10

Baumweitebeschränkte Graphen

Nun betrachten wir Parametrisierungen, die die Weite eines Graphen messen, wenn dieser in einer speziellen Baumstruktur repräsentiert wird. Entlang dieser Baumstruktur können viele an sich schwere Probleme auf Graphen mit beschränktem Parameter effizient im Sinne von FPT-Algorithmen gelöst werden.

10.1 Grundlagen

In diesem Abschnitt definieren wir so genannte Baumdekompositionen für Graphen. Die Güte einer solchen Dekomposition messen wir durch die Baumweite des zugehörigen Graphen. Der Begriff Baumweite (englisch: *treewidth*) wurde 1986 von Robertson und Seymour [RS86] eingeführt.

Definition 10.1 (Baumweite). *Es seien $G = (V_G, E_G)$ ein Graph, $T = (V_T, E_T)$ ein Baum und $\mathscr{X} = \{X_u \mid X_u \subseteq V_G,\ u \in V_T\}$ eine Menge von Teilmengen von V_G. Wir nennen das Paar $(\mathscr{X}, T)$ eine* Baumdekomposition *für G, wenn die folgenden drei Bedingungen erfüllt sind:* Baumdekomposition

1. $\bigcup_{u \in V_T} X_u = V_G$.
2. *Für jede Kante $\{w_1, w_2\} \in E_G$ gibt es einen Knoten $u \in V_T$, sodass $w_1 \in X_u$ und $w_2 \in X_u$.*
3. *Für jeden Knoten $w \in V_G$ ist der Teilgraph von T, der durch die Knoten $u \in V_T$ mit $w \in X_u$ induziert wird, zusammenhängend.*

Die Mengen $X_u \in \mathscr{X}$ werden als Taschen *(englisch:* bags*) bezeichnet.* Tasche

Die Weite einer Baumdekomposition $(\mathscr{X}, T)$ *ist die Zahl* Weite einer Baumdekomposition

$$\max_{u \in V_T} |X_u| - 1.$$

Die Baumweite eines Graphen *G (kurz* Baumweite(G)*) ist die geringste Weite aller möglichen Baumdekompositionen für G.* Baumweite

F. Gurski et al., *Exakte Algorithmen für schwere Graphenprobleme*, eXamen.press,
DOI 10.1007/978-3-642-04500-4_10,

Anmerkung 10.2. 1. Die einigen Lesern vielleicht etwas merkwürdig erscheinende „-1“ in der oben angegebenen Definition der Weite einer Baumdekomposition dient nur der Normierung, sodass Bäume die Baumweite 1 besitzen (siehe Satz 10.25).

2. In der Literatur findet man anstelle der dritten Bedingung in Definition 10.1 sehr häufig die folgende Bedingung:

$$\text{Für alle } u,v,w \in V_T \text{ gilt: Wenn } v \text{ auf dem Weg von } u \text{ nach } w \text{ in } T \text{ liegt, dann gilt } X_u \cap X_w \subseteq X_v. \tag{10.1}$$

Beide Bedingungen sind jedoch äquivalent, sodass wir hier die in der Definition angegebene Bedingung verwenden, da diese wesentlich leichter verifiziert werden kann.

Übung 10.3. Zeigen Sie, dass die in (10.1) angegebene Bedingung äquivalent zur dritten Bedingung in Definition 10.1 ist.

Beispiel 10.4 (Baumdekomposition). Wir betrachten den Graphen G in Abb. 10.1 und eine Baumdekomposition $(\mathscr{X},T)$ für G in Abb. 10.2. Man kann leicht überprüfen, dass die drei Bedingungen aus Definition 10.1 erfüllt sind. Da

$$\max_{u \in V_T} |X_u| - 1 = 3 - 1 = 2$$

gilt, hat die angegebene Baumdekomposition $(\mathscr{X},T)$ die Weite 2.

Da es keine Baumdekomposition der Weite 1 für G gibt, hat G die Baumweite 2.

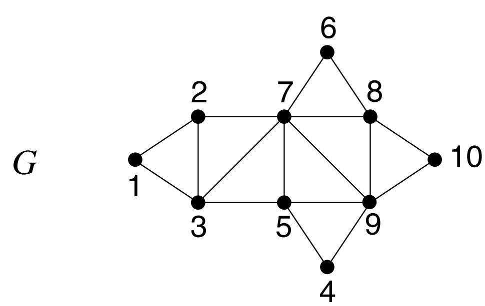

Abb. 10.1. Ein Graph für Beispiel 10.4

Durch eine Einschränkung der Baumstruktur einer Baumdekomposition zu einer Wegstruktur erhalten wir einen weiteren Graphparameter, der ebenfalls von Robertson und Seymour eingeführt wurde [RS83].

Wegdekomposition
Wegweite

Definition 10.5 (Wegweite). *Falls der Baum T in Definition 10.1 ein Weg ist, so heißt $(\mathscr{X},T)$ eine* Wegdekomposition. *Die* Wegweite *eines Graphen G (kurz mit* Wegweite(G) *bezeichnet) ist die geringste Weite aller möglichen Wegdekompositionen für G.*

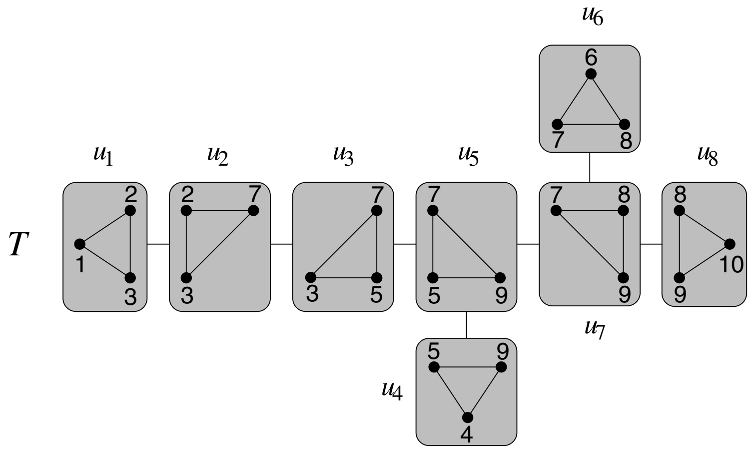

Abb. 10.2. Eine Baumdekomposition $(\mathscr{X}, T)$ der Weite 2 für den Graphen aus Abb. 10.1

Beispiel 10.6 (Wegdekomposition). Wir betrachten nochmals den Graphen G in Abb. 10.1 und eine Wegdekomposition $(\mathscr{X}, T)$ für G in Abb. 10.3. Man kann leicht überprüfen, dass die drei Bedingungen aus Definition 10.1 erfüllt sind und dass T ein Weg ist. Da

$$\max_{u \in V_T} |X_u| - 1 = 4 - 1 = 3$$

gilt, hat die angegebene Wegdekomposition $(\mathscr{X}, T)$ die Weite 3.

Da es keine Wegdekomposition der Weite 2 für G gibt, hat G die Wegweite 3.

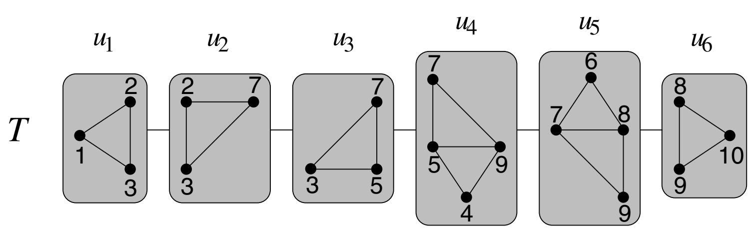

Abb. 10.3. Eine Wegdekomposition $(\mathscr{X}, T)$ der Weite 3 für den Graphen aus Abb. 10.1

Anmerkung 10.7. Jede Wegdekomposition ist offenbar auch eine Baumdekomposition. Weiterhin kann man für jeden Graphen $G = (V, E)$ durch

$$(\mathscr{X}, T) = (\{V\}, (\{u_1\}, \emptyset))$$

eine Baumdekomposition und Wegdekomposition der Weite $|V| - 1$ angeben. Damit gilt für jeden Graphen $G = (V, E)$ offensichtlich

$$\text{Baumweite}(G) \leq \text{Wegweite}(G) \leq |V| - 1.$$

Da die Baumweite bzw. die Wegweite in der Laufzeit von parametrisierten Algorithmen im Exponenten vorkommt, sind stets Baumdekompositionen und Wegdekompositionen mit möglichst geringer Weite von Interesse. Leider ist das Problem, eine solche optimale Baum- oder Wegdekompositionen zu finden, bzw. schon das Bestimmen der Weite einer solchen Dekomposition schwer. Als Entscheidungsproblem lassen sich diese Probleme folgendermaßen ausdrücken:

BAUMWEITE

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

WEGWEITE

Das analoge Problem für die Wegweite lautet wie folgt.

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

Satz 10.8 (Arnborg, Corneil und Proskurowski [ACP87]). BAUMWEITE *ist* NP-*vollständig und* WEGWEITE *ist* NP-*vollständig.* **ohne Beweis**

Die parametrisierten Problemvarianten von BAUMWEITE und WEGWEITE ergeben sich wie folgt:

p-BAUMWEITE

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

p-WEGWEITE

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

Satz 10.9 (Bodlaender [Bod96]). *Es gibt ein Polynom p und einen Algorithmus, der zu einem gegebenen Graphen $G = (V,E)$ eine Baumdekomposition der Weite $k = \mathrm{Baumweite}(G)$ in der Zeit $\mathcal{O}(|V| \cdot 2^{p(k)})$ bestimmt.* **ohne Beweis**

Eine analoge Aussage gilt auch für die Wegweite. Die Beweise dieser Aussagen sind sehr umfangreich und die auftretenden Polynome p haben einen hohen Grad.

Korollar 10.10. p-BAUMWEITE $\in$ FPT *und* p-WEGWEITE $\in$ FPT.

Folglich sind für jedes feste $k \in \mathbb{N}$ die Probleme, ob ein gegebener Graph eine Baumweite von höchstens k bzw. eine Wegweite von höchstens k hat, in linearer Zeit entscheidbar, und im positiven Fall kann auch eine Baumdekomposition bzw. Wegdekomposition der Weite k bestimmt werden.

Wir geben nun Abschätzungen der Baumweite und Wegweite von einigen Graphen an, die bei der Bestimmung der Baumweite und Wegweite weiterer Graphen hilfreich sind. Eine untere Schranke für die Baumweite eines Graphen liefern stets vollständige Teilgraphen.

Lemma 10.11 (Bodlaender und Möhring [BM93]). *Es seien G ein Graph und $(\mathscr{X}, T)$ mit $T = (V_T, E_T)$ eine Baumdekomposition für G. Für jeden vollständigen Teilgraphen $K = (V_K, E_K)$ von G gibt es einen Knoten $u \in V_T$, sodass $V_K \subseteq X_u$ gilt.*

Beweis. Es seien G ein Graph und $(\mathscr{X}, T)$ eine Baumdekomposition für G. Wir zeigen die Aussage induktiv über die Anzahl n der Knoten in $K = (V_K, E_K)$.

Ist $n = 1$, so gibt es einen Knoten u in T mit $V_K \subseteq X_u$ nach der Definition der Baumdekomposition.

Es seien nun $n > 1$ und $x \in V_K$. Nach Induktionsannahme gibt es einen Knoten u in T, sodass $V_K - \{x\} \subseteq X_u$. Wir suchen nun einen Knoten in T, der alle Elemente aus V_K enthält. Es sei $T' = (V', E')$ der Teilbaum von T, der durch die Knoten u' induziert wird, für die $x \in X_{u'}$ gilt.

Ist $u \in V'$, so gilt $V_K - \{x\} \subseteq X_u$ nach Induktionsannahme und $\{x\} \subseteq X_u$ nach Definition von V', also gilt $V_K \subseteq X_u$.

Wir nehmen nun an, dass $u \notin V'$ gilt, siehe Abb. 10.4. Es sei $v \in V'$ der Knoten aus T', der die kleinste Distanz zu u in T hat. Wir zeigen, dass $V_K \subseteq X_v$ gilt. Aufgrund der Definition von T' liegt x in X_v. Also müssen wir noch für jeden Knoten y aus $V_K - \{x\}$ zeigen, dass y in X_v ist. Offenbar enthält jeder Weg in T mit einem Startknoten aus T' und dem Zielknoten u stets den Knoten v. Da $x, y \in V_K$ und K eine Clique in G ist, enthält der Graph G eine Kante $\{x, y\}$, und somit gibt es einen Knoten $v' \in V'$, sodass $x, y \in X_{v'}$ gilt. Da der Knoten y auch in $V_K - \{x\} \subseteq X_u$ liegt und der Weg von v' nach u in T den Knoten v enthält, muss aufgrund der dritten Bedingung der Definition von Baumdekompositionen (siehe Definition 10.1) auch $y \in X_v$ gelten. ❑

Eine analoge Aussage gibt es auch für vollständig bipartite Graphen.

Lemma 10.12 (Bodlaender und Möhring [BM93]). *Es seien G ein Graph und $(\mathscr{X}, T)$ eine Baumdekomposition für G. Weiter gelte $V_1, V_2 \subseteq V$ und $\{\{v_1, v_2\} \mid v_1 \in V_1, v_2 \in V_2\} \subseteq E$. Es gibt einen Knoten $u \in V_T$, sodass $V_1 \subseteq X_u$ oder $V_2 \subseteq X_u$ gilt.* **ohne Beweis**

Das folgende Lemma wird im Beweis einiger der folgenden Aussagen benötigt (z. B. beim Beweis von Korollar 10.15).

Lemma 10.13. *Ein Graph mit einer Baumweite von höchstens k hat stets einen Knoten vom Grad höchstens k.*

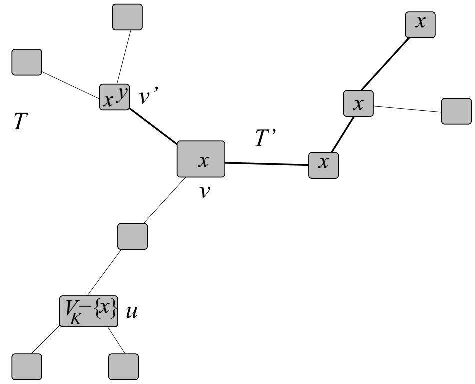

Abb. 10.4. Zum Beweis von Lemma 10.11

Beweis. Es sei G ein Graph mit Baumweite k. Betrachte eine Baumdekomposition $(\mathcal{X}, T)$, $T = (V_T, E_T)$, der Weite k für G, sodass $|V_T|$ minimal ist. Ein beliebiger Knoten $w \in V_T$ sei als Wurzel ausgezeichnet. Weiter sei a ein Blatt im Wurzelbaum T. Wegen der Minimalität von $|V_T|$ gibt es einen Knoten $u \in X_a$, sodass $u \notin X_b$ für alle $b \in V_T$ mit $b \neq a$ gilt. (Wenn jeder Knoten aus der Tasche X_a auch in der Tasche X_c des Vorgängers c von a in T vorkäme, so wäre a in T überflüssig, was im Widerspruch zur Minimalität von $|V_T|$ steht. Aufgrund der Zusammenhangseigenschaft von Baumdekompositionen – siehe die dritte Bedingung in Definition 10.1 – kommt der Knoten u auch in keiner anderen Tasche X_b mit $b \neq a$ vor.) Somit sind alle Nachbarn von u in G in der Tasche X_a enthalten, und da X_a höchstens $k+1$ Knoten enthält, hat u den Grad höchstens k in G. ❑

Aus dem obigen Beweis folgt unmittelbar auch die folgende interessante Aussage über Baumdekompositionen.

Korollar 10.14. *Ist G ein Graph mit n Knoten und Baumweite k, so gibt es eine Baumdekomposition $(\mathcal{X}, T)$ der Weite k für G, sodass T ein Wurzelbaum mit höchstens n Blättern ist.*

Eine weitere einfache Folgerung aus Lemma 10.13 ist die folgende Aussage.

Korollar 10.15. *Jeder k-fach zusammenhängende Graph hat eine Baumweite von mindestens k.*

Beweis. Ein Graph ist genau dann k-fach zusammenhängend, wenn es zwischen je zwei Knoten mindestens k Wege gibt. Folglich hat in einem k-fach zusammenhängenden Graphen jeder Knoten einen Grad von mindestens k. Mit Lemma 10.13 folgt nun die Behauptung. ❑

Weiterhin gilt, dass Graphen der Baumweite k stets eine in der Knotenanzahl lineare Kantenanzahl besitzen.

Lemma 10.16. *Ist $G = (V,E)$ ein Graph mit einer Baumweite von höchstens k, so gilt:*

$$|E| \leq k \cdot |V| - \frac{1}{2}k(k+1).$$

Beweis. Es sei $G = (V,E)$ ein Graph mit einer Baumweite von höchstens k. Wir zeigen die Aussage induktiv über die Knotenanzahl n von G.

Falls G weniger als $k+1$ Knoten hat, so ist Baumweite$(G) < k$.

Falls G genau $k+1$ Knoten hat, so ist G ein Teilgraph eines vollständigen Graphen mit $k+1$ Knoten und hat somit höchstens

$$\frac{1}{2}k(k+1) = k(k+1) - \frac{1}{2}k(k+1) = k|V| - \frac{1}{2}k(k+1)$$

Kanten.

Es seien nun G ein Graph mit mehr als $k+1$ Knoten und $(\mathscr{X}, T)$ eine Baumdekomposition der Weite k für G. Nach Lemma 10.13 gibt es in G einen Knoten u mit einem Grad von höchstens k. Der Graph

$$G - u = (V', E') = (V - \{u\}, E - \{\{u,z\} \in E \mid z \in V\})$$

hat eine Baumweite von höchstens k. Somit gilt:

$$\begin{aligned}
|E| &\leq |E'| + k \\
&\leq k \cdot (n-1) - \frac{1}{2}k(k+1) + k \\
&= k \cdot n - \frac{1}{2}k(k+1).
\end{aligned}$$

Damit ist das Lemma gezeigt. ❏

Für einige spezielle Graphen ist die Baumweite bekannt.

Lemma 10.17.
1. *Für jedes $n \geq 1$ hat der vollständige Graph K_n die Baumweite* $n-1$.
2. *Für alle $n,m \geq 0$ hat der vollständig bipartite Graph $K_{n,m}$ die Baumweite* $\min\{n,m\}$.
3. *Für alle $n,m \geq 0$ hat der Gittergraph $G_{n,m}$ die Baumweite* $\min\{n,m\}$.

Beweis.

1. Wir haben bereits in Anmerkung 10.7 festgehalten, dass jeder Graph mit n Knoten eine Baumweite von höchstens $n-1$ besitzt. Folglich gilt dies auch für den K_n.
   Angenommen, die Baumweite des K_n wäre echt kleiner als $n-1$. Dann hätte der K_n nach Lemma 10.13 einen Knoten vom Grad höchstens $n-2$, was jedoch offensichtlich ein Widerspruch ist.

2. Es sei $n \geq m$ und die Knoten in der Partition des $K_{n,m}$ seien mit $\{u_1, \ldots, u_n\}$ bzw. $\{w_1, \ldots, w_m\}$ bezeichnet. Dann liefert
$$(\{X_{q_1}, \ldots, X_{q_n}\}, (\{q_1, \ldots, q_n\}, \{\{q_1, q_2\}, \ldots, \{q_{n-1}, q_n\}\}))$$
mit $X_{q_1} = \{w_1, \ldots, w_m, u_1\}, \ldots, X_{q_n} = \{w_1, \ldots, w_m, u_n\}$ eine Baumdekomposition der Weite
$$\max_{q \in \{q_1, \ldots, q_n\}} |X_q| - 1 = m + 1 - 1 = m = \min\{n, m\}$$
für $K_{n,m}$, und somit ist die Baumweite des $K_{n,m}$ höchstens $\min\{n,m\}$.
Die untere Schranke folgt wieder aus Lemma 10.13. Hätte der $K_{n,m}$ eine Baumweite von $\min\{n,m\} - 1$ oder kleiner, so hätte der $K_{n,m}$ einen Knoten vom Grad höchstens $\min\{n,m\} - 1$, was jedoch offensichtlich ein Widerspruch ist.
3. Die obere Schranke ist einfach (siehe Übung 10.18) und die untere Schranke etwas aufwändiger zu zeigen. ❑

Übung 10.18. Zeigen Sie, dass Gittergraphen $G_{n,m}$ für $n, m \geq 0$ eine Baumweite von höchstens $\min\{n,m\}$ haben.

Der folgende Satz liefert einige Zusammenhänge zwischen der Baumweite eines Graphen und den Baumweiten seiner Teilgraphen.

Satz 10.19. *Es sei G ein Graph.*

1. *Die Baumweite eines jeden Teilgraphen von G ist durch die Baumweite von G nach oben beschränkt.*
2. *Die Baumweite von G ist gleich dem Maximum der Baumweiten der Zusammenhangskomponenten von G.*
3. *Die Baumweite von G ist gleich dem Maximum der Baumweiten der zweifachen Zusammenhangskomponenten von G.*

Beweis.

1. Es seien $G = (V,E)$ ein Graph, $(\mathcal{X}, T)$ eine Baumdekomposition für G der Weite k und $G' = (V', E)$ ein Teilgraph von G. Entfernt man alle Knoten aus $V - V'$ aus den Taschen X_u, $u \in V_T$, so erhält man eine Baumdekomposition für G' mit einer Weite von höchstens k.
2. Nach der ersten Aussage des Satzes ist die Baumweite jeder Zusammenhangskomponente von G höchstens so groß wie die Baumweite von G, und somit ist auch auch das Maximum der Baumweiten der Zusammenhangskomponenten von G höchstens so groß wie die Baumweite von G.
Es seien $(\mathcal{X}_1, T_1), \ldots, (\mathcal{X}_r, T_r)$ die Baumdekompositionen für die r Zusammenhangskomponenten von G. Eine Baumdekomposition $(\mathcal{X}, T)$ für G erhält man durch die disjunkte Vereinigung der r Bäume $T_1, \ldots, T_r$, in die $r - 1$ Kanten eingefügt werden, sodass ein Baum T einsteht. Die Menge $\mathcal{X}$ ist die Vereinigung der Taschen aus allen $\mathcal{X}_1, \ldots, \mathcal{X}_r$. Somit ist auch die Baumweite von G höchstens so groß wie das Maximum der Baumweiten der r Zusammenhangskomponenten von G.

3. Nach der ersten Aussage des Satzes ist die Baumweite jeder zweifachen Zusammenhangskomponente von G höchstens so groß wie die Baumweite von G, und somit ist auch auch das Maximum der Baumweiten der zweifachen Zusammenhangskomponenten von G höchstens so groß wie die Baumweite von G.
   Es sei $G = (V,E)$ ein Graph. Wir nehmen an, dass G zusammenhängend ist (sonst wenden wir die zweite Aussage des Satzes an). Es sei $A \subseteq V$ die Menge der Artikulationspunkte von G. Weiter seien $(\mathscr{X}_1,T_1),\ldots,(\mathscr{X}_r,T_r)$ die Baumdekompositionen für die r zweifachen Zusammenhangskomponenten $Z_1,\ldots,Z_r$ von G. Es seien T die disjunkte Vereinigung der r Bäume $T_1,\ldots,T_r$ und $\mathscr{X}$ die Vereinigung der Taschen aus allen $\mathscr{X}_1,\ldots,\mathscr{X}_r$. Für jeden Artikulationspunkt $x \in A$ fügen wir einen weiteren Knoten u_x zu T hinzu und definieren $X_{u_x} = \{x\}$. Für jede zweifache Zusammenhangskomponente Z_j, die den Knoten x enthält, fügen wir eine Kante zwischen dem Knoten u_x und einem Knoten aus dem Baum T_j mit $x \in X_v$ hinzu. So erhalten wir eine Baumdekomposition für den Graphen G. Somit ist die Baumweite von G höchstens so groß wie das Maximum der Baumweiten der r zweifachen Zusammenhangskomponenten von G. ❑

Die erste Aussage aus Satz 10.19 impliziert, dass für jede natürliche Zahl k die Menge der Graphen mit einer Baumweite von höchstens k abgeschlossen bezüglich Teilgraphenbildung (und somit natürlich auch bezüglich der Bildung induzierter Teilgraphen) ist.

Die ersten beiden Aussagen aus Satz 10.19 gelten auch für die Wegweite.

Satz 10.20. *Es sei G ein Graph.*

1. *Die Wegweite eines jeden Teilgraphen von G ist durch die Wegweite von G nach oben beschränkt.*
2. *Die Wegweite von G ist gleich dem Maximum der Wegweiten der Zusammenhangskomponenten von G.*

ohne Beweis

Wir übertragen nun die oben für Graphen definierten Begriffe Baumweite und Wegweite auf Graphklassen.

Definition 10.21 (Baumweite und Wegweite von Graphklassen). *Eine Graphklasse $\mathscr{G}$ hat eine* beschränkte Baumweite (bzw. Wegweite)*, falls es eine natürliche Zahl k gibt, sodass alle Graphen in $\mathscr{G}$ eine Baumweite (bzw. Wegweite) von höchstens k haben. Wenn k die kleinste Zahl mit dieser Eigenschaft ist, so heißt k die* Baumweite (bzw. Wegweite) *von $\mathscr{G}$.*

Als Beispiele für Graphklassen mit beschränkter Baumweite betrachten wir nun die Klasse der serienparallelen Graphen und die Klasse der Halingraphen. Serienparallele Graphen sind spezielle Multigraphen, d. h. Graphen, die zwischen zwei Knoten auch mehr als eine Kante haben können.

serienparalleler Graph

Definition 10.22 (serienparalleler Graph). *Ein* serienparalleler Graph *ist ein Tripel (G,s,t), wobei $G = (V,E)$ ein Graph mit $s,t \in V$ ist, der sich mit den folgenden drei Regeln aufbauen lässt:*

1. *Der Graph (G,s,t) mit $G = (\{s,t\},\{\{s,t\}\})$ ist ein serienparalleler Graph. Der Knoten s wird als* Startknoten von G *und der Knoten t als* Zielknoten von G *bezeichnet.*

Abb. 10.5. Ein serienparalleler Graph

2. *Es seien (G_1,s_1,t_1) und (G_2,s_2,t_2) zwei serienparallele Graphen mit Start- und Zielknoten s_1 und t_1 bzw. s_2 und t_2. Dann liefern die folgenden beiden Operationen wieder einen serienparallelen Graphen.*

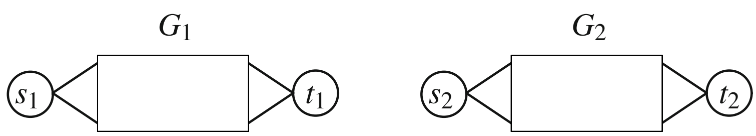

Abb. 10.6. Zwei serienparallele Graphen, G_1 und G_2

   a) *(serielle Kombination) Der Graph (G,s_1,t_2), der durch Identifikation der Knoten t_1 und s_2 entsteht, ist ein serienparalleler Graph. Der Startknoten von G ist s_1 und der Zielknoten von G ist t_2.*

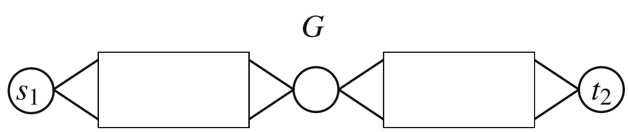

Abb. 10.7. Serielle Kombination serienparalleler Graphen

   b) *(parallele Kombination) Der Graph (G,s,t), der durch Identifikation der Knoten s_1 und s_2 zu s sowie der Knoten t_1 und t_2 zu t entsteht, ist ein serienparalleler Graph. Der Startknoten von G ist s und der Zielknoten von G ist t.*

Halingraph

Definition 10.23 (Halingraph). *Ein Halingraph $G = (V, E_T \cup E_C)$ besteht aus einem Baum (V,E_T), in dem alle Knoten in $V - V_B$ (wobei $V_B \subseteq V$ die Menge der Blätter von (V,E_T) ist) den Grad mindestens drei haben, und aus einem Kreis (V_B,E_C).*

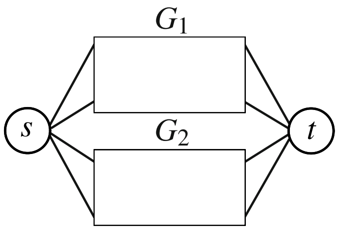

Abb. 10.8. Parallele Kombination serienparalleler Graphen

Mit anderen Worten, ein Halingraph ist ein Graph, der aus einer Einbettung eines Baums ohne Knoten vom Grad 2 in der Ebene entsteht, wobei die Knoten vom Grad 1 des Baums durch einen Kreis verbunden werden, der keine der Baumkanten schneidet.

Beispiel 10.24 (Halingraph). In Abb. 10.1 zeigen wir drei Beispiele für Halingraphen. Die Baumkanten sind schwarz und die Kreiskanten grau gezeichnet.

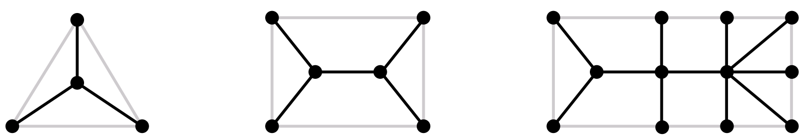

Abb. 10.9. Drei Halingraphen

Bekannte Beispiele für Graphklassen mit einer beschränkten Baumweite fassen wir im folgenden Satz zusammen. Die Beweise sind einfach (siehe Übung 10.26).

Satz 10.25. *1. Ein Graph G hat genau dann die Baumweite* 0*, wenn G keine Kanten enthält.*
2. Ein Graph G hat genau dann die Baumweite 1*, wenn G ein Wald ist, d. h. genau dann, wenn jede Zusammenhangskomponente von G ein Baum ist.*
3. Serienparallele Graphen haben eine Baumweite von höchstens 2*.*
4. Halingraphen haben eine Baumweite von höchstens 3*.*

ohne Beweis

Unbeschränkte Baumweite haben nach Lemma 10.17 hingegen

- die Klasse aller vollständigen Graphen, $\{K_n \mid n \in \mathbb{N}\}$,
- die Klasse aller vollständig bipartiten Graphen, $\{K_{n,m} \mid n,m \in \mathbb{N}\}$, und
- die Klasse aller Gittergraphen, $\{G_{n,m} \mid n,m \in \mathbb{N}\}$.

Übung 10.26. 1. Zeigen Sie, dass Wälder eine Baumweite von höchstens 1 haben.
2. Zeigen Sie, dass serienparallele Graphen eine Baumweite von höchstens 2 haben.
   Hinweis: Stellen Sie den Aufbau eines serienparallelen Graphen in einer binären Baumstruktur dar und transformieren Sie diese in eine Baumdekomposition. (Somit haben die Kreise C_n offenbar auch die Baumweite 2.)
3. Zeigen Sie, dass Halingraphen eine Baumweite von höchstens 3 haben.
   Hinweis: Nutzen Sie die zugrunde liegende Baumstruktur (also den Graphen ohne die Kanten zwischen den Blättern), um eine Baumdekomposition zu definieren. Lösen Sie die Aufgabe zunächst für eine binäre Baumstruktur und versuchen Sie anschließend, Ihre Lösung zu verallgemeinern.

Die Wegweite stellt im Vergleich zur Baumweite eine echte Einschränkung dar. Zum Beispiel hat ein vollständiger binärer Baum der Höhe $h \geq 2$ (und mit $2^{h+1}-1$ Knoten) die Baumweite 1 und die Wegweite $h-1$.

Übung 10.27. Zeigen Sie, dass ein vollständiger binärer Baum der Höhe $h \geq 2$ (mit $2^{h+1}-1$ Knoten) eine Wegweite von höchstens $h-1$ hat.

Zwei mit der Baumweite sehr eng verwandte Parameter betrachten wir in den folgenden beiden Definitionen.

k-Baum

Definition 10.28 (*k*-Baum und partieller *k*-Baum). *Die Klasse der* k-Bäume *enthält genau die Graphen, die wie folgt konstruiert werden können:*

1. *Für jedes $k \geq 0$ ist der vollständige Graph K_k ein k-Baum.*
2. *Sind $G = (V,E)$ ein k-Baum, $v \notin V$ ein Knoten und $G' = (V',E')$ ein vollständiger Teilgraph mit k Knoten von G, so ist auch der Graph $(V \cup \{v\}, E \cup \{\{v,v'\} \mid v' \in V'\})$ ein k-Baum.*

partieller *k*-Baum

Ein Graph $G = (V,E)$ heißt partieller k-Baum, *falls es einen k-Baum $\hat{G} = (V,\hat{E})$ mit $E \subseteq \hat{E}$ gibt.*

Satz 10.29. *Die Klasse der partiellen k-Bäume ist gleich der Klasse der Graphen mit einer Baumweite von höchstens k.* **ohne Beweis**

Übung 10.30. Zeigen Sie Satz 10.29.

Nach der Baumweite und der Wegweite führen wir nun noch einen dritten Parameter ein, die Verzweigungsweite.

Verzweigungs-dekomposition
Ordnung einer Kante

Definition 10.31 (Verzweigungsweite). *Eine* Verzweigungsdekomposition eines Graphen *G besteht aus einem Paar (T, f), sodass T ein binärer Baum ist und f eine Bijektion zwischen den Kanten von G und den Blättern von T. Die* Ordnung einer Kante *e* im Baum *T ist die Anzahl der Knoten in G, die mit zwei Kanten aus G, etwa e_1 und e_2, inzident sind, sodass der einfache Weg zwischen den Blättern $f(e_1)$ und $f(e_2)$ in T die Kante e enthält. Die* Weite einer Verzweigungsdekomposition *ist die maximale Ordnung aller Kanten in T.*

Weite einer Verzweigungsdekomposition
Verzweigungsweite

Die Verzweigungsweite eines Graphen *G ist die minimale Weite aller Verzweigungsdekompositionen für G.*

Satz 10.32. *Für jeden Graphen G gilt der folgende Zusammenhang zwischen seiner Baumweite (kurz* $\mathrm{BW}(G)$*) und seiner Verzweigungsweite (kurz* $\mathrm{VW}(G)$*):*

$$\max\{\mathrm{VW}(G), 2\} \leq \mathrm{BW}(G) + 1 \leq \max\left\{\left\lfloor \frac{3}{2} \cdot \mathrm{VW}(G) \right\rfloor, 2\right\}.$$

ohne Beweis

Übung 10.33. Zeigen Sie Satz 10.32.

Um Baumdekompositionen algorithmisch nutzen zu können, ist es sinnvoll, Baumdekompositionen $(\mathscr{X}, T)$ zu betrachten, in denen T ein spezieller binärer Wurzelbaum ist. Solche Baumdekompositionen nennen wir schön und definieren sie wie folgt.

Definition 10.34 (schöne Baumdekomposition). *Eine Baumdekomposition* $(\mathscr{X}, T)$ *mit binärem Wurzelbaum* $T = (V, E, r)$ *und* $\mathscr{X} = \{X_u \mid u \in V\}$ *der Weite k heißt* schöne Baumdekomposition*, falls die folgenden Bedingungen erfüllt sind:*

schöne Baumdekomposition

1. *Hat ein Knoten u in T zwei Kinder v und w, so gilt* $X_u = X_v = X_w$*. Hier wird u ein* Join-Knoten *genannt.*
2. *Hat ein Knoten u in T genau ein Kind v, so gilt eine der beiden folgenden Bedingungen:*
   a) $|X_u| = |X_v| + 1$ *und* $X_v \subset X_u$*. Hier wird u ein* Introduce-Knoten *genannt.*
   b) $|X_u| = |X_v| - 1$ *und* $X_u \subset X_v$*. Hier wird u ein* Forget-Knoten *genannt.*
3. *Ist ein Knoten u ein Blatt von T, so gilt* $|X_u| = 1$*.*

Join-Knoten

Introduce-Knoten

Forget-Knoten

Satz 10.35 (Kloks [Klo94]). *Es sei G ein Graph mit n Knoten und Baumweite k. Dann gibt es eine schöne Baumdekomposition* $(\mathscr{X}, T)$ *der Weite k für G, sodass die Knotenanzahl von T in* $\mathscr{O}(k \cdot n)$ *liegt.*

Beweis. Es seien $G = (V, E)$ ein Graph mit n Knoten und Baumweite k und $(\mathscr{X}, T)$ eine Baumdekomposition der Weite k für G. Wir wählen einen beliebigen Knoten w aus $T = (V_T, E_T)$ aus, sodass $T = (V_T, E_T, w)$ ein Wurzelbaum ist. Wir zeigen nun, wie man $(\mathscr{X}, T)$ in eine schöne Baumdekomposition $(\mathscr{X}', T')$ der Weite k für G transformieren kann. Dazu ersetzen wir die Knoten aus T wie folgt.

Falls ein Knoten u aus T mehr als ein Kind hat, etwa die Kinder $u_1, \ldots, u_d$, $d > 1$, dann ersetzen wir u durch $d-1$ Knoten $u'_1, \ldots, u'_{d-1}$ und d Knoten $v'_1, \ldots, v'_d$. Der Vorgänger von u'_1 ist der Vorgänger von u in T. Die Knoten u'_i, $i \in \{1, \ldots, d-1\}$, haben als ein Kind jeweils den Knoten v'_i. Die Knoten u'_i für $i \in \{1, \ldots, d-2\}$ haben als ein weiteres Kind jeweils den Knoten u'_{i+1}. Der Knoten u'_{d-1} hat als zweites Kind (neben v'_{d-1}) den Knoten v'_d. Die Knoten v'_i, $i \in \{1, \ldots, d\}$, haben als einziges Kind jeweils den Knoten u_i, d. h. die ursprünglichen Kinder von u. Weiterhin sind die Taschen definiert durch $X_{u'_i} = X_u$, $i \in \{1, \ldots, d-1\}$, und $X_{v'_i} = X_u$, $i \in \{1, \ldots, d\}$, siehe Abb. 10.10.

Falls u genau ein Kind v hat und sich die Taschen X_u und X_v nicht nur um genau einen Knoten unterscheiden, d. h., falls u kein Introduce- oder Forget-Knoten ist, so ersetzen wir die Kante $\{v, u\}$ in T durch einen Weg

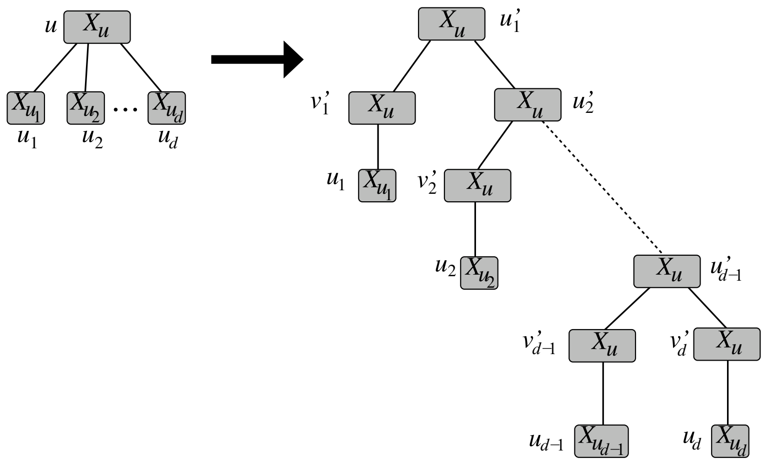

Abb. 10.10. Eine schöne Baumdekomposition für Knoten mit mehr als einem Kind

$$(v, \underbrace{v_1, \ldots, v_{|X_v|-|X_u \cap X_v|}}_{\text{Forget-Knoten}}, \underbrace{u_1, \ldots, u_{|X_u|-|X_u \cap X_v|-1}}_{\text{Introduce-Knoten}}, u),$$

auf dem Forget-Knoten gefolgt von Introduce-Knoten liegen. Genauer gesagt werden zuerst durch $|X_v| - |X_u \cap X_v|$ Forget-Knoten alle Knoten aus der Tasche X_v entfernt, die nicht in der Tasche X_u liegen, bis dem letzten Forget-Knoten $v_{|X_v|-|X_u \cap X_v|}$ die Tasche $X_{v_{|X_v|-|X_u \cap X_v|}} = X_u \cap X_v$ zugeordnet ist. Dann werden mittels einzelner Introduce-Knoten ausgehend von $X_u \cap X_v$ alle Knoten aus $X_u - X_v$ aufgenommen, bis schließlich die Tasche X_u wieder aufgebaut ist, siehe Abb. 10.11.

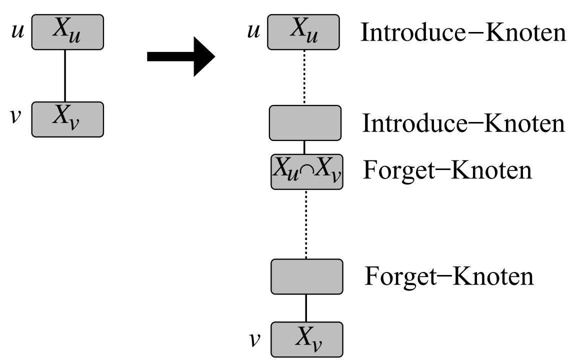

Abb. 10.11. Eine schöne Baumdekomposition für Knoten mit genau einem Kind

Falls u ein Blatt in dem so erhaltenen Baum T ist und die zugehörige Tasche X_u mehr als einen Knoten enthält (d. h., falls $|X_u| = d > 1$ gilt), so ersetzen wir das Blatt

u durch einen Weg mit d Introduce-Knoten, damit das neue Blatt u' einer Tasche $X_{u'}$ entspricht, die $|X_{u'}| = 1$ erfüllt.

Man macht sich leicht klar, dass die angegebene Konstruktion eine schöne Baumdekomposition $(\mathscr{X}', T')$ der Weite k für G liefert. Die Knotenanzahl des so erhaltenen Baums T' kann wie folgt abgeschätzt werden. Offenbar kann man jedem Forget-Knoten von $(\mathscr{X}', T')$ einen Knoten des Graphen zuordnen. Somit kann es aufgrund der dritten Bedingung in der Definition von Baumdekompositionen (siehe Definition 10.1) nicht mehr als n Forget-Knoten geben. Da der ursprüngliche Baum T nach Korollar 10.14 höchstens n Blätter besitzt, hat auch der Baum T' höchstens n Blätter. Da T' ein binärer Baum ist, gibt es bei n Blättern höchstens $n-1$ Join-Knoten. Die Introduce-Knoten wurden in unserer Transformation zur Ersetzung der Blätter und der Knoten mit genau einem Kind benötigt. Da es von beiden Arten von Knoten höchstens n in T gibt und da jede Menge X_u mit $u \in V_T$ höchstens $k+1$ Knoten enthält, liegt die Knotenanzahl von T' in $\mathscr{O}(k \cdot n)$. ❑

Anmerkung 10.36. Falls die ursprüngliche Baumdekomposition $(\mathscr{X}, T)$ im Beweis von Satz 10.35 eine Wegdekomposition ist, ist die in diesem Beweis konstruierte schöne Baumdekomposition $(\mathscr{X}', T')$ ebenfalls eine Wegdekomposition.

Übung 10.37. Transformieren Sie die Baumdekomposition aus Abb. 10.2 bezüglich der Wurzel u_1 in eine schöne Baumdekomposition.

Übung 10.38. Transformieren Sie die Wegdekomposition aus Abb. 10.3 bezüglich der Wurzel u_1 in eine schöne Baumdekomposition. (Wie in Anmerkung 10.36 erwähnt wurde, stellt die so erhaltene schöne Baumdekomposition natürlich wieder eine Wegdekomposition dar.)

Anmerkung 10.39. Wir parametrisieren im Folgenden Entscheidungsprobleme in der Baumweite des Eingabegraphen. Streng genommen ist dies nach unserer Definition kein zulässiger Parameter, da sich der Parameter „Baumweite des Eingabegraphen" nach Satz 10.8 nicht in polynomieller Zeit aus dem Eingabegraphen bestimmen lässt (es sei denn, es würde $\mathrm{P} = \mathrm{NP}$ gelten). Da der Parameter jedoch durch einen FPT-Algorithmus bestimmt werden kann, verwenden wir für diese allgemeinere und ebenfalls gebräuchliche Definition der Parametrisierung 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 Baumweite von höchstens k einschränkt und k als Parameter wählt.

10.2 Unabhängige Menge und Knotenüberdeckung

Unser erstes Problem besteht darin, die Größe einer möglichst großen unabhängigen Menge in einem gegebenen Graphen zu bestimmen. Als Parametrisierung wählen wir hier die Baumweite des Eingabegraphen (siehe Anmerkung 10.39).

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

p^*-tw-UNABHÄNGIGE MENGE

Die folgende Lösung beruht auf einer dynamischen Programmierung entlang einer schönen Baumdekomposition für den Eingabegraphen. Dazu benötigen wir einige Notationen. Es seien $G=(V_G,E_G)$ ein Graph mit Baumweite k und $(\mathscr{X},T)$ mit Wurzelbaum $T=(V_T,E_T,r)$ und $\mathscr{X}=\{X_u \mid u \in V_T\}$ eine Baumdekomposition der Weite k für G. Für einen Knoten $u \in V_T$ definieren wir $T_u=(V_u,E_u)$ als den Teilbaum von T mit Wurzel u und $\mathscr{X}_u$ als die Menge aller Taschen X_v, $v \in V_u$. Weiterhin sei G_u der Teilgraph von G, der durch die Baumdekomposition $(\mathscr{X}_u,T_u)$ definiert wird.

Die Idee beim Algorithmenentwurf entlang von Baumdekompositionen beruht häufig auf der Eigenschaft, dass für jeden Knoten u in T nur die $k+1$ Knoten der Tasche X_u aus dem durch G_u definierten Teilgraphen noch eine Verbindung zu einem Knoten im Restgraphen bekommen können. Hingegen können Knoten aus G_u, die nicht in X_u liegen, keine weiteren inzidenten Kanten in der Tasche X_u oder in einer Tasche X_w mit $w \notin V_u$ erhalten. Somit kann man sich bei Lösungsansätzen mittels dynamischer Programmierung entlang einer Baumdekomposition bei der Bestimmung von Teillösungen häufig auf die höchstens $k+1$ Knoten der aktuellen Tasche beschränken.

Zur Lösung von p^*-tw-UNABHÄNGIGE MENGE verwenden wir für jeden Knoten u von T eine Datenstruktur $F(u)$. Für jeden Knoten $u \in V_T$ ist $F(u)$ ein 2^{k+1}-Tupel, das für jede Teilmenge $X \subseteq X_u$ eine natürliche Zahl a_X enthält, d. h.,

$$F(u) = (a_X \mid X \subseteq X_u).$$

Der Wert von a_X ist die Größe einer größten unabhängigen Menge $U \subseteq V_{G_u}$ im Graphen G_u, sodass $U \cap X_u = X$. Das heißt, a_X ist die Größe einer größten unabhängigen Menge U im Graphen G_u, sodass alle Knoten aus X zu U gehören und alle Knoten aus $X_u - X$ nicht zu U gehören. Falls keine solche unabhängige Menge existiert (insbesondere also, falls es in X zwei adjazente Knoten gibt), setzen wir $a_X = -\infty$.

Beispiel 10.40 (Datenstruktur $F(u)$). Es seien $k=2$ und G_u der in Abb. 10.12 dargestellte Graph mit $X_u=\{x,y,z\}$.

Dann ist $F(u)=(a_\emptyset, a_{\{x\}}, a_{\{y\}}, a_{\{z\}}, a_{\{x,y\}}, a_{\{x,z\}}, a_{\{y,z\}}, a_{\{x,y,z\}})$, wobei die Werte a_X, $X \subseteq X_u = \{x,y,z\}$, wie folgt lauten:

X	$\emptyset$	$\{x\}$	$\{y\}$	$\{z\}$	$\{x,y\}$	$\{x,z\}$	$\{y,z\}$	$\{x,y,z\}$
a_X	1	2	2	2	$-\infty$	$-\infty$	3	$-\infty$

Somit lässt sich unser Algorithmus für p^*-tw-UNABHÄNGIGE MENGE folgendermaßen formulieren:

1. Bestimme eine Baumdekomposition für den Eingabegraphen $G=(V_G,E_G)$ (siehe Satz 10.9).

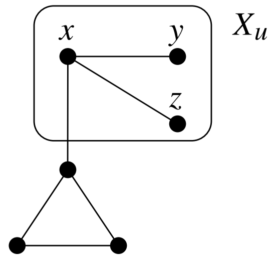

Abb. 10.12. Ein Beispiel zur Datenstruktur für p^*-tw-UNABHÄNGIGE MENGE

2. Transformiere diese Baumdekomposition in eine schöne Baumdekomposition $(\mathscr{X}, T)$ für G (siehe Satz 10.35).
3. Mittels dynamischer Programmierung entlang des Baums T mit der Wurzel r kann $F(r)$ durch einen Durchlauf in Bottom-up-Reihenfolge der Knoten mit der folgenden Aktion auf den Knoten berechnet werden:
   a) Falls u ein Blatt in T ist, sodass $X_u = \{v_1\}$ für ein $v_1 \in V_G$ ist, definieren wir
   $$F(u) = (a_\emptyset, a_{\{v_1\}}) = (0, 1).$$
   b) Falls u ein Join-Knoten mit den Kindern v und w in T ist, sind $F(v) = (a_X \mid X \subseteq X_v)$ und $F(w) = (b_X \mid X \subseteq X_w)$ bereits bekannt. Weiterhin bestimmen wir den Wert i_X als die Größe einer kardinalitätsmaximalen unabhängigen Menge in $X \subseteq X_u$ und definieren
   $$F(u) = (c_X \mid X \subseteq X_u) \text{ mit } c_X = a_X + b_X - i_X$$
   für alle $X \subseteq X_u$.
   c) Falls u ein Introduce-Knoten mit einem Kind v in T ist, sind $X_u - X_v = \{v'\}$ für ein $v' \in V_G$ und $F(v) = (a_X \mid X \subseteq X_v)$ bereits bekannt. Dann definieren wir
   $$F(u) = (b_X \mid X \subseteq X_u)$$
   mit $b_X = a_X$ und
   $$b_{X \cup \{v'\}} = \begin{cases} a_X + 1, & \text{falls } v' \text{ zu keinem Knoten aus } X \text{ adjazent ist,} \\ -\infty & \text{sonst} \end{cases}$$
   für alle $X \subseteq X_v$.
   d) Falls u ein Forget-Knoten mit einem Kind v in T ist, sind $X_v - X_u = \{v'\}$ für ein $v' \in V_G$ und $F(v) = (a_X \mid X \subseteq X_v)$ bereits bekannt. Dann definieren wir
   $$F(u) = (b_X \mid X \subseteq X_u) \text{ mit } b_X = \max\{a_X, a_{X \cup \{v'\}}\}$$
   für alle $X \subseteq X_u$. Da $v' \in X_v - X_u$ ist und da die Graphen G_u und G_v identisch sind, wird dabei über die Größen der unabhängigen Mengen, die v' nicht enthalten, und denen, die v' enthalten, maximiert.

4. Mit dem Tupel $F(r)$ für die Wurzel r von T können wir die Größe einer kardinalitätsmaximalen unabhängigen Menge im durch $(\mathscr{X},T)$ definierten Graphen G bestimmen durch
$$\alpha(G) = \max_{a \in F(r)} a.$$

Satz 10.41. *Die Unabhängigkeitszahl eines Graphen $G=(V,E)$ mit einer Baumweite von höchstens k kann bei gegebener Baumdekomposition in der Zeit*

$$\mathscr{O}(|V| \cdot k \cdot 2^{k+1})$$

berechnet werden.

Beweis. In einer schönen Baumdekomposition $(\mathscr{X},T)$ hat $T=(V_T,E_T)$ höchstens $\mathscr{O}(k \cdot |V|)$ Knoten und für jeden Knoten $u \in V_T$ kann $F(u)$ in der Zeit $\mathscr{O}(2^{k+1})$ initialisiert bzw. aus den Tupeln des Kindes bzw. der Kinder von u bestimmt werden. ❑

Korollar 10.42. *Das Problem p^*-tw-*UNABHÄNGIGE MENGE *ist fest-Parameter-berechenbar.*

Korollar 10.43. *Das Problem* UNABHÄNGIGE MENGE *kann für jeden Graphen $G=(V,E)$ einer Graphklasse mit beschränkter Baumweite (z. B. der Klasse der Bäume, der serienparallelen Graphen, der Halingraphen usw.) in der Zeit $\mathscr{O}(|V|)$ gelöst werden.*

Es gilt die folgende interessante Beziehung zwischen der Baumweite und der Unabhängigkeitszahl von Graphen.

Satz 10.44 (Chlebikova [Chl02]). *Für jeden Graphen $G=(V,E)$ gilt*

$$\alpha(G) \leq |V| - \text{Baumweite}(G).$$

Beweis. Es sei $G=(V,E)$ ein Graph. Gilt $\alpha(G)=|V|$, so enthält G keine Kanten und die Aussage ist klar. Andernfalls teilen wir die Knotenmenge V in $r \geq 2$ Teilmengen $V_1,\ldots,V_r$ auf. Dabei ist die Menge V_1 eine unabhängige Menge maximaler Größe, d. h., $|V_1|=\alpha(G)$, und die restlichen $|V|-\alpha(G)$ Mengen V_i, $2 \leq i \leq r$, enthalten genau einen Knoten, d. h., $|V_i|=1$. Somit ist G ein Teilgraph des vollständigen $(|V|-\alpha(G)+1)$-partiten Graphen $K_{\alpha(G),1,\ldots,1}$. Nach Satz 10.19 folgt

$$\text{Baumweite}(G) \leq \text{Baumweite}(K_{\alpha(G),1,\ldots,1}) = |V| - \alpha(G).$$

Die Abschätzung $\text{Baumweite}(K_{\alpha(G),1,\ldots,1}) \leq |V|-\alpha(G)$ kann man durch Angabe einer Baumdekomposition der Weite $|V|-\alpha(G)$ leicht nachweisen und die Abschätzung $\text{Baumweite}(K_{\alpha(G),1,\ldots,1}) \geq |V|-\alpha(G)$ folgt, da der Graph $K_{\alpha(G),1,\ldots,1}$ einen vollständigen Teilgraphen der Größe $|V|-\alpha(G)+1$ enthält. ❑

Das Problem KNOTENÜBERDECKUNG lässt sich mittels der Gleichung (3.4) von Gallai sehr leicht auf das Problem UNABHÄNGIGE MENGE zurückführen. Dies nutzen wir aus, um die folgende parametrisierte Variante von KNOTENÜBERDECKUNG auf baumweitebeschränkten Graphklassen durch einen effizienten Fest-Parameter-Algorithmus zu lösen.

p^*-tw-KNOTENÜBERDECKUNG

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

Korollar 10.45. *Die Größe einer kardinalitätsminimalen Knotenüberdeckung eines Graphen $G = (V,E)$ mit einer Baumweite von höchstens k kann bei gegebener Baumdekomposition in der Zeit $\mathcal{O}(|V| \cdot k \cdot 2^{k+1})$ berechnet werden.*

Korollar 10.46. *Das Problem p^*-tw-*KNOTENÜBERDECKUNG *ist fest-Parameter-berechenbar.*

Es gilt die folgende Beziehung zwischen der Baumweite und der Knotenüberdeckungszahl von Graphen.

Satz 10.47. *Für jeden Graphen G gilt:*

$$\text{Baumweite}(G) \leq \tau(G).$$

Beweis. Die Abschätzung folgt leicht aus Satz 10.44 und Gleichung (3.4). ❑

Übung 10.48. Geben Sie jeweils eine unendliche Graphklasse $\mathcal{G}$ an, für die die folgende Bedingung erfüllt ist:

1. Für jeden Graphen $G \in \mathcal{G}$ gilt: $\alpha(G) < \text{Baumweite}(G)$.
2. Für jeden Graphen $G \in \mathcal{G}$ gilt: $\omega(G) < \text{Baumweite}(G)$.

Übung 10.48 zeigt somit, dass der Parameter τ in Satz 10.47 offenbar nicht durch α oder ω ersetzt werden kann.

Somit können wir nun einen alternativen Beweis für die schon in Satz 5.48 erwähnte Aussage angeben.

Korollar 10.49. *Das Problem p-*KNOTENÜBERDECKUNG *liegt in* FPT.

Beweis. Nach Korollar 10.45 kann die Größe einer kardinalitätsminimalen Knotenüberdeckung eines Graphen $G = (V,E)$ mit einer Baumweite von höchstens k in der Zeit $\mathcal{O}(|V| \cdot k \cdot 2^{k+1})$ berechnet werden. Nach Satz 10.47 liegt diese Laufzeit in $\mathcal{O}(|V| \cdot s \cdot 2^{s+1})$. ❑

Anmerkung 10.50. Es stellt sich nun die Frage, ob auch das Problem p-UNABHÄNGIGE MENGE fest-Parameter-berechenbar bezüglich des Parameters s ist.

Da die Baumweite eines Graphen die Unabhängigkeitszahl des Graphen deutlich überschreiten kann (siehe Übung 10.48), lässt sich eine Beziehung wie in Satz 10.47 für α statt τ nicht zeigen. Dies ist auch nicht verwunderlich, da ein solches Ergebnis implizieren würde, dass das Problem p-UNABHÄNGIGE MENGE fest-Parameter-berechenbar bezüglich des Parameters s wäre. Dies wäre unter der Annahme, dass $\text{FPT} \neq \text{W}[1]$ gilt, ein Widerspruch zu der Tatsache, dass p-UNABHÄNGIGE MENGE W[1]-vollständig ist.

p^*-tw-GEWICHTETE UNABHÄNGIGE MENGE

Die angegebene Lösung für p^*-tw-UNABHÄNGIGE MENGE kann leicht zu einer Lösung für die folgende gewichtete Problemvariante erweitert werden.

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

Übung 10.51. Geben Sie analog zur Lösung für p^*-tw-UNABHÄNGIGE MENGE eine Lösung für p^*-tw-GEWICHTETE UNABHÄNGIGE MENGE an.

10.3 Clique

p^*-tw-CLIQUE

Unser nächstes Problem besteht darin, die Größe einer möglichst großen Clique in einem gegebenen Graphen zu bestimmen. Als Parametrisierung wählen wir wieder die Baumweite des Eingabegraphen (siehe Anmerkung 10.39).

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

Es seien $G = (V_G, E_G)$ ein Graph der Baumweite k und $(\mathscr{X}, T)$ eine Baumdekomposition der Weite k, mit dem Wurzelbaum $T = (V_T, E_T, r)$ und der Menge $\mathscr{X} = \{X_u \mid u \in V_T\}$ der Taschen. Um die Größe einer kardinalitätsmaximalen Clique in G zu bestimmen, kann man offensichtlich eine analoge Lösung wie für das Problem p^*-tw-UNABHÄNGIGE MENGE angeben.

Übung 10.52. Bestimmen Sie die Größe einer kardinalitätsmaximalen Clique in einem gegebenen Graphen G analog zur Lösung für das Problem p^*-tw-UNABHÄNGIGE MENGE.

Weiterhin liefert Satz 10.11 über Cliquen und Baumdekompositionen eine einfache Möglichkeit, das Problem p^*-tw-CLIQUE auf baumweitebeschränkten Graphklassen zu lösen. Für einen Graphen G der Baumweite k kann der Wert $\omega(G)$ berechnet werden durch

$$\omega(G) = \max_{u \in V_T} \max_{\substack{C \subseteq X_u \\ G[C] \text{ ist eine Clique}}} |C|.$$

Satz 10.53. *Die Cliquenzahl eines Graphen $G = (V,E)$ mit einer Baumweite von höchstens k kann bei gegebener Baumdekomposition in der Zeit*

$$\mathcal{O}(|V| \cdot (k+1)^2 \cdot 2^{k+1})$$

bestimmt werden.

Beweis. In einer Baumdekomposition $(\mathcal{X}, T)$ für einen Graphen $G = (V,E)$ hat T nach Korollar 10.14 höchstens $\mathcal{O}(|V|)$ Knoten. Für jeden Knoten u in T enthält die Tasche X_u höchstens $k+1$ Elemente, womit sich höchstens 2^{k+1} Teilmengen von X_u ergeben. Das Testen einer Teilmenge benötigt höchstens $(k+1)^2$ Schritte. ❑

Korollar 10.54. *Das Problem p^*-tw-*CLIQUE *ist fest-Parameter-berechenbar.*

Korollar 10.55. *Das Problem* CLIQUE *kann für jeden Graphen $G = (V,E)$ einer Graphklasse mit beschränkter Baumweite (z. B. der Klasse der Bäume, der serien-parallelen Graphen, der Halingraphen usw.) in der Zeit $\mathcal{O}(|V|)$ gelöst werden.*

Aus Lemma 10.11 und Satz 10.19 folgt offenbar die folgende Beziehung zwischen der Baumweite und der Cliquenzahl von Graphen.

Korollar 10.56. *Für jeden Graphen $G = (V,E)$ gilt:*

$$\text{Baumweite}(G) \geq \omega(G) - 1.$$

10.4 Partition in unabhängige Mengen

Nun betrachen 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 Baumweite des Eingabegraphen (siehe Anmerkung 10.39).

p^*-tw-PARTITION IN UNABHÄNGIGE MENGEN

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

Es seien $G = (V_G, E_G)$ ein Graph der Baumweite k und $(\mathscr{X}, T)$ eine Baumdekomposition der Weite k für G, mit dem Wurzelbaum $T = (V_T, E_T, r)$ und der Menge $\mathscr{X} = \{X_u \mid u \in V_T\}$ der Taschen.

Zur Lösung von p^*-tw-PARTITION IN UNABHÄNGIGE MENGEN verwenden wir für jeden Knoten u von T eine Datenstruktur $F(u)$. Für jeden Knoten $u \in V_T$ enthält $F(u)$ für jede Partition von V_{G_u} in unabhängige Mengen $V_1, \ldots, V_s$ ein 2^{k+1}-Tupel $t = (\ldots, a_X, \ldots, a)$, das für jede nicht leere Teilmenge $X \subseteq X_u$ einen booleschen Wert $a_X \in \{0,1\}$ und eine natürliche Zahl a enthält. Für die Partition $V_1, \ldots, V_s$ von V_{G_u} in unabhängige Mengen sei genau dann $a_X = 1$, wenn $V_i \cap X_u = X$ für ein $1 \leq i \leq s$ gilt. Der Wert a im Tupel $t = (\ldots, a_X, \ldots, a)$ gibt die Anzahl der für den Teilgraphen G_u benötigten unabhängigen Mengen an.

Die Größe von $F(u)$, $u \in V_T$, ist polynomiell in der Größe von G beschränkt, da jedes Element von $F(u)$ genau 2^{k+1} Einträge hat, wobei $2^{k+1} - 1$ der Einträge einen Wert aus $\{0,1\}$ haben und einer einen Wert aus $\{1, \ldots, |V_G|\}$. Es gilt:

$$|F(u)| \leq 2^{2^{k+1}-1} \cdot |V_G|.$$

Somit lässt sich unsere Lösung für p^*-tw-PARTITION IN UNABHÄNGIGE MENGEN wie folgt formulieren:

1. Bestimme eine Baumdekomposition für den Eingabegraphen $G = (V_G, E_G)$ (siehe Satz 10.9).
2. Transformiere diese Baumdekomposition in eine schöne Baumdekomposition $(\mathscr{X}, T)$ für G (siehe Satz 10.35).
3. Mittels dynamischer Programmierung entlang des Baums T mit der Wurzel r kann $F(r)$ durch einen Durchlauf in Bottom-up-Reihenfolge der Knoten mit der folgenden Aktion auf den Knoten berechnet werden:
   a) Falls u ein Blatt in T ist, sodass $X_u = \{v_1\}$ für ein $v_1 \in V_G$ ist, definieren wir
   $$F(u) = \{(a_{\{v_1\}}, a)\} = \{(1,1)\}.$$
   b) Falls u ein Join-Knoten mit den Kindern v und w in T ist, sind $F(v)$ und $F(w)$ bereits bekannt. Wir nehmen in $F(u)$ also genau die Tupel $(\ldots, a_X, \ldots, \max\{a,b\})$ auf, für die $(\ldots, a_X, \ldots, a) \in F(v)$, $(\ldots, b_X, \ldots, b) \in F(w)$ und $a_X = b_X$ für alle $X \subseteq X_u$ gilt.
   c) Falls u ein Introduce-Knoten mit einem Kind v in T ist, sind $F(v)$ und $X_u - X_v = \{v'\}$ für ein $v' \in V_G$ bereits bekannt.
   Wir betrachten nun jede Partition von X_v in unabhängige Mengen, um eine neue unabhängige Menge aufzunehmen, die nur aus dem Knoten v' besteht, und um eine bereits existierende unabhängige Menge einer Partition von X_v gegebenenfalls um den Knoten v' zu erweitern.
   Formal definieren wir $F(u)$ wie folgt:
      i. Für jedes Tupel $t = (\ldots, a_X, \ldots, a) \in F(v)$ fügen wir ein Tupel der Form $t' = (\ldots, b_X, \ldots, b)$ zu $F(u)$ hinzu, das die Werte aus t enthält, d. h., $b_X = a_X$ für $X \subseteq X_v$. Zusätzlich enthält t' die Werte $b_{\{v'\}} = 1$ und $b_X = 0$ für alle restlichen $X \subseteq X_u$, d. h. für alle Mengen $X \subseteq X_u$ mit mindestens

zwei Elementen, die v' enthalten. Weiterhin setzen wir in t' den Wert $b = \max\{a, \sum_{X \subseteq X_u} b_X\}$.

ii. Dann fügen wir für jedes Tupel $t = (\ldots, a_X, \ldots, a) \in F(v)$ und jedes $X \subseteq X_v$, sodass $a_X = 1$ in $F(v)$ und $X \cup \{v'\}$ eine unabhängige Menge im Graphen G ist, ein Tupel $t' = (\ldots, b_X, \ldots, b)$ zu $F(u)$ hinzu, das die gleichen Werte wie das Tupel t enthält, für das jedoch $b_X = 0$ und zusätzlich $b_{X \cup \{v'\}} = 1$ gilt. Für alle noch nicht definierten b_Y mit $Y \subseteq X_u$ setzen wir $b_Y = 0$. Schließlich definieren wir $b = a$.

d) Falls u ein Forget-Knoten mit einem Kind v in T ist, sind $F(v)$ und $X_v - X_u = \{v'\}$ für ein $v' \in V_G$ bereits bekannt.
Hier gibt es in jeder Partition von X_v genau eine unabhängige Menge X, die den Knoten v' enthält. Falls X nur den Knoten v' enthält, so bleibt von X in X_u kein Knoten mehr übrig, und andernfalls wissen wir, dass $X - \{v'\}$ weiterhin eine unabhängige Menge mit mindestens einem Knoten aus X_u ist.
Formal definieren wir $F(u)$ wie folgt. Für jedes Tupel $t = (\ldots, a_X, \ldots, a) \in F(v)$ fügen wir ein Tupel $t' = (\ldots, b_X, \ldots, b)$ in $F(u)$ ein, das wie folgt definiert ist. Für jedes $X \subseteq X_v$, $X \neq \{v'\}$, setzen wir die Werte $b_{X - \{v'\}} = a_X$ und $b = a$.

In allen vier Fällen enthält die Menge $F(u)$ genau $2^{|X_u|}$ Einträge.

4. Mittels der Menge $F(r)$ für die Wurzel r von T können wir die Färbungszahl im durch $(\mathscr{X}, T)$ definierten Graphen G bestimmen durch

$$\chi(G) = \min_{(\ldots, a_X, \ldots, a) \in F(r)} a.$$

Satz 10.57. *Die Färbungszahl eines Graphen $G = (V, E)$ mit einer Baumweite von höchstens k kann bei gegebener Baumdekomposition in der Zeit*

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

für eine berechenbare Funktion f bestimmt werden.

Übung 10.58. Geben Sie die in Satz 10.57 erwähnte berechenbare Funktion f in Abhängigkeit von k an.

Korollar 10.59. *Das Problem p^*-tw-*PARTITION IN UNABHÄNGIGE MENGEN *ist fest-Parameter-berechenbar.*

Es gilt die folgende interessante Beziehung zwischen der Baumweite und der Färbungszahl von Graphen.

Satz 10.60 (Chlebikova [Chl02]). *Für jeden Graphen $G = (V, E)$ gilt:*

$$\chi(G) \leq \text{Baumweite}(G) + 1.$$

Beweis. Wir zeigen die Ungleichung induktiv über die Anzahl n der Knoten in G. Die Aussage gilt offenbar für alle Graphen mit $n \leq 2$ Knoten. Es sei G ein Graph mit $n+1$ Knoten. Nach Lemma 10.13 gibt es in G einen Knoten v in G mit $\deg(v) \leq$

Baumweite(G). Nach Induktionnsannahme bzw. der ersten Aussage von Satz 10.19 gilt:

$$\chi(G-v) \leq \text{Baumweite}(G-v)+1 \leq \text{Baumweite}(G)+1.$$

Da $\deg(v) \leq \text{Baumweite}(G)$ gilt, gibt es unter den Baumweite$(G)+1$ möglichen unabhängigen Mengen einer Partition von $G-v$ mindestens eine unabhängige Menge, mit der v nicht verbunden ist und der v somit zugeordnet werden kann. Also folgt auch

$$\chi(G) \leq \text{Baumweite}(G)+1,$$

und der Beweis ist abgeschlossen. ❑

10.5 Partition in Cliquen

Das Problem, die Knotenmenge in einem gegebenen Graphen G in möglichst wenige Cliquen zu partitionieren, d. h., die Cliquenüberdeckungszahl $\theta(G)$ zu bestimmen, kann bezüglich der Parametrisierung Baumweite des Eingabegraphen fast analog zum letzten Abschnitt gelöst werden.

p^*-tw-PARTITION IN CLIQUEN

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

Es seien $G = (V_G, E_G)$ ein Graph der Baumweite k und $(\mathscr{X}, T)$ eine Baumdekomposition der Weite k für G, mit dem Wurzelbaum $T = (V_T, E_T, r)$ und der Menge $\mathscr{X} = \{X_u \mid u \in V_T\}$ der Taschen.

Zur Lösung von p^*-tw-PARTITION IN CLIQUEN muss die Datenstruktur F aus Abschnitt 10.4 nur geringfügig modifiziert werden. Für Knoten $u \in V_T$ enthält $F(u)$ nun für jede Partition von V_{G_u} in Cliquen $V_1, \ldots, V_s$ ein 2^{k+1}-Tupel $t = (\ldots, a_X, \ldots, a)$, das für jede nicht leere Teilmenge $X \subseteq X_u$ einen booleschen Wert $a_X \in \{0,1\}$ und eine natürliche Zahl a enthält. Für die Partition $V_1, \ldots, V_s$ von V_{G_u} in Cliquen ist genau dann $a_X = 1$, wenn $V_i \cap X_u = X$ für ein $1 \leq i \leq s$ gilt. Der Wert a gibt die Anzahl der für den Teilgraphen G_u benötigten Cliquen an.

Offensichtlich kann $F(r)$ entlang des Baums T, ähnlich zu den in Abschnitt 10.4 angegebenen Operationen, berechnet und aus $F(r)$ leicht die Cliquenüberdeckungszahl $\theta(G)$ bestimmt werden.

Satz 10.61. *Die Cliquenüberdeckungszahl eines Graphen $G = (V,E)$ mit einer Baumweite von höchstens k kann bei gegebener Baumdekomposition in der Zeit*

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

für eine berechenbare Funktion f bestimmt werden.

Korollar 10.62. *Das Problem p^*-tw-*PARTITION IN CLIQUEN *ist fest-Parameter-berechenbar.*

10.6 MSO_2-definierbare Grapheigenschaften

Es gibt eine Vielzahl von weiteren Graphenproblemen, die bezüglich der Parametrisierung „Baumweite des Eingabegraphen“ mit FPT-Algorithmen gelöst werden können. Eines der grundlegendsten Resultate über die Existenz effizienter Algorithmen für Graphen mit beschränkter Baumweite ist das folgende Ergebnis von Courcelle [Cou92]. Alle Grapheigenschaften, die sich in monadischer Logik zweiter Ordnung (siehe Definition 4.39) auf relationale Graphenstrukturen $\lceil G \rceil$ über der Signatur $\{\overleftarrow{E}, \overrightarrow{E}\}$ (siehe Beispiel 4.42) definieren lassen, sind auf Graphen mit beschränkter Baumweite in linearer Zeit entscheidbar. Dies gilt auch für eine *erweiterte monadische Logik zweiter Ordnung*, die so genannte $EMSO_2$ (englisch: *extended monadic second order logic*), siehe auch Arnborg, Lagergren und Seese [ALS91]. Zu der Erweiterung gehört zum Beispiel auch die *Zähllogik* $CMSO_2$ (englisch: *counting monadic second order logic*) mit dem Mengenprädikat $\mathrm{Card}_{p,q}(X)$, siehe Abschnitt 4.7. Dieses Resultat von Courcelle besagt, dass es für jede positive Zahl $k \in \mathbb{N}$ und jede Grapheigenschaft Π aus MSO_2 einen Algorithmus gibt, der für Graphen $G = (V,E)$ mit einer Baumweite von höchstens k in der Zeit $\mathcal{O}(f(k) \cdot |V|)$ entscheidet, ob G die Eigenschaft Π erfüllt. Der Wert $f(k)$ ist selbstverständlich sehr groß bezüglich k, aber eben unabhängig von der Größe von G. Alle Grapheigenschaften aus MSO_2 sind deshalb fest-Parameter-berechenbar bezüglich dem Parameter „Baumweite des Eingabegraphen“. Dies gilt zum Beispiel auch für die Grapheigenschaften „Hamilton-Kreis”, siehe Behauptung 4.43. Da MSO_1 eine Teilmenge von MSO_2 ist (siehe Abschnitt 4.7), sind auch alle Grapheigenschaften aus MSO_1 fest-Parameter-berechenbar bezüglich dem Parameter „Baumweite des Eingabegraphen“.

$EMSO_2$

p^*-tw-HAMILTON-KREIS

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

Satz 10.63. *Das Problem p^*-tw-*HAMILTON-KREIS *ist fest-Parameter-berechenbar.*

10.7 Literaturhinweise

Das Konzept der Baumweite wurde bereits 1976 von Halin [Hal76] eingeführt. Die Definition von Robertson und Seymour [RS86], an die wir uns hier halten, ist die bekannteste. Die eng mit der Baumweite verwandten Begriffe der partiellen k-Bäume und der Verzweigungsweite wurden von Rose [Ros74] bzw. von Robertson und Seymour [RS91] eingeführt.

Fomin, Kratsch, Todinca und Villanger [FKTV08] zeigten, dass die Baumweite eines Graphen mit n Koten in der Zeit $\tilde{\mathcal{O}}(1,8899^n)$ berechnet werden kann. Weitere Ergebnisse über Graphklassen mit beschränkter Baumweite können zum Beispiel in den Arbeiten von Bodlaender [Bod86, Bod88, Bod98] nachgelesen werden. Auf Graphklassen mit beschränkter Baumweite eingeschränkte Algorithmen findet man beispielsweise in den Arbeiten [Arn85, AP89, Hag00, INZ03, KZN00, ZFN00].

Interessante Übersichtsarbeiten zur Baumweite und eng damit verknüpfter Parameter sind Bodlaender [Bod98] und Bodlaender und Koster [BK08] zu verdanken.