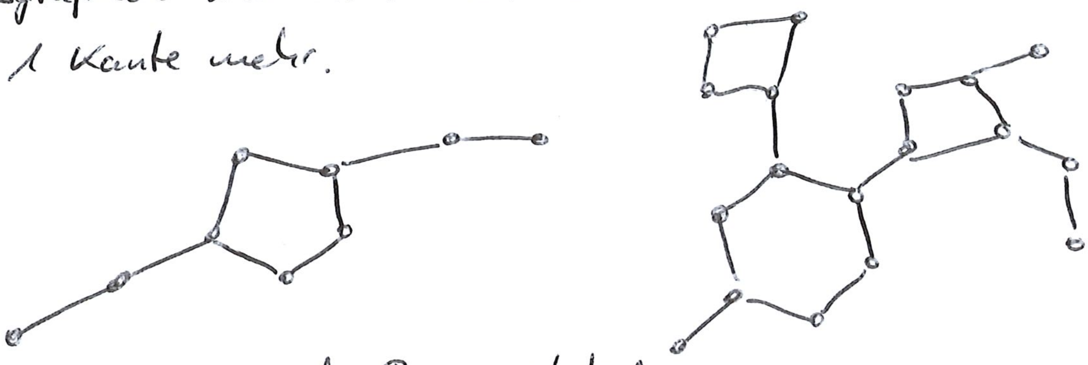
|VL 9|

Kapitel 5 – Tree Width

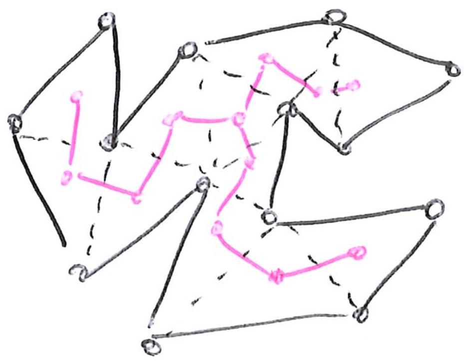
Bereits gesehen:

- Vertex Cover auf Bäumen ist einfach.
- auf Kreisgraphen ist einfach.
- NP schwer auf $\Delta \geq 3$ Graphen (nicht Bäume :))

Kreisgraphen sind nicht sehr anders als Bäume. Sie haben nur 1 Kante mehr.



sehr Baum-ähnlich



Auch wenn gestrichelte Kanten hinzugefügt werden, sieht das immer noch sehr baum-ähnlich aus.

Dazu Fragen:

- Kann man dies Ähnlichkeit messen?
- Können wir die Schwere von Problemen daran messen?
- Wie schwer ist es überhaupt diese Ähnlichkeit zu messen?

Definition 5.1 (Tree Decomposition)

Eine <u>Baumzerlegung</u> (Tree Decomposition) eines Graphen G ist ein Tupel $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$, wobei T ein Baum ist mit Knoten t, welche Teilmengen $X_t \subseteq V(G)$, ein "Bag" (Tasche), entsprechen. Dabei gelten 3 Punkte:

(T1) $\bigcup_{t \in V(T)} X_t = V(G)$

(T2) $\forall \{u,v\} \in E(G): \exists t \in V(T)$ mit $u,v \in X_t$

(T3) $\forall u \in V(G)$: $T[\{t \in V(T) \mid u \in X_t\}]$ ist zusammenhgd.

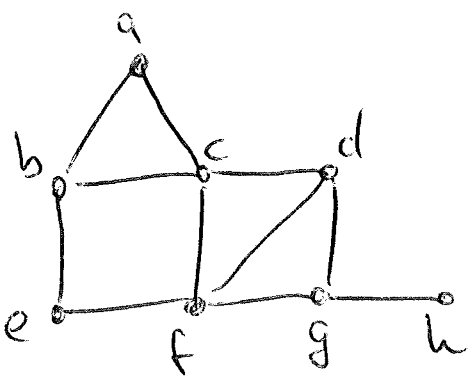
Die <u>Weite</u> (width) einer Baumzerlegung $\mathcal{T}$ ist

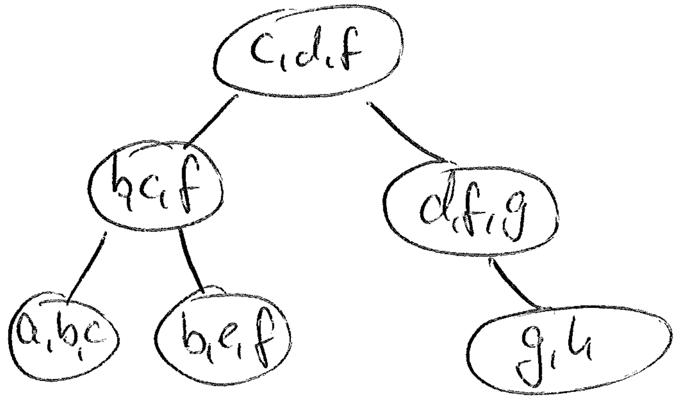
$$\text{width}(\mathcal{T}) := \max_{t \in V(T)} (|X_t| - 1)$$

Die <u>Baumweite</u> (tree width) eines Graphen G ist

$$tw(G) := \min_{\substack{\text{Baumzerlegung} \\ \mathcal{T} \text{ von } G}} (\text{width}(\mathcal{T}))$$

Beispiel:





$\Rightarrow tw(G) \leq 2$

③

Kann man die TW auch von unten beschränken?

Relativ einleuchtend:

$\forall v \in V$: $\cancel{\ }$ $tw(G) \geq tw(G - \{v\})$

$\forall e \in E$: $\cancel{\ }$ $tw(G) \geq tw(G - \{e\})$

Also: Teilgraphen können nicht plötzlich eine Größere TW besitzen; jede Tree Decomp für G ist auch eine TD für $\underbrace{G - \{v\}}_{\text{Lösche } v \text{ aus jedem Bag}}$ bzw $\underbrace{G - \{e\}}_{\text{tue nichts}}$.

Claim:
Sei $e \in E$. ~~Der~~ Für den Graph G/e nach Kontrahieren von e (verschmelze die Endpunkte) gilt:

$$tw(G) \geq tw(G/e)$$

Beweis:
Ersetze für $\{u,v\} = e$ jedes Vorkommen von u durch v.
$\Rightarrow$ Zusammenhang für jedes $w \in V$ bleibt erhalten, alle Kanten von G/e sind weiterhin abgedeckt. Bag-größe kann nicht größer werden. $\square$

(Sub)Graphen, die durch diese 3 Operation entstehen, heißen auch **Minor**.

Definition 5.2

Seien H und G Graphen. H ist ein <u>Minor</u> von G, wenn dieser durch wiederholtes Durchführen der folgenden Operationen entsteht:

1. Löschen eines Knotens
2. Löschen einer Kante
3. Kontrahieren einer Kante

Lemma 5.3

Sei G ein Graph und H ein Minor von G.
Dann gilt $tw(H) \leq tw(G)$.

Im vorherigen Beispiel ist $H = \triangle$ ein Minor.
Wenn wir zeigen können, das ~~H~~ $tw(H) \geq 2$ gilt, dann gilt $tw(G) = 2$. $\Rightarrow$ Hausaufgabe.

Wie hilft uns das jetzt beim Lösen von Problemen?
Die Hoffnung: Probleme sind einfacher zu lösen, wenn sie eine beschränkte Baumweite besitzen.
↳ TW als Parameter!

Dafür schaut man sich besondere "schöne" ("nice") Zerlegungen an:

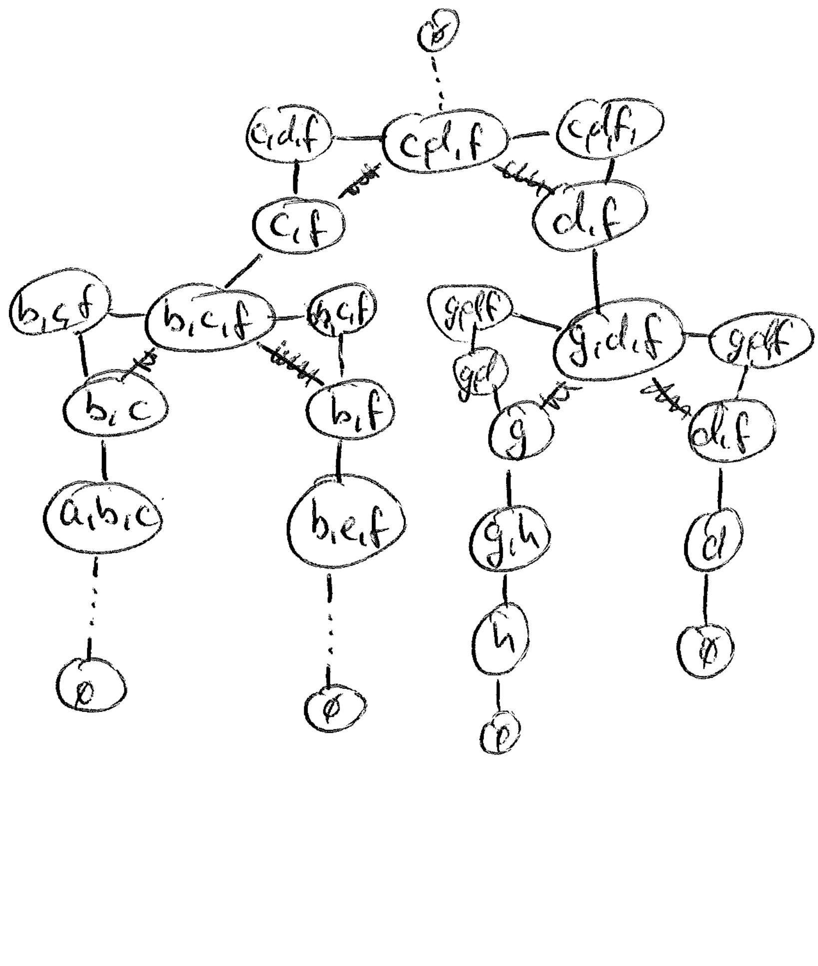
(5)

Definition 5.4

Eine gewurzelte Baumzerlegung heißt "nice", wenn folgendes gilt:

1. $X_r = \emptyset$ für die Wurzel // empty bags
   $X_\ell = \emptyset$ für jedes Blatt
2. Jeder Nicht-Blatt-Knoten ist einer der Typen:
   a) Einführungsknoten: Knoten t mit exakt einem Kind t' mit $X_t = X_{t'} \cup \{v\}$ für $v \notin X_{t'}$
   b) Vergessensknoten: Knoten t mit einem Kind t' mit $X_t = X_{t'} \setminus \{w\}$ für ein $w \in X_{t'}$
   c) Joinknoten: Knoten t mit zwei Kindern t_1, t_2 mit $X_t = X_{t_1} \overset{=}{\cup} X_{t_2}$

Beispiel mit vorherigem Graph



Lemma 5.5

Sei G ein Graph mit einer Baumzerlegung T mit $width(T) \leq k$. Dann kann eine nice Baumzerlegung T' in Zeit $O(k^2 \cdot max(|V(T)|, |V(G)|))$ mit $width(T') \leq k$ und $|V(T')| \in O(k \cdot |V(G)|)$.

Damit:

- Wenn t ~~$\in T$~~ ein Knoten in T, dann entspricht ~~$V(t_T)$~~ V_t allen Knoten von allen Bags im Subbaum.
- Der Unterschied zu einem der Kinder ist maximal ein ~~...~~ Knoten im Bag
- Je Bag gibt es 2^{k+1} Teilmengen.
- Definiere Lösung eines Subtrees über Lösungen dessen Subtrees.

Am Beispiel Vertex Cover

Sei $OPT_t(S)$ = Größe des kleinsten Vertex Covers in $G[V_t]$, mit S im Vertex Cover

Ist t ein Blatt, existiert nur $S = \emptyset$ und damit

$OPT_t(\emptyset) = 0$ für t Blatt. Rekursionsanfang

Betrachte also $t \neq$ Blatt.
Unterscheide:

1. t ist <u>Einführungsknoten</u>, d.h. $X_t = X_{t'} \cup \{v\}$.

Für jedes $S \subseteq X_t$ ist nun

v soll im VC sein.

$$\cancel{DP}\, OPT_t(S) = \begin{cases} \cancel{DP}\, OPT_{t'}(S \setminus \{v\}) + 1, & \text{falls } v \in S \\ OPT_{t'}(S), & \text{falls } \forall u \in X_t: \{u,v\} \in E \Rightarrow u \in S \\ \infty, & \text{sonst} \end{cases}$$

v soll nicht im VC sein

ungültiges S

2. t ist <u>Vergessensknoten</u>, d.h. $X_t = X_{t'} \setminus \{v\}$

Für jedes $S \subseteq X_t$ ist nun

$$OPT_t(S) = \min(\underbrace{OPT_{t'}(S)}_{v \text{ nicht in VC}}, \underbrace{OPT_{t'}(S \cup \{v\})}_{v \text{ in das VC}})$$

3. t ist Join-Knoten, d.h. $X_t = X_{t_1} = X_{t_2}$

Für jedes $S \subseteq X_t$ ist nun

$$OPT_t(S) = \cancel{DP}\, OPT_{t_1}(S) + OPT_{t_2}(S) \underbrace{- |S|}_{\text{wird sonst doppelt gezählt.}}$$

Mit bottom-up-Ansatz (Löse zuerst Blätter, dann bis hoch zur Wurzel) lässt sich das ohne Rekursion berechnen.

Die Laufzeit ist dann prinzipell:

$$2^{k+1} k \cdot \#\text{Knoten} \in O(2^{k+1} k^2 n)$$

$\Rightarrow$ Vertex Cover parametrisiert nach Baumweite ist in FPT.

Da ein Independent Set einfach $V(G) \setminus VC$ für min Vertex Cover VC ist, gilt auch: Independent Set parametrisiert nach Baumweite ist in FPT.

Problem: Baumweite zu bestimmen ist NP-schwer.

Aber: Baumweite ist in FPT und approximierbar

Lösung ist höchstens Konstanten Faktor schlechter als OPT

↳ Daher relaxiert man in der Theorie in der Regel, dass der Parameter in poly. Zeit berechenbar sein muss und sagt, der ist einfach gegeben.