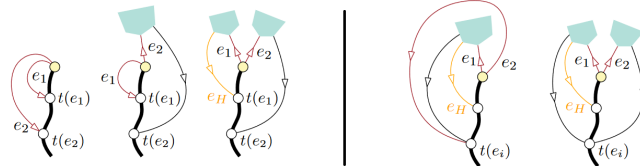
### Schritt 2: LR-Zerlegung auf Baumkanten erweitern

- Sei  $e = vw$  eine Baumkante. Betrachte Rückkante  $e_H$  von  $e$  deren Rückkehrpunkt die höchste DFS-Zahl hat, aber nicht  $v$  ist
- Füge  $e$  der Menge, in der  $e_H$  ist, hinzu

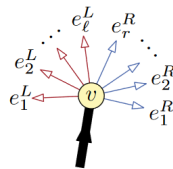


### Schritt 3: Reihenfolge der ausgehenden Kanten festlegen

- Für Knoten  $v$  und ausgehende Kanten  $e_1, e_2 \in L$  definiere  $e_1 \prec e_2$  wenn
  - entweder  $t(e_2) < t(e_1)$
  - oder  $t(e_2) = t(e_1)$  und  $e_1$  hat noch weiteren Rückkehrpunkt

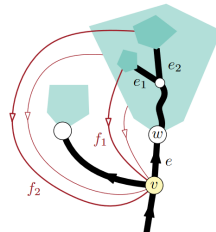


- Im Uhrzeigersinn ist die Ordnung der  $L$ -Kanten um  $v$  dann  $e_1^L \prec e_2^L \prec \dots \prec e_l^L$
- $L$ -Kanten sind also von innen nach außen sortiert
- Für  $R$ -Kanten ist die Sortierung genau andersrum



### Schritt 4: Reihenfolge eingehender Kanten an $v$ zur gleichen Baumkante $e = (v, w)$

- Betrachte eingehende Kanten in  $L$ :  $L(e) = \{\text{Rückkante von } e \text{ eingehend an } v\} \cap L$

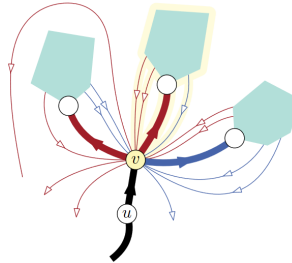


- Je zwei  $f_1, f_2 \in L(e)$  kommen von einer Gabel  $e_1, e_2$ , wobei  $f_i$  Rückkante von  $e_i$  ist
- Aus Schritt 3 haben die Kanten eine Ordnung bezüglich  $\prec$  am Gabelknoten.  
O.B.d.A. sei nun  $e_1 \prec e_2$
- Setze dann  $f_2 \prec f_1$  an  $v$ , also genau umgekehrt
- Verfahre genauso für eingehende Kanten in  $R$

**Schritt 5:** Inzidente Kanten an  $v$  zyklisch sortieren

- $e_1^L, \dots, e_t^L, e_r^R, \dots, e_1^R$  sortierte ausgehende Kanten
- Betrachte  $L(e_i^L), R(e_i^L)$  bzw.  $L(e_j^R), R(e_j^R)$  mit in Schritt 4 festgelegten Reihenfolge
- Für Nichtbaumkanten  $e_i^L, e_j^R$  sind diese Mengen leer
- Definiere nun die Reihenfolge im Uhrzeigersinn an  $v$  als:

$$(u, v), L(e_1^L), e_1^L, R(e_1^L), \dots, L(e_1^R), e_1^R, R(e_1^R)$$



Dieser ganze Prozess kann in  $\mathcal{O}(n)$  durchgeführt werden.