Algorithmen für Planare Graphen

Vorlesung am 28.06.2022

Torsten Ueckerdt | 28. Juni 2022

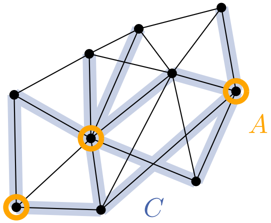
Wiederholung: Mixed-Max-Cut

Definition.

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

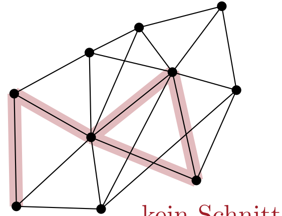
$$C = \{uv \in E \colon |A \cap \{u, v\}| = 1\}$$

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



Problem MIXED-MAX-CUT.

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



Mixed-Max-Cut – Überblick

Eingabe: $G_0 = (V, E_0)$ planar.

1. $\downarrow$

Trianguliere zu $G = (V, E)$.

2. $\downarrow$

Dualisiere zu $G^* = (F, E^*)$.

3. $\downarrow$

Modifiziere zu $G' = (V', E')$.

4. $\searrow$

Ausgabe: $C_0 = C \cap E_0$ Mixed-Max-Cut in G_0.

$\uparrow$

$C = (C^*)^*$ Mixed-Max-Cut in G.

$\uparrow$

$C^* = C' \cap E^*$ gew.max. gerade Menge in G^*

$\uparrow$

$C' = E' - M$ gew.max. 2-Faktor in G'

$\nearrow$

Berechne ein gewichtsminimales perfektes Matching M in G'.

Mixed-Max-Cut – Überblick

Eingabe: $G_0 = (V, E_0)$ planar.

1. $\downarrow$

Trianguliere zu $G = (V, E)$.

2. $\downarrow$

Dualisiere zu $G^* = (F, E^*)$.

3. $\downarrow$

Modifiziere zu $G' = (V', E')$.

4. $\searrow$

Berechne ein gewichtsminimales perfektes Matching M in G'.

$\nearrow$

$C' = E' - M$ gew.max. 2-Faktor in G'

$\uparrow$

$C^* = C' \cap E^*$ gew.max. gerade Menge in G^*

$\uparrow$

$C = (C^*)^*$ Mixed-Max-Cut in G.

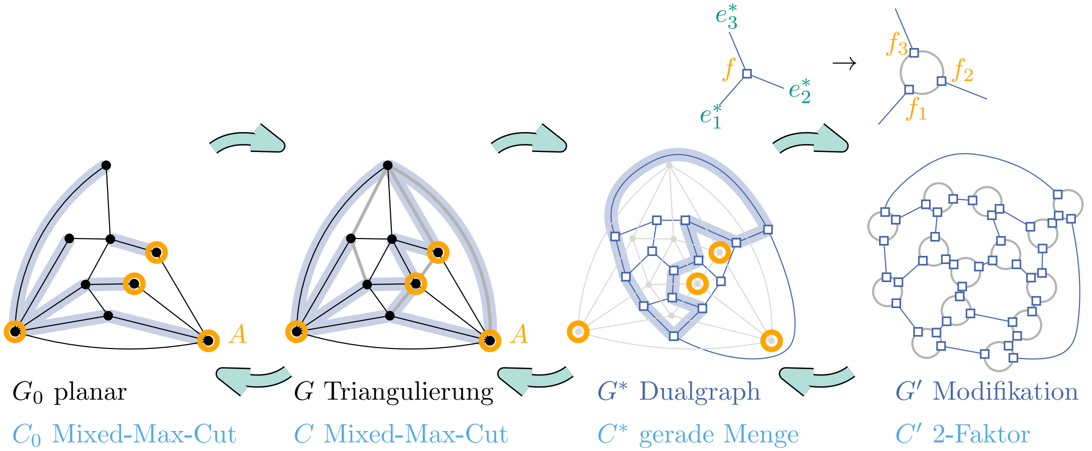
$\uparrow$

Ausgabe: $C_0 = C \cap E_0$ Mixed-Max-Cut in G_0.

Fragen für heute:

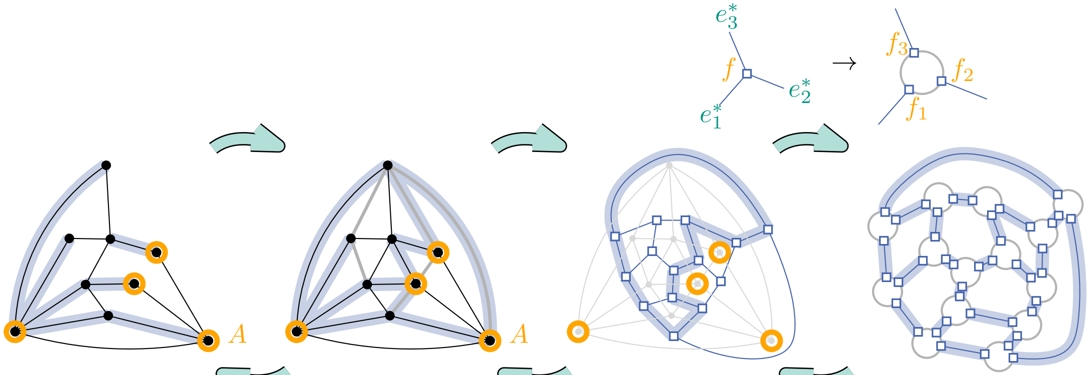
- Wie berechnen wir M? (Existenz?)
- Was ist die Laufzeit?
- Warum ist $C_0 \neq \emptyset$?

Existenz



e_3^*
f
e_2^*
e_1^*
$\rightarrow$
f_3
f_2
f_1
A
A
G_0 planar
C_0 Mixed-Max-Cut
G Triangulierung
C Mixed-Max-Cut
G^* Dualgraph
C^* gerade Menge
G' Modifikation
C' 2-Faktor

Existenz



G_0 planar
C_0 Mixed-Max-Cut

G Triangulierung
C Mixed-Max-Cut

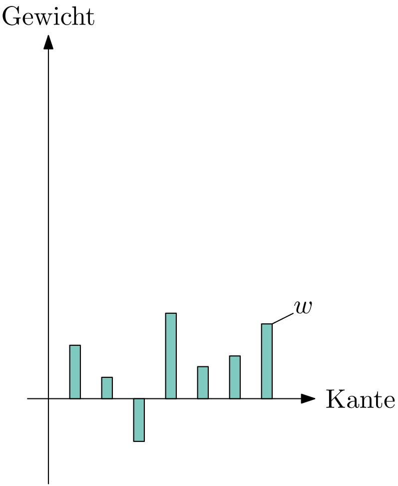
G^* Dualgraph
C^* gerade Menge

G' Modifikation
C' 2-Faktor

Berechnung und Laufzeit

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

Beweis: Reduziere auf gewichtsmaximales Matching.



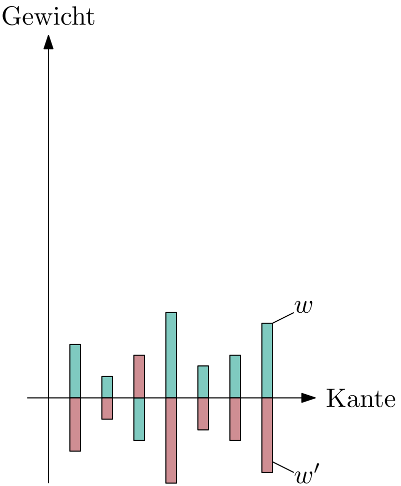
Berechnung und Laufzeit

Satz.

In planaren Graphen können gewichtsminimale perfekte Matchings in $\mathcal{O}(n^{\frac{3}{2}})$ berechnet werden.

Beweis: Reduziere auf gewichtsmaximales Matching.

- $w'(e) := -w(e)$.
  - Maximal bezüglich w' $\Leftrightarrow$ minimal bezüglich w.

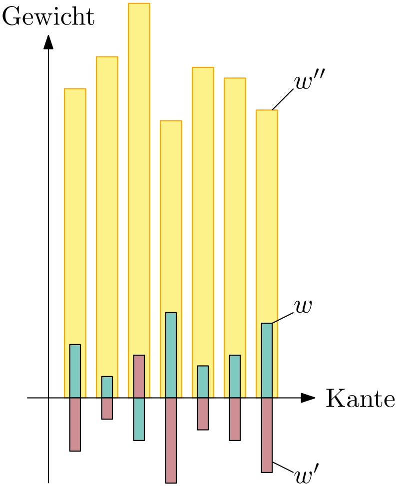


Berechnung und Laufzeit

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

Beweis: Reduziere auf gewichtsmaximales Matching.

- $w'(e) := -w(e)$.
  - Maximal bezüglich w' $\Leftrightarrow$ minimal bezüglich w.
- $w''(e) := W + w'(e)$ für $W > |V| \cdot \max_{e \in E'}(|w'(e)|)$.
  - Max. Matching bzgl. w'' hat die größtmögliche Anzahl Kanten.
  - Damit sind gewichtsmaximale Matchings bzgl. w'' perfekt.
- Max. Matchings bzgl. w'' entsprechen also genau den min. perfekten Matchings bezüglich w.



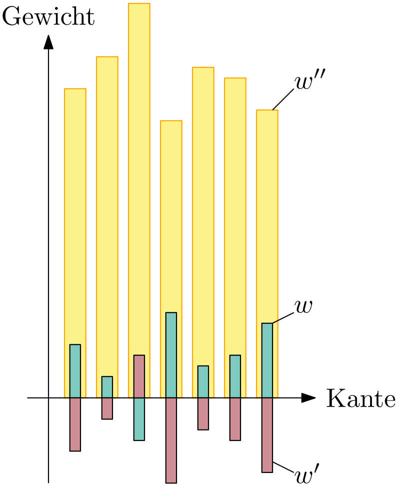
Berechnung und Laufzeit

Satz.

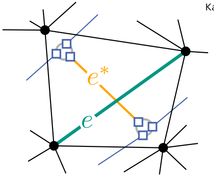
In planaren Graphen können gewichtsminimale perfekte Matchings in $\mathcal{O}(n^{\frac{3}{2}})$ berechnet werden.

Beweis: Reduziere auf gewichtsmaximales Matching.

- $w'(e) := -w(e)$.
  - Maximal bezüglich w' $\Leftrightarrow$ minimal bezüglich w.
- $w''(e) := W + w'(e)$ für $W > |V| \cdot \max_{e \in E'}(|w'(e)|)$.
  - Max. Matching bzgl. w'' hat die größtmögliche Anzahl Kanten.
  - Damit sind gewichtsmaximale Matchings bzgl. w'' perfekt.
- Max. Matchings bzgl. w'' entsprechen also genau den min. perfekten Matchings bezüglich w.
- Fertig, da MAX MATCHING in $\mathcal{O}(n^{\frac{3}{2}})$. ✓



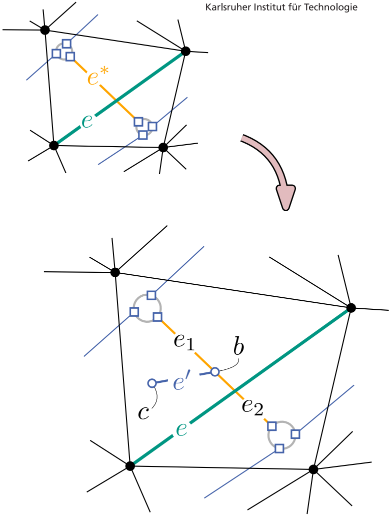
$C_0 \neq \emptyset$ sicherstellen



- Für eine Kante $e \in E_0$ wollen wir erzwingen, dass $e \in C_0$.
- Also soll e^* nicht in M' sein:
  - Das kann man durch lokale Modifikation erreichen.
  - Unterteile e^* mit Knoten b (dadurch entstehen e_1, e_2).
  - $w(e_1) = w(e^*), w(e_2) = 0$.
  - Füge Kante $e' = bc$ mit neuem Knoten c hinzu.
  - $w(e') = 0$.
  - Jedes perfekte Matching muss dann $e' = bc$ enthalten.
- Dadurch wird $e \in C_0$ garantiert. ✓

$C_0 \neq \emptyset$ sicherstellen

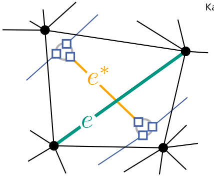
- Für eine Kante $e \in E_0$ wollen wir erzwingen, dass $e \in C_0$.
- Also soll e^* nicht in M' sein:
  - Das kann man durch lokale Modifikation erreichen.
  - Unterteile e^* mit Knoten b (dadurch entstehen e_1, e_2).
  - $w(e_1) = w(e^*), w(e_2) = 0$.
  - Füge Kante $e' = bc$ mit neuem Knoten c hinzu.
  - $w(e') = 0$.
  - Jedes perfekte Matching muss dann $e' = bc$ enthalten.
- Dadurch wird $e \in C_0$ garantiert. ✓

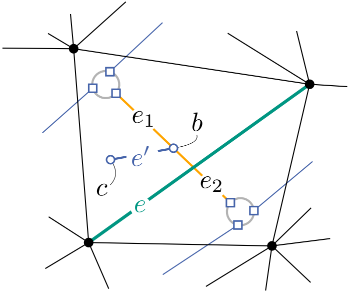


$C_0 \neq \emptyset$ sicherstellen

- Für eine Kante $e \in E_0$ wollen wir erzwingen, dass $e \in C_0$.
- Also soll e^* nicht in M' sein:
  - Das kann man durch lokale Modifikation erreichen.
  - Unterteile e^* mit Knoten b (dadurch entstehen e_1, e_2).
  - $w(e_1) = w(e^*), w(e_2) = 0$.
  - Füge Kante $e' = bc$ mit neuem Knoten c hinzu.
  - $w(e') = 0$.
  - Jedes perfekte Matching muss dann $e' = bc$ enthalten.
- Dadurch wird $e \in C_0$ garantiert. ✓

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





Mixed-Max-Cut – Überblick

Eingabe: $G_0 = (V, E_0)$ planar.

1\. $\downarrow$

Trianguliere zu $G = (V, E)$.

2\. $\downarrow$

Dualisiere zu $G^* = (F, E^*)$.

3\. $\downarrow$

Modifiziere zu $G' = (V', E')$.

4\. $\searrow$

Berechne ein gewichtsminimales perfektes Matching M in G'.

$\nearrow$

$C' = E' - M$ gew.max. 2-Faktor in G'

$\uparrow$

$C^* = C' \cap E^*$ gew.max. gerade Menge in G^*

$\uparrow$

$C = (C^*)^*$ Mixed-Max-Cut in G.

$\uparrow$

Ausgabe: $C_0 = C \cap E_0$ Mixed-Max-Cut in G_0.

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

Planarität testen

Karlsruher Institut für Technologie

Problem

Gegeben ein Graph G als Adjazenzliste, entscheide ob G planar ist.

Möglichkeit:

- Teste auf K_5- oder $K_{3,3}$-Unterteilung.
- Test auf H-Unterteilung von G:
  - Algorithmus von Karawabayashi-Reed (2012)
  - Laufzeit $\leq c(H) \cdot |V(G)|^2$
  - Theoretisch effizient aber nicht praktikabel, da $c(H) \approx 2 \uparrow\uparrow\uparrow |V(H)|$

Pfeil-Notation von Knuth:

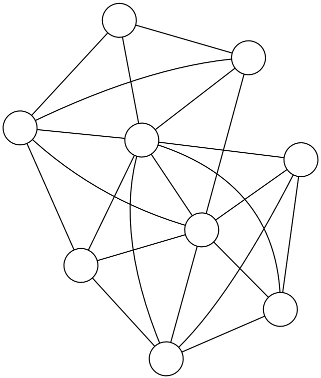
- $a \uparrow b := a^b$
- $$\underbrace{a \uparrow \cdots \uparrow}_{k\text{-fach}} b := \underbrace{a \underbrace{\uparrow \cdots \uparrow}_{k-1\text{-fach}} (\cdots (a \underbrace{\uparrow \cdots \uparrow}_{k-1\text{-fach}} a))}_{b\text{-fach}}$$

- Alle effizienten Planaritätstests sind nichttrivial.
- Lineare Laufzeit $O(|V(G)|)$ ist aber möglich.
- Hier **LR-Planarität**.

LR-Planarität

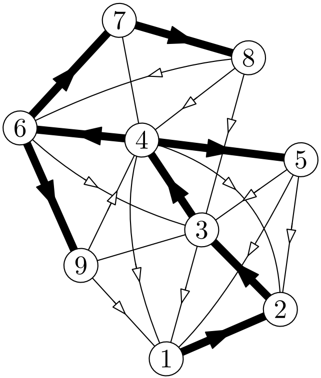
- Annahme:
  - G hat keine Mehrfachkanten.
  - G hat keine Schlingen.
  - G hat keine Brücken.

    $\Leftrightarrow$ Also liegt jede Kante auf einem Kreis.
  - G ist zusammenhängend.



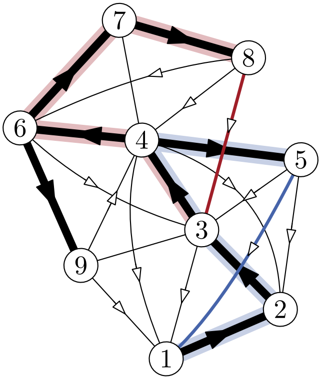
LR-Planarität

- Annahme:
  - G hat keine Mehrfachkanten.
  - G hat keine Schlingen.
  - G hat keine Brücken.
    - $\Leftrightarrow$ Also liegt jede Kante auf einem Kreis.
  - G ist zusammenhängend.
- Vorbereitendes Vorgehen:
  - Tiefensuche (DFS) von beliebigem Startknoten.
  - Nummeriere Knoten in Explorierungsreihenfolge.
  - Orientiere Baumkanten in Explorierungsreihenfolge.
  - Orientiere Nichtbaumkanten ebenfalls.



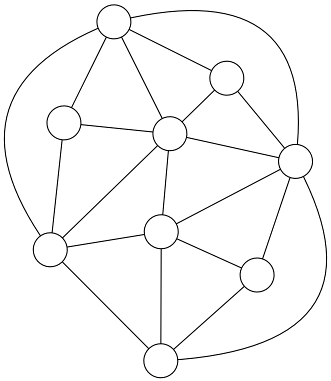
LR-Planarität

- Annahme:
  - G hat keine Mehrfachkanten.
  - G hat keine Schlingen.
  - G hat keine Brücken.
    - $\Leftrightarrow$ Also liegt jede Kante auf einem Kreis.
  - G ist zusammenhängend.
- Vorbereitendes Vorgehen:
  - Tiefensuche (DFS) von beliebigem Startknoten.
  - Nummeriere Knoten in Explorierungsreihenfolge.
  - Orientiere Baumkanten in Explorierungsreihenfolge.
  - Orientiere Nichtbaumkanten ebenfalls.
- Erste Beobachtungen:
  - Nichtbaumkanten verbinden immer zwei Knoten auf gemeinsamen Wurzel-Blatt-Pfad.
  - Jede Nichtbaumkante e schließt einen eindeutigen Kreis mit den Baumkanten (Fundamentalkreis zu e).



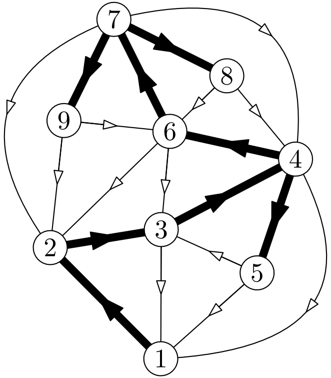
Was hilft der Baum?

- 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.
- Formal:
  - Ist der Fundamentalkreis von e im Uhrzeigersinn orientiert, so ist e R-Kante, ansonsten L-Kante, dieser Zeichnung.



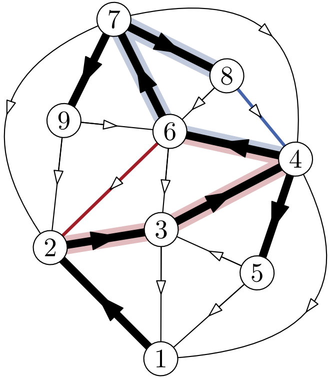
Was hilft der Baum?

- 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.
- Formal:
  - Ist der Fundamentalkreis von e im Uhrzeigersinn orientiert, so ist e *R*-Kante, ansonsten *L*-Kante, dieser Zeichnung.



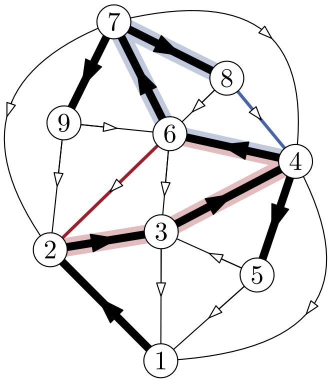
Was hilft der Baum?

- 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.
- Formal:
  - Ist der Fundamentalkreis von e im Uhrzeigersinn orientiert, so ist e R-Kante, ansonsten L-Kante, dieser Zeichnung.



Was hilft der Baum?

- 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.
- Formal:
  - Ist der Fundamentalkreis von e im Uhrzeigersinn orientiert, so ist e R-Kante, ansonsten L-Kante, dieser Zeichnung.
- Um herauszufinden, ob ein Graph G planar ist, müssen wir
  - entweder eine Einteilung der Nichtbaumkanten in L- und R-Kanten finden bei der keine Überschneidungen entstehen
    $\Rightarrow$ das werden wir LR-Zerlegung von G nennen
  - oder zeigen, dass eine solche Einteilung nicht existiert.



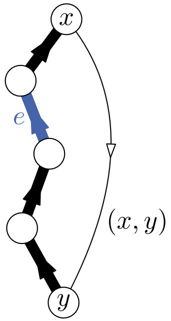
Rückkehr- und Tiefpunkte

Erinnerung: Wir betrachten einen DFS-Baum mit Orientierung.

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.



Rückkehr- und Tiefpunkte

Erinnerung: Wir betrachten einen DFS-Baum mit Orientierung.

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.

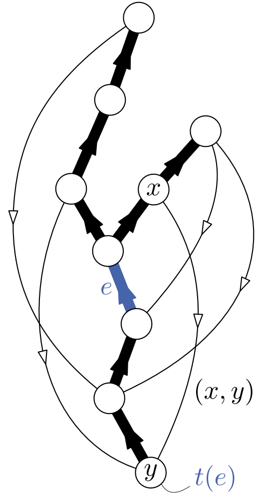
Ist e eine Nichtbaumkante, so ist e seine einzige und eigene Rückkante mit eindeutigem Rückkehrpunkt.

Definition.

Sei e eine Kante. Der tiefste Rückkehrpunkt von e

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

nennt man den **Tiefpunkt von** e.



Gabeln und Konflikte

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).