

# **Mitschrieb Planare Graphen SS 2015**

Robin

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>4</b>
1.1	$K_5$ ist nicht planar . . . . .	5
1.2	$K_{3,3}$ ist nicht planar . . . . .	6
1.3	Themen . . . . .	6
1.4	Vierfarbensatz . . . . .	6
1.5	Satz (1977 Appel & Haken) . . . . .	7
<b>2</b>	<b>Grundlegende Eigenschaften planarer Graphen</b>	<b>8</b>
2.1	planare Einbettung: . . . . .	8
2.2	Satz von Euler (1790): . . . . .	8
2.3	Dualität von Schnitten und Kreisen . . . . .	10
2.4	Minor bzw. Unterteilung . . . . .	10
2.5	Satz von Kuratowski (1930) . . . . .	10
2.5.1	Vorbereitung des Beweises . . . . .	11
<b>3</b>	<b>Färbung planarer Graphen</b>	<b>12</b>
3.1	Färbungsproblem (k-Färbung) . . . . .	12
3.2	Listenfärbungsproblem . . . . .	12
3.2.1	Beweis der schärferen Behauptung per Induktion . . . . .	13
<b>4</b>	<b>Matching</b>	<b>15</b>
4.1	Matching-Algorithmus für pl. Graph $G = (V, E)$ . . . . .	17
4.2	Mixed-Max-Cut in pl. Graphen . . . . .	17
4.3	Beweis zu Folie (Kozykel und st-Schnitte) . . . . .	20
4.4	Beweis zu Folie "Betrachte Fluss von auf P" . . . . .	20
<b>5</b>	<b>Das Menger-Problem</b>	<b>21</b>
5.1	Satz von Menger . . . . .	21
5.2	Menger-Problem . . . . .	21
5.2.1	Menger-Problem in pl. Graphen; kantendisj. Variante . . . . .	22
5.2.2	zu Schritt 1 . . . . .	23
5.2.3	Zu Schritt 2 . . . . .	23
5.2.4	Konstruktion der Kreise $C_1 \dots C_l$ . . . . .	23
5.2.5	Zu Schritt 2 (Wdh) . . . . .	23
5.2.6	Zu Schritt 3 . . . . .	24
5.3	Das knotendisjunkte Menger-Problem . . . . .	25

<b>6</b>	<b>Das Problem von Okamura und Seymour</b>	<b>27</b>
6.1	Kantendisjunktes Wegpackungsproblem . . . . .	27
6.1.1	2 Phasen . . . . .	28
6.1.2	Instanz mit Klammerstruktur . . . . .	28
6.1.3	Korrektheit von Phase 1 . . . . .	29
6.2	Zu Phase 2 . . . . .	29
<b>7</b>	<b>Planartest in Linearzeit</b>	<b>32</b>
7.1	Beweis zum Lemma defekte Kanten bilden Pfad . . . . .	32

---

Der Mitschrieb sollte vollständig sein.

Zeichnungen / Graphenbilder fehlen leider, die kann ich nicht flüssig malen, also falls jemand Lust hat die rein zu machen...

# 1 Einführung

ungerichteter, einfacher Graph  $G = (V, E)$

Einbettung (in die Ebene):

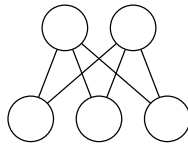
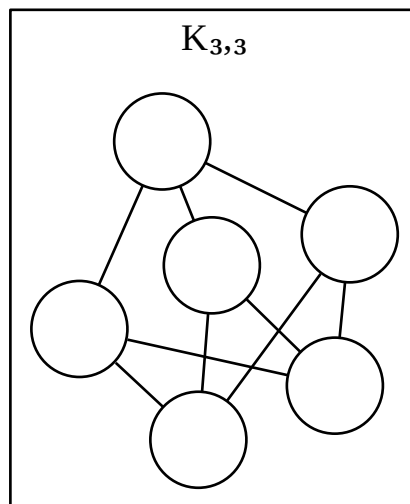
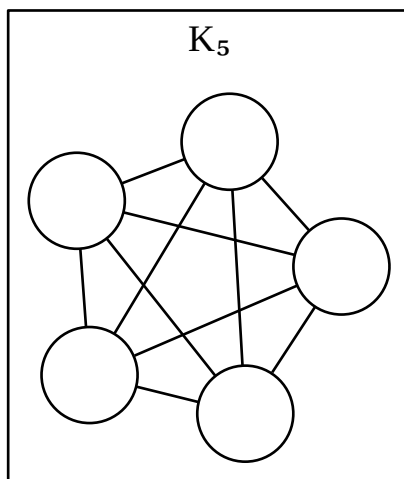
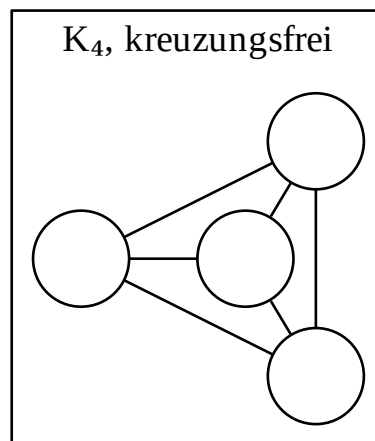
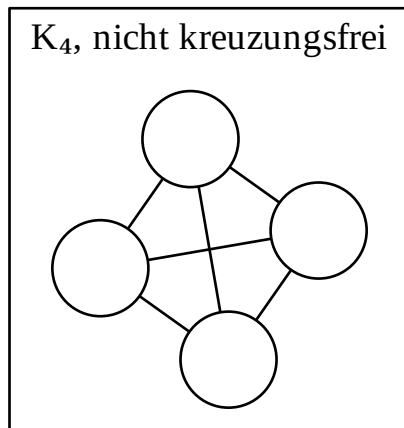


Abbildung 1.1: Einbettung von  $K_{2,3}$

- Koordinaten für Knoten, Kanten als Geraden bzw. Kurven  
     $\rightsquigarrow$  geometrische Einbettung
- Anordnung der Kanten in Adjazenzliste  
     $\rightsquigarrow$  kombinatorische Einbettung

Ein Graph, der sich kreuzungsfrei in die Ebene einbetten lässt ist planar. Beispiele:



$K_5$  und  $K_{3,3}$  sind *nicht* planar.

## 1.1 $K_5$ ist nicht planar

angenommen es gäbe planare Einbettung von  $K_5$ , o.B.d.A. mit Knoten 1 kreuzungsfrei zu Knoten 2,3,4 und 5 verbunden, im Uhrzeigersinn. Einbettung der Kante  $\{2,4\}$  zerlegt Ebene in zwei Gebiete: “Inneres” von 1241 und “Äußeres” von 1241  $\Rightarrow$  Kante  $\{3,5\}$  muss dann Kante  $\{2,4\}$  kreuzen

(todo: Illustration)

## 1.2 $K_{3,3}$ ist nicht planar

$$K_{3,3} = (V, E), V := \{1, 2, 3, g, w, s\}, E := \{\{i, j\} : i \in \{1, 2, 3\}, j \in \{g, w, s\}\}$$

angenommen  $K_{3,3}$  besitze kreuzungsfreie Einbettung, o.B.d.A. mit Kreis  $w1g2s3w$  kreuzungsfrei mit  $\{1,s\}$  im Inneren des Kreises  $\Rightarrow \{2,w\}$  im Äußeren. Dann ist  $\{3,g\}$  nicht mehr kreuzungsfrei einbettbar.

---

Graphen, die  $K_5$  oder  $K_{3,3}$  “enthalten” können also nicht planar sein.

## 1.3 Themen

- Charakterisierung planarer Graphen “Satz von Kuratowski”
- Planaritätstest geht in Linearzeit
- es gibt Probleme, die i.A. NP-schwer sind, und in planaren Graphen polynomial lösbar, z.B. Max-Cut
- polynomial lösbare Probleme, die in pl. Graphen effizienter (oft in Linearzeit) gelöst werden können, z.B. Matchings, Menger, MaxFlow ...

## 1.4 Vierfarbensatz

Jede Landkarte kann mit 4 Farben gefärbt werden. “Dualgraph” zu Landkarte ist planarer Graph (Länder als Knoten, Kanten als Grenzen)  $\rightarrow$  jeder pl. Graph kann mit 4 Farben gefärbt werden

### Einschub: Konzept des Dualgraphen

planarer Graph  $G = (V, E)$  mit pl. Einbettung mit Facettenmenge  $F$  von  $G$  bzgl. dieser Einbettung. Definiere dazu Dualgraph  $G^* = (V^*, E^*)$  wie folgt:

- zu jeder Facette  $f \in F$  def. Knoten  $v_f$  in  $V^*$
- zu jeder Kante  $e = \{u, v\} \in E$  gibt es eine Kante  $e^* \in E^*$ , die die beiden Knoten in  $V^*$  zu den Facetten in  $F$ , die durch  $e$  getrennt werden, verbindet.

$G = (V, E)$ ,  $F = f_1, f_2, f_3$  (todo: Graph mit Dualgraph)

$G = (V, E)$ , andere Darstellung (todo: Graph)

Eigenschaften des Dualgraphen:

- i) Dualgraph  $G^*$  zu einfachem planaren Graphen ist i.A. nicht einfach

- ii) Dualgraph  $G^*$  ist wieder planar und  $(G^*)^* = G$  (wobei  $G^*$  Dualgraph zu  $G$ )
- iii) planarer Graph  $G$  kann verschiedene Dualgraphen haben (zu verschiedenen Einbettungen)

## 1.5 Satz (1977 Appel & Haken)

Jeder pl. Graph kann mit höchstens 4 Farben gefärbt werden. Beweis mit Fallunterscheidung.

### “Folklore”

pl. Graph kann mit 5 Farben gefärbt werden.

Beweis per Induktion über Anz. Knoten  $n$ .

**IS**  $G$  habe  $n$  Knoten,  $v$  sei Knoten in  $G$  mit min. Grad.

Dann ist  $d(v) \leq 5$  (Beweis später)

Entscheidender Fall:  $d(v) = 5$ :

sei Facette von  $v$  so eingebettet: (*todo: Bild*) Dann ex. gegenüberliegende Knoten  $i, j \in \{1, 2, 3, 4, 5\}$  mit  $\{i, j\} \notin E$ .

Verschmelze  $i$  und  $j$  und färbe nach IV entspr. Graph mit 5 Farben. Dies induziert 5-Färbung von  $G-v$ , in der  $i$  und  $j$  dieselbe Farbe haben und entsprechend  $\{1, 2, 3, 4, 5\}$  nur 4 Farben benutzen.  $\Rightarrow$  Beh.

(*todo: Bild*)

## 2 Grundlegende Eigenschaften planarer Graphen

### 2.1 planare Einbettung:

Graph  $G = (V, E)$  kann dargestellt werden indem man die Knoten aus  $V$  auf Punkte im  $\mathbb{R}^2$  und die Kanten aus  $E$  auf Jordan-Kurven (d.h. stetige sich selbst nicht kreuzende Kurven) zwischen den Endpunkten abdeckt.

$G$  heißt *planar* wenn es eine Darstellung gibt, bei der sich die Kanten höchstens in einem gemeinsamen Endpunkt berühren.

- planare Einstellung zerlegt Ebene in *Facetten* (Gebiete, Flächen)
- planare Einbettung, die durch ihre Facetten bzw. die Reihenfolge der Kanten in Adjazenzlisten beschrieben ist, heißt *kombinatorische Einbettung*
- planare Einbettung, die durch Koordinaten der Punkte beschrieben ist, heißt *geometrische Einbettung*

Facettenmenge  $\mathcal{F}$ ,  $|\mathcal{F}| = f$

### 2.2 Satz von Euler (1790):

In einem zusammenhängenden nichtleeren planaren Graph  $G = (V, E)$  gilt für jede planare Einbettung (geg. durch  $\mathcal{F}$ ), dass

$$n - m + f = 2$$

(wobei  $|V| = n, |E| = m, |\mathcal{F}| = f$ )

**Beweis** per Induktion über  $m$ :

**IA:**  $m = 0$ , es ist  $n = 1, f = 1 \Rightarrow$  Beh.

Sei also  $m \geq 1$

*Fall 1:*  $G$  enthalte einen Kreis

$\Rightarrow$  es existiert  $l \in E$  so dass  $G' := G - e = (V, E \setminus e)$  ebenfalls zusammenhängend und  $e$  an zwei Facetten grenzt die zu einer Facette in  $G'$  werden.



$\Rightarrow f'$  #Facetten von  $G'$  erfüllt

$$f' = f - 1 \xRightarrow{IV} n - (m - 1) + f' = 2$$

$$\Rightarrow n - m + f = 2$$

*Fall 2:*  $G$  enthält keinen Kreis, ist also Baum und  $|\mathcal{F}| = 1$ . Für beliebige  $e \in E$  zerfällt  $G' = G - e$  in zwei Zusammenhangskomponenten  $G_1 = (V_1, E_2)$  und  $G_2 = (V_2, E_2)$  und nach IV:

$$n_1 - m_1 + f_1 = 2, n_2 - m_2 + f_2 = 2$$

Da

$$n = n_1 + n_2, m = m_1 + m_2 - 1$$

$$\Rightarrow n - m + f = n_1 + n_2 - m_1 - m_2 - 1 + 1 = \underbrace{(n_1 - m_1 + 1)}_2 + \underbrace{(n_2 - m_2 + 1)}_2 - 2 = 2$$

### Folgerungen:

- #Facetten ist für jede planare Einbettung von  $G$  gleich
- #Kanten eines Baumes mit  $n$  Knoten ist  $n-1$

*Lemma:* Ein planarer Graph mit  $n$  Knoten ( $n \geq 3$ ) hat höchstens  $3n - 6$  Kanten.

*Beweis:* o.B.d.A sei  $G$  maximal planar (d.h. Hinzunahme weiterer Kanten zerstört Planarität)

(todo: Bild)

Dann ist für jede planare Einbettung jede Facette ein Dreieck und jede Kante grenzt an genau zwei Facetten.

$$3f = 2m$$

$\stackrel{=}{\text{mit Euler}}$

$$3(2 - n + m) = 6 - 3n + 3m$$

*Lemma:* Sei  $G$  pl. Graph mit mind 3 Knoten.  $d_{max}(G)$  bezeichne Maximalgrad in  $G$ ,  $n_i$  #Knoten von Grad  $i$ .

Dann gilt:

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

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

Da  $m \leq 3n - 6$  folgt

$$6 \sum_{i=0}^{d_{\max}(G)} n_i = 6n \geq 2m + 12 = \sum_{i=0}^{d_{\max}(G)} i \cdot n_i + 12$$

*Folgerung:* Jeder planare Graph enthält mind. einen Knoten  $v$  mit  $d(v) \leq 5$ .

---

## 2.3 Dualität von Schnitten und Kreisen

*(todo: Bild Dualgraph)*

Planarer Graph  $G$  mit Einbettung  $\mathcal{F}_i$  Dualgraph  $G^*$  dazu. Dann gilt:

Ein Schnitt in  $G$  ( $\hat{=}$  entspr. Kantenmenge) induziert eine Menge von Kreisen in  $G^*$  und umgekehrt.

## 2.4 Minor bzw. Unterteilung

*(todo: Bild  $G'$  Subgraph von  $G$ )*

$G' = (V', E')$  heißt *Subgraph* von  $G = (V, E)$  wenn  $V' \subseteq V$  und  $E' \subseteq E$ .

$G' = (V', E')$  heißt *Unterteilung* von  $G = (V, E)$  wenn  $G'$  aus  $G$  entsteht indem man Kanten von  $G$  durch einfache Wege ersetzt.

Ein Graph  $H$  heißt *Minor* von  $G$  wenn  $H$  aus  $G$  entsteht durch Löschen von Knoten oder/und Kanten und/oder Knotenkontraktion von Knoten von Grad 2.

$H$  ist *Minor* von  $G$  falls  $G$  eine Unterteilung von  $H$  als Subgraph enthält.

*(todo: Bild  $G'$  Unterteilung von  $G$ )*

*(todo: Bild  $G'$  Minor von  $G$ )*

## 2.5 Satz von Kuratowski (1930)

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

“ $\Rightarrow$ ” klar, da  $K_5$  und  $K_{3,3}$  nicht planar.

“ $\Leftarrow$ ”: Es ist also “nur” zu zeigen: Wenn  $G$  nicht planar, dann enthält  $G$  einen  $K_5$  oder  $K_{3,3}$  als Minor.

### 2.5.1 Vorbereitung des Beweises

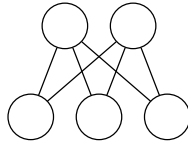


Abbildung 2.1:  $K_{3,2}$

Nenne Graph der  $K_{3,2}$  als Minor enthält  $\theta$ -Graph ( $\theta$  Minor von  $K_{3,2}$ )

(Beginn Vorlesung 2015-04-21)

Siehe Beweisfolien ([kuratowski\\_slides.pdf](#))

(Beginn Vorlesung 2015-04-29)

## 3 Färbung planarer Graphen

(Kap.4 im Skript; “Listenfärbung” nicht im Skript, siehe [coloring\\_slides.pdf](#))

### 3.1 Färbungsproblem (k-Färbung)

geg.  $G = (V, E)$ ,  $k$  Farben

**Problem** Existiert *korrekte* Färbung der Knoten aus  $V$  mit diesen  $k$  Farben, d.h. falls  $\{u, v\} \in E \implies \text{Farbe}(u) \neq \text{Farbe}(v)$

### 3.2 Listenfärbungsproblem

geg.  $G = (V, E)$ ,  $k \in \mathbb{N}$

**Problem** Gibt es für *jede* Zuordnung von Listen  $S_v$  zu Knoten  $v \in V$  mit  $|S_v| = k$  eine korrekte Färbung der Knoten bei der jeder Knoten eine Farbe aus seiner Liste enthält?

**Beobachtung** Listenfärbung ist Verallgemeinerung von Färbungsproblem.

**Satz** Jeder planare Graph ist 5-listenfärbbar.

**Beweis** Induktion über  $|V| = n$  (benutzen nicht, dass  $v$  exist. mit  $d(v) \leq 5$ ).

beweisen schärfere Behauptung:

Falls  $G$  planar und

- jede innere Facette Dreieck
- äußere Facette durch Kreis  $C = v_1 v_2 \dots v_k v_1$  begrenzt
- $v_1$  mit Farbe 1 gefärbt
- $v_2$  mit Farbe 2 gefärbt
- jeder Knoten mit Liste von mind. 3 Farben assoziiert
- jeder Knoten aus  $G - C$  mit Liste von mind. 5 Farben assoziiert

dann folgt:  $G$  korrekt färbbar

Offensichtlich folgt daraus 5-Listenfärbbarkeit.

### 3.2.1 Beweis der schärferen Behauptung per Induktion

Falls  $G = (V, E)$  planar und  $|V| = 3$  trivial

**Induktionsschritt**  $G = (V, E)$  pl. und  $|V| \geq 4$ , Kreis  $C$  der äußeren Facette begrenzt zwei Fälle:  $C$  enthält Sehne  $\{v, w\}$  im Inneren oder nicht

(todo: bild)

#### Fall 1: $C$ enthält Sehne $\{v, w\}$

$\{v, w\}$  induziert eindeutig bestimmte Kreise  $C_1$  und  $C_2$  welche jeweils Subproblem  $G_1$  und  $G_2$  induzieren. o.B.d.A. enthalte  $C_1$  Kante  $\{v_1, v_2\}$  (und damit  $v_1, v_2$  nicht beide auf  $C_2$ ). Wende IV auf  $C_1$  an und dann IV auf  $C_2$  wobei  $v$  und  $w$  Rolle von  $v_1, v_2$  spielen.  $\Rightarrow$  Färbung von  $G_1$  und  $G_2$  ind. korrekte Färbung von  $G$ .

#### Fall 2: $C$ enthält keine Sehne

Seien  $v_{k-1}, u_1, u_2, \dots, u_l, v_1$  die Nachbarn von  $v_k$ . Da alle inneren Facetten Dreiecke ist  $v_{k-1}u_1 \dots u_lv_1$  Weg  $P$  und  $(C - v_k) \cup P = C'$  wird Kreis der äußere Facette begrenzt. "Reserviere" zwei Farben aus Liste von  $v_k$  und entferne diese ggf. aus Listen von  $u_1, \dots, u_l$ . Wende IV auf durch  $C'$  induz. Graph an. Höchstens eine der beiden reservierten Farben wird für  $v_{k-1}$  verwendet, die andere kann für  $v_k$  verwendet werden.

#### Satz

Nicht jeder planare Graph ist 4-listenfärbbar.

#### Beweis

konst. Gegenbeispiel, d.h. planarer Graph mit Listenzuweisung mit Listen  $S_v, |S_v| = 4$ , so dass Graph nicht korrekt färbbar unter Berücksichtigung der  $S_v$ .

Kern der Konstruktion:

(todo: bild)

hat "vis-à-vis-Eigenschaft", d.h. in korrekte Färbung müssen mind. zwei gegenüberliegende Eckknoten dieselbe Farben haben. (klar!)

---

(Beginn Vorlesung 2015-05-12)

Bemerkung zu Planar Separator Theorem: Linearzeitimplementierung

PST: pl.  $G = (V, E)$ ; exist Separator  $S$  der  $G$  in  $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$  trennt mit

1.  $|V_1|, |V_2| \leq \frac{2}{3}n$
2.  $|s| \leq 4\sqrt{n}$

## 4 Matching

$G = (V, E)$ , ein Matching  $M \subseteq E$  sodass keine zwei Kanten aus  $M$  gemeinsame Endknoten haben.

$w : E \rightarrow \mathbb{R}$

- Finde  $M \subseteq E$  Matching mit max. Gewicht, wobei  $w(M) = \sum_{l \in M} w(l)$
- Finde  $M \subseteq E$  Matching mit max. Kardinalität, (Fall  $w(l) = 1$  f.a.  $l \in E$ )

Beide Probleme sind auch für bel. Graphen in P.

(todo: bild)

alternierender Weg bzgl.  $M \rightarrow$  Vertauschen der Kanten auf Weg aus  $M$  mit Kanten auf Weg, die nicht in  $M$  sind resultiert in größerem Matching  $M^*$

- 
- Ein bezüglich einem Matching  $M$  *alternierender Weg* ist ein einfacher Weg oder einfach Kreis, dessen Kanten abwechselnd in  $M$  und  $E \setminus M$  sind.
  - Alternierender Weg  $P$  (bezeichne entsprechende Kantenmenge) ist *erhöhender Weg* falls

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

und  $P$  entweder Kreis (gerader Länge) oder dessen erste und letzte Kante beide in  $M$  sind oder inzident zu einem ungematchten Knoten.

### Beobachtung

$M$  Matching,  $P$  erhöhender Weg bzgl.  $M \Rightarrow M' = (M \setminus P) \cup (P \setminus M)$  wieder Matching mit  $w(M') > w(M)$ .

### Lemma

Sei  $G = (V, E)$ ,  $w : E \rightarrow \mathbb{R}$ ,  $M$  Matching in  $G$ . Dann ist  $w(M)$  maximal genau dann wenn es keinen erhöhenden Weg bzgl.  $M$  gibt.

## Beweis

“ $\Rightarrow$ ” klar

“ $\Leftarrow$ ” sei  $M$  nicht max. Matching in  $G$  und es existiert kein bzgl.  $M$  erhöhender Weg. Dann exist. Matching  $M^*$  mit  $w(M^*) > w(M)$ . Betrachte Subgraph  $G_{M^* \triangle M}$  von  $G$  der durch

$$M^* \triangle M = M \cup M^* \setminus (M \cap M^*)$$

induziert wird. In diesem Graph haben alle Knoten Grad 1 und Grad 2 und er besteht aus einfachen Wegen und Kreisen.

Falls kein Kreis in  $G_{M \triangle M^*}$  erhöhend bzgl.  $M$  so exist. in  $G_{M \triangle M^*}$  ein inklusionsmaximaler Weg, der Weg  $P$  in  $G$  induziert mit  $w(P \cap M^*) > w(P \cap M)$

$\Rightarrow$  beide Endkanten von  $P$  gehören zu  $M$  oder eine Endkante gehört nicht zu  $M$  und ist inzident zu einem Knoten  $v$ ,  $v$  nicht durch  $M$  gematcht.

$\Rightarrow P$  erhöhend bzgl.  $M$ . (widerspruch)

## Lemma

$G = (V, E), w : E \rightarrow \mathbb{R}, v \in V, M$  Matching in  $G - v$  (Graph induziert durch  $V \setminus \{v\}$ )

Dann gilt:

1. Falls es keinen bzgl.  $M$  erhöhenden Weg in  $G$  gibt mit Endknoten  $v$ , so hat  $M$  auch in  $G$  max. Gewicht
2. Falls es bzgl.  $M$  erhöhenden Weg in  $G$  gibt mit Endknoten  $v$  und  $w(P \cap E \setminus M) - w(P \cap M)$  maximal unter allen solchen erhöhenden Wegen, so ist  $M^* = M \triangle P$  Matching maximalen Gewichts in  $G$ .

(todo: bild i) ii))

## Beweis

erhöhender Weg bzgl.  $M$  in  $G$  muss  $v$  als Endknoten haben. Sei  $M^*$  max. Matching in  $G$   
 $\Rightarrow M \triangle M^*$  ist Menge von alternierenden Kreisen und Wegen bzgl.  $M$  bzw.  $M^*$  in  $G$

$P$  erhöhender Weg bzgl.  $M$  in  $G_{M \triangle M^*} \Rightarrow P$  erhöhender Weg bzgl.  $M$  in  $G$ .

Da  $G_{M \triangle M^*}$  höchstens bzgl.  $M$  erhöhender Weg  $P^*$  mit Endknoten  $v$  enthält gilt  $w(M) - w(P^* \cap M) = w(M^*) - w(P^* \cap M^*)$

Gewicht des Matching  $M'$ , das durch erhöhen entlang  $P^*$  entsteht ist:

$$w(M') = w(M) - w(P^* \cap M) + w(P^* \cap E \setminus M) = w(M) - w(P^* \cap M) + w(P^* \cap M^*)$$

$\Rightarrow$

$$w(M') = w(M^*)$$



---

(Beginn Vorlesung 2015-05-20)

## 4.1 Matching-Algorithmus für pl. Graph $G = (V, E)$

1. Zerlege  $G$  in  $G_1, G_2$  durch Separator  $S$  entspr. Planar-Separator-Theorem und berechne rekursiv in  $G_1$  und  $G_2$  Matchings  $M_1$  bzw.  $M_2$  maximalen Gewichts; bezeichne  $M = M_1 \cup M_2$
2. Solange  $S \neq \emptyset$ 
  - wähle  $v \in S, S := S \setminus \{v\}$  und berechne mit Lemma aus  $M'$  matching max. Gewichts in  $G' + v$

---

$$t(n) = t(c_1 n) + t(c_2 n) + c_3 \cdot \sqrt{n} \cdot t'(n)$$

$t'(n)$  Laufzeit für Lemma,  $c_1, c_2, c_3$  Konstante;  $c_1, c_2 \leq \frac{2}{3}, c_1 + c_2 \leq 1$

Mit Master-Theorem kann  $t(n)$  abgeschätzt werden durch

$$t(n) \in O(n^{\frac{3}{2}}) \text{ falls } t'(n) \in O(n)$$

$$t(n) \in O(n^{\frac{3}{2}} \log n) \text{ falls } t'(n) \in O(n \log n)$$

## 4.2 Mixed-Max-Cut in pl. Graphen

$G = (V, E), S \subseteq E$  Schnitt von  $G$  falls durch  $E \setminus S$  induz. Subgraph unzusammenhängend und für alle  $\{u, v\} \in S$   $u$  und  $v$  in verschiedenen Zusammenhangskomponenten dieses Subgraphs.

Kantengewichte  $w : E \rightarrow \mathbb{R}$

### Mixed-Max Cut

Finde Schnitt  $S$  mit  $w(s) = \sum_{l \in S} w(l)$  maximal.

Ist in bel. Graphen NP-schwer.

## Beobachtung

MIXED-MAX CUT Problem und MIXED-MIN CUT Problem äquivalent.

Spezialfall: MIN CUT Problem mit  $w : E \rightarrow \mathbb{R}_0^+$  ist auch für bel. Graphen in P

polynomialer Algorithmus für MIXED-MAX CUT in pl. Graphen:

verwende:

- Dualität von Schnitten und Kreisen
- max. Matching bzw. Planar Separator Theorem

Laufzeit in  $O(n^{3/2} \log n)$ .

Es gilt:  $G$  enthält Euler-Kreis g.d.w.  $E$  kantendisjunkte Vereinigung einfacher Kreise g.d.w. für alle  $v \in V$  ist Knotengrad  $d(v)$  gerade.

Dualität von Schnitt in  $G$  und Menge von einf. Kreisen (= Kantenmenge, in der f.a. Knoten  $v$   $d(v)$  gerade (= gerade Menge)) in Dualgraph  $G^*$  (bzgl. bel. pl. Einbettung)

(todo: bild gewichteter dualgraph)

**Schritt 1** trianguliere  $G$  in  $O(n)$ ; zusätzliche Kanten erhalten Gewicht 0

**Schritt 2** berechne in  $O(n)$  Dualgraph bzgl. bel. pl. Einbettung;  $G^*$  ist dann 3-regulär (d.h. für alle  $v$ :  $d(v) = 3$ )

**Schritt 3** konstruiere zu  $G^*$  Graph  $G'$  so dass perfektes Matching min. Gewichts in  $G'$  eine gerade Menge (bzw. Menge von Kreisen) max. Gewichts in  $G^*$  induziert.

**Schritt 4** berechne in  $O(n^{3/2} \log n)$  solch ein Matching bzw. gerade Menge

**Schritt 5** falls diese gerade Menge nicht leer, gib entspr. Schnitt aus. Ansonsten "Sonderfall"

Matching  $M$  in  $G = (V, E)$  mit  $|V|$  gerade heißt perfekt falls  $|M| = \frac{|V|}{2}$

---

## zu Schritt 3

beachte dass  $G^*$  3-regulär, Matching ergibt zwei Fälle:

(todo: Dreieck mit kante an jeder Ecke)

Fall 1: (todo: Alle drei äußeren Kanten gematcht) Fall 2: (todo: Eine kante von dreieck, eine äußere) (todo: bild)

$G'$  entsteht aus  $G^*$  indem jeder Knoten durch Dreieck ersetzt wird. Sei  $m$  #Kanten in  $G^*$ ,  $n$  #Knoten in  $G^* \Rightarrow 3n = 2m \Rightarrow n$  gerade  $\Rightarrow$  #Knoten in  $G'$  gerade

#### zu Schritt 4

konstruiere perfektes Matching min. Gewichts in  $G'$

**Beobachtung:** M perfektes Matching min. Gewichts in  $G = (V, E)$  mit  $w : E \rightarrow \mathbb{R}$  g.d.w. M perfektes Matching max. Gewichts bzgl. Gewichts fkt.  $\bar{w} : E \rightarrow \mathbb{R}$  mit  $\bar{w}(l) := W - w(l)$ , W geeignet gewählte Konstante.

*Erzwinge* dass Matching max. Gewichts *perfekt* ist:

- zu M perfekt betrachte

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

, wobei  $w_{max} = \max_{l \in E} w(l)$

- zu M' nicht perfekt gilt

$$\bar{w}(M') \leq \left(\frac{n}{2} - 1\right)(W - w_{min}), w_{min} = \min_{l \in E} w(l)$$

Wähle also W so dass  $\frac{n}{2} \cdot (W - w_{max}) > \left(\frac{n}{2} - 1\right)(W - w_{min})$

#### zu Schritt 5

Komplementmenge von perfekten Matching min. Gewichts in  $G'$  induziert gerade Menge max. Gewichts in  $G^*$  und damit max. Schnitt in G.

Es kann sein, dass resultierende Menge leer ist! Passiert wenn max. Schnitt negatives Gewicht hat.

→ *Sonderfall:* Wollen nichttrivialen Schnitt erzwingen.

betrachte wieder Schritt 3: erzwinge, dass in perfekten Matching minimalen Gewichts für mindestens ein Knoten v aus  $G^*$  Fall 2 eintritt.

*Vorgehensweise:* betrachte alle Knoten v aus  $G^*$  und  $G^* - v$  sowie durch perfektes Matching in  $G'$  induzierte Matching in  $G^* - v$  und berechne mit "Matching-Lemma" Matching in  $G^*$ .

Wähle M mit  $w(M) = \min_{v \in V^*} w(M_v)$

*Frage:* Wie kann man dabei Fall 2 an v erzwingen?

(Beginn Vorlesung 2015-05-26)

### 4.3 Beweis zu Folie (Kozykel und st-Schnitte)

- 1) s, t auf selber Seite von  $C^*$   $\Rightarrow$  P kreuzt  $C^*$  gleich oft in jeder Richtung  $\Rightarrow$  C enthält selbe Zahl von Kanten in P und  $\text{rev}(P) \Rightarrow \pi(C) = 0$
- 2) s rechts, t links  $\Rightarrow$  P kreuzt einmal mehr von rechts nach links  $\Rightarrow \pi(C) = 1$
- 3) analog  $\Rightarrow \pi(C) = -1$

C s, t-Schnitt  $\Rightarrow$  P kreuzt  $C^*$  von rechts nach links  $\Rightarrow \pi(C) = 1$ .  $\pi(C) = 1 \Rightarrow$  Fall 2; s rechts, t links.  $\Rightarrow$  C st-Schnitt.

### 4.4 Beweis zu Folie "Betrachte Fluss von auf P"

**Beweis:**

" $\Rightarrow$ " Angenommen  $G_\lambda^*$  enthält neg. Kreis  $C^*$

$$0 > c(\lambda, C^*) = \sum_{e \in C^*} c(e) = \sum_{e \in C^*} c(e) - \lambda \sum_{e \in C^*} \pi(e) = \underbrace{\sum_{e \in C^*} c(e)}_{\geq 0} - \underbrace{\lambda \sum_{e \in C^*} \pi(e)}_{> 0}$$

$$\Rightarrow \pi(C) = 1. \Rightarrow C \text{ ist st-Schnitt}$$

Außerdem  $\sum_{e \in C^*} c(e) < \lambda \Rightarrow$  st-Schnitt mit Kap.  $< \lambda$ .

" $\Leftarrow$ ":  $G_\lambda^*$  enthält keinen neg. Kreis  $\Rightarrow$  kürzeste Wege wohldef.

Wähle o in  $G_\lambda^*$  bel. Ursprung.

$\text{dist}(\lambda, p)$ : Distanz von p zu o in  $G_\lambda^*$ .

Def:  $\phi(\lambda, e) := \text{dist}(\lambda, \text{head}(e^*)) - \text{dist}(\lambda, \text{tail}(e^*)) + \lambda \cdot \pi(e)$

Zeige:  $\phi$  ist gültiger st-Fluss.

$$1) \text{ Für } v \in V \text{ gilt: } \sum_W \phi(v \rightarrow w) = \sum_W \pi(v \rightarrow w)$$

$$\Rightarrow \phi(\lambda, \cdot) \text{ ist Fluss mit Wert } \lambda.$$

$$2) \text{ slack}(\lambda, e^*) := \text{dist}(\lambda, \text{tail}(e^*)) + c(\lambda, e) - \text{dist}(\lambda, \text{head}(e^*))$$

$$\text{Gilt: } \text{slack}(\lambda, e) = c(e) - \phi(\lambda, e)$$

$$\phi(\lambda, e) \leq c(e) \Leftrightarrow \text{slack}(\lambda, e) \geq 0.$$

$$\text{Wäre } \text{slack}(\lambda, e) < 0 \Rightarrow \text{dist}(\lambda, \text{head}(e^*)) > \text{dist}(\lambda, \text{tail}(e^*)) + c(\lambda, e^*)$$

(widerspruch)

Max  $\lambda$  sodass kein neg. Kreis in  $G_\lambda^*$  ist Länge eines kürzesten ts-Wege in  $G_\lambda^*$ .

## 5 Das Menger-Problem

Zur Erinnerung:  $S \subset V$  heißt *Separator* in  $G = (V, E)$ , falls  $G - S$  unzusammenhängend.  $S \subset E$  heißt *Schnitt* in  $G = (V, E)$  falls der durch  $E \setminus S$  induzierte Teilgraph von  $G$  unzusammenhängend.

Zu  $u, v \in V$  def:

$$K_G(u, v) := \begin{cases} |V| - 1 & \text{falls } \{u, v\} \in E \\ \min_{S \subset V} |S| & \text{sonst} \end{cases}$$

und

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

$$\lambda_G(u, v) := \min_{S \subset E} |S|$$

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

Zwei Wege heißen *kantendisjunkt*, wenn sie keine gemeinsame Kante enthalten und (intern) *knotendisjunkt* wenn sie (außer Anfangs- und Endknoten) keinen Knoten gemeinsam haben.

### 5.1 Satz von Menger

Seien  $s$  und  $t$  Knoten in  $G = (V, E)$  ( $\{s, t\} \notin E$  bei knotendisj. Version)

- $K_G(s, t) \geq k$  g.d.w. es mind.  $k$  paarweise knotendisj.  $s$ - $t$ -Wege gibt
- $\lambda_G(s, t) \geq k$  g.d.w. es mind.  $k$  paarweise kantendisj.  $s$ - $t$ -Wege gibt

### 5.2 Menger-Problem

Zu  $G = (V, E)$ ,  $s, t \in V$  finde max. Anzahl knotendisj. bzw. kantendisj.  $s$ - $t$ -Wege.

### 5.2.1 Menger-Problem in pl. Graphen; kantendisj. Variante

#### Linearzeitalgorithmus basierend *Right-First-DFS*

*Spezialfall:*  $s$  und  $t$  liegen auf derselben Facette

(*todo: Bild*)

RIGHT-FIRST  $\hat{=}$  im Gegenuhrzeigersinn nächste freie Kante in Adjazenzliste (relativ zur aktuellen eingehenden Kante)

---

Linearzeitalgorithmus bestehend aus 4 Schritten;  $G = (V, E)$  planar eingebettet, o.B.d.A.  $t$  auf äußerer Facette.

#### 1. Schritt

ersetze  $G = (V, E)$  durch  $\vec{G} = (V, \vec{E})$  indem  $\{u, v\} \in E$  durch  $(u, v)$  und  $(v, u)$  ersetzt wird. (In  $O(n)$ )

#### 2. Schritt

berechne in  $O(n)$  Menge gerichteter einfacher kantendisj. Kreise  $\vec{C}_1, \dots, \vec{C}_l$  und konstruiere (in  $O(n)$ ) aus  $\vec{G}$  Graph  $\vec{G}_C = (V, \vec{E}_C)$ , der entsteht in dem alle Knoten auf den  $\vec{C}_i$  umgedreht werden.

#### 3. Schritt

berechne in  $\vec{G}_C$  in  $O(n)$  mittels RIGHT-FIRST-DFS eine max. Anzahl kantendisj. gerichteter s-t-Wege.

#### 4. Schritt

berechne aus dem in Schritt 3 gefundenen s-t-Wege in  $\vec{G}_C$  gleiche Anzahl kantendisj. s-t-Wege in  $G$  in  $O(n)$ .

---

### 5.2.2 zu Schritt 1

(todo: Bild)

**Lemma:** Seien  $p_1, \dots, p_r$  kantendisjunkte gerichtete s-t-Wege in  $\vec{G}$ . Dann enthält  $P = \{\{u, v\} \in E : \text{genau eine der Kanten } (u, v) \text{ und } (v, u) \text{ liegt auf einem der } p_i\}$  gerade r kantendisjunkte s-t-Wege in G.

**Beweis:** zwei Fälle (todo: Bilder)

### 5.2.3 Zu Schritt 2

$C_1, \dots, C_l$  in  $\vec{G}$  so dass 1.  $\vec{G}_C$  enthält keine Rechtskreise, d.h. keine Kreise deren Inneres rechts liegt. 2. Sei  $\vec{P}_C \subseteq \vec{E}_C$  Menge der Kanten auf kantendisjunkten s-t-Wege in  $\vec{G}_C$  und  $\vec{P} \subseteq \vec{E}$  wobei

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

Dann induziert  $\vec{P}$  k kantendisjunkte gerichtete s-t-Wege in  $\vec{G}$  g.d.w.  $\vec{P}_C$  k kantendisjunkte gerichtete s-t-Wege in  $\vec{G}_C$  induziert.

### 5.2.4 Konstruktion der Kreise $C_1 \dots C_l$

sei f Facette in G bzw.  $\vec{G}$ ; def. Abstand von f von äußerer Facette als

$dist(f) :=$  Länge eines kürzesten Weges von Dualknoten zu f zu Dualknoten zu äußerer Facette  $f_0$  in  $G^*$

def.  $C_i$  als Vereinigung der einfachen Kreise in G für die alle Facetten f im Inneren  $dist(f) \geq i$  erfüllen.  $\vec{C}_i$  sei entsprechend Rechtskreis in  $\vec{G}$ .

Drehe alle diese  $\vec{C}_i$  um  $\rightarrow \vec{G}_C$

(Beginn Vorlesung 2015-06-03)

### 5.2.5 Zu Schritt 2 (Wdh)

$C_1, \dots, C_l$  geeignete Kreise in G;  $\vec{C}_1, \dots, \vec{C}_l$  entsprechende Rechtskreise in  $\vec{G}$  drehe  $\vec{C}_1, \dots, \vec{C}_l$  um  $\sim \rightarrow \vec{G}_C$ ;  $\vec{G}_C$  enthält keine Rechtskreise, da für jeden Rechtskreis in  $\vec{G}$  beim Übergang zu  $\vec{G}_C$  mindestens eine Kante des Kreises umgedreht wird.

Sei  $\vec{P}_C \subset \vec{E}_C$  Kantenmenge zu k s-t-Wege in  $\vec{G}_C$ . Konstruiere dazu Kantenmenge  $\vec{P}$  in  $\vec{G}$

$$\vec{P} := \text{siehe oben}$$

(todo: Bild)

### 5.2.6 Zu Schritt 3

Berechnung einer max. Anzahl s-t-Wege in  $G_C$  in  $O(n)$

- Schleife über ausgehende Kanten aus s
  - RIGHT-FIRST-DFS-Tiefensuche
  - Suchschritt: rechteste freie auslaufende Kante in Bezug auf Referenzkante
    - 2 Varianten welche "Referenzkante"
    - \* **Weihe**: aktuell einlaufende Kante
    - \* **Coupry**: erste einlaufende Kante

### Korrektheitsbeweis zu Schritt 3

Zu beweisen:  $\vec{P}_C$  enthält maximale Anzahl s-t-Wege. Benutze dazu gewichtete Variante des Satz von Menger, d.h. konstruiere s-t-Schnitt, der entsprechende Kapazität hat.

Schnitt A wird induziert durch geeigneten Kreis  $\vec{K}$  in  $\vec{G}_C$  mit folgenden Eigenschaften:

- $s \in \text{Inneres}(\vec{K})$  oder auf  $\vec{K}$
- $t \in \text{Äußeres}(\vec{K})$
- $|A := \{(u, v) \in \vec{E}_C : u \text{ auf } \vec{K}, v \in \text{Äußeres}(\vec{K})\}|$  ist gleich #s-t-Wege in  $\vec{P}_C$

$\vec{K}$  wird mittels LEFT-FIRST Rückwärtssuche von s aus in  $\vec{P}_C$  konstruiert

(todo: Bild: Kreis mit t ausserhalb: Varianten wie K aussieht: 1. S liegt auf Kreis, 2. S liegt in Kreis)

### Lemma

Betrachte  $\vec{G}_C = (V, \vec{E}_C)$  und  $\vec{K}$ , dann ist jede Kante  $(u, v) \in \vec{E}_C$  mit u auf  $\vec{K}$  und  $v \in \text{Äußeres}(\vec{K})$  durch einen s-t-Wege aus  $\vec{P}_C$  besetzt

### Beweis

- wenn  $p_1, \dots, p_l$  s-t-Wege und linkskreise von s nach s. Dann gehört keine der Kanten  $(X, Y), X \in \text{Äußeres}(\vec{K})$  und y auf  $\vec{K}$  zu einem der  $p_i$  (todo: bild)

wegen Left-First in Graph induziert durch  $p_1, \dots, p_l$   $(u, y) \notin p_i$  für alle  $1 \leq i \leq l$   
 $\Rightarrow$  für  $(u, v)$  mit u auf  $\vec{K}, v \in \text{Äußeres}(\vec{K})$  kann nicht auf einem Linkskreis aus  $p_1, \dots, p_l$  liegen.

- betrachte  $(u, v)$  mit u auf  $\vec{K}, v \in \text{Äußeres}(\vec{K})$  und  $(u, w)$  mit w auf  $\vec{K}$  (todo: bild)

Annahme:  $(u, v)$  gehört zu keinem der  $p_1, \dots, p_l$

betrachte Referenzkante zu  $(u, w)$  in RIGHT-FIRST-SUCHE (Schritt 3)



Referenzkante “liegt wie” (*todo: grün*) oder (*todo: blau*), aber dann hätte RIGHT-FIRST *nicht*  $(u,w)$  sondern  $(u,v) \Rightarrow$  Widerspruch

### 5.3 Das knotendisjunkte Menger-Problem

geg.  $G = (V, E)$ ,  $s, t \in V$ . Finde eine maximale Anzahl paarweise (intern) knotendisj. s-t-Wege in  $G$ .

#### Satz von Menger

maximale Anzahl knotendisj. s-t-Wege ist gleich der Größe eines minimalen Separators, der  $s$  und  $t$  trennt.

Linearzeitalgorithmus für knotendisjunktes Menger-Problem in planaren Graphen basiert auf RIGHT-FIRST-Suche.

#### Schritt 1

überführe  $G = (V, E)$  in gerichteten Graph  $\vec{G} = (V, \vec{E})$  indem alle Kanten mit  $u, v \in V \setminus \{s, t\}$  durch zwei gerichtete Kanten ersetzt werden. Kanten  $\{s, v\}$  werden nur durch  $(s, v)$  und  $\{v, t\}$  nur durch  $(v, t)$  ersetzt.

#### Schritt 2

Seien  $l_1, \dots, l_r \in E$  die aus  $s$  auslaufenden Kanten, führe Schleife über  $l_1, \dots, l_r$  aus, wobei im  $i$ -ten Durchlauf RIGHT-FIRST-Suche startend bei  $s$  mit  $l_i$  ausgeführt wird. Durchlauf endet in  $t$  oder in  $s$ .

RIGHT-FIRST-Suche kann zu “Konflikt” führen, d.h. Suche trifft auf einen bereits belegten Knoten

- belegter Knoten: (*todo: bild*)

zwei mögliche Konfliktsituationen:

- Konflikt von links,
- Konflikt von rechts

1) Konflikt von links (mit anderem Pfad oder selben Pfad früher)

$\Rightarrow$  BACKTRACK-REMOVE: ein Schritt zurück

2) Konflikt von rechts

- a) zwischen zwei Wegen, gefundener Weg und Suchweg: Vorangehende Wegabschnitte von gefundenem Weg und Suchweg werden vertauscht  $\Rightarrow$  Konflikt von links
- b) Suchweg trifft von rechts auf sich selbst

Trick: verhindere, dass ein Rechtskreis durchlaufen wird, indem nach vorherigen Durchlauf des entsprechenden Linkskreis geeignete Kante gelöscht wird.

Man kann beweisen, dass für Kante  $(v, w)$  die zu Rechtskreis mit Konflikt rechts in  $v$  führt zuvor umgekehrte Kante  $(w, v)$  in einem Linkskreis mit Konflikt von links in  $v$  durchlaufen wird.

$\rightsquigarrow$  lösche Kante  $(v, w)$  die nach RIGHT-FIRST-Auswahlregel nächste Kante wäre (\*) nachdem  $(w, v)$  zuvor von denselben Suchweg belegt wurde.

zweiter Trick: führe globalen Zähler ein, der um 1 erhöht wird, wenn neue Suche in  $s$  startet oder Wege umorganisiert werden wegen “Konflikt von rechts” setze Wert des “lokalen” Zählers um besuchten Knoten  $v$  auf Wert des globalen Zählers  $\rightsquigarrow$  Situation (\*) wird erkannt, da Wert des lokalen Zählers an  $v$  und  $w$  gleich.

## Laufzeit

$O(n)$  klar, da jede Kante höchstens einmal besucht wird und RIGHT-FIRST-Auswahlschritt in  $O(1)$ ; Referenzkante zu Knoten  $v$  bleibt gleich.

## Korrektheit

wichtigster Teil ist Beweis, dass “gelöschte” Kante wegen “Backtrack-Remove” oder wegen (\*) für Lösung nicht benötigt wird.

Sei  $k$  maximale Anzahl s-t-Wege in  $G$  und  $a_1, \dots, a_n$  gelöschte Kanten (in dieser Reihenfolge).

Beweise per Induktion über  $a_1 \dots a_n$  dass  $G_i := G \setminus \{a_1, \dots, a_n\}$  ebenfalls  $k$  s-t-Wege enthält.

## 6 Das Problem von Okamura und Seymour

### 6.1 Kantendisjunktes Wegpackungsproblem

Gegeben  $G = (V, E)$  und Paare ausgezeichneter Knoten  $\{s_1, t_1\}, \dots, \{s_k, t_k\}, s_i, t_i \in V$

Finde paarweise kantendisjunkte  $s_i - t_i$ -Wege.

NP-vollständig, sogar wenn  $G$  planar. Mögliche Spezialfälle:

- $G$  planar und  $s_i, t_i$  für alle  $1 \leq i \leq k$  auf dem Rand derselben Facette
- $G$  planar und sogar  $G := (V, E \cup D)$  wobei  $D := \{\{s_i, t_i\} : s_i, t_i \text{ Terminale}\}$  planar

$D$  für "Demand-Kanten"

$G = (V, E), D = \{\{s_i, t_i\} : 1 \leq i \leq k, s_i, t_i \in V\}, X \subseteq V$ . Dann heißt

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

**Kapazität** von  $X$  (bzw. Kantenschnitt induziert durch  $X$ ).

und

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

**Dichte** von  $X$ .

$\text{fcap}(X) := \text{cap}(X) - \text{dens}(X)$  **freie Kapazität** von  $X$ .

Offensichtlich  $\text{fcap}(X) \geq 0$  für alle  $X \subseteq V$  notwendige Bedingung für Lösbarkeit.

(todo: Bild)

**Kapazitätsbedingung**  $\text{fcap}(X) \geq 0$  für alle Schnitte  $X \subseteq V$  ist nicht hinreichend für Lösbarkeit!

**Geradheitsbedingung** Für alle  $X \subseteq V$   $\text{fcap}(X)$  gerade ist weitere Bedingung an Instanz

(Beginn Vorlesung 2015-06-16)

#### Satz von Okamura und Seymour

Falls Geradheitsbedingung erfüllt, so ist Kapazitätsbedingung hinreichend für Lösbarkeit

## Lemma

Es gilt  $fcap(X)$  gerade für alle  $X \subseteq V$  g.d.w.  $fcap(v)$  gerade für alle  $v \in V$ .

$$fcap(\{v\}) = \frac{d(v) - dens(v)}{\text{Grad} \quad \# \text{Terminale auf } v}$$

## Beweis

$\Rightarrow$  trivial

$\Leftarrow$ : sei  $fcap(v)$  gerade für alle  $v \in V$ . Betrachte  $X \subseteq V$ :

$$cap(X) = \sum_{v \in X} cap(v) - 2 \cdot |\{\{u, v\} \in E : u, v \in X\}|$$

$$dens(X) = \sum_{v \in X} dens(v) - 2 \cdot |\{\{s_i, t_i\} \in D : s_i, t_i \in X\}|$$

$$fcap(X) = \sum_{v \in X} cap(v) - \sum_{v \in X} dens(v) - 2 \cdot |\{\{u, v\} \in E : u, v \in X\}| + 2 \cdot |\{\{s_i, t_i\} \in D : s_i, t_i \in X\}|$$

$\rightarrow fcap(X)$  gerade

Geradheitsbedingung überprüfen: Es reicht für alle  $v \in V$  den Grad von  $v$  und  $\# s_i, t_i$  auf  $V$  zu zählen.

Linearzeitalgorithmus für  $G = (V, E)$  pl., Terminalpaare  $D$  auf äußerer Facette und Geradheitsbedingung erfüllt.

### 6.1.1 2 Phasen

1. Konstruiere aus  $G, D$  "einfache Instanz" mit "Klammerstruktur" und berechne mittels RIGHT-FIRST Tiefensuche Lösungswege  $q_1, \dots, q_k$ . Diese induzieren gerichteten "Hilfsgraph".
2. Berechne mittels gerichteter RIGHT-FIRST Tiefensuche in Hilfsgraph Lösungswege  $p_1 \dots p_k$  die jeweils  $s_i$  mit  $t_i$  verbinden.

### 6.1.2 Instanz mit Klammerstruktur

$G, D = \{\{s_i, t_i\} : s_i, t_i \in V\}$  alle auf äußerer Facette

(todo: bild bild)

daraus Instanz mit Klammerstruktur konstruieren:

- wähle beliebiges Terminal als "Startterminal"  $s$

- gehe im Gegenuhrzeigersinn startend von  $s$  um äußere Facette und ordne erste Terminal aus einem Paar  $\{s_i, t_i\}$  eine "(" zu und zweitem Auftreten eine ")"

$\rightsquigarrow$  Klammerausdruck; paare Terminal entsprechend Klammerausdruck

$\rightsquigarrow$  Instanz  $G = (V, E), D' = \{\{s'_i, t'_i\} : s'_i, t'_i \in V\}$  entsprechende Klammerung

Konstruiere mittels RF-Suche Lösung  $q_1 \dots q_k$  zu  $G, D'$ ; Reihenfolge in der Wege bezeichnet werden nach Reihenfolge der  $t'_i$ , d.h. "von innen nach außen" in Klammerstruktur

### 6.1.3 Korrektheit von Phase 1

#### Beobachtung

- keine zwei Wege  $g_i, g_j$  kreuzen sich und
- kein  $g_i$  kreuzt sich selbst (wegen RF-Auswahlregel)

Sei  $\vec{G} = (V, \vec{E})$  Graph der durch  $q_1 \dots q_k$  induziert wird. Dann enthält  $\vec{G}$  keinen Rechtskreis. (*todo: bild*)  $\Rightarrow$  induktiv über  $q_i, 1 \leq i \leq k$  kann gezeigt werden, dass  $q'_i$  die "richtigen Terminale" verbindet.

- Phase in  $O(n)$  klar. in

(Beginn Vorlesung 2015-06-17)

## 6.2 Zu Phase 2

Sei von Startterminal im Gegenuhrzeigersinn jeweils  $s_i$  vor  $t_i$  und Indizierung entsprechend Auftreten der  $t_i$ .

Für  $i = 1$  bis  $k$

RFS von  $s_i$  aus bis zu einem  $t_j$  }  $\rightsquigarrow p_i$

falls  $j \neq i$  stop

gib  $p_1, \dots, p_k$  aus

#### Laufzeit

$O(n)$  amortisiert mit UNION-FIND wie kantendisj. Menger

## Korrektheit

Algorithmus findet entweder

- i) mit Wegen  $p_i \dots p_k$  die jeweils  $s_i$  mit  $t_i$  verbinden oder
- ii)  $p_1, \dots, p_{i-1}$  "korrekt" und  $p_i$  verbindet  $s_i$  mit  $t_j, i \neq j \implies i < j$

$\rightsquigarrow$  Prozedur, die Weg  $p$  berechnet, so dass  $p$  einen Schnitt  $X$  induziert, der

- im Fall i) "saturiert" ist, d.h.  $fcap(X) = 0$
- im Fall ii) "übersaturiert" ist, d.h.  $fcap(X) < 0$

$\implies$  Korrektheit

Prozedur für Schritt X:

Rückwärts-LFS startend von Terminal  $t_i$  bzw.  $t_j$  wo Weg  $p_i$  endet in Graph, der durch  $p_1 \dots p_i$  induziert wird.

## Lemma

Sei  $A$  Menge der Kanten  $\{u, v\}$  aus  $G$  mit  $u$  auf  $p$  und  $v$  rechts von  $p$ . Jede Kante  $\{u, v\} \in A$  gehört zu  $\vec{G}$  und genau dann in Orientierung  $(u, v)$  wenn sie durch einen der  $p_1 \dots p_i$  besetzt ist.

## Beweis

Wenn  $\{u, v\}$  durch ein  $p_i$  besetzt, dann wegen Konstruktion von  $p$  in Orientierung  $(u, v)$ .

*Fall 1:* es existiere  $(u, v)$  mit  $(u, v)$  nicht durch  $p_1 \dots p_i$  besetzt. (*todo: bild*) Widerspruch zu RFS in Phase 2

*Fall 2:* es existiert  $\{u, v\} \in A, (u, v), (v, u) \notin \vec{G} \implies$  Widerspruch zu RFS in Phase 1

## Lemma

Sei  $X$  Schnitt induziert durch  $p$  (Knoten rechts von  $p$ ). Falls  $p_i \ s_i - t_i - \text{Weg}$ , so ist  $fcap(X) = 0$  ansonsten  $fcap(X) < 0$ .

## Beweis

Kanten  $\{u, v\}$  auf  $p$ ,  $v$  rechts von  $p$  entweder zu Weg  $p_j$  gehört mit  $1 \leq j \leq i, s_j \in V \setminus X$  und  $t_j \in X$  oder zu Weg  $q_j$  aus erster Phase mit  $s'_j \in X$  und  $t'_j \in V \setminus X$ .

Wenn  $p_i$  korrekter  $s_i - t_i$ -Weg so gibt

$$cap(X) = |\{\{s_j, t_j\} : s_j \in V \setminus X, t_j \in X, 1 \leq j \leq i\}| + |\{\{s'_j, t'_j\} : s'_j \in X\}|$$

und  $t'_j \in V \setminus X, \{s'_i, t'_j\} \notin \{\{s_1, t_1\}, \dots, \{s_{i-1}, t_{i-1}\}\} = dens(X)$

Wenn  $p_i$  nicht korrekt, d.h.  $s_i$  mit  $t_j, i < j$  verbindet, so ist  $cap(X) < dens(X)$ , da  $s_i \in V \setminus X, t_i \in X$ .

## 7 Planartest in Linearzeit

siehe Folien

### 7.1 Beweis zum Lemma defekte Kanten bilden Pfad

- 1) es existiert Knoten mit  $\geq 3$  defekte Kanten  
     $\leadsto$  3 Rot-Schwarz-Wechsel  $\Rightarrow$  Widerspruch zu rot-konsekutiv
- 2) Wenn Pfad zwischen zwei defekten Kanten existiert, sind alle Kanten auf Pfad defekt: klar, weil äußerer Teil Teilbaum von innerem Teil ist