11

Cliquenweitebeschränkte Graphen

In diesem Kapitel werden wir einen zweiten Ansatz zur Lösung schwieriger Graphenprobleme auf speziellen Baumstrukturen kennen lernen. Im Gegensatz zum Ansatz über die Baumweite wird es im folgenden Ansatz auch möglich sein, im Sinne der Fest-Parameter-Algorithmik solche Instanzen effizient zu lösen, die beliebig dichte Graphen (z. B. vollständige Graphen oder vollständig bipartite Graphen) enthalten. Dazu werden wir den Graphparameter Cliquenweite und seinen algorithmischen Nutzen vorstellen.

11.1 Grundlagen

Der Begriff der Cliquenweite (englisch: *clique-width*) von Graphen wurde um 1994 von Courcelle und Olariu eingeführt. Die Cliquenweite basiert auf rekursiven Operationen auf Graphen mit Knotenmarkierungen. Zur Erinnerung: Für eine natürliche Zahl k bezeichnen wir mit $[k]$ die k-elementige Menge $\{1,\ldots,k\}$ von natürlichen Zahlen. Ein *k-markierter Graph* $G=(V_G,E_G,\mathrm{lab}_G)$ ist ein Graph $G=(V_G,E_G)$, dessen Knoten mit einer Markierungsabbildung $\mathrm{lab}_G : V_G \to [k]$ markiert werden. Ein Graph, der aus genau einem mit $i \in [k]$ markierten Knoten besteht, wird im Folgenden kurz mit $\bullet_i$ bezeichnet. $\bullet_i$

Definition 11.1 (Cliquenweite für knotenmarkierte Graphen).

- *Die Graphklasse* CW_k *ist rekursiv wie folgt definiert:* CW_k
  1. *Der* k-markierte Graph $\bullet_i$ *ist für* $i \in [k]$ *in* CW_k. k-markierter Graph $\bullet_i$
  2. *Es seien* $G=(V_G,E_G,lab_G) \in \mathrm{CW}_k$ *und* $J=(V_J,E_J,lab_J) \in \mathrm{CW}_k$ *zwei knotendisjunkte k-markierte Graphen. Dann ist der* k-markierte Graph $G \oplus J = (V',E',\mathrm{lab}')$ *in* CW_k*, der definiert ist durch* $V' = V_G \cup V_J$*,* $E' = E_G \cup E_J$ *und* k-markierter Graph $G \oplus J$

$$lab'(u) = \begin{cases} lab_G(u), & \text{falls } u \in V_G, \\ lab_J(u), & \text{falls } u \in V_J, \end{cases}$$

  für alle $u \in V'$.

F. Gurski et al., *Exakte Algorithmen für schwere Graphenprobleme*, eXamen.press,
DOI 10.1007/978-3-642-04500-4_11,

3. Es seien $i, j \in [k]$ zwei verschiedene natürliche Zahlen und $G = (V_G, E_G, lab_G) \in \mathrm{CW}_k$ ein k-markierter Graph. Dann ist

k-markierter Graph $\rho_{i \to j}(G)$

a) der k-markierte Graph $\rho_{i \to j}(G) = (V_G, E_G, \mathrm{lab}')$ *in* CW_k, *mit*

$$lab'(u) = \begin{cases} lab_G(u), & \text{falls } lab_G(u) \neq i, \\ j, & \text{falls } lab_G(u) = i, \end{cases}$$

für alle $u \in V_G$, und

k-markierter Graph $\eta_{i,j}(G)$

b) es ist der k-markierte Graph $\eta_{i,j}(G) = (V_G, E', \mathrm{lab}_G)$ *mit*

$$E' = E_G \cup \{\{u,v\} \mid lab(u) = i,\ lab(v) = j\}$$

in CW_k.

Cliquenweite-k-Ausdruck

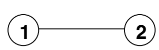
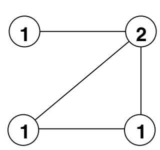
- *Einen Ausdruck X mit den Operationen $\bullet_i$, $\oplus$, $\rho_{i \to j}$ und $\eta_{i,j}$ für $i, j \in [k]$ nennen wir* Cliquenweite-k-Ausdruck *oder kurz* k-Ausdruck.
- *Ein k-markierter Graph G hat eine* Cliquenweite von höchstens k, *falls G in* CW_k *ist.*

Cliquenweite eines markierten Graphen

- *Die* Cliquenweite eines markierten Graphen *G (kurz* Cliquenweite(G)*) ist die kleinste natürliche Zahl k, sodass G in* CW_k *ist.*

Beispiel 11.2 (Cliquenweite-2-Ausdruck). In Tabelle 11.1 werden einige Cliquenweite-2-Ausdrücke und die definierten markierten Graphen angegeben.

Tabelle 11.1. Cliquenweite-2-Ausdrücke und die definierten 2-markierten Graphen

Graph	Cliquenweite-2-Ausdruck
G_1: 	$\bullet_1$
G_2: 	$\bullet_2$
G_3: 	$G_1 \oplus G_2$
G_4: 	$\eta_{1,2}(G_3)$
G_5: 	$\rho_{2 \to 1}(G_4)$
G_6: 	$\eta_{1,2}(G_5 \oplus G_3)$

In Definition 11.1 wurde der Begriff der Cliquenweite für Graphen mit Knotenmarkierungen eingeführt. Für Graphen ohne Knotenmarkierungen wollen wir dies wie folgt auf die obige Definition zurückführen.

Definition 11.3 (Cliquenweite für Graphen ohne Knotenmarkierungen). *Die* Cliquenweite eines Graphen $G = (V, E)$ *(wir schreiben auch kurz* Cliquenweite(G)*) ist die kleinste natürliche Zahl k, für die es eine Abbildung $\ell : V \to [k]$ gibt, sodass der k-markierte Graph (V, E, ℓ) eine Cliquenweite von höchstens k hat.* Cliquenweite

Alle vollständigen Graphen bzw. Wege haben eine Cliquenweite von höchstens 2 bzw. 3, wie das folgende Beispiel zeigt.

Beispiel 11.4 (Cliquenweite).

1. Jeder vollständige Graph K_n mit $n \geq 1$ Knoten kann wie folgt mit Hilfe der Operationen der Cliquenweite aufgebaut werden:
$$\begin{aligned} X_{K_1} &= \bullet_1, \\ X_{K_2} &= \eta_{1,2}(\bullet_1 \oplus \bullet_2), \\ X_{K_n} &= \eta_{1,2}(\rho_{2\to 1}(X_{K_{n-1}}) \oplus \bullet_2), \quad \text{falls } n \geq 3. \end{aligned}$$
Somit hat jeder vollständige Graph K_n, $n \geq 1$, eine Cliquenweite von höchstens 2.
2. Jeder Weg P_n mit $n \geq 1$ Knoten kann wie folgt mit Hilfe der Operationen der Cliquenweite aufgebaut werden:
$$\begin{aligned} X_{P_1} &= \bullet_1, \\ X_{P_2} &= \eta_{1,2}(\bullet_1 \oplus \bullet_2), \\ X_{P_3} &= \eta_{2,3}(\eta_{1,2}(\bullet_1 \oplus \bullet_2) \oplus \bullet_3), \\ X_{P_n} &= \eta_{2,3}(\rho_{3\to 2}(\rho_{2\to 1}(X_{P_{n-1}})) \oplus \bullet_3), \quad \text{falls } n \geq 4. \end{aligned}$$
Somit hat jeder Weg P_n, $n \geq 1$, eine Cliquenweite von höchstens 3.

Übung 11.5. Zeigen Sie die folgenden Aussagen durch Angabe eines geeigneten Cliquenweite-3-Ausdrucks:

- Der Weg P_5 (mit 5 Knoten) hat eine Cliquenweite von höchstens 3.
- Der Kreis C_6 (mit 6 Knoten) hat eine Cliquenweite von höchstens 3.
- Der Gittergraph (bzw. „Domino“) $G_{2,3}$ hat eine Cliquenweite von höchstens 3.

Übung 11.6. Geben Sie jeweils möglichst große n an, sodass

- der Weg P_n mit n Knoten eine Cliquenweite von höchstens 2 hat,
- der Kreis C_n mit n Knoten eine Cliquenweite von höchstens 2 hat,
- der Kreis C_n mit n Knoten eine Cliquenweite von höchstens 3 hat.

Begründen Sie Ihre Angaben kurz.

Übung 11.7. Beweisen oder widerlegen Sie die folgenden Aussagen:

1. Für jeden Graphen G gilt: $\alpha(G) < \text{Cliquenweite}(G)$.
2. Für jeden Graphen G gilt: $\tau(G) < \text{Cliquenweite}(G)$.
3. Für jeden Graphen G gilt: $\omega(G) < \text{Cliquenweite}(G)$.
4. Für jeden Graphen G gilt: $\chi(G) < \text{Cliquenweite}(G)$.
5. Für jeden Graphen G gilt: $\theta(G) < \text{Cliquenweite}(G)$.

Genau wie die Baumweite hat auch die Cliquenweite wichtige algorithmische Anwendungen. Über die rekursive Definition der Cliquenweite-Ausdrücke erkennt man leicht, dass jeder Cliquenweite-Ausdruck X für einen Graphen G auch eine Baumstruktur für G liefert. Wir bezeichnen diesen Baum als *Cliquenweite-k-Ausdrucksbaum*. Der Cliquenweite-k-Ausdrucksbaum $T = (V_T, E_T, \text{lab}_T)$ zu einem Cliquenweite-k-Ausdruck X ist ein binärer Wurzelbaum, dessen Knoten mit den Operationen des zugehörigen k-Ausdrucks markiert sind.

Definition 11.8 (Cliquenweite-*k*-Ausdrucksbaum).

Cliquenweite-k-Ausdrucksbaum zu $\bullet_i$

- *Der* Cliquenweite-k-Ausdrucksbaum T zum k-Ausdruck $\bullet_i$ *besteht aus genau einem Knoten r (der Wurzel von T), der mit $\bullet_i$ markiert wird.*

Cliquenweite-k-Ausdrucksbaum zu $X_1 \oplus X_2$

- *Der* Cliquenweite-k-Ausdrucksbaum T zum k-Ausdruck $X_1 \oplus X_2$ *besteht aus der disjunkten Vereinigung der Cliquenweite-k-Ausdrucksbäume T_1 und T_2 von X_1 bzw. X_2, einem zusätzlichen Knoten r (der Wurzel von T), der mit $\oplus$ markiert wird, und zwei zusätzlichen Kanten zwischen dem Knoten r und der Wurzel von T_1 bzw. der Wurzel von T_2.*

Cliquenweite-k-Ausdrucksbaum zu $\rho_{i\to j}(X)$ bzw. $\eta_{i,j}(X)$

- *Der* Cliquenweite-k-Ausdrucksbaum T zu den beiden k-Ausdrücken $\rho_{i\to j}(X)$ bzw. $\eta_{i,j}(X)$ *besteht aus dem Cliquenweite-k-Ausdrucksbaum T' zum Ausdruck X, einem zusätzlichen Knoten r (der Wurzel von T), der mit $\rho_{i\to j}$ bzw. mit $\eta_{i,j}$ markiert wird, und einer zusätzlichen Kante zwischen der Wurzel von T' und dem Knoten r.*
- *In einem Cliquenweite-k-Ausdrucksbaum T nennen wir*
  – *einen mit $\bullet_i$ markierten Knoten ein* Blatt*,* (Blatt)
  – *einen mit $\oplus$ markierten Knoten einen* Vereinigungsknoten*,* (Vereinigungsknoten)
  – *einen mit $\rho_{i\to j}$ markierten Knoten einen* Ummarkierungsknoten *und* (Ummarkierungsknoten)
  – *einen mit $\eta_{i,j}$ markierten Knoten einen* Kanteneinfügeknoten. (Kanteneinfügeknoten)

Beispiel 11.9 (Cliquenweite-*k*-Ausdrucksbaum). Für den Graphen G_6 aus Beispiel 11.2 zeigt Abb. 11.1 einen Cliquenweite-2-Ausdrucksbaum.

Anmerkung 11.10. Betrachtet man einen Cliquenweite-k-Ausdrucksbaum T zu einem Graphen $G = (V,E)$ etwas genauer, so sieht man leicht, dass T genau $|V|$ Blätter und somit genau $|V|-1$ Vereinigungsknoten besitzt. Die Anordnung der Ummarkierungsknoten und Kanteneinfügeknoten zwischen zwei Vereinigungsknoten ist zunächst einmal beliebig und schwer abschätzbar.

Normalform für Cliquenweite-Ausdrücke

Espelage, Gurski und Wanke [EGW03] haben jedoch eine so genannte *Normalform für Cliquenweite-Ausdrücke* gefunden, die u. a. besagt, dass nach einer disjunkten Vereinigung zunächst eine Folge von Kanteneinfügeknoten folgt und dann eine Folge von Ummarkierungsknoten. Das heißt insbesondere, dass es zwischen zwei

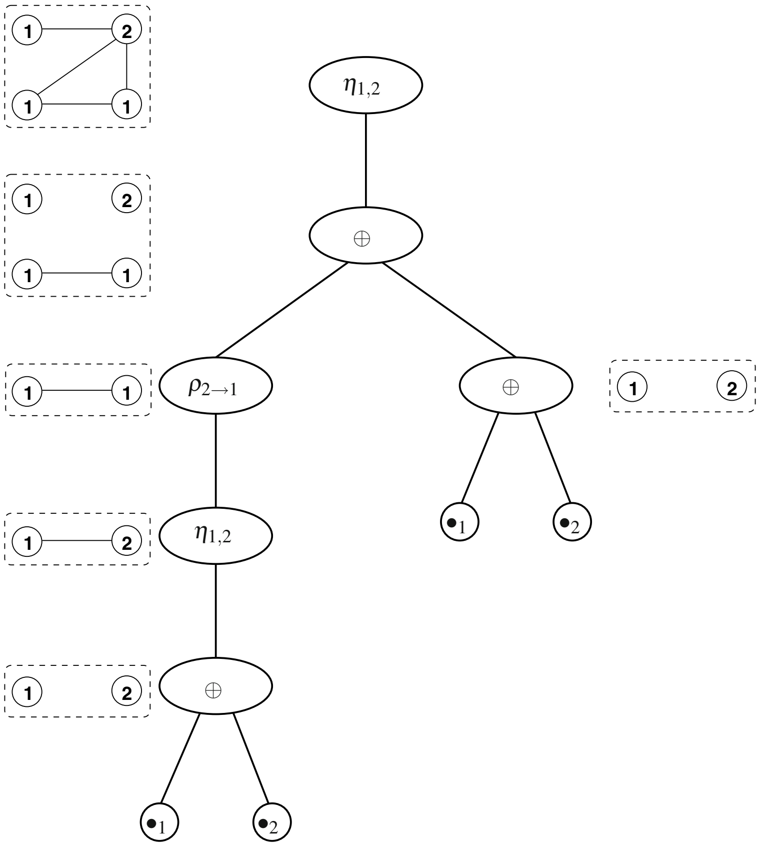


Abb. 11.1. Ein Cliquenweite-2-Ausdrucksbaum für den Graphen G_6 aus Beispiel 11.2

disjunkten Vereinigungen keine Ummarkierung vor einer Kanteneinfügung gibt. Somit gibt es zwischen zwei Vereinigungsknoten offenbar höchstens $k-1$ Ummarkierungsknoten und höchstens $^{k(k-1)}/_2$ Kanteneinfügeknoten, also insgesamt höchstens $(|V|-1)(k-1)$ Ummarkierungsknoten und höchstens $(|V|-1)(^{k(k-1)}/_2)$ Kanteneinfügeknoten.

Es ist bekannt, dass sich die Cliquenweite des Komplementgraphen stets in der Cliquenweite des Ursprungsgraphen beschränken lässt (eine Aussage, die offenbar für die Baumweite nicht gilt).

Satz 11.11 (Courcelle und Olariu [CO00]). *Für jeden Graphen G gilt:*

$$\text{Cliquenweite}(\overline{G}) \leq 2 \cdot \text{Cliquenweite}(G).$$

ohne Beweis

In den folgenden Sätzen vergleichen wir die Baumweite mit der Cliquenweite von Graphen. Es gibt mehrere Schranken für die Cliquenweite eines Graphen in Abhängigkeit von seiner Baumweite. Die beste bekannte Schranke ist im folgenden Satz angegeben.

Satz 11.12 (Corneil und Rotics [CR05]). *Für jeden Graphen G gilt:*

$$\text{Cliquenweite}(G) \leq 3 \cdot 2^{Baumweite(G)-1}.$$

ohne Beweis

Umgekehrt hingegen lässt sich die Baumweite eines Graphen nicht durch einen Ausdruck in seiner Cliquenweite beschränken, wie man sich leicht am Beispiel vollständiger Graphen überlegt. Betrachtet man jedoch nur Graphen, die keine beliebig großen vollständig bipartiten Graphen als Teilgraphen enthalten, so kann man auch die Baumweite eines Graphen durch einen Ausdruck in seiner Cliquenweite beschränken.

Satz 11.13 (Gurski und Wanke [GW00]). *Ist G ein Graph, der den $K_{n,n}$ nicht als Teilgraphen enthält, so gilt:*

$$\text{Baumweite}(G) \leq 3 \cdot (n-1) \cdot \text{Cliquenweite}(G) - 1.$$

ohne Beweis

Es gibt zahlreiche Graphen, die die Voraussetzung von Satz 11.13 erfüllen und für die sich somit aus einer Schranke für die Cliquenweite auch eine Schranke für die Baumweite herleiten lässt.

Korollar 11.14. *Es sei G ein Graph mit einer Cliquenweite von höchstens k.*

1. *Ist G planar, so hat G eine Baumweite von höchstens* $6k-1$.
2. *Hat jeder Knoten in G einen Grad von höchstens d, so hat G eine Baumweite von höchstens* $3kd-1$.

ohne Beweis

Während die Klasse der Graphen mit einer Baumweite von höchstens k bezüglich Teilgraphenbildung abgeschlossen ist (siehe Satz 10.19), ist dies für die Cliquenweite nicht der Fall. Am Beispiel vollständiger Graphen überlegt man sich sehr leicht ein Gegenbeispiel. Es gilt jedoch auch für die Cliquenweite noch der Abschluss bezüglich der Bildung induzierter Teilgraphen.

Satz 11.15. *Es sei G ein Graph. Die Cliquenweite eines jeden induzierten Teilgraphen von G ist durch die Cliquenweite von G nach oben beschränkt.* **ohne Beweis**

Wir definieren nun den Begriff der Cliquenweite von Graphklassen.

Definition 11.16. *Eine Graphklasse $\mathscr{G}$ hat eine* beschränkte Cliquenweite*, falls es eine natürliche Zahl k gibt, sodass alle Graphen in $\mathscr{G}$ eine Cliquenweite von höchstens k haben. Ist k die kleinste Zahl mit dieser Eigenschaft, so heißt k die* Cliquenweite von $\mathscr{G}$.

Graphklasse beschränkter Cliquenweite
Cliquenweite einer Graphklasse

Im folgenden Satz sind einige Mengen von Graphen mit beschränkter Cliquenweite zusamengefasst. Zunächst benötigen wir den folgenden Begriff.

Definition 11.17 (distanzerhaltender Graph). *Ein Graph G ist* distanzerhaltend*, falls in jedem zusammenhängenden induzierten Teilgraphen H von G die Distanz*[53] *zweier Knoten in H gleich der Distanz der beiden Knoten in G ist.*

distanzerhaltender Graph

Beispielsweise sind Bäume und Co-Graphen distanzerhaltend. Nicht distanzerhaltend sind hingegen zum Beispiel die Kreise C_n für $n \geq 5$.

Satz 11.18 (Courcelle und Olariu [CO00], Golumbic und Rotics [GR00]).

1. *Ein Graph hat genau dann die Cliquenweite* 1*, wenn er keine Kanten enthält.*
2. *Ein Graph hat genau dann eine Cliquenweite von höchstens* 2*, wenn er ein Co-Graph ist.*
3. *Distanzerhaltende Graphen haben eine Cliquenweite von höchstens* 3.

ohne Beweis

Dagegen hat die Menge aller Gittergraphen $\{G_{n,m} \mid n,m \in \mathbb{N}\}$ – und somit auch die Menge aller planaren Graphen – eine unbeschränkte Cliquenweite.

Übung 11.19. Zeigen Sie die folgenden Aussagen.

(1) Alle Bäume haben eine Cliquenweite von höchstens 3.
(2) Alle Kreise haben eine Cliquenweite von höchstens 4.

Übung 11.20. Zeigen Sie, dass jeder Co-Graph eine Cliquenweite von höchstens 2 hat und dass jeder Graph mit einer Cliquenweite von höchstens 2 ein Co-Graph ist.

Interessant sind stets Cliquenweite-Ausdrücke mit möglichst geringer Weite. Leider ist das Problem, solche optimalen Ausdrücke zu finden, bzw. schon das Bestimmen der Weite eines solchen Ausdrucks schwierig. Als Entscheidungsproblem lässt sich dies folgendermaßen ausdrücken:

CLIQUENWEITE

CLIQUENWEITE	
Gegeben:	Ein Graph $G = (V,E)$ und eine Zahl $k \in \mathbb{N}$.
Frage:	Hat G eine Cliquenweite von höchstens k?

Satz 11.21 (Fellows, Rosamond, Rotics und Szeider [FRRS06]). *Das Problem* CLIQUENWEITE *ist* NP*-vollständig.* **ohne Beweis**

[53] Die Distanz zweier Knoten wurde in Definition 3.3 eingeführt.

Für jede feste natürliche Zahl $k \leq 3$ kann man in polynomieller Zeit entscheiden, ob ein gegebener Graph eine Cliquenweite von höchstens k hat, und im positiven Fall kann man auch einen Cliquenweite-Ausdruck angeben. Für $k = 1$ und $k = 2$ ist dies nach Satz 11.18 klar. Die Lösung für $k = 3$ ist deutlich aufwändiger (siehe die Arbeit von Corneil, Habib, Lanlignel, Reed und Rotics [CHL+00]). Hingegen ist für jede feste natürliche Zahl $k \geq 4$ die Zeitkomplexität des Problems, zu entscheiden, ob ein gegebener Graph eine Cliquenweite von höchstens k hat, noch offen.

Falls man sich auf Graphklassen mit beschränkter Baumweite einschränkt, so kann man für jede feste natürliche Zahl k in linearer Zeit entscheiden, ob ein gegebener Graph eine Cliquenweite von höchstens k hat (Espelage, Gurski und Wanke [EGW03]). Aufgrund von Satz 11.12 kann man somit offenbar die Cliquenweite eines gegebenen Graphen mit beschränkter Baumweite in linearer Zeit berechnen.

p-CLIQUENWEITE

p-CLIQUENWEITE	
Gegeben:	Ein Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$.
Parameter:	k.
Frage:	Hat G eine Cliquenweite von höchstens k?

Es ist ein offenes Problem, ob das parametrisierte Problem p-CLIQUENWEITE fest-Parameter-berechenbar ist. Zum Problem, Cliquenweite-Ausdrücke für einen gegebenen Graphen zu finden, gibt es bislang nur die folgende Aussage.

Satz 11.22 (Oum [Oum08]). *Für jede natürliche Zahl k gibt es einen Algorithmus, der zu einem gegebenen Graphen $G = (V, E)$ in der Zeit $\mathcal{O}(|V|^3)$ entweder einen Cliquenweite-$(8^k - 1)$-Ausdruck liefert oder aber entscheidet, dass G eine Cliquenweite größer als k hat.* **ohne Beweis**

Einige eng mit der Cliquenweite verwandte Parameter betrachten wir in den Definitionen 11.23, 11.26, 11.29 und 11.32.

Definition 11.23 (NLC-Weite).

NLC$_k$
k-markierter Graph $\bullet_i$

- *Die Graphklasse* NLC$_k$ *ist rekursiv wie folgt definiert:*
  1. *Der* k-markierte Graph $\bullet_i$ *ist für* $i \in [k]$ *in* NLC$_k$.
  2. *Es seien* $G = (V_G, E_G, lab_G) \in \mathrm{NLC}_k$ *und* $J = (V_J, E_J, lab_J) \in \mathrm{NLC}_k$ *zwei knotendisjunkte k-markierte Graphen und* $S \subseteq [k]^2$ *eine Relation. Dann ist auch der* k-markierte Graph

k-markierter Graph $G \times_S J$

$$G \times_S J = (V', E', lab')$$

in NLC$_k$, *der definiert ist durch*

$$\begin{aligned} V' &= V_G \cup V_J, \\ E' &= E_G \cup E_J \cup \{\{u, v\} \mid u \in V_G,\ v \in V_J,\ (lab_G(u), lab_J(v)) \in S\} \quad \textit{und} \end{aligned}$$

$$lab'(u) = \begin{cases} lab_G(u), & \textit{falls } u \in V_G, \\ lab_J(u), & \textit{falls } u \in V_J, \end{cases}$$

für alle $u \in V'$.

3. *Es seien $G = (V_G, E_G, lab_G) \in \mathrm{NLC}_k$ ein k-markierter Graph und $R : [k] \to [k]$ eine totale Funktion. Dann ist auch der* k-markierte Graph

k-markierter Graph $\circ_R(G)$

$$\circ_R(G) = (V_G, E_G, lab')$$

in NLC_k*, dessen Markierungsfunktion definiert ist durch $lab'(u) = R(lab(u))$ für alle $u \in V_G$.*

- *Die* NLC-Weite eines markierten Graphen *G ist die kleinste natürliche Zahl k, sodass G in* NLC_k *ist.*
- *Die* NLC-Weite eines Graphen *$G = (V, E)$ (kurz* NLC-Weite(G)*) ist die kleinste natürliche Zahl k, für die es eine Abbildung $\ell : V \to [k]$ gibt, sodass der k-markierte Graph (V, E, ℓ) eine NLC-Weite von höchstens k hat.*

NLC-Weite eines markierten Graphen

NLC-Weite eines Graphen

Der folgende Satz stellt einen Zusammenhang zwischen der Cliquenweite und der NLC-Weite von Graphen her.

Satz 11.24. *Für jeden Graphen G gilt:*

$$\text{NLC-Weite}(G) \leq \text{Cliquenweite}(G) \leq 2 \cdot \text{NLC-Weite}(G).$$

ohne Beweis

Übung 11.25. Zeigen Sie Satz 11.24.

Definition 11.26 (Rangweite).

- *Eine* Rangdekomposition eines Graphen *G ist ein Paar (T, f), wobei T ein binärer Wurzelbaum ist und f eine Bijektion zwischen den Knoten in G und den Blättern in T.*
- *Der* Rang einer Kante *e* aus *T ist der Rang der* 0-1-*Adjazenzmatrix $M_e = M_{A,B}$, wobei A und B über die Blätter der beiden Teilbäume $T - \{e\}$ bestimmt sind, d. h., $T - \{e\}$ sind die beiden Teilbäume von T, die durch Entfernen der Kante e aus T entstehen.*
- *Die* Weite einer Rangdekomposition *(T, f) ist der maximale Rang aller Kanten in T.*
- *Die* Rangweite eines Graphen *G, kurz* Rangweite(G)*, ist die minimale Weite aller Rangdekompositionen für G.*

Rangdekomposition

Rang einer Kante

Weite einer Rangdekomposition

Rangweite eines Graphen

Der folgende Satz stellt einen Zusammenhang zwischen der Cliquenweite und der Rangweite von Graphen her.

Satz 11.27. *Für jeden Graphen G gilt:*

$$\text{Rangweite}(G) \leq \text{Cliquenweite}(G) \leq 2^{\text{Rangweite}(G)+1} - 1.$$

ohne Beweis

Übung 11.28. Zeigen Sie Satz 11.27.

Nun führen wir den Begriff der modularen Weite von Graphen ein.

Definition 11.29 (modulare Weite).

k-Modul
homogene k-Menge

- *Sei $G = (V,E)$ ein Graph. Eine Knotenmenge $S \subseteq V$ heißt* k-Modul *(bzw.* homogene k-Menge*), falls S in höchstens k Mengen $S_1, \ldots, S_k$ eingeteilt werden kann, sodass jede Menge S_i, $i \in [k]$, ein Modul*[54] *im Graphen $G[(V-S) \cup S_i]$ bildet.*

k-modulare Dekomposition

- *Eine* k-modulare Dekomposition *für einen Graphen G ist ein binärer Wurzelbaum T, sodass die Blätter von T den Knoten von G zugeordnet werden können und für jeden Knoten v in T die Blätter im Teilbaum mit der Wurzel v ein k-Modul von G bilden.*

modulare Weite

- *Die* modulare Weite *eines Graphen G, kurz* Mod-Weite(G)*, ist definiert als die kleinste natürliche Zahl k, sodass es eine k-modulare Dekomposition für G gibt.*

Der folgende Satz stellt einen Zusammenhang zwischen der Cliquenweite, der NLC-Weite und der modularen Weite von Graphen her.

Satz 11.30. *Für jeden Graphen G gilt:*

$$\text{Mod-Weite}(G) \leq \text{NLC-Weite}(G) \leq \text{Cliquenweite}(G) \leq 2 \cdot \text{Mod-Weite}(G).$$

ohne Beweis

Übung 11.31. Zeigen Sie Satz 11.30.

Definition 11.32 (boolesche Weite).

boolesche Dekomposition

- *Eine* boolesche Dekomposition *für einen Graphen G ist ein Paar (T, f), wobei T ein binärer Wurzelbaum ist und f eine Bijektion zwischen den Knoten in G und den Blättern in T.*

Weite einer Kante

- *Offenbar definiert jede Kante e in T eine disjunkte Partition der Knoten von G in genau zwei Mengen, $\{A_e, \overline{A_e}\}$, die sich über f aus den Blättern von T ergeben, falls e aus T entfernt wird. Die* Weite einer Kante e aus T *definieren wir als*

$$\mathit{Weite}(e) = \log_2(|\{S \subseteq \overline{A_e} \mid \text{es gibt eine Menge } X \subseteq A_e \text{ mit } S = \overline{A_e} \cap \bigcup_{x \in X} N(x)\}|).$$

Weite einer booleschen Dekomposition
boolesche Weite

- *Die* Weite einer booleschen Dekomposition (T, f) *ist die maximale Weite aller Kanten in T.*
- *Die* boolesche Weite eines Graphen G*, kurz* Boole-Weite(G)*, ist die minimale Weite aller booleschen Dekompositionen für G.*

Der folgende Satz stellt einen Zusammenhang zwischen der Cliquenweite und der booleschen Weite von Graphen her.

Modul eines Graphen

[54] Für einen Graphen $G = (V,E)$ heißt eine Knotenmenge $M \subseteq V$ *Modul von G*, falls für alle $(v_1, v_2) \in M \times M$ die Bedingung $N(v_1) - M = N(v_2) - M$ gilt, d. h., falls je zwei Knoten v_1 und v_2 die gleiche Nachbarschaft außerhalb von M haben.

Satz 11.33. *Für jeden Graphen G gilt:*

$$\text{Boole-Weite}(G) \leq \text{Cliquenweite}(G) \leq 2^{\text{Boole-Weite}(G)+1}.$$

ohne Beweis

Übung 11.34. Zeigen Sie Satz 11.33.

Anmerkung 11.35. Im Folgenden parametrisieren wir unsere Entscheidungsprobleme wie UNABHÄNGIGE MENGE und CLIQUE in der Cliquenweite des Eingabegraphen. Streng genommen ist dies kein zulässiger Parameter, da sich der Parameter „Cliquenweite des Eingabegraphen" nicht in polynomieller Zeit aus dem Eingabegraphen bestimmen lässt (es sei denn, es würde $\mathrm{P} = \mathrm{NP}$ gelten). Deshalb verwenden wir hier wieder den Zusatz p^* in der Problembezeichnung.

Alternativ erhält man eine in Polynomialzeit berechenbare Parametrisierung eines eingeschränkten Problems, wenn man die Eingabe auf Graphen mit einer Cliquenweite von höchstens k einschränkt und k als Parameter wählt.

11.2 Unabhängige Menge und Knotenüberdeckung

Als Erstes betrachten wir das Problem, in einem gegebenen Graphen G die Größe einer kardinalitätsmaximalen unabhängigen Menge zu bestimmen, d. h., wir bestimmen die Unabhängigkeitszahl $\alpha(G)$. Als Parametrisierung wählen wir hier die Cliquenweite des Eingabegraphen.

p^*-cw-UNABHÄNGIGE MENGE

p^*-cw-UNABHÄNGIGE MENGE	
Gegeben:	Ein Graph $G = (V,E)$ und eine Zahl $s \in \mathbb{N}$.
Parameter:	Cliquenweite(G).
Frage:	Gibt es in G eine unabhängige Menge der Größe mindestens s?

Die Lösungsidee beruht auf einer dynamischen Programmierung entlang eines Cliquenweite-k-Ausdrucksbaums für den Eingabegraphen. Dazu benötigen wir einige Notationen. Es sei $G = (V_G, E_G)$ ein Graph mit einer Cliquenweite von höchstens k und $T = (V_T, E_T)$ ein Cliquenweite-k-Ausdrucksbaum mit Wurzel r für G. Für einen Knoten $u \in V_T$ definieren wir T_u als den Teilbaum von T mit Wurzel u. Offenbar ist für jeden Knoten $u \in V_T$ der Baum T_u ein Cliquenweite-k-Ausdrucksbaum. Weiterhin sei G_u der durch den Ausdrucksbaum T_u definierte Teilgraph von G.

Die Idee beim Algorithmenentwurf entlang von Cliquenweite-k-Ausdrucksbäumen T beruht häufig auf der Eigenschaft, dass sich für jeden Knoten u in T die Knoten im durch den Ausdrucksbaum T_u definierten Teilgraphen G_u in k Mengen einteilen lassen, sodass alle Knoten einer Menge die gleichen Nachbarn im noch folgenden Aufbau des Gesamtgraphen erhalten. Somit kann man sich bei Lösungsansätzen mittels dynamischer Programmierung entlang von Cliquenweite-k-Ausdrucksbäumen bei der Bestimmung von Teillösungen häufig auf die höchstens k Markierungen der Knoten des bisher aufgebauten Teilgraphen beschränken.

Zur Lösung von p^*-cw-UNABHÄNGIGE MENGE verwenden wir für jeden Knoten u von T eine Datenstruktur $F(u)$, nämlich ein $(2^k - 1)$-Tupel

$$F(u) = (a_{\{1\}}, \ldots, a_{\{1,\ldots,k\}}).$$

Jedes dieser Tupel enthält für jede Teilmenge $L \subseteq [k]$, $L \neq \emptyset$, eine positive Zahl a_L, die die Größe einer größten unabhängigen Menge I im Graphen G_u beschreibt, sodass $\{\mathrm{lab}_{G_u}(i) \mid i \in I\} = L$ gilt.

Beispiel 11.36 (Datenstruktur *F*(*u*)). Es sei G_u der in Abb. 11.2 dargestellte 3-markierte Graph.

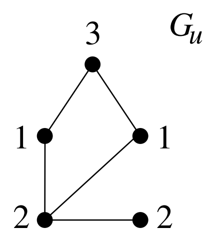


Abb. 11.2. Ein Beispiel zur Datenstruktur für p^*-cw-UNABHÄNGIGE MENGE

Dann ist $F(u) = (a_{\{1\}}, a_{\{2\}}, a_{\{3\}}, a_{\{1,2\}}, a_{\{1,3\}}, a_{\{2,3\}}, a_{\{1,2,3\}})$ mit der folgenden Zuordnung (L, a_L):

L	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
a_L	2	1	1	3	0	2	0

Es ist offensichtlich, dass für jeden Knoten $u \in V_T$ die Größe der Mengen $F(u)$ unabhängig von der Größe des Graphen G_u in k beschränkt werden kann, da $F(u)$ aus genau $2^k - 1$ Zahlen besteht.

Somit lässt sich unser Algorithmus für p^*-cw-UNABHÄNGIGE MENGE wie folgt formulieren:

1. Bestimme einen Cliquenweite-Ausdruck X für den Eingabegraphen $G = (V, E)$ (siehe Satz 11.22).
2. Transformiere diesen Ausdruck X in einen Cliquenweite-Ausdrucksbaum T mit Wurzel r.
3. Mittels dynamischer Programmierung entlang des Baums T mit der Wurzel r kann $F(r)$ durch einen Durchlauf in Bottom-up-Reihenfolge mit der folgenden Aktion auf den Knoten berechnet werden:
   a) Falls u ein mit $\bullet_i$, $i \in [k]$, markiertes Blatt in T ist, definieren wir

$$F(u) = (a_{\{1\}}, \ldots, a_{\{1,\ldots,k\}}),$$

wobei für alle nicht leeren Mengen $L \subseteq [k]$ gilt:

$$a_L = \begin{cases} 1, & \text{falls } L = \{i\}, \\ 0, & \text{falls } L \neq \{i\}. \end{cases}$$

b) Falls u ein Vereinigungsknoten mit den Kindern v und w in T ist, sind $F(v) = (a_{\{1\}}, \ldots, a_{\{1,\ldots,k\}})$ und $F(w) = (b_{\{1\}}, \ldots, b_{\{1,\ldots,k\}})$ bereits bekannt. Wir definieren

$$F(u) = (c_{\{1\}}, \ldots, c_{\{1,\ldots,k\}}),$$

wobei

$$c_L = \max_{L = L_1 \cup L_2} a_{L_1} + b_{L_2}$$

für $L_1, L_2 \subseteq [k]$ gilt.

c) Falls u ein mit $\eta_{i,j}$ markierter Kanteneinfügeknoten mit dem Kind v in T ist, ist $F(v) = (a_{\{1\}}, \ldots, a_{\{1,\ldots,k\}})$ bereits bekannt. Wir definieren

$$F(u) = (b_{\{1\}}, \ldots, b_{\{1,\ldots,k\}}),$$

wobei für alle nicht leeren Mengen $L \subseteq [k]$ gilt:

$$b_L = \begin{cases} a_L, & \text{falls } \{i,j\} \nsubseteq L, \\ 0, & \text{falls } \{i,j\} \subseteq L. \end{cases}$$

Falls sowohl i als auch j in der Markierungsmenge L einer unabhängigen Menge liegen, ist diese Menge nach der Kanteneinfügung nicht mehr unabhängig. Falls höchstens ein Element aus $\{i,j\}$ in L liegt, bleibt L unabhängig.

d) Falls u ein mit $\rho_{i \to j}$ markierter Ummarkierungsknoten mit dem Kind v in T ist, ist $F(v) = (a_{\{1\}}, \ldots, a_{\{1,\ldots,k\}})$ bereits bekannt. Wir definieren in diesem Fall

$$F(u) = (b_{\{1\}}, \ldots, b_{\{1,\ldots,k\}}),$$

wobei für alle nicht leeren Mengen $L \subseteq [k]$ gilt:

$$b_L = \begin{cases} a_L, & \text{falls } i \notin L \text{ und } j \notin L, \\ \max\{a_L, a_{L \cup \{i\}}, a_{(L \cup \{i\}) - \{j\}}\}, & \text{falls } i \notin L \text{ und } j \in L, \\ 0, & \text{falls } i \in L. \end{cases}$$

Falls $i \notin L$ und $j \notin L$ gilt, hat sich die Menge der Knoten in G_u mit einer Markierung aus L nicht verändert, d. h., es gilt $b_L = a_L$.

Falls $i \notin L$ und $j \in L$ gilt, kann ein j-markierter Knoten in G_u vor der Ummarkierung in G_v schon mit j markiert gewesen sein oder aus einem mit i markierten Knoten in G_v entstehen. Im zweiten Fall können die schon in G_v

mit j markierten Knoten mit in L betrachtet werden oder nicht. Somit ergibt sich

$$b_L = \max\{a_L, a_{L\cup\{i\}}, a_{(L\cup\{i\})-\{j\}}\}.$$

Falls $i \in L$ gilt, gibt es in G_u offensichtlich keinen Knoten, der mit i markiert ist, d. h., es gilt $b_L = 0$.

4. Mittels $F(r)$ können wir die Unabhängigkeitszahl des durch T_r definierten Graphen G bestimmen durch
$$\alpha(G) = \max_{a\in F(r)} a.$$

Satz 11.37. *Die Unabhängigkeitszahl eines Graphen $G = (V,E)$ mit einer Cliquenweite von höchstens k kann für eine berechenbare Funktion f in der Zeit*

$$\mathcal{O}(|V| \cdot f(k))$$

bestimmt werden, falls G durch einen Cliquenweite-k-Ausdruck gegeben ist.

Beweis. Es seien $G = (V,E)$ ein Graph mit einer Cliquenweite von höchstens k und X ein zugehöriger Cliquenweite-Ausdruck, der einen Ausdrucksbaum T defininiert. In Anmerkung 11.10 haben wir bereits erläutert, dass es in (der Normalform von) T genau $|V|$ Blätter, $|V| - 1$ Vereinigungsknoten, höchstens $(|V| - 1)(k - 1)$ Ummarkierungsknoten und höchstens $(|V| - 1)(k(k-1)/2)$ Kanteneinfügeknoten gibt.

Da die Mengen $F(u)$ in der Zeit $\mathcal{O}(2^{k+1})$ initialisiert bzw. aus den Tupeln des Kindes bzw. der Kinder von u bestimmt werden können, folgt die Behauptung. ❑

Korollar 11.38. *Das Problem p^*-cw-*UNABHÄNGIGE MENGE *ist fest-Parameter-berechenbar.*

Korollar 11.39. *Das Problem* UNABHÄNGIGE MENGE *kann für jeden Graphen $G = (V,E)$ einer Graphklasse mit beschränkter Cliquenweite (z. B. der Klasse der Co-Graphen, der distanzerhaltenden Graphen usw.) in der Zeit $\mathcal{O}(|V|)$ gelöst werden.*

p^*-cw-KNOTEN-ÜBERDECKUNG

Das Problem KNOTENÜBERDECKUNG lässt sich mittels der Gleichung (3.4) von Gallai sehr leicht auf das Problem UNABHÄNGIGE MENGE zurückführen.

p^*-cw-KNOTENÜBERDECKUNG	
Gegeben:	Ein Graph $G = (V,E)$ und eine Zahl $s \in \mathbb{N}$.
Parameter:	Cliquenweite(G).
Frage:	Gibt es in G eine Knotenüberdeckung der Größe höchstens s?

Korollar 11.40. *Die Größe einer kardinalitätsminimalen Knotenüberdeckung eines Graphen $G = (V,E)$ mit einer Cliquenweite von höchstens k kann in der Zeit*

$$\mathcal{O}(|V| \cdot f(k))$$

für eine berechenbare Funktion f bestimmt werden.

Korollar 11.41. *Das Problem p^*-cw-*KNOTENÜBERDECKUNG *ist fest-Parameter-berechenbar.*

Übung 11.42. Modifizieren Sie die in Abschnitt 11.2 angegebene Lösung für das Problem p^*-cw-UNABHÄNGIGE MENGE, um eine Lösung für das folgende gewichtete Problem zu erhalten:

p^*-cw-GEWICHTETE UNABHÄNGIGE MENGE

p^*-cw-GEWICHTETE UNABHÄNGIGE MENGE	
Gegeben:	Ein Graph $G = (V,E)$, eine Gewichtsfunktion $c : V \to \mathbb{N}$ und eine Zahl $s \in \mathbb{N}$.
Parameter:	Cliquenweite(G).
Frage:	Gibt es in G eine unabhängige Menge $V' \subseteq V$ mit $\sum_{v \in V'} c(v) \geq s$?

11.3 Clique

Als Nächstes betrachten wir das Problem, in einem gegebenen Graphen G die Größe einer kardinalitätsmaximalen Clique zu bestimmen, d. h., wir bestimmen die Cliquenzahl $\omega(G)$. Wieder ist die Cliquenweite des Eingabegraphen unsere Parametrisierung.

p^*-cw-CLIQUE

p^*-cw-CLIQUE	
Gegeben:	Ein Graph $G = (V,E)$ und eine Zahl $s \in \mathbb{N}$.
Parameter:	Cliquenweite(G).
Frage:	Gibt es in G eine Clique der Größe mindestens s?

Satz 11.43. *Die Cliquenzahl eines Graphen $G = (V,E)$ mit einer Cliquenweite von höchstens k kann für eine berechenbare Funktion f in der Zeit*

$$\mathcal{O}(|V| \cdot f(k))$$

bestimmt werden, falls G durch einen Cliquenweite-k-Ausdruck gegeben ist.

Beweis. Falls G eine Cliquenweite von höchstens k hat, konstruieren wir den Komplementgraphen von G, der nach Satz 11.11 eine Cliquenweite von höchstens $2k$ hat, und wenden auf diesen den Algorithmus aus Abschnitt 11.2 an. Die Korrektheit folgt aus Gleichung (3.3). ❑

Korollar 11.44. *Das Problem p^*-cw-*CLIQUE *ist fest-Parameter-berechenbar.*

Korollar 11.45. *Das Problem* CLIQUE *kann für jeden Graphen $G = (V,E)$ einer Graphklasse mit beschränkter Cliquenweite (z. B. der Klasse der Co-Graphen, der distanzerhaltenden Graphen usw.) in der Zeit $\mathcal{O}(|V|)$ gelöst werden.*

11.4 Partition in unabhängige Mengen

p^*-cw-PARTITION IN UNABHÄNGIGE MENGEN

Weiterhin betrachten wir das Problem, die Knotenmenge in einem gegebenen Graphen G in möglichst wenige unabhängige Mengen zu partitionieren, d. h., wir bestimmen die Färbungszahl $\chi(G)$. Als Parametrisierung wählen wir erneut die Cliquenweite des Eingabegraphen.

p^*-cw-PARTITION IN UNABHÄNGIGE MENGEN	
Gegeben:	Ein Graph $G = (V, E)$ und eine Zahl $s \in \mathbb{N}$.
Parameter:	Cliquenweite(G).
Frage:	Gibt es eine Partition von V in s unabhängige Mengen?

Multimenge

In der Datenstruktur zur Lösung des Problems verwenden wir *Multimengen*, d. h. Mengen, die mehrere gleiche Elemente enthalten können. Für eine Multimenge $\mathscr{M}$ mit den Elementen $x_1, \ldots, x_n$ schreiben wir

$$\mathscr{M} = \langle x_1, \ldots, x_n \rangle.$$

Beispielsweise hat die Multimenge $\langle 1,2,2,3,4,4,4 \rangle$ sieben Elemente und nicht vier.

Die Elemente von $\mathscr{M}$ haben keine Ordnung. Die Häufigkeit eines Elements x in einer Multimenge $\mathscr{M}$ wird mit $\psi(\mathscr{M}, x)$ bezeichnet, z. B. ist

$$\psi(\langle 1,2,2,3,4,4,4 \rangle, 4) = 3.$$

Zwei Multimengen $\mathscr{M}_1$ und $\mathscr{M}_2$ heißen *gleich*, falls für jedes Element $x \in \mathscr{M}_1 \cup \mathscr{M}_2$ gilt:

$$\psi(\mathscr{M}_1, x) = \psi(\mathscr{M}_2, x);$$

andernfalls heißen sie *verschieden*. Die leere Multimenge bezeichnen wir mit $\langle \rangle$.

Zur Lösung von p^*-cw-PARTITION IN UNABHÄNGIGE MENGEN für einen Graphen G mit Cliquenweite-Ausdrucksbaum T verwenden wir für jeden Knoten u in T eine Datenstruktur $F(u)$. Für jede Partition der Knotenmenge von G_u in unabhängige Mengen $V_1, \ldots, V_r$, $1 \leq r \leq |V|$, enthält $F(u)$ die Multimenge $\langle \mathrm{lab}(V_1), \ldots, \mathrm{lab}(V_r) \rangle$.

Anschaulich kann man die Anzahl der möglichen verschiedenen Multimengen in $F(u)$ wie folgt darstellen. Jede Multimenge $\mathscr{M}$ enthält Teilmengen von $[k]$, die in der ersten Zeile der folgenden Tabelle angegeben sind. Die Zeile darunter gibt jeweils Anzahl an, wie oft eine Teilmenge in $\mathscr{M}$ vorkommt.

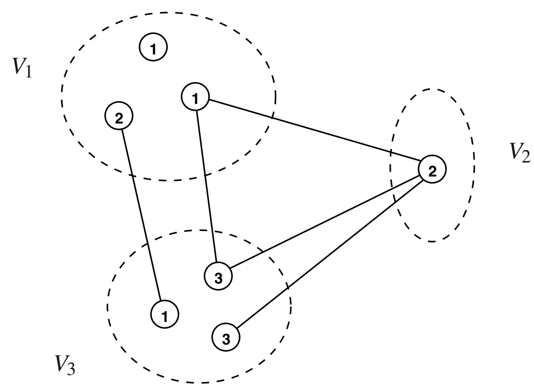
$$\underbrace{\begin{array}{cccccccc}
\{1\} & \cdots & \{k\} & \{1,2\} & \cdots & \{k-1,k\} & \cdots & \{1,\ldots,k\} \\ \hline
0 & & 0 & 0 & & 0 & & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
|V| & & |V| & |V| & & |V| & & |V| \\ \hline
(|V|+1) & \cdots & (|V|+1) & (|V|+1) & \cdots & (|V|+1) & \cdots & (|V|+1)
\end{array}}_{2^k-1} = (|V|+1)^{2^k-1}$$

Somit enthält $F(u)$ höchstens

$$(|V|+1)^{2^k-1}$$

paarweise verschiedene Multimengen, weshalb die Größe von $F(u)$ durch ein Polynom in der Größe von G beschränkt werden kann.

Beispiel 11.46 (Datenstruktur $F(u)$). Abbildung 11.3 zeigt eine Partition eines Graphen G_u in drei unabhängige Mengen und die dazugehörige Multimenge $\mathscr{M} \in F(u)$.



$$F(u) = \{\ \ldots, \langle\ \{1,2\}, \{2\}, \{1,3\}\ \rangle,\ \ldots\ \}$$

Abb. 11.3. Ein Beispiel zur Datenstruktur für p^*-cw-PARTITION IN UNABHÄNGIGE MENGEN. Die Knotenmenge des Graphen G ist in drei unabhängige Mengen V_1, V_2 und V_3 aufgeteilt. $F(u)$ enthält die dazugehörige Multimenge $\langle \text{lab}(V_1), \text{lab}(V_2), \text{lab}(V_3)\rangle$.

Somit lässt sich unser Algorithmus für p^*-cw-PARTITION IN UNABHÄNGIGE MENGEN wie folgt formulieren:

1. Bestimme einen Cliquenweite-Ausdruck X für den Eingabegraphen $G = (V,E)$ (siehe Satz 11.22).
2. Transformiere diesen Ausdruck X in einen Cliquenweite-Ausdrucksbaum T mit Wurzel r.
3. Mittels dynamischer Programmierung entlang des Baums T mit der Wurzel r kann $F(r)$ durch einen Durchlauf in Bottom-up-Reihenfolge mit der folgenden Aktion auf den Knoten berechnet werden:
   a) Falls u ein mit $\bullet_i$, $i \in [k]$, markiertes Blatt in T ist, definieren wir
   $$F(u) = \{\langle\{i\}\rangle\}.$$
   b) Falls u ein Vereinigungsknoten mit den Kindern v und w in T ist, sind $F(v)$ und $F(w)$ bereits bekannt.

Ausgehend von der Menge $D = \{\langle\rangle\} \times F(v) \times F(w)$ wird D um alle Tripel erweitert, die man aus einem Tripel $(\mathscr{M}, \mathscr{M}', \mathscr{M}'') \in D$ erhalten kann, indem man entweder eine Menge L' aus $\mathscr{M}'$ oder eine Menge L'' aus $\mathscr{M}''$ entfernt und die Menge L' bzw. L'' in $\mathscr{M}$ einfügt oder indem man eine Menge L' aus $\mathscr{M}'$ und eine Menge L'' aus $\mathscr{M}''$ entfernt und die Menge $L' \cup L''$ in $\mathscr{M}$ einfügt. D enthält höchstens

$$(|V|+1)^{3(2^k-1)}$$

Tripel und ist somit in polynomieller Zeit berechenbar. Demgemäß definieren wir

$$F(u) = \{\mathscr{M} \mid (\mathscr{M}, \langle\rangle, \langle\rangle) \in D\}.$$

c) Falls u ein mit $\eta_{i,j}$ markierter Kanteneinfügeknoten mit einem Kind v in T ist, ist $F(v)$ bereits bekannt. Wir definieren

$$F(u) = \{\langle L_1, \ldots, L_r\rangle \in F(v) \mid \{i, j\} \not\subseteq L_t \text{ für alle } t \in \{1, \ldots, r\}\}.$$

Falls sowohl i als auch j in der Markierungsmenge L mindestens einer unabhängigen Menge einer Partition der Knotenmenge von G_u liegen, ist diese Menge nach der Kanteneinfügung nicht mehr unabhängig, und deshalb werden die entsprechenden Partitionen nicht in $F(u)$ aufgenommen.

d) Falls u ein mit $\rho_{i\to j}$ markierter Ummarkierungsknoten mit einem Kind v in T ist, ist $F(v)$ bereits bekannt. Wir definieren

$$F(u) = \{\langle \rho_{i\to j}(L_1), \ldots, \rho_{i\to j}(L_r)\rangle \mid \langle L_1, \ldots, L_r\rangle \in F(v)\}.$$

4. Mittels $F(r)$ können wir die Färbungszahl des durch T_r definierten Graphen G bestimmen durch

$$\chi(G) = \min_{\mathscr{M} \in F(r)} |\mathscr{M}|.$$

Satz 11.47. *Die Färbungszahl eines Graphen $G = (V, E)$ mit einer Cliquenweite von höchstens k kann für eine berechenbare Funktion f in der Zeit*

$$\mathscr{O}(|V|^{f(k)})$$

bestimmt werden, falls G durch einen Cliquenweite-k-Ausdruck gegeben ist.

Beweis. Es seien $G = (V, E)$ ein Graph mit einer Cliquenweite von höchstens k und X ein zugehöriger Cliquenweite-Ausdruck, der einen Ausdrucksbaum T definiert. In Anmerkung 11.10 haben wir bereits erläutert, dass es in (der Normalform von) T genau $|V|$ Blätter, $|V|-1$ Vereinigungsknoten, höchstens $(|V|-1)(k-1)$ Ummarkierungsknoten und höchstens $(|V|-1)(k(k-1)/2)$ Kanteneinfügeknoten gibt.

Da die Mengen $F(u)$ in der Zeit $\mathscr{O}(|V|^{f(k)})$ initialisiert bzw. aus den Tupeln des Kindes bzw. der Kinder von u bestimmt werden können, folgt die Behauptung. ❑

Übung 11.48. Geben Sie die in Satz 11.47 erwähnte berechenbare Funktion f in Abhängigkeit von k an.

Korollar 11.49. *Das Problem p^*-cw-*PARTITION IN UNABHÄNGIGE MENGEN *ist in* XP.

Korollar 11.50. *Das Problem* PARTITION IN UNABHÄNGIGE MENGEN *kann für jeden Graphen $G = (V,E)$ einer Graphklasse mit beschränkter Cliquenweite (z. B. der Klasse der Co-Graphen, der distanzerhaltenden Graphen usw.) in der Zeit $\mathcal{O}(|V|^{\mathcal{O}(1)})$ gelöst werden.*

Es stellt sich die Frage, ob p^*-cw-PARTITION IN UNABHÄNGIGE MENGEN sogar in FPT liegt. Dies gilt jedoch nicht, es sei denn, es würde $\mathrm{W}[1] = \mathrm{FPT}$ gelten.

Satz 11.51 (Fomin, Golovach, Lokshtanov und Saurabh [FGLS09]). *Das Problem p^*-cw-*PARTITION IN UNABHÄNGIGE MENGEN *ist* $\mathrm{W}[1]$*-schwer.* **ohne Beweis**

11.5 Partition in Cliquen

Schließlich betrachten wir das Problem, die Knotenmenge in einem gegebenen Graphen G in möglichst wenige Cliquen zu partitionieren, d. h., wir bestimmen die Cliquenüberdeckungszahl $\theta(G)$. Wieder wählen wir die Cliquenweite des Eingabegraphen als Parametrisierung.

p^*-cw-PARTITION IN CLIQUEN

p^*-cw-PARTITION IN CLIQUEN	
Gegeben:	Ein Graph $G = (V,E)$ und eine Zahl $s \in \mathbb{N}$.
Parameter:	Cliquenweite(G).
Frage:	Gibt es eine Partition von V in s Cliquen?

Satz 11.52. *Die Cliquenüberdeckungszahl eines Graphen $G = (V,E)$ mit einer Cliquenweite von höchstens k kann für eine berechenbare Funktion f in der Zeit*

$$\mathcal{O}(|V|^{f(k)})$$

bestimmt werden, falls G durch einen Cliquenweite-k-Ausdruck gegeben ist.

Beweis. Falls G eine Cliquenweite von höchstens k hat, konstruieren wir den Komplementgraphen von G, der nach Satz 11.11 eine Cliquenweite von höchstens $2k$ hat, und wenden auf diesen den Algorithmus aus Abschnitt 11.4 an. Die Korrektheit folgt aus Gleichung (3.9). ❑

Korollar 11.53. *Das Problem p^*-cw-*PARTITION IN CLIQUEN *ist in* XP.

Korollar 11.54. *Das Problem* PARTITION IN CLIQUEN *kann für jeden Graphen* $G = (V,E)$ *einer Graphklasse mit beschränkter Cliquenweite (z. B. der Klasse der Co-Graphen, der distanzerhaltenden Graphen usw.) in der Zeit* $\mathcal{O}(|V|^{\mathcal{O}(1)})$ *gelöst werden.*

Es stellt sich wieder die Frage, ob p^*-cw-PARTITION IN CLIQUEN sogar in FPT liegt. Dies gilt jedoch nicht, es sei denn, $\mathrm{W}[1] = \mathrm{FPT}$ würde gelten.

Korollar 11.55. *Das Problem* p^**-cw-*PARTITION IN CLIQUEN *ist* W[1]*-schwer.*

Beweis. Wenn das Problem p^*-cw-PARTITION IN CLIQUEN in FPT liegt, dann bedeutet dies, dass es einen FPT-Algorithmus A für p^*-cw-PARTITION IN CLIQUEN gibt, sodass A eine Laufzeit von $\mathcal{O}(f(k) \cdot |I|^{\mathcal{O}(1)})$ hat. Aufgrund von Gleichung (3.9) könnte man durch A, angewandt auf den Komplementgraphen der Eingabe, auch das Problem p^*-cw-PARTITION IN UNABHÄNGIGE MENGEN durch einen FPT-Algorithmus lösen. ❑

11.6 $\mathrm{MSO_1}$-definierbare Grapheigenschaften

Ähnlich wie in Abschnitt 10.6 gibt es auch eine grundlegende Aussage über die Existenz effizienter Algorithmen für Graphen mit beschränkter Cliquenweite. Courcelle, Makowsky und Rotics [CMR00] zeigten, dass alle Grapheigenschaften, die sich in monadischer Logik zweiter Ordnung (siehe Definition 4.39) auf relationalen Graphenstrukturen $\lfloor G \rfloor$ über der Signatur $\{E\}$ (siehe Beispiel 4.30) definieren lassen, auf Graphen G mit beschränkter Cliquenweite k in der Zeit $\mathcal{O}(f(k) \cdot |V|)$ entscheidbar sind, wenn ein Cliquenweite-k-Ausdruck für G gegeben ist. Die monadische Logik zweiter Ordnung lässt sich hier ebenfalls erweitern, sodass auch Optimierungsprobleme mit linearen Optimierungsfunktionen definiert werden können, und wird mit

$\mathrm{LinEMSO_1}$ LinEMSO$_1$ (englisch: *linear extended monadic second order logic*) bezeichnet.

Die oben genannte Aussage von Courcelle et al. [CMR00] unterscheidet sich jedoch in den beiden folgenden Punkten von der Aussage aus Abschnitt 10.6 über FPT-Algorithmen für Graphen mit beschränkter Baumweite.

Für Graphen mit beschränkter Cliquenweite werden Grapheigenschaften betrachtet, die sich auf den relationalen Graphenstrukturen $\lfloor G \rfloor$ über der Signatur $\{E\}$ definieren lassen, also $\mathrm{MSO_1}$-Grapheigenschaften. In Abschnitt 10.6 sind es dagegen $\mathrm{MSO_2}$-Grapheigenschaften. Viele interessante $\mathrm{MSO_2}$-Grapheigenschaften sind leider nicht $\mathrm{MSO_1}$-definierbar. Hierzu gehören zum Beispiel die Grapheigenschaften, die testen, ob ein Graph einen Hamilton-Kreis oder ein perfektes Matching besitzt, siehe Behauptung 4.41. Diese Graphenprobleme lassen sich trotzdem auf Graphen mit beschränkter Cliquenweite in polynomieller Zeit lösen, wenn ein Cliquenweite-k-Ausdruck gegeben ist, siehe [EGW01].

Ein weiterer wichtiger Unterschied ist dadurch gegeben, dass bis heute leider kein effizienter Algorithmus bekannt ist, der für einen gegebenen Graphen mit einer Cliquenweite von k einen Cliquenweite-k-Ausdruck berechnet. Es gibt jedoch für jedes k einen Approximationsalgorithmus A_k, der für einen gegebenen Graphen $G = (V,E)$ in der Zeit $\mathcal{O}(|V|^3)$ entweder einen Cliquenweite-k'-Ausdruck berechnet, mit $k' \leq 8^k - 1$, oder rückmeldet, dass die Cliquenweite von G größer als k ist (siehe Satz 11.22). Der Cliquenweite-k'-Ausdruck ist ausreichend für einen Polynomialzeit-Algorithmus zur Entscheidung einer MSO_1-Grapheigenschaft auf G. Der Approximationsalgorithmus A_k entscheidet nicht, ob ein gegebener Graph eine Cliquenweite von höchstens k hat, da er auch für Graphen mit einer Cliquenweite größer als k einen Cliquenweite-k'-Ausdruck ausgeben kann.

Es sind auch Probleme bekannt, die bezüglich des Parameters „Cliquenweite des Eingabegraphen“ nicht fest-Parameter-berechenbar sind (falls FPT $\neq$ W[1] gilt). Zwei dieser Probleme, p^*-cw-PARTITION IN UNABHÄNGIGE MENGEN und p^*-cw-PARTITION IN CLIQUEN, haben wir in Abschnitt 11.4 bzw. 11.5 bereits betrachtet. Zwei weitere Graphenpobleme sind die beiden folgenden:

p^*-cw-HAMILTON-KREIS

p^*-cw-HAMILTON-KREIS	
Gegeben:	Ein Graph $G = (V,E)$.
Parameter:	Cliquenweite(G).
Frage:	Gibt es in G einen Hamilton-Kreis?

p^*-cw-KANTENDOMINIERENDE MENGEN

p^*-cw-KANTENDOMINIERENDE MENGEN	
Gegeben:	Ein Graph $G = (V,E)$ und eine Zahl $s \in \mathbb{N}$.
Parameter:	Cliquenweite(G).
Frage:	Gibt es in G eine kantendominierende Menge E' (d. h., jede Kante aus E gehört entweder zu E' oder ist zu mindestens einer Kante aus E' inzident) der Größe höchstens s?

Satz 11.56 (Fomin, Golovach, Lokshtanov und Saurabh [FGLS09]). *Die Probleme*

- p^*-*cw*-HAMILTON-KREIS *und*
- p^*-*cw*-KANTENDOMINIERENDE MENGEN

sind W[1]-*schwer.*

Die Probleme p^*-cw-HAMILTON-KREIS und p^*-cw-KANTENDOMINIERENDE MENGEN können jedoch wie die beiden Probleme aus Abschnitt 11.4 und 11.5 mittels XP-Algorithmen gelöst werden, siehe [EGW01, KR03].

11.7 Literaturhinweise

Das Konzept der Cliquenweite wurde bereits im Jahre 1994 von Courcelle und Olariu [CO00] eingeführt. Ähnliche Parameter sind die von Wanke [Wan94] eingeführte

NLC-Weite, die von Oum und Seymour [OS06] definierte Rangweite und die modulare Weite, die auf Rao [Rao08] zurückgeht. Die boolesche Weite wurde erstmals von Bui-Xuan, Telle und Vatshelle in [BXTV09] definiert.

Auf Graphklassen mit beschränkter Cliquenweite eingeschränkte Algorithmen findet man zum Beispiel in den Arbeiten [EGW01, KR03, GK03, GW06]. Brandstädt et al. [BELL06] untersuchten alle Graphklassen, die sich durch den Ausschluss verbotener induzierter Teilgraphen mit höchstens vier Knoten definieren lassen, bezüglich ihrer Cliquenweite. Weiterhin wurden bereits alle Graphklassen, die sich durch den Ausschluss aller Erweiterungen des P_4 um einen Knoten definieren lassen, bezüglich ihrer Cliquenweite klassifiziert [BDLM05]. Die Cliquenweite von Graphen mit beschränktem Knotengrad wurde von Lozin und Rautenbach [LR04] betrachtet. Einen sehr umfangreichen Überblick über die Cliquenweite spezieller Graphklassen liefert die Arbeit von Kaminski, Lozin und Milanic [KLM09].

Analog zur Einschränkung der Baumweite in der Wegweite betrachtet man in der Literatur auch Einschränkungen der Cliquenweite und NLC-Weite, die durch den Zusatz „sequentiell" oder „linear" gekennzeichnet werden [FRRS06, GW05]. Erste Exponentialzeit-Algorithmen zur Bestimmung der sequentiellen Cliquenweite, sequentiellen NLC-Weite und NLC-Weite findet man in der Arbeit von Müller und Urner [MU10].

Eine interessante Übersichtsarbeit zur Cliquenweite und zu ähnlichen Graphparametern ist die Schrift von Hlinený, Oum, Seese und Gottlob [HOSG08].