Graphalgorithmen

Johannes Köbler

Institut für Informatik
Humboldt-Universität zu Berlin

SS 2021

Baum- und Pfadweite

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

- Eine Baumzerlegung (kurz TD für tree decomposition) von G ist ein Tripel (V_T, E_T, X), wobei $T = (V_T, E_T)$ ein Baum ist und die Abbildung $X : V_T \to \mathcal{P}(V)$ die folgenden 3 Eigenschaften erfüllt (für (V_T, E_T, X) schreiben wir meist (T, X) und für $X(t)$ meist X_t):
  - Es gilt $V = \bigcup_{t \in V_T} X_t$ (die Mengen $X_t \subseteq V$ heißen Taschen)
  - Für jede Kante $e \in E$ gibt es eine Tasche X_t mit $e \subseteq X_t$
  - Für jeden Knoten $u \in V$ ist der induzierte Teilgraph $T[X^{-1}(u)]$ von T zusammenhängend, wobei $X^{-1}(u) = \{t \in V_T \mid u \in X_t\}$ ist
- Die Weite von (T, X) ist $w(T, X) = \max_{t \in V_T} |X_t| - 1$
- Eine TD (T, X) von G heißt Pfadzerlegung (kurz PD für path decomposition), wenn T ein Pfad ist
- Die Baumweite $tw(G)$ (bzw. Pfadweite $pw(G)$) von G ist die kleinste Weite aller möglichen Baumzerlegungen (bzw. Pfadzerlegungen) von G
- Zudem ist $TW(k) = \{G \mid tw(G) \leq k\}$ und $PW(k) = \{G \mid pw(G) \leq k\}$

Baum- und Pfadweite

Beispiel

- Der leere Graph hat Baum- und Pfadweite $tw(E_n) = pw(E_n) = 0$
- Wir generieren für jeden Knoten $u \in V$ eine Tasche $X_u = \{u\}$, die nur diesen Knoten enthält, und verbinden diese Taschen zu einem Pfad
- Umgekehrt muss die Kantenmenge jedes Graphen G mit $tw(G) = 0$ leer sein, da jede Tasche nur einen Knoten enthält
- $TW(0) = PW(0)$ besteht also aus allen leeren Graphen

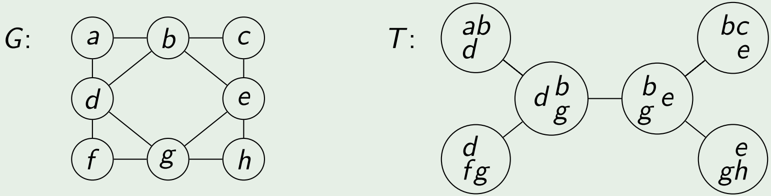
Baum- und Pfadweite

Beispiel

- Jeder Baum $G = (V, E)$ hat eine Baumweite $tw(G) \leq 1$
- Zum Beispiel hat G die TD (T, X) mit $V_T = E \cup \{\{u\} \mid u \in V\}$, $X_t = t$ für $t \in V_T$ und $E_T = \{\{s, t\} \mid s \subset t\}$ mit Weite $w(T, X) \leq 1$
- Zudem können wir aus den TDs (T, X) und (T', X') von zwei knotendisjunkten Graphen G und G' der Weite k und k' wie folgt eine TD von $G_1 \cup G_2$ der Weite $\max\{k, k'\}$ erhalten
- Wir verbinden die beiden Bäume T und T' durch eine beliebige Kante zu einem Baum und lassen die Taschen unverändert
- Somit hat auch jeder Wald eine Baumweite ≤ 1

Beispiel

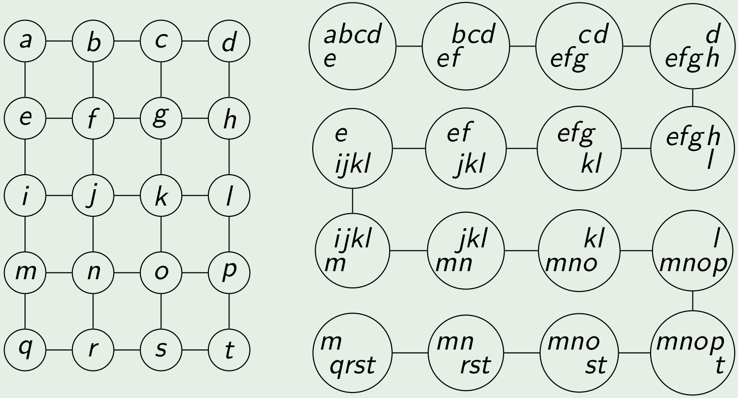
- Folgender Graph G hat eine TD (T, X) der Weite 2:



- Der Baum $T = (\{1, \ldots, 6\}, E_T)$ verbindet die Taschen $X_1 = \{a, b, d\}$, $X_2 = \{b, d, g\}$, $X_3 = \{b, e, g\}$, $X_4 = \{b, e, c\}$, $X_5 = \{e, g, h\}$, $X_6 = \{d, f, g\}$ durch die Kanten in $E_T = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{3, 5\}, \{2, 6\}\}$

Beispiel

- Für den Gittergraphen $G_{k\times\ell}$ mit $k\ell$ Knoten



gilt $tw(G_{k\times\ell}) \leq pw(G_{k\times\ell}) \leq \min\{k,\ell\}$ ◁

Baum- und Pfadweite

Bemerkung

Das Problem, für einen gegebenen Graphen und eine Zahl $w \in \mathbb{N}$ zu prüfen, ob $tw(G) \leq w$ ist, ist NP-vollständig

Satz

Für jeden Graphen G lässt sich eine TD der Weite $k = tw(G)$ in Zeit $2^{O(k^3)}n$ berechnen (ohne Beweis)

Proposition

- Sei $G = (V, E)$ ein Graph und sei $H = (V', E')$ ein Teilgraph von G oder $H = G_{uv}$ für eine Kante $\{u, v\} \in E$
- Dann gilt $tw(H) \leq tw(G)$ und $pw(H) \leq pw(G)$

Beweis.

- Sei (T, X) eine Baum- bzw. Pfadzerlegung von G
- Dann ist (T, X') mit $X'_t = X_t \cap V'$ eine Baum- bzw. Pfadzerlegung von (V', E') mit $w(T, X') \leq w(T, X)$
- Zudem ist (T, X'') mit

$$X''_t = \begin{cases} X_t, & v \notin X_t \\ (X_t \setminus \{v\}) \cup \{u\}, & v \in X_t \end{cases}$$

eine Baum- bzw. Pfadzerlegung von G_{uv} mit $w(T, X'') \leq w(T, X)$ □

Baum- und Pfadweite

Korollar

Die Graphklassen $TW(k)$ und $PW(k)$ sind unter Minorenbildung abgeschlossen

Definition

Eine TD (T, X) heißt kompakt, wenn alle Taschen paarweise unvergleichbar sind, d.h. für alle $s \neq t \in V_T$ gilt $X_s \not\subseteq X_t$ und $X_t \not\subseteq X_s$

- Aufgrund der Definition von Baumzerlegungen gilt im Fall $X_s \subseteq X_t$ für alle Knoten t' auf dem Pfad von s nach t ebenfalls $X_s \subseteq X_{t'}$
- Daher ist (T, X) genau dann kompakt, wenn für alle $\{s, t\} \in E_T$ die Bedingungen $X_s \not\subseteq X_t$ und $X_t \not\subseteq X_s$ gelten
- Es genügt also, die Taschen von benachbarten Baumknoten zu vergleichen

Baum- und Pfadweite

Proposition. Sei $G = (V, E)$ ein Graph.

1. Jede TD (T, X) von G lässt sich effizient in eine kompakte TD (T', X') von G transformieren, die nur Taschen aus X enthält
2. Für jede kompakte TD (T, X) von G gilt $n(T) \leq n(G)$

Beweis.

1. Der Algorithmus durchsucht T ausgehend von einem beliebigen Startknoten
   - Dabei prüft er für jede Kante $\{s, t\} \in E_T$ (wobei s näher am Startknoten liege als t), ob $X_s \subseteq X_t$ oder $X_s \supseteq X_t$ gilt
   - Falls ja, kontrahiert er die Kante $\{s, t\}$ zu s und ersetzt X_s durch $X_s \cup X_t \in \{X_s, X_t\}$
   - Danach setzt er die Suche im Baum (bzw. Pfad) T_{st} mit s als aktuellem Knoten fort
   - Dieser Algorithmus benötigt $O(w(T, X) \cdot n(T))$ Zeit, um (T, X) in eine kompakte Baum- bzw. Pfadzerlegung zu transformieren

Beweis (Fortsetzung).

2. Falls (T, X) eine kompakte TD von $G = (V, E)$ ist, können wir wie folgt eine injektive Abbildung $f : V_T \to V$ definieren, die jedem Baumknoten $t \in V_T$ einen Knoten $v_t \in X_t$ zuordnet

- Beginnend mit $D = \emptyset$ wählen wir ein beliebiges Blatt t in dem Baum $T - D$
- Da (T, X) kompakt ist, enthält die Tasche X_t einen Knoten v_t, der nicht in der Tasche X_s des Nachbarn s von t in $T - D$ vorkommt
- Da dann v_t auch in keiner anderen Tasche eines Knotens in $T - D$ vorkommen kann (sonst wäre $T[X^{-1}(v_t)]$ nicht zusammenhängend), setzen wir $f(t) := v_t$ und fügen t zu D hinzu
- Dies führen wir solange fort, bis $D = V_T$, also f für alle $t \in V_T$ definiert ist

$\square$

Beziehungen zu anderen Graphparametern

- Zuerst zeigen wir, dass $tw(G) \geq \omega(G) - 1$ ist
- Für zwei Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ mit $V_1 \cap V_2 \neq \emptyset$ bezeichne $G_1 \cap G_2$ den Graphen $(V_1 \cap V_2, E_1 \cap E_2)$
- Sind G_1 und G_2 Teilgraphen eines Graphen G, so bezeichne $d(G_1, G_2)$ die minimale Länge eines Pfades P in G, der einen Knoten $v_1 \in V_1$ mit einem Knoten $v_2 \in V_2$ verbindet
- Offenbar gilt genau dann $d(G_1, G_2) = 0$, wenn $V_1 \cap V_2 \neq \emptyset$ ist
- Falls $T_1 = (V_1, E_1)$ und $T_2 = (V_2, E_2)$ Teilbäume eines Baums T sind, gibt es im Fall $d(T_1, T_2) > 0$ genau einen kürzesten Pfad P zwischen T_1 und T_2
- P ist nämlich der einzige Pfad zwischen T_1 und T_2, der keine inneren Knoten aus $V_1 \cup V_2$ enthält
- Gilt dagegen $d(T_1, T_2) = 0$, so ist auch $T_1 \cap T_2$ ein Teilbaum von T
- Da nämlich zu je 2 Knoten u, v in $T' = T_1 \cap T_2$ genau ein u-v-Pfad P in T existiert und dieser Pfad auch in jedem Teilbaum T_i enthalten ist (da T_i zusammenhängend ist), ist P auch ein u-v-Pfad in T'

Beziehungen zu anderen Graphparametern

Lemma

- Für $i = 1, \ldots, k$ seien $T_i = (V_i, E_i)$ und $S = (V, E)$ Teilbäume von T mit $V_1 \cap \cdots \cap V_k \neq \emptyset$
- Dann gilt $d(T_1 \cap \cdots \cap T_k, S) = \max_{1 \leq i \leq k} d(T_i, S)$

Beweis.

- Im Fall $d(T_1 \cap \cdots \cap T_k, S) = 0$, folgt sofort $d(T_i, S) = 0$ für $1 \leq i \leq k$
- Im Fall $d(T_1 \cap \cdots \cap T_k, S) > 0$ führen wir Induktion über k
- Für $k = 1$ ist nichts zu zeigen
- Im Fall $k = 2$ sei $P = (u_0, u_1, \ldots, u_d)$ der kürzeste Pfad zwischen $T_1 \cap T_2$ und S in T
- Dann ist $d \geq 1$, $u_0 \in V_1 \cap V_2$, $u_1 \notin V_1 \cap V_2$, $u_{d-1} \notin V$ und $u_d \in V$
- Zudem ist P genau dann der kürzeste Pfad zwischen T_i und S in T, wenn der Knoten u_1 nicht zu T_i gehört

Beweis (Fortsetzung).

- Da $u_1 \notin V_1 \cap V_2$ ist, gibt es also folgende 3 Unterfälle, in denen jeweils $d = \max_{i=1,2} d(T_i, S)$ gilt:
  - $u_1 \in V_1 \setminus V_2$, d.h. $d = d(T_2, S) > d(T_1, S)$
  - $u_1 \in V_2 \setminus V_1$, d.h. $d = d(T_1, S) > d(T_2, S)$, sowie
  - $u_1 \notin V_1 \cup V_2$, d.h. $d = d(T_1, S) = d(T_2, S)$
- Im Fall $k \geq 3$ sei $T' = T_1 \cap \cdots \cap T_k$ und $T'' = T_1 \cap \cdots \cap T_{k-1}$
- Dann gilt nach IV $d(T'' \cap T_k, S) = \max\{d(T'', S), d(T_k, S)\}$ und $d(T'', S) = \max_{1 \leq i < k} d(T_i, S)$
- Daher folgt

$$d(\underbrace{T_1 \cap \cdots \cap T_k}_{T'' \cap T_k}, S) = \max\{\underbrace{d(T'', S)}_{\max_{1 \leq i < k} d(T_i, S)}, d(T_k, S)\} = \max_{1 \leq i \leq k} d(T_i, S)$$

$\square$

Definition. Sei $S = \{A_i\}_{i \in I}$ ein Mengensystem.

- S hat die Helly-Eigenschaft, wenn für jede Teilmenge $J \subseteq I$ gilt:

$$(\forall i, j \in J : A_i \cap A_j \neq \emptyset) \Rightarrow \bigcap_{j \in J} A_j \neq \emptyset$$

- In diesem Fall heißt S Helly-System

Lemma

Die Knotenmengen $V_1, \ldots, V_k$ von beliebigen Teilbäumen $T_i = (V_i, E_i)$ eines Baumes T bilden ein Helly-System $\{V_i\}_{i=1,\ldots,k}$

Lemma

Die Knotenmengen $V_1, \ldots, V_k$ von beliebigen Teilbäumen $T_i = (V_i, E_i)$ eines Baumes T bilden ein Helly-System $\{V_i\}_{i=1,\ldots,k}$

Beweis.

- Wir zeigen induktiv über k, dass die Teilbäume $T_1, \ldots, T_k$ einen gemeinsamen Knoten haben, wenn dies für je zwei Teilbäume T_i, T_j gilt
- Für $k \leq 2$ ist nichts zu zeigen
- Im Fall $k \geq 3$ gilt nach IV, dass $T_1, \ldots, T_{k-1}$ einen gemeinsamen Knoten haben
- Zudem folgt mit obigem Lemma, dass

$$d(T_1 \cap \cdots \cap T_{k-1}, T_k) = \max_{1 \leq i < k} d(T_i, T_k) = 0$$

ist, und somit auch $T_1, \ldots, T_k$ einen gemeinsamen Knoten haben $\square$

Baum- und Pfadweite

Satz. Sei (T, X) eine TD eines Graphen G und sei C eine Clique in G.

Dann gibt es einen Knoten $s \in V_T$ mit $C \subseteq X_s$

Beweis.

- Wir betrachten für jeden Knoten $u \in C$ den Teilbaum $T_u = T[X^{-1}(u)]$
- Dann haben je 2 Teilbäume T_u und T_v einen gemeinsamen Knoten t:
  - Wegen $\{u, v\} \in E$ gibt es eine Tasche X_t mit $\{u, v\} \in X_t$, d.h. $t \in X^{-1}(u) \cap X^{-1}(v)$
- Wegen der Helly-Eigenschaft haben dann alle Teilbäume T_u, $u \in C$, einen gemeinsamen Knoten s und somit sind alle Knoten $u \in C$ in der Tasche X_s enthalten $\square$

Baum- und Pfadweite

Korollar

1. $tw(K_n) = n - 1$
2. $tw(G) \geq \omega(G) - 1$
3. $TW(1) = \{G \mid G \text{ ist ein Wald}\} = \{G \mid G \text{ ist } K_3\text{-frei}\}$

Beweis.

1 und 2 folgen direkt aus obigem Satz

3. Falls ein Graph G kein Wald ist, muss er den K_3 als Minor enthalten, und es folgt $tw(G) \geq tw(K_3) = 2$ $\square$

Definition

Eine TD (T, X) von G heißt Baumzerlegung in Cliquen (kurz TDC), wenn alle Taschen X_t Cliquen in G sind

Baum- und Pfadweite

Als direkte Folgerung des nächsten Satzes erhalten wir für alle chordalen Graphen G die Gleichheit $tw(G) = \omega(G) - 1$

Satz

Ein Graph G ist genau dann chordal, wenn G eine TDC hat

Beweis.

- Falls G chordal und $(v_1, \ldots, v_n)$ eine PEO für G ist, so sind die Mengen $C_i = N_{G_i}(v_i) \cup \{v_i\}$ Cliquen in $G_i = G[v_1, \ldots, v_i]$ und in G
- Zudem gilt $\bigcup_{i=1}^{n} C_i = V$ und jede Kante von G ist in mindestens einer Clique C_i enthalten
- Daher eignen sich die Cliquen C_t, $t \in V_T = \{1, \ldots, n\}$ als Taschen einer TDC (T, X) mit $X_t = C_t$ für $t \in V_T$
- Um einen Baum $T = (V_T, E_T)$ zu erhalten, verbinden wir jeden Knoten $t > 1$ in V_T mit dem Knoten $parent(t) = \max\{i < t \mid v_i \in C_t \vee i = 1\}$

Beweis (Fortsetzung).

- Induktiv über $k = 1, \ldots, n$ folgt dann nämlich, dass für jedes $i \leq k$ die Baumknoten $t \leq k$ mit $v_i \in C_t$ einen Unterbaum $T_{i,k}$ von T induzieren:
  - Für $i = k$ ist dies klar, da k der einzige Baumknoten mit $v_k \in C_k$ ist
  - Für $i < k$ mit $v_i \notin C_k$ folgt dies direkt aus der IV, da $T_{i,k} = T_{i,k-1}$ ist
  - Für $i < k$ mit $v_i \in C_k = N_{G_k}(v_k) \cup \{v_k\}$ ist v_i wegen

    $$v_i \in N_{G_k}(v_k) \subseteq N_{G_{parent(k)}}(v_{parent(k)}) \cup \{v_{parent(k)}\} = C_{parent(k)}$$

    auch in $C_{parent(k)}$ enthalten und daher induzieren die Baumknoten $t \leq parent(k)$ mit $v_i \in C_t$ nach IV einen Unterbaum $T_{i,parent(k)}$ von T, der mit k durch die Kante $\{parent(k), k\}$ verbunden ist

Beweis (Schluss).

- Sei nun umgekehrt (T, X) eine TDC von G
- Dann erhalten wir aus (T, X) für jeden Kreis $K = (u_1, \ldots, u_\ell, u_1)$ eine TD (T, X') für den induzierten Teilgraphen $G[u_1, \ldots, u_\ell]$ mit den Taschen $X'_t = X_t \cap \{u_1, \ldots, u_\ell\}$
- Wegen $tw(G[u_1, \ldots, u_\ell]) \geq tw(K) \geq 2$ gibt es in (T, X') eine Tasche X'_t der Größe $|X'_t| \geq 3$
- Da X'_t in $G[u_1, \ldots, u_\ell]$ eine Clique induziert, muss K im Fall $\ell \geq 4$ eine Sehne in G haben $\square$

Korollar

Für chordale Graphen $G = (V, E)$ kann eine TDC in Linearzeit berechnet werden.

Beweis.

- Wir berechnen zuerst eine PEO $(v_1, \ldots, v_n)$
- Die zugehörigen Cliquen $C_i = N_{G_i}(v_i) \cup \{v_i\}$ berechnen wir, indem wir in der Adjazenzliste von v_i alle Knoten v_j mit $j < i$ auswählen
- Dabei können wir auch gleich $\mathit{parent}(i) = \max\{j < i \mid v_j \in C_i \vee j = 1\}$ in Linearzeit bestimmen

$\square$

Baum- und Pfadweite

Korollar. Für jeden Graphen G gilt

$$tw(G) = \min\{\omega(H) - 1 \mid H \text{ ist chordaler Supergraph von } G\}$$

Beweis.

- Zum Nachweis von $\leq$ sei H ein chordaler Supergraph von G
- Dann ist $tw(G) \leq tw(H)$, da G ein Teilgraph von H und $tw(H) = \omega(H) - 1$, da H chordal ist
- Zum Nachweis von $\geq$ sei (T, X) eine TD von G der Weite $k = tw(G)$
- Dann können wir G zu einem chordalen Graphen H mit $\omega(H) \leq k + 1$ erweitern, indem wir alle Taschen X_t zu Cliquen auffüllen $\square$

Zudem ist $tw(G) = \min\{k \mid G \text{ ist Teilgraph eines } k\text{-Baums}\}$ (s. Übungen)

Korollar

Für jeden Graphen G gilt $\omega(G) \leq \chi(G) \leq tw(G) + 1$.

Beweis.

- Sei H ein chordaler Supergraph von G mit $tw(G) = \omega(H) - 1$
- Dann folgt $\omega(G) \leq \chi(G) \leq \chi(H) = \omega(H) = tw(G) + 1$ $\square$

Baum- und Pfadweite

Definition. Sei $\mathcal{F} = \{A_u\}_{u \in V}$ eine Familie von Mengen.

Dann ist der Schnittgraph von $\mathcal{F}$ der Graph $G_{\mathcal{F}} = (V, E)$ mit der Kantenmenge

$$E = \left\{ \{u, v\} \in \tbinom{V}{2} \mid A_u \cap A_v \neq \emptyset \right\}$$

- Für jeden Graphen $G = (V, E)$ gibt es eine Familie $\mathcal{F} = \{A_u\}_{u \in V}$ mit $G_{\mathcal{F}} = G$
- Eine einfache Möglichkeit besteht etwa darin, für A_u die Menge $\{e \in E \mid u \in e\}$ aller mit u inzidenten Kanten zu wählen
- Lässt man jedoch nur Familien $\mathcal{F}$ von speziellen Mengen zu, können sich interessante Graphklassen ergeben
- Zum Beispiel lassen sich chordale Graphen als Schnittgraphen von Teilbäumen T_u eines Baums T charakterisieren
- Ein weiteres prominentes Beispiel ist die Klasse der Intervallgraphen

Satz. Sei $G = (V, E)$ ein Graph.

Dann ist G genau dann chordal, wenn G der Schnittgraph $G_{\mathcal{F}}$ einer Familie von Knotenmengen von Teilbäumen T_u eines Baums T ist

Beweis.

- Sei $G = (V, E)$ chordal und sei $(T, X) = (V_T, E_T, X)$ eine TDC von G
- Dann gilt

$$\{u, v\} \in E \ \Rightarrow\ \exists t \in V_T : \{u, v\} \subseteq X_t \ \Leftrightarrow\ X^{-1}(u) \cap X^{-1}(v) \neq \emptyset,$$

- Da die Taschen X_t von (T, X) Cliquen in G sind, gilt auch die Implikation von rechts nach links, d.h. G ist der Schnittgraph der Knotenmengen $X^{-1}(u)$ der Teilbäume $T_u = T[X^{-1}(u)]$ von T
- Ist umgekehrt $G = G_{\mathcal{F}}$ der Schnittgraph einer Familie $\mathcal{F} = (V_u)_{u \in V}$ von Knotenmengen V_u von Teilbäumen T_u eines Baums $T = (V_T, E_T)$, so ist (T, X) mit $X_t = \{u \in V \mid t \in V_u\}$ eine TDC von G und somit G chordal $\square$

Baum- und Pfadweite

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

G heißt Intervallgraph, wenn G Schnittgraph einer Familie $\mathcal{F} = \{I_u\}_{v \in V}$ von nichtleeren Intervallen $I_u = [a_u, b_u] = \{x \in \mathbb{Z} \mid a_u \leq x \leq b_u\}$ mit $a_u, b_u \in \mathbb{Z}$ ist. In diesem Fall heißt $\mathcal{F}$ Intervallrepräsentation von G.

- Es ändert nichts an der Klasse der Intervallgraphen, wenn reelle Intervalle $I_u \subseteq \mathbb{R}$ anstelle von diskreten Intervallen $I_u \subseteq \mathbb{Z}$ zugelassen werden
- Einerseits kann jedes diskrete Intervall auch als reelles Intervall aufgefasst werden, ohne dass sich der Schnittgraph ändert
- Andererseits kann eine Familie von reellen Intervallen in eine von diskreten Intervallen mit gleichem Schnittgraphen überführt werden, indem nur die Endpunkte der Intervalle beibehalten werden und unter Beibehaltung ihrer Reihenfolge auf ganze Zahlen verschoben werden
- Dies gilt auch, wenn neben abgeschlossenen noch (halb-)offene Intervalle zugelassen werden
- Im weiteren Verlauf wird sich die diskrete Version als nützlich erweisen

Definition

Eine PD (T, X) von G heißt Pfadzerlegung in Cliquen (kurz PDC), wenn alle Taschen X_t Cliquen in G sind

Proposition. Für einen Graphen $G = (V, E)$ sind äquivalent:

1. G ist ein Intervallgraph
2. G ist der Schnittgraph von Teilpfaden P_u eines Pfades $P = (V_P, E_P)$
3. G hat eine PDC (P, X)

Baum- und Pfadweite

Proposition. Für einen Graphen $G = (V, E)$ sind äquivalent:

1. G ist ein Intervallgraph
2. G ist der Schnittgraph von Teilpfaden P_u eines Pfades $P = (V_P, E_P)$
3. G hat eine PDC (P, X)

Beweis von 1 $\Rightarrow$ 2:

- Sei $\mathcal{F} = \{I_u\}_{u \in V}$ mit $I_u = [a_u, b_u]$ eine Intervallrepräsentation von G
- Wir wählen für $V_P = \{a_u \mid u \in V\}$ die Menge aller linken Endpunkte und setzen $E_P = \{\{a_u, a_v\} \mid a_u < a_v \wedge \nexists w \in V : a_u < a_w < a_v\}$
- Dann gilt für die Familie $\{P_u\}_{u \in V}$ der durch $V_u = V_P \cap I_u$ induzierten Teilpfade $P_u = P[V_u]$ von P:

$$\{u, v\} \in E \;\Leftrightarrow\; I_u \cap I_v \neq \emptyset \;\Leftrightarrow\; a_u \in I_v \vee a_v \in I_u \;\Leftrightarrow\; V_u \cap V_v \neq \emptyset,$$

d.h. G ist der Schnittgraph der Knotenmengen der Teilpfade P_u von P

Baum- und Pfadweite

Proposition. Für einen Graphen $G = (V, E)$ sind äquivalent:

1. G ist ein Intervallgraph
2. G ist der Schnittgraph von Teilpfaden P_u eines Pfades $P = (V_P, E_P)$
3. G hat eine PDC (P, X)

Beweis von 2 $\Rightarrow$ 3:

- Man verifiziert leicht, dass (P, X) mit $X : p \mapsto X_p = \{u \in V \mid p \in V_u\}$ eine Pfadzerlegung von G ist, wobei $P_u = (V_u, E_u)$ der zu $u \in V$ gehörige Teilpfad von P ist
- Zum Beispiel ist $P[X^{-1}(u)] = P_u$ der durch $X^{-1}(u) = V_u$ induzierte Teilpfad von P
- Zudem ist jede Tasche X_p eine Clique in G, da alle Pfade P_u mit $u \in X_p$ den Knoten p gemeinsam haben

Proposition. Für einen Graphen $G = (V, E)$ sind äquivalent:

1. G ist ein Intervallgraph
2. G ist der Schnittgraph von Teilpfaden P_u eines Pfades $P = (V_P, E_P)$
3. G hat eine PDC (P, X)

Beweis von 3 $\Rightarrow$ 1:

- Wir definieren $z\colon V_P \to \mathbb{N}$ durch $z(p) = d_P(p_0, p)$, wobei p_0 einer der Endknoten von P ist
- Für $u \in V$ sei $I_u = \{z(p) \mid p \in X^{-1}(u)\}$
- Da $X^{-1}(u)$ einen Teilpfad $P_u = P[X^{-1}(u)]$ von P induziert, sind I_u, $u \in V$, Intervalle und da die Taschen X_p Cliquen sind, folgt

$$I_u \cap I_v \neq \emptyset \;\Leftrightarrow\; \exists p \in V_P : \{u, v\} \subseteq X_p \;\Leftrightarrow\; \{u, v\} \in E,$$

  d.h. $\mathcal{F} = \{I_u\}_{u \in V}$ ist eine Intervallrepräsentation von G

Ähnlich wie für chordale Graphen gibt es auch hier einen Linearzeitalgorithmus, der bei Eingabe eines Graphen G eine Intervallrepräsentation bzw. eine PDC für G findet, sofern diese existieren (also G ein Intervallgraph ist)

Definition

- Sei $A \subseteq \Sigma^*$ und $k\colon \Sigma^* \to \mathbb{N}_0$ ein Parameter (beispielsweise die Baumweite des Eingabegraphen)
- Dann heißt A fixed parameter tractable (kurz: FPT) bezüglich k, wenn es eine Funktion $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ und einen Algorithmus gibt, der für jedes $x \in \Sigma^*$ in Zeit $f(k(x)) \cdot |x|^{O(1)}$ entscheidet, ob $x \in A$ ist

- Die Funktion f ist hierbei beliebig, insbesondere kann sie exponentiell (oder noch schneller) wachsen
- Für NP-schwere Probleme sind FPT-Algorithmen dennoch interessant, weil sie das exponentielle Laufzeitverhalten auf den Parameter eingrenzen
- Wird der Parameter als konstant (oder relativ klein) angenommen, ergibt sich für wachsende Eingabegrößen noch eine polynomiell begrenzte Rechenzeit

- Dabei ist die Aussage „A ist in FPT bezüglich der Baumweite“ stärker als „für jede Zahl k ist A für Graphen in $TW(k)$ effizient entscheidbar“
- Letzteres trifft nämlich auch noch bei einer Laufzeit von $n^{f(k)}$ zu (hierfür sagt man, A ist in XP bezüglich k)
- Wichtig ist also, dass der Exponent des Polynoms, das die Rechenzeit beschränkt, unabhängig von k ist
- Viele Probleme, die für allgemeine Graphen NP-schwer sind, sind FPT bezüglich der Baumweite
- Häufig lassen sich sogar Laufzeiten der Form $f(k) \cdot n$ erreichen (die Eingabegröße geht also nur linear in die Laufzeit ein)
- Um einen FPT-Algorithmus bezüglich der Baumweite zu erhalten, bietet es sich an, den Zerlegungsbaum T an einem beliebigen Knoten zu wurzeln und dann von den Blättern aufwärts Teillösungen zu berechnen und diese zu Teillösungen von immer größeren Teilgraphen von G zusammenzusetzen, bis eine Lösung für G vorliegt

Dynamische Programmierung über Baumzerlegungen

Definition. Sei (T, X) eine TD für $G = (V, E)$ und sei $r \in V_T$.

- Dann heißt (T_r, X) Wurzelbaumzerlegung (kurz rTD) von G, wobei der gerichtete Baum T_r aus T durch Orientierung aller Kanten weg von der Wurzel r entsteht
- Zudem definieren wir für $t \in V_T$:

$$T(t) = \{s \in V_T \mid s \text{ ist von } t \text{ aus erreichbar}\}$$

$$V(t) = \bigcup_{s \in T(t)} X_s$$

$$G(t) = G[V(t)]$$

- Die Idee ist nun, für jeden Baumknoten $t \in V_T$ und jede lokale Lösung L_t auf $G[X_t]$ die Information zu speichern, ob (und ggf. wie) sich L_t zu einer Lösung $\hat{L}_t$ auf $G(t)$ erweitern lässt
- Dabei werden wir den Baum T_r post-order traversieren, damit bei der Bearbeitung jedes Knotens t die benötigten Informationen über seine Kinder bereits vorliegen

Dynamische Programmierung über Baumzerlegungen

Damit ergibt sich folgender Meta-Algorithmus

for each $t \in V_T$ (bottom-up) **do**
    for each Lösung L_t auf $G[X_t]$ **do**
        speichere, ob (und wie) L_t mit bereits bekannten Lösungen
        $\hat{L}_{s_1}, \ldots, \hat{L}_{s_k}$ für die Kinder $s_1, \ldots, s_k$ von t zu einer Lösung
        $\hat{L}_t$ auf $G(t)$ kombiniert werden kann
prüfe, ob eine Lösung $\hat{L}_r$ auf $G = G(r)$ existiert

Um diesen Meta-Algorithmus für ein konkretes Problem anzupassen, muss jeweils geklärt werden, was eine lokale Lösung ist, wann lokale Lösungen kompatibel sind und wie diese kombiniert werden können

Färbbarkeit

- Als Beispiel für ein NP-schweres Problem, das FPT in der Baumweite ist, betrachten wir das Färbbarkeitsproblem
- Wir nehmen an, dass der Eingabegraph G zusammen mit einer kompakten TD (T, X) vorliegt
- Als Kandidaten für eine lokale Lösung an einem Baumknoten $t \in V_T$ kommen nur Funktionen $f_t \colon X_t \to \{1, \ldots, k\}$ infrage
- Wir definieren auf diesen Funktionen das Prädikat

$$P(f_t) = 1 :\Leftrightarrow f_t \text{ ist zu einer } k\text{-Färbung } \hat{f}_t \text{ von } G(t) \text{ erweiterbar}$$

- Um $P(f_t)$ im Fall eines Blattes t auszuwerten, ist nur zu verifizieren, ob f_t eine k-Färbung von $G[X_t]$ ist
- Ist t dagegen ein innerer Knoten, muss zusätzlich noch die Erweiterbarkeit von f_t zu einer k-Färbung $\hat{f}_t$ von $G(t)$ geprüft werden
- Hierzu dient nachfolgendes Lemma

Definition

- Zwei Funktionen $f_1 \colon V_1 \to \{1, \ldots, k\}$ und $f_2 \colon V_2 \to \{1, \ldots, k\}$ heißen kompatibel, wenn $f_1(u) = f_2(u)$ für alle $u \in V_1 \cap V_2$ gilt
- Falls zusätzlich $V_1 \subset V_2$ gilt, so heißt f_2 eine Erweiterung von f_1

Lemma

$P(f_t) = 1$ gilt genau dann, wenn $f_t \colon X_t \to \{1, \ldots, k\}$ eine k-Färbung von $G[X_t]$ ist und für jedes Kind s von t eine zu f_t kompatible Funktion $f_s \colon X_s \to \{1, \ldots, k\}$ mit $P(f_s) = 1$ existiert

Lemma

$P(f_t) = 1$ gilt genau dann, wenn $f_t \colon X_t \to \{1, \ldots, k\}$ eine k-Färbung von $G[X_t]$ ist und für jedes Kind s von t eine zu f_t kompatible Funktion $f_s \colon X_s \to \{1, \ldots, k\}$ mit $P(f_s) = 1$ existiert

Beweis von „$\Rightarrow$"

- Sei $\hat{f}_t$ eine k-Färbung von $G(t)$, die f_t erweitert
- Dann ist f_t eine k-Färbung von $G[X_t]$ und für jedes Kind s von t ist die Restriktion $f_s = \hat{f}_t|_{X_s}$ eine zu f_t kompatible Funktion mit $P(f_s) = 1$, da sie sich zu einer k-Färbung $\hat{f}_s = \hat{f}_t|_{V(s)}$ von $G(s)$ erweitern lässt $\square$

Dynamische Programmierung über Baumzerlegungen

Lemma

$P(f_t) = 1$ gilt genau dann, wenn $f_t \colon X_t \to \{1, \ldots, k\}$ eine k-Färbung von $G[X_t]$ ist und für jedes Kind s von t eine zu f_t kompatible Funktion $f_s \colon X_s \to \{1, \ldots, k\}$ mit $P(f_s) = 1$ existiert

Beweis von „$\Leftarrow$“

- Falls t keine Kinder hat, ist jede k-Färbung f_t von $G[X_t]$ auch eine k-Färbung von $G(t)$
- Andernfalls seien $s_1, \ldots, s_d$ die Kinder von t und für $i = 1, \ldots, d$ seien $f_{s_i} \colon X_{s_i} \to \{1, \ldots, k\}$ zu f_t kompatible Funktionen mit $P(f_{s_i}) = 1$ (d.h. f_{s_i} lässt sich zu einer k-Färbung $\hat{f}_{s_i}$ von $G(s_i)$ erweitern)
- Wegen $V(s_i) \cap V(s_j) \subseteq X_t$ für $1 \leq i < j \leq d$ sind dann die Funktionen $f_t, \hat{f}_{s_1}, \ldots, \hat{f}_{s_d}$ paarweise kompatibel
- Somit ist ihre Vereinigung $\hat{f}_t = f_t \cup \hat{f}_{s_1} \cup \cdots \cup \hat{f}_{s_d}$ eine k-Färbung von $G(t)$, die f_t erweitert, d.h. es gilt $P(f_t) = 1$ $\square$

Dynamische Programmierung über Baumzerlegungen

Insgesamt ergibt sich der folgende Algorithmus `TD-color`, der das Prädikat $P(f_t)$ für alle Funktionen $f_t\colon X_t \to \{1,\dots,k\}$ berechnet, und im Fall, dass für die Wurzel r eine Funktion f_r mit $P(f_r) = 1$ gefunden wird, eine zugehörige Erweiterung $\hat{f}_r$ als k-Färbung von G ausgibt

Algorithmus `TD-color`(V, E, T_r, X, k)

for each $t \in V_T$ (bottom-up) **do**
  for each $f_t\colon X_t \to \{1,\dots,k\}$ **do**
    if $\exists u, v \in X_t\colon \{u,v\} \in E \wedge f_t(u) = f_t(v)$ **then**
      $P(f_t) := 0$
    else
      seien s_i, $i = 1,\dots,d$ die Kinder von t
      if $\forall i \, \exists f_{s_i}\colon P(f_{s_i}) = 1$ und f_{s_i} ist zu f_t kompatibel **then**
        $P(f_t) := 1;\ \hat{f}_t := f_t \cup \hat{f}_{s_1} \cup \cdots \cup \hat{f}_{s_d}$
      else
        $P(f_t) := 0$
if $\exists f_r : P(f_r) = 1$ **then output** $\hat{f}_r$ **else output** $\perp$