

Effiziente Algorithmen
für
baumstrukturierte Graphklassen

Inaugural-Dissertation
zur
Erlangung des Doktorgrades der
Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von
Frank Gurski
aus Essen

Düsseldorf
2003

Teile dieser Arbeit wurden bereits in den Schriften [GW00], [EGW01a], [EGW01b], [EGW03] und [GW04] veröffentlicht.

Gedruckt mit der Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. E. Wanke

Koreferent: Prof. Dr. J. Rothe

Tag der mündlichen Prüfung: 03.11.2003

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	7
2.1	Graphen	7
2.2	Baumweite – TW_k	10
2.3	Cliquenweite – CW_k	13
2.4	NLC-Weite – NLC_k	16
3	Inklusionen zwischen Graphklassen	21
3.1	Inklusionen zwischen den Graphklassen NLC_k und CW_k	21
3.2	Inklusionen zwischen den Graphklassen NLC_k , CW_k und TW_k	22
4	Modulare Dekomposition	29
4.1	Ersetzungsmechanismus in Graphen	29
4.2	Module	31
4.3	Quotientgraphen	33
5	NLC-Weite und Cliquenweite von Graphen und Graphklassen	35
5.1	NLC-Weite und Cliquenweite von speziellen Graphen	35
5.2	Graphklassen mit beschränkter NLC-Weite und Cliquenweite	41
5.2.1	NLC-Weite und Cliquenweite von Cographen	41
5.2.2	NLC-Weite und Cliquenweite von Graphen mit wenigen P_4	41
5.2.3	Weitere Graphklassen mit beschränkter NLC-Weite und Cliquenweite	46
5.3	Graphklassen mit unbeschränkter NLC-Weite und Cliquenweite	47
5.4	Graphoperationen auf Graphen mit beschränkter NLC-Weite	48
5.5	Abschätzungen der NLC-Weite und Cliquenweite beliebiger Graphen	53
6	Cliquenweite von Graphen mit beschränkter Baumweite	57
6.1	Erkennungsprobleme	57
6.2	Ersetzbarkeit von k -markierten l -terminalen Graphen	59
6.3	Normalform für Cliquenweite k -Ausdrücke	62
6.4	Überblick	66
6.5	Bestimmung des Verbindungstyps	71
6.6	Ergebnis	88

7	Effiziente Algorithmen für NP-schwere Graphenprobleme	93
7.1	Algorithmen	93
7.2	Ersetzbarkeit von k -markierten Graphen	95
7.3	Effiziente Algorithmen auf Graphen mit beschränkter Cliquesweite	96
7.4	Partition in unabhängige Mengen oder Cliques	99
7.5	Partition in unabhängige Mengen oder Cliques mit beschränkter Größe	101
7.6	l -Coloring	103
7.7	Partition in vollständig bipartite Graphen	104
7.8	Partition in Wälder	106
7.9	Partition in perfekte Matchings	108
7.10	Minimales maximales Matching	110
7.11	Hamilton Weg	111
7.12	Knotendisjunkte Wege	112
7.13	Knotengrad beschränkte Teilgraphen	116
7.14	Knotenpartition	117
7.15	Kantenpartition	118
7.16	Anmerkungen	120
8	NP-vollständige Probleme für Graphen mit beschränkter Cliquesweite	121
8.1	Knotendisjunkte Wege	121
8.1.1	Linearzeit Lösung für Cographen	121
8.1.2	NP-Vollständigkeit für NLC_4 und CW_7	123
	Symbolverzeichnis	129
	Literaturverzeichnis	131
	Index	137

Kapitel 1

Einleitung

Diese Arbeit beschäftigt sich mit der Cliquenweite und der NLC-Weite von ungerichteten Graphen. Die *Cliquenweite* wurde 1994 von Courcelle und Olariu in [CO00] eingeführt und basiert auf drei Operationen für knotenmarkierte Graphen. Die NLC-Weite wurde 1994 von Wanke in [Wan94c] eingeführt und basiert auf lediglich zwei Operationen für knotenmarkierte Graphen. In einem knotenmarkierten Graphen ist jeder Knoten mit einer natürlichen Zahl markiert. Die Operationen der Cliquenweite erlauben zwei bereits definierte Graphen G und J disjunkt zu vereinigen, $G \oplus J$, in einem bereits definierten Graphen G jeden mit i markierten Knoten mit jedem mit j markierten Knoten zu verbinden, $\eta_{i,j}(G)$, und alle mit i markierten Knoten mit j zu markieren, $\rho_{i \rightarrow j}(G)$. Bei den Operationen der NLC-Weite ist die disjunkte Vereinigung mit der Kanteneinfügeoperation kombiniert. Für eine Menge von Markierungspaaren $S \subseteq \{1, \dots, k\} \times \{1, \dots, k\}$ definiert die Operation $G \times_S J$ die disjunkte Vereinigung von G und J , in der anschließend für jedes Paar $(i, j) \in S$ alle i -markierten Knoten aus G mit allen j -markierten Knoten aus J verbunden werden. Die zweite Operation erlaubt die Ummarkierung aller Knoten in einem Schritt. Für eine Funktion $R : \{1, \dots, k\} \rightarrow \{1, \dots, k\}$ ist $\circ_R(G)$ der knotenmarkierte Graph G , in dem jeder ursprünglich mit i markierte Knoten mit $R(i)$ markiert ist. Der rekursive Aufbau der Graphen beginnt immer mit einzelnen markierten Knoten \bullet_i . Die Cliquenweite bzw. NLC-Weite eines Graphen G ist die minimale Anzahl von verschiedenen Markierungen, die insgesamt benötigt werden, um G mit den entsprechenden Operationen aufzubauen. Einen Ausdruck mit k Knotenmarkierungen gebildet mit den Operationen für die Cliquenweite bzw. mit den Operationen für die NLC-Weite nennen wir einen *Cliquenweite k -Ausdruck* bzw. *NLC-Weite k -Ausdruck*. Graphen mit beschränkter Cliquenweite bzw. beschränkter NLC-Weite besitzen aufgrund ihrer rekursiven Definition über k -Ausdrücke eine natürliche Baumstruktur.

Einige der interessantesten Ergebnisse dieser Arbeit stehen im Zusammenhang mit der Baumweite von Graphen. Die Menge der Graphen mit *Baumweite* höchstens k wurde 1986 von Robertson und Seymour in [RS86] definiert und ist äquivalent zur Menge der *partiellen k -Bäume* [Ros74]. Ein Graph ist ein partieller k -Baum, wenn er ein Teilgraph eines k -Baumes ist. k -Bäume sind wie folgt rekursiv definiert. Der vollständige Graph mit k Knoten ist ein k -Baum. Wird in einem k -Baum ein neuer Knoten eingefügt und dieser mit allen Knoten eines vollständigen Teilgraphen mit k Knoten verbunden, so ist das Ergebnis wieder ein k -Baum. Partielle k -Bäume bzw. Graphen mit beschränkter Baumweite besitzen aufgrund ihrer Definition ebenfalls eine Baumstruktur.

Die Cliquenweite, NLC-Weite bzw. Baumweite einer Graphklasse, d.h. einer Menge von

Graphen, ist die kleinste natürliche Zahl k , so dass jeder Graph der Graphklasse Cliquenweite, NLC-Weite bzw. Baumweite höchstens k hat. Falls kein solches k existiert, so sagen wir, dass die Graphklasse *unbeschränkte Cliquenweite*, *unbeschränkte NLC-Weite* bzw. *unbeschränkte Baumweite* hat.

Baumstrukturierte Graphen spielen in der Informatik eine wichtige Rolle, da eine unterliegende Baumstruktur sehr oft eine systematische und effiziente Analyse der entsprechenden Graphen ermöglicht. Für diese effiziente Analyse muss jedoch die Baumstruktur der Graphen explizit gegeben sein bzw. effizient bestimmt werden können. Die Grundidee besteht aus einer dynamischen Programmierung, in der alle Teillösungen für die Teilgraphen berechnet werden, die durch Teilbäume der Baumstruktur repräsentiert werden. Die maximale Anzahl der Teillösungen ist oft unabhängig von der Größe der Teilgraphen. In diesen Fällen können die entsprechenden Graphenprobleme auf den baumstrukturierten Graphen mit dynamischer Programmierung Bottom-Up entlang der gegebenen Baumstruktur in linearer Zeit gelöst werden.

Nach einem Ergebnis von Courcelle, Makowsky und Rotics in [CMR00] sind für alle MSO_1 -definierbaren Grapheigenschaften (Eigenschaften, die sich in monadischer Logik zweiter Ordnung mit Quantifizierungen über Knoten und Knotenmengen formulieren lassen) die Anzahlen der Teillösungen für Teilgraphen, die über Teilbäume einer gegebenen Cliquenweite Baumstruktur definiert werden, endlich. Damit können auf Graphklassen mit beschränkter Cliquenweite bei gegebener Baumstruktur alle MSO_1 -definierbaren Grapheigenschaften, wie zum Beispiel das Problem l -Färbbarkeit für festes l , in linearer Zeit gelöst werden.

Courcelle hat 1990 in [Cou90b] gezeigt, dass für alle MSO_2 -definierbaren Grapheigenschaften (Eigenschaften, die sich in monadischer Logik zweiter Ordnung mit Quantifizierungen über Knoten, Knotenmengen, Kanten und Kantenmengen formulieren lassen) die Anzahlen der Teillösungen für Teilgraphen, die über Teilbäume einer Baumweite Baumstruktur definiert werden, endlich sind. Für Graphen mit beschränkter Baumweite kann eine Baumstruktur nach einem Ergebnis von Bodlaender [Bod96] in linearer Zeit gefunden werden. Somit können auf Graphklassen mit beschränkter Baumweite alle MSO_2 -definierbaren Graphenprobleme, wie zum Beispiel das Hamilton Kreis Problem, in linearer Zeit gelöst werden.

Im Folgenden werden die Ergebnisse dieser Arbeit beschrieben.

In **Kapitel 3** untersuchen wir Inklusionsbeziehungen zwischen Graphklassen mit beschränkter Cliquenweite, Graphklassen mit beschränkter NLC-Weite und Graphklassen mit beschränkter Baumweite.

In [Gur98] wurde gezeigt, dass jeder Graph mit Cliquenweite k die NLC-Weite höchstens k hat und dass jeder Graph mit Cliquenweite k die NLC-Weite höchstens $2k$ hat. Somit hat eine Graphklasse beschränkte NLC-Weite genau dann, wenn sie beschränkte Cliquenweite hat.

Corneil und Rotics zeigen in [CR01], dass jeder Graph mit Baumweite k die Cliquenweite höchstens $3 \cdot 2^{k-1}$ und folglich auch NLC-Weite höchstens $3 \cdot 2^{k-1}$ hat. Somit hat jede Graphklasse mit beschränkter Baumweite auch beschränkte Cliquenweite und beschränkte NLC-Weite. Da der vollständige Graph mit n Knoten die NLC-Weite 1, Cliquenweite 2 und Baumweite $n - 1$ hat, haben Graphklassen mit beschränkter Cliquenweite und Graphklassen mit beschränkter NLC-Weite im Allgemeinen keine beschränkte Baumweite. Damit ist der Begriff der Cliquenweite eine echte Erweiterung gegenüber der Baumweite.

Wir zeigen in dieser Arbeit, dass jeder Graph mit NLC-Weite höchstens k , der den vollständig bipartiten Graphen $K_{n,n}$, für eine natürliche Zahl $n > 1$, nicht als Teilgraphen

enthält, die Baumweite höchstens $3k(n-1) - 1$ hat.

Unser Ergebnis impliziert unmittelbar die folgenden Charakterisierungen von Graphklassen mit beschränkter Baumweite. Ein planarer Graph mit Cliquesweite höchstens k hat Baumweite höchstens $6k - 1$, da er keinen $K_{3,3}$ als Teilgraph enthält. Ein Graph mit Cliquesweite höchstens k , in dem alle Knoten einen Grad höchstens d haben, hat Baumweite höchstens $3kd - 1$, da er keinen $K_{d+1,d+1}$ als Teilgraph enthält. Ein Graph mit Cliquesweite höchstens k , bei dem jeder Teilgraph mit n Knoten höchstens $l \cdot n$ Kanten hat (G ist l -sparse), hat Baumweite höchstens $6kl - 1$, da er keinen $K_{2l+1,2l+1}$ als Teilgraph enthält. Hier gilt auch die umgekehrte Richtung, da jeder Graph mit Baumweite höchstens k auch k -sparse ist.

In **Kapitel 4** wird die Anwendung der modularen Dekomposition auf Graphen mit beschränkter NLC-Weite beschrieben. Ein *Modul* M eines Graphen ist eine Menge von Knoten, die alle die gleichen Nachbarn außerhalb von M besitzen. *Triviale Module* sind Knotenmengen ohne, mit genau einem oder allen Knoten. Ein Graph G ist *prim*, wenn jedes Modul in G trivial ist. Die modulare Dekomposition zerlegt die Knotenmenge von G disjunkt in möglichst viele Module, so dass der *Quotientgraph*, den man erhält, wenn die Knoten innerhalb der Module zu einem Knoten zusammengefasst werden, prim ist. Die durch die Module induzierten Teilgraphen werden anschließend weiter modular zerlegt. Die modulare Dekomposition ist baumstrukturiert und kann nach einem Ergebnis von Cournier und Habib [CH94] in linearer Zeit gefunden werden. Das Interessante an der modularen Dekomposition im Zusammenhang mit der NLC-Weite ist die folgende Beobachtung. Ein Graph G hat genau dann NLC-Weite höchstens k , wenn alle Quotientgraphen der modularen Dekomposition von G NLC-Weite höchstens k haben. Eine Graphklasse hat somit beschränkte NLC-Weite, falls die Menge der primen Quotientgraphen der Graphklasse beschränkte NLC-Weite hat.

In **Kapitel 5** zeigen wir mit den bereits bekannten Quotientgraphen, dass P_4 -reduzierbare Graphen, P_4 -sparse Graphen und P_4 -tidy Graphen NLC-Weite höchstens 2 haben und dass auch $(q, q-4)$ -Graphen und $(q, q-3)$ -Graphen beschränkte NLC-Weite haben. Diese Graphklassen sind über die Anzahl der in einem Graphen zulässigen induzierten Wege der Länge drei definiert.

Graphen, die keinen induzierten Weg der Länge drei enthalten, werden *Cographen* genannt und können rekursiv definiert werden. Mit Hilfe dieser rekursiven Definition kann man zeigen, dass die Menge der Graphen mit NLC-Weite 1 und die Menge der Graphen mit Cliquesweite höchstens 2 genau die Menge der Cographen ist [Wan94c, CO00]. Weiterhin ist aus der Schrift [GR00] bekannt, dass distanzerhaltende Graphen Cliquesweite höchstens 3 haben.

Die unbeschränkte NLC-Weite einer Graphklasse kann man wie folgt durch die Anzahl der verschiedenen Nachbarschaften der Knoten bei balancierten Partitionen der Knotenmenge zeigen. Jeder binäre Baum B kann durch Entfernen von genau einer Kante in zwei Bäume geteilt werden, die je mindestens $\frac{1}{3}$ der Blätter von B enthalten. Durch Anwendung dieser Eigenschaft auf die Baumstruktur eines NLC-Weite k -Ausdrucks erhalten wir unmittelbar die folgende notwendige Bedingung für Graphklassen mit beschränkter NLC-Weite. Die Knotenmenge V eines Graphen $G = (V, E)$ mit NLC-Weite k kann man immer disjunkt in zwei Mengen U, W aufteilen, so dass $|U| \geq \frac{1}{3}|V|$, $|W| \geq \frac{1}{3}|V|$ und alle Knoten u in U höchstens k verschiedene Nachbarschaften $N_W(u) = \{w \in W \mid \{u, w\} \in E\}$ bezüglich der Knoten in W besitzen. Die Anzahl der Nachbarschaften ist durch k beschränkt, weil gleichmarkierte Knoten von allen nachfolgenden Operationen gleich behandelt werden. Mit Hilfe dieser Beobachtung wurde von verschiedenen Autoren gezeigt, dass Intervallgraphen, chordale Graphen, Permutationsgraphen, Splitgraphen, planare Graphen, Gitter und reguläre Graphen unbeschränkte

NLC-Weite haben.

Weiterhin geben wir Graphoperationen an, die die NLC-Weite und Cliquesweite von Graphen nur um eine Konstante oder in Abhängigkeit von k erhöhen.

Aus [Wan94c] und [CO00] ist bekannt, dass der Komplementgraph eines Graphen mit NLC-Weite k (Cliquesweite k) die NLC-Weite k (Cliquesweite höchstens $2k$) hat. Jeder induzierte Teilgraph eines Graphen mit NLC-Weite k (Cliquesweite k) hat NLC-Weite höchstens k (Cliquesweite höchstens k). Der *Potenzgraph* G^d eines Graphen hat die gleiche Knotenmenge wie G und enthält eine Kante zwischen jedem Knotenpaar, zwischen dem es in G einen Weg der Länge höchstens d gibt. Wir zeigen, dass der Graph G^d für einen Graphen G mit NLC-Weite k (Cliquesweite k) die NLC-Weite höchstens $k \cdot 2^{(d-1) \cdot k}$ (Cliquesweite höchstens $k \cdot 2^{d \cdot k}$) hat. Fügt man in einen Graphen mit NLC-Weite k (Cliquesweite k) eine zusätzliche Kante ein oder entfernt man eine vorhandene Kante, so hat der entstehende Graph NLC-Weite höchstens $k + 2$ (Cliquesweite höchstens $k + 2$). Fügt man in einen Graphen mit NLC-Weite k (Cliquesweite k) einen zusätzlichen Knoten mit einer beliebigen Nachbarschaft ein, so hat der neue Graph NLC-Weite höchstens $2k$ (Cliquesweite höchstens $2k + 1$).

Abschätzungen für die NLC-Weite und Cliquesweite beliebiger Graphen werden von Johansson in [Joh98] gezeigt. Jeder Graph mit n Knoten hat NLC-Weite höchstens $\lceil \frac{n}{2} \rceil$ und Cliquesweite höchstens $n - k$, falls $2^k < n - k$. Die exakte Berechnung der NLC-Weite und der Cliquesweite eines Graphen sind die größten offenen Probleme im Zusammenhang mit der NLC-Weite und der Cliquesweite.

In **Kapitel 6** beschäftigen wir uns mit dem Erkennungsproblem für Graphen mit beschränkter Cliquesweite. Für das Problem „Hat ein gegebener Graph die Cliquesweite (NLC-Weite) höchstens k ?“ sind bisher nur für festes $k \leq 3$ bzw. festes $k \leq 2$ effiziente Algorithmen bekannt. „NLC-Weite höchstens 1“ und „Cliquesweite höchstens 2“ entsprechen dem Erkennungsproblem für Cographen und sind damit in Linearzeit entscheidbar [CPS84]. „Cliquesweite höchstens 3“ ist für Graphen mit n Knoten und m Kanten in Zeit $O(n^2 \cdot m)$ entscheidbar [CHL⁺00]. Johansson hat in [Joh00] für das Problem „NLC-Weite höchstens 2“ auf Graphen mit n Knoten einen Algorithmus mit Laufzeit $O(n^4 \log n)$ angegeben. Für $k \geq 4$ bzw. $k \geq 3$ ist die Komplexität der Probleme „Cliquesweite höchstens k “ bzw. „NLC-Weite höchstens k “ noch offen, d.h. weder effizient gelöst noch als NP-vollständig eingeordnet worden.

Wir zeigen in dieser Arbeit, dass sich die Cliquesweite eines Graphen für Graphen mit beschränkter Baumweite effizient bestimmen lässt. Insbesondere wird gezeigt, dass die Grapheigenschaft „Cliquesweite höchstens k “ für jedes feste k die Menge aller l -terminalen Graphen für jedes feste $l \geq 0$ in endlich viele Äquivalenzklassen einteilt. Damit ist das Problem „Cliquesweite höchstens k “ ein sogenanntes *erkennbares* Problem und lässt sich mit Hilfe einer Baumstruktur in Linearzeit auf Graphen mit beschränkter Baumweite lösen. Ein *l -terminaler Graph* ist ein Graph mit l verschiedenen ausgezeichneten Knoten, die sogenannten Terminale. Zwei l -terminale Graphen G und H können zu einem Graphen $G \circ H$ verknüpft werden, indem man die disjunkte Vereinigung von G und H betrachtet und den i -ten terminalen Knoten von G mit dem i -ten terminalen Knoten von H , für $1 \leq i \leq l$, identifiziert. G und H werden *äquivalent* bezüglich der Grapheigenschaft Π genannt ($G \sim_{\Pi, l} H$), falls für alle l -terminalen Graphen J das Ergebnis für Π bei Eingabe von $G \circ J$ und $H \circ J$ das gleiche ist.

Wir definieren zu einem k -markierten l -terminalen Graphen G einen *Verbindungstyp*, der aus einer Menge von *Verbindungsbaum*en besteht. Ein Verbindungsbaum für G beinhaltet die wesentlichen Informationen wie der l -terminale Graph G und ein l -terminaler Graph H , so

dass der Graph $G \circ H$ Cliquesweite höchstens k hat, miteinander verbunden werden. Durch schrittweises Zusammenfassen von Teilwegen in einer Baumstruktur für einen Cliquesweite k -Ausdruck für $G \circ H$ erhalten wir einen Verbindungsbaum für G . Wir zeigen, dass zwei k -markierte l -terminale Graphen bezüglich der Grapheigenschaft Π_k „Cliquesweite höchstens k “ äquivalent sind, falls sie den gleichen Verbindungstyp haben. Damit kann die Anzahl der Äquivalenzklassen von $\sim_{\Pi_k, l}$ durch die Anzahl der verschiedenen Verbindungstypen für alle k -markierten l -terminalen Graphen nach oben abgeschätzt werden. Wir zeigen, dass es für jedes feste $k \geq 1$ und jedes feste $l \geq 0$ endlich viele verschiedene Verbindungstypen gibt, damit hat $\sim_{\Pi_k, l}$ endlich viele Äquivalenzklassen und unsere Behauptung folgt.

Da jeder Graph mit Baumweite r Cliquesweite höchstens $3 \cdot 2^{r-1}$ hat, folgt aus unserem Ergebnis auch ein Linearzeitalgorithmus um die Cliquesweite eines Graphen mit beschränkter Baumweite zu berechnen, indem man „Cliquesweite höchstens k “ für $k = 1, \dots, 3 \cdot 2^{r-1}$ testet.

In **Kapitel 7** nutzen wir die Baumstruktur von Graphen mit beschränkter NLC-Weite und Graphen mit beschränkter Cliquesweite um NP-schwere Probleme in polynomieller Zeit zu lösen.

Wir haben ein Schema zur Lösung von schweren Graphenproblemen auf rekursiv definierten Graphen entwickelt, mit dem wir eine Vielzahl nicht MSO_1 -definierbarer Probleme auf Graphen mit beschränkter NLC-Weite und Graphen mit beschränkter Cliquesweite effizient lösen können. Die algorithmische Grundidee ist die dynamische Programmierung. Sei Π ein Graphenproblem, das auf einem Graphen G gelöst werden soll, der durch einen gegebenen NLC-Weite k -Ausdruck X definiert ist. Falls es eine Abbildung F gibt, die jedem NLC-Weite k -Ausdruck X eine Struktur $F(X)$ zuordnet, so dass für alle NLC-Weite k -Ausdrücke X, Y (1.) die Größe von $F(X)$ polynomiell in der Größe von X beschränkt ist, (2.) die Lösung für Π auf dem durch X definierten Graphen in polynomieller Zeit aus $F(X)$ berechenbar ist, (3.) $F(\bullet_i)$ in Zeit $O(1)$, $F(X \times_S Y)$ in polynomieller Zeit aus $F(X)$ und $F(Y)$ und $F(\circ_R(X))$ in polynomieller Zeit aus $F(X)$ berechenbar sind, dann ist für jeden NLC-Weite k -Ausdruck X die Lösung für Π auf dem durch X definierten Graphen in polynomieller Zeit aus X berechenbar. Dieses Schema funktioniert für alle rekursiv definierten Graphenstrukturen, somit auch für Cliquesweite k -Ausdrücke, indem die Operationen \oplus , $\rho_{i \rightarrow j}$ und $\eta_{i,j}$ verwendet werden.

Mit Hilfe von Multimengen konnten wir für die folgenden NP-schweren, nicht MSO_1 -definierbaren Graphenprobleme geeignete Strukturen $F(X)$ für k -Ausdrücke X definieren: Partition in unabhängige Mengen (Knotenfärbung), Partition in Cliques, Partition in unabhängige Mengen mit beschränkter Größe, Partition in Cliques mit beschränkter Größe, l -Coloring, Partition in vollständig bipartite Graphen, Partition in Wälder, Partition in perfekte Matchings, Minimales maximales Matching, Hamilton Weg, Knotendisjunkte Wege, Knotengrad beschränkte Teilgraphen, Knotenpartition und Kantenpartition.

Wir speichern uns in der Struktur $F(X)$ Informationen über die Markierungen von Teilmengen der Knotenmenge des durch einen k -Ausdruck X gegebenen Eingabegraphen. Da wir voraussetzen, dass k fest ist, das heißt dass k nicht zu der Eingabe unserer Algorithmen gehört, haben unsere Algorithmen eine polynomielle Laufzeit, obwohl die Laufzeiten alle exponentiell in k sind.

Da bisher noch keine effizienten Algorithmen zur Bestimmung eines k -Ausdrucks bekannt sind, setzen wir bei unseren Lösungen voraus, dass mit einem Graphen auch der zugehörige k -Ausdruck mit den Operationen der Cliquesweite oder den Operationen der NLC-Weite gegeben ist. Für distanzerhaltende Graphen kann nach einem Ergebnis von Golumbic und Rotics in [GR00] ein 3-Ausdruck in linearer Zeit gefunden werden und somit kann auf distan-

zerhaltenden Graphen jedes MSO_1 -definierbare Problem in linearer Zeit und jedes unserer obigen Probleme in polynomieller Zeit gelöst werden.

In **Kapitel 8** betrachten wir Graphenprobleme, die für Graphenklassen mit beschränkter Cliquesweite und Graphklassen mit beschränkter NLC-Weite NP-vollständig sind. Da Cographen und Graphklassen mit beschränkter Baumweite beschränkte Cliquesweite und beschränkte NLC-Weite haben, sind die Graphenprobleme, die für Cographen und für Graphklassen mit beschränkter Baumweite NP-vollständig sind, auch für Graphklassen mit beschränkter Cliquesweite und für Graphklassen mit beschränkter NLC-Weite NP-vollständig. Bisher sind jedoch so gut wie keine Graphenprobleme bekannt, die auf Graphklassen mit beschränkter Cliquesweite und auf Graphklassen mit beschränkter NLC-Weite NP-vollständig sind, aber auf Cographen und Graphklassen mit beschränkter Baumweite in Polynomzeit entschieden werden können.

Nishizeki, Vygen und Zhou haben in [NVZ01] gezeigt, dass das Kantendisjunkte Wege Problem für beliebig viele Knotenpaare für Graphen mit Baumweite höchstens 2 NP-vollständig ist. Das Kantendisjunkte Wege Problem kann auf einem Graphen G mit einem Algorithmus für das Knotendisjunkte Wege Problem auf dem Linegraphen von G gelöst werden. Der *Linegraph* eines Graphen G hat einen Knoten für jede Kante in G und zwei Knoten sind adjazent, wenn die entsprechenden Kanten in G adjazent sind. Wir zeigen, dass der Linegraph eines Graphen mit Baumweite k die NLC-Weite höchstens $k + 2$ und die Cliquesweite höchstens $2k + 3$ hat, indem wir zu einem Graphen G mit Baumweite k einen NLC-Weite $k + 2$ -Ausdruck bzw. einen Cliquesweite $2k + 3$ -Ausdruck für den Linegraphen von G angeben. Daraus folgt, dass das Knotendisjunkte Wege Problem auf Graphen mit NLC-Weite höchstens 4 und auf Graphen mit Cliquesweite höchstens 7 NP-vollständig ist. Wir zeigen weiterhin, dass das Knotendisjunkte Wege Problem auf Cographen in linearer Zeit lösbar ist. Auf Graphklassen mit beschränkter Baumweite ist das Knotendisjunkte Wege Problem ebenfalls in linearer Zeit lösbar [Sch94].

Kapitel 2

Grundlagen

In diesem Kapitel stellen wir graphentheoretische Grundbegriffe zusammen, definieren die Baumweite, NLC-Weite und Cliquesweite von Graphen und die zugehörigen Baumstrukturen.

2.1 Graphen

Ein *Graph* ist ein Paar $G = (V_G, E_G)$. V_G ist eine Menge von *Knoten* und $E_G \subseteq \{\{u, v\} \mid u, v \in V_G, u \neq v\}$ ist eine Menge von *Kanten*. Wir betrachten in dieser Arbeit ausschließlich *endliche Graphen*, d.h. Graphen in denen die Knotenmenge V_G endlich ist. Ein Knoten u heißt mit einer Kante e *inzident*, wenn $u \in e$ gilt. Zwei Kanten heißen *adjazent*, falls sie einen Knoten gemeinsam haben. Zwei Knoten u, v sind *adjazent*, falls $\{u, v\} \in E_G$. Die Menge der *Nachbarn* $\{v \mid \{u, v\} \in E_G\}$ eines Knoten u in einem Graphen G bezeichnen wir als *Nachbarschaft* von u bzw. mit $N_G(u)$, oder kurz mit $N(u)$ ¹. Der *Grad* $\deg_G(v)$ eines Knoten v in dem Graphen G ist die Anzahl der Nachbarn des Knoten v in G . Wenn in einem Graphen jeder Knoten den Grad k hat, so bezeichnen wir den Graphen als *k-regulär*. Sind in einem Graphen je zwei Knoten adjazent, so nennen wir den Graphen *vollständig* oder auch eine *Clique*. Wir bezeichnen einen vollständigen Graphen mit n Knoten mit K_n . Einen Graphen, in dem keine zwei Knoten adjazent sind, nennen wir *vollständig unzusammenhängend*. Eine Knotenmenge in einem Graphen, die keine zwei adjazenten Knoten enthält, nennen wir *unabhängig*.

Bildlich kann man einen Graphen darstellen, indem man seine Knoten als Punkte zeichnet und zwei dieser Punkte genau dann durch eine Linie verbindet, wenn die entsprechende Knotenmenge eine Kante ist. Abbildung 2.1 zeigt eine Clique mit vier Knoten und eine Clique mit sechs Knoten.

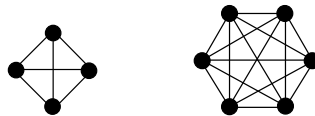


Abbildung 2.1: Die Clique K_4 und die Clique K_6 .

Eine aus Graphen bestehende Menge bezeichnet man als *Graphklasse*. Es gibt viele spezielle Graphklassen, die durch Einschränkungen der allgemeinen Graphdefinition entstehen.

¹Bei weiteren Bezeichnungen, in denen der Graph als Index angegeben wird, lassen wir den Index weg, wenn der Bezug klar ist.

Eine gute Übersicht über bekannte Graphklassen liefert das Buch [BLS99] von Brandstädt, Le und Spinrad.

Ein Graph $J = (V_J, E_J)$ ist ein *Teilgraph* von G , falls V_J eine Teilmenge von V_G ist und E_J eine Teilmenge von E_G ist. J ist ein *induzierter Teilgraph* von G , falls J ein Teilgraph von G ist und zusätzlich $E_J = \{\{u, v\} \in E_G \mid u, v \in V_J\}$ gilt. Für eine Teilmenge der Knotenmenge $U \subseteq V_G$ eines Graphen G bezeichne $G(U)$ den durch die Knoten in der Menge U induzierten Teilgraphen von G . Jeder induzierte Teilgraph ist ein Teilgraph, aber nicht umgekehrt. In Abbildung 2.2 ist ein Graph (links), ein Teilgraph des Graphen (mitte) und ein induzierter Teilgraph des Graphen (rechts) dargestellt.

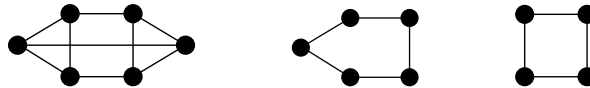


Abbildung 2.2: Ein Graph (links) mit Teilgraph (mitte) und induziertem Teilgraph (rechts)

Ein Graph $J = (V_J, E_J)$ ist ein *Minor* eines Graphen $G = (V_G, E_G)$, falls J aus G durch eine Folge von Kantenlöschungen, Kantenkontraktionen, d.h. Identifikation zweier adjazenter Knoten, und Löschen von isolierten Knoten erhalten werden kann. Jeder Teilgraph eines Graphen G ist somit auch ein Minor von G . Eine Graphklasse \mathcal{L} heißt *abgeschlossen bezüglich Minoren (Teilgraphen, induzierten Teilgraphen)*, falls für jeden Graphen $G \in \mathcal{L}$ alle Minoren (Teilgraphen, induzierten Teilgraphen) von G in \mathcal{L} enthalten sind. Zwei Graphen $G = (V_G, E_G)$ und $H = (V_H, E_H)$ heißen *isomorph* ($G \approx H$), wenn es eine Bijektion $b: V_G \rightarrow V_H$ gibt, so dass für alle $u, v \in V_G$: $\{u, v\} \in E_G \Leftrightarrow \{b(u), b(v)\} \in E_H$.

Es sei \mathcal{G} eine Graphklasse und H ein Graph. \mathcal{G} heißt *H-frei*, falls kein Graph aus \mathcal{G} den Graphen H als induzierten Teilgraphen enthält. Für ein p -Tupel (H_1, \dots, H_p) von Graphen H_i , $1 \leq i \leq p$, heißt eine Graphklasse \mathcal{G} (H_1, \dots, H_p) -frei, falls kein Graph aus \mathcal{G} einen der Graphen H_i , $1 \leq i \leq p$ als induzierten Teilgraphen enthält.

Ein Graph G ist *bipartite*, falls seine Knotenmenge in zwei disjunkte Mengen V_1 und V_2 aufgeteilt werden kann, so dass für jede Kante $\{u, v\}$ in G entweder $u \in V_1$ und $v \in V_2$ oder $v \in V_1$ und $u \in V_2$ gilt. Es gibt also ausschließlich Kanten zwischen V_1 und V_2 . Sind je zwei Knoten aus V_1 und V_2 in einem bipartiten Graphen adjazent, so nennen wir den Graphen *vollständig bipartite*. Haben die Mengen V_1 und V_2 in einem vollständig bipartiten Graphen die Mächtigkeiten n bzw. m , so bezeichnen wir den Graphen kurz mit $K_{n,m}$. Abbildung 2.3 zeigt die vollständig bipartiten Graphen $K_{2,3}$ und $K_{4,4}$.



Abbildung 2.3: Die vollständig bipartiten Graphen $K_{2,3}$ und $K_{4,4}$.

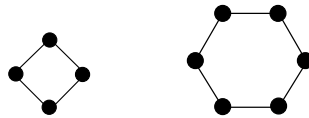
Der *Komplementgraph* \overline{G} eines Graphen G ist der Graph mit der Knotenmenge V_G in dem zwei Knoten genau dann adjazent sind wenn sie in G nicht adjazent sind. Einige Autoren bezeichnen den Komplementgraph eines Graphen G auch mit *co-G*.

Ein (einfacher) *Weg* ist ein Graph $P_n = (V, E)$ mit Knotenmenge $V = \{v_1, \dots, v_n\}$ und

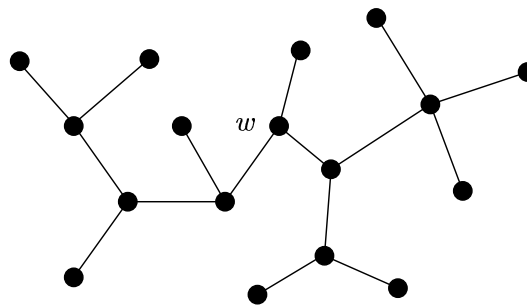
einer Kantenmenge $E = \{\{v_i, v_{i+1}\} \mid 1 \leq i \leq n-1\}$, wir schreiben auch kurz $P_n = (v_1, \dots, v_n)$. v_1 und v_n heißen *Anfangs-* bzw. *Endknoten* des Weges. Die *Länge* eines Weges ist die Anzahl seiner Kanten. Abbildung 2.4 zeigt einen Weg mit vier Knoten und einen Weg mit fünf Knoten. Die *Distanz* ($\text{dist}_G(u, v)$) zweier Knoten u, v eines Graphen G ist die Länge des kürzesten Weges in G der u und v verbindet. Der *Durchmesser* eines Graphen ist die maximale Distanz aller Knotenpaare aus G .

Abbildung 2.4: Der Weg P_4 und der Weg P_5

Ein Graph G heißt *zusammenhängend*, falls es für je zwei Knoten u, v des Graphen einen Weg P (als Teilgraphen) in G gibt, so dass u Anfangsknoten und v Endknoten von P ist. Falls es nicht zwischen jedem Knotenpaar einen solchen Weg gibt, so heißt der Graph *unzusammenhängend*. Die zusammenhängenden induzierten Teilgraphen eines Graphen werden als *Zusammenhangskomponenten* bezeichnet. Ein Weg in dem Anfangs- und Endknoten durch eine Kante verbunden werden, nennt man *Kreis*. Die *Länge* eines Kreises ist die Anzahl seiner Kanten. Abbildung 2.5 zeigt einen Kreis mit vier Knoten und einen Kreis mit sechs Knoten. Wir bezeichnen einen Kreis mit n Knoten mit C_n .

Abbildung 2.5: Der Kreis C_4 und der Kreis C_6

Einen kreisfreien Graphen, d.h. ein Graph der keinen Kreis C_n , $n \geq 3$ als Teilgraphen enthält, bezeichnet man als *Wald*. Ein Graph der zusammenhängend und kreisfrei ist, heißt *Baum*. Abbildung 2.6 zeigt einen Baum. Die Knoten mit Grad 1 in einem Baum heißen *Blätter*, die übrigen Knoten heißen *innere Knoten*. Gelegentlich ist es sinnvoll einen speziellen Knoten des Baumes auszuzeichnen, indem man diesen Knoten *Wurzel* nennt.

Abbildung 2.6: Ein Baum mit Wurzel w

Zur Veranschaulichung der Baumstruktur von Graphen mit beschränkter Cliquesweite und Graphen mit beschränkter NLC-Weite betrachten wir gerichtete Bäume, d.h. Bäume

deren Kanten Richtungen enthalten. Dazu benötigen wir noch die folgenden Begriffe für gerichtete Graphen.

Ein *gerichteter Graph* ist ein Paar $G = (V_G, E_G)$. V_G ist eine Menge von *Knoten* und $E_G \subseteq \{(u, v) \mid u, v \in V_G, u \neq v\}$ ist eine Menge von *gerichteten Kanten*. Ein einfacher *gerichteter Weg* ist ein Graph $P = (\{v_1, \dots, v_n\}, \{(v_i, v_{i+1}) \mid 1 \leq i \leq n-1\})$ mit $v_i \neq v_j$ für $1 \leq i < j \leq n$, wir schreiben auch kurz $P = (v_1, \dots, v_n)$. v_1 heißt *Startknoten* und v_n heißt *Endknoten* von P . Ein *gerichteter Kreis* ist ein gerichteter Weg in dem $v_1 = v_n$ gilt. Ein *gerichteter Baum* ist ein gerichteter Graph, dessen zugrunde liegender ungerichteter Graph, den wir erhalten wenn wir die Richtungen der Kanten vernachlässigen, ein Baum ist. Gerichtete Graphen kann man anschaulich darstellen, indem man die Knoten als Punkte zeichnet und vom Punkt zu Knoten u zum Punkt zu Knoten v eine Linie mit Pfeil nach v zeichnet, genau dann wenn das Paar (u, v) eine Kante ist. Wir betrachten gerichtete Bäume mit einem ausgezeichneten Knoten, den wir *Wurzel* nennen. Es sei T ein gerichteter Baum mit Wurzel w und T' der zugrunde liegende ungerichtete Baum. Für eine Kante $\{u, v\}$ in T' mit $\text{dist}_{T'}(u, w) < \text{dist}_{T'}(v, w)$ heißt der Knoten u *Vater* von v und der Knoten v *Sohn* von u in T . Abbildung 2.7 zeigt einen gerichteten Baum mit Wurzel w .

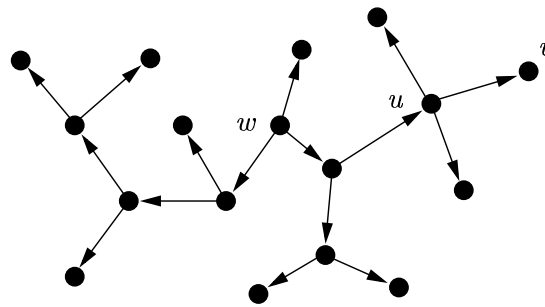


Abbildung 2.7: Ein gerichteter Baum mit Wurzel w . Knoten u ist Vater von Knoten v und Knoten v ist Sohn von Knoten u .

2.2 Baumweite – TW_k

Eine der bekanntesten baumstrukturierten Graphklassen ist die Menge der Graphen mit *Baumweite höchstens k* , für eine natürliche Zahl k . Der Begriff Baumweite wurde 1986 von Robertson und Seymour in [RS86] eingeführt.

Definition 2.2.1 (Baumweite, TW_k [RS86]) *Es sei $G = (V_G, E_G)$ ein Graph, $T = (V_T, E_T)$ ein Baum und $\mathcal{X} = \{X_u \mid X_u \subseteq V_G, u \in V_T\}$. Wir nennen das Paar (\mathcal{X}, T) eine Baumdekomposition für G , wenn die folgenden drei Bedingungen erfüllt sind.*

1. $\cup_{u \in V_T} X_u = V_G$.
2. Für jede Kante $\{w_1, w_2\} \in E_G$, gibt es einen Knoten $u \in V_T$, so dass $w_1 \in X_u$ 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.

Als Weite einer Baumdekomposition ($\mathcal{X} = \{X_u \mid u \in V_T\}$, $T = (V_T, E_T)$) bezeichnen wir die Zahl

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

Die Baumweite eines Graphen G (kurz Baumweite(G)) ist die minimale Weite aller möglichen Baumdekompositionen für G . Der Baum T wird Dekompositionsbaum für G bezüglich k genannt. Die Baumstruktur der Graphen mit Baumweite k wird durch den zugehörigen Dekompositionsbaum gegeben.

Die Menge aller Graphen mit Baumweite höchstens k bezeichnen wir mit TW_k . (Die Abkürzung TW stammt aus der englischen Bezeichnung *treewidth*.)

Eine Graphklasse \mathcal{L} hat beschränkte Baumweite, falls es eine natürliche Zahl k gibt, so dass jeder Graph in \mathcal{L} Baumweite höchstens k hat, d.h. es gibt ein $k \in \mathbb{N}$, so dass $\mathcal{L} \subseteq TW_k$. Falls ein solches k existiert, so ist das minimale k die Baumweite von \mathcal{L} , anderenfalls hat \mathcal{L} unbeschränkte Baumweite.

Abbildung 2.8 zeigt einen Graphen G mit Baumweite 2 und eine Baumdekomposition (\mathcal{X}, T) der Weite 2 für G . Wälder haben Baumweite 1. Eine Clique mit n Knoten hat Baumweite $n - 1$. Damit hat die Menge aller Cliques und auch die Menge aller Cographen unbeschränkte Baumweite. Graphen mit beschränkter Baumweite sind abgeschlossen bezüglich induzierten Teilgraphen, Teilgraphen und Minoren.

Die Klasse der Graphen mit Baumweite höchstens k ist äquivalent zur Klasse der partiellen k -Bäume [Ros74].

Definition 2.2.2 ((partieller) k -Baum) Eine Clique die aus k Knoten besteht ist ein k -Baum. Verbindet man einen neuen Knoten mit allen Knoten einer Clique der Größe k in einem k -Baum, so erhält man wieder einen k -Baum. Ein partieller k -Baum ist ein Teilgraph eines k -Baumes.

Die nun folgenden Definitionen von Graphen mit beschränkter Cliquesweite und Graphen mit beschränkter NLC-Weite resultieren aus einer Reihe von Forschungsarbeiten aus dem Bereich der Graphersetzungs-systeme. In der Theorie der Graphersetzungs-systeme [Roz97] unterscheidet man zwischen *Hyperkantenersetzungs-systemen (HR-Systemen)* und *Knotenersetzungs-systemen (NR-Systemen)*. Der hauptsächliche Unterschied der beiden Modelle besteht in der Art, in der ein Graph (festgelegt durch die rechte Seite einer Produktion) nach dem Entfernen einer nichtterminalen Kante bzw. eines nichtterminalen Knoten (spezifiziert durch die linke Seite einer Produktion) in der Umgebung eingebettet wird. Die Umgebung wird durch die Knoten bestimmt, die mit der ausgetauschten Kante inzident bzw. mit dem ausgetauschten Knoten adjazent waren. In Kantenersetzungs-systemen werden bei der Einbettung bestimmte Knoten des eingesetzten Graphen mit der Umgebung identifiziert, bei den Knotenersetzungs-systemen dagegen werden neue Kanten zwischen den Knoten des eingesetzten Graphen und der Umgebung hinzugefügt. Das Hinzufügen neuer Kanten wird dabei über Knotenmarkierungen kontrolliert. Die Ableitungsbäume spiegeln die Baumstrukturen der Graphen wieder.

Courcelle zeigte in den Schriften [Cou90a, Cou90b], dass alle MSO_2 -definierbaren Eigenschaften (MSO_1 -definierbaren Eigenschaften) auf den Sprachen kontextfreier HR-Systeme (kontextfreier NR-Systeme) entscheidbar sind. Daraus folgte unmittelbar ein Linearzeit-Algorithmus für jeden Graphen, der mit einem gegebenen HR-System definiert werden kann,

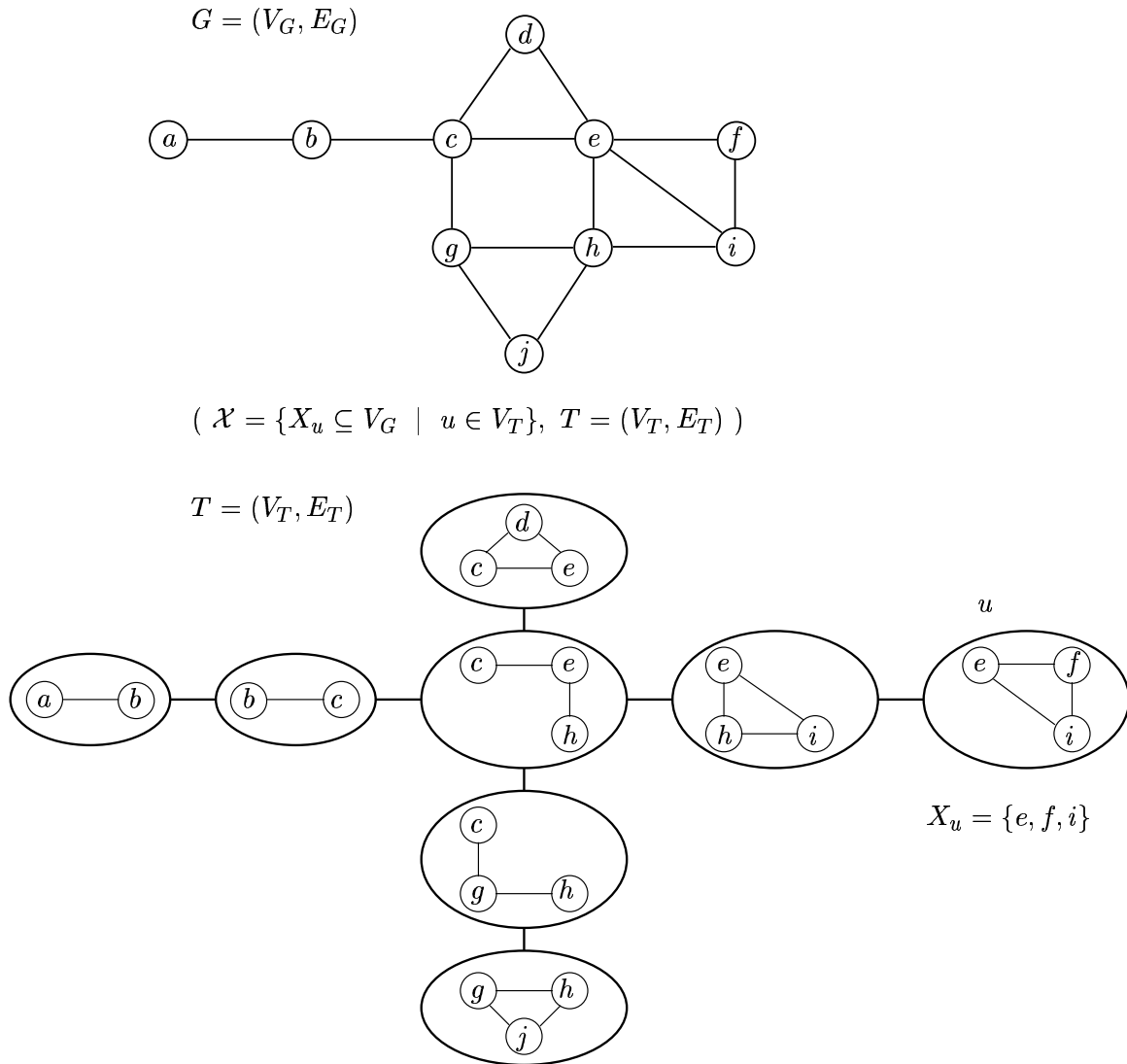


Abbildung 2.8: Ein Graph $G = (\{a, b, c, d, e, f, g, h, i, j\}, \{\{a, b\}, \{b, c\}, \{c, d\}, \{c, e\}, \{d, e\}, \{e, f\}, \{e, h\}, \{e, i\}, \{f, i\}, \{h, i\}, \{h, g\}, \{h, j\}, \{g, h\}, \{c, g\}\})$ mit Baumweite 2 und eine dazugehörige Baumdekomposition (\mathcal{X}, T) der Weite 2. Die Knoten in T entsprechen Teilmengen der Knotenmenge von G . Zum Beispiel besteht die Menge X_u zu Knoten u in T aus der Knotenmenge $\{e, f, i\} \subseteq V_G$.

siehe auch [LW93, Wan94b]. Voraussetzung ist jedoch die Vorgabe der Ableitungsstruktur. Es war bekannt, dass für jedes HR-System eine Zahl k existiert, so dass alle Graphen in der Sprache des Systems eine Baumweite höchstens k besitzen. So ergab sich das Ergebnis, dass alle MSO₂-definierbaren Eigenschaften auf Graphen mit beschränkter Baumweite in linearer Zeit entscheidbar sind, siehe Kapitel 6.2. Für NR-Systeme war jedoch keine entsprechende Graphklasse bekannt. Dies führte zur Definition der Graphen mit beschränkter Cliquesweite in [CO00] bzw. zur Definition der Graphen mit beschränkter NLC-Weite in [Wan94c].

2.3 Cliquesweite – CW_k

Um den Verknüpfungsmechanismus der Cliquesweite zu definieren benötigen wir Graphen, deren Knoten markiert sind.

Es sei k eine natürliche Zahl und $[k]$ bezeichne die Menge der ganzen Zahlen $\{1, \dots, k\}$. Ein k -markierter Graph $G = (V_G, E_G, \text{lab}_G)$ ist ein Graph $G = (V_G, E_G)$ dessen Knoten mit einer Markierungsabbildung $\text{lab}_G : V_G \rightarrow [k]$ markiert werden. Ein Graph, der aus genau einem mit $i \in [k]$ markierten Knoten besteht, wird im Folgenden mit \bullet_i bezeichnet. Ein k -markierter Graph $J = (V_J, E_J, \text{lab}_J)$ heißt k -markierter Teilgraph von G , falls $V_J \subseteq V_G$, $E_J \subseteq E_G$ und $\text{lab}_J(u) = \text{lab}_G(u)$ für alle $u \in V_J$. J ist ein k -markierter induzierter Teilgraph, falls zusätzlich noch $E_J = \{\{u, v\} \in E_G \mid u, v \in V_J\}$ gilt. G und J sind isomorph, falls es eine Bijektion $b : V_G \rightarrow V_J$ gibt, so dass $\{u, v\} \in E_G$ genau dann wenn $\{b(u), b(v)\} \in E_J$ und für jeden Knoten $u \in V_G$ gilt $\text{lab}_G(u) = \text{lab}_J(b(u))$.

Der Begriff der *Cliquesweite* wurde 1994 von Courcelle und Olariu in [CO00] eingeführt. Die zur Definition der Cliquesweite verwendeten Operationen sind bereits 1993 von Courcelle, Engelfriet und Rozenberg in [CER93, Roz97] zur Beschreibung von kontextfreien Graphsprachen mit verschachtelten rekursiven Funktionen eingesetzt worden.

Definition 2.3.1 (Cliquesweite, CW_k [CO00]) Die Graphklasse CW_k ist rekursiv wie folgt definiert.

1. Der k -markierte Graph \bullet_i ist für $i \in [k]$ in CW_k .
2. Es seien $G = (V_G, E_G, \text{lab}_G) \in CW_k$ und $J = (V_J, E_J, \text{lab}_J) \in CW_k$ zwei knotendisjunkte k -markierte Graphen. Dann ist der k -markierte Graph $G \oplus J := (V', E', \text{lab}')$ mit $V' := V_G \cup V_J$, $E' := E_G \cup E_J$ und

$$\text{lab}'(u) := \begin{cases} \text{lab}_G(u) & \text{falls } u \in V_G \\ \text{lab}_J(u) & \text{falls } u \in V_J \end{cases}, \forall u \in V'$$

in CW_k .

3. Es seien $i, j \in [k]$ zwei verschiedene natürliche Zahlen und $G = (V_G, E_G, \text{lab}_G) \in CW_k$ ein k -markierter Graph, dann ist

(a) der k -markierte Graph $\rho_{i \rightarrow j}(G) := (V_G, E_G, \text{lab}')$ mit

$$\text{lab}'(u) := \begin{cases} \text{lab}_G(u) & \text{falls } \text{lab}_G(u) \neq i \\ j & \text{falls } \text{lab}_G(u) = i \end{cases}, \forall u \in V_G$$

in CW_k und

(b) der k -markierte Graph $\eta_{i,j}(G) := (V_G, E', \text{lab}_G) \in CW_k$ mit

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

in CW_k .

Einen Ausdruck X mit den Operationen \bullet_i , \oplus , $\eta_{i,j}$, $\rho_{i \rightarrow j}$ mit $i, j \in [k]$, nennen wir *Cliquesweite k -Ausdruck* oder kurz *k -Ausdruck*. Der Graph der durch einen Ausdruck X definiert wird, wird mit $\text{val}(X)$ bezeichnet. Die Menge CW_k besteht somit aus der Menge aller Graphen $\text{val}(X)$ für Cliquesweite k -Ausdrücke X .

Ein k -markierter Graph G hat Cliquesweite höchstens k , falls $G \in CW_k$, d.h. CW_k besteht aus der Menge aller k -markierten Graphen mit Cliquesweite höchstens k . Die Cliquesweite eines markierten Graphen G (kurz Cliquesweite(G)) ist die kleinste natürliche Zahl k , so dass $G \in CW_k$. Die Baumstruktur für G ist durch die Baumstruktur des Ausdrucks für G gegeben.

Die Operation $\eta_{i,j}$ wird auch Kanteneinfügeoperation, die Operation $\rho_{i \rightarrow j}$ wird Ummarkierungsoperation und die Operation \oplus wird Vereinigungsoperation genannt.

Eine Graphklasse \mathcal{L} hat beschränkte Cliquesweite, falls es eine natürliche Zahl k gibt, so dass jeder Graph in \mathcal{L} Cliquesweite höchstens k hat, d.h. es gibt ein $k \in \mathbb{N}$ mit $\mathcal{L} \subseteq CW_k$. Falls ein solches k existiert, so ist das minimale k die Cliquesweite von \mathcal{L} , sonst hat \mathcal{L} unbeschränkte Cliquesweite.

Der Cliquesweite 2-Ausdruck

$$X_1 = \eta_{1,2}(\bullet_1 \oplus \bullet_2)$$

definiert den 2-markierten Graphen G_1 in Abbildung 2.9, der 2-markierte Graph G_1 ist somit in der Graphklasse CW_2 enthalten und hat Cliquesweite 2. Der Cliquesweite 2-Ausdruck

$$X_2 = \eta_{1,2}((\rho_{2 \rightarrow 1}(\eta_{1,2}(\bullet_1 \oplus \bullet_2))) \oplus \bullet_2)$$

definiert den 2-markierten Graphen G_2 in Abbildung 2.10, der 2-markierte Graph G_2 ist somit aus CW_2 und hat Cliquesweite 2. Der folgende Cliquesweite 3-Ausdruck beschreibt den 3-markierten Graphen $G_3 \in CW_3$ in Abbildung 2.11.

$$X_3 = \rho_{1 \rightarrow 2}(\eta_{2,3}(((\eta_{1,2}(\bullet_1 \oplus \bullet_2)) \oplus (\eta_{1,2}(\bullet_1 \oplus \bullet_2))) \oplus \bullet_3))$$

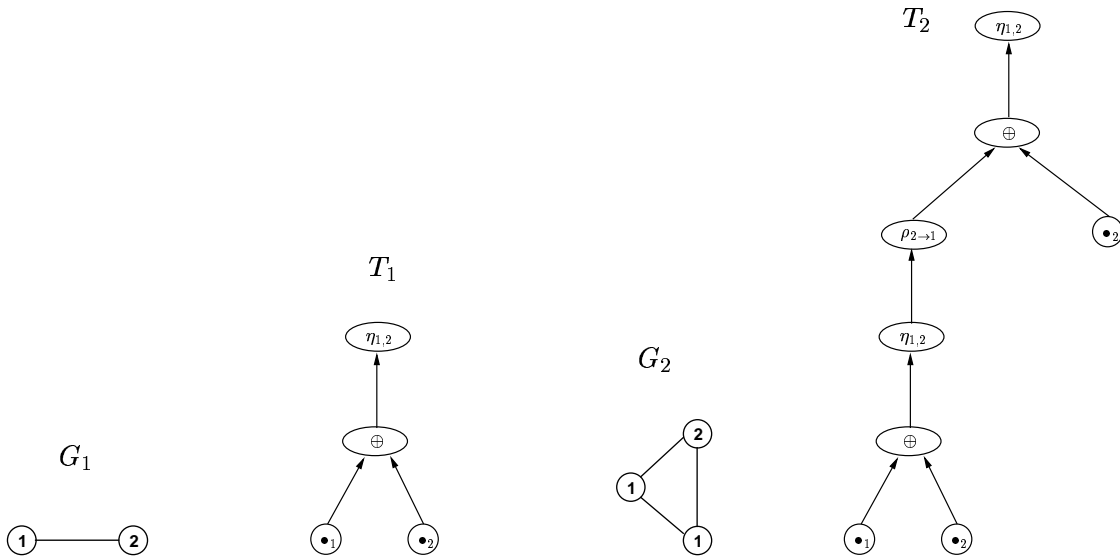


Abbildung 2.9: $X_1 = \eta_{1,2}(\bullet_1 \oplus \bullet_2)$, $G_1 = \text{val}(X_1)$ und der Cliquesweite 2-Ausdrucksbaum T_1 für X_1

Abbildung 2.10: $X_2 = \eta_{1,2}((\rho_{2 \rightarrow 1}(\eta_{1,2}(\bullet_1 \oplus \bullet_2))) \oplus \bullet_2)$, $G_2 = \text{val}(X_2)$ und der Cliquesweite 2-Ausdrucksbaum T_2 für X_2

Ein unmarkierter Graph $G = (V, E)$ hat Cliquesweite höchstens k , falls es eine Markierung $\text{lab} : V \rightarrow [k]$ der Knoten von G gibt, so dass der markierte Graph $G' = (V, E, \text{lab})$

Cliquesweite höchstens k hat. Die *Cliquesweite* eines *unmarkierten* Graphen $G = (V, E)$ (kurz *Cliquesweite*(G)) ist die kleinste natürliche Zahl k , so dass es eine Markierung $\text{lab} : V \rightarrow [k]$ gibt, so dass der markierte Graph (V, E, lab) Cliquesweite höchstens k hat.

Hat ein markierter Graph G Cliquesweite höchstens k und hat G zwei verschieden markierte Knoten die mit i bzw. j markiert sind, dann hat $\rho_{i \rightarrow j}(G)$ ebenfalls Cliquesweite höchstens k . Da eine Ummarkierung die Cliquesweite eines Graphen nicht erhöht, können wir einen unmarkierten Graphen auch als Graphen betrachten dessen Knoten alle mit 1 markiert sind. Damit können wir den Begriff Graph im Zusammenhang mit der Cliquesweite ohne Konflikte sowohl für markierte und unmarkierte Graphen verwenden. Für einen markierten Graphen $G = (V, E, \text{lab})$ bezeichnen wir mit $\text{unlab}(G)$ den zugehörigen unmarkierten Graphen (V, E) . Für jeden markierten Graphen G gilt also

$$\text{Cliquesweite}(\text{unlab}(G)) \leq \text{Cliquesweite}(G).$$

Die Graphen $\text{unlab}(G_1)$ und $\text{unlab}(G_2)$ haben Cliquesweite 2, da es zu $\text{unlab}(G_1)$ und $\text{unlab}(G_2)$ einen 2-Ausdruck X_1 bzw. X_2 mit zwei Markierungen gibt, jedoch keinen Aufbau mit einer Markierung. Der Graph $\text{unlab}(G_3)$ hat die Cliquesweite 3, da man den unmarkierten Weg mit fünf Knoten mit drei Markierungen wie mit X_3 konstruieren kann, jedoch nicht mit zwei Markierungen.

Zur Veranschaulichung der Baumstruktur der Graphen mit beschränkter Cliquesweite definieren wir den *Cliquesweite k -Ausdrucksbaum*. Der Cliquesweite k -Ausdrucksbaum $T = (V_T, E_T, \text{lab}_T)$ zu einem Cliquesweite k -Ausdruck X ist ein gerichteter Baum mit einer ausgezeichneten Wurzel. Die Knoten von T sind mit den Operationen des zugehörigen k -Ausdrucks markiert und die Kanten sind von den Söhnen zu den Vätern gerichtet.

Definition 2.3.2 (Cliquesweite k -Ausdrucksbaum) *Der Cliquesweite k -Ausdrucksbaum T zum k -Ausdruck \bullet_i besteht aus genau einem Knoten r (der Wurzel von T) der mit \bullet_i markiert wird. Der Cliquesweite k -Ausdrucksbaum T zu den k -Ausdrücken $\eta_{i,j}(X)$ und $\rho_{i \rightarrow j}(X)$ besteht aus dem Cliquesweite k -Ausdrucksbaum T' zum Ausdruck X , einem zusätzlichen Knoten r (der Wurzel von T) der mit $\eta_{i,j}$ bzw. mit $\rho_{i \rightarrow j}$ markiert wird und einer zusätzlichen gerichteten Kante von der Wurzel von T' zu Knoten r . Der Cliquesweite k -Ausdrucksbaum T zum k -Ausdruck $X_1 \oplus X_2$ besteht aus der disjunkten Vereinigung der Cliquesweite 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 gerichteten Kanten von den Wurzeln von T_1 und T_2 zu Knoten r . Die Wurzel von T_1 ist der linke Sohn von r und die Wurzel von T_2 ist der rechte Sohn von r .*

Ein Knoten aus T der mit \bullet_i , $\eta_{i,j}$, $\rho_{i \rightarrow j}$ bzw. \oplus markiert ist, wird Blatt, Kanteneinfügenknoten, Ummarkierungsknoten bzw. Vereinigungsknoten genannt.

Abbildungen 2.9, 2.10 und 2.11 zeigen drei Beispiele für Cliquesweite k -Ausdrücke und dazugehörige Cliquesweite k -Ausdrucksbäume.

Es gibt eine Bijektion zwischen den Blättern des Cliquesweite k -Ausdrucksbaumes T für einen Cliquesweite k -Ausdruck X und den Knoten im Graphen $\text{val}(X)$. Für einen Knoten u aus dem Ausdrucksbaum T sei $T(u)$ der Teilbaum von T der durch den Knoten u und alle Knoten v , für die es einen gerichteten Weg von v nach u in T gibt, induziert wird. Der Baum $T(u)$ ist ein gerichteter geordneter Baum mit Wurzel u und ist ebenfalls ein Cliquesweite k -Ausdrucksbaum. Der *Teilausdruck* $X(u)$, der durch den Baum $T(u)$ definiert wird, kann einfach durch einen Inorder Durchlauf von $T(u)$ (durchlaufe $T(u)$ beginnend an der Wurzel

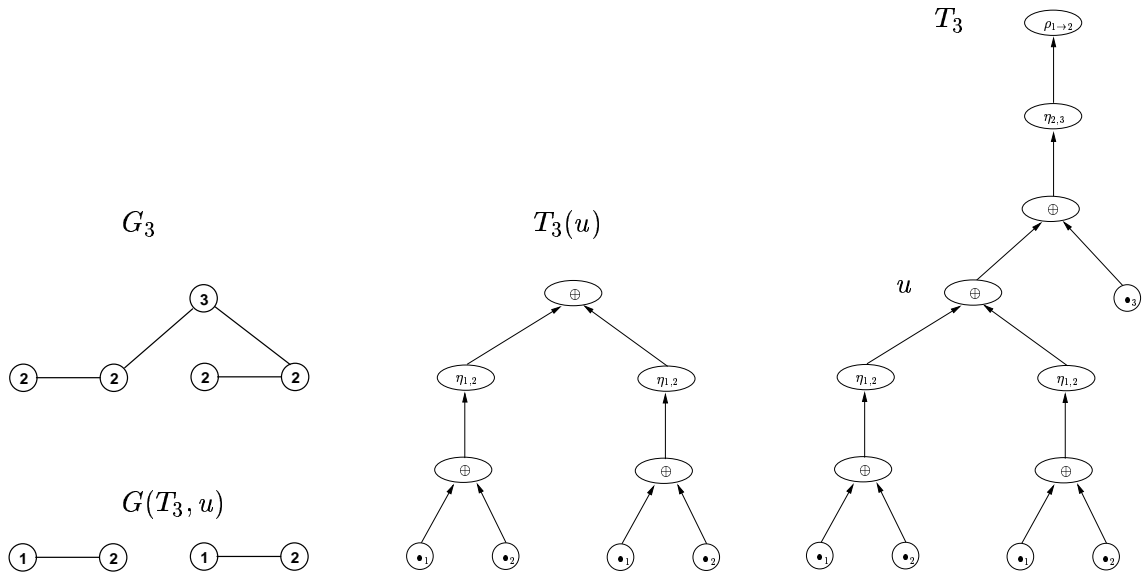


Abbildung 2.11: $X_3 = \rho_{1 \rightarrow 2}(\eta_{2,3}(((\eta_{1,2}(\bullet_1 \oplus \bullet_2)) \oplus (\eta_{1,2}(\bullet_1 \oplus \bullet_2))) \oplus \bullet_3))$, $G_3 = \text{val}(X_3)$ und der Cliquesweite 3-Ausdrucksbaum T_3 für X_3 . $X(u) = (\eta_{1,2}(\bullet_1 \oplus \bullet_2)) \oplus (\eta_{1,2}(\bullet_1 \oplus \bullet_2))$ ist ein Teilausdruck von X_3 , $T_3(u)$ ist der Teilbaum von T_3 mit Wurzel u und definiert den Graphen $G(T_3, u)$.

rekursiv, wobei zuerst der linke Teilbaum, dann die Wurzel und dann der rechte Teilbaum durchlaufen wird) erhalten werden. Die Knoten des Graphen G' seien die Knoten aus G die den Blättern aus $T(u)$ entsprechen. Die Kanten in G' und die Markierungen der Knoten in G' sind durch den Ausdruck $X(u)$ definiert. Für zwei Knoten u, v aus G' ist jede Kante $\{u, v\}$ aus G' auch in G enthalten, aber nicht umgekehrt. Zwei Knoten die in G' gleich markiert sind, sind auch in G gleichmarkiert, jedoch nicht umgekehrt. Der markierte Graph G' wird mit $G(T, u)$ oder kurz mit $G(u)$ bezeichnet, vergleiche Abbildung 2.11. Man kann zu jedem Cliquesweite k -Ausdruck einen Cliquesweite k -Ausdrucksbaum bestimmen und umgekehrt. Es ist jedoch bisher noch unbekannt wie schwer es ist, zu einem gegebenen Graphen einen Cliquesweite k -Ausdruck mit minimalem k zu finden.

2.4 NLC-Weite – NLC_k

Die Graphklasse NLC_k der Graphen mit *NLC-Weite* höchstens k ist von Wanke 1994 in [Wan94c] definiert. Die Abkürzung NLC steht für den *node label controlled* Einbettungsmechanismus, der ursprünglich für Graphgrammatiken definiert wurde.

Definition 2.4.1 (NLC-Weite, NLC_k [Wan94c]) 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 $G = (V_G, E_G, \text{lab}_G) \in NLC_k$ und $J = (V_J, E_J, \text{lab}_J) \in NLC_k$ zwei knotendisjunkte k -markierte Graphen und $S \subseteq [k]^2$ eine Relation, dann ist auch der k -markierte

Graph $G \times_S J := (V', E', lab')$ mit $V' := V_G \cup V_J$,

$$E' := E_G \cup E_J \cup \{\{u, v\} \mid u \in V_G, v \in V_J, (lab_G(u), lab_J(v)) \in S\},$$

und

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

in NLC_k .

3. Es sei $G = (V_G, E_G, lab_G) \in NLC_k$ ein k -markierter Graph und $R : [k] \rightarrow [k]$ eine totale Funktion, dann ist auch der k -markierte Graph $\circ_R(G) := (V_G, E_G, lab')$ mit $lab'(u) := R(lab(u)), \forall u \in V_G$ in NLC_k .

Einen Ausdruck X mit den Operationen $\bullet_i, \times_S, \circ_R$ mit $i \in [k], S \subseteq [k]^2, R : [k] \rightarrow [k]$, nennen wir NLC-Weite k -Ausdruck oder kurz k -Ausdruck. Der Graph der durch einen k -Ausdruck X definiert wird, wird mit $val(X)$ bezeichnet. Die Menge NLC_k besteht somit aus der Menge aller Graphen $val(X)$ für NLC-Weite k -Ausdrücke X .

Ein k -markierter Graph G hat NLC-Weite höchstens k , falls $G \in NLC_k$, d.h. NLC_k ist die Menge aller k -markierter Graphen mit NLC-Weite höchstens k . Die NLC-Weite eines markierten Graphen G (kurz NLC-Weite(G)) ist die kleinste natürliche Zahl k , so dass $G \in NLC_k$. Die Baumstruktur für G ist durch die Baumstruktur des Ausdrucks für G gegeben.

Die Operation \circ_R wird Ummarkierungsoperation und die Operation \times_S wird Vereinigungsoperation genannt.

Eine Graphklasse \mathcal{L} hat beschränkte NLC-Weite, falls es eine natürliche Zahl k gibt, so dass jeder Graph in \mathcal{L} NLC-Weite höchstens k hat, d.h. es existiert ein $k \in \mathbb{N}$ mit $\mathcal{L} \subseteq NLC_k$. Falls ein solches k existiert, so ist das minimale k die NLC-Weite von \mathcal{L} , sonst hat \mathcal{L} unbeschränkte NLC-Weite.

Der NLC-Weite 2-Ausdruck

$$X_1 = \bullet_1 \times_{\{(1,2)\}} \bullet_2$$

definiert den in Abbildung 2.12 dargestellten 2-markierten Graphen $G_1 = val(X_1)$, der 2-markierte Graph G_1 ist somit in der Graphklasse NLC_2 enthalten und hat NLC-Weite 2. Der NLC-Weite 2-Ausdruck

$$X_2 = (\circ_{\{(1,1),(2,1)\}}(\bullet_1 \times_{\{(1,2)\}} \bullet_2)) \times_{\{(1,2)\}} \bullet_2$$

definiert den 2-markierten Graphen G_2 in Abbildung 2.13, der 2-markierte Graph G_2 ist somit aus NLC_2 und hat NLC-Weite 2. Der NLC-Weite 3-Ausdruck

$$X_3 = ((\bullet_1 \times_{\{(1,2)\}} \bullet_2) \times_{\emptyset} (\bullet_1 \times_{\{(1,2)\}} \bullet_2)) \times_{\{(2,3)\}} \bullet_3$$

beschreibt den 3-markierten Graphen $G_3 \in NLC_3$ in Abbildung 2.14.

Ein unmarkierter Graph $G = (V, E)$ hat NLC-Weite höchstens k , falls es eine Markierung $lab : V \rightarrow [k]$ der Knoten von G gibt, so dass der markierte Graph $G' = (V, E, lab)$ NLC-Weite höchstens k hat. Die NLC-Weite eines unmarkierten Graphen $G = (V, E)$ (kurz NLC-Weite(G)) ist die kleinste natürliche Zahl k , so dass es eine Abbildung $lab : V \rightarrow [k]$ gibt, so dass (V, E, lab) NLC-Weite höchstens k hat.

Hat ein markierter Graph G NLC-Weite höchstens k und hat G zwei verschieden markierte Knoten die mit i bzw. j markiert sind, dann hat der Graph $\circ_{\{(i,j)\}}(G)$ ebenfalls NLC-Weite

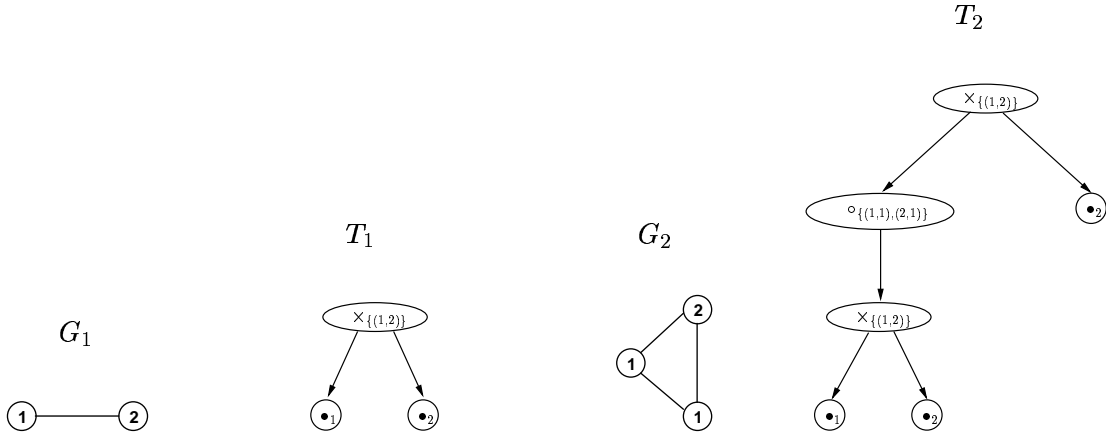


Abbildung 2.12: $X_1 = \bullet_1 \times_{\{(1,2)\}} \bullet_2$, $G_1 = \text{val}(X_1)$ und der NLC-Weite 2-Ausdrucksbaum T_1 für X_1

Abbildung 2.13: $X_2 = (\circ_{\{(1,1),(2,1)\}} (\bullet_1 \times_{\{(1,2)\}} \bullet_2)) \times_{\{(1,2)\}} \bullet_2$, $G_2 = \text{val}(X_2)$ und der NLC-Weite 2-Ausdrucksbaum T_2 für X_2

höchstens k . Da eine Ummarkierung die NLC-Weite eines Graphen nicht erhöht, können wir einen unmarkierten Graphen auch als Graphen betrachten dessen Knoten alle mit 1 markiert sind. Damit können wir den Begriff Graph auch im Zusammenhang mit der NLC-Weite ohne Konflikte sowohl für markierte und unmarkierte Graphen verwenden. Für jeden markierten Graphen G gilt also

$$\text{NLC-Weite}(\text{unlab}(G)) \leq \text{NLC-Weite}(G).$$

Die Graphen $\text{unlab}(G_1)$ und $\text{unlab}(G_2)$ in Abbildung 2.12 bzw. 2.13 haben NLC-Weite 1, da es zu $\text{unlab}(G_1)$ und $\text{unlab}(G_2)$ einen NLC-Weite 1-Ausdruck X'_1 bzw. X'_2 gibt.

$$X'_1 = \bullet_1 \times_{\{(1,1)\}} \bullet_1$$

$$X'_2 = (\bullet_1 \times_{\{(1,1)\}} \bullet_1) \times_{\{(1,1)\}} \bullet_1$$

Der Graph $\text{unlab}(G_3)$ in Abbildung 2.14 hat die NLC-Weite 2, da der folgende NLC-Weite 2-Ausdruck einen Weg mit fünf Knoten definiert.

$$X'_3 = ((\bullet_1 \times_{\{(1,2)\}} \bullet_2) \times_{\emptyset} (\bullet_1 \times_{\{(1,2)\}} \bullet_2)) \times_{\{(2,1)\}} \bullet_1$$

Zur Veranschaulichung der Baumstruktur von Graphen mit beschränkter NLC-Weite definieren wir den *NLC-Weite k -Ausdrucksbaum*. Der NLC-Weite k -Ausdrucksbaum $T = (V_T, E_T, \text{lab}_T)$ zu einem NLC-Weite k -Ausdruck ist ein gerichteter Baum mit einer ausgezeichneten Wurzel. Die Knoten von T sind mit den Operationen des k -Ausdrucks markiert und die Kanten sind von den Vätern zu den Söhnen gerichtet.

Definition 2.4.2 (NLC-Weite k -Ausdrucksbaum) Der NLC-Weite k -Ausdrucksbaum T zum k -Ausdruck \bullet_i besteht aus genau einem Knoten r (der Wurzel von T) der mit \bullet_i markiert wird. Der NLC-Weite k -Ausdrucksbaum T zum k -Ausdruck $\circ_R(G)$ besteht aus dem NLC-Weite k -Ausdrucksbaum T' von G , einem zusätzlichen Knoten r (der Wurzel von T) der mit \circ_R markiert wird und einer zusätzlichen gerichteten Kante von r zu der Wurzel von T' . Der NLC-Weite k -Ausdrucksbaum T zum k -Ausdruck $G \times_S J$ besteht aus der disjunkten

Vereinigung der NLC-Weite k -Ausdrucksbäume T_G und T_J von G bzw. J , einem zusätzlichen Knoten r (der Wurzel von T) der mit \times_S markiert wird und zwei zusätzlichen gerichteten Kanten von Knoten r zu den Wurzeln von T_G und T_J . Der linke Sohn von r ist die Wurzel von T_G und der rechte Sohn von r ist die Wurzel von T_J .

Ein Knoten aus T der mit \bullet_i , \circ_R bzw. \times_S markiert ist, wird Blatt, Ummarkierungsknoten bzw. Vereinigungsknoten genannt.

Abbildungen 2.12, 2.13 und 2.14 zeigen jeweils einen NLC-Weite k -Ausdrucksbaum und den zugehörigen Graphen. Die Richtungen der Kanten in einem NLC-Weite k -Ausdrucksbaum sind genau entgegengesetzt zu den Richtungen der Kanten eines Cliquenweite k -Ausdrucksbaumes definiert. Die Richtungen sind jeweils so gewählt um die Beweise in denen ein Ausdrucksbaum durchlaufen wird möglichst einfach zu gestalten. Die Richtung der Kanten im Cliquenweite k -Ausdrucksbaum verwenden wir in Kapitel 6. Die Richtungen der Kanten eines NLC-Weite k -Ausdrucksbaumes verwenden wir im Kapitel 3.2.

Wenn aus dem Zusammenhang klar wird, ob wir einen Cliquenweite k -Ausdrucksbaum oder einen NLC-Weite k -Ausdrucksbaum betrachten, schreiben wir auch einfacher k -Ausdrucksbaum. Wenn zusätzlich noch k aus dem Zusammenhang klar ist, schreiben wir auch kurz *Ausdrucksbaum*.

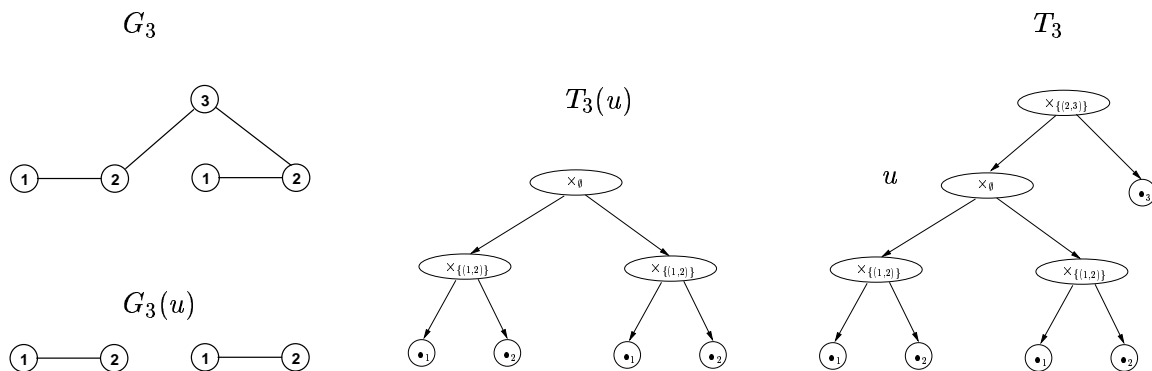


Abbildung 2.14: $X_3 = ((\bullet_1 \times_{\{(1,2)\}} \bullet_2) \times_\emptyset (\bullet_1 \times_{\{(1,2)\}} \bullet_2)) \times_{\{(2,3)\}} \bullet_3$, $G_3 = \text{val}(X_3)$ und der NLC-Weite 3-Ausdrucksbaum T_3 für X_3 . $X_3(u) = (\bullet_1 \times_{\{(1,2)\}} \bullet_2) \times_\emptyset (\bullet_1 \times_{\{(1,2)\}} \bullet_2)$ ist ein Teilausdruck von X_3 der durch den Teilbaum $T_3(u)$ von T_3 definiert wird. $G_3(u)$ ist der 3-markierte Graph der durch X_3 und $T_3(u)$ beschrieben wird.

Es gibt eine Bijektion zwischen den Knoten von G und den Blättern des NLC-Weite k -Ausdrucksbaumes T eines NLC-Weite k -Ausdrucks für G . Für einen Knoten u aus T sei $T(u)$ der Teilbaum von T der durch den Knoten u und alle Knoten v , für die es einen gerichteten Weg von u nach v in T gibt, induziert wird. Der Baum $T(u)$ ist ein gerichteter geordneter Baum mit Wurzel u und ist ebenfalls ein NLC-Weite k -Ausdrucksbaum. Der *Teilausdruck* $X(u)$ der durch den Baum $T(u)$ definiert wird, kann einfach durch einen Inorder Durchlauf von $T(u)$ erhalten werden. Der Ausdruck $X(u)$ beschreibt einen möglicherweise ummarkierten induzierten Teilgraphen $G(u)$ von G . Die Knoten von $G(u)$ sind die Knoten von G die den Blättern im Teilbaum $T(u)$ entsprechen. Die Kanten von $G(u)$ sind die Kanten von G für die beide Endknoten in $G(u)$ liegen. Die Markierungen der Knoten in $G(u)$ sind über den Ausdruck zu $T(u)$ definiert. Diese Markierungen sind nicht unbedingt die endgültigen

Markierungen der Knoten im Graphen G , da die Knoten aus $G(u)$ durch die Operationen der Knoten auf dem Weg von der Wurzel von T zum Vater von u ummarkiert werden können. In Abbildung 2.14 sind diese Notationen an einem Beispiel erläutert. Man kann aus jedem NLC-Weite k -Ausdruck einen NLC-Weite k -Ausdrucksbaum konstruieren und umgekehrt. Es ist bisher jedoch noch nicht effizient möglich aus einem gegebenen Graphen einen zugehörigen NLC-Weite k -Ausdruck mit minimalem k zu konstruieren.

Kapitel 3

Inklusionen zwischen Graphklassen

In diesem Kapitel werden vorhandene Inklusionen zwischen den in dem vorhergehenden Kapitel definierten Graphklassen beschrieben. Eine Graphklasse hat beschränkte Cliquesweite, genau dann wenn sie beschränkte NLC-Weite hat. Graphklassen mit beschränkter Baumweite haben auch beschränkte NLC-Weite und beschränkte Cliquesweite. Graphklassen mit beschränkter NLC-Weite haben jedoch nur unter zusätzlichen Voraussetzungen beschränkte Baumweite. Wir geben hier solche Bedingungen an und zeigen die entsprechenden Baumweiteschränken.

3.1 Inklusionen zwischen den Graphklassen NLC_k und CW_k

In [Gur98] und in [Joh98] wurde gezeigt, dass man zu jedem Cliquesweite k -Ausdruck einen äquivalenten (d.h. die markierten Graphen der beiden Ausdrücke sind isomorph) NLC-Weite k -Ausdruck angeben kann und zu jedem NLC-Weite k -Ausdruck einen äquivalenten Cliquesweite $2k$ -Ausdruck konstruieren kann. Damit gilt für jeden Graphen G

$$\text{NLC-Weite}(G) \leq \text{Cliquesweite}(G) \quad (3.1)$$

$$\text{Cliquesweite}(G) \leq 2 \cdot \text{NLC-Weite}(G) \quad (3.2)$$

und für die entsprechenden Graphklassen gilt für jede natürliche Zahl k

$$\text{CW}_k \subseteq \text{NLC}_k \quad (3.3)$$

$$\text{NLC}_k \subseteq \text{CW}_{2k}. \quad (3.4)$$

Eine Graphklasse hat somit beschränkte NLC-Weite, genau dann wenn sie beschränkte Cliquesweite hat. Für vollständig unzusammenhängende Graphen (Graphen mit NLC-Weite 1 und Cliquesweite 1) bzw. für vollständige Graphen (Graphen mit NLC-Weite 1 und Cliquesweite 2) werden die beiden Abschätzungen (3.1) und (3.2) angenommen. Aus (3.3) und (3.4) folgt eine Inklusionskette.

$$\text{CW}_1 \subseteq \text{NLC}_1 \subseteq \text{CW}_2 \subseteq \text{NLC}_2 \subseteq \text{CW}_4 \subseteq \text{NLC}_4 \subseteq \text{CW}_8 \subseteq \text{NLC}_8 \dots$$

Die Mengen NLC_1 und CW_2 definieren sogar genau die gleichen Mengen von Graphen, wenn man die Knotenmarkierungen der Graphen vernachlässigt, siehe Theorem 5.2.2. Es sind jedoch nicht alle Graphklassen CW_k , $\text{NLC}_{k'}$, $k, k' \geq 2$ bezüglich Inklusionen vergleichbar. Beispielsweise sind NLC_2 und CW_3 unvergleichbar. $\overline{C_6} \in \text{NLC}_2$ aber $\overline{C_6} \notin \text{CW}_3$ und $P_{10} \in \text{CW}_3$ aber $P_{10} \notin \text{NLC}_2$.

3.2 Inklusionen zwischen den Graphklassen NLC_k , CW_k und TW_k

In den Schriften [Wan94c, CO00, CR01, GM03] werden Abschätzungen für die Cliquesweite und NLC-Weite von Graphen mit beschränkter Baumweite angegeben. Die beste bekannte Abschätzung für die Cliquesweite von Graphen mit Baumweite k ist die Schranke von Corniel und Rotics in [CR01] welche zeigt, dass Graphen mit Baumweite k Cliquesweite höchstens $3 \cdot 2^{k-1}$ haben. Da die NLC-Weite eines Graphen stets kleiner oder gleich seiner Cliquesweite ist, beträgt auch die NLC-Weite von Graphen mit Baumweite k höchstens $3 \cdot 2^{k-1}$. Für einen beliebigen Graphen G gilt somit

$$\text{NLC-Weite}(G) \leq \text{Cliquesweite}(G) \leq 3 \cdot 2^{\text{Baumweite}(G)-1} \quad (3.5)$$

und für die entsprechenden Graphklassen gelten für jede natürliche Zahl k die folgenden Inklusionen.

$$\text{TW}_k \subseteq \text{CW}_{3 \cdot 2^{k-1}} \subseteq \text{NLC}_{3 \cdot 2^{k-1}} \quad (3.6)$$

Daraus folgt, dass jede Menge von Graphen mit beschränkter Baumweite auch beschränkte NLC-Weite und beschränkte Cliquesweite hat. Somit haben Graphklassen mit unbeschränkter NLC-Weite auch unbeschränkte Baumweite. Für Bäume mit Durchmesser mindestens 9 (Graphen mit Baumweite 1, Cliquesweite 3 und NLC-Weite 3) gilt in der Abschätzung (3.5) die Gleichheit.

Eine Menge von Graphen mit beschränkter NLC-Weite (bzw. beschränkter Cliquesweite) hat im Allgemeinen keine beschränkte Baumweite. Beispielsweise hat die Menge aller Cliques $\{K_n \mid n \geq 1\}$ NLC-Weite 1 (Cliquesweite 2), jedoch unbeschränkte Baumweite, da die Clique K_n die Baumweite $n - 1$ hat. Jeder Graph mit n Knoten und Baumweite k hat höchstens $kn - \frac{1}{2}k(k+1)$ Kanten, siehe zum Beispiel [Bod98], und somit eine in der Knotenanzahl lineare Kantenanzahl.

Unter der zusätzlichen Voraussetzung dass die Graphen keine beliebig großen vollständigen bipartiten Graphen $K_{n,n}$ als Teilgraphen enthalten, können wir bei beschränkter NLC-Weite die folgende Schranke für die Baumweite angeben.

Theorem 3.2.1 *Es sei G ein Graph mit NLC-Weite k und G enthalte den vollständigen bipartiten Graphen $K_{n,n}$, für eine natürliche Zahl $n > 1$, nicht als Teilgraph, dann hat G Baumweite höchstens $3k(n - 1) - 1$.*

Beweis Es sei $G = (V_G, E_G)$ ein Graph mit NLC-Weite k der den vollständig bipartiten Graphen $K_{n,n}$, für eine natürliche Zahl $n > 1$, nicht als Teilgraph enthält. Zu G gibt es einen NLC-Weite k -Ausdrucksbaum $T_G = (V_{T_G}, E_{T_G})$. Zwischen den Knoten aus G und den Blättern aus T_G gibt es eine Bijektion die wir in diesem Beweis mehrfach verwenden werden.

Wir definieren nun eine Baumdekomposition $(\mathcal{X} = \{X_u \mid u \in V_T\}, T = (V_T, E_T))$ mit Weite höchstens $3k(n - 1) - 1$ aus dem NLC-Weite k -Ausdrucksbaum T_G zu G .

Der Baum $T = (V_T, E_T)$ der Baumdekomposition wird mit Hilfe des NLC-Weite k -Ausdrucksbaums T_G für G wie folgt definiert.

1. Die Knotenmenge von T ist die Knotenmenge von T_G .

2. Es gibt eine Kante $\{u, v\}$ in E_T genau dann wenn es eine gerichtete Kante von u nach v in T_G gibt.

Zur Definition der Mengen X_u der Knoten u aus T führen wir einige Bezeichnungen ein. Für einen Knoten u aus T_G und eine Markierung $t \in [k]$, sei $A(u, t)$ die Menge aller Knoten aus $G(u)$ die mit t markiert sind. $B(u, t)$ sei die Menge aller Knoten aus G die nicht in $G(u)$ enthalten sind, aber in G zu einem Knoten aus $A(u, t)$ adjazent sind, siehe Abbildung 3.1. Alle Knoten aus $A(u, t)$ sind zu allen Knoten aus $B(u, t)$ adjazent, da gleich markierte Knoten aus $G(u)$ von allen Operationen der Knoten auf dem Weg von der Wurzel von T_G zum Vater von u gleichbehandelt werden. Daher hat für jedes u aus T_G und jedes $t \in [k]$ entweder die Menge $A(u, t)$, die Menge $B(u, t)$ oder beide Mengen weniger als n Knoten, sonst enthält G den $K_{n,n}$ als Teilgraphen, im Widerspruch zu unserer Annahme.

Nun definieren wir die Mengen $X_u \subseteq V_G$ für die Knoten u aus T in Abhängigkeit von der Markierung der Knoten u in T_G .

1. Falls der Knoten u mit \bullet_i , $i \in [k]$, markiert ist, d.h. u ist ein Blatt in T_G :

Dann wird X_u als die Menge definiert, die aus dem Knoten aus G besteht der zu dem Blatt u aus T_G gehört.

2. Falls der Knoten u mit \circ_R , $R: [k] \rightarrow [k]$, markiert ist:

Es sei v der Sohn von u in T_G . X_u wird als die Vereinigung aller Mengen $A(v, t)$ und $B(u, t)$, $t \in [k]$, die weniger als n Knoten enthalten, definiert.

3. Falls der Knoten u mit \times_S , $S \subseteq [k]^2$, markiert ist:

Es seien v_1, v_2 die beiden Söhne von u in T_G . Dann wird X_u als die Vereinigung der Mengen $A(v_1, t)$, $A(v_2, t)$ und $B(u, t)$, $t \in [k]$, die weniger als n Knoten enthalten, definiert.

Es bleibt zu zeigen, dass (\mathcal{X}, T) eine Baumdekomposition der Weite höchstens $3k(n-1) - 1$ für G ist.

1. $\cup_{u \in V_T} X_u = V_G$.

Da jeder Knoten aus G einem Blatt in T_G entspricht, sind alle Knoten aus G schon in der Vereinigung der Mengen X_u , für die Blätter u aus T , enthalten.

2. $|X_u| \leq 3k(n-1)$ für jedes $u \in V_T$.

Dies folgt unmittelbar, da X_u entweder nur aus einem Knoten besteht oder die Vereinigung von $2k$ bzw. $3k$ Mengen der Größe höchstens $n-1$ ist.

3. Für jede Kante $\{w_1, w_2\}$ aus E_G gibt es einen Knoten $u \in V_T$, so dass $w_1, w_2 \in X_u$.

Es sei $\{w_1, w_2\}$ eine Kante aus E_G und u ein Knoten aus T_G , so dass u zwei Söhne v_1 und v_2 hat, w_1 ein Knoten aus $G(v_1)$ und w_2 ein Knoten aus $G(v_2)$ ist. Die Verbindung zwischen w_1 und w_2 , d.h. die Kante $\{w_1, w_2\} \in E_G$, wird in der Operation \times_S in Knoten u eingefügt. Seien t_1 und t_2 die Markierungen von w_1 bzw. w_2 im Graphen $G(v_1)$ bzw. im Graphen $G(v_2)$. Falls v_1 der linke Sohn und v_2 der rechte Sohn von u ist, dann ist $(t_1, t_2) \in S$. Weiterhin gilt $w_1 \in A(v_1, t_1)$, $w_2 \in A(v_2, t_2)$, $w_2 \in B(v_1, t_1)$ und $w_1 \in B(v_2, t_2)$, aufgrund der Definition dieser Mengen.

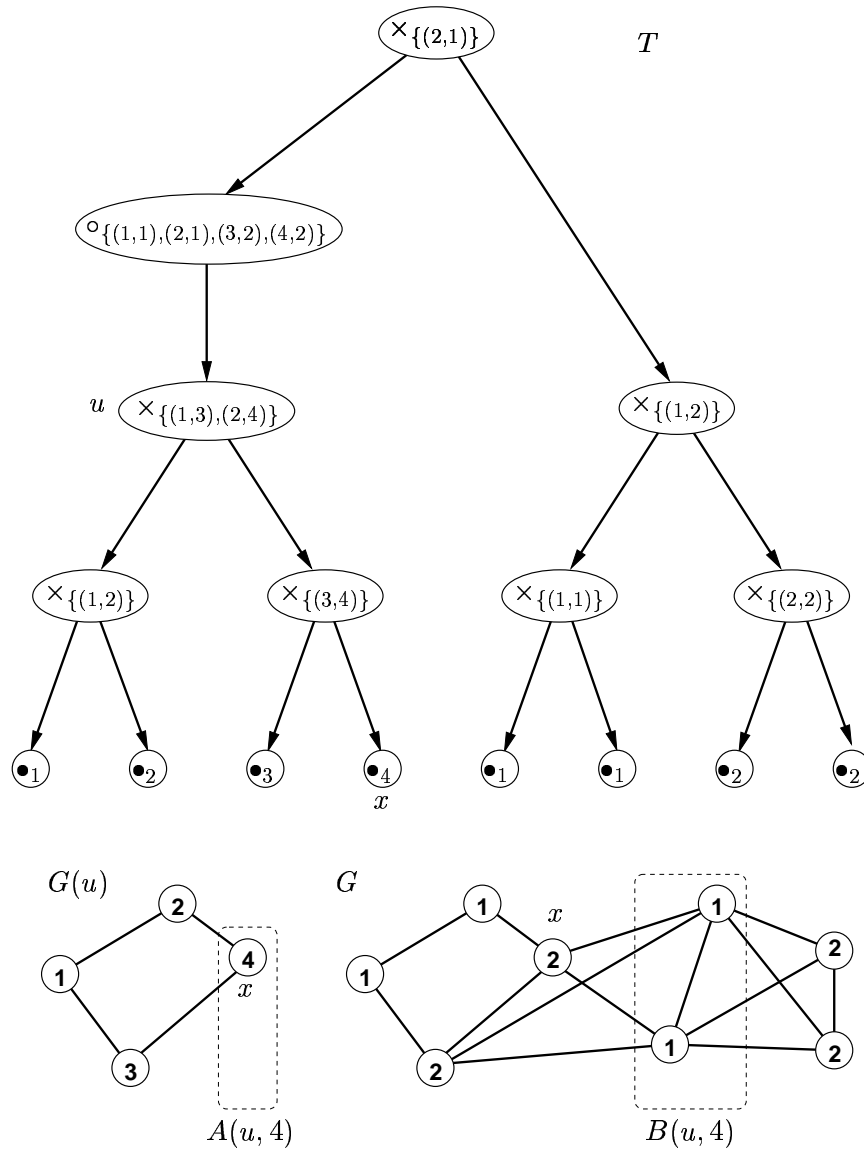


Abbildung 3.1: T ist ein NLC-Weite 2-Ausdrucksbaum für den Graphen G . Der Teilbaum $T(u)$ definiert den Graphen $G(u)$. $A(u, 4)$ besteht aus der Menge aller Knoten aus $G(u)$ die mit 4 markiert sind. $B(u, 4)$ ist die Menge aller Knoten aus G die nicht in $G(u)$ enthalten sind, aber in G zu einem Knoten aus $A(u, 4)$ adjazent sind.

- (a) Falls $|A(v_1, t_1)| < n$ und $|A(v_2, t_2)| < n$, dann sind w_1 und w_2 beide in X_u enthalten, da $A(v_1, t_1) \subseteq X_u$ und $A(v_2, t_2) \subseteq X_u$.
- (b) Falls $|A(v_1, t_1)| \geq n$ und $|A(v_2, t_2)| \geq n$, dann ist der $K_{n,n}$ ein Teilgraph von $G(u)$, im Widerspruch zu unserer Annahme.
- (c) Wir nehmen an, dass $|A(v_1, t_1)| < n$ und $|A(v_2, t_2)| \geq n$ gilt. Den Fall $|A(v_1, t_1)| \geq n$ und $|A(v_2, t_2)| < n$ beweist man analog. Falls $|A(v_2, t_2)| \geq n$, dann ist $|B(v_2, t_2)| < n$ und $B(v_2, t_2) \subseteq X_{v_2}$, somit ist $w_1 \in X_{v_2}$. Wir betrachten nun die Knoten im NLC-Weite k -Ausdrucksbaum T_G auf dem Weg vom Knoten v_2 zum Blatt das zum Knoten w_2 von G gehört. Sei r_1 dieses Blatt, r_{i+1} sei der

Vater von r_i für $i \geq 1$ und r_s sei der Knoten v_2 . Sei l_i die Markierung des Knotens w_2 im Graphen $G(r_i)$ für $i = 1, \dots, s$. Es gilt $A(r_i, l_i) \subseteq A(r_{i+1}, l_{i+1})$ und $B(r_{i+1}, l_{i+1}) \subseteq B(r_i, l_i)$ für $i = 1, \dots, s-1$. Es sei j , $1 \leq j \leq s$, der kleinste Index, so dass $|A(r_j, l_j)| \geq n$. Dann ist X_{r_j} die Menge die die beiden Knoten w_1 und w_2 enthält. Da $|A(r_{j-1}, l_{j-1})| < n$, wissen wir, dass $A(r_{j-1}, l_{j-1}) \subseteq X_{r_j}$ und damit ist $w_2 \in X_{r_j}$. Da $|A(r_j, l_j)| \geq n$, wissen wir dass $|B(r_j, l_j)| < n$, $B(r_j, l_j) \subseteq X_{r_j}$ und damit ist auch $w_1 \in X_{r_j}$.

4. Für jeden Knoten $w \in V_G$ ist der Teilgraph von T , der durch die Knoten u mit $w \in X_u$ induziert wird, zusammenhängend.

Es sei w ein Knoten aus G und r_1 sei das Blatt im NLC-Weite k -Ausdrucksbaum T_G , das w entspricht. Sei r_{i+1} der Vater von r_i , für $i = 1, \dots, h-1$, so dass r_h die Wurzel von T_G ist. Für alle diese Knoten r_i , $1 \leq i \leq h$, wissen wir, dass w ein Knoten aus $G(r_i)$ ist und somit ist $w \notin B(r_i, t)$ für jedes $t \in [k]$. Sei l_i die Markierung des Knoten w im Graphen $G(r_i)$, für $i = 1, \dots, h$. Falls $w \in X_{r_i}$, für ein $i > 1$, dann ist $w \in A(r_{i-1}, l_{i-1})$. Daher induzieren die Knoten r_i , mit $w \in X_{r_i}$, einen Weg in dem Dekompositionsbaum T . Wir nennen die Knoten r_1, \dots, r_h den *Wurzelpfad* des Knotens w in T_G . Man beachte, dass der Knoten w im Teilgraph $G(u)$, für einen Knoten u aus T_G , enthalten ist, genau dann wenn u im Wurzelpfad von w in T_G enthalten ist.

Es sei v ein Knoten aus $V_{T_G} - \{r_1, \dots, r_h\}$, so dass $w \in X_v$. Da v nicht zum Wurzelpfad gehört, muss es eine Markierung $t \in [k]$ geben, so dass $w \in B(v, t)$ und $|B(v, t)| < n$. Daher enthält der Teilgraph $G(v)$ mindestens einen Knoten w' der mit t markiert ist und der in G zu w adjazent ist. Nun zeigen wir, dass es immer einen Weg in T_G von einem Knoten r_i mit $|A(r_{i-1}, t_{i-1})| < n$ des Wurzelpfades des Knotens w in T_G zum Knoten v gibt. Daraus folgt, dass der Teilgraph von T , der durch die Knoten u mit $w \in X_u$ induziert wird, zusammenhängend ist.

- (a) Falls der Vater von v einer der Knoten r_i des Wurzelpfades, mit $|A(r_{i-1}, t_{i-1})| < n$ ist, dann ist die Behauptung gezeigt.
- (b) Falls der Vater von v einer der Knoten r_i des Wurzelpfades, mit $|A(r_{i-1}, t_{i-1})| \geq n$ ist, dann hat der Teilgraph $G(r_{i-1})$ höchstens n Knoten die mit t_{i-1} (die Markierung des Knoten w im Graphen $G(r_{i-1})$) markiert sind. Da gleichmarkierte Knoten gleichbehandelt werden, werden alle diese Knoten in G mit Knoten w' aus $G(v)$ verbunden. Somit gilt $A(r_{i-1}, t_{i-1}) \subseteq B(v, t)$ und damit $|B(v, t)| \geq |A(r_{i-1}, t_{i-1})| \geq n$, dies ist ein Widerspruch zu der Annahme dass $|B(v, t)| < n$.
- (c) Falls der Vater v' des Knoten v nicht zum Wurzelpfad $\{r_1, \dots, r_h\}$ gehört, dann sei t' die Markierung der Knoten aus $G(v')$ die in $G(v)$ mit t markiert sind. Dann ist $B(v', t') \subseteq B(v, t)$, $|B(v', t')| \leq |B(v, t)| < n$ und damit $B(v', t') \subseteq X_{v'}$. Da w in $B(v', t')$ enthalten sein muss, gilt $w \in X_{v'}$. Dieses Argument können wir bis hin zu den Knoten des Wurzelpfades in T_G wiederholen.

□

Die Schranke für die Baumweite aus Theorem 3.2.1 ist für $k = 1$ und $n = 2$ scharf. Der vollständige Graph mit drei Knoten K_3 hat NLC-Weite 1, enthält den $K_{2,2}$ nicht als Teilgraph und hat Baumweite $3k(n-1) - 1 = 2$.

Ein Graph $G = (V_G, E_G)$ heißt l -sparse, für $l \in \mathbb{N}$, falls $|E_G| \leq l \cdot |V_G|$. G heißt *uniformly l -sparse* falls jeder Teilgraph von G l -sparse ist. Eine Graphklasse \mathcal{G} ist *uniformly l -sparse*, falls alle Graphen aus \mathcal{G} uniformly l -sparse sind [Cou03]. Aus dem Theorem 3.2.1 folgen die nächsten beiden Korollare.

Korollar 3.2.2 *Es sei G ein Graph mit Cliquenweite (NLC-Weite) höchstens k .*

1. *Falls der vollständige bipartite Graph $K_{n,n}$, für eine natürliche Zahl $n > 1$, nicht Teilgraph von G ist, dann hat G Baumweite höchstens $3k(n-1) - 1$.*
2. *Falls G uniformly l -sparse ist, dann hat G Baumweite höchstens $6kl - 1$.*
3. *Falls es einen Graphen mit n Knoten gibt, der nicht Minor von G ist, dann hat G Baumweite höchstens $3k(n-1) - 1$.*
4. *Falls G planar¹ ist, dann hat G Baumweite höchstens $6k - 1$.*
5. *Falls jeder Knoten in G den Grad höchstens d hat, dann hat G Baumweite höchstens $3kd - 1$.*

Beweis

1. Aus Abschätzung (3.1) in Kapitel 3.1 ($\text{NLC-Weite}(G) \leq \text{Cliquenweite}(G)$) und Theorem 3.2.1 folgt die Behauptung.
2. Falls ein Graph G uniformly l -sparse ist, dann ist der vollständige bipartite Graph $K_{2l+1, 2l+1}$ nicht Teilgraph von G .
3. Falls es einen Graphen mit n Knoten gibt, der kein Minor von G ist, dann ist der vollständige Graph K_n kein Minor von G , damit ist der $K_{n,n}$ kein Minor von G und somit ist der $K_{n,n}$ nicht Teilgraph von G .
4. Planare Graphen enthalten den $K_{3,3}$ nicht als Teilgraphen.
5. Graphen in denen jeder Knoten den Grad höchstens d hat, enthalten den $K_{d+1, d+1}$ nicht als Teilgraphen.

□

Korollar 3.2.3 *Es sei \mathcal{L} eine Graphklasse mit Cliquenweite (NLC-Weite) höchstens k .*

1. *Wenn die Menge \mathcal{L}' aller bipartiter planarer Teilgraphen, aller Teilgraphen oder aller Minoren der Graphen in \mathcal{L} Cliquenweite höchstens l hat, dann hat \mathcal{L} Baumweite höchstens $3k(\lceil \frac{l^2}{2} \rceil - 1) - 1$.*
2. *Falls \mathcal{L} bezüglich bipartiten planaren Teilgraphen, Teilgraphen oder Minoren abgeschlossen ist, dann hat \mathcal{L} Baumweite höchstens $3k(\lceil \frac{k^2}{2} \rceil - 1) - 1$.*

Beweis

¹Ein Graph heißt *planar*, wenn er in die Ebene einbettbar ist, d.h. isomorph ist zu einem Graphen in dem sich zwei Kanten nur in einem Knoten treffen.

1. Das $l \times l$ -Gitter² ist bipartite, planar, hat die Cliquesweite $l+1$ (siehe [GR00]) und ist ein Teilgraph des $K_{\lfloor \frac{l^2}{2} \rfloor, \lfloor \frac{l^2}{2} \rfloor}$. Falls \mathcal{L} Cliquesweite l hat, so ist das $l \times l$ -Gitter kein Teilgraph eines Graphen aus \mathcal{L} und damit ist der $K_{\lfloor \frac{l^2}{2} \rfloor, \lfloor \frac{l^2}{2} \rfloor}$ nicht Teilgraph eines Graphen aus \mathcal{L} .
2. Folgt aus (1.).

□

Die Korollare 3.2.2 und 3.2.3 kann man nutzen um Mengen von Graphen mit beschränkter Baumweite zu charakterisieren. Es sei \mathcal{L} eine Graphklasse mit beschränkter Cliquesweite. Die Menge \mathcal{L} hat beschränkte Baumweite, genau dann wenn

1. jeder Graph aus \mathcal{L} den vollständig bipartiten Graphen $K_{n,n}$ für ein $n > 1$ nicht als Teilgraphen enthält,
2. \mathcal{L} uniformly l -sparse für ein $l \in \mathbb{N}$ ist,
3. es einen Graphen gibt, der nicht Minor jedes Graphen aus \mathcal{L} ist oder
4. die Menge aller bipartiten planaren Teilgraphen, Teilgraphen oder Minoren der Graphen aus \mathcal{L} beschränkte Cliquesweite hat.

Die Rückrichtungen folgen aus Korollar 3.2.2 (1.), (2.), (3.) und Korollar 3.2.3 (1.). Die Hinrichtungen folgen aus der folgenden Beobachtung. Wenn ein Graph G Baumweite höchstens k hat, dann (1.) enthält G den vollständig bipartiten Graphen $K_{k+1, k+1}$ nicht als Teilgraph, (2.) ist G uniformly k -sparse, (3.) ist der vollständige Graph K_{k+2} kein Minor von G und (4.) hat jeder bipartite planare Teilgraph, Teilgraph und Minor von G Baumweite höchstens k und damit Cliquesweite höchstens $3 \cdot 2^{k-1}$, siehe [CR01].

Die in Theorem 3.2.1 und Korollar 3.2.2 und 3.2.3 gezeigten Ergebnisse stammen aus einer Reihe von Ergebnissen die aus dem Bereich der Graphersetzungs-systeme bekannt sind [Cou97]. Eine Menge \mathcal{L} von Graphen, die durch ein Knotenersetzungs-system (NR-System) definiert ist, ist durch ein Hyperkantenersetzungs-system (HR-System) definierbar, genau dann wenn es eine natürliche Zahl $n > 1$ gibt, so dass der $K_{n,n}$ nicht Teilgraph eines Graphen aus \mathcal{L} ist [Cou97]. Dieses Ergebnis konnte in Theorem 3.2.1 auf die den Graphgrammatiken entsprechenden Graphklassen übertragen werden.

Es können jedoch nicht alle Ergebnisse bezüglich Hyperkantenersetzungs-systemen und Knotenersetzungs-systemen direkt auf Graphen mit beschränkter Baumweite und Graphen mit beschränkter NLC-Weite übertragen werden. Courcelle hat in [Cou97] gezeigt, dass eine Menge \mathcal{L} von Graphen die durch ein Knotenersetzungs-system definiert ist, durch ein Hyperkantenersetzungs-system definiert werden kann, genau dann wenn es eine natürliche Zahl k gibt, so dass für jeden Graphen aus $H \in \mathcal{L}$ gilt $|E_H| \leq k \cdot |V_H|$ (\mathcal{L} ist k -sparse). Es sei $G_n = K_n \cup \overline{K_{n^2}}$ der Graph der aus einer Clique mit n Knoten disjunkt vereinigt mit einem vollständig unzusammenhängendem Graphen mit n^2 Knoten besteht. Mann sieht leicht, dass die Graphklasse $\{G_n \mid n \in \mathbb{N}\}$ 1-sparse ist und NLC-Weite 1 hat, jedoch unbeschränkte Baumweite hat. Dieses Ergebnis kann man nur auf Graphen mit beschränkter NLC-Weite

²Ein $n \times m$ -Gitter ist ein Graph mit Knotenmenge $\{1, \dots, n\} \times \{1, \dots, m\}$ und Kantenmenge $\{(i, j), (i', j')\} \mid |i - i'| + |j - j'| = 1\}$.

übertragen, wenn man die Bedingung $|E_H| \leq k \cdot |V_H|$ für jeden Teilgraphen der Graphen aus \mathcal{L} voraussetzt (\mathcal{L} ist *uniformly k -sparse*, siehe Korollar 3.2.2).

Wanke hat in [Wan94c] auch eine eingeschränkte Version der NLC-Weite, die sogenannte *lineare NLC-Weite* definiert. Mindestens einer der an einer \times_S -Operation mit nicht leerem S beteiligten Graphen besteht hier aus genau einem Knoten. Die daraus resultierenden Graphklassen enthalten ebenfalls schon die Menge der Graphen mit beschränkter Baumweite, jedoch nicht die Menge der Cographen.

Kapitel 4

Modulare Dekomposition

In diesem Kapitel betrachten wir die modulare Dekomposition als eine weitere baumstrukturierte Graphzerlegung. Bei dieser Zerlegung werden Knoten zusammengefasst, die die gleiche Nachbarschaft im Restgraphen besitzen. In den resultierenden Quotientgraphen haben keine zwei Knoten die gleiche Nachbarschaft. Wir zeigen, dass ein Graph G genau dann NLC-Weite höchstens k hat, wenn alle Quotientgraphen von G NLC-Weite höchstens k haben.

4.1 Ersetzungsmechanismus in Graphen

Um die modulare Dekomposition auf Graphen mit beschränkter NLC-Weite anzuwenden, definieren wir einen Ersetzungsmechanismus in Graphen.

Definition 4.1.1 (Ersetzung in Graphen) *Es sei $G = (V, E)$ ein Graph mit n Knoten, $V = \{v_1, \dots, v_n\}$ und G_{v_1}, \dots, G_{v_n} seien n knotendisjunkte Graphen. Der Graph*

$$G[G_{v_1}, \dots, G_{v_n}]$$

ist der Graph den man erhält, indem man in die disjunkte Vereinigung der Graphen G_{v_1}, \dots, G_{v_n} für jede Kante $\{v_i, v_j\} \in E$ alle Kanten zwischen $V_{G_{v_i}}$ und $V_{G_{v_j}}$ einfügt.

Ist $G = (V, E, lab)$ ein markierter Graph und sind auch die Graphen G_{v_1}, \dots, G_{v_n} markiert, so entsprechen die Markierungen der Knoten in $G[G_{v_1}, \dots, G_{v_n}]$ den Markierungen der Knoten in den Graphen G_{v_1}, \dots, G_{v_n} .

Als Beispiel betrachten wir den unmarkierten Graphen $G = (V_G, E_G)$ mit $V_G = \{v_1, v_2, v_3, v_4, v_5\}$ in Abbildung 4.1, der auch als *house* bezeichnet wird. Weiterhin seien G_1

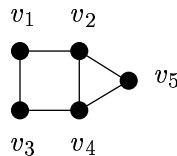
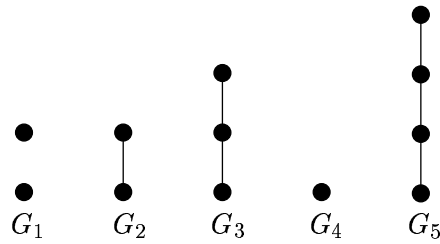


Abbildung 4.1: Der Graph $G = (V_G, E_G)$

der Graph der aus zwei nicht adjazenten Knoten besteht, G_2 sei der Graph der aus zwei

Abbildung 4.2: Die Graphen G_1, G_2, G_3, G_4, G_5

adjazenten Knoten besteht, G_3 sei der Weg der Länge zwei, G_4 bestehe aus genau einem Knoten und G_5 sei der Weg der Länge drei, siehe Abbildung 4.2. Für diese Graphen erhält man mittels $G[G_1, G_2, G_3, G_4, G_5]$ den in Abbildung 4.3 dargestellten Graphen H .

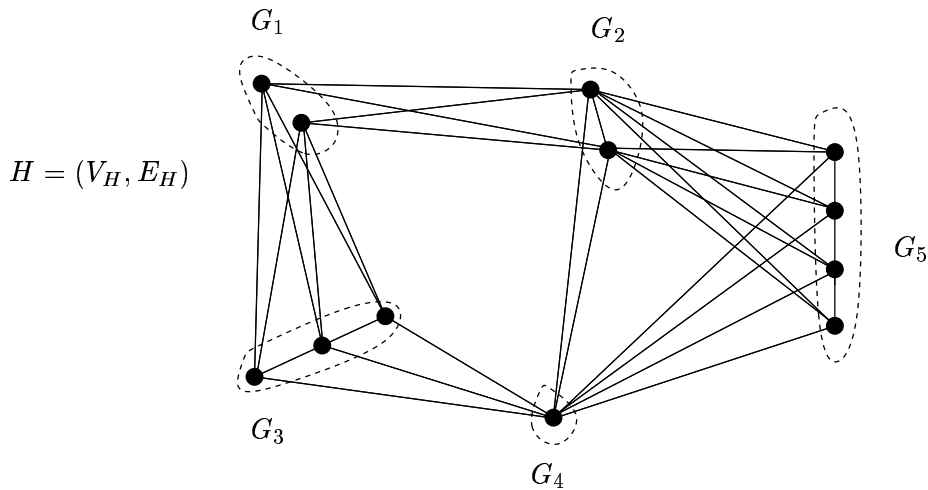


Abbildung 4.3: Der Graph H entsteht mittels der Graphersetzung $H = G[G_1, G_2, G_3, G_4, G_5]$ aus dem Graphen G aus Abbildung 4.1 und den Graphen G_1, G_2, G_3, G_4, G_5 aus Abbildung 4.2. Die Knotenmengen der Teilgraphen G_1, G_2, G_3, G_4, G_5 bilden (starke) Module in H .

Das folgende Theorem liefert einen Zusammenhang zwischen der NLC-Weite (Cliquesweite) der eingesetzten Graphen und der NLC-Weite (Cliquesweite) des Ergebnisgraphen bei der Graphersetzung.

Theorem 4.1.2 *Es sei G ein Graph mit n Knoten und G_{v_1}, \dots, G_{v_n} seien n knotendisjunkte Graphen, dann gilt*

$$NLC\text{-Weite}(G[G_{v_1}, \dots, G_{v_n}]) = \max \left\{ \begin{array}{l} NLC\text{-Weite}(G) \\ NLC\text{-Weite}(G_{v_1}) \\ \vdots \\ NLC\text{-Weite}(G_{v_n}) \end{array} \right\}$$

und

$$\text{Cliquesweite}(G[G_{v_1}, \dots, G_{v_n}]) = \max \left\{ \begin{array}{l} \text{Cliquesweite}(G) \\ \text{Cliquesweite}(G_{v_1}) \\ \vdots \\ \text{Cliquesweite}(G_{v_n}) \end{array} \right\}.$$

Weiterhin kann man einen NLC-Weite (bzw. Cliquesweite) k -Ausdruck für jeden der Graphen

$$G, G_{v_1}, \dots, G_{v_n}$$

aus einem NLC-Weite (bzw. Cliquesweite) k -Ausdruck für

$$G[G_{v_1}, \dots, G_{v_n}]$$

bestimmen und umgekehrt.

Beweis Es sei G ein Graph mit n Knoten v_1, \dots, v_n ; G_{v_1}, \dots, G_{v_n} seien n knotendisjunkte Graphen und

$$k := \max \{ \text{NLC-Weite}(G), \text{NLC-Weite}(G_{v_1}), \dots, \text{NLC-Weite}(G_{v_n}) \}.$$

Dann gibt es zu G einen NLC-Weite k -Ausdrucksbaum T . Ebenso existieren zu G_{v_1}, \dots, G_{v_n} jeweils ein NLC-Weite k -Ausdrucksbaum T_{v_1}, \dots, T_{v_n} . Die n Blätter in T entsprechen den Knoten v_1, \dots, v_n aus G . Ersetzt man in T das mit \bullet_{l_i} markierte Blatt zu Knoten v_i für $i = 1, \dots, n$ durch einen Knoten v'_i , markiert diesen Knoten mit \circ_R , $R = \{(1, l_i), \dots, (k, l_i)\}$ und macht die Wurzel von T_{v_i} zum Sohn von v'_i , so erhält man einen NLC-Weite k -Ausdrucksbaum für $G[G_{v_1}, \dots, G_{v_n}]$. Damit gilt $\text{NLC-Weite}(G[G_{v_1}, \dots, G_{v_n}]) \leq k$.

Wir nehmen an, dass $\text{NLC-Weite}(G[G_{v_1}, \dots, G_{v_n}]) = k_1 < k$, dann gibt es einen NLC-Weite k_1 -Ausdrucksbaum T für $G[G_{v_1}, \dots, G_{v_n}]$. Aus T kann man wie folgt einen NLC-Weite k_1 -Ausdrucksbaum T_{v_i} für jeden der Graphen G_{v_i} , $i = 1, \dots, n$ gewinnen. T_{v_i} ist der induzierte Teilbaum von T der die Blätter von T , die den Knoten in G_{v_i} entsprechen und alle Knoten aus T auf den Wegen von diesen Blättern zur Wurzel von T enthält. Vereinigungsknoten die nach dieser Auswahl nur noch einen Sohn besitzen, werden durch diesen Sohn ersetzt.

Weiterhin kann man aus T einen NLC-Weite k_1 -Ausdrucksbaum T' für G konstruieren, indem man aus T den Teilbaum auswählt, der als Blätter zu jedem der Graphen G_{v_1}, \dots, G_{v_n} genau ein Blatt und alle Wegen dieser n Blätter zur Wurzel von T' enthält, dabei werden Vereinigungsknoten mit einem Sohn durch diesen Sohn ersetzt. Somit kann man sowohl die Graphen G_{v_1}, \dots, G_{v_n} als auch den Graphen G mit k_1 Markierungen konstruieren. Dies ist ein Widerspruch zur Wahl von k .

Die Aussagen zur Cliquesweite beweist man analog, siehe [CMR00]. \square

4.2 Module

Ein *Modul* M in einem Graphen G ist eine Teilmenge der Knotenmenge von G , die alle die gleichen Nachbarn außerhalb von M besitzen. Triviale Module sind Knotenmengen ohne, mit genau einem oder allen Knoten.

Definition 4.2.1 ((triviales, starkes) Modul, primer Graph) Es sei $G = (V, E)$ ein Graph. Eine Knotenmenge $M \subseteq V$ heißt Modul, falls

$$\forall u_1, u_2 \in M, \quad \forall v \in V - M, \quad \{u_1, v\} \in E \Leftrightarrow \{u_2, v\} \in E.$$

Ein Modul M heißt trivial, falls $M = \emptyset$, $|M| = 1$ oder $M = V$. Ein Modul M heißt stark, falls für jedes Modul M_1 entweder $M \cap M_1 = \emptyset$ gilt oder ein Modul das andere enthält, d.h. $M \subseteq M_1$ oder $M_1 \subseteq M$. Ein Graph G heißt prim, falls jedes Modul in G trivial ist.

Für markierte Graphen ist der Begriff Modul analog definiert. In Abbildung 4.3 bilden die eingekreisten Knotenmengen Module in H . Diese Module sind alle stark, da sie disjunkt sind. Das Modul das aus der Knotenmenge V_{G_4} besteht ist ein triviales Modul. Die Knotenmengen V_{G_1} , V_{G_2} , V_{G_3} und V_{G_5} sind nicht triviale Module in H . Der Graph G in Abbildung 4.1 ist prim. Weitere prime Graphen sind der P_2 und der P_4 . Cographen mit mehr als zwei Knoten sind nach Definition 5.2.1 nicht prim. Mit Hilfe von Modulen kann man jeden Graphen eindeutig zerlegen. Die Grundlage dafür liefert das folgende Theorem.

Theorem 4.2.2 ([Gal67, Hab71, HM79, Sum71]) Es sei G ein Graph mit mindestens zwei Knoten, dann gilt genau eine der folgenden drei Bedingungen.

1. G ist unzusammenhängend und kann in seine Zusammenhangskomponenten zerlegt werden (parallele Dekomposition)
2. \bar{G} ist unzusammenhängend und G kann in seine Zusammenhangskomponenten von \bar{G} zerlegt werden (serielle Dekomposition) oder
3. G und \bar{G} sind zusammenhängend. Es gibt eine Menge $Y \subseteq V$ und eine eindeutige Zerlegung P von V , so dass
 - (a) $|Y| > 3$,
 - (b) $G(Y)$ ist ein maximaler primer Teilgraph von G und
 - (c) für jedes $S \in P$ ist S ein Modul und $|S \cap Y| = 1$.

Zerlegt man einen Graphen nach Theorem 4.2.2, so kann man diese Zerlegung in einer Baumstruktur darstellen. Der *modulare Dekompositionsbaum* [GV97, BM83, CH94] eines Graphen G ist ein Baum, den wir mit $MD(G)$ bezeichnen. Die Blätter von $MD(G)$ sind genau die Knoten von G . Die Menge der Blätter im Teilbaum zu einem inneren Knoten von $MD(G)$ bilden ein starkes Modul von G . Der Dekompositionsbaum hat entsprechend Theorem 4.2.2 drei Typen von inneren Knoten: parallele (Bedingung 1), serielle (Bedingung 2) und prime Knoten (Bedingung 3). Die inneren Knoten in $MD(G)$ werden entsprechend mit P , S bzw. N markiert. Die Menge der Blätter eines primen Knoten entsprechen einem maximalen (nicht vergrößerbaren) primen Teilgraphen in G . Der Baum $MD(G)$ zu einem Graphen G ist bis auf Isomorphie eindeutig. Abbildung 4.4 zeigt einen modularen Dekompositionsbaum $MD(H)$ für den Graphen H aus Abbildung 4.3. Die *modulare Dekomposition* eines Graphen G besteht aus dem Baum $MD(G)$ und einer Abbildung die jedem inneren Knoten h in $MD(G)$ den Quotientgraphen $G(h)$ zuordnet.

Enthält der Dekompositionsbaum $MD(G)$ der modularen Dekomposition eines Graphen G nur parallele und serielle innere Knoten, so ist G ein *Cograph*. Der Baum $MD(G)$ wird in diesem Fall auch *Cobaum* für G genannt. Cographen werden in den Schriften [Ler71, CLSB81,

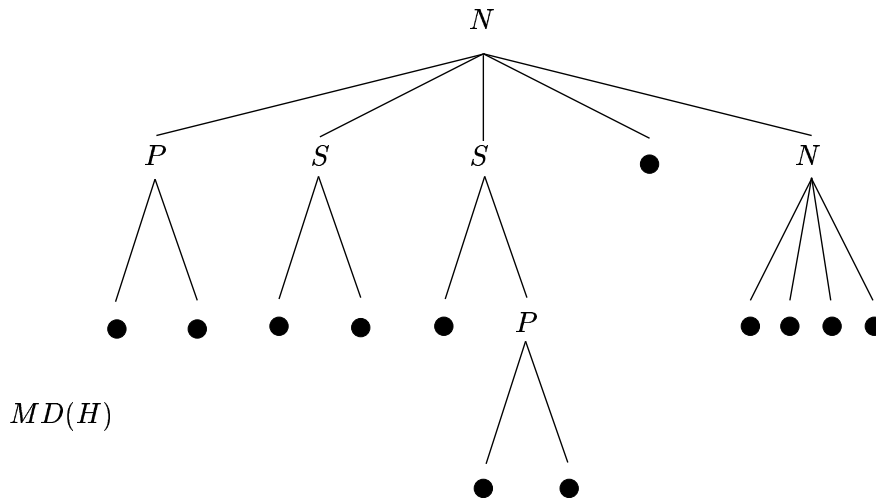


Abbildung 4.4: Ein modularer Dekompositionsbaum $MD(H)$ für den Graph H aus Abbildung 4.3

CPS85, BLS99] untersucht. Es gibt einen Linearzeit Erkennungsalgorithmus für Cographen von Corneil, Perl und Stewart in [CPS85] der von McConnell und Spinrad in [MS94] und von Cournier und Habib in [CH94] zu einem Linearzeitalgorithmus zur Bestimmung der modularen Dekomposition erweitert wurde.

Theorem 4.2.3 ([MS94, CH94]) *Die modulare Dekomposition eines Graphen kann in Linearzeit berechnet werden.*

4.3 Quotientgraphen

Hier zeigen wir, dass zur Bestimmung der NLC-Weite eines Graphen nur die NLC-Weiten der wie folgt definierten *Quotientgraphen* in der modularen Dekomposition des Graphen zu betrachten sind.

Definition 4.3.1 (Quotientgraph) *Es sei h ein innerer Knoten des modularen Dekompositionsbaumes $MD(G)$ eines Graphen G . Mit $M(h)$ sei das zu h gehörige Modul bezeichnet, das aus der Menge der Knoten von G besteht, die in den Blättern des Teilbaumes von $MD(G)$ mit Wurzel h vorkommen. Es sei $\{h_1, \dots, h_r\}$ die Menge der Söhne von h in $MD(G)$. Der Graph $G(h) = (V(h), E(h))$ wird als Quotientgraph des Moduls $M(h)$ bezeichnet. $V(h) = \{h_1, \dots, h_r\}$ und $E(h) = \{(h_i, h_j) \mid \exists u, v (u \in M(h_i) \wedge v \in M(h_j) \wedge (u, v) \in E)\}$.*

Nach der Definition eines Moduls gilt die folgende Beobachtung. Ist ein Knoten aus $M(h_i)$ zu einem Knoten aus $M(h_j)$ adjazent, so ist jeder Knoten aus $M(h_i)$ zu jedem Knoten aus $M(h_j)$ adjazent. Aus der Konstruktion von $MD(G)$ folgt unmittelbar das folgende Lemma.

Lemma 4.3.2 *Es sei G ein Graph, h sei ein innerer Knoten des modularen Dekompositionsbaumes $MD(G)$ und $L(h)$ sei die Markierung von h . $G(h)$ sei der Quotientgraph des Moduls $M(h)$.*

- Falls $L(h) = S$, so ist der Graph $G(h)$ vollständig.
- Falls $L(h) = P$, so ist der Graph $G(h)$ vollständig unzusammenhängend.
- Falls $L(h) = N$, so ist der Graph $G(h)$ prim.

Das folgende Theorem zeigt, dass man die NLC-Weite eines Graphen mit Hilfe der in der modularen Dekomposition des Graphen auftretenden Quotientgraphen bestimmen kann.

Theorem 4.3.3 Für jeden Graphen G gilt

$$\text{NLC-Weite}(G) = \max \{ \text{NLC-Weite}(H) : H \text{ ist primier Quotientgraph in } MD(G) \}.$$

Beweis Zu einem Graphen G wird mit Hilfe von $MD(G)$ ein Aufbau für G mit der Graphersetzung aus Definition 4.1.1 konstruiert. Sei r die Wurzel von $MD(G)$. $MD(G)$ wird bezüglich r in Sphären S_i , $i = 0, \dots, h$ eingeteilt. Die Sphäre S_i besteht aus genau den inneren Knoten aus $MD(G)$ die die Distanz i zu r in $MD(G)$ haben. S_0 besteht also nur aus r . Initial sei G der Graph der aus genau einem Knoten besteht. Nun wird G für jede Sphäre S_i , $i = 0, 1, \dots, h$ wie folgt ersetzt. Die Knoten in einer Sphäre S_i seien h_1, \dots, h_{n_i} , dann wird G durch $G[G(h_1), \dots, G(h_{n_i})]$ ersetzt. Nach dem Durchlaufen aller Sphären erhalten wir den Graphen G den wir mittels der Graphersetzung aufbauen wollen.

Da die Quotientgraphen zu inneren Knoten der modularen Dekomposition isomorph zu einem vollständigen Graphen (NLC-Weite 1), vollständig unzusammenhängenden Graphen (NLC-Weite 1) oder zu einem primen Graphen sind und alle in der angegebenen Konstruktion ersetzten Graphen Quotientgraphen zu inneren Knoten aus $MD(G)$ sind, folgt die Behauptung aus Theorem 4.1.2. \square

Für den Graphen H aus Abbildung 4.3 gilt nach Theorem 4.3.3

$$\text{NLC-Weite}(H) = \max \{ \text{NLC-Weite}(G_5), \text{NLC-Weite}(\text{house}) \} = \max \{ 2, 2 \} = 2$$

In Kapitel 5.2.2 wird die NLC-Weite von Graphklassen mit wenigen P_4 mit Hilfe der modularen Dekomposition bestimmt. Courcelle, Makowsky und Rotics haben die Aussage aus Theorem 4.3.3 in [CMR00] für die Cliquesweite gezeigt. Somit kann man sich bei der Analyse von Graphen bezüglich der NLC-Weite und der Cliquesweite auf prime Graphen beschränken.

Kapitel 5

NLC-Weite und Cliquesweite von Graphen und Graphklassen

In diesem Kapitel bestimmen wir die NLC-Weiten und Cliquesweiten einiger spezieller Graphen und Graphklassen. Weiterhin geben wir Abschätzungen für die NLC-Weite und die Cliquesweite beliebiger Graphen an und untersuchen wie sich die NLC-Weite (Cliquesweite) eines Graphen beim Anwenden von Graphoperationen verändert. Die untersuchten Graphen seien alle unmarkiert, bzw. alle Knoten seien mit 1 markiert.

5.1 NLC-Weite und Cliquesweite von speziellen Graphen

Wir geben in diesem Abschnitt die NLC-Weite und Cliquesweite von Wegen und Kreisen und deren Komplementen an.

NLC-Weite von Graphen

In der Tabelle 5.2 geben wir in der zweiten Spalte die *NLC-Weite von Wegen* P_n , $n = 1, \dots, 9$ und in der dritten Spalte einen NLC-Weite k -Ausdruck X_n für P_n an. In Tabelle 5.1 sind die Graphen P_4 und P_5 und ihre Komplemente dargestellt. Die Graphen P_4 , P_5 , P_6 , P_7 , P_8 und P_9 haben NLC-Weite > 1 , da es keine Cographen sind, siehe Theorem 5.2.2. Ein NLC-Weite 2-Ausdruck für diese Graphen ist in Tabelle 5.2 angegeben. Den Weg P_{10} kann man nicht mit zwei Markierungen konstruieren. Der P_{10} hat NLC-Weite 3 und da Graphen mit beschränkter NLC-Weite abgeschlossen bezüglich induzierten Teilgraphen sind (Kapitel 5.4), hat auch jeder Weg P_n mit $n \geq 10$ NLC-Weite 3 und kann mit dem folgenden NLC-Weite 3-Ausdruck X_n aufgebaut werden.

$$X_3 = (\bullet_1 \times_{\{(1,2)\}} \bullet_2) \times_{\{(2,3)\}} \bullet_3$$

$$X_n = (\circ_{\{(1,1),(2,1),(3,2)\}} (X_{n-1})) \times_{\{(2,3)\}} \bullet_3, \quad n \geq 4$$

Der Komplementgraph jedes Weges P_n hat aufgrund des Komplementabschlusses der Graphklassen NLC_k (Kapitel 5.4) die gleiche NLC-Weite wie der Weg P_n .

Mit Hilfe der NLC-Weiten der Wege kann man Aussagen über den Durchmesser von Graphen mit NLC-Weite höchstens k machen. Graphen mit NLC-Weite 1 haben alle einen

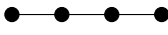
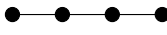

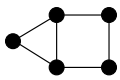


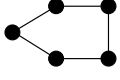
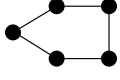
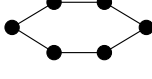
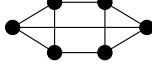
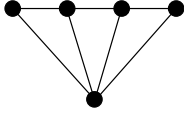
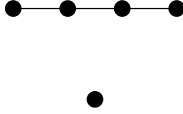
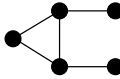
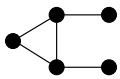
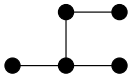
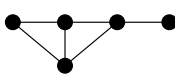
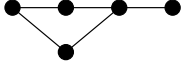
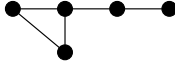


P_4 (2,3) 	$\overline{P_4} \approx P_4$ (2,3) 
P_5 (2,3) 	$\overline{P_5} \approx \text{house}$ (2,3) 
C_4 (1,2) 	$\overline{C_4}$ (1,2) 
C_5 (2,3) 	$\overline{C_5} \approx C_5$ (2,3) 
C_6 (2,3) 	$\overline{C_6}$ (2,4) 
gem (2,3) 	co-gem (2,3) 
bull (2,3) 	bull \approx bull (2,3) 
chair (2,3) 	co-chair (2,3) 
P (2,3) 	co-P (2,3) 
diamond (1,2) 	co-diamond (1,2) 

Tabelle 5.1: Spezielle Graphen und deren Komplemente. Die NLC-Weite und die Cliquenweite der Graphen sind jeweils in Klammern angegeben (NLC-Weite(G), Cliquenweite(G)).

Durchmesser höchstens 2 und Graphen mit NLC-Weite höchstens 2 haben alle einen Durchmesser höchstens 8. Graphen mit NLC-Weite höchstens k , für $k \geq 3$ haben beliebig große Durchmesser.

P_1	1	$X_1 = \bullet_1$
P_2	1	$X_2 = \bullet_1 \times_{\{(1,1)\}} \bullet_1$
P_3	1	$X_3 = (\bullet_1 \times_{\emptyset} \bullet_1) \times_{\{(1,1)\}} \bullet_1$
P_4	2	$X_4 = ((\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\emptyset} \bullet_1) \times_{\{(1,1)\}} \bullet_1$
P_5	2	$X_5 = X_4 \times_{\{(2,2)\}} \bullet_2$
P_6	2	$X_6 = (\bullet_1 \times_{\{(1,2)\}} \bullet_2) \times_{\{(2,2)\}} X_4$
P_7	2	$X_7 = (\bullet_1 \times_{\{(1,1)\}} (\bullet_1 \times_{\{(1,2)\}} \bullet_2)) \times_{\{(2,2)\}} X_4$
P_8	2	$X_8 = X_4 \times_{\{(2,2)\}} X_4$
P_9	2	$X_9 = (X_4 \times_{\emptyset} X_4) \times_{\{(2,2)\}} \bullet_2$

 Tabelle 5.2: NLC-Weite von Wegen P_n und ein k -Ausdruck X_n für P_n , $n = 1, \dots, 9$.

Tabelle 5.3 zeigt die *NLC-Weite von Kreisen* C_n , $n = 1, \dots, 12$ und einen NLC-Weite k -Ausdruck X_n . Tabelle 5.1 zeigt die Kreise C_4 , C_5 und C_6 und ihre Komplemente. Die Kreise C_n mit $n \leq 4$ haben NLC-Weite 1. C_5 und C_6 haben NLC-Weite 2. C_7 , C_8 , C_9 , C_{10} , C_{11} und C_{12} haben NLC-Weite 3. Kreise C_n mit $n \geq 13$ haben NLC-Weite 4 und können mit dem folgenden NLC-Weite 4-Ausdruck X_n aufgebaut werden.

$$P_{n-1} = (\circ_{\{(1,1),(2,2),(3,2),(4,3)\}} (P_{n-2})) \times_{\{(3,4)\}} \bullet_4, \quad n \geq 6$$

$$P_4 = ((\bullet_1 \times_{\{(1,2)\}} \bullet_2) \times_{\{(2,3)\}} \bullet_3) \times_{\{(3,4)\}} \bullet_4$$

$$X_n = P_{n-1} \times_{\{(1,2),(4,2)\}} \bullet_2$$

C_1	1	$X_1 = \bullet_1$
C_2	1	$X_2 = \bullet_1 \times_{\{(1,1)\}} \bullet_1$
C_3	1	$X_3 = (\bullet_1 \times_{\{(1,1)\}} \bullet_1) \times_{\{(1,1)\}} \bullet_1$
C_4	1	$X_4 = (\bullet_1 \times_{\emptyset} \bullet_1) \times_{\{(1,1)\}} (\bullet_1 \times_{\emptyset} \bullet_1)$
C_5	2	$X_5 = ((\bullet_1 \times_{\{(1,2)\}} \bullet_2) \times_{\{(1,1)\}} (\bullet_1 \times_{\{(1,2)\}} \bullet_2)) \times_{\{(2,1)\}} \bullet_1$
C_6	2	$X_6 = (((\bullet_1 \times_{\{(1,2)\}} \bullet_2) \times_{\emptyset} (\bullet_1 \times_{\{(1,2)\}} \bullet_2)) \times_{\{(1,1)\}} \bullet_1) \times_{\{(2,2)\}} \bullet_2$
C_7	3	$X_7 = ((\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\{(1,1)\}} (\bullet_1 \times_{\{(1,3)\}} \bullet_3)) \times_{\{(2,2),(3,3)\}} ((\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\{(1,3)\}} \bullet_3)$
C_8	3	$X_8 = (\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\{(1,1)\}} (\bullet_1 \times_{\{(1,3)\}} \bullet_3) \times_{\{(2,2),(3,3)\}} ((\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\{(1,1)\}} (\bullet_1 \times_{\{(1,3)\}} \bullet_3))$
C_9	3	$X_9 = ((\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\{(1,1)\}} (\bullet_1 \times_{\{(1,3)\}} \bullet_3)) \times_{\{(2,2),(3,3)\}} (((\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\emptyset} (\bullet_1 \times_{\{(1,3)\}} \bullet_3)) \times_{\{(1,1)\}} \bullet_1)$
C_{10}	3	$X_{10} = (((\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\emptyset} (\bullet_1 \times_{\{(1,3)\}} \bullet_3)) \times_{\{(1,1)\}} \bullet_1) \times_{\{(2,2),(3,3)\}} (((\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\emptyset} (\bullet_1 \times_{\{(1,3)\}} \bullet_3)) \times_{\{(1,1)\}} \bullet_1)$
C_{11}	3	$X_{11} = (((((\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\{(1,1)\}} (\bullet_1 \times_{\{(1,3)\}} \bullet_3)) \times_{\emptyset} (((\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\emptyset} (\bullet_1 \times_{\{(1,3)\}} \bullet_3)) \times_{\{(1,1)\}} \bullet_1)) \times_{\{(2,2)\}} \bullet_2) \times_{\{(3,3)\}} \bullet_3$
C_{12}	3	$X_{12} = ((((((\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\emptyset} (\bullet_1 \times_{\{(1,3)\}} \bullet_3)) \times_{\{(1,1)\}} \bullet_1) \times_{\emptyset} (((\bullet_2 \times_{\{(2,1)\}} \bullet_1) \times_{\emptyset} (\bullet_1 \times_{\{(1,3)\}} \bullet_3)) \times_{\{(1,1)\}} \bullet_1)) \times_{\{(2,2)\}} \bullet_2) \times_{\{(3,3)\}} \bullet_3$

 Tabelle 5.3: NLC-Weite von Kreisen C_n und ein k -Ausdruck X_n für C_n , $n = 1, \dots, 12$.

Da die NLC-Weite eines Graphen gleich der NLC-Weite des zugehörigen Komplementgraphen ist, siehe Kapitel 5.4, ist die NLC-Weite des $\overline{C_n}$ gleich der NLC-Weite des Kreises C_n .

Cliquesweiten von Graphen

Tabelle 5.4 zeigt die *Cliquesweite von Wegen* P_n , $n = 1, \dots, 5$. Der Weg mit zwei bzw. drei Knoten hat Cliquesweite 2, da er eine Kante bzw. zwei Kanten enthält und mit Cliquesweite 1 keine Kante erzeugt werden kann. Wege mit vier oder mehr Knoten haben alle Cliquesweite 3 und können mit dem folgenden Cliquesweite 3-Ausdruck konstruiert werden.

$$X_n = \eta_{2,3}(\rho_{3 \rightarrow 2}(\rho_{2 \rightarrow 1}(X_{n-1})) \oplus \bullet_3), \quad n \geq 4$$

$$X_3 = \eta_{2,3}(\eta_{1,2}(\bullet_1 \oplus \bullet_2) \oplus \bullet_3)$$

P_1	1	$X_1 = \bullet_1$
P_2	2	$X_2 = \eta_{1,2}(\bullet_1 \oplus \bullet_2)$
P_3	2	$X_3 = \eta_{1,2}((\bullet_1 \oplus \bullet_1) \oplus \bullet_2)$
P_4	3	$X_4 = \eta_{2,3}((\eta_{1,2}(\bullet_1 \oplus \bullet_2) \oplus \bullet_2) \oplus \bullet_3)$
P_5	3	$X_5 = \eta_{2,3}((\eta_{1,2}(\bullet_1 \oplus \bullet_2) \oplus (\eta_{1,2}(\bullet_1 \oplus \bullet_2))) \oplus \bullet_3)$

Tabelle 5.4: Cliquesweite von Wegen P_n und ein k -Ausdruck X_n für P_n , $n = 1, \dots, 5$.

Tabelle 5.5 zeigt die *Cliquesweite von Komplementen von Wegen* $\overline{P_n}$, $n = 1, \dots, 5$. $\overline{P_1}$ und $\overline{P_2}$ haben Cliquesweite 1, da sie keine Kanten enthalten. $\overline{P_3}$ hat Cliquesweite 2. Für $n \geq 4$ hat der Graph $\overline{P_n}$ Cliquesweite 3 und kann mit dem folgenden Cliquesweite 3-Ausdruck aufgebaut werden.

$$X_n = \eta_{1,3}(\rho_{3 \rightarrow 2}(\rho_{2 \rightarrow 1}(X_{n-1})) \oplus \bullet_3), \quad n \geq 4$$

$$X_3 = \eta_{1,3}((\bullet_1 \oplus \bullet_2) \oplus \bullet_3)$$

$\overline{P_1}$	1	$X_1 = \bullet_1$
$\overline{P_2}$	1	$X_2 = \bullet_1 \oplus \bullet_1$
$\overline{P_3}$	2	$X_3 = \eta_{1,2}(\bullet_1 \oplus \bullet_2) \oplus \bullet_2$
$\overline{P_4}$	3	$X_4 = \eta_{2,3}((\eta_{1,2}(\bullet_1 \oplus \bullet_2) \oplus \bullet_2) \oplus \bullet_3)$
$\overline{P_5}$	3	$X_5 = \eta_{1,2}((\rho_{2 \rightarrow 1}(\eta_{2,3}(\eta_{1,2}(\bullet_1 \oplus \bullet_2) \oplus \eta_{1,3}(\bullet_1 \oplus \bullet_3)))) \oplus \bullet_2)$

Tabelle 5.5: Cliquesweite von Komplementen von Wegen $\overline{P_n}$ und ein k -Ausdruck X_n für $\overline{P_n}$, $n = 1, \dots, 5$.

Die Tabelle 5.6 zeigt die *Cliquesweite von Kreisen* C_n , $n = 1, \dots, 6$. Kreise mit höchstens 4 Knoten haben Cliquesweite 2. Der C_5 und der C_6 haben Cliquesweite 3. Für $n \geq 7$ haben

Kreise C_n Cliquesweite 4. Einen Cliquesweite 4-Ausdruck X_n für C_n kann man analog zum NLC-Weite 4-Ausdruck angeben.

$$P_4 = \eta_{3,4}(\eta_{2,3}(\eta_{1,2}(\bullet_1 \oplus \bullet_2) \oplus \bullet_3)) \oplus \bullet_4$$

$$P_n = \eta_{4,3}((\rho_{4 \rightarrow 3}(\rho_{3 \rightarrow 2}(P_{n-1}))) \oplus \bullet_4), \quad n \geq 5$$

$$X_n = \eta_{1,4}(P_n)$$

C_1	1	$X_1 = \bullet_1$
C_2	2	$X_2 = \eta_{1,2}(\bullet_1 \oplus \bullet_2)$
C_3	2	$X_3 = \eta_{1,2}(\rho_{2 \rightarrow 1}(\eta_{1,2}(\bullet_1 \oplus \bullet_2)) \oplus \bullet_2)$
C_4	2	$X_4 = \eta_{1,2}((\bullet_1 \oplus \bullet_1) \oplus (\bullet_2 \oplus \bullet_2))$
C_5	3	$X_5 = \eta_{1,3}((\rho_{3 \rightarrow 2}(\eta_{2,3}((\eta_{1,2}(\bullet_1 \oplus \bullet_2)) \oplus (\eta_{1,3}(\bullet_1 \oplus \bullet_3)))))) \oplus \bullet_3)$
C_6	3	$X_6 = \eta_{2,3}(\bullet_3 \oplus (\rho_{3 \rightarrow 1}(\eta_{1,3}(\bullet_3 \oplus ((\eta_{1,2}(\bullet_1 \oplus \bullet_2)) \oplus (\eta_{1,2}(\bullet_1 \oplus \bullet_2)))))))$

Tabelle 5.6: Cliquesweite von Kreisen C_i und ein k -Ausdruck X_n für C_n , $n = 1, \dots, 6$.

Die Tabelle 5.7 zeigt die *Cliquesweite von Komplementen von Kreisen* $\overline{C_n}$. Die Komplemente der Kreise mit höchstens drei Knoten haben Cliquesweite 1. Der $\overline{C_4}$ hat Cliquesweite 2. $\overline{C_5}$ hat Cliquesweite 3. Für $n \geq 6$ haben $\overline{C_n}$ Cliquesweite 4. Einen Cliquesweite 4-Ausdruck X_n für $\overline{C_n}$ kann man wie folgt angeben.

$$X_n = \eta_{2,4}(\rho_{4 \rightarrow 3}(\rho_{3 \rightarrow 2}(\eta_{1,4}(X_{n-1}))) \oplus \bullet_4), \quad n \geq 5$$

$$X_4 = \eta_{2,4}(\eta_{1,3}((\bullet_1 \oplus \bullet_2) \oplus \bullet_3) \oplus \bullet_4)$$

$\overline{C_1}$	1	$X_1 = \bullet_1$
$\overline{C_2}$	1	$X_2 = \bullet_1 \oplus \bullet_1$
$\overline{C_3}$	1	$X_3 = (\bullet_1 \oplus \bullet_1) \oplus \bullet_1$
$\overline{C_4}$	2	$X_4 = \eta_{1,2}(\bullet_1 \oplus \bullet_2) \oplus \eta_{1,2}(\bullet_1 \oplus \bullet_2)$
$\overline{C_5}$	3	$X_5 = \eta_{1,3}((\rho_{3 \rightarrow 2}(\eta_{2,3}((\eta_{1,2}(\bullet_1 \oplus \bullet_2)) \oplus (\eta_{1,3}(\bullet_1 \oplus \bullet_3)))))) \oplus \bullet_3)$
$\overline{C_6}$	4	$X_6 = \eta_{2,4}(\eta_{1,3}(\eta_{3,4}(\bullet_3 \oplus \bullet_4) \oplus (\rho_{3 \rightarrow 2}(\rho_{4 \rightarrow 1}(\eta_{1,4}(\eta_{3,4}(\eta_{2,3}(\eta_{1,2}(((\bullet_1 \oplus \bullet_2) \oplus \bullet_3) \oplus \bullet_4))))))))))$

Tabelle 5.7: Cliquesweite von Komplementen von Kreisen C_n und ein k -Ausdruck X_n für $\overline{C_n}$, $n = 1, \dots, 6$.

Graphklassen

Nun werden Graphklassen daraufhin untersucht, ob sie beschränkte oder unbeschränkte NLC-Weite (Cliquesweite) haben. Eine gute Übersicht über bekannte Graphklassen liefert das Buch [BLS99] von Brandstädt, Le und Spinrad, dort sind ca. 200 Graphklassen beschrieben, davon sind bisher nur wenige bezüglich ihrer NLC-Weite eingeordnet worden. An der Universität Rostock wurde ein Java Applet entwickelt, mit dem man Inklusionen zwischen Graphklassen testen kann, siehe [ISG]. In Abbildung 5.1 sind einige Graphklassen bezüglich ihrer Inklusionen dargestellt.

Nach Kapitel 3.1 hat eine Graphklasse genau dann beschränkte NLC-Weite, wenn sie beschränkte Cliquesweite hat. Deshalb braucht man sich insbesondere bei Untersuchungen bezüglich unbeschränkter NLC-Weite (unbeschränkter Cliquesweite) einer Graphklasse nur auf eines der beiden Modelle Cliquesweite bzw. NLC-Weite zu beschränken. Bei Graphklassen mit beschränkter Cliquesweite ist die NLC-Weite höchstens so groß wie die Cliquesweite.

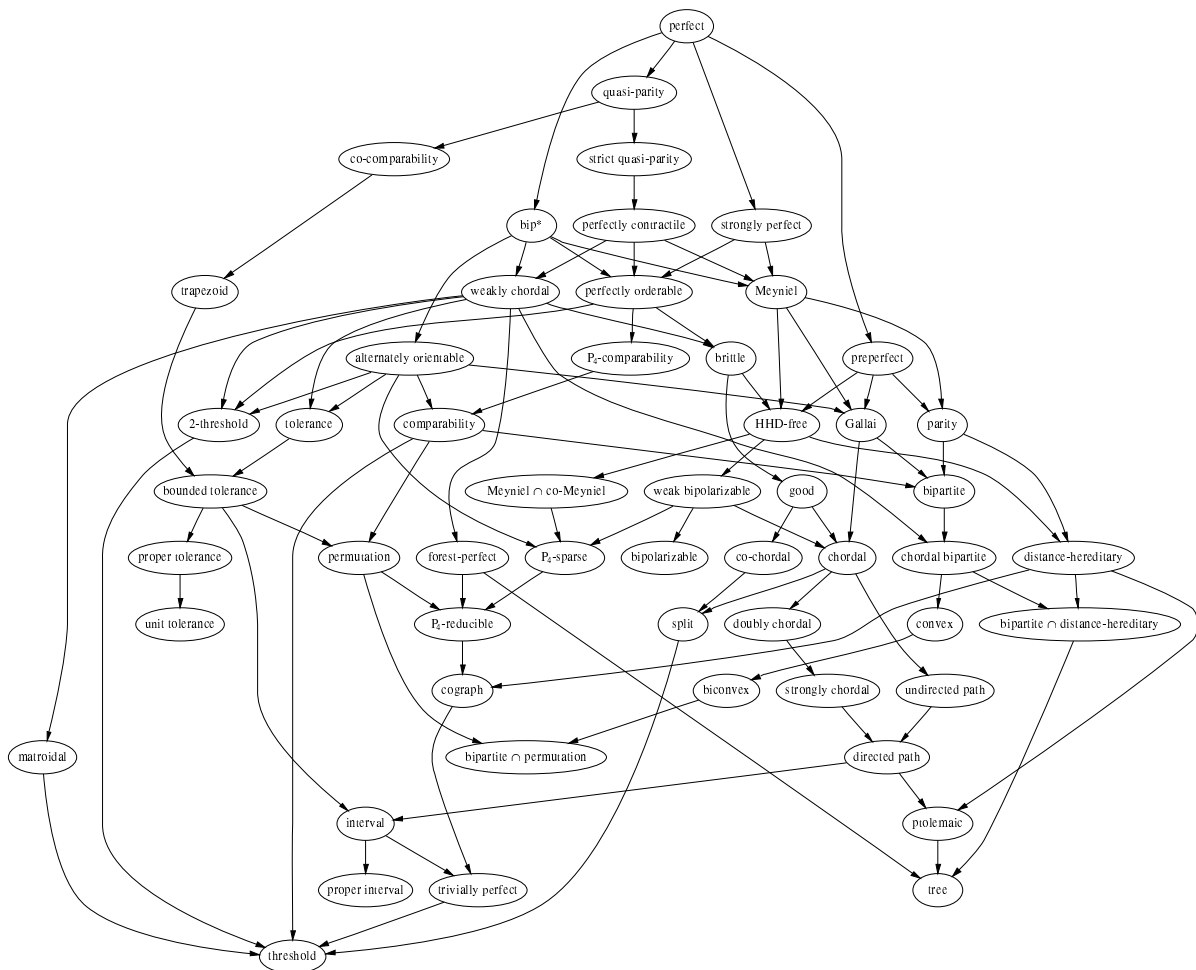


Abbildung 5.1: Die abgebildete Graphklassenhierarchie stellt Inklusionsbeziehungen zwischen den als Knoten angegebenen Graphklassen dar. Eine gerichtete Kante von Graphklasse \mathcal{G}_1 nach Graphklasse \mathcal{G}_2 bedeutet dabei, dass jeder Graph aus der Graphklasse \mathcal{G}_2 auch in der Graphklasse \mathcal{G}_1 enthalten ist.

5.2 Graphklassen mit beschränkter NLC-Weite und Cliquesweite

Eine Menge von Graphen \mathcal{G} hat beschränkte NLC-Weite (beschränkte Cliquesweite), falls es eine natürliche Zahl k gibt, so dass jeder Graph aus \mathcal{G} NLC-Weite (Cliquesweite) höchstens k hat. Das minimale k wird als NLC-Weite (Cliquesweite) der Graphklasse bezeichnet.

5.2.1 NLC-Weite und Cliquesweite von Cographen

Die NLC-Weite und die Cliquesweite von Cographen lassen sich mit Hilfe der folgenden Definition leicht bestimmen.

Definition 5.2.1 (Cograph) *Cographen bestehen aus*

1. einem einzelnen Knoten,
2. aus der disjunkten Vereinigung zweier Cographen G_1, G_2 ($G_1 \cup G_2$) oder
3. aus zwei Cographen G_1, G_2 , die vollständig miteinander verbunden werden ($G_1 \times G_2$).

Beispiele für Cographen sind vollständige Graphen, vollständig unzusammenhängende Graphen und vollständig bipartite Graphen. Ein Graph ist ein Cograph genau dann wenn er keinen P_4 als induzierten Teilgraphen enthält, siehe [Ler71]. Mit Hilfe der rekursiven Definition für Cographen kann man jeden Ausdruck mit den Operationen für Cographen in einen NLC-Weite 1-Ausdruck und einen Cliquesweite 2-Ausdruck umformen und umgekehrt. Für eine Graphklasse \mathcal{G} sei $\text{unlab}(\mathcal{G}) = \{\text{unlab}(G) \mid G \in \mathcal{G}\}$.

Theorem 5.2.2 ([Wan94c, CO00])

$$\text{unlab}(NLC_1) = \text{unlab}(CW_2) = \text{Cographen}$$

5.2.2 NLC-Weite und Cliquesweite von Graphen mit wenigen P_4

Es sind viele Erweiterungen von Cographen definiert worden, indem man die Anzahl der im Graphen vorkommenden P_4 beschränkte. Für viele dieser Graphklassen sind die primen Quotientgraphen, die in einer modularen Dekomposition auftreten, bekannt. Damit kann man wie in Kapitel 4 beschrieben die NLC-Weite und die Cliquesweite dieser Graphklassen über die NLC-Weite bzw. Cliquesweite der primen Quotientgraphen in der modularen Dekomposition bestimmen.

P_4 -reduzierbare Graphen

P_4 -reduzierbare Graphen wurden 1989 von Jamison und Olariu in [JO89] eingeführt. P_4 -reduzierbare Graphen liefern eine Verallgemeinerung von Cographen und sind Permutationsgraphen, siehe Abbildung 5.1.

Definition 5.2.3 (P_4 -reduzierbare Graphen [JO89]) *Ein Graph heißt P_4 -reduzierbar, falls jeder seiner Knoten zu höchstens einem induzierten P_4 gehört.*

Lemma 5.2.4 *Sei G ein P_4 -reduzierbarer Graph, dann gilt $\text{NLC-Weite}(G) \leq 2$.*

Beweis In [GV97] wird gezeigt, dass die primen Quotientgraphen in der modularen Dekomposition von P_4 -reduzierbaren Graphen isomorph zu einem P_4 oder zu einem bull sind. Nach Theorem 4.3.3 ist die NLC-Weite eines Graphen G gleich dem Maximum der NLC-Weiten der primen Quotientgraphen in der modularen Dekomposition von G . Da der P_4 und der bull NLC-Weite 2 haben, hat jeder P_4 -reduzierbare Graph NLC-Weite höchstens 2. \square

In [CMR00] nutzen Courcelle, Makowsky und Rotics ebenfalls die Quotientgraphen der P_4 -reduzierbaren Graphen um zu zeigen, dass P_4 -reduzierbare Graphen Cliquesweite höchstens 3 haben. In [GV97] werden neben P_4 -reduzierbaren Graphen auch erweitert P_4 -reduzierbare Graphen definiert. Ein Graph heißt *erweitert P_4 -reduzierbar*, falls jeder seiner Knoten zu einem induzierten C_5 oder höchstens einem induzierten P_4 gehört. In [GV97] wurde gezeigt, dass die primen Quotientgraphen von erweitert P_4 -reduzierbaren Graphen isomorph zu einem P_4 , zu einem bull oder zu einem C_5 sind. Mit Hilfe dieser Quotientgraphen wurde in [CMR00] gezeigt, dass die Cliquesweite von erweitert P_4 -reduzierbaren Graphen höchstens 3 ist. Da P_4 , bull und C_5 NLC-Weite 2 haben, haben erweitert P_4 -reduzierbare Graphen NLC-Weite höchstens 2.

P_4 -sparse Graphen

Die Klasse der P_4 -sparse Graphen wurde 1985 von Hoang in [Hoa85] eingeführt. P_4 -sparse Graphen liefern eine Verallgemeinerung von Cographen.

Definition 5.2.5 ((extended) P_4 -sparse Graphen [Hoa85]) *Ein Graph G ist P_4 -sparse, falls jede Menge von fünf Knoten in G höchstens einen P_4 in G induziert. Ein Graph G heißt extended P_4 -sparse, falls jede Menge von 5 Knoten entweder einen C_5 in G induziert oder höchstens einen P_4 in G induziert.*

Um die NLC-Weite von (extended) P_4 -sparse Graphen zu bestimmen definiert man *Spinnen (Spiders)*, die als Quotientgraphen in der modularen Dekomposition von (extended) P_4 -sparse Graphen auftreten. Spinnen sind spezielle Splitgraphen¹ und sind folgendermaßen definiert.

Definition 5.2.6 (Spinne) *Ein Graph $G = (V, E)$ ist eine Spinne, falls die Knotenmenge V disjunkt in drei Mengen S , K und R aufgeteilt werden kann, so dass folgende drei Bedingungen gelten.*

- (i) S ist eine unabhängige Menge, K induziert eine Clique in G und $|S| = |K| \geq 2$.
- (ii) $|R| \leq 1$. Falls $R \neq \emptyset$, so ist der Knoten in R zu allen Knoten aus K adjazent und zu keinem Knoten aus S adjazent.
- (iii) Es gibt eine Bijektion f zwischen den Knoten aus S und den Knoten aus K , so dass für alle Knoten x in S entweder
 - (a) $N(x) = \{f(x)\}$ (dünne Spinne) oder
 - (b) $N(x) = K - \{f(x)\}$ (dicke Spinne) gilt.

¹Ein Graph G ist ein *Splitgraph*, falls seine Knotenmenge in zwei disjunkte Mengen V_1 und V_2 aufgeteilt werden kann, so dass V_1 eine Clique in G induziert und V_2 eine unabhängige Menge in G ist.

Das Tripel (S, K, R) heißt Spinnenpartition für G .

Dicke Spinnen sind Komplemente dünner Spinnen. Jede so definierte Spinne ist primer Graph. In der Literatur werden Spinnen auch für $|R| > 1$ definiert, vergleiche [JO92], dann unterscheidet man zwischen *primen Spinnen* ($|R| \leq 1$) und *Spinnen*.

Ein Graph G ist *distanzerhaltend*, falls in jedem zusammenhängenden induzierten Teilgraphen H von G die Distanz zweier Knoten in H gleich der Distanz der beiden Knoten in G ist.

Lemma 5.2.7 *Jede dünne Spinne ist distanzerhaltend.*

Beweis Es sei G eine dünne Spinne mit einer Spinnenpartition (S, K, R) und $G_1 = (S_1, K_1, R_1)$, mit $S_1 \subseteq S$, $K_1 \subseteq K$ und $R_1 \subseteq R$, sei ein zusammenhängender induzierter Teilgraph von G . Es seien x und y zwei Knoten aus G_1 . Aus der folgenden Tabelle folgt, dass die Distanz zweier Knoten aus G_1 in G gleich der Distanz der beiden Knoten in G_1 ist, damit ist G distanzerhaltend.

$x \in$	$y \in$	
K_1	K_1	$\text{dist}_G(x, y) = 1 = \text{dist}_{G_1}(x, y)$
S_1	S_1	$\text{dist}_G(x, y) = 3 = \text{dist}_{G_1}(x, y)$
K_1	R_1	$\text{dist}_G(x, y) = 1 = \text{dist}_{G_1}(x, y)$
S_1	R_1	$\text{dist}_G(x, y) = 2 = \text{dist}_{G_1}(x, y)$
S_1	K_1	$N(x) = \{y\}$ $\text{dist}_G(x, y) = 1 = \text{dist}_{G_1}(x, y)$
		$N(x) \neq \{y\}$ $\text{dist}_G(x, y) = 2 = \text{dist}_{G_1}(x, y)$

□

Dicke Spinnen (S, K, R) sind im Allgemeinen nicht distanzerhaltend, wie ein Gegenbeispiel mit $|S| = |K| = 5$ zeigt. Courcelle, Makowsky und Rotics haben in [CMR00] gezeigt, dass Spinnen Cliquesweite höchstens 4 besitzen. Für dicke Spinnen ist diese Schranke scharf, da dicke Spinnen für $|S| = |K| = 2$ und $|S| = |K| = 3$ Cliquesweite 3 haben und für $|S| = |K| \geq 4$ Cliquesweite 4 haben. Für dünne Spinnen kann die Abschätzung der Cliquesweite wie folgt verbessert werden.

Lemma 5.2.8 *Es sei G eine dünne Spinne, dann gilt $\text{Cliquesweite}(G) = 3$.*

Beweis Da jeder distanzerhaltende Graph Cliquesweite höchstens 3 hat, siehe [GR00], folgt mit Lemma 5.2.7 bereits, dass die Cliquesweite einer dünnen Spinne höchstens 3 beträgt.

Weiterhin kann man für eine dünne Spinne G mit einer zugehörigen Spinnenpartition (S, K, R) mit $S = \{s_1, \dots, s_n\}$ und $K = \{k_1, \dots, k_n\}$ wie folgt einen Cliquesweite 3-Ausdruck angeben. Nach der Definition einer dünnen Spinne gibt es eine Bijektion $f : S \rightarrow K$, so dass für $x \in S$ gilt $N(x) = \{f(x)\}$. Wir können ohne Einschränkungen annehmen, dass $f(s_i) = k_i$ gilt, d.h. $N(s_i) = \{k_i\}$. Es sei G_j , $1 \leq j \leq n$, der Teilgraph von G der durch die Knoten aus $S_j := \{s_1, \dots, s_j\}$ und $K_j := \{k_1, \dots, k_j\}$ induziert wird. Wir markieren hier die Knoten im folgenden Cliquesweite 3-Ausdruck X_j für G_j neben der Markierung aus $[k]$ noch mit dem Knotenindex s_i bzw. k_i .

$$X_1 := \eta_{1,2}(\bullet_{1,(s_1)} \oplus \bullet_{2,(k_1)})$$

$$X_j := \rho_{3 \rightarrow 2}(\eta_{2,3}(X_{j-1} \oplus \eta_{1,3}(\bullet_{1,(s_j)} \oplus \bullet_{3,(k_j)}))), \quad j \geq 2$$

Für $R = \emptyset$ definiert der Cliquesweite 3-Ausdruck X_n den Graphen G und falls $R = \{r\}$ definiert der Cliquesweite 3-Ausdruck $X'_n = \eta_{2,3}(X_n \oplus \bullet_3)$ den Graphen G .

Da jede dünne Spinne den P_4 als induzierten Teilgraphen enthält, hat jede dünne Spinne Cliquesweite 3. \square

Das folgende Lemma zeigt die NLC-Weite von Spinnen.

Lemma 5.2.9 *Es sei G eine Spinne, dann gilt $NLC\text{-Weite}(G) = 2$.*

Beweis Es sei $G = (V, E)$ eine Spinne und (S, K, R) eine Spinnenpartition für G . Es seien $S = \{s_1, \dots, s_n\}$, $K = \{k_1, \dots, k_n\}$, $n \geq 2$ und $R = \{r\}$ oder $R = \emptyset$. Nach Definition 5.2.6 (iii) ist G entweder eine dünne Spinne, für die $N(s_i) = \{f(s_i)\}$ gilt, oder eine dicke Spinne, für die $N(s_i) = K - \{f(s_i)\}$, $i = 1, \dots, n$, gilt. Wir können ohne Einschränkungen annehmen, dass $f(s_i) = k_i$ gilt, d.h. $N(s_i) = \{k_i\}$ bzw. $N(s_i) = K - \{k_i\}$.

Wir zeigen nun, dass jede Spinne G NLC-Weite höchstens 2 hat, indem wir einen NLC-Weite 2-Ausdruck für G angeben. Wir betrachten zunächst nur den durch die Mengen S und K induzierten Teilgraphen von G . In dem aufgebauten markierten Graphen werden die Knoten aus S mit 1 markiert und die Knoten aus K mit 2 markiert.

1. Falls G eine dünne Spinne ist, d.h. $N(s_i) = \{f(s_i)\} = \{k_i\}$, definieren rekursiv die NLC-Weite 2-Ausdrücke

$$X_1 := \bullet_{1,(s_1)} \times_{\{(1,2)\}} \bullet_{2,(k_1)}$$

$$X_j := (\bullet_{1,(s_j)} \times_{\{(1,2)\}} \bullet_{2,(k_j)}) \times_{\{(2,2)\}} X_{j-1}, \quad j \geq 2$$

Wir beweisen nun induktiv, dass $\text{val}(X_j)$, $j \geq 2$ der Teilgraph von G ist, der durch die Knoten aus $S_j := \{s_1, \dots, s_j\}$ und $K_j := \{k_1, \dots, k_j\}$ induziert wird.

- Induktionsanfang ($j = 2$): $\text{val}(X_2)$ ist ein mit 1-2-2-1 markierter P_4 . Je zwei Knoten aus K sind adjazent, je zwei Knoten aus S sind nicht adjazent und $N(s_i) = \{k_i\}$, $j = 1, 2$.
 - Induktionsschritt ($j - 1 \rightarrow j$): Gelte die Behauptung für $\text{val}(X_{j-1})$. $\text{val}(X_j)$ entsteht aus $\text{val}(X_{j-1})$ indem jeder mit 2 markierte Knoten in dem Graph $\text{val}(\bullet_{1,(s_j)} \times_{\{(1,2)\}} \bullet_{2,(k_j)})$ mit jedem mit 2 markiertem Knoten in $\text{val}(X_{j-1})$ verbunden wird. In $\text{val}(X_j)$ sind also je zwei Knoten aus S_j nicht adjazent, da in S_{j-1} nach Induktionsannahme je zwei Knoten nicht adjazent sind und s_j zu keinem $s \in S_{j-1}$ verbunden wird. Je zwei Knoten aus K_j sind adjazent, da nach Induktionsannahme je zwei Knoten aus K_{j-1} adjazent sind und k_j mit allen Knoten aus K_{j-1} verbunden wird. $N(s_i) = \{k_i\}$ gilt für $i < j$ nach Induktionsannahme und für $i = j$, da s_j adjazent zu k_j ist.
2. Falls G eine dicke Spinne ist, d.h. $N(s_i) = K - \{f(s_i)\} = K - \{k_i\}$, definieren wir die NLC-Weite 2-Ausdrücke

$$X_1 := \bullet_{1,(s_1)} \times_{\emptyset} \bullet_{2,(k_1)}$$

$$X_j := (\bullet_{1,(s_j)} \times_{\emptyset} \bullet_{2,(k_j)}) \times_{\{(1,2),(2,1),(2,2)\}} X_{j-1}, \quad j \geq 2$$

Wir beweisen induktiv, dass $\text{val}(X_j)$, $j \geq 2$ der Teilgraph von G ist, der durch die Knoten aus $S_j := \{s_1, \dots, s_j\}$ und $K_j := \{k_1, \dots, k_j\}$ induziert wird.

- Induktionsanfang ($j = 2$): $\text{val}(X_2)$ ist ein mit 1-2-2-1 markierter P_4 . Je zwei Knoten aus K sind adjazent, je zwei Knoten aus S sind nicht adjazent und $N(s_i) = K_2 - \{k_i\}$, $i = 1, 2$.
- Induktionsschritt ($j-1 \rightarrow j$): Gelte die Behauptung für $\text{val}(X_{j-1})$. $\text{val}(X_j)$ entsteht aus $\text{val}(X_{j-1})$ indem man in dem Graphen $\text{val}(\bullet_{1,(s_j)} \times_{\emptyset} \bullet_{2,(k_j)})$ jeden mit $2, (1), [2]$ markierten Knoten mit jedem mit $2, (2), [1]$ markierten Knoten in $\text{val}(X_{j-1})$ verbindet. In $\text{val}(X_j)$ sind also je zwei Knoten aus S_j nicht adjazent, da in S_{j-1} nach Induktionsannahme je zwei Knoten nicht adjazent sind und s_j zu keinem $s \in S_{j-1}$ verbunden wird. Je zwei Knoten aus K_j sind adjazent, da nach Induktionsannahme je zwei Knoten aus K_{j-1} adjazent sind und k_j mit allen Knoten aus K_{j-1} verbunden wird. $N(s_i) = K - \{k_i\}$ gilt für $i < j$ nach Induktionsannahme und für $i = j$, da s_j nicht adjazent zu k_j ist und s_j mit allen k_i , $i < j$ verbunden wird und k_j mit allen s_i , $i < j$ verbunden wird.

$\text{val}(X_n)$ ist also der Teilgraph von G der durch $S \cup K = V - R$ induziert wird. In $\text{val}(X_n)$ sind alle Knoten aus K mit 2 markiert, damit ist G der Graph $\text{val}(X_n \times_{\{(2,2)\}} \bullet_{2,(r)})$, falls $R = \{r\}$ bzw. der Graph $\text{val}(X_n)$, falls $R = \emptyset$.

Jede Spinne enthält einen P_4 als induzierten Teilgraphen, somit hat jede Spinne NLC-Weite genau 2. \square

Damit kann nun gezeigt werden, dass P_4 -sparse und extended P_4 -sparse Graphen beschränkte NLC-Weite haben.

Lemma 5.2.10 *Jeder P_4 -sparse Graph und jeder extended P_4 -sparse Graph hat NLC-Weite höchstens 2.*

Beweis Giakoumakis und Vanherpe zeigen in der Schrift [GV97], dass die primen Quotientgraphen in der modularen Dekomposition für P_4 -sparse Graphen isomorph zu einer Spinne sind. Da $\text{NLC-Weite}(\text{Spinne}) = 2$ folgt aus Theorem 4.3.3 die Behauptung. Für extended P_4 -sparse Graphen sind die primen Quotientgraphen nach einem Ergebnis in [GV97] isomorph zu einer Spinne oder zu einem C_5 . Aus $\text{NLC-Weite}(\text{Spinne}) = 2$ und $\text{NLC-Weite}(C_5) = 2$ folgt mit Theorem 4.3.3 die Behauptung. \square

In [CMR00] wurde mit Hilfe der bekannten Quotientgraphen gezeigt, dass P_4 -sparse Graphen und extended P_4 -sparse Graphen Cliquesweite höchstens 4 haben.

P_4 -tidy Graphen

P_4 -tidy Graphen wurden 1995 von Rusu eingeführt. P_4 -tidy Graphen verallgemeinern die extended P_4 -sparse Graphen.

Definition 5.2.11 (P_4 -tidy Graphen [GRT97]) *Es sei G ein Graph und X sei die Knotenmenge eines induzierten P_4 von G . Ein Knoten v außerhalb von X heißt Partner von X , falls die Knoten in der Menge $X \cup \{v\}$ mindestens zwei P_4 in G induzieren. G heißt P_4 -tidy, falls jeder induzierte P_4 in G höchstens einen Partner hat.*

Lemma 5.2.12 *Es sei G ein P_4 -tidy Graph, dann gilt $\text{NLC-Weite}(G) \leq 2$.*

Beweis In [GRT97] wird gezeigt, dass die primen Quotientgraphen für P_4 -tidy Graphen isomorph zu einer Spinne, zu einem C_5 , zu einem P_5 oder zu einem $\overline{P_5}$ sind. Da $\text{NLC-Weite}(\text{Spinne}) = 2$, $\text{NLC-Weite}(C_5) = 2$, $\text{NLC-Weite}(P_5) = \text{NLC-Weite}(\overline{P_5}) = 2$ folgt die Behauptung mit Theorem 4.3.3. \square

In [CMR00] wurde mit Hilfe der bekannten Quotientgraphen gezeigt, dass P_4 -tidy Graphen Cliquesweite höchstens 4 haben.

(q, t) -Graphen

Babel und Olariu führen in dem Artikel [BO98] die Klasse der (q, t) -Graphen ein.

Definition 5.2.13 ((q, t)-Graphen [BO98]) *Ein Graph G ist ein (q, t) -Graph, falls jede Menge von q Knoten aus G höchstens t disjunkte induzierte P_4 enthält.*

Es sei immer $q \geq 4$. $(4, 0)$ -Graphen entsprechen genau den Cographen und $(5, 1)$ -Graphen den P_4 -sparse Graphen. Es werden nun die Fälle $t = q - 4$ und $t = q - 3$ untersucht.

Lemma 5.2.14 *Es sei G ein $(q, q - 4)$ -Graph, dann gilt $\text{NLC-Weite}(G) \leq \lceil \frac{q}{2} \rceil$.*

Beweis Babel und Olariu haben in [BO98] gezeigt, dass die primen Quotientgraphen für $(q, q - 4)$ -Graphen isomorph zu einer Spinne oder zu einem Graphen mit höchstens q Knoten sind. Da $\text{NLC-Weite}(\text{Spinne}) = 2$, $\text{NLC-Weite}(G) \leq \lceil \frac{q}{2} \rceil$, falls $|V_G| \leq q$ und $q \geq 4$ folgt aus Theorem 4.3.3 die Behauptung. \square

In [CMR00] wurde nach diesem Schema gezeigt, dass $(q, q - 4)$ -Graphen Cliquesweite höchstens q haben. Nun bestimmen wir die NLC-Weite von $(q, q - 3)$ -Graphen.

Lemma 5.2.15 *Es sei G ein $(q, q - 3)$ -Graph mit $q \geq 7$, dann gilt $\text{NLC-Weite}(G) \leq \max\{6, \lceil \frac{q}{2} \rceil\}$.*

Beweis Die primen Quotientgraphen von $(q, q - 3)$ -Graphen sind nach [Bab98] isomorph zu einem prim p -tree, einem disc (d.h. ein C_n oder ein $\overline{C_n}$), einer Spinne oder zu einem Graphen mit höchstens q Knoten.

Ein prim p -tree hat NLC-Weite höchstens 6, da er nach [MR99] Cliquesweite höchstens 6 hat. Ein disc hat NLC-Weite höchstens 4. $\text{NLC-Weite}(G) \leq \lceil \frac{q}{2} \rceil$, falls $|V_G| \leq q$ und $\text{NLC-Weite}(\text{Spinne}) = 2$. Damit folgt mit Theorem 4.3.3 die Behauptung. \square

In [MR99] haben Makowsky und Rotics gezeigt, dass $(q, q - 3)$ -Graphen mit $q \geq 7$ Cliquesweite höchstens q haben.

5.2.3 Weitere Graphklassen mit beschränkter NLC-Weite und Cliquesweite

Graphklassen mit Baumweite k haben, wie bereits in Kapitel 3.2 erwähnt, Cliquesweite höchstens $3 \cdot 2^{k-1}$. Damit haben Wälder (Graphen mit Baumweite 1) NLC-Weite und Cliquesweite höchstens 3 und series-parallele Graphen (Graphen mit Baumweite höchstens 2) NLC-Weite und Cliquesweite höchstens 6.

Golumbic und Rotics zeigen in [GR00], dass distanzerhaltende Graphen Cliquesweite höchstens 3 und damit auch NLC-Weite höchstens 3 haben.

Brandstädt und andere Autoren untersuchen die Cliquesweite von Graphen die sich durch verbotene Erweiterungen des P_4 um einen Knoten (wie zum Beispiel gem, P_5 , siehe Tabelle 5.1) ergeben und zeigen, dass $(P_5, \text{diamond})$ -freie Graphen [Bra01], $(P_5, \text{co-}P_5, \text{co-chair})$ -freie Graphen [BM03], $(P_5, \text{co-chair, gem})$ -freie Graphen [BM03], $(P_5, \text{co-chair, bull})$ -freie Graphen [BM03], $(\text{chair, co-}P, \text{gem})$ -freie Graphen [BLV03], $(\text{bull, chair, co-chair})$ -freie Graphen [BHL03], (P_5, gem) -freie Graphen [BLM02a], (gem, co-gem) -freie Graphen [BLM02b], chordale gem-freie Graphen (distanzerhaltende Graphen) und chordale co-gem-freie Graphen [BLM02a] beschränkte Cliquesweite und damit auch beschränkte NLC-Weite haben. In [BDLM02] untersuchen Brandstädt und andere die Cliquesweite von (G_1, G_2, G_3) -freien Graphen für alle Erweiterungen G_i , $1 \leq i \leq 3$, des P_4 um einen Knoten.

5.3 Graphklassen mit unbeschränkter NLC-Weite und Cliquesweite

Eine Menge \mathcal{G} von Graphen hat unbeschränkte NLC-Weite (bzw. unbeschränkte Cliquesweite) falls es keine natürliche Zahl k gibt, so dass jeder Graph aus \mathcal{G} NLC-Weite (bzw. Cliquesweite) höchstens k hat.

Wir geben eine notwendige Bedingung für Graphen mit NLC-Weite höchstens k an.

Theorem 5.3.1 *Sei $G = (V, E)$ ein Graph mit NLC-Weite höchstens k , dann existiert eine balancierte Partition U_1, U_2 von V , d.h. es gibt $U_1, U_2 \subseteq V$, $U_1 \cap U_2 = \emptyset$, $U_1 \cup U_2 = V$, $|U_1| \geq \frac{1}{3}|V|$, $|U_2| \geq \frac{1}{3}|V|$ und es gilt:*

$$|N(U_1)| \leq k \text{ und } |N(U_2)| \leq 2^k$$

oder

$$|N(U_2)| \leq k \text{ und } |N(U_1)| \leq 2^k,$$

wobei $N(U_1) = \{N(u) \cap U_2 \mid u \in U_1\}$, $N(U_2) = \{N(u) \cap U_1 \mid u \in U_2\}$ und $N(u) = \{v \mid \{u, v\} \in E\}$.

Beweis Es sei $G = (V, E)$ ein Graph mit NLC-Weite höchstens k . In jedem Baum $B = (V, E)$ gibt es eine Kante $e \in E$, so dass B beim Entfernen dieser Kante (die Endknoten bleiben erhalten) in genau zwei Bäume zerfällt die je mindestens $\frac{1}{3}$ und je höchstens $\frac{2}{3}$ der Blätter von B enthalten. Somit gibt es auch in jedem NLC-Weite k -Ausdrucksbaum T für G eine gerichtete Kante $e = (u, v)$ mit $u, v \in V_T$ mit dieser Eigenschaft. Wir wissen nun, dass der Teilbaum $T(u)$ mindestens $\frac{1}{3} \cdot |V|$ Blätter enthält, die Knoten aus G entsprechen und dass die Anzahl der restlichen Blätter in T ebenfalls mindestens $\frac{1}{3} \cdot |V|$ beträgt. Es sei nun U_1 die Menge der Knoten in G die den Blättern in $T(u)$ entsprechen und U_2 sei die Menge der Knoten in $V - U_1$. Die Knoten u_1 in $G(u)$ haben höchstens k verschiedene Markierungen. Somit gibt es höchstens k verschiedene Nachbarschaften $N_{U_2}(u_1) = \{u_2 \in U_2 \mid \{u_1, u_2\} \in E_G\}$ der Knoten aus U_1 bezüglich der Menge U_2 , da alle Knoten die in $G(u)$ gleich markiert sind im weiteren Verlauf der Konstruktion in T gleich behandelt werden. Die Knoten aus der Menge U_2 haben höchstens 2^k verschiedene Nachbarschaften in der Menge U_1 . Wählt man U_2 als die Menge der Knoten in G die den Blättern in $T(u)$ entsprechen und $U_1 = V - U_2$, so enthält man die zweite Aussage analog. \square

Die folgende Umkehrung des letzten Theorem ist ein mögliches Hilfsmittel um zu zeigen, dass eine Graphklasse unbeschränkte NLC-Weite hat.

Korollar 5.3.2 *Ein Graph $G = (V, E)$ hat NLC-Weite $> k$, falls für alle balancierten Partition U_1, U_2 von V mit $U_1, U_2 \subseteq V$, $U_1 \cap U_2 = \emptyset$, $U_1 \cup U_2 = V$, $|U_1| \geq \alpha|V|$, $|U_2| \geq \alpha|V|$:*

$$|N(U_1)| > k \text{ und } |N(U_2)| > k$$

Da auch jeder Graph mit Cliquesweite k einen Cliquesweite k -Ausdrucksbaum besitzt, kann man die letzten beiden Aussagen auch für die Cliquesweite zeigen.

Mit Hilfe einer ähnlichen Beobachtung wurde von Golubic und Rotics in [GR00] gezeigt, dass Permutationsgraphen², Intervallgraphen³ (und damit auch chordale⁴ Graphen, da diese die Menge der Intervallgraphen beinhalten), $n \times n$ -Gitter (und somit auch bipartite und planare Graphen, da jedes Gitter bipartite und planar ist) unbeschränkte Cliquesweite haben. Weiterhin haben Makowsky und Rotics in [MR99] gezeigt, dass Splitgraphen (und damit auch P_5 -freie Graphen, da Splitgraphen alle P_5 -frei sind), $(q, q - 3)$ -Graphen, $4 \leq q \leq 6$, $(q, q - 1)$ -Graphen, $q \geq 4$, $(6, 3)$ -Graphen und $(7, 5)$ -Graphen unbeschränkte Cliquesweite haben. Weiterhin haben Linegraphen von Cliques nach [KR01] und k -reguläre Graphen für $k \geq 3$ unbeschränkte Cliquesweite.

5.4 Graphoperationen auf Graphen mit beschränkter NLC-Weite

Hier werden Graphoperationen betrachtet, die die NLC-Weite und Cliquesweite k eines Graphen um eine Konstante oder in Abhängigkeit von k erhöhen.

Kantenkomplementbildung

Für jeden Graphen G und den dazugehörigen Komplementgraphen \overline{G} gilt

$$\text{NLC-Weite}(G) = \text{NLC-Weite}(\overline{G}) \quad (5.1)$$

Man kann einen NLC-Weite k -Ausdruck für G wie folgt in einen NLC-Weite k -Ausdruck für \overline{G} umformen [Wan94c].

$$\begin{array}{c|c} G & \overline{G} \\ \hline \bullet_t & \bullet_t \\ \circ_R(H) & \circ_R(\overline{H}) \\ H_1 \times_S H_2 & \overline{H}_1 \times_{\overline{S}} \overline{H}_2 \end{array}$$

wobei $\overline{S} = \{(i, j) \in [k]^2 \mid (i, j) \notin S\}$, $t \in [k]$, $H, H_1, H_2 \in \text{NLC}_k$. Somit sind Graphklassen mit beschränkter NLC-Weite abgeschlossen bezüglich Komplementbildung. Für die Cliquesweite eines Graphen G zeigen Courcelle und Olariu in [CO00]

$$\text{Cliquesweite}(G) \leq 2 \cdot \text{Cliquesweite}(\overline{G}). \quad (5.2)$$

Für Graphen mit Cliquesweite 1 hat der Komplementgraph Cliquesweite 2. Graphen mit Cliquesweite 2 sind Cographen, die abgeschlossen bezüglich Komplementbildung sind, somit hat das Komplement eines Graphen mit Cliquesweite 2 die Cliquesweite höchstens 2. Ob die Schranke (5.2) für Graphen mit Cliquesweite $k \geq 3$ angenommen wird, ist bisher unbekannt.

²Ein Graph $G = (V, E)$ ist ein *Permutationsgraph*, wenn es eine Bijektion $f : V \rightarrow \{1, \dots, |V|\}$ und eine Permutation π gibt, so dass $\{u, v\} \in E \Leftrightarrow f(u) < f(v)$ und $\pi_{f(u)} > \pi_{f(v)}$.

³Ein Graph heißt *Intervallgraph*, wenn seine Knoten durch reelle Intervalle dargestellt werden können, so dass zwei Knoten genau dann adjazent sind, wenn sich die zugehörigen Intervalle schneiden.

⁴Ein Graph heißt *chordal*, falls er keine induzierten Kreise der Länge ≥ 4 enthält.

Induzierte Teilgraphbildung

Für jeden induzierten Teilgraphen H eines Graphen G gilt

$$\text{NLC-Weite}(H) \leq \text{NLC-Weite}(G),$$

da man aus einem NLC-Weite k -Ausdruck für G einfach einen NLC-Weite k -Ausdruck für H gewinnen kann, indem man nur die Knoten aus H betrachtet. Ebenso gilt für die Cliquesweite eines Graphen G und einen beliebigen induzierten Teilgraphen H von G

$$\text{Cliquesweite}(H) \leq \text{Cliquesweite}(G).$$

Somit sind Graphklassen mit beschränkter NLC-Weite und Graphklassen mit beschränkter Cliquesweite abgeschlossen bezüglich induzierten Teilgraphen, siehe auch [Wan94c, CO00]. Graphklassen mit beschränkter NLC-Weite sind ebenso wie Graphklassen mit beschränkter Cliquesweite nicht abgeschlossen bezüglich Teilgraphen im Allgemeinen, da eine Clique mit n Knoten NLC-Weite 1 (Cliquesweite 2) hat, jedoch nicht jeder Graph in NLC_1 (bzw. nicht in CW_2) enthalten ist. Damit sind Graphklassen mit beschränkter Cliquesweite und Graphklassen mit beschränkter NLC-Weite auch nicht bezüglich Minoren abgeschlossen.

Potenz eines Graphen

Definition 5.4.1 (Potenzgraph, G^d) Für einen Graphen $G = (V_G, E_G)$ und eine natürliche Zahl d bezeichne G^d den Potenzgraph zu G . G^d hat die Knotenmenge V_G . Zwei Knoten sind in G^d genau dann adjazent, wenn sie in G die Distanz höchstens d haben.

Lemma 5.4.2 Es sei G ein Graph mit NLC-Weite k , dann gilt $\text{NLC-Weite}(G^2) \leq k \cdot 2^k$.

Beweis Aus einem NLC-Weite k -Ausdruck X für G kann wie folgt ein $k \cdot 2^k$ -Ausdruck X' für G^2 konstruiert werden. Jeder Knoten x aus G^2 wird in X' mit einem Paar (t, L) markiert. $t \in [k]$ ist die Markierung des Knotens x im NLC-Weite k -Ausdruck X für G und $L \subseteq [k]$ ist die Menge der Markierungen der Nachbarn des Knoten x in G über Kanten aus G . Für einen Knoten x in G^2 bezeichne $\text{lab}(x)$ das Paar $(t, L) \in [k] \times 2^{[k]}$ mit dem x markiert ist.

1. Falls $X = \bullet_t$, so setze $X' = \bullet_{(t, \emptyset)}$
2. Falls $X = \circ_R(X_1)$, so setze $X' = \circ_{R'}(X'_1)$, wobei $R'((t, L)) = (R(t), \{R(t_i) \mid t_i \in L\})$ die Ummarkierungsoperation auf G^2 ist. X'_1 kann nach (1.) oder (3.) aus X_1 bestimmt werden.
3. Falls $X = X_1 \times_S X_2$, so setze $X' = \circ_{R'_1}(X'_1) \times_{S'} \circ_{R'_2}(X'_2)$.

Da durch eine \times_S -Operation Kanten eingefügt werden, ändert sich die Nachbarschaft der Knoten die an der \times_S Operation beteiligt sind und somit müssen die L Mengen der Knoten im Graph $\text{val}(X'_1 \times_{S'} X'_2)$ aktualisiert werden. Da diese Aktualisierung für die Knoten aus $V_{\text{val}(X'_1)}$ und für die Knoten aus $V_{\text{val}(X'_2)}$ unterschiedlich ist, wird diese vor dem Einfügen der Kanten wie folgt vorgenommen und beeinflusst die Korrektheit der eingefügten Kanten nicht.

Für alle $(a, b) \in S$ und alle $x \in V_{\text{val}(X'_1)}$ mit $\text{lab}(x) = (a, L)$, für ein $L \subseteq 2^{[k]}$, setze $\text{lab}(x) = (a, L \cup \{b\})$ und für alle $y \in V_{\text{val}(X'_2)}$ mit $\text{lab}(y) = (b, L)$, für ein $L \subseteq 2^{[k]}$, setze

$\text{lab}(y) = (b, L \cup \{a\})$. Dies wird durch die Ummarkierungen $R_1''(a, L) = (a, L \cup \{b\})$ für alle $(a, b) \in S$ und alle $L \subseteq 2^{[k]}$ und $R_2''(b, L) = (b, L \cup \{a\})$ für alle $(a, b) \in S$ und alle $L \subseteq 2^{[k]}$ erreicht.

Die Verbindungsrelation $\times_{S'}$ für die beiden Graphen $\text{val}(X_1')$ und $\text{val}(X_2')$ wird wie folgt definiert.

$$S' = \{((a, L), (b, L')) \mid (a, b) \in S, L, L' \subseteq 2^{[k]}\} \cup \\ \{((x, L), (b, L')) \mid (a, b) \in S, x \in [k], a \in L \subseteq 2^{[k]}, L' \subseteq 2^{[k]}\} \cup \\ \{((a, L), (y, L')) \mid (a, b) \in S, y \in [k], b \in L' \subseteq 2^{[k]}, L \subseteq 2^{[k]}\}$$

$\text{val}(X')$ entsteht aus der disjunkten Vereinigung von $\text{val}(\circ_{R_1''}(X_1'))$ und $\text{val}(\circ_{R_2''}(X_2'))$ in der für jedes Paar $((t, L), (t', L')) \in S'$ jeder mit (t, L) markierte Knoten aus $\text{val}(\circ_{R_1''}(X_1'))$ mit jedem mit (t', L') markierten Knoten aus $\text{val}(\circ_{R_2''}(X_2'))$ verbunden wird.

X_1' und X_2' können nach (1.), (2.) oder (3.) aus X_1 und X_2 bestimmt werden.

□

Da die Abschätzung aus Theorem 5.4.2 sehr grob ist, wird diese wahrscheinlich nicht angenommen. Für beliebige Potenzen $d \geq 2$ ist die NLC-Weite ebenfalls beschränkt wie das folgende Lemma zeigt.

Lemma 5.4.3 *Es sei G ein Graph mit NLC-Weite k , dann gilt $\text{NLC-Weite}(G^d) \leq k \cdot 2^{(d-1) \cdot k}$.*

Beweisidee Aus dem NLC-Weite k -Ausdruck X für G wird ein NLC-Weite $k \cdot 2^{(d-1) \cdot k}$ -Ausdruck X' für G^d konstruiert. Jeder Knoten x in G^d wird mit einem d -Tupel (t, L_1, \dots, L_{d-1}) markiert. t ist die Markierung des Knotens x im NLC-Weite k -Ausdruck für G und L_i , $1 \leq i \leq d-1$, ist die Menge der Markierungen der Knoten y in G mit $\text{dist}_G(x, y) = i$. X' kann analog zum Beweis von Lemma 5.4.2 aus X bestimmt werden. □

Eine Analoge Aussagen kann man auch für die Cliquesweite zeigen.

Lemma 5.4.4 *Es sei G ein Graph mit Cliquesweite k , dann gilt $\text{Cliquesweite}(G^2) \leq k \cdot 2^{k+1}$.*

Beweis Da $\text{NLC-Weite}(G) \leq \text{Cliquesweite}(G) = k$ folgt mit Lemma 5.4.2, dass $\text{NLC-Weite}(G^2) \leq k \cdot 2^k$ und da $\text{Cliquesweite}(G) \leq 2 \cdot \text{NLC-Weite}(G)$ folgt $\text{Cliquesweite}(G^2) \leq k \cdot 2^{k+1}$. □

Analog zum Beweis des letzten Lemmas folgt aus Lemma 5.4.3 das nächste Lemma.

Lemma 5.4.5 *Es sei G ein Graph mit Cliquesweite k , dann gilt $\text{Cliquesweite}(G^d) \leq k \cdot 2^{d \cdot k}$.*

Einfügen und Löschen einer Kante

Die NLC-Weite und die Cliquesweite eines Graphen erhöht sich beim Einfügen einer Kante höchstens um 2.

Lemma 5.4.6 (Kanten einfügen) *Es sei $G = (V_G, E_G)$ ein Graph mit NLC-Weite k , $x, y \in V_G$, $\{x, y\} \notin E_G$, dann hat der Graph $G' = (V_G, E_G \cup \{x, y\})$ NLC-Weite höchstens $k + 2$.*

Beweisidee Man kann aus einem NLC-Weite k -Ausdrucksbaum T für G einen NLC-Weite $k + 2$ -Ausdrucksbaum für G' konstruieren, indem man die Blätter u und v in T die den Knoten x bzw. y in G entsprechen mit $k + 1$ bzw. mit $k + 2$ markiert. Auf den Wegen von u und v zur Wurzel von T' werden die beiden Knoten wie zuvor in T behandelt. Dazu müssen Vereinigungsoperationen (d.h. Kanteneinfügungen) an denen u bzw. v beteiligt waren durch die entsprechenden Paare ergänzt werden. Weiterhin muss in der Vereinigungsoperation in der sich die beiden Knoten u und v treffen das Paar $(k + 1, k + 2)$ eingefügt werden. \square

Diese Schranke aus dem letzten Lemma wird auch wirklich angenommen. Der Graph G in Abbildung 5.2 hat NLC-Weite 2, der Graph G' in Abbildung 5.2 hat NLC-Weite 4 und beide Graphen unterscheiden sich genau durch die Kante $e = \{x, y\}$.

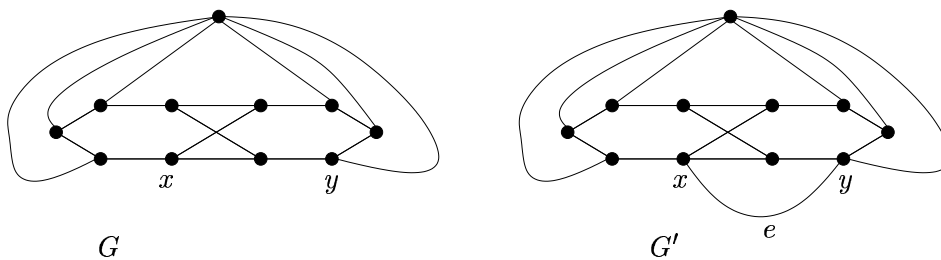


Abbildung 5.2: Der Graph G hat NLC-Weite 2 und der Graph G' mit der zusätzlichen Kante $e = \{x, y\}$ hat NLC-Weite 4.

Wenn man in einem Graphen eine Kante entfernt, so erhöht sich die NLC-Weite des Graphen ebenfalls höchstens um 2.

Lemma 5.4.7 (Kanten löschen) *Es sei $G = (V_G, E_G)$ ein Graph mit NLC-Weite k , $x, y \in V_G$, $\{x, y\} \in E_G$, dann hat der Graph $G' = (V_G, E_G - \{x, y\})$ NLC-Weite höchstens $k + 2$.*

Beweis Man kann wie im Beweis des vorhergehenden Lemmas eine Konstruktion für den Graphen G' , der aus G durch Entfernen der Kante $\{x, y\}$ hervorgeht, angeben.

Oder man kann das Löschen einer Kante durch das Einfügen einer Kante im Komplementgraphen zurückführen. Sei \bar{G} der Komplementgraph von G . In \bar{G} sind die Knoten x und y dann nicht adjazent und die NLC-Weite von \bar{G} ist gleich der NLC-Weite von G , also gleich k . Wenn wir in \bar{G} eine Kante zwischen x und y einfügen, so erhalten wir einen Graphen G'' mit NLC-Weite höchstens $k + 2$. Dann bilden wir den Komplementgraphen von G'' und erhalten den Graphen G' , der mit Ausnahme der Kante $\{x, y\}$, genau die Kanten aus G enthält mit NLC-Weite(G') $\leq k + 2$. \square

Ein analoges Ergebnis gilt auch für die Cliquesweite. Man kann wie im Beweis von Lemma 5.4.6 einen Cliquesweite k -Ausdrucksbaum zu einem Cliquesweite $(k + 2)$ -Ausdrucksbaum für den Graphen mit einer Kante mehr bzw. mit einer Kante weniger modifizieren.

Einfügen eines Knotens

Lemma 5.4.8 *Fügt man in einen Graphen mit NLC-Weite k einen Knoten mit einer beliebigen Nachbarschaft ein, so erhält man einen Graphen mit NLC-Weite höchstens $2k$.*

Beweisidee Es sei G ein Graph mit NLC-Weite k in den wir einen Knoten x einfügen wollen, den so erhaltenen Graphen bezeichnen wir mit G' . Man kann einen NLC-Weite k -Ausdrucksbaum T für G zu einem NLC-Weite $2k$ -Ausdrucksbaum T' für G' modifizieren, indem man die Knoten in G' die Nachbarn von x werden sollen, wie bisher mit den Markierungen $1, \dots, k$ behandelt und die Nichtnachbarn von x mit den Markierungen $k+1, \dots, 2k$ behandelt wie in T . In einem letzten Schritt wird der Knoten x mit allen Knoten die eine Markierung aus $[k]$ besitzen verbunden. \square

Wenn man in einen Graphen mit NLC-Weite 1 einen Knoten hinzufügt, so kann der resultierende Graph NLC-Weite 2 haben. Der Graph H der aus dem Graphen G in Abbildung 5.2 durch Löschen von Knoten x entsteht, hat NLC-Weite 2. Der Graph G' in Abbildung 5.2 kann durch Einfügen eines Knotens in H mit Nachbarschaft $N_G(x) \cup \{y\}$ erhalten werden und hat NLC-Weite 4. Somit wird die Schranke aus Lemma 5.4.8 für $k=1$ und $k=2$ wirklich angenommen.

Eine Analoge Abschätzung kann man auch für die Cliquesweite angeben, man benötigt hier jedoch noch eine Markierung mehr um im letzten Schritt den einzufügenden Knoten zu markieren. Bei dem NLC-Weite Aufbau war dies nicht nötig, da die Kanteneinfügeoperation der NLC-Weite Kanten immer zwischen zwei Graphen einfügt, die Kanteneinfügeoperation der Cliquesweite die Kanten jedoch zwischen den angegebenen Markierungen in einem Graphen einfügt.

Lemma 5.4.9 *Fügt man in einen Graphen mit Cliquesweite k einen Knoten mit einer beliebigen Nachbarschaft ein, so erhält man einen Graphen mit Cliquesweite höchstens $2k+1$.*

Da Graphen mit beschränkter NLC-Weite und Graphen mit beschränkter Cliquesweite abgeschlossen bezüglich induzierten Teilgraphen sind, kann man beliebig viele Knoten aus einem Graphen entfernen ohne dass sich die NLC-Weite bzw. die Cliquesweite erhöht.

Komplementierung der Nachbarschaft eines Knotens (Switching)

Wir betrachten hier die in der Literatur als *switching* bekannte Operation, die in einem Graphen für genau einen Knoten mit Nachbarschaft $N(x)$ alle Kanten von x zu Knoten aus $N(x)$ entfernt und alle Kanten von x zu Knoten aus $V - N(x)$ einfügt, siehe [Sei92].

Lemma 5.4.10 *Es sei $G = (V_G, E_G)$ ein Graph mit NLC-Weite k , $x \in V_G$, dann hat der Graph $G' = (V_G, E_G - \{\{x, y\} \mid y \in V_G \wedge \{x, y\} \in E_G\} \cup \{\{x, y\} \mid y \in V_G \wedge \{x, y\} \notin E_G\})$ NLC-Weite höchstens $k+1$.*

Beweisidee Aus einer Konstruktion für G mit k Markierungen kann man eine Konstruktion für G' mit $k+1$ Markierungen angeben. Dazu verändert man den NLC-Weite k -Ausdrucksbaum T zu G zu einem NLC-Weite $k+1$ -Ausdrucksbaum zu G' folgendermaßen. Das Blatt u in T das dem Knoten x entspricht wird mit $k+1$ markiert. Weiterhin wird auf dem Weg in T von u zur Wurzel von T jede \times_S Operation in einem Knoten v in T folgendermaßen modifiziert. Sei v_1 der linke Sohn von v und v_2 der rechte Sohn von v in T . Ist u im Teilbaum $T(v_1)$ enthalten, so wird S durch $S \cup \{(k+1, l) \mid (\text{lab}_{G(v)}(u), l) \notin S, l \in [k]\}$ ersetzt. $\text{lab}_{G(v)}(u)$

bezeichne dabei die Markierung des Knotens u im Graphen $G(v)$. Ist u im Teilbaum $T(v_2)$ enthalten, so wird S entsprechend durch $S \cup \{(l, k+1) \mid (l, \text{lab}_{G(v)}(u)) \notin S, l \in [k]\}$ ersetzt. \square

Es sei G der linke Graph in Abbildung 5.3 mit NLC-Weite 1. Komplementiert man die Nachbarschaft des Knotens x , so erhält man den P_4 mit NLC-Weite 2, siehe Abbildung 5.3 rechts.

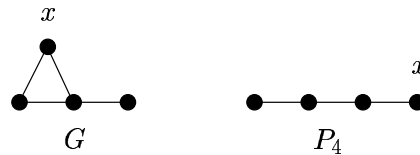


Abbildung 5.3: Komplementiert man die Nachbarschaft des Knotens x im Graphen G mit NLC-Weite 1, so erhält man den P_4 mit NLC-Weite 2.

Betrachtet man das Verhalten der Cliquesweite eines Graphen G beim Komplementieren der Nachbarschaft eines Knotens in G , so kann man aufgrund der Kanteneinfügeoperation bei der Cliquesweite nicht wie im Beweis von Lemma 5.4.10 mit einer Hilfsmarkierung einen Cliquesweite Ausdruck für den entstehenden Graphen angeben. Man kann jedoch den Knoten dessen Nachbarschaft man Komplementieren will zuerst aus dem Graphen entfernen und den Knoten dann mit der neuen Nachbarschaft einfügen und man erhält mit Lemma 5.4.9 die folgende Cliquesweiteabschätzung.

Lemma 5.4.11 *Es sei $G = (V_G, E_G)$ ein Graph mit Cliquesweite k , $x \in V_G$, dann hat der Graph $G' = (V_G, E_G - \{\{x, y\} \mid y \in V_G \wedge \{x, y\} \in E_G\} \cup \{\{x, y\} \mid y \in V_G \wedge \{x, y\} \notin E_G\})$ Cliquesweite höchstens $2k + 1$.*

5.5 Abschätzungen der NLC-Weite und Cliquesweite beliebiger Graphen

In diesem Abschnitt werden obere Schranken für die NLC-Weite und die Cliquesweite eines Graphen $G = (V_G, E_G)$ mit Hilfe der Knotenanzahl und Kantenanzahl angeben.

Eine Schranke die für jeden Graphen anhand der Knotenanzahl oder Kantenanzahl seine NLC-Weite oder Cliquesweite exakt berechnet kann es nicht geben. Ein Graph mit n Knoten kann NLC-Weite 1 (Cliquesweite 1) aber auch NLC-Weite $\lceil \frac{n}{2} \rceil$ (Cliquesweite n) haben, ebenso kann die NLC-Weite (Cliquesweite) bei m Kanten 1 (2) sein, im Allgemeinen jedoch nicht durch eine Konstante beschränkt werden.

Man kann jeden Graphen G mit $|V_G|$ vielen Knotenmarkierungen sowohl mit den NLC-Weite als auch mit den Cliquesweite Operationen aufbauen, indem jeder Knoten eine unterschiedliche Markierung erhält und man die Kanten gemäß der vorhandenen Adjazenzen einfügt. Ummarkierungsoperationen sind in diesem Aufbau nicht notwendig.

Johansson hat in [Joh98] gezeigt, dass man im diese triviale Schranke im NLC-Weite Modell halbieren kann, indem man den Graphen G in zwei Graphen der Größe $\lceil \frac{|V_G|}{2} \rceil$ bzw. $\lfloor \frac{|V_G|}{2} \rfloor$ aufteilt. Diese beiden Graphen können beide mit höchstens $\lceil \frac{|V_G|}{2} \rceil$ Markierungen aufgebaut werden und mittels einer \times_S Operation zu G verknüpft werden, da man im NLC-Weite Modell beim Kanteneinfügen zwischen den Markierungen eines linken und eines rechten Graphen

unterscheidet. Damit gilt für jeden Graphen G

$$\text{NLC-Weite}(G) \leq \lceil \frac{|V_G|}{2} \rceil. \quad (5.3)$$

Nach Lemma 5.4.6 erhöht sich die NLC-Weite eines Graphen beim Einfügen einer zusätzlichen Kante höchstens um 2. Da Graphen mit drei Kanten NLC-Weite höchstens 2 haben, gilt für Graphen mit mindestens drei Kanten die folgende Abschätzung.

$$\text{NLC-Weite}(G) \leq 2 \cdot |E_G| - 4 \quad (5.4)$$

Da die Definition der NLC-Weite die disjunkte Vereinigung (\times_{\emptyset}) zweier Graphen erlaubt, kann man sich bei der Bestimmung der NLC-Weite eines Graphen G auf dessen Zusammenhangskomponenten $\text{ZK}(G)$ beschränken und es gilt

$$\text{NLC-Weite}(G) = \max_{G_i \in \text{ZK}(G)} \text{NLC-Weite}(G_i) \quad (5.5)$$

Ein zusammenhängender Graph mit n Knoten hat mindestens $n - 1$ Kanten. Damit ist die Abschätzung (5.4) nie besser als die Abschätzung (5.3). Für Graphen mit höchstens sieben Knoten kann man die Abschätzung 5.3 wie folgt verbessern.

Lemma 5.5.1 *Es sei $G = (V_G, E_G)$ ein Graph.*

1. $|V_G| \leq 3 \Rightarrow \text{NLC-Weite}(G) = 1$
2. $|V_G| = 4 \Rightarrow \text{NLC-Weite}(G) = \begin{cases} 1, & \text{falls } G \not\approx P_4 \\ 2, & \text{falls } G \approx P_4 \end{cases}$
3. $|V_G| \leq 6 \Rightarrow \text{NLC-Weite}(G) \leq 2$
4. $|V_G| \leq 7 \Rightarrow \text{NLC-Weite}(G) \leq 3$

Beweis Eine Übersicht aller Graphen mit höchstens sechs Knoten findet man zum Beispiel in [Har69].

- 1.-3. Alle Graphen mit höchstens vier Knoten, mit Ausnahme des P_4 , sind Cographen. Für alle Graphen mit höchstens 6 Knoten gibt es einen NLC-Weite 2-Ausdruck.
4. In jedem Graphen G mit 7 Knoten gibt es immer einen Knoten v mit $\deg_G(v) \in \{0, 1, 2, 4, 5, 6\}$, da nicht alle Knoten den Grad 3 haben können. Da $\text{NLC-Weite}(G) = \text{NLC-Weite}(\overline{G})$, siehe Kapitel 5.4, gilt, können wir annehmen, dass es einen Knoten v in G gibt mit $\deg_G(v) \in \{0, 1, 2\}$, sonst betrachten wir den Komplementgraphen \overline{G} von G und zeigen, dass \overline{G} NLC-Weite höchstens 3 hat.

Es sei G' der Graph G ohne den Knoten v und der zu v in G inzidenten Kanten. Wir können wie folgt einen NLC-Weite 3-Ausdruck X' für G' angeben und mit diesem einen NLC-Weite 3-Ausdruck X für G angeben.

- (a) Falls $\deg_G(v) = 0$, so gibt es, da G' genau 6 Knoten enthält, nach (3.) einen NLC-Weite 2-Ausdruck X' für G' .

$$X = \bullet_{1,(v)} \times_{\emptyset} X'$$

- (b) Falls $\deg_G(v) = 1$, so kann der Graph G' nach (3.) mit einem NLC-Weite 2-Ausdruck X'' aufgebaut werden. Wir modifizieren X'' zu einem NLC-Weite 3-Ausdruck X' , indem wir den Knoten aus G' der in G zu v adjazent ist mit der Markierung 3 versehen, ihn bezüglich der Kanteneinfügungen wie bisher behandeln, und die restlichen Knoten in X' wie im NLC-Weite 2-Ausdruck X'' mit Markierung 1 bzw. 2 markieren.

$$X = \bullet_{1,(v)} \times_{\{(1,3)\}} X'$$

- (c) Falls $\deg_G(v) = 2$, dann seien v_1, v_2 die beiden Knoten aus G' die in G zu v adjazent sind. Wir teilen $V_{G'}$ in zwei Mengen V_1, V_2 mit je drei Knoten auf, so dass $v_i \in V_i, i = 1, 2$. Es seien X_1 und X_2 ein NLC-Weite 3-Ausdruck für den durch V_1 bzw. V_2 induzierten Teilgraph von G' , so dass der Knoten v_i in $\text{val}(X_i), i = 1, 2$, enthalten ist und mit 1 markiert ist und die anderen beiden Knoten in X_i mit 2 bzw. 3 markiert sind. Aus X_1 und X_2 kann man mittels einer Vereinigungsoperation $X_1 \times_S X_2$ einen NLC-Weite 3-Ausdruck X' für G' aufbauen.

$$X = \bullet_{1,(v)} \times_{\{(1,1)\}} X'$$

Der Kreis mit sieben Knoten C_7 hat die NLC-Weite 3.

□

Für die Cliquesweite eines Graphen kann man die triviale Schranke der Knotenanzahl nicht stark verbessern. Johansson zeigt in [Joh98], dass für Graphen mit n Knoten $\text{Cliquesweite}(G) \leq n - k$, falls $2^k + k \leq n - k$. Das folgende Lemma verbessert diese Abschätzung.

Lemma 5.5.2 *Es sei $G = (V, E)$ ein Graph mit n Knoten.*

$$\text{Cliquesweite}(G) \leq n - k, \text{ falls } 2^k < n - k \quad (5.6)$$

Beweis Es sei $G = (V, E)$ ein Graph und $V_1 \cup V_2$ sei eine disjunkte Partition der Knotenmenge V in zwei Mengen der Größe $n - k$ bzw. k . Es seien G_1 und G_2 die durch die Mengen V_1 bzw. V_2 induzierten Teilgraphen von G . Man kann einen Graphen $J_1 \in \text{CW}_{n-k}$ konstruieren in dem jeder Knoten unterschiedlich mit einer Zahl aus $\{1, \dots, n - k\}$ markiert ist, so dass $G_1 = \text{unlab}(J_1)$ gilt.

Die Knoten aus V_1 können in höchstens 2^k Nachbarschaftsklassen bezüglich der Knoten aus V_2 eingeteilt werden. Zwei Knoten $u, v \in V_1$ gehören dabei zur gleichen Nachbarschaftsklasse, falls u und v die gleichen Nachbarn in der Menge V_2 haben. Wir markieren die Knoten in J_1 so um dass jede dieser Nachbarschaftsklassen mit einer Zahl aus $\{1, \dots, 2^k\}$ markiert ist. Dies ist möglich, falls $n - k \geq 2^k$.

Nun werden die Knoten aus G_2 bezüglich ihrer Nachbarschaft in V_1 in die Konstruktion aufgenommen. Ein beliebiger Knoten v aus G_2 wird mit $2^k + 1$ markiert. Dies ist möglich, falls $n - k \geq 2^k + 1$. Dann wird v mit seiner Nachbarschaft in J_1 verbunden. Nun können wir die Knoten in J_1 ummarkieren. Es werden nur noch höchstens 2^{k-1} Markierungen für höchstens 2^{k-1} Nachbarschaftsklassen in J_1 bezüglich Knoten aus $V_2 - \{v\}$ benötigt. Somit haben wir 2^{k-1} freie Markierungen von denen wir eine nutzen um den nächsten Knoten aus G_2 zu markieren und diesen mit Knoten aus J_1 und Knoten aus G_2 zu verbinden die wir schon in unsere Konstruktion eingefügt haben, falls es eine solche Verbindung in G gibt. Nachdem alle

Knoten aus V_2 in dem Cliquesweite Aufbau eingefügt sind, haben wir einen Graphen J mit $G = \text{unlab}(J)$ konstruiert. \square

Aus Schranke (5.6) folgt, dass für Graphen mit $n \geq 4$ Knoten die Cliquesweite höchstens $n - 1$ beträgt, für $n \geq 7$ ist die Cliquesweite höchstens $n - 2$, für $n \geq 12$ ist die Cliquesweite höchstens $n - 3$.

Da sich die Cliquesweite eines Graphen beim Einfügen einer zusätzlichen Kante um höchstens 2 erhöht und Graphen mit drei Kanten Cliquesweite höchstens 3 haben, gilt für Graphen mit mindestens drei Kanten die folgende Abschätzung.

$$\text{Cliquesweite}(G) \leq 2 \cdot |E_G| - 3 \quad (5.7)$$

Die Definition der Cliquesweite erlaubt die disjunkte Vereinigung (\oplus) zweier Graphen. Somit kann man sich auch bei der Bestimmung der Cliquesweite eines Graphen G auf dessen Zusammenhangskomponenten $\text{ZK}(G)$ beschränken und es gilt

$$\text{Cliquesweite}(G) = \max_{G_i \in \text{ZK}(G)} \text{Cliquesweite}(G_i) \quad (5.8)$$

Da ein zusammenhängender Graph mit n Knoten mindestens $n - 1$ Kanten hat, ist die Abschätzung (5.7) nie besser als die Abschätzung (5.6). Mit Hilfe von Konstruktionen aller möglicher Graphen mit der angegebenen Knotenanzahl findet man folgende Aussagen.

Lemma 5.5.3 *Es sei $G = (V_G, E_G)$ ein Graph.*

1. $|V_G| = 1 \Rightarrow \text{Cliquesweite}(G) = 1$
2. $|V_G| \leq 3 \Rightarrow \text{Cliquesweite}(G) \leq 2$
3. $|V_G| = 4 \Rightarrow \text{Cliquesweite}(G) \leq \begin{cases} 2, & \text{falls } G \not\approx P_4 \\ 3, & \text{falls } G \approx P_4 \end{cases}$
4. $|V_G| \leq 5 \Rightarrow \text{Cliquesweite}(G) \leq 3$
5. $|V_G| = 6 \Rightarrow \text{Cliquesweite}(G) \leq \begin{cases} 3, & \text{falls } G \not\approx \overline{C_6} \\ 4, & \text{falls } G \approx \overline{C_6} \end{cases}$

Die Betrachtungen von zufälligen Graphen von Johansson in [Joh98] zeigen, dass die Schranken für die NLC-Weite (5.3) und für die Cliquesweite (5.6) eines Graphen im Allgemeinen optimal sind.

Theorem 5.5.4 (Johansson [Joh98]) *Es sei $0 < p < 1$, $0 < \alpha < 1$ und $G(n, p)$ sei ein Graph mit n Knoten mit Kantenwahrscheinlichkeit p . $\text{Pr}[x]$ bezeichne die Wahrscheinlichkeit eines Ereignisses x .*

$$\lim_{n \rightarrow \infty} \text{Pr}[\text{NLC-Weite}(G(n, p)) \leq \frac{\alpha n}{2}] = 0$$

$$\lim_{n \rightarrow \infty} \text{Pr}[\text{Cliquesweite}(G(n, p)) \leq \alpha n] = 0$$

Kapitel 6

Cliquenweite von Graphen mit beschränkter Baumweite

Im ersten Abschnitt dieses Kapitels geben wir einen Überblick über die bisher bekannten Methoden zur Bestimmung der Cliquenweite und NLC-Weite beliebiger Graphen an. Im zweiten Abschnitt werden bekannte Ergebnisse über das Lösen von Graphenproblemen auf Graphen mit beschränkter Baumweite angegeben. Im dritten Abschnitt definieren wir eine Normalform für Cliquenweite k -Ausdrücke. In den folgenden Abschnitten zeigen wir, dass man auf Graphklassen mit beschränkter Baumweite für jedes feste k in linearer Zeit entscheiden kann, ob ein gegebener Graph Cliquenweite höchstens k hat. Aus diesem Ergebnis folgt auch ein Linearzeitalgorithmus zur Berechnung der Cliquenweite für Graphklassen mit beschränkter Baumweite. Da unser Beweis technisch sehr aufwendig ist, geben wir im vierten Abschnitt die Grundidee für unseren Beweis an.

6.1 Erkennungsprobleme

Das folgende Problem wird als *Erkennungsproblem* für festes $k \in \mathbb{N}$ für Graphen mit beschränkter NLC-Weite bezeichnet.

Problem 6.1.1 (NLC-Weite höchstens k)

EINGABE: Ein Graph $G = (V, E)$

FRAGE: $NLC\text{-Weite}(G) \leq k$?

Für $k = 1$ und $k = 2$ ist das Problem 6.1.1 effizient gelöst. „NLC-Weite höchstens 1“ entspricht dem Erkennungsproblem für Cographen und ist in Linearzeit entscheidbar, siehe [CPS84]. Johansson hat in [Joh00] gezeigt, dass „NLC-Weite höchstens 2“ auf Graphen mit n Knoten in Zeit $O(n^4 \log n)$ entscheidbar ist. Die beiden Algorithmen finden sogar einen NLC-Weite k -Ausdruck, falls der gegebene Graph NLC-Weite höchstens 1 bzw. NLC-Weite höchstens 2 hat. Ein k -Ausdruck ist für Anwendungen, wie zum Beispiel das Lösen von schweren Graphenproblemen auf Graphen mit beschränkter NLC-Weite von Bedeutung, vergleiche Kapitel 7. Für $k \geq 3$ ist die Komplexität von Problem 6.1.1 bisher noch offen.

Johansson hat einen Approximationsalgorithmus entwickelt mit dem man einen NLC-Weite $(\log n \cdot k)$ -Ausdruck in der Zeit $O(n^{2 \cdot k + 1})$ für Graphen mit NLC-Weite k mit n Knoten bestimmen kann, siehe [Joh01]. Der Approximationsalgorithmus hat für festes k eine polyno-

mielle Laufzeit, liefert aber wie es bei Approximationsalgorithmen üblich ist nicht immer das exakte Ergebnis.

Ein NLC-Weite k -Ausdruck, der keine Ummarkierungsoperationen enthält, kann für einen gegebenen markierten Graphen $G = (V, E, \text{lab})$ in polynomieller Zeit berechnet werden, falls ein solcher Ausdruck für G existiert. Der entsprechende Erkennungsalgorithmus sucht nach einer möglichen zuletzt angewendeten \times_S -Operation. Für jede Relation $S \subseteq [k]^2$ wird die Knotenmenge V systematisch disjunkt in zwei Mengen V_1 und V_2 aufgeteilt. Gestartet wird mit einer beliebigen einelementigen Menge V_1 . Anschließend wird ein Knoten v_2 aus V_2 entfernt und in V_1 eingefügt, falls V_1 einen Knoten v_1 enthält mit $(\text{lab}(v_1), \text{lab}(v_2)) \in S$ und $\{v_1, v_2\} \notin E$ bzw. $(\text{lab}(v_1), \text{lab}(v_2)) \notin S$ und $\{v_1, v_2\} \in E$. Wird eine Aufteilung gefunden, so ist $G = G|_{V_1} \times_S G|_{V_2}$, und die Suche kann mit den durch die Knotenmengen V_1 und V_2 induzierten Teilgraphen $G|_{V_1}$ bzw. $G|_{V_2}$ fortgesetzt werden.

Analog definiert man das Erkennungsproblem für festes $k \in \mathbb{N}$ für Graphen mit beschränkter Cliquesweite.

Problem 6.1.2 (Cliquesweite höchstens k)

EINGABE: Ein Graph $G = (V, E)$.

FRAGE: $\text{Cliquesweite}(G) \leq k$?

Für $k = 1$, $k = 2$ und $k = 3$ ist Problem 6.1.2 effizient gelöst. Die Menge der Graphen mit Cliquesweite 1 besteht nur aus vollständig unzusammenhängenden Graphen, womit „Cliquesweite höchstens 1“ leicht in Linearzeit entscheidbar ist. „Cliquesweite höchstens 2“ ist in Linearzeit entscheidbar, da die Menge der Graphen mit Cliquesweite höchstens 2 gleich der Menge der Cographen ist, siehe [CPS84]. „Cliquesweite höchstens 3“ ist für Graphen mit n Knoten und m Kanten in Zeit $O(n^2 \cdot m)$ entscheidbar, siehe [CHL⁺00]. Die Algorithmen finden auch einen Cliquesweite k -Ausdruck falls der gegebene Graph Cliquesweite höchstens 1, Cliquesweite höchstens 2 bzw. Cliquesweite höchstens 3 hat. Für $k \geq 4$ ist die Komplexität von Problem 6.1.2 noch offen.

Mit dem Approximationsalgorithmus von Johansson [Joh01] für das Problem „NLC-Weite höchstens k “ folgt auch für die Cliquesweite ein Approximationsalgorithmus. Einen Cliquesweite $(2 \log n \cdot k)$ -Ausdruck kann in Zeit $O(n^{2 \cdot k+1})$ für Graphen mit Cliquesweite k mit n Knoten bestimmt werden, siehe [Joh01].

Algorithmen mit exponentieller Laufzeit zum Lösen der Probleme 6.1.1 und 6.1.2 können durch testen aller möglichen Dekompositionen des gegebenen Graphen gefunden werden. Eine Implementierung ist aber aufgrund der Laufzeit nur für Graphen mit höchstens 20 Knoten praktisch anwendbar.

Falls wir in der Eingabe einen Graphen gegeben haben und das minimale k suchen, so dass der gegebene Graph NLC-Weite höchstens k bzw. Cliquesweite höchstens k hat, so erhalten wir die folgenden bisher noch nicht effizient gelösten *Minimierungsprobleme*.

Problem 6.1.3 (NLC-Weite)

EINGABE: Ein Graph $G = (V, E)$

GESUCHT: $\min_k : \text{NLC-Weite}(G) \leq k$

Problem 6.1.4 (Cliquesweite)**EINGABE:** Ein Graph $G = (V, E)$ **GESUCHT:** $\min_k : \text{Cliquesweite}(G) \leq k$

Da jeder Graph G NLC-Weite höchstens $\lceil \frac{|V_G|}{2} \rceil$ (Cliquesweite höchstens $|V_G|$) hat, kann man durch testen von „NLC-Weiten höchstens k “, $k = 1, \dots, \lceil \frac{|V_G|}{2} \rceil$ („Cliquesweiten höchstens k “, $k = 1, \dots, |V_G|$) mit einem exponentiellen Algorithmus, auch für das Problem der Berechnung der NLC-Weite (Cliquesweite) einen Algorithmus mit exponentieller Laufzeit angeben.

6.2 Ersetzbarkeit von k -markierten l -terminalen Graphen

In diesem Abschnitt werden die wichtigsten Ergebnisse über das Lösen schwerer Graphenprobleme auf Graphklassen mit beschränkter Baumweite zusammengefasst.

Die meisten Bottom-Up Algorithmen zur Entscheidung einer Grapheigenschaft Π auf einem baumstrukturierten Graphen G basieren auf der Idee einen Teilgraphen von G durch einen kleineren sogenannten *ersetzbaren* Teilgraphen zu ersetzen. Die Ersetzung wird durch einen Verknüpfungsmechanismus definiert, der für verschiedene Graphmodelle unterschiedlich definiert ist. Der Begriff der *Ersetzbarkeit* kann für jeden Verknüpfungsmechanismus definiert werden. Damit funktionieren die dynamischen Bottom-Up Techniken im Prinzip für alle baumstrukturierten Graphen mehr oder weniger gut. Zur Untersuchung von baumweitebeschränkten Graphen benötigen wir sogenannte *l -terminale Graphen* und eine Operation die wir mit \circ bezeichnen, die zwei l -terminale Graphen durch Identifizierung von Knoten kombiniert. Da wir uns hier hauptsächlich für markierte Graphen interessieren, definieren wir k -markierte l -terminale Graphen. Terminale Graphen werden auch *sourced graphs* genannt, vergleiche [ALS91].

Definition 6.2.1 (k -markierter l -terminaler Graph) Ein k -markierter l -terminaler Graph ist ein Tupel

$$G = (V_G, E_G, P_G, \text{lab}_G),$$

wobei (V_G, E_G, lab_G) ein k -markierter Graph und $P_G = (x_1, \dots, x_l)$ eine Liste von $l \geq 0$ paarweise verschiedenen Knoten aus V_G ist. Die Knoten in P_G heißen terminale Knoten oder kurz Terminale. Knoten x_i , $1 \leq i \leq l$, heißt i -ter terminaler Knoten von G . Die restlichen Knoten in $V_G - P_G$ heißen innere Knoten von G .

Es seien $H = (V_H, E_H, P_H, \text{lab}_H)$ und $J = (V_J, E_J, P_J, \text{lab}_J)$ zwei knotendisjunkte k -markierte l -terminale Graphen, so dass der i -te terminale Knoten aus P_H , für $i = 1, \dots, l$, die gleiche Markierung hat wie der i -te terminale Knoten aus P_J . Der Graph

$$H \circ J$$

entsteht aus der disjunkten Vereinigung der k -markierten Graphen (V_H, E_H, lab_H) und (V_J, E_J, lab_J) in der die entsprechenden terminalen Knoten identifiziert werden, d.h., für $i = 1, \dots, l$ wird der i -te terminale Knoten aus H mit dem i -ten terminalen Knoten aus J identifiziert und Mehrfachkanten werden gelöscht.

Definition 6.2.2 (Ersetzbarkeit von k -markierten l -terminalen Graphen) Es sei Π eine Grapheigenschaft, d.h., $\Pi : \mathcal{G}_k \rightarrow \{0, 1\}$, wobei \mathcal{G}_k die Menge aller k -markierter Graphen

ist. Zwei k -markierte l -terminale Graphen H_1 und H_2 heißen ersetzbar bezüglich Π , dies bezeichnen wir mit $H_1 \sim_{\Pi, l} H_2$, falls für alle k -markierten l -terminalen Graphen J gilt:

$$\Pi(H_1 \circ J) = \Pi(H_2 \circ J)$$

Die Relation $\sim_{\Pi, l}$ ist eine Äquivalenzrelation. Abbildungen 6.1 und 6.2 zeigen drei 4-markierte 3-terminale Graphen H_1 , H_2 und J . Die Graphen H_1 und H_2 sind bezüglich der Grapheigenschaft $\Pi : \Pi(G) = G$ hat einen Hamilton Kreis (zur Problemdefinition vergleiche Kapitel 7.1), nicht ersetzbar. Der Graph $H_1 \circ J$ besitzt einen Hamilton Kreis und der Graph $H_2 \circ J$ enthält keinen Hamilton Kreis.

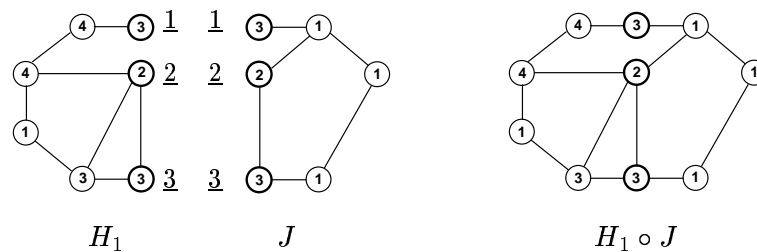


Abbildung 6.1: Zwei 4-markierte 3-terminale Graphen H_1 , J und der verknüpfte 4-markierte Graph $H_1 \circ J$. Die unterstrichenen Zahlen repräsentieren die Nummerierung der terminalen Knoten.

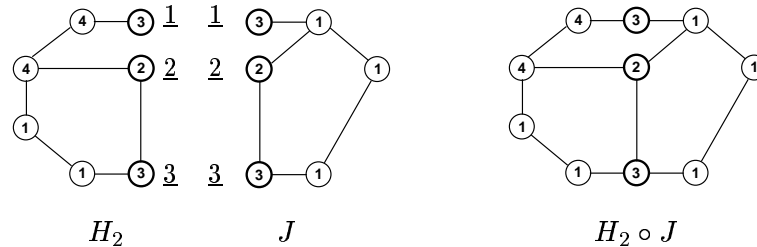


Abbildung 6.2: Zwei 4-markierte 3-terminale Graphen H_2 , J und der verknüpfte 4-markierte Graph $H_2 \circ J$.

Wir wollen im Weiteren die Grapheigenschaften Π betrachten für die die Relation $\sim_{\Pi, l}$ für jedes $l \geq 0$ endlich viele Äquivalenzklassen hat.

Definition 6.2.3 (erkennbar) Eine Grapheigenschaft Π heißt erkennbar, falls die Relation $\sim_{\Pi, l}$ für jedes $l \geq 0$ einen endlichen Index hat, d.h. falls die Relation $\sim_{\Pi, l}$ für jede natürliche Zahl $l \geq 0$ endlich viele Äquivalenzklassen hat.

Falls Π erkennbar ist, dann ist Π für festes l in linearer Zeit für alle k -markierten Graphen mit Baumweite höchstens l mit Hilfe von dynamischer Programmierung entscheidbar,

vergleiche [Cou90b]. Die Eingabe für den Entscheidungsalgorithmus besteht aus einer Baumdekomposition (\mathcal{X}, T) der Weite l mit einem binären Dekompositionsbaum T mit ausgezeichneter Wurzel. Der Algorithmus berechnet Bottom-Up mit dynamischer Programmierung für jeden Teilbaum von T mit Wurzel u in konstanter Zeit die Äquivalenzklasse des zugehörigen l -terminalen Graphen aus den Äquivalenzklassen der Teilbäume der zwei Söhne von u .

Die wichtigste Voraussetzung für eine Bottom-Up Verarbeitung ist somit die Existenz der Baumdekomposition. Arnborg, Corneil und Proskurowski zeigten in [ACP87], dass die Frage, gegeben ein Graph G und eine Zahl k , hat G die Baumweite k , ein NP-vollständiges Entscheidungsproblem ist. Für konstante Werte von k existieren jedoch polynomielle Algorithmen, die eine Baumdekomposition berechnen, falls es eine solche gibt. Der asymptotisch schnellste sequentielle Algorithmus für die Berechnung einer Baumdekomposition für festes k ist der Linearzeit-Algorithmus von Bodlaender in [Bod96]. Die niedrigste bekannte Komplexitätsklasse, welche die Menge aller Graphen mit Baumweite k für festes k enthalten, die eine MSO_2 -definierbare Eigenschaft erfüllen, ist die Komplexitätsklasse LOGCFL, siehe [Wan94a].

Das gleiche Schema kann genutzt werden um Graphenprobleme auf Graphen mit beschränkter NLC-Weite zu lösen, siehe Kapitel 7.2. Näheres zur Lösung von schweren Graphenproblemen auf Graphklassen mit beschränkter Baumweite und auf Graphklassen mit beschränkter NLC-Weite findet man in den Schriften [Arn85, AP89, ALS91, Bod97, Bod98, CMR00, Cou90a, Cou90b, Cou92, EGW01a, KR01, LW93, Wan94c].

Es gibt eine ganze Klasse von erkennbaren Graphenproblemen die man mittels Ausdrucksmitteln aus der Logik beschreiben kann. Man unterscheidet die Monadische Logik zweiter Ordnung für Graphen mit Quantifizierungen über Knoten und Knotenmengen, kurz MSO_1 ¹ und die ausdrucksstärkere Monadische Logik zweiter Ordnung für Graphen mit Quantifizierungen über Knoten, Knotenmengen, Kanten und Kantenmengen, kurz MSO_2 . Näheres zur Monadischen Logik für Graphen findet man in den Schriften [Cou90a, CM93]. Es sei $\Pi : \mathcal{G}_k \rightarrow \{0, 1\}$ eine Grapheigenschaft. Falls es eine Formel mit den Operationen der Monadischen Logik mit Quantifizierungen gemäss MSO_1 bzw. MSO_2 gibt, so dass ein Graph G genau dann Π erfüllt, wenn G die Formel erfüllt, so nennen wir Π *MSO_1 -definierbar* bzw. *MSO_2 -definierbar*.

Die Eigenschaft Π ob ein Graph einen Hamilton Kreis besitzt ist wie folgt MSO_2 -definierbar. $\Pi(G) \Leftrightarrow \exists E' \subseteq E_G : (V_G, E')$ ist ein Kreis. Ein Graph ist ein Kreis genau dann wenn der Graph zusammenhängend ist und jeder Knoten den Grad zwei hat. Zwei Knoten u und v eines Graphen G sind adjazent ($\text{adj}(u, v)$) genau dann wenn $\exists e \in E : u \in e \wedge v \in e$. $\text{deg}(u) \geq k \Leftrightarrow \exists u_1, \dots, u_k \in V : u_1 \neq u_2, u_1 \neq u_3, \dots, u_1 \neq u_k, \dots, u_{k-1} \neq u_k \wedge \text{adj}(u, u_1) \wedge \dots \wedge \text{adj}(u, u_k)$. $\text{deg}(u) = k \Leftrightarrow \text{deg}(u) \geq k \wedge \neg(\text{deg}(u) \geq k + 1)$. G ist zusammenhängend $\Leftrightarrow \forall U \subseteq V_G$ mit $1 \leq |U| < |V_G| : \exists u \in U \wedge \exists v \in V_G - U \wedge \exists e \in E$ mit $u \in e$ und $v \in e$. $|U| < |V| : \exists w \in V : \neg(w \in U)$. Π ist nicht MSO_1 -definierbar, siehe Kapitel 7.11.

Theorem 6.2.4 (Courcelle [Cou90b]) *Jede MSO_2 -definierbare Grapheigenschaft ist erkennbar.*

Die Lösungsidee zu Theorem 6.2.4 beruht darauf, dass Teilgraphen, welche durch Teilbäume des Dekompositionsbaumes des Eingabegraphen definiert sind, sich bezüglich der zu untersuchenden Eigenschaft in Äquivalenzklassen einteilen lassen. Für alle MSO_2 -definierbaren Eigenschaften ist die Anzahl der Äquivalenzklassen unabhängig von der Teilgraphgröße. Dadurch lassen sich mit Bottom-Up Analysemethoden alle MSO_2 -definierbaren Eigenschaften auf Graphen mit beschränkter Baumweite in linearer Zeit entscheiden. Zum

¹Die Abkürzung MSO steht für Monadic Second Order Logic.

Beispiel ist das MSO_2 -definierbare und im allgemeinen NP-vollständige Graphenproblem Hamiltonkreis in linearer Zeit auf Graphen mit beschränkter Baumweite lösbar.

Die Umkehrung von Theorem 6.2.4 gilt jedoch nicht, da nicht jede erkennbare Grapheneigenschaft auch MSO_2 -definierbar ist. Zum Beispiel ist die Menge aller Graphen mit einer geraden Anzahl paarweise nicht adjazenter Knoten ($\{\overline{K_{2n}} \mid n \in \mathbb{N}\}$) erkennbar, aber nicht MSO_2 -definierbar, siehe [Cou90b]. Die Äquivalenz zwischen Erkennbarkeit und MSO_2 -Definierbarkeit gilt jedoch für Graphklassen mit beschränkter Baumweite, siehe [Lap98].

Erweiterungen der MSO_2 Logik erlauben sogar die Formulierung von Optimierungsproblemen die ebenfalls erkennbar sind, genaueres dazu findet man in den Schriften [ALS91, BPT92, CM93]. Zum Beispiel sind die NP-schweren Probleme ein minimales maximales Matching in einem Graphen zu finden (siehe Kapitel 7.10 zur Problemdefinition) und einen minimalen t -Spanner in einem Graphen zu finden in der erweiterten MSO_2 Logik definierbar und somit in linearer Zeit auf Graphen mit beschränkter Baumweite lösbar.

6.3 Normalform für Cliquesweite k -Ausdrücke

Wir definieren eine *Normalform* für die in Definition 2.3.1 eingeführten Cliquesweite k -Ausdrücke. Diese Normalform ändert die Graphen, die man mittels Cliquesweite k -Ausdrücken definieren kann nicht, sondern ist nur für einige Beweise hilfreich.

Um die Definition der Normalform möglichst einfach zu gestalten nummerieren wir die Knoten in einem Graphen $G = \text{val}(X)$ für einen Cliquesweite k -Ausdruck X wie folgt. Falls $G = \text{val}(\bullet_t)$, dann ist der einzige Knoten aus G der *erste Knoten* in G . Es sei $G = \text{val}(Y_1 \oplus Y_2)$. Wenn $\text{val}(Y_1)$ n Knoten und $\text{val}(Y_2)$ m Knoten enthält, dann ist der i -te Knoten von G der i -te Knoten aus $\text{val}(Y_1)$, falls $i \leq n$ und der $(i - n)$ -te Knoten aus $\text{val}(Y_2)$ falls $i > n$. Der i -te Knoten von $\text{val}(\eta_{i,j}(Y))$ und von $\text{val}(\rho_{i \rightarrow j}(Y))$ ist der i -te Knoten aus $\text{val}(Y)$. Zwei k -Ausdrücke X und Y heißen *äquivalent*, dies bezeichnen wir mit $X \equiv Y$, falls die beiden Graphen $\text{val}(X)$ und $\text{val}(Y)$ bezüglich der Knotennummerierung isomorph sind, d.h.

1. $\text{val}(X)$ und $\text{val}(Y)$ haben die gleiche Knotenanzahl n ,
2. der i -te Knoten in $\text{val}(X)$, $1 \leq i \leq n$, hat die gleiche Markierung wie der i -te Knoten in $\text{val}(Y)$ und
3. es gibt eine Kante zwischen dem i -ten und dem j -ten Knoten in $\text{val}(X)$, $1 \leq i, j \leq n$, genau dann wenn es eine Kante zwischen dem i -ten und j -ten Knoten in $\text{val}(Y)$ gibt.

Anderenfalls sind X und Y *nicht äquivalent*, was wir mit $X \not\equiv Y$ bezeichnen.

Wenn zwei k -Ausdrücke X und Y äquivalent sind, dann müssen diese nicht gleich sein. Sind zwei k -markierte Graphen $\text{val}(X)$ und $\text{val}(Y)$, die durch die beiden k -Ausdrücke X und Y definiert sind, isomorph, dann müssen die Ausdrücke X und Y nicht äquivalent sein, siehe Tabelle 6.1.

Definition 6.3.1 (Normalform) Die Normalform für Cliquesweite k -Ausdrücke ist wie folgt definiert.

1. Der k -Ausdruck \bullet_t , für $t \in [k]$, ist in Normalform.
2. Falls Y_1 und Y_2 zwei k -Ausdrücke in Normalform sind, dann ist der k -Ausdruck

$$\rho_{i_n \rightarrow j_n}(\cdots \rho_{i_1 \rightarrow j_1}(\eta_{i'_n, j'_n}(\cdots \eta_{i'_1, j'_1}(Y_1 \oplus Y_2) \cdots)) \cdots)$$

$$\begin{array}{lcl}
 X_1 = \eta_{2,3}((\eta_{1,2}(\bullet_1 \oplus \bullet_2)) \oplus \bullet_3) & \equiv & X_2 = \eta_{1,2}(\bullet_1 \oplus (\eta_{2,3}(\bullet_2 \oplus \bullet_3))) \\
 \begin{array}{c} \textcircled{1} \textsubscript{1} \text{---} \textcircled{2} \textsubscript{2} \text{---} \textcircled{3} \textsubscript{3} \end{array} & & \begin{array}{c} \textcircled{1} \textsubscript{1} \text{---} \textcircled{2} \textsubscript{2} \text{---} \textcircled{3} \textsubscript{3} \end{array} \\
 X_3 = \eta_{1,2}(\bullet_1 \oplus \bullet_2) & \not\equiv & X_4 = \eta_{1,2}(\bullet_2 \oplus \bullet_1) \\
 \begin{array}{c} \textcircled{1} \textsubscript{1} \text{---} \textcircled{2} \textsubscript{2} \end{array} & & \begin{array}{c} \textcircled{2} \textsubscript{1} \text{---} \textcircled{1} \textsubscript{2} \end{array} \\
 X_5 = (\eta_{1,2}(\bullet_2 \oplus \bullet_1)) \oplus \bullet_2 & \not\equiv & X_6 = \bullet_2 \oplus (\eta_{1,2}(\bullet_1 \oplus \bullet_2)) \\
 \begin{array}{c} \textcircled{2} \textsubscript{1} \text{---} \textcircled{1} \textsubscript{2} \quad \textcircled{2} \textsubscript{3} \end{array} & & \begin{array}{c} \textcircled{2} \textsubscript{1} \quad \textcircled{1} \textsubscript{2} \text{---} \textcircled{2} \textsubscript{3} \end{array}
 \end{array}$$

Tabelle 6.1: Die Indizes rechts unter den Knoten der Graphen stellen die Nummerierung der Knoten bezüglich des k -Ausdrucks dar. Die 3-Ausdrücke X_1 und X_2 sind äquivalent, da $\text{val}(X_1)$ und $\text{val}(X_2)$ die gleiche Knotenanzahl haben, der i -te Knoten in $\text{val}(X_1)$ hat die gleiche Markierung wie der i -te Knoten in $\text{val}(X_2)$, $1 \leq i \leq 3$, und es gibt eine Kante zwischen dem i -ten und j -ten Knoten in $\text{val}(X_1)$, $1 \leq i, j \leq 3$, genau dann wenn es eine Kante zwischen dem i -ten und j -ten Knoten in $\text{val}(X_2)$ gibt. Die Ausdrücke X_1 und X_2 sind jedoch nicht gleich.

Die 2-Ausdrücke X_3 und X_4 sind nicht äquivalent, da der erste Knoten aus X_3 nicht die gleiche Markierung hat wie der erste Knoten aus X_4 .

Die 2-Ausdrücke X_5 und X_6 sind nicht äquivalent, da es eine Kante zwischen dem ersten Knoten aus X_5 und dem zweiten Knoten aus X_5 in $\text{val}(X_5)$ gibt, jedoch gibt es keine Kante zwischen dem ersten Knoten von X_6 und dem zweiten Knoten aus X_6 in $\text{val}(X_6)$. Die 2-markierten Graphen $\text{val}(X_5)$ und $\text{val}(X_6)$ sind isomorph.

für $i_1, j_1, \dots, i_n, j_n, i'_1, j'_1, \dots, i'_n, j'_n \in [k]$ in Normalform falls die folgenden Bedingungen erfüllt sind.

(a) Für jede Kanteneinfügeoperation $\eta_{i'_l, j'_l}$, $1 \leq l \leq n'$, gilt

$$\begin{aligned}
 \eta_{i'_l, j'_l}(\eta_{i'_{l-1}, j'_{l-1}}(\dots \eta_{i'_1, j'_1}(Y_1 \oplus Y_2) \dots)) &\not\equiv \eta_{i'_{l-1}, j'_{l-1}}(\dots \eta_{i'_1, j'_1}(Y_1 \oplus Y_2) \dots), \\
 \eta_{i'_l, j'_l}(Y_1) &\equiv Y_1 \quad \text{und} \quad \eta_{i'_l, j'_l}(Y_2) \equiv Y_2.
 \end{aligned}$$

(b) Für jede Ummarkierungsoperation $\rho_{i_l \rightarrow j_l}$, $1 \leq l \leq n$, hat der Graph

$$\text{val}(\rho_{i_{l-1} \rightarrow j_{l-1}}(\dots \rho_{i_1 \rightarrow j_1}(\eta_{i'_n, j'_n}(\dots \eta_{i'_1, j'_1}(Y_1 \oplus Y_2) \dots)) \dots))$$

einen Knoten der mit i_l markiert ist und einen Knoten der mit j_l markiert ist und $i_l \notin \{j_1, \dots, j_{l-1}\}$.

(c) Für jedes Paar verschiedener Markierungen $i, j \in [k]$ gilt

i. falls der Graph $\text{val}(Y_1)$ einen mit i markierten Knoten und einen mit j markierten Knoten enthält, dann gilt

$$\begin{aligned}
 &\rho_{i_l \rightarrow j_l}(\dots \rho_{i_1 \rightarrow j_1}(\eta_{i'_n, j'_n}(\dots \eta_{i'_1, j'_1}(Y_1 \oplus Y_2) \dots)) \dots) \\
 &\not\equiv \rho_{i_l \rightarrow j_l}(\dots \rho_{i_1 \rightarrow j_1}(\eta_{i'_n, j'_n}(\dots \eta_{i'_1, j'_1}(\rho_{i \rightarrow j}(Y_1) \oplus Y_2) \dots)) \dots)
 \end{aligned}$$

und

ii. falls der Graph $\text{val}(Y_2)$ einen mit i markierten Knoten und einen mit j markierten Knoten enthält, dann gilt

$$\begin{aligned} & \rho_{i_l \rightarrow j_l}(\cdots \rho_{i_1 \rightarrow j_1}(\eta_{i'_n, j'_n}(\cdots \eta_{i'_1, j'_1}(\text{Y}_1 \oplus \text{Y}_2) \cdots)) \cdots) \\ \neq & \rho_{i_l \rightarrow j_l}(\cdots \rho_{i_1 \rightarrow j_1}(\eta_{i'_n, j'_n}(\cdots \eta_{i'_1, j'_1}(\text{Y}_1 \oplus \rho_{i \rightarrow j}(\text{Y}_2)) \cdots)) \cdots). \end{aligned}$$

Ist X ein k -Ausdruck in Normalform, so sind die Operationen zwischen zwei Vereinigungsoperationen so angeordnet, dass erst alle Kanteneinfügeoperationen erfolgen und dann die Ummarkierungsoperationen. Kanten werden so früh wie möglich eingefügt und Knoten werden so früh wie möglich ummarkiert. Nach Definition 6.3.1(2.(a)) fügt eine Kanteneinfügeoperation $\eta_{i', j'}$ mindestens eine Kante zwischen einem Knoten aus $\text{val}(Y_1)$ und einem Knoten aus $\text{val}(Y_2)$ ein, jedoch keine Kanten zwischen zwei Knoten aus $\text{val}(Y_1)$ oder zwischen zwei Knoten aus $\text{val}(Y_2)$. Nach Definition 6.3.1(2.(b)) markiert eine Ummarkierungsoperation $\rho_{i_l \rightarrow j_l}$ mindestens einen Knoten mit einer Markierung die schon ein anderer Knoten besitzt. Jede Ummarkierungsoperation verringert somit die Anzahl der Markierungen der Knoten im Graphen. Kein Knoten wird zwischen zwei Vereinigungsoperationen mehr als einmal ummarkiert. Keine der Operationen $\rho_{i_{l-1} \rightarrow j_{l-1}}, \dots, \rho_{i_1 \rightarrow j_1}$ markiert einen Knoten mit i_l . Damit ist i_l nicht an den Ummarkierungsoperationen $\rho_{i_{l-1} \rightarrow j_{l-1}}, \dots, \rho_{i_1 \rightarrow j_1}$ beteiligt, d.h. $i_l \notin \{i_{l-1}, \dots, i_1, j_{l-1}, \dots, j_1\}$ und j_l wurde zuvor nicht ummarkiert, d.h. $j_l \notin \{i_{l-1}, \dots, i_1\}$. Nach Definition 6.3.1(2.(c)) ist die Anzahl der Markierungen, die von Graphen der Teilausdrücke verwendet werden, stets minimal.

Man sieht leicht, dass falls der k -Ausdruck $\rho_{i \rightarrow j}(Y)$ in Normalform ist, so ist auch der k -Ausdruck Y in Normalform. Ist der k -Ausdruck $\eta_{i', j'}(Y)$ in Normalform ist, so ist auch der k -Ausdruck Y in Normalform. Ist der k -Ausdruck $X \oplus Y$ in Normalform ist, so sind auch die k -Ausdrücke X und Y in Normalform.

Theorem 6.3.2 *Zu jedem Cliquesweite k -Ausdruck X gibt es einen äquivalenten Cliquesweite k -Ausdruck in Normalform.*

Beweis Wir zeigen wie ein beliebiger Cliquesweite k -Ausdruck X in einen äquivalenten Cliquesweite k -Ausdruck in Normalform umgeformt werden kann.

Die folgenden Umformungen können angewendet werden um einen k -Ausdruck X in einen äquivalenten k -Ausdruck umzuformen in dem keine Kanteneinfügeoperation direkt auf eine Ummarkierungsoperation folgt.

Es sei $Z = \eta_{i', j'}(\rho_{i \rightarrow j}(Y))$ ein Teilausdruck von X .

1. Falls $\{i, j\} \cap \{i', j'\} = \emptyset$, so kann Z durch $\rho_{i \rightarrow j}(\eta_{i', j'}(Y))$ ersetzt werden, da die beiden Markierungen i und j der Ummarkierungsoperation nicht an der Kanteneinfügeoperation beteiligt sind.
2. Falls $i \in \{i', j'\}$, dann können wir die Kanteneinfügeoperation $\eta_{i', j'}$ entfernen, da sie keine Kanten erzeugt.
3. Falls $i \notin \{i', j'\}$ und $j \in \{i', j'\}$, dann unterscheiden wir die folgenden beiden Fälle. Falls $j = i'$ gilt, so kann Z durch $\rho_{i \rightarrow j}(\eta_{i', j'}(\eta_{i, j'}(Y)))$ ersetzt werden und falls $j = j'$ gilt, so wird Z durch $\rho_{i \rightarrow j}(\eta_{i', j'}(\eta_{i', i}(Y)))$ ersetzt.

Mit Hilfe dieser Umformungsschritte können wir jeden k -Ausdruck X in einen äquivalenten k -Ausdruck umformen, in dem alle Kanteneinfügeoperationen und Ummarkierungsoperationen in der in Definition 6.3.1 geforderten Reihenfolge auftreten.

Nun zeigen wir die Behauptung mit vollständiger Induktion nach der Anzahl der Vereinigungsknoten in X und nach dem Aufbau von X . Die folgenden Umformungen verändern die Anzahl der Vereinigungsknoten in den Teilausdrücken nicht.

Es sei

$$X = \rho_{i_n \rightarrow j_n}(\cdots \rho_{i_1 \rightarrow j_1}(\eta_{i'_{n'}, j'_{n'}}(\cdots \eta_{i'_1, j'_1}(\bullet_t) \cdots)) \cdots)$$

ein k -Ausdruck ohne Vereinigungsoperationen. X ist äquivalent zu dem k -Ausdruck \bullet_j mit $j \in \{j_1, \dots, j_n, t\}$.

Nun sei

$$X = \eta_{i'_{l'}, j'_{l'}}(\eta_{i'_{l'-1}, j'_{l'-1}}(\cdots \eta_{i'_1, j'_1}(Y_1 \oplus Y_2) \cdots))$$

ein k -Ausdruck, wobei

$$X = \eta_{i'_{l'-1}, j'_{l'-1}}(\cdots \eta_{i'_1, j'_1}(Y_1 \oplus Y_2) \cdots)$$

in Normalform sei.

1. Falls $\text{val}(Y_1)$ nicht alle Kanten zwischen Knoten die mit $i'_{l'}$ und Knoten die mit $j'_{l'}$ markiert sind enthält, dann formen wir den k -Ausdruck $\eta_{i'_{l'}, j'_{l'}}(Y_1)$ in einen äquivalenten k -Ausdruck Y'_1 in Normalform um und ersetzen im Ausdruck X den Ausdruck Y_1 durch Y'_1 . Die Umformung von $\eta_{i'_{l'}, j'_{l'}}(Y_1)$ in Normalform ist nach Induktionsannahme möglich. Die gleiche Umformung können wir auch für Y_2 durchführen, falls dies notwendig ist.
2. Falls

$$\eta_{i'_{l'}, j'_{l'}}(\eta_{i'_{l'-1}, j'_{l'-1}}(\cdots \eta_{i'_1, j'_1}(Y_1 \oplus Y_2) \cdots)) \equiv \eta_{i'_{l'-1}, j'_{l'-1}}(\cdots \eta_{i'_1, j'_1}(Y_1 \oplus Y_2) \cdots)$$

dann können wir die Operation $\eta_{i'_{l'}, j'_{l'}}$ aus X entfernen, da diese keine Kanten erzeugt.

In beiden Fällen erhalten wir einen äquivalenten k -Ausdruck in Normalform.

Es sei

$$X = \rho_{i_n \rightarrow j_n}(\cdots \rho_{i_1 \rightarrow j_1}(\eta_{i'_{n'}, j'_{n'}}(\cdots \eta_{i'_1, j'_1}(Y_1 \oplus Y_2) \cdots)) \cdots)$$

ein k -Ausdruck, in dem der Teilausdruck

$$Z_0 := \eta_{i'_{n'}, j'_{n'}}(\cdots \eta_{i'_1, j'_1}(Y_1 \oplus Y_2) \cdots)$$

in Normalform ist.

Falls der Graph $\text{val}(Y_1)$ einen Knoten enthält der mit i markiert ist und einen Knoten enthält der mit j markiert ist, $i, j \in [k]$, $i \neq j$ und

$$\begin{aligned} & \rho_{i_l \rightarrow j_l}(\cdots \rho_{i_1 \rightarrow j_1}(\eta_{i'_{n'}, j'_{n'}}(\cdots \eta_{i'_1, j'_1}(Y_1 \oplus Y_2) \cdots)) \cdots) \\ \equiv & \rho_{i_l \rightarrow j_l}(\cdots \rho_{i_1 \rightarrow j_1}(\eta_{i'_{n'}, j'_{n'}}(\cdots \eta_{i'_1, j'_1}(\rho_{i \rightarrow j}(Y_1) \oplus Y_2) \cdots)) \cdots), \end{aligned}$$

dann formen wir den Teilausdruck $\rho_{i \rightarrow j}(Y_1)$ in einen äquivalenten k -Ausdruck Y'_1 in Normalform um und ersetzen im k -Ausdruck X den Teilausdruck Y_1 durch Y'_1 . Die Umformung von

$\rho_{i \rightarrow j}(Y_1)$ in Normalform ist nach Induktionsannahme möglich. Dannach formen wir Z_0 mit dem neuen Teilausdruck Y_1 in Normalform um, falls dies notwendig ist. Diese Umformung ist nach der Induktionsannahme und den obigen Umformungen möglich. Die gleiche Umformung erfolgt für Y_2 , falls dies notwendig ist. Die Umformungsschritte können höchstens $k - 1$ mal für Y_1 und Y_2 wiederholt werden, da nach jeder Umformung die Anzahl der Markierungen in $\text{val}(Y_1)$ und $\text{val}(Y_2)$ um eins geringer wird.

Nun berechnen wir für jede Markierung $l \in [k]$ die Markierung $h(l)$ in die l durch die n Operationen $\rho_{i_1 \rightarrow j_1}, \dots, \rho_{i_n \rightarrow j_n}$ in dieser Reihenfolge ummarkiert wird. Die Funktion $h : [k] \rightarrow [k]$ kann als gerichteter Graph $H = (V_H, E_H)$ mit Knotenmenge $V_H = [k]$ und Kantenmenge $E_H = \{(l, h(l)) \mid l \in [k]\}$ dargestellt werden. Jeder Knoten l in H hat genau eine auslaufende gerichtete Kante $(l, h(l))$.

Zuerst entfernen wir alle gerichteten Kanten (l_1, l_2) aus H , für die der Graph $\text{val}(Z_0)$ keinen mit l_1 markierten Knoten enthält und alle Kanten (l_1, l_2) mit $l_1 = l_2$, da diese Kanten keine Ummarkierungen von Knoten in $\text{val}(Z_0)$ repräsentieren. Dann werden alle Kantenpaare $(l_1, l_2), (l_2, l_3)$ in H betrachtet und wir ersetzen gleichzeitig im Ausdruck Z_0 alle Markierungen l_1 durch l_2 und l_2 durch l_1 , wir entfernen die Kanten $(l_1, l_2), (l_2, l_3)$ aus H und fügen falls $l_1 \neq l_3$ eine neue Kante (l_1, l_3) zu H hinzu. Dazu beachte man, dass Z_0 in Normalform bleibt, wenn wir zwei Markierungen vertauschen. Dann betrachten wir alle Kanten (l_1, l_2) für die der Graph $\text{val}(Z_0)$ keinen mit l_2 markierten Knoten enthält. Wir ersetzen im Ausdruck Z_0 alle Markierungen l_1 durch l_2 und alle Markierungen l_2 durch l_1 und entfernen die Kante (l_1, l_2) aus H .

Mit Hilfe der verbleibenden Kanten in H definieren wir eine neue Ummarkierung für die Knoten in Z_0 . Wir beginnen mit dem Ausdruck Z_0 und löschen schrittweise eine Kante (l_1, l_2) aus H und wenden die Operation $\rho_{l_1 \rightarrow l_2}$ auf den aktuellen Ausdruck Z_i an um einen neuen Ausdruck $Z_{i+1} = \rho_{l_1 \rightarrow l_2}(Z_i)$ zu erhalten. So erhalten wir einen k -Ausdruck in Normalform der äquivalent zu unserem ursprünglichen Ausdruck ist. \square

Der Beweis von Theorem 6.3.1 nutzt einen Ummarkierungstrick um eine Ummarkierungsoperation $\rho_{i_l \rightarrow j_l}(X)$ zu vermeiden, falls der Graph $\text{val}(X)$ keinen Knoten enthält, der mit j_l markiert ist. Diese Ummarkierung ersetzt im Graph $\text{val}(X)$ gleichzeitig jede Markierung i_l durch j_l und jede Markierung j_l durch i_l . Den so erhaltenen Ausdruck bezeichnen wir mit $X_{i_l \leftrightarrow j_l}$. Ist X in Normalform, so ist auch $X_{i_l \leftrightarrow j_l}$ in Normalform und $\rho_{i_l \rightarrow j_l}(X) \equiv X_{i_l \leftrightarrow j_l}$.

6.4 Überblick

In diesem Abschnitt soll erläutert werden wie der Beweis in den nächsten Abschnitten abläuft. Π_k sei die Grapheigenschaft „Cliquesweite höchstens k “ und $\sim_{\Pi_k, l}$ sei die Äquivalenzrelation die wir in Kapitel 6.2 für k -markierte l -terminale Graphen definiert haben. Wir wollen zeigen, dass $\sim_{\Pi_k, l}$ die Menge aller k -markierter l -terminaler Graphen für jedes $k \geq 1$ und jedes $l \geq 0$ in endlich viele Äquivalenzklassen einteilt. Damit wird unmittelbar folgen, dass die Grapheigenschaft „Cliquesweite höchstens k “ in Linearzeit für Graphen mit Baumweite höchstens l entscheidbar ist.

Wir definieren zu jedem k -markierten l -terminalen Graphen G einen *Verbindungstyp*, der aus einer Menge von *Verbindungsbäumen* besteht. Wir zeigen, dass zwei k -markierte l -terminale Graphen bezüglich der Grapheigenschaft Cliquesweite höchstens k ersetzbar sind, falls sie den gleichen Verbindungstyp haben. Damit kann die Anzahl der Äquivalenzklassen von $\sim_{\Pi_k, l}$ durch die Anzahl der verschiedenen Verbindungstypen für alle k -markierten

l -terminalen Graphen nach oben abgeschätzt werden. Falls es für jedes feste $k \geq 1$ und jedes feste $l \geq 0$ endlich viele verschiedene Verbindungstypen gibt, dann hat $\sim_{\Pi_k, l}$ endlich viele Äquivalenzklassen und unsere Behauptung folgt.

Wir beschreiben den Beweis mittels einer vereinfachten Version des Verbindungsbaumes. Um diesen Baum von dem später gebrauchten Verbindungsbaum zu unterscheiden, nennen wir die einfachere Version *starker Verbindungsbaum*. H und J seien zwei k -markierte l -terminale Graphen, so dass der k -markierte Graph $G = H \circ J$ Cliquesweite höchstens k hat, siehe Abbildung 6.3. X sei ein Cliquesweite k -Ausdruck für $H \circ J$ und T sei der Cliquesweite k -Ausdrucksbaum, vergleiche Definition 2.3.2, zu X . Der Baum T kann wie folgt in zwei Teilbäume T_H und T_J eingeteilt werden. Der Teilbaum T_H beschreibt den k -markierten Teilgraphen von $H \circ J$ der durch die Knoten von H induziert wird. T_H besteht aus den Blättern von T , die Knoten aus H entsprechen und allen Knoten von T auf dem Weg dieser Blätter zur Wurzel von T . T_J wird auf die gleiche Weise mit Hilfe der Knoten aus J definiert. Jeder Knoten aus T ist in mindestens einem der beiden Teilbäume T_H oder T_J enthalten. Einige Knoten aus T sind in beiden Bäumen T_H und T_J enthalten. Zum Beispiel die Wurzel von T ist sowohl in T_H als auch in T_J enthalten. Die Blätter aus T die identifizierten terminalen Knoten von H und J entsprechen und alle Knoten auf den Wegen von diesen Blättern zur Wurzel von T sind ebenfalls in beiden Bäumen T_H und T_J enthalten.

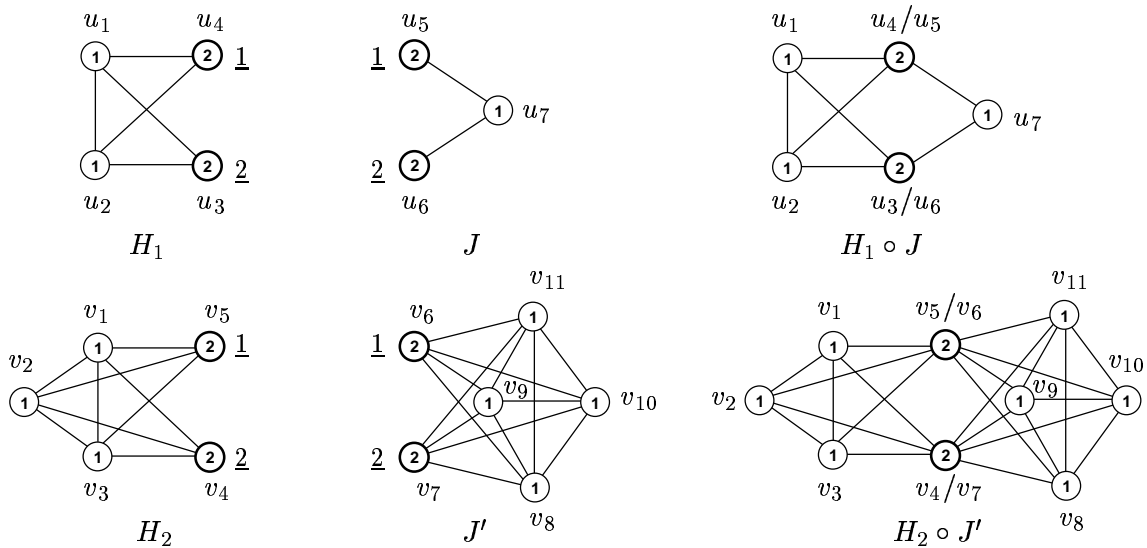


Abbildung 6.3: Vier 2-markierte 2-terminale Graphen H_1, H_2, J und J' und die 2-markierten Graphen $H_1 \circ J$ und $H_2 \circ J'$

Der gemeinsame Teil der beiden Teilbäume T_H und T_J , wir bezeichnen diesen im Folgenden mit C , definiert einen *starken Verbindungsbaum* für H . Die Blätter im gemeinsamen Teil C sind entweder Blätter aus T oder Vereinigungsknoten. Falls ein Blatt in C einem Knoten aus $H \circ J$ entspricht, der durch die Identifizierung der i -ten terminalen Knoten aus H und J erhalten wurde, dann markieren wir den Knoten in C mit dem Index i . Es sei u ein Vereinigungsknoten des k -Ausdrucksbaumes T und u_l und u_r seien der linke bzw. der rechte Sohn von u in T . Falls u im gemeinsamen Teil C enthalten ist, aber u_l oder u_r nicht im gemeinsamen Teil C vorkommt, dann fügen wir in C einen linken Sohn v_l bzw. einen rechten

Sohn v_r ein, so dass wir einen geordneten binären Baum erhalten in dem jeder Vereinigungsknoten einen linken und einen rechten Sohn hat. Falls u_l (u_r) ein Knoten aus T_H ist aber nicht in T_J enthalten ist, dann wird das in C eingefügte Blatt v_l (bzw. v_r) mit der Menge L der Markierungen der Knoten im k -markierten Graphen $G(T, u_l)$ (bzw. $G(T, u_r)$) markiert. In Abbildung 6.4 wird ein Beispiel für einen solchen eingefügten Knoten und eine zusätzliche Markierung gezeigt. Eine nicht leere Markierungsmenge $L \subseteq [k]$ in C bedeutet, dass das Blatt einen Teilbaum aus T_H und nicht einen Teilbaum aus T_J repräsentiert. Diese Blätter werden auch *innere Blätter* und die restlichen Blätter werden *äußere Blätter* genannt. Die Bezeichnungen „innere“ und „äußere“ sollen andeuten, dass das linke Argument H der innere Graph ist zu dem wir den Verbindungsbaum bestimmen und dass das rechte Argument J der äußere Graph ist, d.h die Umgebung von H darstellt. Der so entstandene markierte Baum C wird starker Verbindungsbaum für H genannt. Um alle starken Verbindungsbäume für H zu erhalten, müssen wir alle k -markierten l -terminalen Graphen J betrachten für die der Graph $H \circ J$ Cliquesweite höchstens k hat. Weiterhin müssen alle k -Ausdrücke für $H \circ J$ betrachtet werden. Die Menge aller starken Verbindungsbäume für H ist der *starke Verbindungstyp* für H .

Nun soll erläutert werden, warum zwei k -markierte l -terminale Graphen H_1 und H_2 mit dem gleichen starken Verbindungstyp bezüglich der Grapheigenschaft Cliquesweite höchstens k ersetzbar sind. Der Graph $H_1 \circ J$ habe Cliquesweite höchstens k , für einen k -markierten l -terminalen Graphen J . X sei ein Cliquesweite k -Ausdruck für $H_1 \circ J$. T sei der k -Ausdrucksbaum für X und T_{H_1} und T_J seien die beiden Teilbäume von T die zu H_1 bzw. zu J gehören. Der gemeinsame Teil von T_{H_1} und T_J definiert einen starken Verbindungsbaum C für H_1 . Dieser Baum C ist nach unserer Annahme auch ein starker Verbindungsbaum für H_2 . Es muss also mindestens einen k -markierten l -terminalen Graphen J' geben, so dass $H_2 \circ J'$ Cliquesweite höchstens k hat. Es gibt dann einen k -Ausdruck X' für $H_2 \circ J'$ zu dem ein k -Ausdrucksbaum T' gehört, so dass der gemeinsame Teil der beiden Teilbäume T'_{H_2} und $T'_{J'}$ den gleichen starken Verbindungsbaum C für H_2 definieren. Nun können wir im k -Ausdrucksbaum T den Teilbaum T_{H_1} durch den Teilbaum T'_{H_2} austauschen, siehe Abbildung 6.5. Dies kann durch Ersetzung der Teilbäume, die zu den inneren Blättern gehören, erfolgen. T'' sei der durch diese Ersetzung entstandene k -Ausdrucksbaum.

Man kann leicht sehen, dass der k -Ausdrucksbaum T'' den k -markierten Graphen $H_2 \circ J$ beschreibt. Die Knoten aus H_2 und J sind in dem k -markierten Graphen der durch T'' definiert wird genau so markiert wie in den k -markierten Graphen $H_2 \circ J'$ und $H_1 \circ J$. Zwei Knoten aus H_2 oder zwei Knoten aus J sind adjazent, genau dann wenn sie in $H_2 \circ J'$ bzw. in $H_1 \circ J$ adjazent sind. Dies gilt, da die Teilbäume die durch Wege von den beteiligten Blätter hin zu den Wurzeln in den beiden k -Ausdrucksbäumen T'' und T bzw. in den beiden k -Ausdrucksbäumen T'' und T' gleich sind.

Die L -Markierungen der inneren Blätter im starken Verbindungsbaum C ist notwendig um sicherzustellen, dass T'' keine Kante zwischen einem inneren Knoten u_1 aus H_2 der nicht in J enthalten ist und einem inneren Knoten u_2 aus J der nicht in H_2 enthalten ist, erzeugt. Falls der Graph der durch T'' definiert wird eine solche verbotene Kante enthält, dann besitzt auch der Graph $H_1 \circ J$ mindestens eine verbotene Kante, da der entsprechende Teilgraph von H_1 mindestens einen Knoten mit der gleichen Markierung wie u_1 in H_2 hätte. Falls für jeden k -markierten l -terminalen Graphen J , der Graph $H_1 \circ J$ Cliquesweite höchstens k hat, genau dann wenn der Graph $H_2 \circ J$ Cliquesweite höchstens k hat, dann sind H_1 und H_2 bezüglich der Grapheigenschaft Cliquesweite höchstens k ersetzbar.

Nun betrachten wir die Größe der starken Verbindungsbäume. Die Größe des gemeinsamen

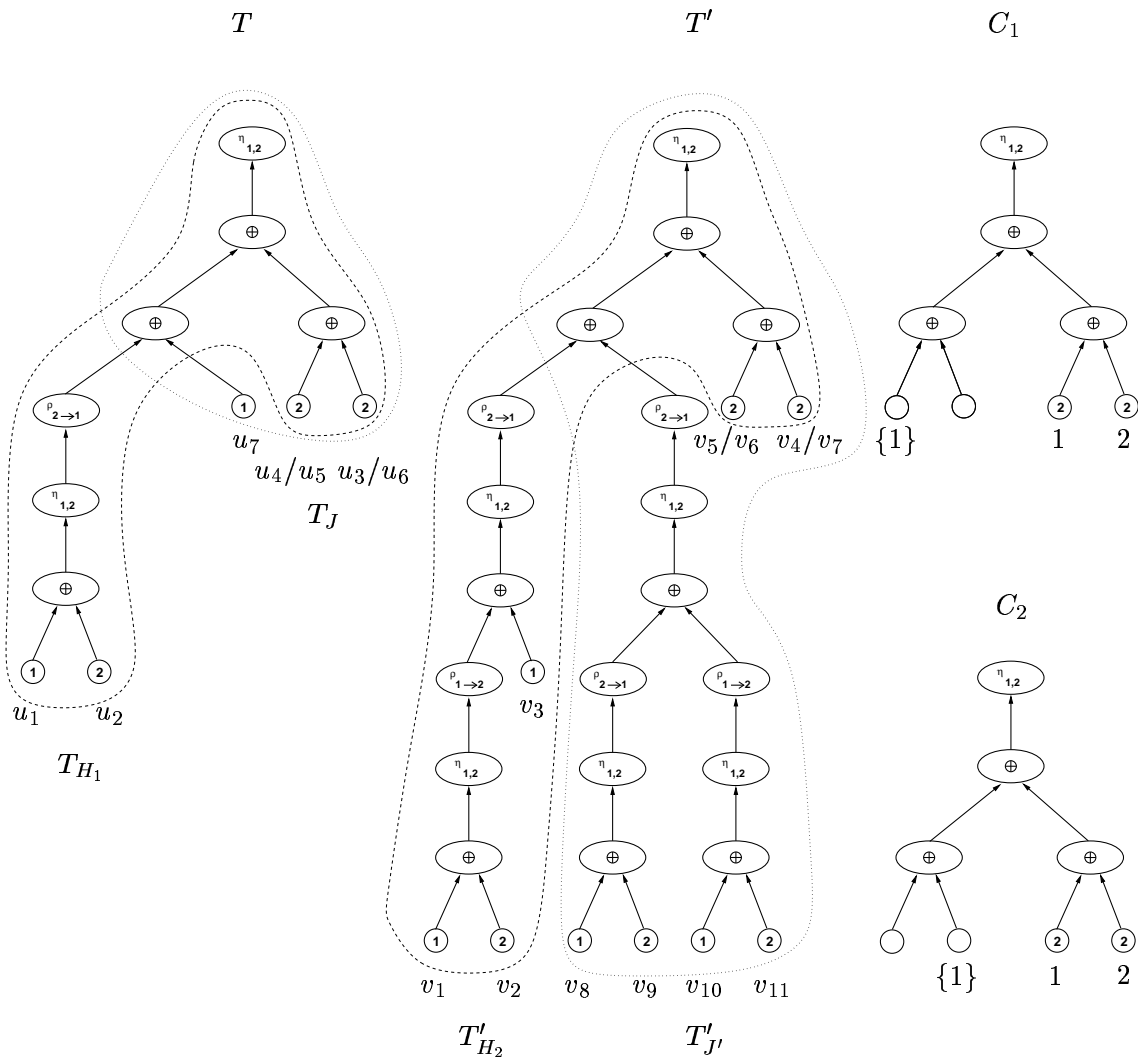


Abbildung 6.4: Ein Cliqueweite 2-Ausdrucksbaum T für den Graphen $H_1 \circ J$ und ein Cliqueweite 2-Ausdrucksbaum T' für den Graphen $H_2 \circ J'$. C_1 ist ein starker Verbindungsbaum für H_1 und H_2 . C_2 ist ein starker Verbindungsbaum für J und J' .

Teils der beiden Teilbäume T_H und T_J kann nicht durch eine Konstante, die nur von k und l abhängt, beschränkt werden. Die nächsten Abschnitte beschäftigen sich damit zu zeigen, dass es für jeden k -markierten Graphen $H \circ J$ mit Cliqueweite höchstens k mindestens einen k -Ausdrucksbaum T gibt, so dass die notwendige Information des gemeinsamen Teils der Bäume T_H und T_J beschränkt werden kann. Diese Information wird weiterhin baumstrukturiert sein und wird in den nächsten Abschnitten als Verbindungsbaum definiert.

Wir werden schrittweise zeigen, dass es einen k -Ausdrucksbaum für $H \circ J$ gibt in dem die Wege im gemeinsamen Teil von T_H und T_J die folgende Form haben. Wir teilen die Operationen der Knoten in T in H -Operationen und J -Operationen ein. Eine H -Operation markiert einen Knoten aus H um oder fügt eine zu einem Knoten aus H inzidente Kante ein. Eine J -Operation behandelt entsprechend Knoten aus J . Einige Knoten aus T sind sowohl

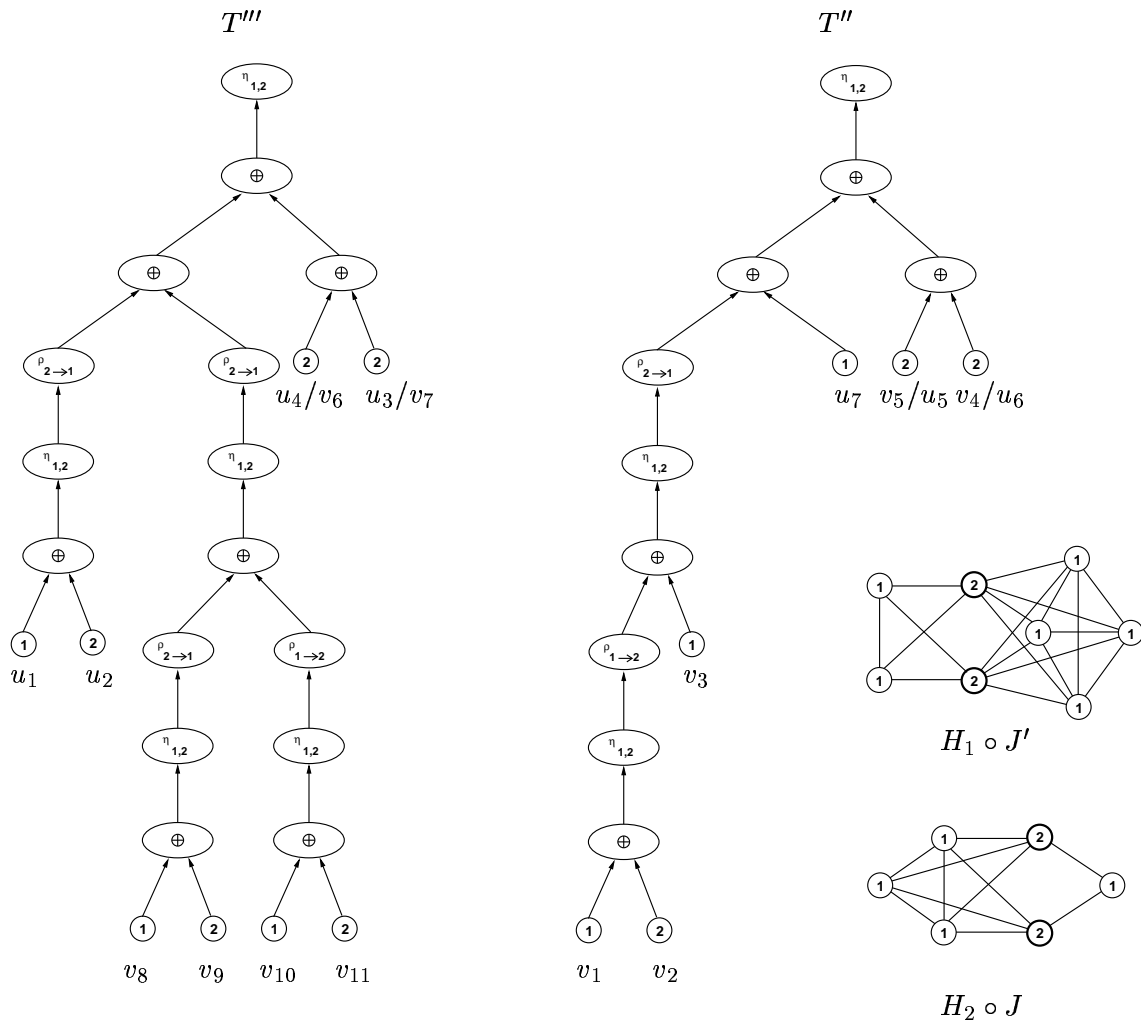


Abbildung 6.5: Ein Cliquesweite 2-Ausdrucksbaum T''' für den Graphen $H_1 \circ J'$, ein Cliquesweite 2-Ausdrucksbaum T'' für den Graphen $H_2 \circ J$ und die 2-markierten Graphen $H_1 \circ J'$ und $H_2 \circ J$.

H -Operationen als auch J -Operationen. Im nächsten Abschnitt werden wir zeigen, dass es immer einen k -Ausdrucksbaum T für $H \circ J$ gibt, so dass im gemeinsamen Teil von T_H und T_J die Anzahl der Wechsel zwischen H -Operationen und J -Operationen entlang eines Weges von einem Blatt zur Wurzel von T durch eine Konstante, die nur von k und l und nicht von der Größe von H oder der Größe von J abhängt, beschränkt werden kann. Diese Eigenschaft ermöglicht es uns eine Verbindungsstruktur beschränkter Größe zu definieren, die wir dann den Verbindungsbaum für H nennen. Die wesentliche Idee besteht darin, dass wir Teilwege mit Operationen desgleichen Typs durch sogenannte *Brückenknoten* ersetzen.

6.5 Bestimmung des Verbindungstyps

Wir nehmen nun an, dass wir einen k -markierten l -terminalen Graphen $H = (V_H, E_H, P_H, \text{lab}_H)$ und einen k -markierten l -terminalen Graphen $J = (V_J, E_J, P_J, \text{lab}_J)$ gegeben haben, so dass H und J knotendisjunkt sind und der k -markierte Graph

$$G = (V_G, E_G, \text{lab}_G) = H \circ J$$

Cliquenweite höchstens k hat.

Wir teilen die Knotenmenge V_G in drei disjunkte Mengen U_H, U_J, U_P ein, so dass $U_H \cup U_J \cup U_P = V_G$. Die Menge $U_H = V_H - P_H$ enthält die inneren Knoten von H , $U_J = V_J - P_J$ enthält die inneren Knoten von J und U_P enthält die identifizierten terminalen Knoten aus H und J . Die Knotenmenge U_P enthält genau l Knoten, da die l terminalen Knoten von H mit den l terminalen Knoten von J identifiziert werden. Der Graph G enthält keine Kante zwischen einem Knoten aus U_H und einem Knoten aus U_J , vergleiche Abbildung 6.6.

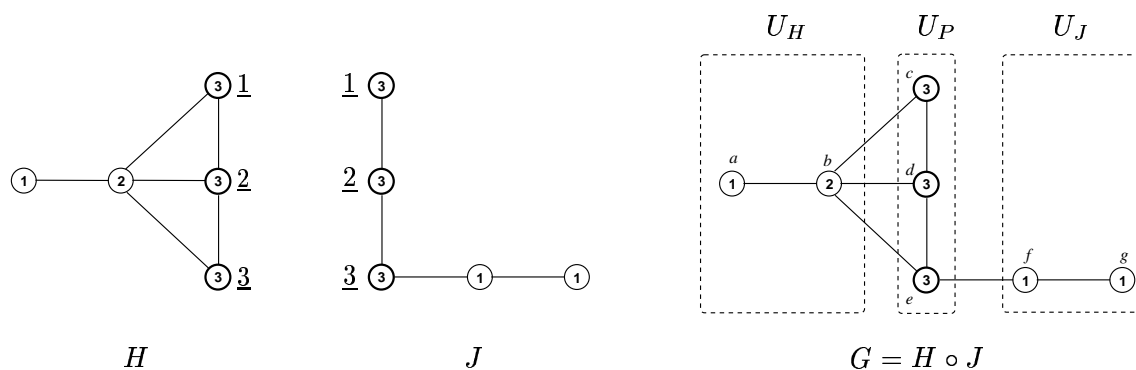


Abbildung 6.6: Zwei 3-markierte 3-terminale Graphen H und J , der 3-markierte Graph $G = H \circ J$ und die Aufteilung der Knotenmenge von G in U_H , U_P und U_J .

Es sei T ein Cliquenweite k -Ausdrucksbaum für $G = H \circ J$. Der Teilbaum T_P von T ist der Baum der aus den l Blättern von T besteht, die den Knoten aus U_P entsprechen und allen Wegen dieser Knoten zu der Wurzel von T , siehe Abbildung 6.7. Somit ist die Wurzel von T auch die Wurzel von T_P . Der Baum T_P ist im Allgemeinen kein Ausdrucksbaum. Falls H und J keine inneren Knoten besitzten, so ist T_P ein Ausdrucksbaum und T und T_P sind sogar gleich.

Wir wollen zeigen, dass es für jedes Paar H, J mindestens einen k -Ausdrucksbaum T für G gibt, so dass T_P eine spezielle Form hat. Diese Form beinhaltet die wesentlichen Informationen wie die beiden Graphen H und J miteinander verbunden werden. Wir zeigen, dass die Größe dieser Verbindungsinformationen nur von k und l abhängt und nicht von der Größe von H oder der Größe von J .

Die folgenden vier Abschnitte beginnen jeweils mit einem Lemma welches zeigt, dass wir jeweils ohne Einschränkungen einen spezielleren k -Ausdrucksbaum T betrachten können. Die Einschränkungen sind als Eigenschaften formuliert die der Baum T erfüllen muss. Die Lemmas zeigen, dass dies jeweils ohne Beschränkung der Allgemeinheit möglich ist.

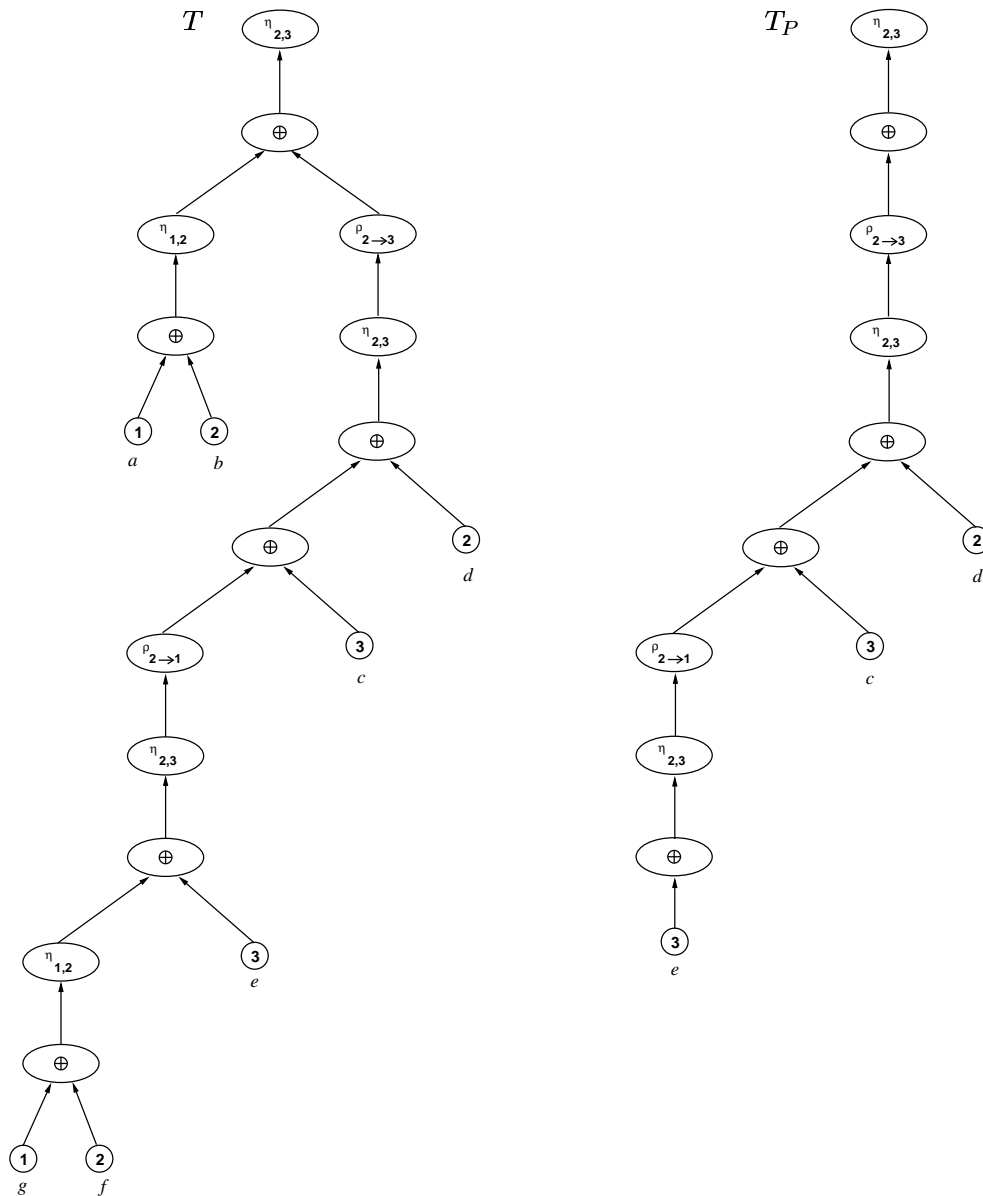


Abbildung 6.7: Ein Cliquesweite 3-Ausdrucksbaum T für den 3-markierten Graphen G aus Abbildung 6.6 und der Teilbaum T_P von T .

Einteilung in Wege vom Typ 1, 1.a und 1.b

Lemma 6.5.1 *Es gibt immer einen k -Ausdrucksbaum T für G , der die folgende Eigenschaft erfüllt.*

Eigenschaft 6.5.2 *Es sei u_1 ein Vereinigungsknoten in T , so dass einer seiner Söhne u_0 in T_P liegt und der andere Sohn u'_0 nicht in T_P liegt. Dann gehören die Knoten in $G(u'_0)$ entweder alle zu U_H oder alle zu U_J .*

Beweis Da $G(u'_0)$ keine Knoten aus U_P enthält, sind die Knoten aus $G(u'_0)$ alle aus $U_H \cup U_J$. Falls die Knoten aus $G(u'_0)$ nicht alle aus U_H oder nicht alle aus U_J sind, dann seien T_H und T_J die beiden k -Ausdrucksbäume, die die Teilgraphen von $G(u'_0)$ die durch die Knoten aus U_H bzw. durch die Knoten aus U_J induziert werden, definieren. Die Bäume T_H und T_J können leicht aus $T(u'_0)$ konstruiert werden, indem Teilbäume deren Blätter Knoten aus U_J bzw. Knoten aus U_H entsprechen, aus $T(u'_0)$ entfernt werden. Ein Vereinigungsknoten in T_H bzw. T_J der nun nur noch einen Sohn hat, kann durch diesen Sohn ersetzt werden. Wir ersetzen den Teilbaum $T(u'_0)$ durch den Baum T_H , fügen einen neuen Vereinigungsknoten v_0 zwischen u_1 und u_0 ein und fügen eine Kante zwischen v_0 und der Wurzel von T_J ein. Somit wird die Wurzel von T_J zum nicht in T_P enthaltenen Sohn von v_0 . Der so erhaltene Ausdrucksbaum definiert denselben Graphen wie der ursprüngliche Ausdrucksbaum und u_1 erfüllt nun Eigenschaft 6.5.2. Diese Transformation kann für alle Vereinigungsknoten vorgenommen werden die Eigenschaft 6.5.2 nicht erfüllen, vergleiche dazu Abbildung 6.8. \square

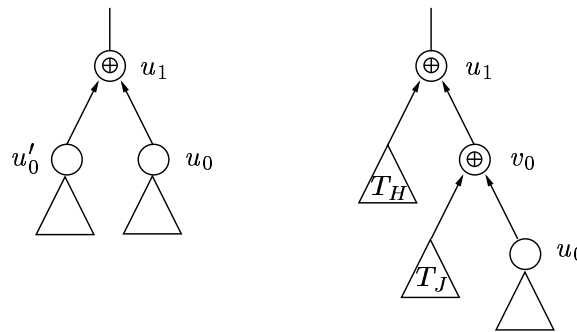


Abbildung 6.8: Ein Umformungsschritt aus dem Beweis von Lemma 6.5.1

Es sei X der Cliquesweite k -Ausdruck zu dem Ausdrucksbaum T der Eigenschaft 6.5.2 erfülle. Mit Hilfe der Umformungen aus dem Beweis von Theorem 6.3.2 erhält man einen zu X äquivalenten k -Ausdruck in Normalform der auch Eigenschaft 6.5.2 erfüllt, da die Umformungsschritte zur Erlangung der Normalform nur Ummarkierungs- und Kanteneinfügeknöten umordnen und Eigenschaft 6.5.2 für T nicht verändern. Wir nehmen nun an, dass T Eigenschaft 6.5.2 erfüllt und dass X in Normalform ist.

Es sei u_1 ein Vereinigungsknoten in T_P , so dass einer seiner Söhne u_0 in T_P liegt und der andere Sohn u'_0 nicht in T_P liegt. Wir definieren $\xi(u_1) := 0$ oder $\xi(u_1) := 1$, falls alle Knoten aus $G(u'_0)$ aus U_H sind bzw. falls alle Knoten aus $G(u'_0)$ aus U_J sind. In allen anderen Fällen und in den Fällen wo u_1 kein Vereinigungsknoten ist sei $\xi(u_1)$ undefiniert. Zur besseren Lesbarkeit schreiben wir im Weiteren stets $\xi(u_1) = H$ anstelle von $\xi(u_1) = 0$ und $\xi(u_1) = J$ anstelle von $\xi(u_1) = 1$. Dies soll nicht bedeuten, dass $\xi(u_1)$ der Graph H oder der Graph J ist, sondern nur andeuten, dass alle Knoten aus $G(u'_0)$ aus der Menge U_H bzw. U_J sind. Nach Lemma 6.5.1 können wir annehmen, dass $\xi(u_1)$ für alle Vereinigungsknoten u_1 in T_P , die genau einen Sohn nicht in T_P haben, definiert ist.

Der Baum T_P hat l Blätter und besteht aus höchstens $2l - 1$ maximalen Wegen $p = (u_1, \dots, u_{s'})$, $s' \geq 1$, so dass u_1 ein Vereinigungsknoten in T_P ist, der entweder zwei Söhne in T_P besitzt oder der nur einen Sohn in T_P besitzt, der ein Blatt ist. Der letzte Knoten $u_{s'}$ von p ist entweder die Wurzel von T_P oder der Sohn eines Vereinigungsknoten dessen Söhne beide

in T_P enthalten sind. Alle Graphen $G(u_s)$ für $s = 1, \dots, s'$ enthalten die gleichen Knoten aus U_P . Wir nennen einen solchen Weg p in T_P *1-Weg* oder *Weg vom Typ 1*. Jeder Knoten aus T_P der kein Blatt ist, ist in genau einem dieser Wege vom Typ 1 enthalten.

Ein maximaler Teilweg $(u_1, \dots, u_{r'}, \dots, u_{s'})$ in T_P , so dass u_1 ein Vereinigungsknoten ist, $u_2, \dots, u_{r'}$ Kanteneinfügeknoten und $u_{r'+1}, \dots, u_{s'}$ Ummarkierungsknoten sind, wird *Rahmen* von T_P genannt. Jeder Rahmen enthält höchstens $\binom{k}{2} + k$ Knoten, da es einen Vereinigungsknoten u_1 , höchstens $\binom{k}{2}$ Kanteneinfügeknoten $u_2, \dots, u_{r'}$ und höchstens $k - 1$ Ummarkierungsknoten $u_{r'+1}, \dots, u_{s'}$ gibt. Abbildung 6.9 zeigt den Aufbau eines Rahmens.

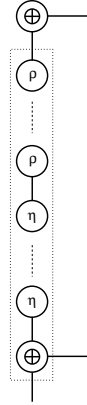


Abbildung 6.9: Ein Rahmen beginnt immer mit einem Vereinigungsknoten auf den dann Kanteneinfügeknoten und Ummarkierungsknoten folgen.

Der erste Rahmen eines 1-Weges wird *Weg vom Typ 1.a* genannt. Der restliche Teil eines 1-Weges, falls er nicht leer ist, heißt *Weg vom Typ 1.b*, siehe Abbildung 6.10. Es gibt in T_P höchstens $2l - 1$ Wege vom Typ 1.a und höchstens $2l - 1$ Wege vom Typ 1.b. Jeder 1.a-Weg hat höchstens $\binom{k}{2} + k$ Knoten, da er ein Rahmen ist. Für jeden Vereinigungsknoten u_1 in einem 1.b-Weg gilt entweder $\xi(u_1) = H$ oder $\xi(u_1) = J$.

Einteilung in Wege vom Typ 2.a und 2.b

Für einen Knoten u_s des k -Ausdrucksbaumes T , sei $L_H(u_s)$, $L_J(u_s)$ und $L_P(u_s)$ die Menge der Markierungen der Knoten in $G(u_s)$ die aus der Menge U_H , U_J bzw. U_P sind. Die Schnittmengen

$$L_H(u_s) \cap L_J(u_s), \quad L_P(u_s) \cap L_H(u_s), \quad L_P(u_s) \cap L_J(u_s) \quad \text{und} \quad L_P(u_s) \cap L_H(u_s) \cap L_J(u_s)$$

werden im Folgenden durch

$$L_{H \cap J}(u_s), \quad L_{P \cap H}(u_s), \quad L_{P \cap J}(u_s) \quad \text{und} \quad L_{P \cap H \cap J}(u_s)$$

abgekürzt.

Lemma 6.5.3 *Es gibt immer einen k -Ausdrucksbaum T für G , so dass der k -Ausdruck X in Normalform ist und T die Eigenschaften 6.5.2 und 6.5.4 erfüllt.*

Eigenschaft 6.5.4 *Es sei u_s ein Ummarkierungsknoten in T_P , der mit $\rho_{i \rightarrow j}$ markiert ist und u_{s-1} sei der Sohn von u_s . Falls $i \in L_P(u_{s-1})$, dann ist $j \in L_P(u_{s-1})$.*

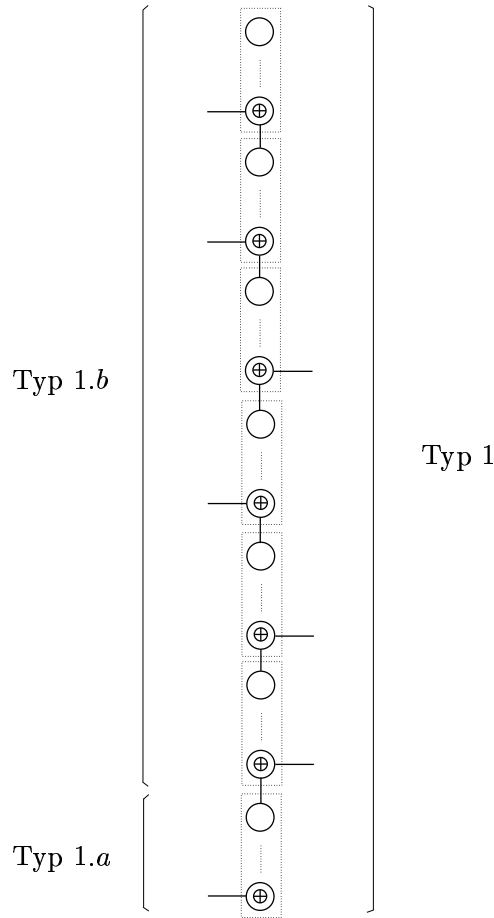


Abbildung 6.10: Jeder 1-Weg p wird in einen Weg vom Typ 1.a und einen Weg vom Typ 1.b unterteilt. Der Weg vom Typ 1.a ist der erste Rahmen in p . Der restliche Teil von p ist der Weg vom Typ 1.b, der auch leer sein kann.

Beweis Wir beweisen die Behauptung induktiv über die Höhe von T_P . Wir nehmen an, dass X in Normalform ist und dass T die Eigenschaft 6.5.2 erfüllt. Es sei $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ ein Rahmen in T_P und $u_s, r' < s \leq s'$ sei ein Ummarkierungsknoten der mit $\rho_{i \rightarrow j}$ markiert ist. Es sei $i \in L_P(u_{s-1})$. Weiterhin können wir nach Induktionsannahme annehmen, dass $T(u_{s-1})$ die Eigenschaft 6.5.4 schon erfüllt.

Falls $j \notin L_P(u_{s-1})$, dann ersetzen wir im Teilbaum $T(u_{r'})$ gleichzeitig jede Markierung i durch Markierung j und jede Markierung j durch Markierung i . Der Ausdruck zu dem so erhaltenen Teilbaum $T(u_{r'})$ ist weiterhin in Normalform und $T(u_{r'})$ erfüllt die Eigenschaften 6.5.2 und 6.5.4. Da i nicht an den Ummarkierungsoperationen der Knoten $u_{r'+1}, \dots, u_{s-1}$ beteiligt ist, ist der so erhaltene Ausdruck X in Normalform, beschreibt den gleichen Graphen wie zuvor und $T(u_s)$ erfüllt die Eigenschaften 6.5.2 und 6.5.4. \square

Es sei u_{s-1} der Sohn eines Ummarkierungsknotens u_s in T_P . Nach Lemma 6.5.3 können wir annehmen, dass

$$L_P(u_{s-1}) \supseteq L_P(u_s)$$

gilt. Falls $L_P(u_s) = L_P(u_{s-1})$, so gilt für die Mengen $L_{P \cap H}(u_s)$ und $L_{P \cap J}(u_s)$ die umgekehrte Inklusion, d.h.

$$L_{P \cap H}(u_{s-1}) \subseteq L_{P \cap H}(u_s) \quad \text{und} \quad L_{P \cap J}(u_{s-1}) \subseteq L_{P \cap J}(u_s),$$

da eine Ummarkierung eines Knotens mit einer Markierung aus $L_{P \cap H}(u_{s-1})$ oder aus $L_{P \cap J}(u_{s-1})$ auch eine Ummarkierung eines Knotens mit einer Markierung aus $L_P(u_{s-1})$ ist.

Damit können wir jeden 1.b-Weg p in *Wege vom Typ 2.a* und *Wege vom Typ 2.b* einteilen. Die 2.a-Wege sind die Rahmen $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ in p für die mindestens eine der folgenden Bedingungen erfüllt ist.

1. Es gibt einen Ummarkierungsknoten u_s , $r' < s \leq s'$, so dass

$$L_P(u_{s-1}) \supsetneq L_P(u_s), \quad L_{P \cap H}(u_{s-1}) \subsetneq L_{P \cap H}(u_s) \quad \text{oder} \quad L_{P \cap J}(u_{s-1}) \subsetneq L_{P \cap J}(u_s).$$

- 2.

$$L_{P \cap H}(u_0) \subsetneq L_{P \cap H}(u_1) \quad \text{oder} \quad L_{P \cap J}(u_0) \subsetneq L_{P \cap J}(u_1),$$

wobei u_0 der in T_P enthaltene Sohn des Vereinigungsknoten u_1 ist.

Diese Eigenschaft ist äquivalent zu

$$L_P(u_0) \supsetneq L_P(u_{s'}), \quad L_{P \cap H}(u_0) \subsetneq L_{P \cap H}(u_{s'}) \quad \text{oder} \quad L_{P \cap J}(u_0) \subsetneq L_{P \cap J}(u_{s'}),$$

wobei u_0 der in T_P enthaltene Sohn des Vereinigungsknoten u_1 ist.

Die 2.a Wege sind die Rahmen q der 1.b-Wege für die sich entweder die Anzahl der Markierungen in L_P verkleinert oder sich die Anzahl der Markierungen in $L_{P \cap H}$ oder in $L_{P \cap J}$ erhöht. Die 2.b-Wege sind die restlichen Teile der 1.b-Wege. In einem 2.b-Weg p sind alle Mengen $L_P(u_s)$ gleich, alle Mengen $L_{P \cap H}(u_s)$ gleich, alle Mengen $L_{P \cap J}(u_s)$ gleich und alle Mengen $L_{P \cap H \cap J}(u_s)$ für alle Knoten u_s aus p gleich, sogar für den in T_P enthaltenen Sohn u_0 des ersten Knoten u_1 . Vergleiche dazu auch Abbildung 6.11.

Für einen Rahmen $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ in einem 2.b-Weg sei

$$L_P(q) = L_P(u_{s'}), \quad L_{P \cap H}(q) = L_{P \cap H}(u_{s'}),$$

$$L_{P \cap J}(q) = L_{P \cap J}(u_{s'}) \quad \text{und} \quad L_{P \cap H \cap J}(q) = L_{P \cap H \cap J}(u_{s'}).$$

Wir gebrauchen hier q anstelle eines Knotens aus q als Argument um zu betonen, dass die obigen Mengen für alle Knoten aus q und für u_0 (dem in T_P enthaltenen Sohn von u_1) gleich sind. Es ist leicht zu sehen, dass es in jedem 1.b-Weg p höchstens $3k - 1$ Wege vom Typ 2.a und damit höchstens $3k$ Wege vom Typ 2.b gibt. Die folgende Tabelle zeigt ein Worst-Case Beispiel für $k = 3$. Die j -te Zeile gibt dabei die Markierungen des letzten Knotens u_i des j -ten 2.a-Rahmen an.

Beispiel:

j	$L_P(u_i)$	$L_{P \cap H}(u_i)$	$L_{P \cap J}(u_i)$
1	{1, 2, 3}	{1}	\emptyset
2	{1, 2, 3}	{1, 2}	\emptyset
3	{1, 2, 3}	{1, 2, 3}	\emptyset
4	{1, 2, 3}	{1, 2, 3}	{1}
5	{1, 2, 3}	{1, 2, 3}	{1, 2}
6	{1, 2, 3}	{1, 2, 3}	{1, 2, 3}
7	{1, 2}	{1, 2}	{1, 2}
8	{1}	{1}	{1}

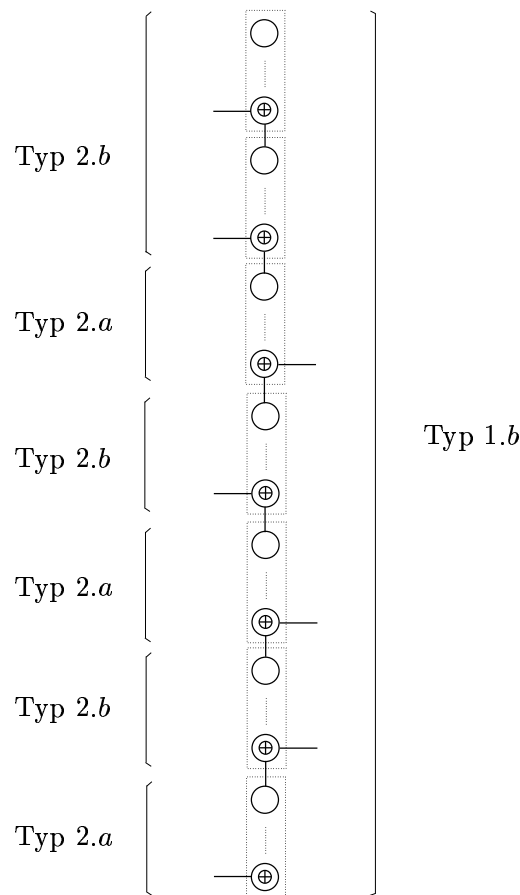


Abbildung 6.11: Jeder 1.b-Weg wird in Wege vom Typ 2.a und Wege vom Typ 2.b unterteilt. Die 2.a-Wege sind die Rahmen q der 1.b-Wege, so dass sich entweder die Anzahl der Markierungen in L_P verkleinert oder sich die Anzahl der Markierungen in $L_{P \cap H}$ oder $L_{P \cap J}$ erhöht.

Einteilung in Wege vom Typ 3.a und 3.b

Lemma 6.5.5 *Es gibt immer einen k -Ausdrucksbaum T für G , so dass der k -Ausdruck X zu T in Normalform ist und T die Eigenschaften 6.5.2, 6.5.4 und 6.5.6 erfüllt.*

Eigenschaft 6.5.6 *Es sei p ein 2.b-Weg in T_P und $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ sei ein Rahmen aus p , so dass u_s , $r' < s \leq s'$, ein Ummarkierungsknoten ist, der mit $\rho_{i \rightarrow j}$ markiert ist. Falls $i \in L_{H \cap J}(u_{s-1})$, dann ist $j \in L_{H \cap J}(u_1)$.*

Vor dem Beweis von Lemma 6.5.5 soll betont werden, dass wir zeigen dass j in der Menge $L_{H \cap J}(u_1)$ und nicht nur in der Menge $L_{H \cap J}(u_{s-1})$ enthalten ist.

Beweis Es sei X in Normalform, T erfülle die Eigenschaften 6.5.2 und 6.5.4 und $T(u_{s-1})$ erfülle zusätzlich Eigenschaft 6.5.6 für ein s , $r' < s \leq s'$. Es sei $i \in L_{H \cap J}(u_{s-1})$.

Falls $j \in L_P(q)$, dann folgt aus der Annahme $i \in L_{H \cap J}(u_{s-1})$ und der Ummarkierung $\rho_{i \rightarrow j}$ im Knoten u_s , dass $j \in L_{P \cap H \cap J}(u_s) = L_{P \cap H \cap J}(q)$ und damit $j \in L_{H \cap J}(u_1)$.

Falls $j \notin L_P(q)$ und $j \notin L_{H \cap J}(u_1)$ gilt, dann ersetzen wir in dem Teilbaum $T(u_{r'})$ gleichzeitig jede Markierung i durch Markierung j und jede Markierung j durch Markierung i . Der Ausdruck zu dem so erhaltenen Teilbaum $T(u_{r'})$ ist weiterhin in Normalform und $T(u_{r'})$ erfüllt weiterhin die Eigenschaften 6.5.2, 6.5.4 und 6.5.6. Es seien $\rho_{i_1 \rightarrow j_1}, \dots, \rho_{i_{l-1} \rightarrow j_{l-1}}$ die Ummarkierungsoperationen der Knoten $u_{r'+1}, \dots, u_{s-1}$. Die Markierung i ist nicht an diesen Ummarkierungen beteiligt, d.h. $i \notin \{i_1, \dots, i_{l-1}, j_1, \dots, j_{l-1}\}$. Die Markierung j wird durch diese Ummarkierungen nicht verändert, d.h. $j \notin \{i_1, \dots, i_{l-1}\}$ und keine dieser Ummarkierungsoperationen verändert eine Markierung aus $L_{H \cap J}(u_{r'})$ zu Markierung j im ursprünglichen Ausdruck, da der ursprüngliche Ausdrucksbaum $T(u_{r'})$ die Eigenschaft 6.5.6 erfüllt und $j \notin L_{H \cap J}(u_1)$. Somit ist der Ausdruck des so erhaltenen Baumes $T(u_s)$ in Normalform und $T(u_s)$ erfüllt nun die Eigenschaften 6.5.2, 6.5.4 und 6.5.6. \square

Für einen Knoten u_s aus T_P und eine Markierung $j \in [k]$, sei

$$\text{forb}_P(u_s, j)$$

die Menge aller Markierungen $i \in L_P(u_s)$, so dass der Graph $G(u_s)$ zwei nicht adjazente Knoten enthält die mit i bzw. j markiert sind. Ist die Menge $\text{forb}_P(u_s, j)$ leer, so hat der Graph $G(u_s)$ keinen mit j markierten Knoten oder jeder Knoten aus $G(u_s)$ der mit j markiert ist ist zu jedem Knoten aus $G(u_s)$ adjazent, der mit einer Markierung aus $L_P(u_s)$ markiert ist. Man beachte, dass die Menge $L_P(u_s)$ für alle Knoten u_s aus T_P nicht leer ist.

Es sei $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ ein Rahmen aus T_P , so dass u_1 ein Vereinigungsknoten ist. u_0 sei der in T_P enthaltene Sohn von u_1 . Falls eine der Kanteneinfügeoperationen u_r , $1 < r \leq r'$ mit $\eta_{i,j}$ markiert ist, dann ist $i \notin \text{forb}_P(u_0, j)$ und $j \notin \text{forb}_P(u_0, i)$, da $\eta_{i,j}$ sonst eine verbotene Kante zwischen zwei Knoten aus $G(u_0)$ einfügen würde.

Es sei u_s ein Ummarkierungsknoten aus T_P , der mit $\rho_{i \rightarrow j}$ markiert ist. Falls $i \notin L_P(u_{s-1})$ dann gilt

$$\text{forb}_P(u_s, j) = \text{forb}_P(u_{s-1}, j) \cup \text{forb}_P(u_{s-1}, i).$$

In $G(u_s)$ ist ein mit $o \in L_P(u_{s-1})$ markierter Knoten nicht adjazent zu einem mit j markierten Knoten, genau dann wenn in $G(u_{s-1})$ ein Knoten der mit $o \in L_P(u_{s-1})$ markiert ist, nicht adjazent zu einem mit i oder j markierten Knoten ist.

Lemma 6.5.7 *Der k -Ausdrucksbaum T erfülle die Eigenschaften 6.5.2, 6.5.4 und 6.5.6 und der k -Ausdruck X zu T sei in Normalform. Es sei p ein 2.b-Weg aus T_P , $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ sei ein Rahmen aus p und u_0 sei der in T_P enthaltene Sohn von u_1 . Falls ein Knoten u_s , $r' < s \leq s'$, mit $\rho_{i \rightarrow j}$ markiert ist und falls $i \in L_{H \cap J}(u_{s-1})$, dann ist*

$$\text{forb}_P(u_0, i) \subsetneq \text{forb}_P(u_{s'}, j) \quad \text{und} \quad \text{forb}_P(u_0, j) \subsetneq \text{forb}_P(u_{s'}, i).$$

Beweis Da i nicht an den Ummarkierungsoperationen der Knoten $u_{r'+1}, \dots, u_{s-1}$ beteiligt ist, ist i auch in der Menge $L_{H \cap J}(u_1)$ enthalten. Nach Eigenschaft 6.5.6 wissen wir, dass $j \in L_{H \cap J}(u_1)$ und damit gilt $i, j \in L_{H \cap J}(u_1)$. Es sei u'_0 der nicht in T_P enthaltene Sohn von u_1 . Wir nehmen ohne Beschränkung der Allgemeinheit an, dass $\xi(u_1) = H$ gilt. Da i und j beide in $L_J(u_1)$ enthalten sind und da die Knoten aus $G(u_0)$ alle aus U_H sind, enthält der Graph $G(u_0)$ mindestens einen Knoten der mit i markiert ist und mindestens einen Knoten der mit j markiert ist.

Falls die Markierung i oder die Markierung j an einer der Kanteneinfügeoperationen $\eta_{i',j'}$ der Knoten $u_2, \dots, u_{r'}$ beteiligt ist, dann muss die andere Markierung aus $\{i', j'\}$ in $L_P(q) - L_{H \cup J}(u_1)$ (analog zu den bisherigen Abkürzungen definieren wir auch hier $L_{H \cup J}(u_1) := L_H(u_1) \cup L_J(u_1)$) enthalten sein, d.h. nicht in $L_{H \cup J}(u_1)$ enthalten sein. Sonst wird eine verbotene Kante zwischen einem Knoten aus U_H und einem Knoten aus U_J eingefügt, da i und j beide in $L_H(u_1)$ und beide in $L_J(u_1)$ enthalten sind. Nach der Normalformeigenschaft 2.(a) fügt keine Kanteneinfügeoperation eine neue Kante zwischen zwei Knoten aus $G(u'_0)$ oder zwischen zwei Knoten aus $G(u_0)$ ein. Somit fügt jede dieser Kanteneinfügeoperationen, an der Markierung i oder j beteiligt ist, eine Kante zwischen einem mit i oder j markiertem Knoten aus $G(u'_0)$ und einem mit einer Markierung aus $L_P(q) - L_{H \cup J}(u_1)$ markiertem Knoten aus $G(u_0)$ ein.

Falls jede Markierung aus $\text{forb}_P(u_0, i)$ auch in $\text{forb}_P(u_0, j)$ enthalten ist, dann würde eine zusätzliche Ummarkierung $\rho_{i \rightarrow j}$, angewandt auf den Ausdruck der durch den Baum $T(u_0)$ definiert wird, den Graphen $G(u_{s'})$ nicht verändern. Dies widerspricht unserer Normalformeigenschaft 2.(c). Falls jede Markierung aus $\text{forb}_P(u_0, j)$ in $\text{forb}_P(u_0, i)$ enthalten ist, dann würde eine zusätzliche Markierung $\rho_{j \rightarrow i}$, angewandt auf den Ausdruck der durch den Baum $T(u_0)$ definiert wird, den Graphen $G(u_{s'})$ nicht verändern. Auch dies widerspricht unserer Normalformeigenschaft 2.(c).

Also muss es mindestens eine Markierung in $\text{forb}_P(u_0, i)$ geben, die nicht in $\text{forb}_P(u_0, j)$ enthalten ist und mindestens eine Markierung in $\text{forb}_P(u_0, j)$ geben, die nicht in $\text{forb}_P(u_0, i)$ enthalten ist. Da $\text{forb}_P(u_0, i) \subseteq \text{forb}_P(u_{s-1}, i)$ und $\text{forb}_P(u_0, j) \subseteq \text{forb}_P(u_{s-1}, j)$ gilt, folgt unsere Behauptung. \square

Wir teilen nun jeden 2.b-Weg p in Wege vom Typ 3.a und Wege vom Typ 3.b ein. Die 3.a-Wege sind die Rahmen $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ aus p für die

$$L_{H \cap J}(u_0) \subsetneq L_{H \cap J}(u_{s'}) \quad \text{oder} \quad \text{forb}_P(u_0, j) \subsetneq \text{forb}_P(u_{s'}, j)$$

für ein $j \in L_P(u_0) \cup L_{H \cap J}(u_0)$ gilt, wobei u_0 der in T_P enthaltene Sohn von u_1 ist.

Die obigen Mengen können ihre Größe innerhalb eines Rahmens in einem 2.b-Weg wie folgt ändern.

1. $L_{H \cap J}(u_0) \subseteq L_{H \cap J}(u_1)$ und $\text{forb}_P(u_0, j) \subseteq \text{forb}_P(u_1, j)$, da ein Vereinigungsknoten die Mengen $L_{H \cap J}(u_0)$ und $\text{forb}_P(u_0, j)$ nicht verkleinern kann.
2. $L_{H \cap J}(u_{r-1}) = L_{H \cap J}(u_r)$ und $\text{forb}_P(u_0, j) \subseteq \text{forb}_P(u_r, j)$ für $r = 2, \dots, r'$, da die Kanteneinfügeoperationen die Markierungen nicht verändern und keine Kanten zwischen Knoten aus $G(u_0)$ einfügen. Die Kanteneinfügeoperationen können keine Markierungen aus $\text{forb}_P(u_0, j)$ entfernen, jedoch aus $\text{forb}_P(u_1, j)$.
3. Es sei $u_s, r' < s \leq s'$ ein Ummarkierungsknoten der mit $\rho_{j \rightarrow j'}$ markiert ist.

(a) Falls $j \notin L_P(u_{s-1}) \cup L_{H \cap J}(u_{s-1})$, dann gilt $L_{H \cap J}(u_{s-1}) \subseteq L_{H \cap J}(u_s)$.

(b) Falls $j \in L_P(u_{s-1}) \cup L_{H \cap J}(u_{s-1})$, dann ist $j \in L_{H \cap J}(u_{s-1})$, da wir einen 2.b-Weg betrachten. Nach Lemma 6.5.5 ist $j' \in L_{H \cap J}(u_{s-1})$ und damit ist $L_{H \cap J}(u_{s-1}) \supsetneq L_{H \cap J}(u_s)$. Nach Lemma 6.5.7 gilt $\text{forb}_P(u_s, j) = \emptyset$, $\text{forb}_P(u_0, j) \subsetneq \text{forb}_P(u_s, j')$ und $\text{forb}_P(u_0, j') \subsetneq \text{forb}_P(u_s, j')$.

Die Größe der Mengen $\text{forb}_P(u_{s-1}, j)$ für ein $j \in L_P(u_{s-1}) \cup L_{H \cap J}(u_{s-1})$ kann nur im Fall 3.(b) kleiner werden, falls ein $j \in L_{H \cap J}(u_{s-1})$ in ein $j' \in L_{H \cap J}(u_{s-1})$ ummarkiert wird. In diesem Fall ist $\text{forb}_P(u_s, j) = \emptyset$, da $G(u_s)$ keinen Knoten enthält, der mit j markiert ist.

Wir wollen zeigen, dass sich die Anzahl der 3.a-Wege (3.a-Rahmen) durch $(k+1)^{k+1}$ beschränken lässt. Für einen Knoten u_s sei $\alpha(u_s) = (z_0, \dots, z_{k'})$ ein Vektor, wobei $k' = |L_P(u_s)|$ und z_t , $0 \leq t \leq k'$ die Anzahl der Mengen $\text{forb}_P(u_s, j)$, $j \in L_P(u_s) \cup L_{H \cap J}(u_s)$ der Größe t ist. Wir nennen einen Vektor $(z'_0, \dots, z'_{k'})$ größer als Vektor $(z_0, \dots, z_{k'})$, dies bezeichnen wir mit

$$(z'_0, \dots, z'_{k'}) > (z_0, \dots, z_{k'}),$$

falls es ein t , $0 \leq t \leq k'$ gibt, so dass $z'_t > z_t$ und $z'_{t'} = z_{t'}$, für $t' = t+1, \dots, k'$. Für jeden 3.a-Weg $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ gilt $\alpha(u_{s'}) > \alpha(u_0)$ wobei u_0 der Sohn von u_1 ist, der nicht in T_P enthalten ist. Damit ist die Anzahl der 3.a-Wege durch $(k+1)^{k+1}$ beschränkt. Diese Schranke ist jedoch nicht scharf.

Die restlichen Teile eines 2.b-Weges sind die 3.b-Wege, siehe Abbildung 6.12. In einem 3.b-Weg p sind für alle Knoten u_s aus p und für den in T_P enthaltenen Sohn des ersten Knotens aus p alle Mengen $L_{H \cap J}(u_s)$ gleich. Um dieses hervorzuheben definieren wir

$$L_{H \cap J}(q) = L_{H \cap J}(u_{s'})$$

für Rahmen $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ eines 3.b-Weges p . Die Mengen $\text{forb}_P(u_s, j)$, für $j \in L_P(u_0) \cup L_{H \cap J}(u_0)$, müssen nicht für alle Knoten u_s aus p gleich sein. In einem Rahmen $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ in einem 3.b-Weg p kann es vorkommen, dass es ein r , $1 \leq r < r'$, gibt, so dass die Menge $\text{forb}_P(u_r, j)$ eine Markierung enthält, die nicht in $\text{forb}_P(u_0, j)$ enthalten ist. Wir wissen aber, dass für alle $s = r', \dots, s'$ $\text{forb}_P(u_0, j) = \text{forb}_P(u_s, j)$ gilt.

Einteilung in Wege vom Typ 4

Zur Einteilung der 3.b-Wege in Wege vom Typ 4 benötigen wir drei weitere Lemmas. Das erste Lemma gilt bereits für Wege vom Typ 1.b, wir gebrauchen es aber nur für Wege vom Typ 3.b.

Lemma 6.5.8 *Es sei $(u_1, \dots, u_{r'}, \dots, u_{s'})$ ein Rahmen in einem 1.b-Weg p , so dass u_r , $1 < r \leq r'$, ein Kanteneinfügeknoten ist der mit $\eta_{i', j'}$ markiert ist. Falls $\xi(u_1) = H$ ($\xi(u_1) = J$), dann fügt die Operation $\eta_{i', j'}$ nur Kanten zwischen Knoten aus $G(u'_0)$ die mit Markierungen aus $L_H(u_1)$ ($L_J(u_1)$) markiert sind und Knoten aus $G(u_0)$ die nicht mit Markierungen aus $L_J(u_1)$ ($L_H(u_1)$) markiert sind ein.*

Beweis Falls $\xi(u_1) = H$ ($\xi(u_1) = J$), dann gehören alle Knoten aus $G(u'_0)$ zu U_H (U_J). Da $\eta_{i', j'}$ mindestens eine Kante zwischen einem Knoten aus $G(u'_0)$ und einem Knoten aus $G(u_0)$ einfügt und da es keine Kanten zwischen einem Knoten aus U_H und einem Knoten aus U_J gibt, muss eine der Markierungen aus $\{i', j'\}$ aus $L_H(u_1)$ ($L_J(u_1)$) sein und die andere Markierung aus $\{i', j'\}$ kann nicht aus $L_J(u_1)$ ($L_H(u_1)$) sein. \square

Das nächste Lemma zeigt, dass die Ummarkierungsoperationen in einem Rahmen $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ in einem 3.b-Weg mit $\xi(u_1) = H$ ($\xi(u_1) = J$) nur innere Knoten aus H (aus J) ummarkieren.

Lemma 6.5.9 *Es sei $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ ein Rahmen eines 3.b-Weges p , so dass u_s , $r' < s \leq s'$, ein Ummarkierungsknoten ist, der mit $\rho_{i \rightarrow j}$ markiert ist.*

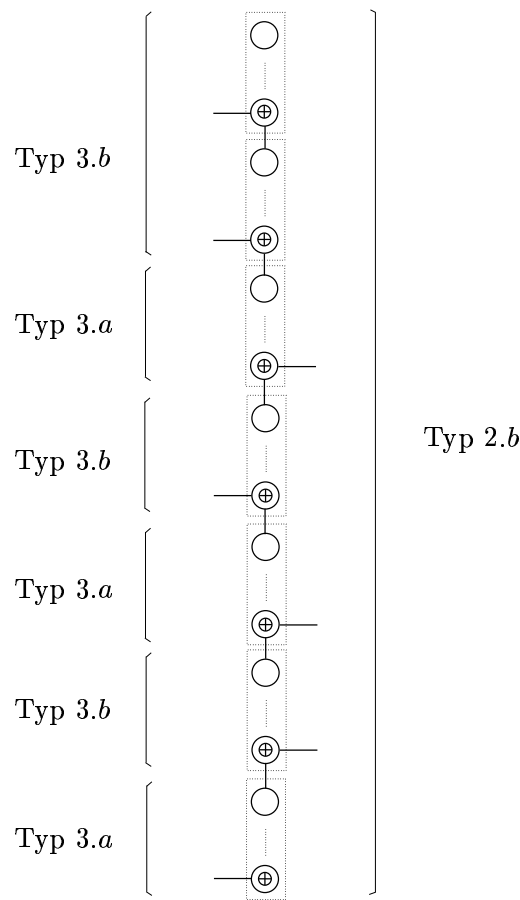


Abbildung 6.12: Jeder 2.b-Weg wird in Wege vom Typ 3.a und Wege vom Typ 3.b unterteilt.

1. Falls $\xi(u_1) = H$, dann ist $i \in L_H(u_{s-1}) - L_P(q) - L_J(u_{s-1})$ und $j \in L_H(u_{s-1})$.
2. Falls $\xi(u_1) = J$, dann ist $i \in L_J(u_{s-1}) - L_P(q) - L_H(u_{s-1})$ und $j \in L_J(u_{s-1})$.

Beweis Da in einem 3.b-Weg p , die Markierungen aus $L_P(q)$ und aus $L_{H \cap J}(q)$ nicht ummarkiert werden, kann i nur in $L_H(u_{s-1}) - L_P(q) - L_J(u_{s-1})$ oder in $L_J(u_{s-1}) - L_P(q) - L_H(u_{s-1})$ liegen.

Falls $i \in L_H(u_{s-1}) - L_P(q) - L_J(u_{s-1})$, dann ist $j \in L_H(u_{s-1})$, sonst folgt $L_{P \cap H}(u_s) \supsetneq L_{P \cap H}(u_{s-1})$ oder $L_{H \cap J}(u_s) \supsetneq L_{H \cap J}(u_{s-1})$. Beides ist in einem 3.b-Weg nicht möglich. Andererseits, wenn $i \in L_J(u_{s-1}) - L_P(q) - L_H(u_{s-1})$ erhalten wir dass $j \in L_J(u_{s-1})$, sonst folgt $L_{P \cap J}(u_s) \supsetneq L_{P \cap J}(u_{s-1})$ oder $L_{H \cap J}(u_s) \supsetneq L_{H \cap J}(u_{s-1})$.

Sei u_0 der Sohn von u_1 der in T_P liegt. Wir nehmen an, dass $\xi(u_1) = H$, $i \in L_J(u_{s-1}) - L_P(q) - L_H(u_{s-1})$ und $j \in L_J(u_{s-1})$ gilt. Dann enthält $G(u_0)$ einen Knoten der mit i markiert ist und einen Knoten der mit j markiert ist.

1. Falls j nicht an den Kanteneinfügeoperationen der Knoten u_2, \dots, u_r beteiligt ist, dann kann die Markierung i in $G(u_0)$ in j ummarkiert werden ohne dass sich der Graph $G(u_{s'})$ ändert. Dies widerspricht jedoch unserer Normalformeingenschaft 2.(c).

2. Falls j an den Kanteneinfügeoperationen der Knoten u_2, \dots, u_r beteiligt ist, dann ist j in $L_{H \cap J}(u_1) = L_{H \cap J}(q) = L_{H \cap J}(u_0)$ enthalten und somit ist $\text{forb}_P(u_{s'}, j) = \text{forb}_P(u_0, j)$ und wir können im Graphen $G(u_0)$ die Markierung i in j ummarkieren ohne den Graphen $G(u_{s'})$ zu ändern. Auch dies widerspricht unserer Normalformseigenschaft 2.(c).

Somit gilt $i \in L_H(u_{s-1}) - L_P(q) - L_J(u_{s-1})$ und $j \in L_H(u_{s-1})$. Falls $\xi(u_1) = J$, erhalten wir $i \in L_J(u_{s-1}) - L_P(q) - L_H(u_{s-1})$ und $j \in L_J(u_{s-1})$. \square

Im Beweis des nächsten Lemmas werden wir Rahmen innerhalb eines Weges vom Typ 3.b vertauschen. Es sei p ein Weg, der aus zwei aufeinanderfolgenden Rahmen besteht, d.h.

$$p = (u_1, \dots, u_{r'}, \dots, u_{s'}, u_{s'+1}, \dots, u_{r''}, \dots, u_{s''}),$$

wobei u_1 und $u_{s'+1}$ Vereinigungsknoten, $u_2, \dots, u_{r'}$ und $u_{s'+2}, \dots, u_{r''}$ Kanteneinfügeknöten und $u_{r'+1}, \dots, u_{s'}$ und $u_{r''+1}, \dots, u_{s''}$ Ummarkierungsknoten sind. Es seien u'_0 und u_0 die Söhne von u_1 , u_0 sei in T_P enthalten und u''_0 sei der nicht in T_P enthaltene Sohn von $u_{s'+1}$.

Wenn wir die beiden Rahmen in p vertauschen, erhalten wir einen neuen Weg

$$p' = (u_{s'+1}, \dots, u_{r''}, \dots, u_{s''}, u_1, \dots, u_{r'}, \dots, u_{s'}).$$

In dem zugehörigen Ausdrucksbaum hat der Vereinigungsknoten $u_{s'+1}$ die beiden Söhne u''_0 und u_0 und der Vereinigungsknoten u_1 hat die beiden Söhne u'_0 und $u_{s''}$. Die Links-Rechts Ordnung der Söhne von u_1 und $u_{s'+1}$ wird dabei nicht verändert. Falls u'_0 der linke Sohn (rechte Sohn) von u_1 im ursprünglichen Ausdrucksbaum ist, dann ist u'_0 der linke Sohn (rechte Sohn) von u_1 im neuen Ausdrucksbaum und falls u''_0 der linke Sohn (rechte Sohn) von $u_{s'+1}$ im ursprünglichen Ausdrucksbaum ist dann ist u''_0 der linke Sohn (rechte Sohn) von $u_{s'+1}$ im neuen Ausdrucksbaum.

Die Umordnung ändert den Ausdruck der durch den ursprünglichen Ausdrucksbaum $T(u_{s''})$ definiert ist wie folgt, siehe dazu Abbildung 6.13. Es seien X_1, X_2, X_3 die Ausdrücke die durch die Ausdrucksbäume $T(u'_0)$, $T(u_0)$ und $T(u''_0)$ definiert werden. Ohne Beschränkung der Allgemeinheit sei u'_0 der linke Sohn von u_1 und u''_0 der rechte Sohn von $u_{s'+1}$. Es seien $\eta_{i_2, j_2}, \dots, \eta_{i_{r'}, j_{r'}}$ die Kanteneinfügeoperationen der Knoten $u_2, \dots, u_{r'}$ und $\rho_{i_{r'+1}, j_{r'+1}}, \dots, \rho_{i_{s'}, j_{s'}}$ seien die Ummarkierungsoperationen der Knoten $u_{r'+1}, \dots, u_{s'}$, es seien $\eta_{i_{s'+2}, j_{s'+2}}, \dots, \eta_{i_{r''}, j_{r''}}$ die Kanteneinfügeoperationen der Knoten $u_{s'+2}, \dots, u_{r''}$ und $\rho_{i_{r''+1}, j_{r''+1}}, \dots, \rho_{i_{s''}, j_{s''}}$ seien die Ummarkierungsoperationen der Knoten $u_{r''+1}, \dots, u_{s''}$. Dann ist der ursprüngliche Ausdruck der durch $T(u_{s''})$ definiert ist

$$\rho_{i_{s''} \rightarrow j_{s''}} (\dots \rho_{i_{r''+1} \rightarrow j_{r''+1}} (\eta_{i_{r''}, j_{r''}} (\dots \eta_{i_{s'+2}, j_{s'+2}} (Y \oplus X_3) \dots)) \dots),$$

wobei

$$Y = \rho_{i_{s'} \rightarrow j_{s'}} (\dots \rho_{i_{r'+1} \rightarrow j_{r'+1}} (\eta_{i_{r'}, j_{r'}} (\dots \eta_{i_2, j_2} (X_1 \oplus X_2) \dots)) \dots).$$

Der Ausdruck des neuen Ausdrucksbaumes $T(u_{s'})$, den wir nach dem Vertauschen der beiden Rahmen erhalten, ist

$$\rho_{i_{s'} \rightarrow j_{s'}} (\dots \rho_{i_{r'+1} \rightarrow j_{r'+1}} (\eta_{i_{r'}, j_{r'}} (\dots \eta_{i_2, j_2} (X_1 \oplus Y') \dots)) \dots),$$

wobei

$$Y' = \rho_{i_{s''} \rightarrow j_{s''}} (\dots \rho_{i_{r''+1} \rightarrow j_{r''+1}} (\eta_{i_{r''}, j_{r''}} (\dots \eta_{i_{s'+2}, j_{s'+2}} (X_2 \oplus X_3) \dots)) \dots).$$

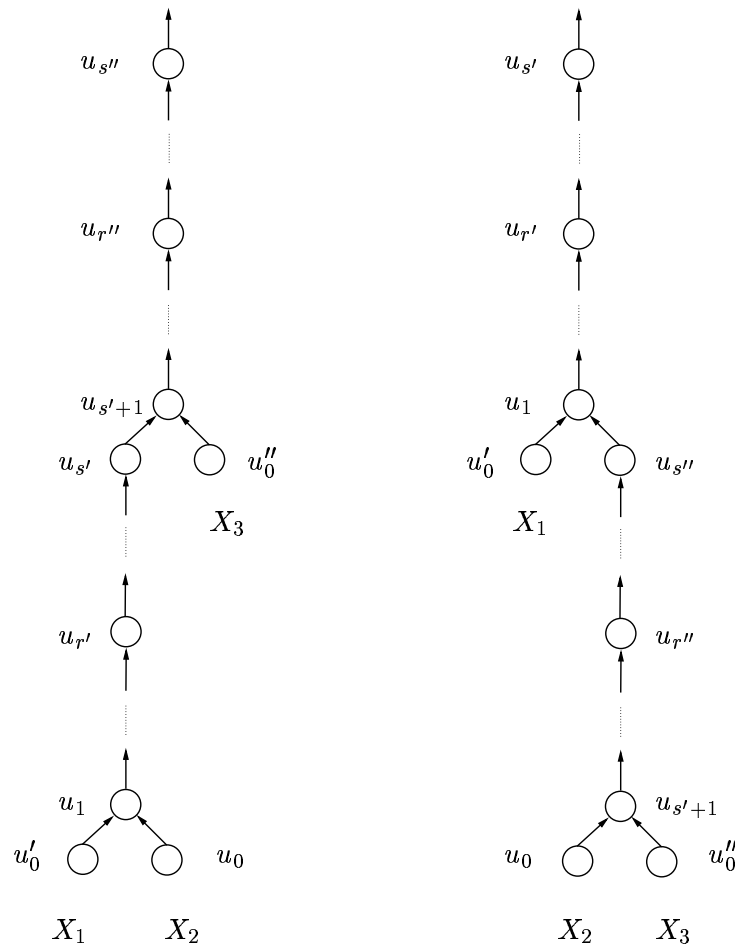


Abbildung 6.13: Die Teilbäume vor und nach dem Vertauschen zweier Rahmen

Der neue Ausdruck und der ursprüngliche Ausdruck müssen nicht äquivalent sein, die Reihenfolge der Blätter in den Bäumen bleibt jedoch unverändert.

Wir benötigen im Weiteren die folgenden Bezeichnungen. Es sei u_s ein Knoten aus T . Die Markierungen $L_H(u_s) - L_P(u_s) - L_J(u_s)$ und $L_J(u_s) - L_P(u_s) - L_H(u_s)$ heißen die *freien Markierungen* in $G(u_s)$. In einem Weg vom Typ 3.b werden nach Lemma 6.5.9 nur freie Markierungen ummarkiert. Die Knoten aus $G(u_s)$, die mit freien Markierungen markiert sind, heißen *freie Knoten* aus $G(u_s)$.

Lemma 6.5.10 *Es gibt immer einen k -Ausdruck X in Normalform, so dass der Baum T die Eigenschaften 6.5.2, 6.5.4, 6.5.6 und zusätzlich noch Eigenschaft 6.5.11 erfüllt.*

Eigenschaft 6.5.11 *Jeder 3.b-Weg p kann in höchstens $3 \cdot 2^{2(k+1)}$ Wege p' unterteilt werden, so dass für alle Rahmen $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ aus p' $\xi(u_1) = H$ oder für alle Rahmen $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ aus p' $\xi(u_1) = J$ gilt.*

Beweis Es seien q_1, \dots, q_t die Rahmen in einem 3.b-Weg p , d.h. $p = q_1 \odot \dots \odot q_t$, wobei die Operation \odot die Hintereinanderschaltung zweier Rahmen beschreibe.

Für einen Rahmen $q = (u_1, \dots, u_{s'})$ aus p sei $\xi(q) = \xi(u_1)$,

$$\begin{aligned} \text{frei}_H(q) &= |L_H(u_{s'}) - L_P(u_{s'}) - L_J(u_{s'})|, \\ \text{frei}_J(q) &= |L_J(u_{s'}) - L_P(u_{s'}) - L_H(u_{s'})| \end{aligned}$$

und

$$\begin{aligned} \min_H(p) &= \min\{\text{frei}_H(q_1), \dots, \text{frei}_H(q_t)\} \quad \text{und} \\ \min_J(p) &= \min\{\text{frei}_J(q_1), \dots, \text{frei}_J(q_t)\}. \end{aligned}$$

Es seien r_1, r_2 , $1 \leq r_1 \leq r_2 \leq t$, so gewählt, dass $r_2 - r_1$ maximal ist und entweder

$$\begin{aligned} \text{frei}_H(q_{r_1}) &= \min_H(p) \quad \text{und} \quad \text{frei}_J(q_{r_2}) = \min_J(p) \quad \text{oder} \\ \text{frei}_J(q_{r_1}) &= \min_J(p) \quad \text{und} \quad \text{frei}_H(q_{r_2}) = \min_H(p). \end{aligned}$$

Falls q_{i_1}, \dots, q_{i_n} , $1 \leq i_1 < i_2 < \dots < i_n \leq t$, die Rahmen sind für die

$$\text{frei}_H(q_{r_{i_1}}) = \text{frei}_H(q_{r_{i_2}}) = \dots = \text{frei}_H(q_{r_{i_n}}) = \min_H(p)$$

und falls q_{j_1}, \dots, q_{j_m} , $1 \leq j_1 < j_2 < \dots < j_m \leq t$, die Rahmen sind für die

$$\text{frei}_J(q_{r_{j_1}}) = \text{frei}_J(q_{r_{j_2}}) = \dots = \text{frei}_J(q_{r_{j_m}}) = \min_J(p),$$

dann ist entweder $r_1 = i_1$ und $r_2 = j_m$ oder $r_1 = j_1$ und $r_2 = i_n$.

Wir teilen den Weg p in drei Teile $p_{\text{vorne}}, p_{\text{mitte}}, p_{\text{hinten}}$ ein, so dass $p = p_{\text{vorne}} \odot p_{\text{mitte}} \odot p_{\text{hinten}}$. Der Teilweg p_{mitte} beginnt mit dem Rahmen q_{r_1} und endet mit dem Rahmen q_{r_2} , siehe Abbildung 6.14.

Wenn die Wege p_{vorne} und p_{hinten} nicht leer sind, so werden sie in der gleichen Weise wie p unterteilt. Da

$$\begin{aligned} \min_J(p_{\text{vorne}}) &> \min_J(p) \quad \text{und} \quad \min_H(p_{\text{hinten}}) > \min_H(p) \quad \text{oder} \\ \min_H(p_{\text{vorne}}) &> \min_H(p) \quad \text{und} \quad \min_J(p_{\text{hinten}}) > \min_J(p), \end{aligned}$$

erhält man mit dieser Unterteilung höchstens $2^{2(k+1)}$ viele Wege p_{mitte} .

Wir nehmen an, dass $\text{frei}_H(q_{r_1}) = \min_H(p)$ und dass $\text{frei}_J(q_{r_2}) = \min_J(p)$. Den Fall wo $\text{frei}_J(q_{r_1}) = \min_J(p)$ und $\text{frei}_H(q_{r_2}) = \min_H(p)$ gilt, beweist man analog. Dann gilt $\xi(q_{r_1}) = H$ und $\xi(q_{r_2}) = J$. Wir vertauschen die Rahmen im Weg p_{mitte} , so dass im neuen Weg q_{r_1} der erste Rahmen ist, dann alle Rahmen q mit $\xi(q) = J$ folgen und zuletzt die noch verbleibenden Rahmen q mit $\xi(q) = H$. Wenn wir alle Rahmen q aus p_{mitte} , in denen $\xi(q) = J$ gilt, an den Anfang schieben, dann werden automatisch die Rahmen q aus p_{mitte} , in denen $\xi(q) = H$ gilt (außer dem Rahmen q_{r_1}), an das Ende von p_{mitte} geschoben. Durch diese Umordnung erhalten wir höchstens $3 \cdot 2^{2(k+1)}$ Wege in denen alle Rahmen q die gleichen $\xi(q)$ Werte gleich haben.

Die Anordnung der Rahmen q , für die $\xi(q) = J$ bzw. $\xi(q) = H$ gilt, wird dabei nicht geändert, d.h. sie ist gleich der Anordnung dieser Rahmen im alten Weg p_{mitte} . Die Anordnung der Knoten in einem Rahmen wird bei der Umordnung der Rahmen ebenfalls nicht verändert. Um sicherzustellen, dass der so erhaltene Ausdruck äquivalent zu dem ursprünglichem Ausdruck ist, markieren wir die freien Markierungen wie folgt um.

Für jeden Knoten u_s des neuen Wegs p_{mitte} definieren wir schrittweise eine Bijektion $b_{u_s} : [k] \rightarrow [k]$ um in den Operationen für den Graphen $G(u_s)$ die Markierung $b_{u_s}(i)$ anstatt der Markierung i zu verwenden. Für alle Markierungen $i \in L_P(u_s) \cup L_{H \cap J}(u_s)$ wird $b_{u_s}(i) = i$ gelten, da diese Markierungen in einem Weg vom Typ 3.b nicht ummarkiert werden.

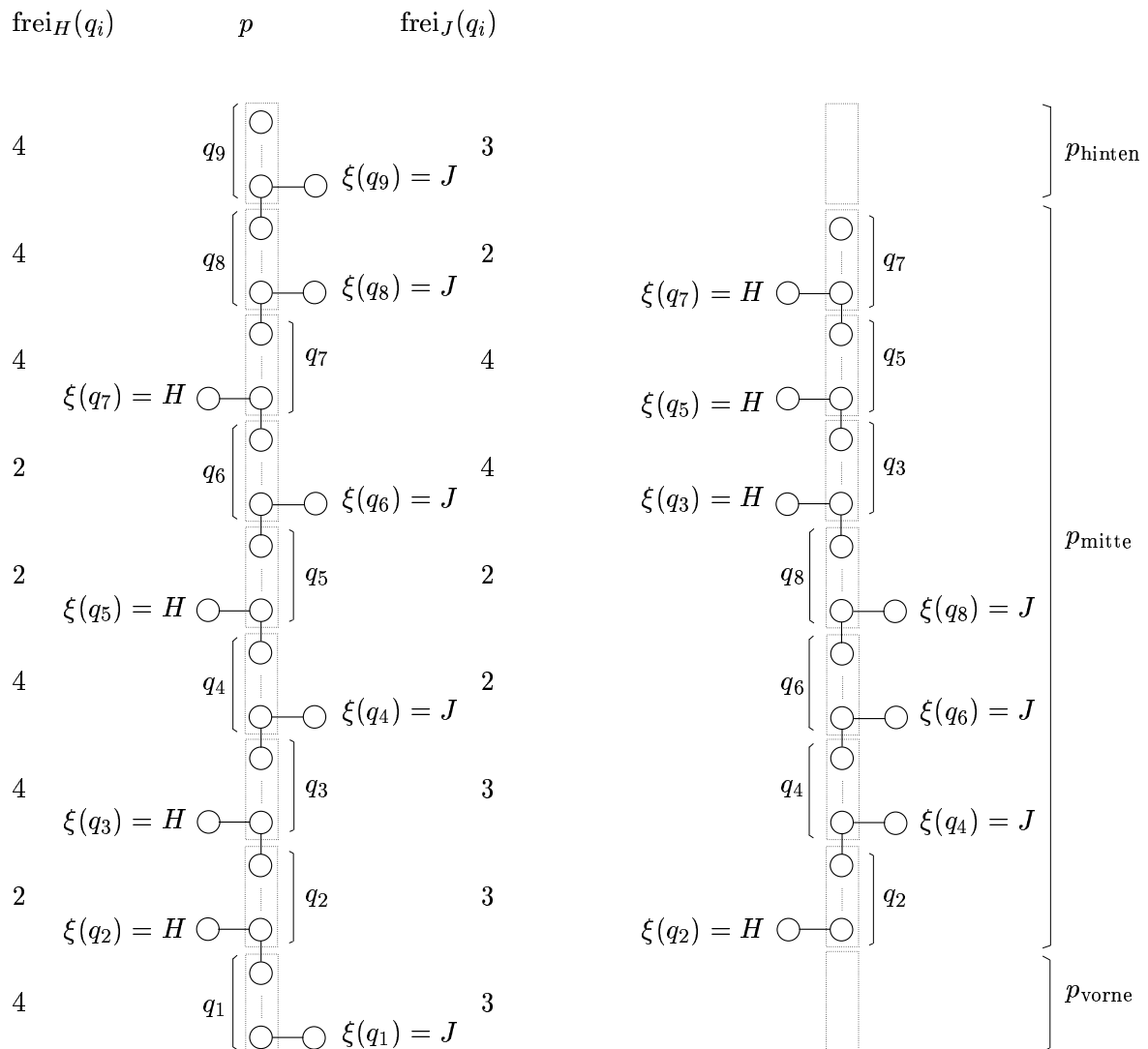


Abbildung 6.14: Die Aufteilung eines Weges p in drei Teile $p_{\text{vorne}}, p_{\text{mitte}}, p_{\text{hinten}}$, mit $\min_H(p) = 2$, $\min_J(p) = 2$, $r_1 = 2$ und $r_2 = 8$.

Die Bijektionen für die Knoten aus q_{r_1} sind die Identitäten. Die weiteren Bijektionen werden schrittweise in Abhängigkeit von den Operationen der Knoten auf dem neuen Weg p_{mitte} definiert.

Wir betrachten die Knoten u_s auf dem neuen Weg p_{mitte} in der gegebenen Reihenfolge und beginnen mit dem Vater des letzten Knotens im Rahmen q_{r_1} .

1. Falls u_s ein Vereinigungsknoten ist, dann seien u'_{s-1} und u_{s-1} die beiden Söhne von u_s , wobei u'_{s-1} nicht in T_P liegt. Die Bijektion b_{u_s} ist die Bijektion $b_{u_{s-1}}$ des Sohnes u_{s-1} .

Dann ersetzen wir im Ausdruck, der durch den Teilbaum $T(u'_{s-1})$ definiert wird, jede Markierung i durch $b_{u_s}(i)$ und prüfen ob es einen Knoten w in $G(u'_{s-1})$ gibt der im ursprünglichen Baum $T(u_s)$ ein freier Knoten war, im neuen Baum $T(u_s)$ jedoch nicht

mehr frei ist.

Falls es einen solchen Knoten w gibt der ursprünglich mit i markiert war und nun mit $b_{u_s}(i)$ markiert ist, dann wählen wir eine beliebige Markierung j , so dass $b_{u_s}(j)$ bisher noch nicht gebraucht wurde, d.h. so dass $b_{u_s}(j) \notin L_P(u_s) \cup L_H(u_s) \cup L_J(u_s)$ bezüglich des bisher aufgebauten Ausdrucks des Baumes $T(u_s)$ gilt. Wir definieren $b_{u_s}(i) := b_{u_s}(j)$ und $b_{u_s}(j) := b_{u_s}(i)$ und ersetzen im Ausdrucksbaum $T(u_{s-1})$ gleichzeitig jede Markierung $b_{u_s}(i)$ durch $b_{u_s}(j)$ und jede Markierung $b_{u_s}(j)$ durch $b_{u_s}(i)$.

2. Falls u_s ein mit $\eta_{i,j}$ markierter Kanteneinfügeknoten ist, dann ist die Bijektion b_{u_s} gleich der Bijektion $b_{u_{s-1}}$ und der Knoten u_s wird mit der Operation $\eta_{b_{u_s}(i), b_{u_s}(j)}$ markiert.
3. Falls u_s ein mit $\rho_{i \rightarrow j}$ markierter Ummarkierungsknoten ist, dann ist die Bijektion b_{u_s} gleich der Bijektion $b_{u_{s-1}}$ und der Knoten u_s wird mit der Operation $\rho_{b_{u_s}(i) \rightarrow b_{u_s}(j)}$ markiert.

Für den letzten Knoten u_s des neuen Weges p_{mitte} wird eine zusätzliche Ummarkierung des durch $T(u_s)$ erhaltenen Ausdrucks durchgeführt, so dass die freien Knoten in $G(u_s)$ genauso markiert sind wie die freien Knoten in dem Graph der durch den letzten Knoten des ursprünglichen Weges p_{mitte} definiert wurde.

Alle diese Ummarkierungen sind möglich, da für alle Knoten u_s im ersten Teil von p_{mitte} $\text{frei}_H(u_s) = \min_H(p)$ gilt und für alle Knoten u_s im zweiten Teil von p_{mitte} $\text{frei}_J(u_s) = \min_J(p)$ gilt. Somit gibt es immer genügend ungebrauchte Markierungen um die freien Knoten umzumarkieren. Man beachte, dass die Mengen $L_P(u_s)$ und $L_{H \cap J}(u_s)$ für die Knoten des Weges p_{mitte} gleich bleiben.

Es bleibt noch zu zeigen, dass der neue Ausdruck äquivalent zum bisherigen Ausdruck ist. Es sei $q = (u_1, \dots, u_{r'}, \dots, u_{s'})$ ein Rahmen im ursprünglichen Weg p_{mitte} mit $\xi(q) = H$. Der Fall $\xi(q) = J$ verläuft analog und ist sogar einfacher. Der Rahmen q kann durch die Umordnung an das Ende von p_{mitte} verschoben werden. Es seien u_0 und u'_0 die beiden Söhne von u_1 , wobei u_0 in T_P enthalten sei. Der Knoten u'_0 ist auch im neuen Ausdrucksbaum ein Sohn von u_1 , da die Söhne der Vereinigungsknoten die nicht in T_P liegen durch die Umordnung der Rahmen nicht verändert werden.

Wir betrachten eine Kanteneinfügeoperation $\eta_{i,j}$ eines Knoten u_r , $1 < r \leq r'$, im Rahmen q im ursprünglichen Ausdrucksbaum. Nach Lemma 6.5.8 wissen wir, dass diese Kanteneinfügeoperation $\eta_{i,j}$ des Knoten u_r nur Kanten zwischen Knoten aus $G(u'_0)$ und Knoten aus $G(u_0)$ einfügt. Weiterhin wissen wir, dass eine der Markierungen i, j in der Menge $L_H(u_1)$ enthalten ist und die andere Markierung nicht in der Menge $L_J(u_1)$ enthalten ist. Ohne Beschränkung der Allgemeinheit nehmen wir an, dass $j \in L_H(u_1)$ und $i \notin L_J(u_1)$.

Die Umordnung der Rahmen ändert die Reihenfolge der Blätter im Ausdrucksbaum nicht, vergleiche Abbildung 6.13. Weiterhin wird die Ordnung der Rahmen q im Weg p_{mitte} mit gleichen $\xi(q)$ nicht verändert. Da $\eta_{i,j}$ nur Kanten zwischen Knoten aus U_H und Knoten aus $U_H \cup U_P$ einfügt, werden alle diese Kanten auch durch die entsprechende Kanteneinfügeoperation $\eta_{b_{u_s}(i), b_{u_s}(j)}$ im neuen Ausdrucksbaum eingefügt.

Wir nehmen nun an, dass die Kanteneinfügeoperation $\eta_{b_{u_s}(i), b_{u_s}(j)}$ im neuen Ausdruck eine Kante einfügt, die im Graphen der durch den ursprünglichen Ausdruck definiert wurde nicht enthalten ist. Dann muss einer der Knoten dieser Kante in U_J liegen. Dieser Knoten kann nur mit $b_{u_s}(j)$ markiert sein, da $i \notin L_J(u_1)$ für den Knoten u_1 aus dem ursprünglichen Ausdrucksbaum und nach unserer Ummarkierung gilt auch $b_{u_r}(i) \notin L_J(u_1)$ für den Knoten

u_1 des neuen Ausdrucksbaumes. Es gilt nun $b_{u_r}(j) \in L_{H \cap J}(u_1)$, $b_{u_r}(i) \in L_P(u_1)$ und $b_{u_r}(i) \notin \text{forb}_P(u_s, b_{u_r}(j))$, für Knoten u_1 aus dem neuen Ausdrucksbaum. Da alle Mengen $L_P(u_s)$ für alle Knoten u_s aus p_{mitte} gleich sind, alle Mengen $L_{H \cap J}(u_s)$ für alle Knoten u_s aus p_{mitte} gleich sind und da alle Mengen $\text{forb}_P(u_s, j)$ für alle Knoten u_s aus p_{mitte} gleich sind und da diese Markierungen durch unsere Ummarkierungsprozedur nicht verändert wurden wissen wir, dass alle Kanten die durch $\eta_{b_{u_s}(i), b_{u_s}(j)}$ erzeugt werden auch im Graph, der durch den ursprünglichen Ausdruck definiert wird, enthalten sind. Dies widerspricht unserer Annahme.

Somit sind der ursprüngliche und der neue Ausdruck äquivalent. Beachte, dass die Normalformeneigenschaft und die Eigenschaften 6.5.2, 6.5.4 und 6.5.6 durch die Umordnung der Rahmen nicht verletzt werden. \square

Mit Lemma 6.5.10 können wir jeden 3.b-Weg in $3 \cdot 2^{2(k+1)}$ Wege vom Typ 4 unterteilen. Die Wege vom Typ 4 sind die Teile der Wege p_{mitte} in denen für alle Rahmen q die Werte $\xi(q)$ gleich sind.

Der Verbindungstyp von H

Wir haben die Wege in T_P nun folgendermaßen unterteilt. Es gibt

1. höchstens $2l - 1$ Wege vom Typ 1.a,
2. höchstens $(2l - 1) \cdot (3k - 1)$ Wege vom Typ 2.a,
3. höchstens $(2l - 1) \cdot 3k \cdot (k + 1)^{k+1}$ Wege vom Typ 3.a und
4. höchstens $(2l - 1) \cdot 3k \cdot (k + 1)^{k+1} \cdot 3 \cdot 2^{2(k+1)}$ Wege vom Typ 4.

Jeder Knoten aus T_P der kein Blatt ist, ist in genau einem dieser Wege vom Typ 1.a, 2.a, 3.a oder 4 enthalten. Die Wege vom Typ 1.a, 2.a und 3.a haben höchstens $\binom{k}{2} + k$ Knoten, da diese Wege Rahmen sind. Für alle Rahmen $q = (u_1, \dots, u_{s'})$ in einem Weg vom Typ 4 sind sowohl alle $\xi(u_1)$ Werte gleich und auch alle Mengen $\text{forb}_P(u_{s'}, j)$ für alle $j \in L_P(q) \cup L_{H \cap J}(p)$ gleich. Für einen Knoten u_s aus T_P sei $L_P(u_s)$ die Menge der *terminalen Markierungen*, $L_H(u_s)$ die Menge der *inneren Markierungen* und $L_J(u_s)$ die Menge der *äußeren Markierungen* für den Knoten u_s .

Wir ersetzen nun jeden 4-Weg $p = (u_1, \dots, u_{s'})$ aus T_P , der aus mehr als einem Rahmen besteht, durch einen sogenannten *Brückenknoten* v . Es sei u_0 der Sohn von u_1 der in T_P enthalten ist und $u_{s'+1}$ der Vater von $u_{s'}$ in T_P . Der Weg $p = (u_0, u_1, \dots, u_{s'}, u_{s'+1})$ wird durch den Weg $(u_0, v, u_{s'+1})$ ersetzt. Der Knoten v wird ein *innerer* Brückenknoten genannt, falls $\xi(u_1) = H$ und ein *äußerer* Brückenknoten genannt, falls $\xi(u_1) = J$. Jeder Brückenknoten repräsentiert einen 4-Weg. In einer weiteren Ersetzung können die Knoten u_0 und $u_{s'+1}$ Brückenknoten sein. An jedem Brückenknoten speichern wir die ob es sich um einen inneren oder einen äußeren Brückenknoten handelt, weiterhin speichern wir die Menge der terminalen Markierungen $L_P(u_{s'})$, die Menge der inneren Markierungen $L_H(u_{s'})$, die Menge der äußeren Markierungen $L_J(u_{s'})$ und alle Paare $(\text{forb}_P(u_{s'}, j), j)$, für alle $j \in L_P(u_{s'}) \cup L_{H \cap J}(u_{s'})$.

Ein Vereinigungsknoten u_1 wird ein *innerer Vereinigungsknoten* genannt, falls $\xi(u_1) = H$ und ein *äußerer Vereinigungsknoten* genannt, falls $\xi(u_1) = J$. An jedem Vereinigungsknoten u_s in T_P für den $\xi(u_s)$ definiert ist, speichern wir ob u_s ein innerer oder ein äußerer Vereinigungsknoten ist.

An jedem Knoten u_s in T_P der kein Brückenknoten ist speichern wir die Cliquesweite Operation, die Menge aller terminalen Markierungen $L_P(u_s)$, die Menge aller inneren Markierungen $L_H(u_s)$, die Menge aller äußeren Markierungen $L_J(u_s)$ und alle Paare $(\text{forb}_P(u_{s'}, j), j)$ für alle $j \in L_P(u_s) \cup L_{H \cap J}(u_s)$. Falls ein Blatt u_s aus T_P einem Knoten aus G entspricht der durch die Identifizierung des i -ten terminalen Knotens aus H mit dem i -ten terminalen Knotens aus J hervorgeht, so markieren wir das Blatt u_s zusätzlich mit dem Index i . Der so erhaltene Baum C wird ein *Verbindungsbaum* für den k -markierten l -terminalen Graphen H genannt.

Die Menge aller paarweise verschiedenen Verbindungsbäume für H bezüglich aller k -markierter l -terminaler Graphen J wird der *Verbindungstyp* für H genannt. Zwei Verbindungsbäume C_1, C_2 für H heißen *äquivalent*, falls es eine Bijektion b zwischen den Knoten aus C_1 und den Knoten aus C_2 gibt, so dass

1. der Knoten u_{s-1} ein Sohn (linker Sohn, rechter Sohn) von Knoten u_s in C_1 ist, genau dann wenn $b(u_{s-1})$ ein Sohn (linker Sohn, rechter Sohn) von Knoten $b(u_s)$ in C_2 ist,
2. der Knoten u_s aus C_1 ein äußerer oder innerer Vereinigungsknoten ist, genau dann wenn der Knoten $b(u_s)$ aus C_2 ein äußerer bzw. innerer Vereinigungsknoten ist,
3. der Knoten u_s aus C_1 ein äußerer oder innerer Brückenknoten ist, genau dann wenn der Knoten $b(u_s)$ aus C_2 ein äußerer bzw. innerer Brückenknoten ist,
4. der Knoten u_s aus C_1 und der Knoten $b(u_s)$ aus C_2 die gleichen terminalen Markierungsmengen, die gleichen inneren Markierungsmengen und die gleichen äußeren Markierungsmengen und die gleichen Paare $(\text{forb}_P(u_{s'}, j), j)$ und die gleiche Cliquesweiteoperation speichern,
5. der Knoten u_s aus C_1 und der Knoten $b(u_s)$ aus C_2 denselben Index speichern, falls u_s und $b(u_s)$ Blätter sind, die durch die Identifizierung zweier terminaler Knoten entstanden sind.

Die beiden Begriffe Verbindungsbaum und Verbindungstyp sind stets bezüglich der Grapheneigenschaft Cliquesweite höchstens k definiert. Wir lassen diesen Zusatz zur besseren Lesbarkeit gelegentlich weg.

6.6 Ergebnis

Aus dem nächsten Theorem wird unmittelbar die Hauptaussage dieses Kapitels folgen.

Theorem 6.6.1 *Wenn zwei k -markierte l -terminale Graphen den gleichen Verbindungstyp bezüglich der Grapheneigenschaft Cliquesweite höchstens k haben, dann sind die Graphen bezüglich der Grapheneigenschaft Cliquesweite höchstens k ersetzbar.*

Beweis Es seien H_1 und H_2 zwei k -markierte l -terminale Graphen die den gleichen Verbindungstyp haben. Es sei J ein k -markierter l -terminaler Graph, so dass $H_1 \circ J$ Cliquesweite höchstens k hat. Wir werden zeigen, dass der Graph $H_2 \circ J$ ebenfalls Cliquesweite höchstens k hat. Dies zeigt, dass H_1 und H_2 bezüglich der Eigenschaft Cliquesweite höchstens k ersetzbar sind.

Es sei T_1 ein k -Ausdrucksbaum für $H_1 \circ J$ der einen Verbindungsbaum C definiert. Es sei $T_{1,P}$ der Teilbaum von T_1 der durch die Blätter in T_1 , die den identifizierten terminalen Knoten aus H_1 und J entsprechen, und allen Knoten auf den Wegen von diesen Blätter zur Wurzel von T_1 induziert wird.

Da H_1 und H_2 den gleichen Verbindungstyp haben, ist C auch ein Verbindungsbaum für H_2 bezüglich eines k -markierten l -terminalen Graphen J' . Es sei T' ein k -Ausdrucksbaum für den k -markierten Graphen $H_2 \circ J'$. T' definiert den gleichen Verbindungsbaum C . Es sei T'_P der Teilbaum von T' der durch die Blätter von T' die zu identifizierten terminalen Knoten von H_2 und J' gehören und allen Wegen dieser Blätter zur Wurzel von T' induziert wird.

Da T_1 und T' den gleichen Verbindungsbaum C definieren, gibt es Bijektion zwischen den Knoten in T_1 , T' und C . Für einen Knoten u aus C schreiben wir nun u^C um anzudeuten, dass u ein Knoten aus C ist. Ist v der entsprechende Knoten in T_1 schreiben wir u^{T_1} anstatt v . Der entsprechende Knoten in T' wird analog mit $u^{T'}$ bezeichnet. Die gleiche Notation gebrauchen wir auch für Rahmen und Wege.

Wir wollen einen neuen k -Ausdrucksbaum T_2 aus T_1 und T' konstruieren, so dass T_2 den Graphen $H_2 \circ J$ definiert. Dazu beginnen wir mit einer Kopie T'_1 des Ausdrucksbaumes T_1 . $T'_{1,P}$ sei bezüglich T'_1 definiert genau wie wir $T_{1,P}$ bezüglich T_1 definiert haben. Es sei u_1^C ein innerer Vereinigungsknoten und $u_0^{T'_1}$ der Sohn von $u_1^{T'_1}$ der nicht im Baum $T'_{1,P}$ liegt und es sei $u_0^{T'}$ der Sohn von $u_1^{T'}$ der nicht im Baum T'_P liegt. Nach unserer Notation ist es nun klar aus welchen Bäumen die Knoten sind. Für jeden solchen Knoten u_0^C ersetzen wir in der Kopie von T'_1 die Teilbäume $T'_1(u_0^{T'_1})$ durch die Teilbäume $T'(u_0^{T'})$. Es sei u^C ein innerer Brückenknoten, p^{T_1} sei der entsprechende 4-Weg in T_1 und $p^{T'}$ sei der entsprechende 4-Weg in T' . Für jeden solchen Knoten ersetzen wir in unserer bisherigen Kopie T'_1 von T_1 den 4-Weg p^{T_1} durch den 4-Weg $p^{T'}$. Bei dieser Ersetzung werden auch alle Teilbäume der Söhne der Vereinigungsknoten von p^{T_1} und $p^{T'}$, die nicht in $T'_{1,P}$ bzw. nicht in T'_P enthalten sind, betrachtet. Den so erhaltenen Baum bezeichnen wir mit T_2 . Es ist klar, dass T_2 ein k -Ausdrucksbaum ist.

Es bleibt zu zeigen, dass der k -Ausdrucksbaum T_2 den Graphen $H_2 \circ J$ definiert. Es sei $T_{2,P}$ der Teilbaum von T_2 der durch die Blätter von T_2 , die identifizierte terminale Knoten aus H_2 und J darstellen, und allen Wegen von diesen Knoten zur Wurzel von T_2 induziert wird.

Zuerst zeigen wir, dass die Knoten im k -markierten Graph $H_2 \circ J$ genauso markiert sind wie in dem k -markierten Graph der durch den k -Ausdrucksbaum T_2 beschrieben wird. Es gibt eine Bijektion zwischen den Knoten aus $H_2 \circ J$ und den Knoten des Graphen der durch T_2 beschrieben wird, da T_2 aus den Bäumen T_1 und T' konstruiert wurde, die die Graphen J bzw. H_2 definieren. Es seien T_{2,H_2} und $T_{2,J}$ die Teilbäume von T_2 die durch die Blätter der Knoten aus H_2 bzw. Knoten aus J und allen Wegen von diesen Blättern zur Wurzel der Bäume induziert werden. Die Knoten aus H_2 sind im Graph $H_2 \circ J$ genauso markiert wie in dem Graphen der durch den k -Ausdrucksbaum T_2 beschrieben wird, da die Knoten nur durch Operationen aus T_{2,H_2} ummarkiert werden, die nicht zu äußeren 4-Wegen aus $T_{2,P}$ gehören. (In diesem Zusammenhang bedeutet innerer 4-Weg dass der Weg aus T_1 bzw. T' kopiert wurde.) Das gleiche gilt für die Knoten aus J , da diese Knoten nur durch Operationen aus $T_{2,J}$ ummarkiert werden, die nicht zu inneren 4-Wegen aus $T_{2,P}$ gehören.

Nun werden wir zeigen, dass alle Kanten aus $H_2 \circ J$ auch im Graphen der durch den Baum T_2 definiert wird, enthalten sind. Es seien T'_{H_2} und $T_{1,J}$ die Teilbäume von T' bzw. von T_1 , die durch die Blätter aus dem Graphen H_2 bzw. aus dem Graphen J und durch alle Wege von diesen Blättern zur Wurzel des Baumes T' bzw. T_1 , induziert werden. Es sei e eine Kante aus

$H_2 \circ J$. Sind beide Endknoten von e aus H_2 oder beide Endknoten aus J , so wird e durch eine Kanteneinfügeoperation $u_s^{T'}$ bzw. $u_s^{T_1}$, die auch im Baum T'_{H_2} bzw. im Baum $T'_{1,J}$ enthalten ist, erzeugt. Aus dem Aufbau von T_2 folgt nun, dass der Knoten $u_s^{T_2}$ in T_2 enthalten ist und die entsprechende Kante auch in dem durch den Baum T_2 definierten Graphen enthalten ist. Somit sind alle Kanten aus $H_2 \circ J$ in dem Graphen der durch T_2 definiert wird enthalten.

Der interessantere Teil besteht darin zu zeigen, dass jede Kanteneinfügeoperation aus T_2 keine Kante erzeugt, die nicht in $H_2 \circ J$ enthalten ist. Eine Kanteneinfügeoperation $u_s^{T_2}$ aus T_2 die nicht zu $T_{2,P}$ gehört erzeugt nur Kanten die auch in $H_2 \circ J$ enthalten sind, da der entsprechende Teilbaum $T_2(u_s^{T_2})$ entweder vollständig aus T' oder vollständig aus dem Baum T_1 kopiert wurde.

Wir nehmen nun an, dass die Kanteneinfügeoperation im Knoten $u_s^{T_2}$ auch zum Baum $T_{2,P}$ gehört. Es sei $\eta_{i,j}$ die Kanteneinfügeoperation des Knotens $u_s^{T_2}$. Falls $u_s^{T_2}$ nicht aus einem 4-Weg aus $T_{2,P}$, der aus mehr als einem Rahmen besteht, stammt dann ist $u_s^{T_2}$ auch in C enthalten. Dann gibt es auch eine entsprechende Kanteneinfügeoperation im Knoten $u_s^{T_1}$ bzw. $u_s^{T'}$ im Baum T_1 bzw. im Baum T' , die wie der Knoten $u_s^{T_2}$ in T_2 markiert ist. Diese gleiche Markierung garantiert uns, dass die Kanteneinfügeoperation $\eta_{i,j}$ eine neue Kante zwischen einem Knoten der mit i markiert ist und einem Knoten der mit j markiert ist erzeugt, wenn sie mindestens eine solche Kante in T_1 oder in T' erzeugt.

Falls $u_s^{T_2}$ aus einem 4-Weg p^{T_2} stammt, der auch in $T_{2,P}$ enthalten ist und aus mehr als einem Rahmen besteht, dann sei ohne Beschränkung der Allgemeinheit p^{T_2} aus T' kopiert, d.h. p^{T_2} ist ein innerer 4-Weg mit $\xi(q^{T_2}) = H_2$ für alle Rahmen q^{T_2} in p^{T_2} . Es sei u^C der entsprechende innere Brückenknoten für p^{T_2} und es sei u'^C der Sohn von u^C in C . Der Knoten u'^C kann ein Brückenknoten oder ein gewöhnlicher Knoten sein. Falls u'^C kein Brückenknoten ist dann sichert uns die gleiche Markierung von u'^{T_2} und $u'^{T'}$, dass $\eta_{i,j}$ eine neue Kante zwischen einem Knoten der mit i markiert ist und einem Knoten der mit j markiert ist, falls mindestens eine solche Kante in T' erzeugt wird.

Falls u'^C ein Brückenknoten ist, dann sei v^{T_2} der letzte Knoten des Weges in T_2 der in C durch den Knoten u'^C ersetzt wurde. Falls u'^C ein innerer (äußerer) Brückenknoten ist, dann sichert uns die gleiche Markierung der Knoten v^{T_2} und $v^{T'}$ (bzw. v^{T_1}), dass $\eta_{i,j}$ eine neue Kante zwischen einem Knoten der mit i markiert ist und einem Knoten der mit j markiert ist erzeugt, falls mindestens eine solche Kante in T' erzeugt wird. Somit ist jede Kante im Graphen der durch T_2 definiert wird auch in dem Graphen $H_2 \circ J$ enthalten und umgekehrt. \square

Aus Theorem 6.6.1 und der Tatsache, dass es nur endlich viele Verbindungstypen für feste natürliche Zahlen l und k gibt folgt, dass die Äquivalenzrelation $\sim_{\Pi_k, l}$ endlich viele Äquivalenzklassen hat, wobei Π_k die Grapheigenschaft „Cliquesweite höchstens k “ ist.

Korollar 6.6.2 *Für jedes feste k gibt es einen Linearzeitalgorithmus um Cliquesweite höchstens k auf Graphen mit beschränkter Baumweite zu entscheiden.*

Da die Cliquesweite eines Graphen mit Baumweite l höchstens $3 \cdot 2^{l-1}$ beträgt, siehe [CR01], gibt es einen Algorithmus der die Cliquesweite eines Graphen mit beschränkter Baumweite minimiert.

Korollar 6.6.3 *Es gibt einen Linearzeitalgorithmus der die Cliquesweite eines Graphen mit beschränkter Baumweite berechnet.*

Aus dem gezeigten Ergebnis kann jedoch nicht gefolgert werden, dass die erkennbare Grapheigenschaft „Cliquesweite höchstens k “ auch MSO_2 -definierbar ist. Die Äquivalenz zwischen erkennbaren Grapheigenschaften und MSO_2 -definierbaren Grapheigenschaften gilt nur für spezielle Graphklassen, wie zum Beispiel für Graphklassen mit beschränkter Baumweite [Lap98], jedoch nicht für alle Graphklassen.

Es ist bisher offen, ob man auch die NLC-Weite eines Graphen für Graphen mit beschränkter Baumweite effizient berechnen kann. Will man analog zu dem Beweis dieses Kapitels zeigen, dass die Grapheigenschaft Π'_k „NLC-Weite höchstens k “ die Menge aller k -markierten l -terminalen Graphen bezüglich der Äquivalenzrelation $\sim_{\Pi'_k, l}$ in endlich viele Äquivalenzklassen einteilt und einen NLC-Weite k -Ausdruck X anstelle eines Cliquesweite k -Ausdrucks verwendet, ist es nicht immer möglich den NLC-Weite k -Ausdrucksbaum zu X so zu modifizieren, dass Eigenschaft 6.5.2 erfüllt ist. Zum Beispiel kann man im NLC-Weite 1-Ausdruck

$$X = \bullet_{1,(p)} \times_{\{(1,1)\}} (\bullet_{1,(u)} \times_{\{(1,1)\}} \bullet_{1,(v)})$$

die beiden Knoten $u \in U_H$ und $v \in U_J$ nur mit Hilfe einer Hilfsmarkierung nacheinander an den Knoten $p \in U_P$ anfügen. Das hier gezeigte Ergebnis kann auch für die NLC-Weite gelten, der Beweis muss aber anders geführt werden.

Kapitel 7

Effiziente Algorithmen für NP-schwere Graphenprobleme

Im ersten Abschnitt dieses Kapitels geben wir einige Begriffe zur Laufzeit von Algorithmen an. Im zweiten Abschnitt definieren wir die Ersetzbarkeit von markierten Graphen und damit eine ganze Klasse von effizient lösbaren Problemen auf Graphen mit beschränkter Cliquesweite. Im dritten Abschnitt beschreiben wir ein Schema zur effizienten Lösung von Graphenproblemen auf rekursiv definierten Graphen. In den folgenden Abschnitten verwenden wir dieses Schema um effiziente Lösungen für zwölf NP-schwere Graphenprobleme auf Graphen mit beschränkter Cliquesweite anzugeben.

7.1 Algorithmen

Um ein Problem mit Hilfe eines Computers zu lösen benötigt man einen *Algorithmus*, d.h. ein allgemeines Berechnungsverfahren das zu einer Eingabe die Lösung des Problems bestimmt. Algorithmen werden in der Informatik bezüglich ihrer Laufzeit klassifiziert. Praktisch anwendbar sind die Algorithmen A zu denen es ein Polynom p gibt, so dass A für alle Eingaben der Größe n nach höchstens $p(n)$ Schritten das Ergebnis liefert. Solche Algorithmen arbeiten in *Polynomzeit*, bzw. heißen *polynomiell* oder *effizient*. Die Menge aller Probleme die in polynomieller Zeit gelöst werden können bezeichnen wir wie üblich mit P . Zum Beispiel das Problem eine gegebene Menge von Zahlen zu sortieren liegt in P , es gibt Sortieralgorithmen die n Zahlen in $O(n \cdot \log n)$ Schritten sortieren. Falls das Polynom den Grad eins hat, so heißt der Algorithmus auch *Linearzeit* Algorithmus. Eine weitere wichtige Komplexitätsklasse in der Informatik ist die Klasse NP , eine Abkürzung für nichtdeterministisch Polynomzeit. Ein nichtdeterministischer Algorithmus hat im Gegensatz zu einem deterministischen Algorithmus in einem gewissen Zustand die Möglichkeit in verschiedene Folgezustände überzugehen. Die Menge aller Probleme, die in polynomieller Zeit mit einem nichtdeterministischen Algorithmus gelöst werden können, bezeichnen wir wie üblich mit NP . Nichtdeterministische Algorithmen sind nur aus theoretischer Sicht interessant, da es keine Computer gibt, die diese berechnen. Es ist offensichtlich, dass $P \subseteq NP$ gilt, ob jedoch P eine echte Teilmenge von NP ist, ist bis heute eines der wichtigsten offenen Probleme der Informatik. Die schwierigsten Probleme der Klasse NP fasst man in der Klasse der *NP-vollständigen* Probleme $NP-C$ zusammen. Ein Problem Π ist NP -vollständig, wenn es in NP liegt und jedes andere Problem aus NP auf Π in polynomieller Zeit reduzierbar ist (die letzte Eigenschaft bezeichnet

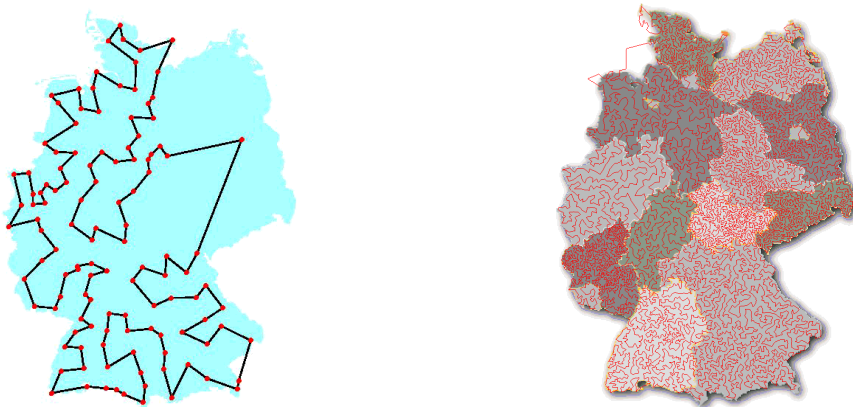


Tabelle 7.1: Groetschel fand 1977 die kürzeste Rundreise für 120 Städte im heutigen Westdeutschland (links), siehe [Gö77]. Applegate, Bixby, Chvátal und Cook fanden 2001 die kürzeste Rundreise für 15112 Städte in Deutschland (rechts), siehe [ABCC01].

man auch als *NP-schwer*). Ist ein Problem aus der Klasse NPC, so gibt es vermutlich keinen polynomiellen Algorithmus der das Problem auf einem wirklichen Rechner löst. Näheres zur NP-Vollständigkeit findet man zum Beispiel in dem Buch [Pap94].

Ein klassisches Graphenproblem ist das *Problem des Handlungsreisenden* (*Traveling Salesman Problem*, kurz *TSP* genannt). Ein Vertreter möchte zum Besuch seiner Kunden seine Rundreise möglichst effizient organisieren. Stellt man sich die zu besuchenden Städte als Knoten eines Graphen vor und nimmt als Länge der Kanten zwischen zwei Knoten die Verbindungsdauer um zwischen den zugehörigen Städten zu reisen, so gibt es mindestens eine Rundreise durch alle Städte, bei der keine Stadt mehrfach besucht wird und bei der die Gesamtdauer der Reise minimiert wird. Ein *Hamilton Kreis* in einem Graphen G ist ein Kreis in G , der alle Knoten von G genau einmal durchläuft. Damit kann das Traveling Salesman Problem folgendermaßen definiert werden. Gegeben ist ein vollständiger Graph $G = (V, E)$ wobei jeder Kante $e \in E$ positive Kosten $k(e)$ zugeordnet sind. Gesucht ist ein Hamilton Kreis C in G mit minimalen Kosten $\sum_{e \in E_C} k(e)$. Da G vollständig ist, gibt es auf jeden Fall einen Hamilton Kreis und man muss sich darauf konzentrieren einen Hamilton Kreis mit minimalen Kosten zu finden. Die Anzahl der möglichen Rundreisen bei n Städten beträgt $(n-1)! = (n-1) \cdot (n-2) \cdot (n-3) \cdot \dots \cdot 2 \cdot 1$. Die besten bekannten Algorithmen zur Lösung des TSP haben eine Laufzeit aus $O(2^n)$. Das Traveling Salesman Problem ist NP-vollständig. Im Jahre 1977 konnte man minimale Rundreisen für bis zu 120 Städte bestimmen, siehe Tabelle 7.1. Nimmt man an dass sich die Rechengeschwindigkeit von Computern seit 1977 alle 1,5 Jahre verdoppelt hat, so hat sich bis zum Jahre 2001 die Rechengeschwindigkeit 16 mal verdoppelt und man kann nun mit dem Algorithmus von 1977 Rundreisen über 136 Städte bestimmen. In Deutschland gibt es aber über 5000 Städte. Die bisher größte Instanz für das TSP wurde durch einen geschickten Algorithmenentwurf von Applegate, Bixby, Chvátal und Cook im Jahre 2001 gelöst und kann eine Rundreise über 15112 Städte finden, siehe [ABCC01]. In allgemeinen Graphen dagegen ist es bereits schwierig zu entscheiden, ob es überhaupt einen Hamilton Kreis gibt. Der Satz von Dirac liefert eine hinreichende Bedingung für die Existenz eines Hamilton Kreises. Besitzen in einem Graphen $G = (V, E)$ alle Knoten einen Grad größer gleich k , $k > 1$ und ist $|E| \leq 2k$, so besitzt G einen Hamilton Kreis. Richard Karp hat 1972

bewiesen, dass das Problem „Gegeben ein Graph G , enthält G einen Hamilton Kreis?“ NP-vollständig ist. Eine Variante des Hamilton Kreis Problems besteht darin, dass man auf die Forderung, dass die Rundreise im Startknoten endet verzichtet. In diesem Fall spricht man von einem *Hamilton Weg*. Das Problem in einem Graphen einen Hamilton Weg zu finden ist ebenfalls NP-vollständig. Aufgrund dieses Ergebnisses ist es sehr unwahrscheinlich, dass man einen Algorithmus findet, der in realistischer Zeit die Lösung der Probleme Hamilton Kreis und Hamilton Weg liefert. Um Lösungen für diese Probleme zu erhalten, sucht man nach effizienten Näherungslösungen oder man schränkt die Eingabe auf spezielle Graphklassen ein. Sowohl das Hamilton Kreis Problem als auch das Hamilton Weg Problem lassen sich auf Graphen mit beschränkter Cliquesweite effizient lösen, siehe Kapitel 7.11.

7.2 Ersetzbarkeit von k -markierten Graphen

Zur Beschreibung von effizienten Bottom-Up Algorithmen auf Graphklassen mit beschränkter Cliquesweite, definieren wir die *Ersetzbarkeit* von k -markierten Graphen. Der Verknüpfungsmechanismus ist hier die Vereinigungsoperation der NLC-Weite.

Definition 7.2.1 (Ersetzbarkeit von k -markierten Graphen) *Es sei \mathcal{G}_k die Menge aller k -markierter Graphen $G = (V, E, lab)$ mit $lab : V \rightarrow [k]$. Weiterhin sei $\Pi : \mathcal{G}_k \rightarrow \{0, 1\}$ eine Grapheigenschaft. Zwei k -markierte Graphen $G_1, G_2 \in \mathcal{G}_k$ heißen ersetzbar bezüglich Π , wir bezeichnen dies mit $G_1 \sim_{\Pi, k} G_2$, falls für alle k -markierten Graphen $H \in \mathcal{G}_k$ und alle Relationen $S \subseteq [k]^2$ gilt:*

$$\Pi(G_1 \times_S H) = \Pi(G_2 \times_S H)$$

Die Relation $\sim_{\Pi, k}$ ist eine Äquivalenzrelation. Wir betrachten nun die Grapheigenschaften Π , so dass die Relation $\sim_{\Pi, k}$ endliche viele Äquivalenzklassen hat näher.

Definition 7.2.2 (1-erkennbar [Wan94c]) *Eine Grapheigenschaft Π für die die Relation $\sim_{\Pi, k}$ für jede natürliche Zahl k einen endlichen Index hat, heißt 1-erkennbar.*

Alle 1-erkennbaren Grapheigenschaften Π können mit Bottom-Up Algorithmen in linearer Zeit für alle $G \in \mathcal{G}_k$ entschieden werden, wenn zusätzlich ein k -Ausdruck bzw. ein k -Ausdrucksbaum T für G gegeben ist. Der Linearzeitalgorithmus berechnet Bottom-Up die Äquivalenzklassen für jeden Graphen G' der durch einen vollständigen Teilbaum T' von T dargestellt wird. Die Äquivalenzklasse des Graphen G' , der durch den Teilbaum T' mit Wurzel u' definiert wird, kann in Zeit $O(1)$ aus den Äquivalenzklassen der beiden Graphen die durch die beiden Teilbäume $T' - \{u'\}$ definiert werden, bzw. im Falle eines Knotens u' mit einem Sohn aus der Äquivalenzklasse des Graphen der durch den Teilbaum $T' - \{u'\}$ definiert wird, bestimmt werden.

Mit Hilfe von logischen Formeln kann man Grapheigenschaften beschreiben die 1-erkennbar sind. Dazu betrachtet man die Probleme die MSO_1 -definierbar sind, d.h. die Probleme zu denen es eine Formel mit Quantifizierungen über Knoten und Knotenmengen gibt. Ein Graph $G = (V_G, E_G)$ heißt r -färbbar, falls es eine disjunkte Partition der Knotenmenge V in r unabhängige Mengen gibt. Die Grapheigenschaft $\Pi_r : \Pi_r(G) = G$ ist r -färbbar ist für jedes feste r wie folgt MSO_1 -definierbar. $\Pi_r(G) \Leftrightarrow \exists X_1, \dots, X_r \subseteq V_G (\text{Partition}(X_1, \dots, X_r) \wedge \text{unabhängig}(X_1) \wedge \dots \wedge \text{unabhängig}(X_r))$, wobei $\text{Partition}(X_1, \dots, X_r) \Leftrightarrow \forall u \in V_G (u \in X_1 \vee \dots \vee u \in X_r) \wedge \neg \exists u \in V_G (u \in X_1 \wedge u \in X_2 \vee \dots \vee u \in X_{r-1} \wedge u \in X_r)$ und

$unabhängig(X_i) \Leftrightarrow \forall u, v \in X_i : \neg\{u, v\} \in E_G$. Das Problem Π_r ist für jedes $r \geq 3$ NP-vollständig, siehe Problem [GT4] in [GJ79].

Theorem 7.2.3 (Courcelle, Makowsky, Rotics [CMR00, CMR01]) *Jede MSO_1 -definierbare Grapheigenschaft ist 1-erkennbar.*

Die Lösungsidee zu Theorem 7.2.3 beruht darauf, dass Teilgraphen, welche durch Teilausdrücke definiert sind, sich bezüglich der zu untersuchenden Eigenschaft in Äquivalenzklassen einteilen lassen. Für alle MSO_1 -definierbaren Eigenschaften ist die Anzahl der Äquivalenzklassen unabhängig von der Teilgraphgröße. Dadurch lassen sich mit Bottom-Up Analysemethoden mit Hilfe eines gegebenen Cliquesweite k -Ausdrucks oder eines NLC-Weite k -Ausdrucks alle MSO_1 -definierbaren Eigenschaften auf Graphen mit beschränkter Cliquesweite und auf Graphen mit beschränkter NLC-Weite in linearer Zeit entscheiden. Somit ist das MSO_1 -definierbare Problem, ob ein gegebener Graph l -färbbar ist, für jedes feste l , auf Graphklassen mit beschränkter Cliquesweite bzw. beschränkter NLC-Weite, bei gegebenem k -Ausdruck für den Eingabegraphen, in linearer Zeit entscheidbar.

Erweiterungen der MSO_1 Logik erlauben die Formulierung von Optimierungsproblemen. Beispielsweise das Finden der maximalen Clique (Problem [GT19] in [GJ79]) in einem Graphen ist in einer Erweiterung der MSO_1 Logik definierbar und auf Graphen mit beschränkter Cliquesweite bei gegebenem k -Ausdruck effizient lösbar. Die Aussage aus Theorem 7.2.3 ist nach einem Ergebnis in [CMR00] nicht auf alle MSO_2 -definierbaren Probleme erweiterbar.

Für distanzerhaltende Graphen kann nach einem Ergebnis von Golumbic und Rotics in [GR00] ein Cliquesweite 3-Ausdruck in linearer Zeit gefunden werden. Somit ist jedes MSO_1 -definierbare Problem auf distanzerhaltenden Graphen in Linearzeit lösbar. Diese Aussage ebenso für die Graphklassen P_4 -tidy Graphen, P_4 -sparse Graphen und $(q, q - 4)$ Graphen, siehe [CMR00]. Für allgemeine Graphklassen mit beschränkter Cliquesweite können MSO_1 -definierbare Probleme in Zeit $O(f(|V_G|))$ gelöst werden, wobei $f(|V_G|)$ die bisher unbekannte Zeit zum Finden des k -Ausdrucks ist.

7.3 Effiziente Algorithmen auf Graphen mit beschränkter Cliquesweite

Es gibt eine Vielzahl von Grapheigenschaften und Graphenproblemen, welche nicht MSO_1 -definierbar sind, aber dennoch auf Graphklassen mit beschränkter Cliquesweite effizient lösbar sind. Wir wollen hier anhand von zwölf Beispielen erläutern wie man schwere Graphenprobleme auf Graphklassen mit beschränkter Cliquesweite bzw. Graphklassen mit beschränkter NLC-Weite mit Hilfe des folgenden Schemas effizient lösen kann.

Theorem 7.3.1 (Wanke [Wan94c]) *Es sei Π ein Graphenproblem, das auf einem Graphen $val(X)$ gelöst werden soll, der durch einen gegebenen NLC-Weite k -Ausdruck X definiert ist. Falls es eine Abbildung F gibt, die jedem NLC-Weite k -Ausdruck X eine Struktur $F(X)$ zuordnet, so dass für alle NLC-Weite k -Ausdrücke X, Y*

1. die Größe von $F(X)$ polynomiell in der Größe von X beschränkt ist,
2. $F(\bullet_i)$, $i \in [k]$, in Zeit $O(1)$ berechenbar ist,
3. $F(\circ_R(X))$, $R : [k] \rightarrow [k]$, in polynomieller Zeit aus $F(X)$ berechenbar ist,

4. $F(X \times_S Y)$, $S \subseteq [k] \times [k]$, in polynomieller Zeit aus $F(X)$ und $F(Y)$ berechenbar ist und
5. $\Pi(\text{val}(X))$ in polynomieller Zeit aus $F(X)$ berechenbar ist,

dann ist für jeden NLC-Weite k -Ausdruck X die Lösung für Π auf $\text{val}(X)$ in polynomieller Zeit aus X berechenbar.

Die Eingabe für den Algorithmus besteht aus einem NLC-Weite k -Ausdruck bzw. aus einem NLC-Weite k -Ausdrucksbaum. Der Algorithmus berechnet Bottom-Up mittels dynamischer Programmierung für jeden Teilbaum mit Wurzel u die Äquivalenzklasse des zugehörigen Graphen aus den Äquivalenzklassen der Teilbäume der zwei Söhne von u , bzw. im Falle einer Ummarkierungsoperation aus der Äquivalenzklasse des Teilbaumes des Sohnes von u . Die Lösung des Problems bestimmt man mit Hilfe der Äquivalenzklasse der Wurzel des gegebenen NLC-Weite k -Ausdrucksbaumes bzw. mit Hilfe der Äquivalenzklasse der innersten Operation des gegebenen NLC-Weite k -Ausdrucks.

Die Schwierigkeit beim Lösen von Graphenproblemen nach diesem Schema besteht darin eine geeignete Datenstruktur F zu finden, so dass die Bedingungen aus Theorem 7.3.1 erfüllt sind. Mit Hilfe dieser Methode wurden in [Wan94c] das Hamilton Kreis Problem und das Simple Max Cut Problem auf Graphen mit beschränkter NLC-Weite gelöst. Beide Probleme sind NP-vollständig und nicht 1-erkennbar. Dieses Schema ist auf alle rekursiv definierten Graphenstrukturen anwendbar, also auch für Graphen mit beschränkter Cliquesweite, indem die Operationen \oplus , $\rho_{i \rightarrow j}$ und $\eta_{i,j}$ verwendet werden.

Theorem 7.3.2 *Es sei Π ein Graphenproblem, das auf einem Graphen $\text{val}(X)$ gelöst werden soll, der durch einen gegebenen Cliquesweite k -Ausdruck X definiert ist. Falls es eine Abbildung F gibt, die jedem Cliquesweite k -Ausdruck X eine Struktur $F(X)$ zuordnet, so dass für alle Cliquesweite k -Ausdrücke X, Y*

1. die Größe von $F(X)$ polynomiell in der Größe von X beschränkt ist,
2. $F(\bullet_i)$, $i \in [k]$, in Zeit $O(1)$ berechenbar ist,
3. $F(\rho_{i \rightarrow j}(X))$ und $F(\eta_{i,j}(X))$, $i, j \in [k], i \neq j$, in polynomieller Zeit aus $F(X)$ berechenbar sind,
4. $F(X \oplus Y)$ in polynomieller Zeit aus $F(X)$ und $F(Y)$ berechenbar ist und
5. $\Pi(\text{val}(X))$ in polynomieller Zeit aus $F(X)$ berechenbar ist,

dann ist für jeden Cliquesweite k -Ausdruck X die Lösung für Π auf $\text{val}(X)$ in polynomieller Zeit aus X berechenbar.

Kobler und Rotics haben in [KR01] mit einer Struktur F , die die Bedingungen von Theorem 7.3.2 erfüllt die Probleme Dominating Set und list- q -coloring mit Kosten in der Knoten- und Kantenversion und das Chromatic Number Problem auf Graphen mit beschränkter Cliquesweite gelöst. Todinca löst in [Tod03] nach diesem Schema das l -Coloring Problem auf Graphen mit beschränkter Cliquesweite.

Wir wollen das Schema aus Theorem 7.3.1 bzw. Theorem 7.3.2 verwenden, um die Probleme Partition in unabhängige Mengen oder Cliques, Partition in unabhängige Mengen oder Cliques mit beschränkter Größe, l -Coloring, Partition in vollständig bipartite Graphen,

Partition in Wälder, Partition in perfekte Matchings, Minimales maximales Matching, Hamilton Weg, Knotendisjunkte Wege, Knotengrad beschränkte Teilgraphen, Knotenpartition und Kantenpartition zu lösen. Diese Probleme sind alle NP-vollständig, siehe zum Beispiel [GJ79]. Weiterhin sind diese Probleme nicht MSO_1 -definierbar, so dass eine Lösung nicht aus dem Ergebnis von Courcelle, Makowsky und Rotics (Theorem 7.2.3) resultiert. Dies kann man häufig mit einer unendlichen Anzahl von Äquivalenzklassen bzgl. der Relation aus Definition 7.2.1 zeigen. Einige der hier gelösten Probleme sind nicht einmal MSO_2 -definierbar.

Bei unseren Lösungen sei immer vorausgesetzt, dass mit einem Graphen auch der zugehörige k -Ausdruck mit den Operationen der Cliquesweite oder den Operationen der NLC-Weite gegeben ist. Bisher sind noch keine effizienten Algorithmen zur Bestimmung eines k -Ausdrucks für $k \geq 4$ bzw. $k \geq 3$ bekannt. Da jeder Cliquesweite k -Ausdruck in einen äquivalenten NLC-Weite k -Ausdruck umgewandelt werden kann und jeder NLC-Weite k -Ausdruck in einen äquivalenten Cliquesweite $2k$ -Ausdruck umgewandelt werden kann, gebrauchen wir in den Lösungen unserer Probleme jeweils ohne Einschränkungen das Graphenmodell das die Datenstruktur F möglichst einfach macht. Wir speichern uns in der Struktur $F(X)$ Informationen über die Markierungen von Knotenmengen des durch einen k -Ausdruck gegebenen Eingabegraphen $\text{val}(X)$. Da wir voraussetzen, dass k fest ist, das heißt dass k nicht zu der Eingabe des Algorithmus gehört, haben unsere Algorithmen eine polynomielle Laufzeit, obwohl die Laufzeiten alle exponentiell in k sind.

Da der Komplementgraph eines Graphen mit Cliquesweite k (NLC-Weite k) Cliquesweite höchstens $2k$ (bzw. NLC-Weite k) hat und man einen Cliquesweite $2k$ -Ausdruck (bzw. NLC-Weite k -Ausdruck) für den Komplementgraphen effektiv angeben kann, ist mit einem Problem Π unmittelbar auch das Problem $\bar{\Pi} : \bar{\Pi}(G) := \Pi(\bar{G})$ auf dem Komplementgraphen lösbar.

Ebenso hat mit einem Graphen G auch der Potenzgraph G^d beschränkte NLC-Weite und man kann zu einem NLC-Weite k -Ausdruck für G einen NLC-Weite $k \cdot 2^{(d-1)k}$ -Ausdruck für G^d angeben. Damit kann man die in den folgenden Abschnitten gelösten Probleme auch auf dem Potenzgraphen des Eingabegraphen lösen.

Zur Lösung der meisten der nun folgenden Probleme verwenden wir *Multimengen*, d.h. Mengen die mehrere gleiche Elemente enthalten können. Für eine Multimenge \mathcal{M} mit den Elementen x_1, \dots, x_n schreiben wir

$$\mathcal{M} = \langle x_1, \dots, x_n \rangle.$$

Die Elemente von \mathcal{M} haben keine Ordnung. Die Häufigkeit eines Elementes x in einer Multimenge \mathcal{M} wird mit $\psi(\mathcal{M}, x)$ bezeichnet. Zwei Multimengen \mathcal{M}_1 und \mathcal{M}_2 heißen *gleich*, falls für jedes Element $x \in \mathcal{M}_1 \cup \mathcal{M}_2$ gilt $\psi(\mathcal{M}_1, x) = \psi(\mathcal{M}_2, x)$, sonst heißen sie *verschieden*. Die leere Multimenge bezeichnen wir mit $\langle \rangle$.

Weiterhin benötigen wir die folgenden Funktionen. Für zwei verschiedene natürliche Zahlen i, j sei $\rho_{i \rightarrow j} : [k] \rightarrow [k]$

$$\rho_{i \rightarrow j}(i') = \begin{cases} j & \text{falls } i = i' \\ i' & \text{falls } i \neq i' \end{cases} \quad \text{für alle } i' \in [k],$$

für $L \subseteq [k]$ sei

$$\rho_{i \rightarrow j}(L) = \{\rho_{i \rightarrow j}(i') \mid i' \in L\},$$

für $R : [k] \rightarrow [k]$ sei

$$R(L) = \{R(i') \mid i' \in L\}$$

und für $L \subseteq [k]^r$, $r > 1$, sei

$$\rho_{i \rightarrow j}(L) = \{(\rho_{i \rightarrow j}(i_1), \dots, \rho_{i \rightarrow j}(i_r)) \mid (i_1, \dots, i_r) \in L\}.$$

Für eine Knotenmenge $U \subseteq V$ in einem Graphen $G = (V, E, \text{lab})$ sei $\text{lab}(U) = \{\text{lab}(u) \mid u \in U\}$ und $\text{lab}^2(U)$ die Menge aller Markierungen i , so dass U mindestens zwei Knoten, die mit i markiert sind, enthält.

7.4 Partition in unabhängige Mengen oder Cliques

Das Problem die Knotenmenge eines Graphen G in disjunkte unabhängige Mengen einzuteilen wird auch als *Färbungsproblem* bezeichnet. Das Entscheidungsproblem „Gegeben ist ein Graph G , ist G r -färbbar?“ ist für jede feste natürliche Zahl $r \geq 3$ NP-vollständig, siehe Problem [GT4] in [GJ79] und MSO₁-definierbar und somit aufgrund Theorem 7.2.3 auf Graphen mit beschränkter Cliquesweite lösbar.

Wir betrachten hier die Minimierungsversion des Problems in dem r mit zur Eingabe gehört. Dieses Problem ist nicht MSO₁-definierbar und damit nicht unmittelbar durch Theorem 7.2.3 auf Graphen mit beschränkter Cliquesweite lösbar. Die minimale Anzahl der notwendigen unabhängigen Knotenmengen nennt man die *Färbungszahl* oder *Chromatic Number* $\chi(G)$ des Graphen G .

Problem 7.4.1 (Partition in unabhängige Mengen (in Cliques))

EINGABE: Ein Graph $G = (V, E)$ und eine natürliche Zahl $r \leq |V|$.

FRAGE: Gibt es eine Partition von V in r disjunkte Mengen V_1, \dots, V_r , so dass $V_1 \cup \dots \cup V_r = V$ und keine der Mengen V_t , $1 \leq t \leq r$, enthält zwei adjazente Knoten (jede der Mengen V_t , $1 \leq t \leq r$, induziert einen vollständigen Teilgraphen in G)?

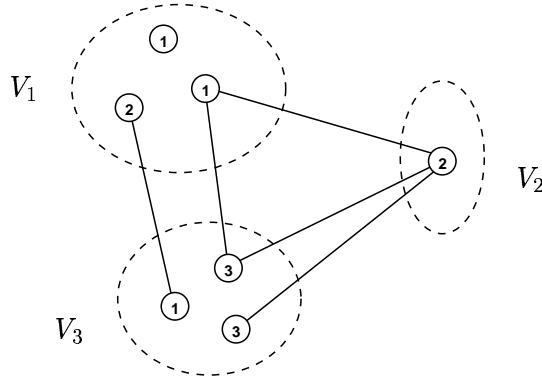
Es sei $G = (V, E, \text{lab}) = \text{val}(X)$ ein Graph der durch einen Cliquesweite k -Ausdruck X definiert ist. Für eine disjunkte Partition von V in r , $0 \leq r \leq |V|$, unabhängige Mengen V_1, \dots, V_r sei \mathcal{M} die Multimenge $\langle \text{lab}(V_1), \dots, \text{lab}(V_r) \rangle$. $F(X)$ sei die Menge aller paarweise verschiedenen Multimengen \mathcal{M} , für alle disjunkten Partitionen der Knotenmenge V in unabhängige Mengen.

Anschaulich kann man die Anzahl der möglichen verschiedenen Multimengen in $F(X)$ wie folgt darstellen. Jede Multimenge \mathcal{M} enthält Teilmengen von $[k]$, die in der ersten Zeile der folgenden Tabelle angegeben sind, jeweils in einer der darunterstehenden Anzahl.

$$\begin{array}{ccccccc}
 \{1\} & \dots & \{k\} & \{1, 2\} & \dots & \{k-1, k\} & \dots & \{1, \dots, k\} \\
 \hline
 0 & & 0 & 0 & & 0 & & 0 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 |V| & & |V| & |V| & & |V| & & |V| \\
 \hline
 (|V|+1) & \dots & (|V|+1) & (|V|+1) & \dots & (|V|+1) & \dots & (|V|+1) = (|V|+1)^{2^k-1} \\
 \underbrace{\hspace{10em}} & & & & & & & \\
 & & & & & & & 2^k-1
 \end{array}$$

Damit enthält $F(X)$ höchstens

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



$$F(X) = \{ \dots, \langle \{1, 2\}, \{2\}, \{1, 3\} \rangle, \dots \}$$

Abbildung 7.1: Die Knotenmenge des Graphen $G = \text{val}(X)$ ist in drei unabhängige Mengen V_1 , V_2 und V_3 aufgeteilt. $F(X)$ enthält die dazugehörige Multimenge $\langle \text{lab}(V_1), \text{lab}(V_2), \text{lab}(V_3) \rangle$.

paarweise verschiedene Multimengen und somit kann die Größe von $F(X)$ durch ein Polynom in der Größe von X beschränkt werden. Abbildung 7.1 zeigt eine Partition eines Graphen $G = \text{val}(X)$ in drei unabhängige Mengen und die dazugehörige Multimenge $\mathcal{M} \in F(X)$.

Die folgenden Überlegungen zeigen, dass $F(\bullet_i)$ in konstanter Zeit berechenbar ist, $F(X \oplus Y)$ in polynomieller Zeit aus $F(X)$ und $F(Y)$ berechenbar ist und $F(\eta_{i,j}(X))$ und $F(\rho_{i \rightarrow j}(X))$ in polynomieller Zeit aus $F(X)$ berechenbar sind.

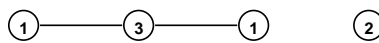
1. $F(\bullet_i) = \{\langle \{i\} \rangle\}$, $i \in [k]$
2. Ausgehend von der Menge $D = \{\langle \rangle\} \times F(X) \times F(Y)$ wird D um alle Tripel erweitert, die man aus einem Tripel $(\mathcal{M}, \mathcal{M}', \mathcal{M}'') \in D$ erhalten kann, indem man eine Menge L' aus \mathcal{M}' oder eine Menge L'' aus \mathcal{M}'' entfernt und die Menge L' bzw. L'' in \mathcal{M} einfügt oder indem man eine Menge L' aus \mathcal{M}' und eine Menge L'' aus \mathcal{M}'' entfernt und die Menge $L' \cup L''$ in \mathcal{M} einfügt.

D enthält höchstens $(|V| + 1)^{3(2^k - 1)}$ Tripel und ist somit in polynomieller Zeit berechenbar.

$$F(X \oplus Y) = \{\mathcal{M} \mid (\mathcal{M}, \langle \rangle, \langle \rangle) \in D\}$$

3. $F(\eta_{i,j}(X)) = \{\langle L_1, \dots, L_r \rangle \in F(X) \mid \{i, j\} \not\subseteq L_t \text{ für } t = 1, \dots, r\}$
4. $F(\rho_{i \rightarrow j}(X)) = \{\langle \rho_{i \rightarrow j}(L_1), \dots, \rho_{i \rightarrow j}(L_r) \rangle \mid \langle L_1, \dots, L_r \rangle \in F(X)\}$

Für den 3-markierten Graphen



ist $X = \eta_{1,3}((\bullet_1 \oplus \bullet_1) \oplus \bullet_3) \oplus \bullet_2$ ein Cliquesweite 3-Ausdruck und $F(X)$ kann mit Hilfe

der Mengen $F(Y)$, für Teilausdrücke Y von X , wie folgt berechnet werden.

$$\begin{aligned}
 F(\bullet_1) &= \{\{\{1\}\}\} \\
 F(\bullet_1 \oplus \bullet_1) &= \{\{\{1\}\}, \{\{1\}, \{1\}\}\} \\
 F(\bullet_3) &= \{\{\{3\}\}\} \\
 F((\bullet_1 \oplus \bullet_1) \oplus \bullet_3) &= \{\{\{1, 3\}\}, \{\{1, 3\}, \{1\}\}, \{\{1\}, \{3\}\}, \{\{1\}, \{1\}, \{3\}\}\} \\
 F(\eta_{1,3}((\bullet_1 \oplus \bullet_1) \oplus \bullet_3)) &= \{\{\{1\}, \{3\}\}, \{\{1\}, \{1\}, \{3\}\}\} \\
 F(\bullet_2) &= \{\{\{2\}\}\} \\
 F(X) &= \{\{\{1\}, \{1\}, \{2\}, \{3\}\}, \{\{1\}, \{2\}, \{3\}\}, \{\{1\}, \{1, 2\}, \{3\}\}, \\
 &\quad \{\{1\}, \{1\}, \{2, 3\}\}, \{\{1\}, \{2, 3\}\}, \{\{1, 2\}, \{3\}\}\}
 \end{aligned}$$

Es gibt eine Partition der Knotenmenge des Graphen $\text{val}(X)$ in r unabhängige Mengen, genau dann wenn es eine Multimenge $\mathcal{M} \in F(X)$ gibt die aus genau r Mengen besteht. Eine Partition von $G = (V, E, \text{lab})$ in Cliques entspricht einer Partition des Komplementgraphen $\overline{G} = (V, \overline{E}, \text{lab})$, $\overline{E} = \{\{u, v\} \mid u, v \in V, u \neq v, \{u, v\} \notin E\}$, in unabhängige Mengen. Ein Cliquesweite $2k$ -Ausdruck \overline{X} für $\text{val}(X)$ kann leicht in polynomieller Zeit aus X berechnet werden, siehe [CO00].

Damit ist das folgende Ergebnis gezeigt.

Theorem 7.4.2 *Zu einem gegebenen k -Ausdruck X , für eine feste natürliche Zahl k , sind die Probleme Partition in unabhängige Mengen und Partition in Cliques für $\text{val}(X)$ in Polynomzeit in der Größe von X lösbar.*

7.5 Partition in unabhängige Mengen oder Cliques mit beschränkter Größe

Wenn wir unabhängige Mengen bzw. Cliques mit beschränkter Größe suchen, so erhalten wir das folgende Problem.

Problem 7.5.1 (Partition in unabhängige Mengen (Cliques) beschränkter Größe)
EINGABE: Ein Graph $G = (V, E)$ und eine natürliche Zahl $r \leq |V|$.

FRAGE: Gibt es eine Partition von V in r disjunkte Mengen V_1, \dots, V_r der Größe s , so dass $V_1 \cup \dots \cup V_r = V$ und keine der Mengen V_t , $1 \leq t \leq r$, enthält zwei in G adjazente Knoten (jede der Mengen V_t , $1 \leq t \leq r$, induziert in G einen vollständigen Teilgraphen)?

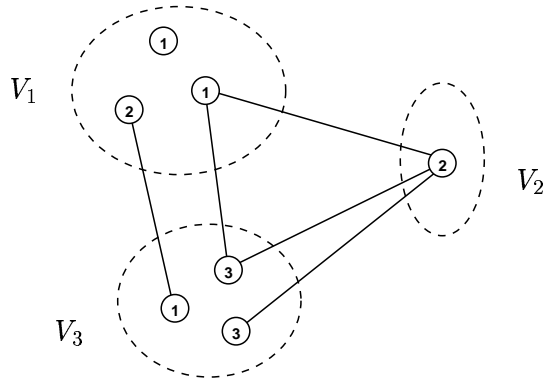
Es sei $G = (V, E, \text{lab}) = \text{val}(X)$ für einen Cliquesweite k -Ausdruck X . Weiterhin sei s eine natürliche Zahl. Zu einer disjunkten Partition von V in r , $0 \leq r \leq |V|$, unabhängige Mengen V_1, \dots, V_r der Größe höchstens s sei \mathcal{M} die Multimenge $\langle (\text{lab}(V_1), |V_1|), \dots, (\text{lab}(V_r), |V_r|) \rangle$. $F(X)$ sei die Menge aller paarweise verschiedenen Multimengen \mathcal{M} für alle möglichen Partitionen von V in disjunkte Mengen der Größe höchstens s .

Es gibt $(2^k - 1) \cdot s$ verschiedene Paare (L, x) , $L \subseteq [k]$, $1 \leq x \leq s$ und jedes Paar tritt höchstens $|V|$ mal in einer Multimenge auf. Damit enthält $F(X)$ höchstens

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

paarweise verschiedene Multimengen und die Größe von $F(X)$ ist polynomiell in der Größe von X beschränkt. Abbildung 7.2 zeigt eine Partition eines Graphen $\text{val}(X)$ in drei unabhängige Mengen der Größe höchstens drei und die zugehörige Multimenge $\mathcal{M} \in F(X)$.

$F(\bullet_i)$, $F(X \oplus Y)$, $F(\eta_{i,j}(X))$ und $F(\rho_{i \rightarrow j}(X))$ können wie folgt berechnet werden.



$$F(X) = \{ \dots, \langle (\{1, 2\}, 3), (\{2\}, 1), (\{1, 3\}, 3) \rangle, \dots \}$$

Abbildung 7.2: Die Knotenmenge des Graphen $G = \text{val}(X)$ ist in drei unabhängige Mengen V_1, V_2, V_3 der Größe höchstens drei eingeteilt. $F(X)$ enthält die dazugehörige Multimenge $\langle (\text{lab}(V_1), |V_1|), (\text{lab}(V_2), |V_2|), (\text{lab}(V_3), |V_3|) \rangle$.

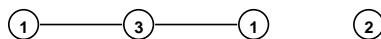
1. $F(\bullet_i) = \{ \langle (\{i\}, 1) \rangle \}, i \in [k]$
2. Ausgehend von der Menge $D = \{ \langle \rangle \} \times F(X) \times F(Y)$ wird D um alle Tripel erweitert, die man aus einem Tripel $(\mathcal{M}, \mathcal{M}', \mathcal{M}'') \in D$ erhalten kann, indem man ein Paar (L', x') aus \mathcal{M}' oder ein Paar (L'', x'') aus \mathcal{M}'' entfernt und das Paar (L', x') bzw. (L'', x'') in \mathcal{M} einfügt oder indem man ein Paar (L', x') aus \mathcal{M}' und ein Paar (L'', x'') aus \mathcal{M}'' entfernt und das Paar $(L' \cup L'', x' + x'')$ in \mathcal{M} einfügt, falls $x' + x'' \leq s$.

D enthält höchstens $(|V| + 1)^{3(2^k - 1) \cdot s}$ Tripel und ist somit in Polynomzeit berechenbar.

$$F(X \oplus Y) = \{ \mathcal{M} \mid (\mathcal{M}, \langle \rangle, \langle \rangle) \in D \}$$

3. $F(\eta_{i,j}(X)) = \{ \langle (L_1, x_1), \dots, (L_r, x_r) \rangle \in F(X) \mid \{i, j\} \not\subseteq L_t \text{ für } t = 1, \dots, r \}$
4. $F(\rho_{i \rightarrow j}(X)) = \{ \langle (\rho_{i \rightarrow j}(L_1), x_1), \dots, (\rho_{i \rightarrow j}(L_r), x_r) \rangle \mid \langle (L_1, x_1), \dots, (L_r, x_r) \rangle \in F(X) \}$

Für den 3-markierten Graphen



ist $X = \eta_{1,3}((\bullet_1 \oplus \bullet_1) \oplus \bullet_3) \oplus \bullet_2$ ein Cliquesweite 3-Ausdruck. $F(X)$ kann mit Hilfe der Mengen $F(Y)$, für Teilausdrücke Y von X , wie folgt berechnet werden.

$$\begin{aligned}
 F(\bullet_1) &= \{\langle\langle\{1\}, 1\rangle\rangle\} \\
 F(\bullet_1 \oplus \bullet_1) &= \{\langle\langle\{1\}, 2\rangle\rangle, \langle\langle\{1\}, 1\rangle, \langle\{1\}, 1\rangle\rangle\} \\
 F(\bullet_3) &= \{\langle\langle\{3\}, 1\rangle\rangle\} \\
 F((\bullet_1 \oplus \bullet_1) \oplus \bullet_3) &= \{\langle\langle\{1, 3\}, 3\rangle\rangle, \langle\langle\{1, 3\}, 2\rangle, \langle\{1\}, 1\rangle\rangle, \\
 &\quad \langle\langle\{1\}, 2\rangle, \langle\{3\}, 1\rangle\rangle, \langle\langle\{1\}, 1\rangle, \langle\{1\}, 1\rangle, \langle\{3\}, 1\rangle\rangle\} \\
 F(\eta_{1,3}((\bullet_1 \oplus \bullet_1) \oplus \bullet_3)) &= \{\langle\langle\{1\}, 2\rangle, \langle\{3\}, 1\rangle\rangle, \langle\langle\{1\}, 1\rangle, \langle\{1\}, 1\rangle, \langle\{3\}, 1\rangle\rangle\} \\
 F(\bullet_2) &= \{\langle\langle\{2\}, 1\rangle\rangle\} \\
 F(X) &= \{\langle\langle\{1\}, 1\rangle, \langle\{1\}, 1\rangle, \langle\{2\}, 1\rangle, \langle\{3\}, 1\rangle\rangle, \\
 &\quad \langle\langle\{1\}, 2\rangle, \langle\{2\}, 1\rangle, \langle\{3\}, 1\rangle\rangle, \langle\langle\{1\}, 1\rangle, \langle\{1, 2\}, 2\rangle, \langle\{3\}, 1\rangle\rangle, \\
 &\quad \langle\langle\{1\}, 1\rangle, \langle\{1\}, 1\rangle, \langle\{2, 3\}, 2\rangle\rangle, \langle\langle\{1\}, 2\rangle, \langle\{2, 3\}, 2\rangle\rangle, \\
 &\quad \langle\langle\{1, 2\}, 3\rangle, \langle\{3\}, 1\rangle\rangle\}
 \end{aligned}$$

Es gibt eine Partition von $\text{val}(X)$ in unabhängige Mengen der Größe s , genau dann wenn es eine Multimenge $\mathcal{M} \in F(X)$ gibt, so dass für alle $(L, x) \in \mathcal{M}$ gilt $x = s$. Mit der Kantenkomplementeigenschaft der Graphen mit beschränkter Cliquesweite gilt nun das folgende Theorem.

Theorem 7.5.2 *Zu einem gegebenen k -Ausdruck X , für eine feste natürliche Zahl k , können die Probleme Partition in unabhängige Mengen beschränkter Größe und Partition in Cliques beschränkter Größe für den Graphen $\text{val}(X)$ in polynomieller Zeit gelöst werden.*

Es sei Π_1 das Graphproblem, ob es eine Partition eines gegebenen Graphen in Cliques der Größe drei gibt. Π_1 wird auch *Partition in Triangles* (Problem [GT11] in [GJ79]) genannt. Die Relation $\sim_{\Pi_1, k}$ aus Definition 7.2.1 hat für $k = 1$ und zwei 1-markierte Graphen $K_{n,n}, \overline{K_m}$, $n, m \in \mathbb{N}$ wegen

$$\Pi_1(K_{n,n} \times_{\{(1,1)\}} \overline{K_m}) = \Pi_1(K_{n,n,m}) \Leftrightarrow n = m$$

unendlich viele Äquivalenzklassen und somit ist Π_1 nach Theorem 7.2.3 nicht MSO_1 -definierbar. Das Problem Π_1 ist MSO_2 -definierbar.

In Definition 6.2.2 wird eine zu Definition 7.2.1 analoge Äquivalenzrelation $\sim_{\Pi, l}$ für l -terminale Graphen definiert. Für jede MSO_2 -definierbare Eigenschaft Π hat $\sim_{\Pi, l}$ für jedes $l \geq 0$ endlich viele Äquivalenzklassen. Es sei Π_2 das Graphproblem, ob es eine Partition eines gegebenen Graphen in unabhängige Mengen der Größe drei gibt. Die Relation $\sim_{\Pi_2, l}$ aus Definition 6.2.2 hat für $l = 0$ und zwei 0-terminale Graphen $K_n \cup K_n$ und K_m , $n, m \in \mathbb{N}$ wegen

$$\Pi_2((K_n \cup K_n) \circ K_m) = \Pi_2(K_n \cup K_n \cup K_m) \Leftrightarrow n = m$$

unendlich viele Äquivalenzklassen und somit ist Π_2 nicht erkennbar. Damit ist Π_2 nicht MSO_2 -definierbar und effizient auf Graphen mit beschränkter Cliquesweite lösbar.

7.6 l -Coloring

Problem 7.6.1 (l -Coloring)

EINGABE: Ein Graph $G = (V, E)$ und eine natürliche Zahl $r \leq |V|$.

FRAGE: Gibt es eine Färbung $f : V \rightarrow W$, $W \subseteq \mathbb{N}$, $|W| = r$, von V , so dass $f(u) \neq f(v)$, für jedes Knotenpaar u, v mit $\text{dist}(u, v) \leq l$ gilt?

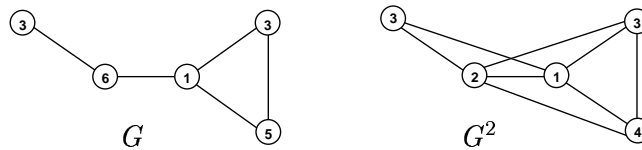


Abbildung 7.3: Ein Graph G mit $\chi_2(G) = 4$ mit den vier Knotenmarkierungen 1,3,5 und 6 und der quadrierte Graph G^2 mit $\chi(G^2) = 4$

Die minimale Anzahl von natürlichen Zahlen (Farben) zur Färbung der Knoten eines Graphen G mit einem l -Coloring bezeichnen wir mit $\chi_l(G)$. Ein Graph G hat ein l -Coloring mit r Farben, genau dann wenn der Potenzgraph G^l r -färbbar ist, es gilt also

$$\chi_l(G) = \chi(G^l).$$

Damit kann das l -Coloring Problem für einen Graphen G durch das Problem Partition in unabhängige Mengen (siehe Kapitel 7.4) auf G^l gelöst werden. Einen k -Ausdruck X' für G^l kann man aus einem Ausdruck X für G berechnen, siehe Kapitel 5.4.

In Abbildung 7.3 ist ein Graph G angegeben in dessen Knoten ein 2-Coloring mit vier verschiedene Zahlen eingezeichnet ist. Der Graph G^2 hat die Färbungszahl 4.

Theorem 7.6.2 *Zu einem gegebenen k -Ausdruck X , für eine feste natürliche Zahl k , kann das l -Coloring Problem für $\text{val}(X)$ in Polynomzeit gelöst werden.*

Todincă löst das l -Coloring Problem in [Tod03] direkt auf Graphen mit beschränkter Cliquesweite, indem er eine Struktur $F(X)$ angibt, die die Bedingungen aus Theorem 7.3.2 erfüllt. Todincă's Lösung hat den Vorteil, dass sie eine bessere Laufzeit hat.

7.7 Partition in vollständig bipartite Graphen

Problem 7.7.1 (Partition in vollständig bipartite Graphen)

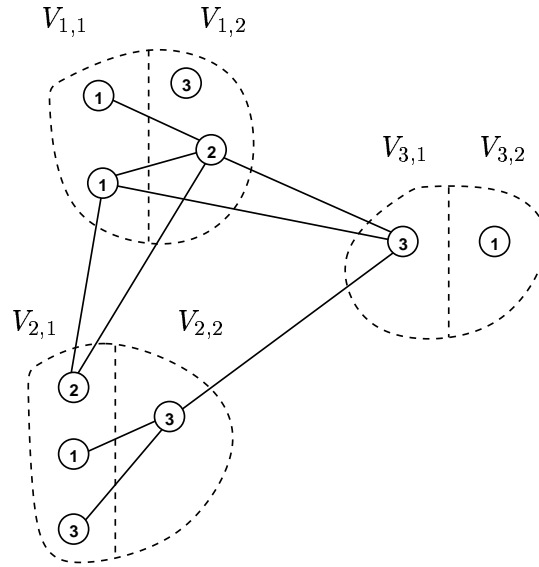
EINGABE: Ein Graph $G = (V, E)$ und eine natürliche Zahl $r \leq |V|$.

FRAGE: Gibt es eine Partition von V in r disjunkte Mengen V_1, \dots, V_r , so dass $V_1 \cup \dots \cup V_r = V$ und jede Menge V_t , $1 \leq t \leq r$, einen vollständig bipartiten Teilgraphen in G induziert?

Es sei $G = (V, E, \text{lab}) = \text{val}(X)$ für einen Cliquesweite k -Ausdruck X . $V_{1,1}, V_{1,2}, \dots, V_{r,1}, V_{r,2}$ sei eine disjunkte Partition der Knotenmenge V in unabhängige Mengen, so dass die Teilgraphen G_t , $t = 1, \dots, r$, die durch die Knoten $V_{t,1} \cup V_{t,2}$ in G induziert werden, bipartite (nicht unbedingt vollständig bipartite) Graphen sind. \mathcal{M} sei die Multimenge die für jedes Paar $(V_{t,1}, V_{t,2})$ ein Tripel $(L_1, L_2, L_{\overline{E}})$ enthält, das wie folgt definiert ist. $L_1 = \text{lab}(V_{t,1})$, $L_2 = \text{lab}(V_{t,2})$ und $L_{\overline{E}}$ ist die Menge aller Paare von Knotenmarkierungen (i, j) , so dass es zwei nicht adjazente Knoten $u \in V_{t,1}$, $v \in V_{t,2}$ mit $\text{lab}(u) = i$, $\text{lab}(v) = j$ gibt. $F(X)$ sei die Menge aller paarweise verschiedenen Multimengen \mathcal{M} für alle möglichen so definierten disjunkten Partitionen der Knotenmenge V in bipartite Graphen. Abbildung 7.4 zeigt eine Partition eines Graphen $\text{val}(X)$ in drei bipartite Graphen und die dazugehörige Multimenge $\mathcal{M} \in F(X)$.

$F(X)$ enthält höchstens

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



$$F(X) = \left\{ \dots \left\langle \begin{pmatrix} \{1\}, & \{2, 3\}, & \{(1, 3)\} \\ \{1, 2, 3\}, & \{3\}, & \{(2, 3)\} \\ \{3\}, & \{1\}, & \{(3, 1)\} \end{pmatrix} \right\rangle \dots \right\}$$

Abbildung 7.4: Der Graph $G = \text{val}(X)$ ist in drei bipartite Graphen eingeteilt. $F(X)$ enthält die dazugehörige Multimenge.

paarweise verschiedene Multimengen, die jeweils r , $0 \leq r \leq |V|$, Tripel $(L_1, L_2, L_{\overline{E}})$ enthalten. L_1 und L_2 sind Teilmengen von $[k]$ und $L_{\overline{E}}$ ist eine Teilmenge von $[k] \times [k]$.

$F(\bullet_i)$, $F(X \oplus Y)$, $F(\eta_{i,j}(X))$ und $F(\rho_{i \rightarrow j}(X))$ können wie folgt berechnet werden.

1. $F(\bullet_i) = \{ \langle (\{i\}, \emptyset, \emptyset), (\emptyset, \{i\}, \emptyset) \rangle \}$, $i \in [k]$
2. Ausgehend von der Menge $D = \{ \langle \rangle \} \times F(X) \times F(Y)$ wird D um alle Tripel erweitert, die man aus einem Tripel $(\mathcal{M}, \mathcal{M}', \mathcal{M}'') \in D$ erhalten kann, indem man ein Tripel $(L'_1, L'_2, L'_{\overline{E}})$ aus \mathcal{M}' oder ein Tripel $(L''_1, L''_2, L''_{\overline{E}})$ aus \mathcal{M}'' entfernt und dieses Tripel in \mathcal{M} eingefügt oder indem man beide Tripel entfernt und das Tripel

$$(L'_1 \cup L''_1, L'_2 \cup L''_2, L'_{\overline{E}} \cup L''_{\overline{E}} \cup L'_1 \times L''_2 \cup L''_1 \times L'_2)$$

in \mathcal{M} einfügt.

$$F(X \oplus Y) = \{ \mathcal{M} \mid (\mathcal{M}, \langle \rangle, \langle \rangle) \in D \}$$

3. Wir nennen eine Multimenge $\mathcal{M} \in F(X)$ *zulässig* für die Operation $\eta_{i,j}$, falls für jedes Tripel $(L_1, L_2, L_{\overline{E}}) \in \mathcal{M}$: $\{i, j\} \not\subseteq L_1$ und $\{i, j\} \not\subseteq L_2$ gilt.

Falls \mathcal{M} für die Operation $\eta_{i,j}$ zulässig ist, dann sei $\eta_{i,j}(\mathcal{M})$ die Multimenge die man aus \mathcal{M} erhält, indem man für alle Tripel $(L_1, L_2, L_{\overline{E}}) \in \mathcal{M}$ die Paare (i, j) und (j, i) aus allen Mengen $L_{\overline{E}}$ entfernt.

$$F(\eta_{i,j}(X)) = \{ \eta_{i,j}(\mathcal{M}) \mid \mathcal{M} \in F(X), \mathcal{M} \text{ ist zulässig für } \eta_{i,j} \}$$

4. Für eine Ummarkierungsoperation $\rho_{i \rightarrow j}$ und eine Multimenge $\mathcal{M} \in F(X)$, sei $\rho_{i \rightarrow j}(\mathcal{M})$ die Multimenge die man aus \mathcal{M} erhält, indem man jedes Tripel $(L_1, L_2, L_{\overline{E}}) \in \mathcal{M}$ durch $(\rho_{i \rightarrow j}(L_1), \rho_{i \rightarrow j}(L_2), \rho_{i \rightarrow j}(L_{\overline{E}}))$ ersetzt.

$$F(\rho_{i \rightarrow j}(X)) = \{\rho_{i \rightarrow j}(\mathcal{M}) \mid \mathcal{M} \in F(X)\}.$$

Es gibt eine Partition von $\text{val}(X)$ in r vollständig bipartite Graphen, genau dann wenn es eine Multimenge $\mathcal{M} \in F(X)$ gibt, die aus r Tripeln (L_1, L_2, \emptyset) besteht, in denen die Mengen L_1 und L_2 nicht leer sind.

Theorem 7.7.2 *Zu einem k -Ausdruck X , für eine feste natürliche Zahl k , kann das Problem Partition in vollständige bipartite Graphen für $\text{val}(X)$ in Polynomzeit gelöst werden.*

7.8 Partition in Wälder

Problem 7.8.1 (Partition in Wälder)

EINGABE: Ein Graph $G = (V, E)$ und eine natürliche Zahl $r \leq |V|$.

FRAGE: Gibt es eine Partition von V in r disjunkte Mengen V_1, \dots, V_r , so dass $V_1 \cup \dots \cup V_r = V$ und jede der Mengen V_t , $1 \leq t \leq r$, in G einen Wald induziert?

Es sei $G = (V, E, \text{lab}) = \text{val}(X)$ für einen Cliquesweite k -Ausdruck X . V_1, \dots, V_r sei eine diskunkte Partition der Knotenmenge V , so dass die Teilgraphen G_t , $t = 1, \dots, r$, die in G durch die Knoten aus V_t induziert werden, Wälder sind. Für eine solche Partition von V sei \mathcal{M} die Multimenge die für jede Knotenmenge V_t , $1 \leq t \leq r$, ein Tripel $(L_{\geq 1}, L_{\geq 2}, L_T)$ enthält. $L_{\geq 1} = \text{lab}(V_t)$, $L_{\geq 2} = \text{lab}^2(V_t)$ und L_T ist die Menge aller Tripel $(K_{\geq 1}, K_{\geq 2}, K_{\overline{E}})$, so dass G_t einen Baum $T = (V_T, E_T, \text{lab}_T)$ enthält. $K_{\geq 1} = \text{lab}(V_T)$, $K_{\geq 2} = \text{lab}^2(V_T)$ und $K_{\overline{E}}$ ist die Menge aller zweielementigen Markierungsmengen $\{i, j\}$, so dass V_T zwei Knoten enthält die nicht adjazent sind und mit i bzw. j markiert sind. $F(X)$ sei die Menge aller so definierten paarweise verschiedener Multimengen \mathcal{M} für alle möglichen Partitionen von V in Wälder. Abbildung 7.5 zeigt eine Partition eines Graphen $\text{val}(X)$ in drei Wälder und die zugehörige Multimenge $\mathcal{M} \in F(X)$.

Die Größe von $F(X)$ ist polynomiell in der Größe von X beschränkt, da $F(X)$ höchstens

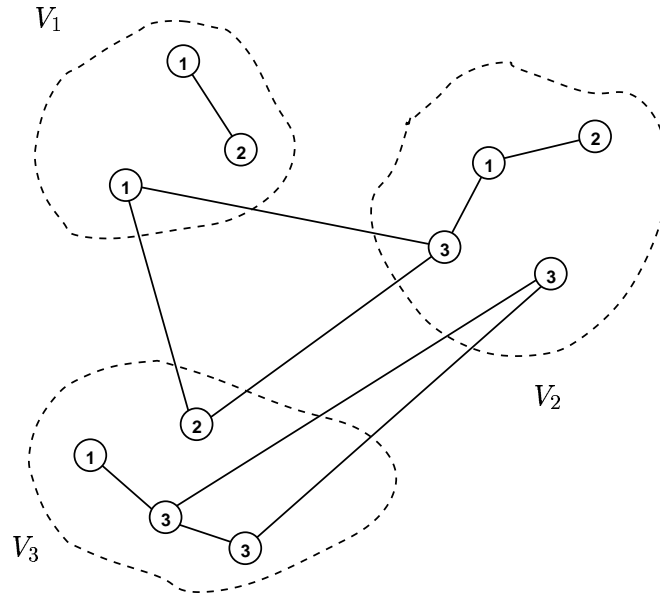
$$(|V| + 1)^{(2^k - 1) \cdot 2^k \cdot 2^{2^k} \cdot 2^{2^k} \cdot 2^{k^2}}$$

paarweise verschiedene Multimengen enthält, jede mit r , $0 \leq r \leq |V|$, Tripeln $(L_{\geq 1}, L_{\geq 2}, L_T)$. $L_{\geq 1}$ ist eine nicht leere Teilmenge von $[k]$, $L_{\geq 2}$ ist eine Teilmenge von $[k]$ und L_T ist eine Menge von Tripeln $(K_{\geq 1}, K_{\geq 2}, K_{\overline{E}})$. $K_{\geq 1}$ ist eine nicht leere Teilmenge von $[k]$, $K_{\geq 2}$ ist eine Teilmenge von $[k]$ und $K_{\overline{E}}$ ist eine Teilmenge von $\{\{i, j\} \mid i, j \in [k], i \neq j\}$.

$F(\bullet_i)$, $F(X \oplus Y)$, $F(\eta_{i,j}(X))$ und $F(\rho_{i \rightarrow j}(X))$ können wie folgt bestimmt werden.

1. $F(\bullet_i) = \{\langle \{\{i\}, \emptyset, \{\{\{i\}, \emptyset, \emptyset\}\} \rangle\}$, $i \in [k]$
2. Initial sei $D = \{\langle \rangle\} \times F(X) \times F(Y)$. D wird um alle Tripel erweitert, die man aus einem Tripel $(\mathcal{M}, \mathcal{M}', \mathcal{M}'') \in D$ erhalten kann, indem man ein Tripel $(L'_{\geq 1}, L'_{\geq 2}, L'_T)$ aus \mathcal{M}' oder ein Tripel $(L''_{\geq 1}, L''_{\geq 2}, L''_T)$ aus \mathcal{M}'' entfernt und dieses Tripel in \mathcal{M} einfügt oder indem man beide Tripel entfernt und das Tripel

$$(L'_{\geq 1} \cup L''_{\geq 1}, L'_{\geq 2} \cup L''_{\geq 2} \cup (L'_{\geq 1} \cap L''_{\geq 1}), L'_T \cup L''_T)$$



$$F(X) = \left\{ \begin{array}{l} \dots \\ \langle (\{1, 2\}, \{1\}, \{(\{1\}, \emptyset, \emptyset), (\{1, 2\}, \emptyset, \emptyset) \}), \\ (\{1, 2, 3\}, \{3\}, \{(\{1, 2, 3\}, \emptyset, \{2, 3\}), (\{3\}, \emptyset, \emptyset) \}), \\ (\{1, 2, 3\}, \{3\}, \{(\{2\}, \emptyset, \emptyset), (\{1, 3\}, \{3\}, \{1, 3\}) \}) \\ \dots \end{array} \right\}$$

Abbildung 7.5: Der Graph $G = \text{val}(X)$ ist in drei Wälder eingeteilt, die durch die drei Mengen V_1, V_2 und V_3 induziert werden. $F(X)$ enthält die zugehörige Multimenge.

in \mathcal{M} einfügt.

$$F(X \oplus Y) = \{ \mathcal{M} \mid (\mathcal{M}, \langle \rangle, \langle \rangle) \in D \}$$

- Wir nennen eine Multimenge $\mathcal{M} \in F(X)$ zulässig für die Operation $\eta_{i,j}$, falls für jedes Tripel $(L_{\geq 1}, L_{\geq 2}, L_T) \in \mathcal{M}$: $\{i, j\} \not\subseteq L_{\geq 2}$ gilt und für jedes Tripel $(K_{\geq 1}, K_{\geq 2}, K_{\overline{E}}) \in L_T$: $\{i, j\} \not\subseteq K_{\overline{E}}$ gilt und falls $i \in L_{\geq 1} - K_{\geq 2}$ dann ist $j \notin K_{\geq 2}$ und falls $j \in L_{\geq 1} - K_{\geq 2}$ dann ist $i \notin K_{\geq 2}$.

Falls \mathcal{M} für die Operation $\eta_{i,j}$ zulässig ist, dann sei $\eta_{i,j}(\mathcal{M})$ die Multimenge, die aus \mathcal{M} durch die folgende Modifikation der Tripel $(L_{\geq 1}, L_{\geq 2}, L_T) \in \mathcal{M}$ erhalten werden kann. Es sei $i \in L_{\geq 1} - L_{\geq 2}$, falls $j \in L_{\geq 1} - L_{\geq 2}$, dann vertausche i und j . Weiterhin seien

$$(K_{1,\geq 1}, K_{1,\geq 2}, K_{1,\overline{E}}), \dots, (K_{s,\geq 1}, K_{s,\geq 2}, K_{s,\overline{E}}) \in L_T$$

alle Tripel, so dass $i \in K_{1,\geq 1}$ und $j \in K_{t,\geq 1}$ für $t = 2, \dots, s$. Diese s Tripel werden aus L_T entfernt und das wie folgt definierte Tripel $(K'_{\geq 1}, K'_{\geq 2}, K'_{\overline{E}})$ wird in L_T eingefügt.

$$\begin{aligned} K'_{\geq 1} &= \bigcup_{1 \leq t \leq s} K_{t,\geq 1}, \\ K'_{\geq 2} &= \bigcup_{1 \leq t \leq s} K_{t,\geq 2} \cup \bigcup_{1 \leq t_1 < t_2 \leq s} K_{t_1,\geq 1} \cap K_{t_2,\geq 1} \text{ und} \\ K'_{\overline{E}} &= \bigcup_{1 \leq t \leq s} K_{t,\overline{E}} \cup \{ \{i', j'\} \mid i' \in K_{t_1,\geq 1}, j' \in K_{t_2,\geq 1}, 1 \leq t_1 < t_2 \leq s \} - \{i, j\} \end{aligned}$$

$$F(\eta_{i,j}(X)) = \{\eta_{i,j}(\mathcal{M}) \mid \mathcal{M} \in F(X), \mathcal{M} \text{ ist zulässig für } \eta_{i,j}\}$$

4. Für eine Ummarkierungsoperation $\rho_{i \rightarrow j}$ sei $\rho_{i \rightarrow j}(\mathcal{M})$ die Multimenge, die man aus \mathcal{M} erhält, indem man die Tripel $(L_{\geq 1}, L_{\geq 2}, L_T) \in \mathcal{M}$ wie folgt modifiziert. Falls beide Markierungen i, j in $L_{\geq 1}$ oder in $K_{\geq 1}$ eines Tripels $(K_{\geq 1}, K_{\geq 2}, K_{\overline{E}}) \in L_T$ liegen, dann wird j in $L_{\geq 2}$ bzw. in $K_{\geq 2}$ eingefügt.

Dann wird in jeder Menge aus jedem Tripel $(L_{\geq 1}, L_{\geq 2}, L_T)$ jedes i durch j ersetzt.

$$F(\rho_{i \rightarrow j}(X)) = \{\rho_{i \rightarrow j}(\mathcal{M}) \mid \mathcal{M} \in F(X)\}$$

Es gibt eine Partition von $\text{val}(X)$ in r Wälder, genau dann wenn es eine Multimenge $\mathcal{M} \in F(X)$ gibt die aus genau r Tripeln besteht.

Theorem 7.8.2 *Zu einem gegebenen k -Ausdruck X , für eine feste natürliche Zahl k , kann das Problem Partition in Wälder für $\text{val}(X)$ in Polynomzeit gelöst werden.*

7.9 Partition in perfekte Matchings

Eine Teilmenge $M \subseteq E$ der Kantenmenge eines Graphen $G = (V, E)$ heißt *Matching*, falls keine zwei Kanten in M einen Knoten gemeinsam haben. Ein Matching M in G heißt *perfekt*, falls es zu jedem Knoten $v \in V$ eine Kante $e \in M$ gibt mit $v \in e$, d.h. in dem durch M in G induzierten Teilgraph hat jeder Knoten den Grad eins.

Problem 7.9.1 (Partition in perfekte Matchings)

Eingabe: *Ein Graph $G = (V, E)$ und eine natürliche Zahl $r \leq |V|$.*

Frage: *Gibt es eine Partition von V in r disjunkte Mengen V_1, \dots, V_r , so dass $V_1 \cup \dots \cup V_r = V$ und jede Menge V_t , $1 \leq t \leq r$, ein perfektes Matching in G induziert?*

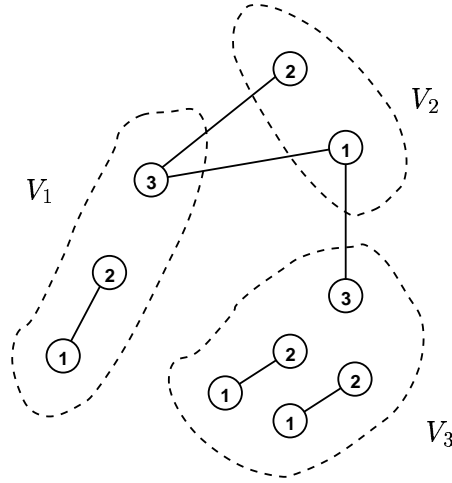
Es sei $G = (V, E, \text{lab}) = \text{val}(X)$ für einen Cliquesweite k -Ausdruck X . Eine Knotenmenge $V' \subseteq V$ heißt *kantenunabhängig* falls der Teilgraph G' von G der durch die Knoten in V' induziert wird keine zwei adjazenten Kanten enthält. V_1, \dots, V_r sei eine disjunkte Partition der Knotenmenge V in r kantenunabhängige Mengen. G_t bezeichne den durch V_t in G induzierten Teilgraphen. Für eine solche Partition von V sei \mathcal{M} die Multimenge die für jede Knotenmenge V_t , $1 \leq t \leq r$, ein 4-Tupel $(L_{\geq 1}, L_{\geq 2}, L_E, L_U)$ enthält das wie folgt definiert ist. $L_{\geq 1} = \text{lab}(V_t)$, $L_{\geq 2} = \text{lab}^2(V_t)$, L_E ist die Menge aller Markierungsmengen $\{i, j\}$, so dass G_t zwei adjazente Knoten enthält die mit i bzw. j markiert sind und L_U ist die Menge aller Knotenmarkierungen i , so dass G_t einen Knoten enthält, der mit i markiert ist der zu keiner Kante aus G_t inzident ist. $F(X)$ sei die Menge aller so definierten paarweise verschiedenen Multimengen \mathcal{M} für alle disjunkten Partitionen der Knotenmenge V in kantenunabhängige Mengen. Abbildung 7.6 zeigt eine Partition eines Graphen $\text{val}(X)$ in drei kantenunabhängige Mengen und die zugehörige Multimenge $\mathcal{M} \in F(X)$.

$F(X)$ enthält höchstens

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

paarweise verschiedene Multimengen, jede enthält r , $0 \leq r \leq |V|$, 4-Tupel $(L_{\geq 1}, L_{\geq 2}, L_E, L_U)$. $L_{\geq 1}$ ist eine nicht leere Teilmenge von $[k]$, $L_{\geq 2}$ und L_U sind Teilmengen von $[k]$ und L_E ist eine Teilmenge von $\{\{i, j\} \mid i, j \in [k], i \neq j\}$.

$F(\bullet_i)$, $F(X \oplus Y)$, $F(\eta_{i,j}(X))$ und $F(\rho_{i \rightarrow j}(X))$ können wie folgt berechnet werden.



$$F(X) = \left\{ \dots \left(\begin{array}{cccc} \{1, 2, 3\}, & \emptyset, & \{\{1, 2\}\}, & \{3\} \\ \{1, 2\}, & \emptyset, & \emptyset, & \{1, 2\} \\ \{1, 2, 3\}, & \{1, 2\}, & \{\{1, 2\}\}, & \{3\} \end{array} \right) \dots \right\}$$

Abbildung 7.6: Eine Aufteilung der Knotenmenge des Graphen $G = \text{val}(X)$ in drei Mengen V_1, V_2 und V_3 , die jeweils ein Matching (jedoch kein perfektes Matching) in G induzieren. $F(X)$ enthält die zugehörige Multimenge.

1. $F(\bullet_i) = \{(\{i\}, \emptyset, \emptyset, \{i\})\}$, $i \in [k]$
2. Ausgehend von $D = \{(\emptyset)\} \times F(X) \times F(Y)$ wird D um alle Tripel erweitert, die man aus einem Tripel $(\mathcal{M}, \mathcal{M}', \mathcal{M}'') \in D$ erhalten kann, indem man ein 4-Tupel $(L'_{\geq 1}, L'_{\geq 2}, L'_E, L'_U)$ aus \mathcal{M}' oder ein 4-Tupel $(L''_{\geq 1}, L''_{\geq 2}, L''_E, L''_U)$ aus \mathcal{M}'' entfernt und dieses 4-Tupel in \mathcal{M} einfügt oder indem man beide 4-Tupel entfernt und das 4-Tupel

$$(L'_{\geq 1} \cup L''_{\geq 1}, L'_{\geq 2} \cup L''_{\geq 2} \cup (L'_{\geq 1} \cap L''_{\geq 1}), L'_E \cup L''_E, L'_U \cup L''_U)$$

in \mathcal{M} einfügt.

$$F(X \oplus Y) = \{\mathcal{M} \mid (\mathcal{M}, \langle \rangle, \langle \rangle) \in D\}$$

3. Die Multimenge $\mathcal{M} \in F(X)$ heißt *zulässig* für die Operation $\eta_{i,j}$, falls für jedes 4-Tupel $(L_{\geq 1}, L_{\geq 2}, L_E, L_U) \in \mathcal{M}$ folgende Eigenschaft erfüllt ist. Falls $\{i, j\} \subseteq L_{\geq 1}$, dann ist $\{i, j\} \subseteq L_{\geq 1} - L_{\geq 2}$ und entweder gilt $\{i, j\} \in L_E$ oder $i, j \in L_U$.

Falls \mathcal{M} für die Operation $\eta_{i,j}$ zulässig ist, dann sei $\eta_{i,j}(\mathcal{M})$ die Multimenge die man aus \mathcal{M} erhält indem man die 4-Tupel $(L_{\geq 1}, L_{\geq 2}, L_E, L_U) \in \mathcal{M}$ wie folgt modifiziert. Falls $i, j \in L_{\geq 1} - L_{\geq 2}$ und $i, j \in L_U$, dann entferne i, j aus L_U und füge $\{i, j\}$ in L_E ein.

$$F(\eta_{i,j}(X)) = \{\eta_{i,j}(\mathcal{M}) \mid \mathcal{M} \in F(X), \mathcal{M} \text{ ist zulässig für } \eta_{i,j}\}$$

4. Für eine Ummarkierungsoperation $\rho_{i \rightarrow j}$ sei $\rho_{i \rightarrow j}(\mathcal{M})$ die Multimenge die man aus \mathcal{M} erhält, indem man die 4-Tupel $(L_{\geq 1}, L_{\geq 2}, L_E, L_U) \in \mathcal{M}$ folgendermaßen modifiziert. Falls $i, j \in L_{\geq 1}$ dann wird j in $L_{\geq 2}$ eingefügt. Dann wird in allen Mengen $L_{\geq 1}, L_{\geq 2}, L_U$ und L_E jede Markierung i durch die Markierung j ersetzt.

$$F(\rho_{i \rightarrow j}(X)) = \{\rho_{i \rightarrow j}(\mathcal{M}) \mid \mathcal{M} \in F(X)\}$$

Es gibt eine Partition von $\text{val}(X)$ in r perfekte Matchings, genau dann wenn es eine Multimenge $\mathcal{M} \in F(X)$ gibt, die aus r 4-Tupeln $(L_{\geq 1}, L_{\geq 2}, L_E, L_U)$ besteht in denen die Menge L_U leer ist.

Theorem 7.9.2 *Zu einem gegebenen k -Ausdruck X , für eine feste natürliche Zahl k , kann das Problem Partition in perfekte Matchings für $\text{val}(X)$ in Polynomzeit gelöst werden.*

7.10 Minimales maximales Matching

Ein Matching $E' \subseteq E$ in einem Graphen $G = (V, E)$ heißt *maximal*, falls jede Kante aus $E - E'$ einen gemeinsamen Endpunkt mit einer Kante aus E' hat.

Problem 7.10.1 (Minimales maximales Matching)

EINGABE: Ein Graph $G = (V, E)$ und eine natürliche Zahl $r \leq |E|$.

FRAGE: Gibt es eine Teilmenge $E' \subseteq E$ mit $|E'| \leq r$, so dass E' ein maximales Matching ist?

Es sei $G = (V, E, \text{lab}) = \text{val}(X)$ für einen Cliquesweite k -Ausdruck X . Für eine Teilmenge der Kanten $E' \subseteq E$ und eine disjunkte Partition von V in drei Mengen V_1, V_2, V_3 , so dass V_1 eine unabhängige Menge ist und E' ein perfektes Matching für die Knoten aus V_2 ist, sei $(\text{lab}(V_1), |E'|, \langle \text{lab}(u_1), \dots, \text{lab}(u_s) \rangle)$ ein Tripel mit $V_3 = \{u_1, \dots, u_s\}$. $F(X)$ sei die Menge aller dieser Tripel für alle diese Kantenteilmengen und Partitionen der Knotenmenge. $F(X)$ enthält dann höchstens

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

Tripel der Größe höchstens $k + 1 + |V|$. Abbildung 7.7 zeigt ein maximales Matching in einem Graphen.

Die Mengen $F(\bullet_i)$, $F(X \oplus Y)$, $F(\eta_{i,j}(X))$ und $F(\rho_{i \rightarrow j}(X))$ können wie folgt in polynomieller Zeit berechnet werden.

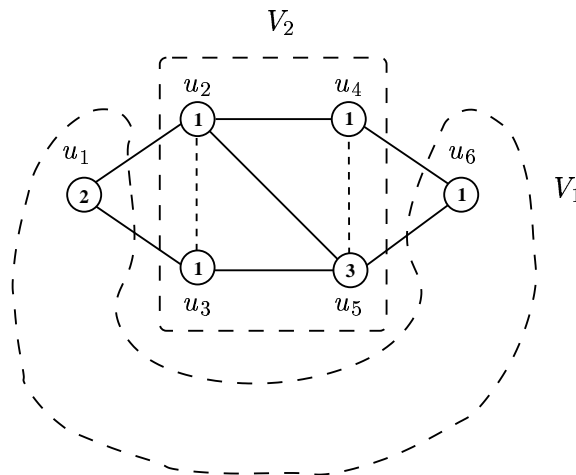
1. $F(\bullet_i) = \{(\emptyset, 0, \langle i \rangle), (\{i\}, 0, \langle \rangle)\}$, $i \in [k]$
2. $F(X \oplus Y) = \{(L \cup R, t_1 + t_2, \mathcal{M} \cup \mathcal{N}) \mid (L, t_1, \mathcal{M}) \in F(X), (R, t_2, \mathcal{N}) \in F(Y)\}$
3. $F(\eta_{i,j}(X))$ kann wie folgt aus $F(X)$ bestimmt werden. Ausgehend von $D = F(X)$ werden alle Tripel (L, k, \mathcal{M}) aus D gelöscht für die $\{i, j\} \subseteq L$ gilt. Dann wird D um alle Tripel $(L, t + 1, \mathcal{M} - \{\{i\}, \{j\}\})$ erweitert, die man aus einem Tripel (L, t, \mathcal{M}) in D mit $\{\{i\}, \{j\}\} \subseteq \mathcal{M}$ erhalten kann.

Die so erhaltene Menge D ist $F(X)$.

4. $F(\rho_{i \rightarrow j}(X)) = \{(\rho_{i \rightarrow j}(L), t, \rho_{i \rightarrow j}(\mathcal{M})) \mid (L, t, \mathcal{M}) \in F(X)\}$

Es gibt eine Teilmenge $E' \subseteq E$ mit $|E'| = r$, so dass $|E'|$ ein maximales Matching in G ist, genau dann wenn es ein Tripel $(L, r, \langle \rangle) \in F(X)$ gibt.

Theorem 7.10.2 *Zu einem gegebenen k -Ausdruck X , für eine feste natürliche Zahl k , kann das Minimale Maximale Matching Problem für $\text{val}(X)$ in Polynomzeit gelöst werden.*



$$V_1 = \{u_1, u_6\}, F(X) = \{ \dots (\{1, 2\}, 2, \langle \rangle) \dots \}$$

Abbildung 7.7: Die Knotenmenge des Graphen $G = \text{val}(X)$ ist in drei Mengen $V_1 = \{u_1, u_6\}$, $V_2 = \{u_2, u_3, u_4, u_5\}$, $V_3 = \emptyset$ eingeteilt. Die gestrichelten Kanten $E' = \{\{u_2, u_3\}, \{u_4, u_5\}\}$ bilden ein maximales Matching der Größe zwei im Graphen der durch die Knoten in V_2 induziert wird. $F(X)$ enthält das Tripel $(\text{lab}(V_1), |E'|, \langle \rangle)$.

7.11 Hamilton Weg

Problem 7.11.1 (Hamilton Weg)

EINGABE: Ein Graph $G = (V, E)$.

FRAGE: Gibt es einen einfachen Weg der Länge $|V| - 1$ in G , d.h. einen Weg der Länge $|V| - 1$ in G der jeden Knoten in G genau einmal besucht?

Es sei $G = (V, E, \text{lab}) = \text{val}(X)$ ein Graph für einen Cliquesweite k -Ausdruck X . Für eine Teilmenge $E' \subseteq E$ der Kantenmenge, so dass der Teilgraph $G' = (V, E', \text{lab})$ keine Kreise enthält und jeder Knoten in G' Grad höchstens 2 hat, sei \mathcal{M} die folgende Multimenge. Für jeden Weg $p = (v_1, \dots, v_r)$, $r \geq 1$, in G' , so dass die Knoten v_1 und v_r Grad höchstens 1 in G' haben, enthält \mathcal{M} eine Multimenge $\langle \text{lab}(v_1), \text{lab}(v_r) \rangle$. $F(X)$ sei die Menge aller Multimengen \mathcal{M} für alle diese Teilmengen $E' \subseteq E$. Es gibt $k(k+1)/2$ verschiedene Multimengen $\langle i, j \rangle$, $i, j \in [k]$. (Da $\langle i, j \rangle = \langle j, i \rangle$ genügt es sich auf Multimengen $\langle i, j \rangle$ mit $i \leq j$ zu beschränken.) Jede Multimenge \mathcal{M} enthält höchstens $|V|$ Multimengen der Größe zwei, da diese Wegen in G entsprechen. Damit enthält $F(X)$ höchstens

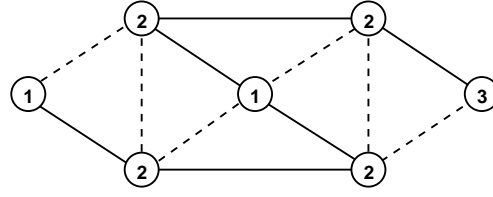
$$(|V| + 1)^{k(k+1)/2}$$

verschiedene Multimengen.

Abbildung 7.8 zeigt einen Teilgraph (Weg) in einem Graphen $\text{val}(X)$ in dem der Anfangs- und Endknoten den Knotengrad 1 haben und die dazugehörige Multimenge $\mathcal{M} \in F(X)$.

$F(\bullet_i)$, $F(X \oplus Y)$, $F(\eta_{i,j}(X))$ und $F(\rho_{i \rightarrow j}(X))$ können wie folgt berechnet werden.

1. $F(\bullet_i) = \{\langle i, i \rangle\}$, $i \in [k]$
2. $F(X \oplus Y) = \{\mathcal{M}' \cup \mathcal{M}'' \mid \mathcal{M}' \in F(X), \mathcal{M}'' \in F(Y)\}$



$$F(X) = \{ \dots, \langle \langle 1, 3 \rangle \rangle, \dots \}$$

Abbildung 7.8: Ein Hamilton Weg in einem Graphen $G = \text{val}(X)$. $F(X)$ enthält die zugehörige Multimenge.

3. $F(\eta_{i,j}(X))$ kann wie folgt aus $F(X)$ bestimmt werden. Ausgehend von $D = F(X)$ wird D um alle Multimengen erweitert, die aus einer Multimenge \mathcal{M} aus D erhalten werden können indem zwei Multimengen $\langle i_1, j_1 \rangle, \langle i_2, j_2 \rangle$, mit $i = i_1$ und $j = j_2$ entfernt werden und die neue Multimenge $\langle i_2, j_1 \rangle$ eingefügt wird. Das so erhaltene D ist $F(\eta_{i,j}(X))$.
4. $F(\rho_{i \rightarrow j}(X)) = \{ \langle \rho_{i \rightarrow j}(i), \rho_{i \rightarrow j}(j) \rangle \mid \langle i, j \rangle \in \mathcal{M} \mid \mathcal{M} \in F(X) \}$

Es gibt einen Hamilton Weg in $\text{val}(X)$ genau dann wenn es eine Multimenge $\mathcal{M} \in F(X)$ gibt, die aus genau einer Multimenge besteht.

Theorem 7.11.2 *Zu einem gegebenen k -Ausdruck X , für eine feste natürliche Zahl k , kann das Hamilton Weg Problem für $\text{val}(X)$ in Polynomzeit gelöst werden.*

Diese Lösung kann man leicht zu einer Lösung für das Hamilton Kreis Problem modifizieren, für Graphen mit beschränkter NLC-Weite ist dieses Problem in [Wan94c] gelöst.

Es sei Π_1 das Problem ob ein gegebener Graph einen Hamilton Kreis enthält. Die Relation $\sim_{\Pi_1, k}$ aus Definition 7.2.1 besitzt für $k = 1$ und zwei 1-markierte Graphen $\overline{K}_n, \overline{K}_m, n, m \in \mathbb{N}$ wegen

$$\Pi_1(\overline{K}_n \times_{\{(1,1)\}} \overline{K}_m) = \Pi_1(K_{n,m}) \Leftrightarrow n = m$$

unendlich viele Äquivalenzklassen und somit ist Π_1 nach Theorem 7.2.3 nicht MSO_1 -definierbar. Das Problem Π_1 ist MSO_2 -definierbar, siehe Kapitel 6.2.

Damit kann man unmittelbar auch das Problem Π_2 , ob der Komplementgraph eines Graphen einen Hamilton Kreis hat, effizient auf Graphen mit beschränkter Cliquesweite lösen. Die Relation aus $\sim_{\Pi_2, l}$ aus Definition 6.2.2 hat für $l = 0$ und zwei 0-terminale Graphen K_n und $K_m, n, m \in \mathbb{N}$ wegen

$$\Pi_2(K_n \circ K_m) = \Pi_2(K_n \cup K_m) = \Pi_1(\overline{K_n \cup K_m}) = \Pi_1(\overline{K_n} \times \overline{K_m}) = \Pi_1(K_{n,m}) \Leftrightarrow n = m$$

unendlich viele Äquivalenzklassen und somit ist Π_2 nach Theorem 6.2.4 nicht einmal MSO_2 -definierbar und effizient auf Graphen mit beschränkter Cliquesweite lösbar.

7.12 Knotendisjunkte Wege

Wir betrachten nun das Problem zwischen gegebenen Knotenpaaren knotendisjunkte Wege zu finden. Zwei Wege $p = (u_1, \dots, u_r)$ und $q = (v_1, \dots, v_{r'})$ heißen *knotendisjunkt*, falls $\{u_1, \dots, u_r\} \cap \{v_2, \dots, v_{r'-1}\} = \emptyset$ und $\{v_1, \dots, v_{r'}\} \cap \{u_2, \dots, u_{r-1}\} = \emptyset$.

Problem 7.12.1 (Knotendisjunkte Wege)

EINGABE: Ein Graph $G = (V, E)$, $2 \cdot l$ paarweise verschiedene Knoten $u_1, \dots, u_{2 \cdot l}$ und l natürliche Zahlen r_1, \dots, r_l .

FRAGE: Gibt es r_t knotendisjunkte Wege der Länge größer gleich 2 in G zwischen u_t und u_{t+l} für $t = 1, \dots, l$, so dass alle $\sum_{t=1}^l r_t$ Wege paarweise knotendisjunkt sind?

Ohne Einschränkung der Allgemeinheit betrachten wir hier nur Wege die mindestens 3 Knoten enthalten. Alle anderen Wege können durch Testen von Adjazenzen der Knotenpaare u_t, u_{t+l} , $1 \leq t \leq l$, gefunden werden.

Für festes l und $r_t = 1$, $1 \leq t \leq l$, kann das Knotendisjunkte Wege Problem für Graphen mit n Knoten in Zeit $O(n^3)$ gelöst werden, siehe [RS95]. Das Knotendisjunkte Wege Problem ist NP-vollständig für $l = 2$, $r_1 = 1$ und unbeschränktes r_2 , siehe [EIS76]. Das Problem ist ebenfalls NP-vollständig wenn die Anzahl l der Knotenpaare unbeschränkt ist und $r_t = 1$, $1 \leq t \leq l$, siehe [MP95].

Wir geben eine Lösung für das Knotendisjunkte Wege Problem für eine feste Anzahl l von Knotenpaaren auf Graphen mit beschränkter Cliquesweite an. Es sei $G = (V, E, \text{lab}) = \text{val}(X)$ ein Graph der durch einen Cliquesweite k -Ausdruck definiert ist. Wir nennen die Knoten $u_1, \dots, u_{2 \cdot l}$ terminale Knoten und alle anderen Knoten nichtterminale Knoten. Wir gehen davon aus, dass wir wissen welcher Teilausdruck \bullet_i im Ausdruck X zu welchem terminalen Knoten u_t gehört.

Wir betrachten jede Teilmenge $E' \subseteq E$, so dass der Teilgraph $G' = (V, E', \text{lab})$ von G die folgenden Bedingungen erfüllt.

1. Jeder nichtterminale Knoten in G' hat den Grad höchstens 2.
2. Jeder Kreis in G' enthält zwei terminale Knoten u_t, u_{t+l} , für ein t mit $1 \leq t \leq l$.
3. Terminale Knoten sind in G' nicht adjazent.
4. Die terminalen Knoten u_t, u_{t+l} , $1 \leq t \leq l$, in G' haben den Grad höchstens r_t .
5. Falls es einen Weg in G' zwischen zwei unterschiedlichen terminalen Knoten gibt, dann ist einer der Knoten u_t und der andere u_{t+l} , für ein t mit $1 \leq t \leq l$.

Für jede solche Teilmenge E' definieren wir wie folgt drei Multimengen \mathcal{M} , \mathcal{N}_2 , \mathcal{N}_3 und eine Menge \mathcal{N}_1 .

1. Definition der Multimenge \mathcal{M} :

\mathcal{M} enthält eine zweielementige Multimenge $\langle i, j \rangle$, $i, j \in [k]$, für jeden Weg p in G' , so dass alle Knoten in p nichtterminale Knoten sind und die beiden (nicht unbedingt verschiedene) Endknoten des Weges mit i bzw. mit j markiert sind.

2. Definition der Menge \mathcal{N}_1 :

\mathcal{N}_1 enthält ein Paar (t, i) , $1 \leq t \leq 2 \cdot l$, $i \in [k]$, für jeden terminalen Knoten u_t der mit i markiert ist.

3. Definition der Multimenge \mathcal{N}_2 :

\mathcal{N}_2 enthält ein Paar (t, i) , $1 \leq t \leq 2 \cdot l$, $i \in [k]$, für jeden Weg von dem terminalen Knoten u_t über nichtterminale Knoten zu einem nichtterminalen Knoten vom Grad 1, der mit i markiert ist.

4. Definition der Multimenge \mathcal{N}_3 :

\mathcal{N}_3 enthält eine natürliche Zahl t , $1 \leq t \leq l$, für jeden Weg von dem terminalen Knoten u_t über nichtterminale Knoten zu dem terminalen Knoten u_{t+l} .

Die Multimenge \mathcal{M} enthält höchstens $|V|$ Multimengen der Größe zwei, da die Multimengen in \mathcal{M} knotendisjunkten Wegen entsprechen. Es gibt $k(k+1)/2$ Multimengen $\langle i, j \rangle$, $i, j \in [k]$. Da jede Multimenge $\langle i, j \rangle$ in \mathcal{M} höchstens $|V|$ mal auftreten kann, gibt es höchstens $(|V|+1)^{k(k+1)/2}$ verschiedene Multimengen \mathcal{M} .

Die Menge N_1 enthält höchstens $2 \cdot l$ Paare, eines für jeden terminalen Knoten. Da jeder in G vorkommende terminale Knoten eine der k möglichen Markierungen besitzt, gibt es höchstens $(k+1)^{2 \cdot l}$ verschiedene Mengen N_1 .

Die Multimenge \mathcal{N}_2 enthält höchstens $|V|$ Paare. Es gibt höchstens $2 \cdot l \cdot k$ Paare (t, i) , $1 \leq t \leq l$, $i \in [k]$. Da jedes Paar (t, i) in \mathcal{N}_2 nur r_t mal auftreten kann und $0 \leq r_t \leq |V|$, gibt es höchstens $(|V|+1)^{2 \cdot l \cdot k}$ paarweise verschiedene Multimengen \mathcal{N}_2 .

Die Multimenge \mathcal{N}_3 enthält höchstens $\sum_{t=1}^l r_t \leq |V|$ natürliche Zahlen. Da jede natürliche Zahl t in \mathcal{N}_3 höchstens r_t mal auftreten kann und $0 \leq r_t \leq |V|$, gibt es höchstens $(|V|+1)^l$ paarweise verschiedene Multimengen \mathcal{N}_3 .

Diese Abschätzungen sind alle nicht scharf.

Es sei X ein Cliquesweite k -Ausdruck für den Graphen G . $F(X)$ sei die Menge aller 4-Tupel

$$(\mathcal{M}, N_1, \mathcal{N}_2, \mathcal{N}_3)$$

die durch alle Teilmengen E' von E definiert werden. Die obigen Abschätzungen zeigen, dass es höchstens

$$(|V|+1)^{k(k+1)/2} \cdot (k+1)^{2 \cdot l} \cdot (|V|+1)^{2 \cdot l \cdot k} \cdot (|V|+1)^l$$

verschiedene 4-Tupel $(\mathcal{M}, N_1, \mathcal{N}_2, \mathcal{N}_3)$ gibt. Damit ist die Größe von $F(X)$ für feste natürliche Zahlen l und k polynomiell in der in der Größe von X beschränkt.

Abbildung 7.9 zeigt ein Beispiel für einen Teilgraphen G' eines Graphen G und das zugehörige 4-Tupel $(\mathcal{M}, N_1, \mathcal{N}_2, \mathcal{N}_3)$.

$F(\bullet_i)$, $F(X \oplus Y)$, $F(\eta_{i,j}(X))$ und $F(\rho_{i \rightarrow j}(X))$ können wie folgt berechnet werden.

1. Falls \bullet_i einen terminalen Knoten u_t definiert, dann ist

$$F(\bullet_i) = \{(\langle \rangle, \{(t, i)\}, \langle \rangle, \langle \rangle)\}$$

sonst ist

$$F(\bullet_i) = \{(\langle i, i \rangle, \emptyset, \langle \rangle, \langle \rangle)\}.$$

2. $F(X \oplus Y) = \{(\mathcal{M}' \cup \mathcal{M}'', N_1' \cup N_1'', \mathcal{N}_2' \cup \mathcal{N}_2'', \mathcal{N}_3' \cup \mathcal{N}_3'' \mid (\mathcal{M}', N_1', \mathcal{N}_2', \mathcal{N}_3') \in F(X), (\mathcal{M}'', N_1'', \mathcal{N}_2'', \mathcal{N}_3'') \in F(Y)\}$
3. $F(\eta_{i,j}(X))$ kann wie folgt aus $F(X)$ berechnet werden. Ausgehend von $D = F(X)$ wird D um alle 4-Tupel erweitert, die aus einem 4-Tupel $(\mathcal{M}, N_1, \mathcal{N}_2, \mathcal{N}_3) \in D$ wie folgt erhalten werden können.
 - (a) Ersetze zwei Multimengen $\langle i, i' \rangle$ und $\langle j, j' \rangle$ für $i', j' \in [k]$ aus \mathcal{M} durch eine Multimenge $\langle i', j' \rangle$,

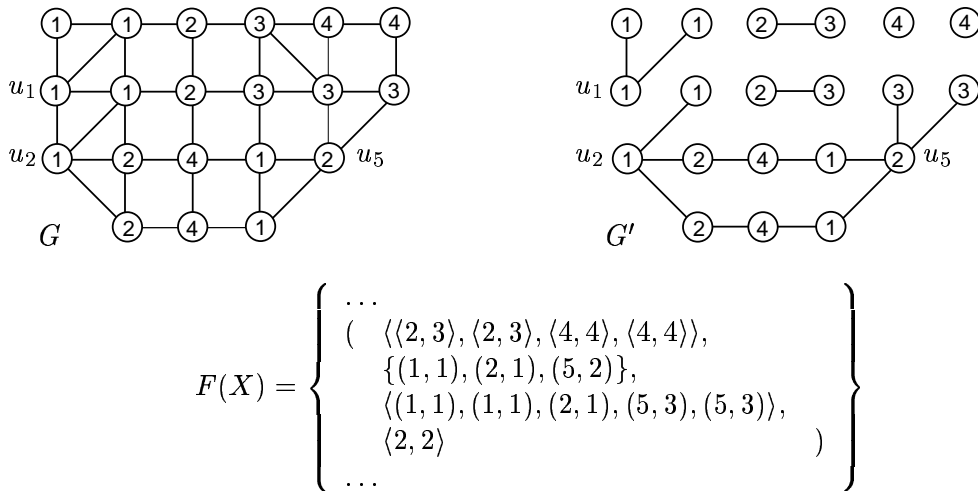


Abbildung 7.9: Der Teilgraph G' von $G = \text{val}(X)$ definiert für $l = 3$, $r_1 \geq 2$ und $r_2 \geq 4$ das in $F(X)$ angegebene 4-Tupel $(\mathcal{M}, N_1, \mathcal{N}_2, \mathcal{N}_3)$.

- (b) entferne eine Multimenge $\langle j, i' \rangle$ aus \mathcal{M} für ein $i' \in [k]$ und füge ein Paar (t, i') in \mathcal{N}_2 für ein t , $1 \leq t \leq 2 \cdot l$, ein falls N_1 ein Paar (t, i) enthält, so dass \mathcal{N}_2 und \mathcal{N}_3 zusammen weniger als r_t Paare (t, j') , $j' \in [k]$, und natürliche Zahlen t enthalten,
- (c) entferne eine Multimenge $\langle i, i' \rangle$ aus \mathcal{M} für ein $i' \in [k]$ und füge ein Paar (t, i') in \mathcal{N}_2 für ein t , $1 \leq t \leq 2 \cdot l$, ein falls N_1 ein Paar (t, j) enthält, so dass \mathcal{N}_2 und \mathcal{N}_3 zusammen weniger als r_t Paare (t, j') , $j' \in [k]$, und natürliche Zahlen t enthalten,
- (d) entferne ein Paar (t, i) aus \mathcal{N}_2 für ein $1 \leq t \leq 2 \cdot l$ und eine Multimenge $\langle j, i' \rangle$ aus \mathcal{M} für ein $i' \in [k]$ und füge ein Paar (t, i') in \mathcal{N}_2 ein,
- (e) entferne ein Paar (t, j) aus \mathcal{N}_2 für ein $1 \leq t \leq 2 \cdot l$ und eine Multimenge $\langle i, i' \rangle$ aus \mathcal{M} für ein $i' \in [k]$ und füge ein Paar (t, i') in \mathcal{N}_2 ein,
- (f) entferne zwei Paare (t, i) , $(t+l, j)$ aus \mathcal{N}_2 für ein $1 \leq t \leq l$ und füge die natürliche Zahl t in \mathcal{N}_3 ein oder
- (g) entferne zwei Paare (t, j) , $(t+l, i)$ aus \mathcal{N}_2 für ein $1 \leq t \leq l$ und füge die natürliche Zahl t in \mathcal{N}_3 ein.

Wenn D nicht durch weitere 4-Tupel verändert werden kann, so ist das so erhaltene D die Menge $F(\eta_{i,j}(X))$.

$$4. F(\rho_{i \rightarrow j}(X)) = \{(\rho_{i \rightarrow j}(\mathcal{M}), \rho_{i \rightarrow j}(N_1), \rho_{i \rightarrow j}(\mathcal{N}_2), \mathcal{N}_3) \mid (\mathcal{M}, N_1, \mathcal{N}_2, \mathcal{N}_3) \in F(X)\}$$

$$\begin{aligned} \text{wobei } \rho_{i \rightarrow j}(\mathcal{M}) &= \langle \langle \rho_{i \rightarrow j}(i'), \rho_{i \rightarrow j}(j') \rangle \mid \langle i', j' \rangle \in \mathcal{M} \rangle, \\ \rho_{i \rightarrow j}(N_1) &= \{ (t, \rho_{i \rightarrow j}(i')) \mid (t, i') \in N_1 \} \text{ und} \\ \rho_{i \rightarrow j}(\mathcal{N}_2) &= \langle (t, \rho_{i \rightarrow j}(i')) \mid (t, i') \in \mathcal{N}_2 \rangle. \end{aligned}$$

Es gibt r_t , $1 \leq t \leq l$, knotendisjunkte Wege in $\text{val}(X)$ zwischen zwei terminalen Knoten u_t und u_{t+l} , genau dann wenn $F(X)$ ein 4-Tupel $(\mathcal{M}, N_1, \mathcal{N}_2, \mathcal{N}_3)$ enthält, so dass \mathcal{N}_3 r_t natürliche Zahlen t enthält.

Theorem 7.12.2 *Zu einem gegebenen k -Ausdruck X , für eine feste natürliche Zahl k , kann das knotendisjunkte Wege Problem für eine feste Anzahl von Knotenpaaren für $\text{val}(X)$ in Polynomzeit gelöst werden.*

Die Datenstrukturen $F(X)$ zur Lösung der folgenden Probleme lassen sich wesentlich einfacher definieren, wenn der Ausdruck X mit NLC-Weite Operationen und nicht mit Cliquesweite Operationen aufgebaut ist. Deshalb verwenden wir in den nächsten drei Beispielen anstelle eines Cliquesweite k -Ausdrucks einen NLC-Weite k -Ausdruck.

7.13 Knotengrad beschränkte Teilgraphen

Wir lösen hier das Problem *Kubischer Teilgraph* (in einem gegebenen Graphen wird ein Teilgraph gesucht in dem jeder Knoten den Grad drei hat), die hier vorgestellte Lösungsmethode ist jedoch für beliebige Knotengrade anwendbar.

Problem 7.13.1 (Kubischer Teilgraph)

EINGABE: *Ein Graph $G = (V, E)$.*

FRAGE: *Gibt es eine nicht leere Teilmenge $E' \subseteq E$, so dass im Graphen (V, E') jeder Knoten den Grad 0 oder 3 hat?*

$G = (V, E, \text{lab}) = \text{val}(X)$ sei ein Graph für einen NLC-Weite k -Ausdruck X . Zu einer natürlichen Zahl s und einer Teilmenge $E' \subseteq E$ der Kanten für die der Graph $G' = (V, E', \text{lab})$ nur Knoten mit Grad höchstens s enthält, definieren wir eine Multimenge \mathcal{M} . \mathcal{M} enthält für jeden Knoten aus V ein Paar $(\text{lab}(u), \text{deg}_{G'}(u))$. $F(X)$ sei die Menge aller paarweise verschiedener Multimengen \mathcal{M} bezüglich s und allen so definierten Knotenteilmengen E' . $F(X)$ hat dann höchstens

$$(|V| + 1)^{k \cdot (s+1)}$$

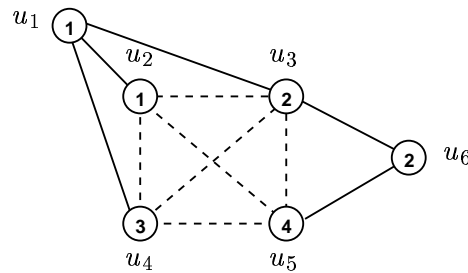
paarweise verschiedene Multimengen. Jede Multimenge enthält $|V|$ Paare (l, d) , wobei l eine Knotenmarkierung aus $[k]$ ist und d eine natürliche Zahl zwischen 0 und s . Abbildung 7.10 zeigt einen kubischen Teilgraphen eines Graphen $\text{val}(X)$ und die dazugehörige Multimenge $\mathcal{M} \in F(X)$.

Die Mengen $F(\bullet_i)$, $F(X \times_S Y)$ und $F(\circ_R(X))$ können wie folgt bestimmt werden.

1. $F(\bullet_i) = \{\langle (i, 0) \rangle\}$, $i \in [k]$
2. Ausgehend von der Menge $D = \{\langle \rangle\} \times F(X) \times F(Y)$ wird D um alle Tripel erweitert die man aus einem Tripel $(\mathcal{M}, \mathcal{M}', \mathcal{M}'') \in D$ erhalten kann, indem man
 - (a) ein Paar (l', d') aus \mathcal{M}' oder ein Paar (l'', d'') aus \mathcal{M}'' entfernt und dieses Paar in \mathcal{M} einfügt oder
 - (b) ein Paar (l', d') aus \mathcal{M}' entfernt, höchstens r , $1 \leq r \leq s - d'$, Paare $(l''_1, d''_1), \dots, (l''_r, d''_r)$ aus \mathcal{M}'' mit $(l', l''_t) \in S$ und $d''_t < s$ für $t = 1, \dots, r$ auswählt, das Paar $(l', d' + r)$ in \mathcal{M} einfügt und jedes (l''_t, d''_t) durch $(l''_t, d''_t + 1)$ ersetzt ohne es aus \mathcal{M}'' zu entfernen.

$$F(X \times_S Y) = \{\mathcal{M} \mid (\mathcal{M}, \langle \rangle, \langle \rangle) \in D\}$$

3. $F(\circ_R(X)) = \{\langle (R(l_1), d_1), \dots, (R(l_t), d_t) \rangle \mid \langle (l_1, d_1), \dots, (l_t, d_t) \rangle \in \mathcal{M}\}$



$$F(X) = \left\{ \begin{array}{c} \dots \\ \langle (1, 0), (1, 3), (2, 3), (3, 3), (4, 3), (2, 0) \rangle \\ \dots \end{array} \right\}$$

Abbildung 7.10: Ein kubischer Teilgraph in einem Graphen $G = \text{val}(X)$. $F(X)$ enthält die zugehörige Multimenge $\langle (\text{lab}(u_1), \text{deg}(u_1)), \dots, (\text{lab}(u_6), \text{deg}(u_6)) \rangle$.

Es gibt einen Teilgraphen von $\text{val}(X)$ mit mindestens einer Kante in dem alle Knoten den Grad 0 oder 3 haben, genau dann wenn es eine Multimenge $\mathcal{M} \in F(X)$ gibt, so dass für alle $(l, d) \in \mathcal{M}$ gilt $d = 3$ oder $d = 0$ und es mindestens ein $(l, d) \in \mathcal{M}$ mit $d = 3$ gibt.

Theorem 7.13.2 *Zu einem gegebenen k -Ausdruck X , für eine feste natürliche Zahl k , ist das Problem Kubischer Teilgraph für $\text{val}(X)$ in Polynomzeit lösbar.*

Mit dem Ergebnis aus Theorem 3.2.1 ist das Problem Kubischer Teilgraph sogar ohne einen gegebenen k -Ausdruck des Eingabegraphen effizient lösbar. Man testet zuerst ob der gegebene Graph G einen $K_{3,3}$ als Teilgraphen enthält. Falls ja, so hat G einen kubischen Teilgraphen und sonst hat G nach Theorem 3.2.1 Baumweite höchstens $6k - 1$. Da das Problem Kubischer Teilgraph MSO_2 -definierbar ist, ist es auf Graphen mit beschränkter Baumweite nach Theorem 6.2.4 in linearer Zeit lösbar.

7.14 Knotenpartition

Problem 7.14.1 (Knotenpartition)

EINGABE: Ein Graph $G = (V, E)$, eine natürliche Zahl $r < |V|$ und eine positive reelle Zahl $\alpha < 1$.

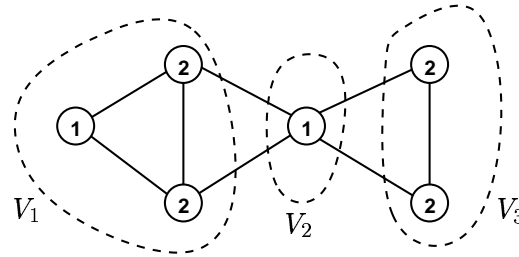
FRAGE: Gibt es in G eine Knotenmenge V_2 der Größe höchstens r , so dass G ohne die Knoten aus V_2 in zwei disjunkte Mengen V_1, V_3 der Größe höchstens $\alpha|V|$ aufgeteilt werden kann?

Es sei $G = (V, E, \text{lab}) = \text{val}(X)$ ein Graph für einen NLC-Weite k -Ausdruck X . $F(X)$ sei die Menge aller 5-Tupel

$$(\text{lab}(V_1), |V_1|, |V_2|, \text{lab}(V_3), |V_3|)$$

für alle disjunkten Partitionen der Knotenmenge V in drei disjunkte Mengen V_1, V_1, V_3 , so dass G keine Kante zwischen einem Knoten aus V_1 und einem Knoten aus V_3 enthält. Dann gibt es in $F(X)$ höchstens

$$2^k \cdot (|V| + 1) \cdot (|V| + 1) \cdot 2^k \cdot (|V| + 1)$$



$$F(X) = \{ \dots (\{1, 2\}, 3, 1, \{2\}, 2) \dots \}$$

Abbildung 7.11: Die Knotenmenge des Graphen $G = \text{val}(X)$ ist in drei Mengen V_1, V_2 und V_3 eingeteilt. $F(X)$ enthält das zugehörige 5-Tupel.

paarweise verschiedene 5-Tupel. Abbildung 7.11 zeigt eine Knotenpartition eines Graphen $\text{val}(X)$ und das zugehörige 5-Tupel in $F(X)$.

Die Mengen $F(\bullet_i)$, $F(X \times_S Y)$ und $F(\circ_R(X))$ werden wie folgt bestimmt.

1. $F(\bullet_i) = \{(\{i\}, 1, 0, \emptyset, 0), (\emptyset, 0, 1, \emptyset, 0), (\emptyset, 0, 0, \{i\}, 1)\}$, $i \in [k]$
2. $F(X \times_S Y)$ besteht aus der Menge aller 5-Tupel $(L_1, x_1, x_2, L_3, x_3)$, die aus zwei Tupeln $(L'_1, x'_1, x'_2, L'_3, x'_3) \in F(X)$ und $(L''_1, x''_1, x''_2, L''_3, x''_3) \in F(Y)$ so konstruiert werden dass $L_1 = L'_1 \cup L''_1$, $x_1 = x'_1 + x''_1$, $x_2 = x'_2 + x''_2$, $L_3 = L'_3 \cup L''_3$, $x_3 = x'_3 + x''_3$, falls es kein Paar $(i, j) \in S$ gibt, so dass $i \in L'_1$ und $j \in L''_3$ oder $i \in L''_3$ und $j \in L'_1$.
3. $F(\circ_R(X)) = \{(R(L_1), x_1, x_2, R(L_3), x_3) \mid (L_1, x_1, x_2, L_3, x_3) \in F(X)\}$.

Es gibt in $\text{val}(X)$ eine Knotenmenge V_2 der Größe höchstens r , so dass $\text{val}(X)$ ohne die Knoten aus V_2 in zwei disjunkte Mengen V_1, V_3 der Größe höchstens $\alpha|V|$ aufgeteilt werden kann, genau dann wenn es ein $(L_1, x_1, x_2, L_3, x_3) \in F(X)$ gibt, so dass

$$x_1 \leq \alpha \cdot |V|, \quad x_3 \leq \alpha \cdot |V| \quad \text{und} \quad x_2 \leq r.$$

Theorem 7.14.2 *Zu einem gegebenen k -Ausdruck X , für eine feste natürliche Zahl k , ist das Problem Knotenpartition für $\text{val}(X)$ für gegebenes r und α in Polynomzeit lösbar.*

7.15 Kantenpartition

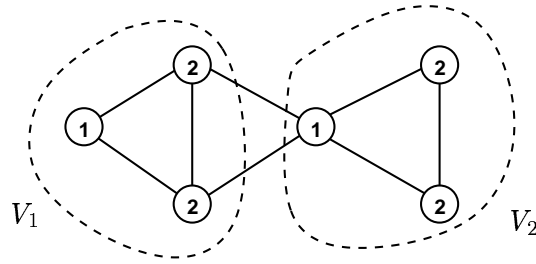
Problem 7.15.1 (Kantenpartition)

EINGABE: Ein Graph $G = (V, E)$, eine natürliche Zahl $r \leq |E|$ und eine positive reelle Zahl $\alpha < 1$.

FRAGE: Gibt es eine Partition der Knotenmenge des Graphen G in zwei disjunkte Mengen V_1, V_2 der Größe höchstens $\alpha|V|$, so dass es höchstens r Kanten zwischen einem Knoten aus V_1 und einem Knoten aus V_2 gibt?

Es sei $G = (V, E, \text{lab}) = \text{val}(X)$ für einen NLC-Weite k -Ausdruck X . $F(X)$ sei die Menge aller Tripel

$$(\langle \text{lab}(u_1), \dots, \text{lab}(u_r) \rangle, x, \langle \text{lab}(u_{r+1}), \dots, \text{lab}(u_n) \rangle)$$



$$F(X) = \{\dots (\langle 1, 2, 2 \rangle, 2, \langle 1, 2, 2 \rangle) \dots\}$$

Abbildung 7.12: Die Knotenmenge des Graphen $G = \text{val}(X)$ ist in zwei Mengen V_1 und V_2 eingeteilt zwischen denen es zwei Kanten gibt. $F(X)$ enthält das zugehörige Tripel.

für alle disjunkten Partitionen von $V = \{u_1, \dots, u_n\}$ in $V_1 = \{u_1, \dots, u_r\}$ und $V_2 = \{u_{r+1}, \dots, u_n\}$, so dass x die Anzahl der Kanten zwischen Knoten aus V_1 und Knoten aus V_2 angibt. $F(X)$ enthält dann höchstens

$$(|V| + 1)^k \cdot (|E| + 1) \cdot (|V| + 1)^k$$

Tripel. Abbildung 7.12 zeigt eine Kantenpartition eines Graphen $\text{val}(X)$ und das zugehörige Tripel in $F(X)$.

Die Mengen $F(\bullet_i)$, $F(X \times_S Y)$ und $F(\circ_R(X))$ können wie folgt bestimmt werden.

1. $F(\bullet_i) = \{(\langle i \rangle, 0, \langle \rangle), (\langle \rangle, 0, \langle i \rangle)\}$, $i \in [k]$.
2. $F(X \times_S Y)$ besteht aus der Menge aller Tripel $(\mathcal{M}_1, x, \mathcal{M}_2)$ die aus zwei Tripeln $(\mathcal{M}'_1, x', \mathcal{M}'_2) \in F(X)$ und $(\mathcal{M}''_1, x'', \mathcal{M}''_2) \in F(Y)$ erhalten werden können, wobei $\mathcal{M}_1 = \mathcal{M}'_1 \cup \mathcal{M}''_1$, $\mathcal{M}_2 = \mathcal{M}'_2 \cup \mathcal{M}''_2$ und

$$x = x' + x'' + \sum_{(i,j) \in S} \psi(\mathcal{M}'_1, i) \cdot \psi(\mathcal{M}''_2, j) + \sum_{(i,j) \in S} \psi(\mathcal{M}'_2, i) \cdot \psi(\mathcal{M}''_1, j).$$

3. $F(\circ_R(X)) = \{(R(\mathcal{M}_1), x, R(\mathcal{M}_2)) \mid (\mathcal{M}_1, x, \mathcal{M}_2) \in F(X)\}$.

Es gibt in $\text{val}(X)$ eine Partition der Knotenmenge in zwei disjunkte Mengen V_1, V_2 der Größe höchstens $\alpha|V|$, so dass es höchstens r Kanten zwischen einem Knoten aus V_1 und einem Knoten aus V_2 gibt, genau dann wenn es ein Tripel $(\mathcal{M}_1, x, \mathcal{M}_2) \in F(X)$ gibt mit

$$\sum_{i \in [k]} \psi(\mathcal{M}_1, i) \leq \alpha|V|, \sum_{i \in [k]} \psi(\mathcal{M}_2, i) \leq \alpha|V| \text{ und } x \leq r.$$

Theorem 7.15.2 *Zu einem gegebenen k -Ausdruck X , für eine feste natürliche Zahl k , ist das Problem Kantenpartition für $\text{val}(X)$ für gegebenes r und α in Polynomzeit lösbar.*

In [Wan94c] ist das Simple Max Cut Problem auf Graphen mit beschränkter NLC-Weite mit einer ähnlichen Datenstruktur gelöst worden.

7.16 Anmerkungen

Die der hier vorgestellten Algorithmen haben alle eine Laufzeit die exponentiell in der Cliquesweite bzw. der NLC-Weite k des Eingabegraphen ist. Damit sind die Algorithmen nur für kleine Werte von k praktisch interessant. Eine genaue Laufzeitabschätzung ist stets von der Implementierung abhängig. Es ist bisher noch offen, ob es Algorithmen für schwere Graphenprobleme auf Graphen mit beschränkter Cliquesweite bzw. auf Graphen mit beschränkter NLC-Weite gibt, die eine Laufzeit aus $O(n^c)$ haben, wobei c unabhängig von k und der Größe des Graphen ist.

Weiterhin ist es interessant Probleme auf Graphen mit beschränkter Cliquesweite bzw. beschränkter NLC-Weite zu lösen ohne dabei einen k -Ausdruck gegeben zu haben, da das Finden eines solchen Ausdrucks möglicherweise schwierig ist. Einen möglichen Ansatz zum Lösen von schwierigen Graphenproblemen ohne gegebene Dekomposition des Eingabegraphen liefern Spinrad und andere Autoren in [BJRS02] über robuste Algorithmen.

Die bisher auf Graphen mit beschränkter Cliquesweite bzw. beschränkter NLC-Weite gelösten Probleme sind alle Knotenpartitionsprobleme, schwieriger scheinen hingegen Kantenpartitionsprobleme. Das Problem die Kantenmenge eines Graphen in möglichst wenige Mengen einzuteilen die jeweils keine zwei adjazenten Kanten enthalten bezeichnet man als *Kantenfärbungsproblem*. Die Anzahl der benötigten Mengen heißt der *Chromatic Index* des Graphen. Die Komplexität des Problems Kantenfärbung ist selbst für Cographen noch offen [CE91].

Kapitel 8

NP-vollständige Probleme für Graphen mit beschränkter Cliquesweite

Da Cographen und Graphklassen mit beschränkter Baumweite beschränkte Cliquesweite und beschränkte NLC-Weite haben, sind die Graphenprobleme die für Cographen und für Graphklassen mit beschränkter Baumweite NP-vollständig sind auch für Graphklassen mit beschränkter Cliquesweite und für Graphklassen mit beschränkter NLC-Weite NP-vollständig. Das Achromatic Number Problem (siehe [GT5] in [GJ79]) ist NP-vollständig für Cographen (siehe [Bod89]) und damit auch NP-vollständig für Graphen mit Cliquesweite höchstens 2 und für Graphen mit NLC-Weite 1.

Bisher sind jedoch so gut wie keine Graphenprobleme bekannt, die auf Graphklassen mit beschränkter Cliquesweite und auf Graphklassen mit beschränkter NLC-Weite NP-vollständig sind, aber auf Cographen und Graphklassen mit beschränkter Baumweite in Polynomzeit entschieden werden können. Wir zeigen hier, dass das Knotendisjunkte Wege Problem auf Cographen in linearer Zeit lösbar ist und für Graphen mit NLC-Weite höchstens 4 und für Graphen mit Cliquesweite höchstens 7 NP-vollständig ist. Auf Graphklassen mit beschränkter Baumweite kann das Problem in linearer Zeit gelöst werden [Sch94].

8.1 Knotendisjunkte Wege

Das Knotendisjunkte Wege Problem ist in Kapitel 7.12 definiert. Wir betrachten hier das Knotendisjunkte Wege Problem für beliebig viele Knotenpaare, d.h. die Anzahl der Knotenpaare gehört nun mit zur Eingabe des Problems. Die NP-Vollständigkeit dieses Problems wurde in der Schrift [MP95] gezeigt.

8.1.1 Linearzeit Lösung für Cographen

Wir zeigen hier, dass das Knotendisjunkte Wege Problem auf Cographen effizient lösbar ist. Es sei G ein Cograph, der durch einen NLC-Weite 1-Ausdruck X gegeben ist. Der Ausdruck X und ein Cobaum für G können in linearer Zeit mit Hilfe des Erkennungsalgorithmus für Cographen in [CPS85] bestimmt werden.

Wir nennen die Knoten u_1, \dots, u_{2l} , wie in Kapitel 7.12, *terminale Knoten* und die restlichen Knoten *nichtterminale Knoten*. Wir nehmen im Folgenden an, dass wir wissen welcher Teilausdruck \bullet_1 zu welchem terminalen Knoten u_t , $1 \leq t \leq 2 \cdot l$ gehört.

Da Cographen genau die Graphen sind die keinen induzierten Weg der Länge drei enthalten, hat das Knotendisjunkte Wege Problem für Cographen, genau dann eine Lösung wenn alle paarweise verschiedenen Wege genau drei Knoten enthalten, d.h. einen Startknoten u_t , einen mittleren Knoten $u_{t,j}$ und einen Endknoten u_{t+l} , $1 \leq t \leq l$, $1 \leq j \leq r_t$. Wege der Länge eins können durch Testen von Adjazenzen vernachlässigt werden.

Diese Wege $(u_t, u_{t,j}, u_{t+l})$ können nur in einem Teilausdruck $Y \times_{\{(1,1)\}} Z$ von X , so dass $\text{val}(Y)$ ($\text{val}(Z)$) beide terminalen Knoten u_t und u_{t+l} enthält und $\text{val}(Z)$ (bzw. $\text{val}(Y)$) den nichtterminalen Knoten $u_{t,j}$ enthält, erzeugt werden. Da alle Knoten die gleiche Markierung 1 haben, wird durch die Operation $\times_{\{(1,1)\}}$ jeder nichtterminale Knoten in $\text{val}(Y)$ ($\text{val}(Z)$) mit jedem terminalen Knoten in $\text{val}(Z)$ (bzw. $\text{val}(Y)$) verbunden.

Es sei G' ein induzierter Teilgraph von G und G' sei durch einen Teilausdruck X' von X definiert. Es sei

1. $R(X')$ die Summe aller r_t , so dass die beiden terminalen Knoten u_t und u_{t+l} in G' enthalten sind,
2. $F(X')$ die Anzahl der nichtterminalen Knoten in G' und
3. $M(X')$ die maximale Anzahl knotendisjunkter Wege $(u_t, u_{t,j}, u_{t+l})$, $1 \leq t \leq l$, $1 \leq j \leq r_t$ im Teilgraph G' , so dass alle Knoten $u_{t,j}$ nichtterminale Knoten sind.

Das Knotendisjunkte Wege Problem hat für G genau dann eine Lösung wenn $R(X) = M(X)$. Wir zeigen nun, dass alle Zahlen $R(X')$, $F(X')$ und $M(X')$ in linearer Zeit bestimmt werden können.

- Es sei $X' = \bullet_1$.

Falls der Knoten in $\text{val}(X')$ ein nichtterminaler Knoten ist, dann ist

$$R(X') = 0, \quad F(X') = 1 \quad \text{und} \quad M(X') = 0$$

sonst ist

$$R(X') = 0, \quad F(X') = 0 \quad \text{und} \quad M(X') = 0.$$

- Es sei $X' = Y \times_{\emptyset} Z$

Es sei $R_{Y,Z}$ die Summe aller r_t , $1 \leq t \leq l$, so dass der Graph $\text{val}(Y)$ einen terminalen Knoten u_t oder u_{t+l} und der Graph $\text{val}(Z)$ den anderen terminalen Knoten u_{t+l} bzw. u_t enthält. Dann ist

$$\begin{aligned} R(X') &= R(Y) + R(Z) + R_{Y,Z}, \\ F(X') &= F(Y) + F(Z) \quad \text{und} \\ M(X') &= M(Y) + M(Z). \end{aligned}$$

Dazu beachte man, dass die Operation \times_{\emptyset} keine zusätzlichen Kanten erzeugt.

- Es sei $X' = Y \times_{\{(1,1)\}} Z$

Dann gilt

$$\begin{aligned}
R(X') &= R(Y) + R(Z) + R_{Y,Z}, \\
F(X') &= F(Y) + F(Z) \text{ und} \\
M(X') &= M(Y) + M(Z) + \min\{R(Y) - M(Y), F(Z) - M(Z)\} \\
&\quad + \min\{R(Z) - M(Z), F(Y) - M(Y)\}
\end{aligned}$$

Die $F(Z) - M(Z)$ nicht gebrauchten nichtterminalen Knoten aus $\text{val}(Z)$ können benutzt werden um noch $R(Y) - M(Y)$ aufzubauende Wege in $\text{val}(Y)$ zu konstruieren und umgekehrt.

Die Berechnungen können durch einen Bottom-Up Durchlauf von X bzw. dem zugehörigen NLC-Weite 1-Ausdrucksbaum zu X in linearer Zeit durchgeführt werden. Die Werte $R_{Y,Z}$ können in einer Vorverarbeitung berechnet werden. Dazu müssen in einem NLC-Weite 1-Ausdrucksbaum für X für jedes Paar (u_t, u_{t+l}) , $1 \leq t \leq l$, der nächste gemeinsame Vorfahre bestimmt werden. Diese Vorverarbeitung kann nach einem Ergebnis in [AGKR02] in linearer Zeit durchgeführt werden. Damit haben wir das folgende Theorem gezeigt.

Theorem 8.1.1 *Das Knotendisjunkte Wege Problem kann auf Cographen, Graphen mit NLC-Weite 1 und auf Graphen mit Cliquenweite höchstens 2 in linearer Zeit gelöst werden.*

8.1.2 NP-Vollständigkeit für NLC_4 und CW_7

Betrachten wir Wege die keine Kante gemeinsam haben, so erhalten wir das *Kantendisjunkte Wege Problem*. Das Kantendisjunkte Wege Problem ist für series-parallele Graphen, d.h. für Graphen mit Baumweite höchstens 2 NP-vollständig [NVZ01].

Das Kantendisjunkte Wege Problem kann für einen Graphen G mit einem Algorithmus für das Knotendisjunkte Wege Problem auf dem Linegraphen von G gelöst werden. Der *Linegraph* zu einem Graphen $G = (V_G, E_G)$ besitzt einen Knoten für jede Kante in G und eine Kante zwischen zwei Knoten, falls die entsprechenden Kanten in G adjazent sind.

Wir zeigen, dass der Linegraph eines Graphen mit Baumweite k die NLC-Weite höchstens $k + 2$ hat. Wir verwenden die zur Baumweite äquivalente Definition der partiellen k -Bäume aus Definition 2.2.2.

Theorem 8.1.2 *Es sei G ein partieller k -Baum, dann hat der Linegraph von G die NLC-Weite höchstens $k + 2$.*

Beweis Es genügt zu zeigen, dass der Linegraph eines k -Baumes G NLC-Weite höchstens $k + 2$ hat, da der Linegraph von jedem Teilgraph von G ein induzierter Teilgraph des Linegraphen von G ist und die Menge der Graphen mit NLC-Weite höchstens $k + 2$ bezüglich induzierten Teilgraphen abgeschlossen ist.

Es sei $G = (V_G, E_G)$ ein k -Baum mit n Knoten und $o = (u_1, \dots, u_n)$ eine Anordnung der n Knoten von G . Es seien $N^-(G, o, i)$ und $N^+(G, o, i)$, $i = 1, \dots, n$, die Nachbarn u_j von Knoten u_i mit $j < i$ bzw. $j > i$. D.h.

$$N^-(G, o, i) := \{u_j \mid \{u_i, u_j\} \in E_G \wedge j < i\}$$

und

$$N^+(G, o, i) := \{u_j \mid \{u_j, u_i\} \in E_G \wedge j > i\}.$$

Die Knotenreihenfolge (u_1, \dots, u_n) heißt *perfekte Eliminationsordnung* (PEO) für G , falls die Knotenmengen $N^+(G, o, i)$ für $i = 1, \dots, n$ einen vollständigen Teilgraphen in G induzieren.

Es gibt immer eine PEO $o = (u_1, \dots, u_n)$ für G , so dass die Knoten aus jeder Menge $N^+(G, o, i)$ für $i = 1, \dots, n - k$ einen vollständigen Teilgraphen mit k Knoten und die Knoten in jeder Menge $N^+(G, o, i)$ für $i = n - k + 1, \dots, n$ einen vollständigen Teilgraphen mit $n - i$ Knoten von G induzieren. Dazu können wir die umgekehrte Anordnung der Knoten aus der rekursiven Definition eines k -Baumes wählen.

Der Aufbau des k -Baumes G kann wie folgt durch einen Baum $T(G, o) = (V_T, E_T)$ dargestellt werden. Es sei $o = (u_1, \dots, u_n)$ eine perfekte Eliminationsordnung für G .

$$\begin{aligned} V_T &:= V_G \\ E_T &:= \{\{u_i, u_j\} \in E_G \mid i < j \wedge \forall j', i < j' < j, \{u_i, u_{j'}\} \notin E_G\}. \end{aligned}$$

Der Graph $T(G, o)$ ist ein Baum, da jeder Knoten u_i , $i < n$, aus $T(G, o)$ genau mit einer Kante $\{u_i, u_j\}$ für $j > i$ inzident ist.

Es sei col eine $k + 1$ -Färbung von G , d.h., $\text{col} : V_G \rightarrow [k + 1]$ ist eine Abbildung mit $\text{col}(u_i) \neq \text{col}(u_j)$ für alle $\{u_i, u_j\} \in E_G$. Man sieht leicht, dass jeder k -Baum $k + 1$ -färbbar ist, wenn man den Knoten u_i mit der Farbe markiert die nicht von den Knoten in der Menge $N^+(G, o, i)$ benutzt wird.

Wir definieren nun rekursiv für $i = 1, \dots, n$ einen NLC-Weite $k + 2$ -Ausdruck X_i . Es sei $N^-(T(G, o), o, i) = \{u_{j_1}, \dots, u_{j_m}\}$ und $N^+(G, o, i) = \{u_{l_1}, \dots, u_{l_r}\}$. Beachte, dass $\{u_{j_1}, \dots, u_{j_m}\}$ durch den Baum $T(G, o)$ und $\{u_{l_1}, \dots, u_{l_r}\}$ durch den Graph G definiert ist.

1. Falls $m = 1$, dann sei $Y_i = X_{j_1}$. Falls $m > 1$, dann sei

$$Y_i = ((X_{j_1} \times_I X_{j_2}) \times_I \dots) \times_I X_{j_m},$$

wobei $I = \{(s, s) \mid s \in [k + 1]\}$ die Identität zwischen den Markierungen $1, \dots, k + 1$ sei.

Der Graph $\text{val}(Y_i)$ besteht aus der disjunkten Vereinigung von m Graphen $\text{val}(X_{j_1}), \dots, \text{val}(X_{j_m})$, in der gleichmarkierte Knoten aus verschiedenen Graphen durch eine Kante verbunden werden. Man beachte, dass die Relation I nur die Knotenmarkierungen $1, \dots, k + 1$ enthält. Die Markierung $k + 2$ wird nur für die Knoten verwandt, die im Weiteren nicht mit anderen Knoten verbunden werden.

2. Falls $r > 0$ dann sei Z_i ein NLC-Weite $k + 1$ -Ausdruck der einen vollständigen Graphen mit r Knoten definiert, die mit $\text{col}(u_{l_1}), \dots, \text{col}(u_{l_r})$ markiert sind. Man beachte, dass diese $r \leq k$ Markierungen paarweise verschieden sind und nicht die Farbe des Knoten u_i enthalten.

3. Wir definieren dann

$$X_i = \begin{cases} \circ_R(Y_i \times_S Z_i) & \text{falls } m > 0 \text{ und } r > 0 \\ Z_i & \text{falls } m = 0 \text{ und } r > 0 \\ \circ_R(Y_i) & \text{falls } m > 0 \text{ und } r = 0 \end{cases}$$

wobei

$$S = \{(s, s) \mid s \in [k + 1]\} \cup \{(\text{col}(u_i), s) \mid s \in [k + 1]\}$$

und

$$R(s) = \begin{cases} s & \text{falls } s \neq \text{col}(u_i) \\ k + 2 & \text{falls } s = \text{col}(u_i) \end{cases}.$$

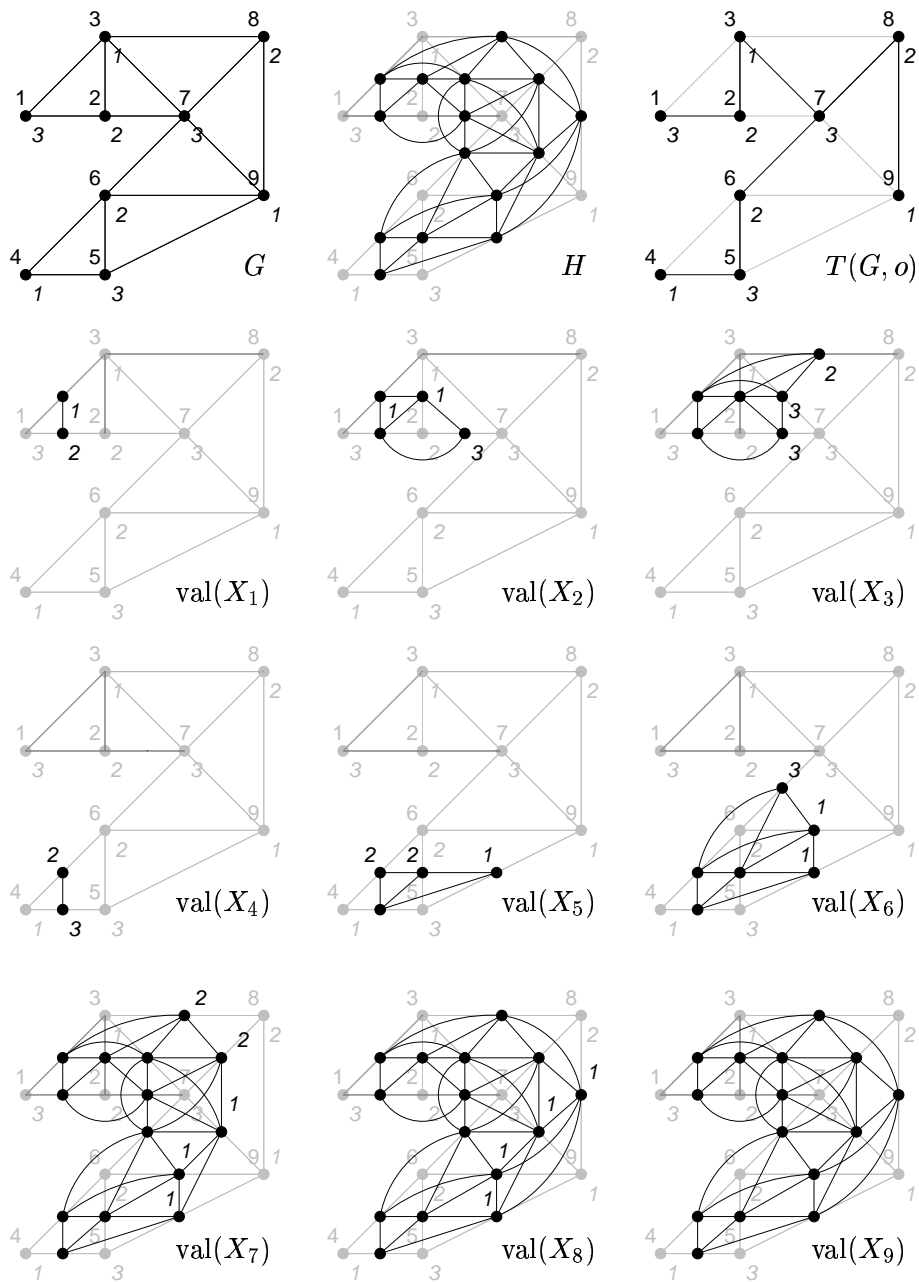


Abbildung 8.1: Ein 2-Baum G und der zugehörige Linegraph H . Die (Nicht-Italic) Markierungen der Knoten von G entsprechen einer perfekten Eliminationsordnung. Die Italic Indizes entsprechen einer 3-Färbung. Die Abbildung zeigt auch den Baum $T(G, o)$ und alle markierten Graphen der NLC-Weite 4-Ausdrücke X_1 bis X_9 . Die Italic Indizes an den Graphen $\text{val}(X_1)$ bis $\text{val}(X_9)$ entsprechen den Markierungen der Knoten. Die Knoten ohne Markierung sind mit 4 markiert.

Abbildung 8.1 zeigt ein Beispiel für den Aufbau des Linegraphen. Wir zeigen nun, dass der NLC-Weite $k + 2$ -Ausdruck X_n den Linegraph des k -Baumes G definiert.

Nach der Definition von Z_i , $i = 1, \dots, n$, gibt es eine Bijektion zwischen den Knoten aus G und den Knoten des Graphen $\text{val}(X_i)$. Ein Knoten im Graph $\text{val}(Z_i)$ der mit s markiert ist, entspricht einer Kante zwischen Knoten u_i und dem Knoten von $N^+(G, o, i)$ der mit s markiert ist. Deshalb gibt es auch eine Bijektion zwischen den Kanten aus G und den Knoten aus dem Graph $\text{val}(X_n)$. Die Knoten im Graph $\text{val}(X_i)$ die mit einer Markierung aus $[k+1]$ markiert sind, entsprechen genau den Kanten aus G mit einem Endknoten aus $\{u_1, \dots, u_i\}$ und einem Endknoten aus $\{u_{i+1}, \dots, u_n\}$.

Wir wollen die Graphen $\text{val}(X_i)$, $i = 1, \dots, n$, näher untersuchen. Es sei G_i , $1 \leq i \leq n$, der Teilgraph von G der durch die Knoten $\{u_1, \dots, u_i\}$ induziert wird. Es sei G'_i die Zusammenhangskomponente von G_i zu der u_i gehört und es sei \tilde{G}_i der Graph G'_i der durch alle Kanten (und deren Endknoten) aus G erweitert wird die genau einen Endknoten in G'_i haben. Eine Induktion nach i zeigt nun, dass X_i den Linegraph von \tilde{G}_i definiert.

Induktionsanfang: Es sei $i = 1$.

Der Graph \tilde{G}_1 besteht aus $1 + |N^+(G, o, 1)|$ Knoten und $|N^+(G, o, 1)|$ Kanten zwischen u_1 und den Knoten aus $N^+(G, o, 1)$. Hier definiert der Graph $\text{val}(X_1)$ einen vollständigen Graphen mit $|N^+(G, o, 1)|$ Knoten, die mit den Farben der Knoten aus $N^+(G, o, 1)$ markiert sind. Der Graph der durch den Ausdruck X_1 definiert wird ist offensichtlich der Linegraph des Graphen \tilde{G}_1 .

Induktionsschritt: Es sei $i > 1$.

Es sei $N^-(T(G, o), o, i) = \{u_{j_1}, \dots, u_{j_m}\}$ und $N^+(G, o, i) = \{u_{l_1}, \dots, u_{l_r}\}$. Falls $m = 0$, dann besteht \tilde{G}_i wie im Falle $i = 1$ aus $1 + |N^+(G, o, i)|$ Knoten und $|N^+(G, o, i)|$ Kanten zwischen u_i und den Knoten aus $N^+(G, o, i)$. Hier definiert der Ausdruck X_i ebenfalls einen vollständigen Graphen mit $|N^+(G, o, i)|$ Knoten, die mit den Farben der Knoten aus der Menge $N^+(G, o, i)$ markiert sind.

Falls $m > 0$, dann definieren wir zunächst einen Ausdruck Y_i der aus der disjunkten Vereinigung der Graphen $\text{val}(X_{j_1}), \dots, \text{val}(X_{j_m})$ besteht in welchen gleich markierte Knoten aus verschiedenen Graphen durch eine Kante verbunden werden. Die gleichmarkierten Knoten aus verschiedenen Graphen müssen durch Kanten verbunden werden, da sie Kanten entsprechen werden, die einen Endknoten aus $\{u_1, \dots, u_{i-1}\}$ haben und den gleichen Endknoten aus der Menge $\{u_i, \dots, u_n\}$ besitzen.

Nach der Induktionsannahme und aufgrund der zusätzlich eingefügten Kanten, beschreibt der Ausdruck $Y_i \times_S Z_i$ den Linegraphen von \tilde{G}_i . Die Relation S verbindet einen Knoten u aus dem Graph $\text{val}(Y_i)$ und einen Knoten v aus dem Graph $\text{val}(Z_i)$ falls eine der folgenden Eigenschaften erfüllt ist.

1. Beide Knoten haben die gleiche Markierung aus der Menge $[k+1]$.

Dann entsprechen u und v zwei adjazenten Kanten $\{u_{i'}, u_{j'}\}$ bzw. $\{u_i, u_{j'}\}$ aus G mit $i' < i < j'$.

2. Die Markierung von u ist die Farbe $\text{col}(u_i)$ von u_i in G .

Dann entspricht u einer Kante $\{u_{i'}, u_i\}$ aus G mit $i' < i$. Diese Kanten sind alle mit den Kanten die Knoten aus $\text{val}(Z_i)$ entsprechen, adjazent.

Die Ummarkierung \circ_R ändert die Markierung $\text{col}(u_i)$ in die Markierung $k+2$, da die Knoten die mit $\text{col}(u_i)$ markiert sind nun nur noch solchen Kanten e aus G entsprechen für die alle den zu e adjazenten Kanten entsprechenden Knoten schon im Graph $\text{val}(X_i)$

vorhanden sind. Da \tilde{G}_n der Graph G ist, haben wir einen NLC-Weite $k + 2$ -Ausdruck für den Linegraphen von G konstruiert. \square

Für $k = 1$ besagt das obige Theorem, dass die Linegraphen von Wäldern NLC-Weite höchstens 3 haben. Da der Linegraph eines Weges P_n der Weg P_{n-1} ist, siehe Abbildung 8.2, und Wege P_n , $n \geq 10$, NLC-Weite 3 haben, ist die Schranke für die NLC-Weite aus Theorem 8.1.2 scharf.

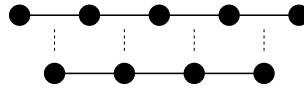


Abbildung 8.2: Der Linegraph des P_n ist der P_{n-1} .

Für einen Graphen $G = (V_G, E_G)$ und $2 \cdot l$ Knoten $(u_1, \dots, u_{2 \cdot l})$ sei $G' = (V_{G'}, E_{G'})$ der Graph G mit $2 \cdot l$ zusätzlichen Knoten $u'_1, \dots, u'_{2 \cdot l}$ und $2 \cdot l$ zusätzlichen Kanten $\{u'_1, u_1\}, \dots, \{u'_{2 \cdot l}, u_{2 \cdot l}\}$. Es sei H der Linegraph von G' und v_i der Knoten aus H der der Kante $\{u'_i, u_i\}$, $1 \leq i \leq 2 \cdot l$, in G' entspricht. Dann gibt es l paarweise verschiedene Knotendisjunkte Wege in H zwischen Knoten v_i und v_{i+l} für $i = 1, \dots, l$ genau dann wenn es es l paarweise verschiedene Kantendisjunkte Wege in G zwischen u_i und u_{i+l} für $i = 1, \dots, l$ gibt. Da G' ein partieller 2-Baum ist, falls G ein partieller 2-Baum ist und da das Kantendisjunkte Wege Problem auf partiellen 2-Bäumen NP-vollständig ist [NVZ01], folgt das folgende Theorem.

Theorem 8.1.3 *Das Knotendisjunkte Wege Problem ist NP-vollständig für Graphen mit NLC-Weite höchstens 4.*

Man kann den Beweis von Theorem 8.1.2 leicht modifizieren und zeigen, dass der Linegraph eines partiellen k -Baumes Cliquesweite höchstens $2k + 3$ hat. Damit folgt aus dem Ergebnis aus [NVZ01] das folgende Theorem.

Theorem 8.1.4 *Das Knotendisjunkte Wege Problem ist NP-vollständig für Graphen mit Cliquesweite höchstens 7.*

Aus Theorem 8.1.2 folgt ebenfalls, dass der Chromatic Index eines Graphen mit beschränkter Baumweite in polynomieller Zeit gefunden werden kann, da das Chromatic Number Problem (Partition in unabhängige Mengen) für Graphen mit beschränkter NLC-Weite in polynomieller Zeit gelöst werden kann, siehe Kapitel 7.4. Dies beweist nochmals ein bereits seit 1990 bekanntes Ergebnis von Bodlaender in [Bod90]. Der Beweis von Theorem 8.1.2 ist konstruktiv, d.h. man kann einen NLC-Weite $k + 2$ -Ausdruck bzw. einen Cliquesweite $2k + 3$ -Ausdruck aus einer gegebenen Baumdekomposition der Weite k aufbauen.

Symbolverzeichnis

Graphen

$\{u, v\}$	ungerichtete Kante zwischen u und v
(u, v)	gerichtete Kante von u nach v
K_n	vollständiger Graph mit n Knoten
$K_{n,m}$	vollständig bipartiter Graph mit n und m Knoten
P_n	Weg der Länge $n - 1$ mit n Knoten
C_n	Kreis der Länge n mit n Knoten
$ V $	Anzahl der Knoten in der Menge V (Größe der Menge V)
$ E $	Anzahl der Kanten in der Menge E
V_G	Knotenmenge des Graphen G
E_G	Kantenmenge des Graphen G
$N_G(u)$	Nachbarschaft des Knotens u in G
$\deg_G(u)$	Grad des Knotens u in G
$\text{dist}_G(u, v)$	Abstand (Distanz) der Knoten u und v in G
$\text{lab}_G(v)$	Markierung des Knotens v in G
$\text{lab}(U)$	Menge der Markierungen der Knoten aus U
$\text{val}(X)$	bezeichnet den zum Ausdruck X gehörigen Graphen
\mathcal{L}	Graphmenge, Graphklasse
$G \times H$	Disjunkte Vereinigung der Graphen G und H und alle Kanten zwischen den Graphen (join)
$G \cup H$	Disjunkte Vereinigung der Graphen G und H (co-join)
$G \approx H$	Graph G und H sind isomorph

Komplexität

P	Klasse der Probleme die in polynomieller Zeit lösbar sind
NP	Klasse der Probleme die nichtdeterministisch in polynomieller Zeit lösbar sind
NPC	Klasse der NP-vollständigen Probleme

$O(f)$ = $\{g : \mathbb{N} \rightarrow \mathbb{N} \mid \exists n_0, c \in \mathbb{N} : \forall n \in \mathbb{N} : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\}$
= Menge der Funktionen die höchstens so stark wachsen wie f

Zahlen und Mengen

\mathbb{N}	Menge der natürlichen Zahlen $\{1, 2, 3, \dots\}$
\mathbb{N}_0	Menge der natürlichen Zahlen und Null $\{0, 1, 2, 3, \dots\}$
\mathbb{Z}	Menge der ganzen Zahlen $\{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$
\mathbb{R}	Menge der reellen Zahlen
$\min A$	Minimum der Elemente in einer Menge A
$\max A$	Maximum der Elemente in einer Menge A
$\log n$	Logarithmus von n
$\lceil r \rceil$	die kleinste natürliche Zahl die größer gleich einer reellen Zahl r ist
$\lfloor r \rfloor$	die größte natürliche Zahl die kleiner gleich einer reellen Zahl r ist
$[k]$	$= \{1, \dots, k\}$ Menge der ganzen Zahlen zwischen 1 und k
M^k	Menge aller k -Tupel (a_1, \dots, a_k) mit Einträgen $a_i \in M$, $1 \leq i \leq k$
2^M	Menge aller Teilmengen einer Menge M
\emptyset	die leere Menge

Literaturverzeichnis

- [ABCC01] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. TSP cuts which do not conform to the template paradigm. In M. Jünger and D. Naddef, editors, *Computational combinatorial optimization: optimal or provably near optimal solutions*, volume 2241 of *LNCS*, pages 261–303. Springer-Verlag, 2001.
- [ACP87] S. Arnborg, D.G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k -tree. *SIAM Journal of Algebraic and Discrete Methods*, 8(2):227–284, 1987.
- [AGKR02] S. Alstrup, C. Gavaille, H. Kaplan, and T. Rauhe. Nearest common ancestors: A survey and a new distributed algorithm. In *Proceedings of the Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 258–264. ACM, 2002.
- [ALS91] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12:308–340, 1991.
- [AP89] S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems on graphs embedded in k -trees. *Discrete Applied Mathematics*, 23:11–24, 1989.
- [Arn85] S. Arnborg. Efficient algorithms for combinatorial problems on graphs with bounded decomposability – A survey. *BIT*, 25:2–23, 1985.
- [Bab98] L. Babel. On graphs with simple P_4 -structure. Preprint, 1998.
- [BDLM02] A. Brandstädt, F. F. Dragan, H.-O. Le, and R. Mosca. New graph classes of bounded clique width. In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 2573 of *LNCS*, pages 57–67. Springer-Verlag, 2002.
- [BHL03] A. Brandstädt, C.T. Hoang, and V.-B. Le. Stability Number of Bull- and Chair-Free Graphs Revisited. *Discrete Applied Mathematics*, 131:39–50, 2003.
- [BJRS02] R. Borie, J.L. Johnson, V. Raghavan, and J.P. Spinrad. Robust polynomial time algorithms on clique-width k graphs. Manuscript, 2002.
- [BLM02a] A. Brandstädt, H.O. Le, and R. Mosca. Chordal co-gem-free and (P_5, gem) -free graphs have bounded clique-width. Manuscript, 2002.
- [BLM02b] A. Brandstädt, H.O. Le, and R. Mosca. Gem- and co-gem-free graphs have bounded clique-width. Manuscript, 2002.
- [BLS99] A. Brandstädt, V.B. Le, and J.P. Spinrad. *Graph Classes: A Survey*. SIAM Monographs on Discrete Math. and Applications. Springer-Verlag, Berlin, Heidelberg, New York, 1999.

-
- [BLV03] A. Brandstädt, H.-O. Le, and J.-M. Vanherpe. Structure and Stability Number of (Chair, Co-P, Gem)-Free Graphs Revisited. *Information Processing Letters*, 86:161–167, 2003.
- [BM83] H. Buer and R.H. Möhring. A fast algorithm for the decomposition of graphs and posets. *Methods of Operations Research*, 8:170–184, 1983.
- [BM03] A. Brandstädt and R. Mosca. On the Structure and Stability Number of P_5 and Co-Chair-Free Graphs. *Discrete Applied Mathematics*, 132:47–65, 2003.
- [BO98] L. Babel and S. Olariu. On the structure of graphs with few P_4 s. *Discrete Applied Mathematics*, 84:1–13, 1998.
- [Bod89] H.L. Bodlaender. Achromatic number is NP-complete for cographs and interval graphs. *Information Processing Letters*, 31(3):135–138, 1989.
- [Bod90] H.L. Bodlaender. Polynomial algorithms for chromatic index and graph isomorphism on partial k -trees. *Journal of Algorithms*, 11(4):631–643, 1990.
- [Bod96] H.L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.
- [Bod97] H.L. Bodlaender. Treewidth: Algorithmic techniques and results. In *Proceedings of Mathematical Foundations of Computer Science*, volume 1295 of *LNCS*, pages 29–36. Springer-Verlag, 1997.
- [Bod98] H.L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209:1–45, 1998.
- [BPT92] R.B. Borie, R.G. Parker, and C.A. Tovey. Automatic generation of lineartime algorithms for predicate calculus description of problems on recursively constructed graph families. *Algorithmica*, 7:555–582, 1992.
- [Bra01] A. Brandstädt. $(P_5, \text{Diamond})$ -Free Graphs Revisited: Structure, Bounded Clique Width and Linear Time Optimization. Manuscript, 2001.
- [CE91] L. Cai and J.A. Ellis. NP-completeness of edge-colouring some restricted graphs. *Discrete Applied Mathematics*, 30:15–27, 1991.
- [CER93] B. Courcelle, J. Engelfriet, and G. Rozenberg. Handle-rewriting hypergraph grammars. *Journal of Computer and System Sciences*, 46:218–270, 1993.
- [CH94] A. Cournier and M. Habib. A new linear time algorithm for modular decomposition. *Lecture Notes in Computer Science*, 787:68–84, 1994.
- [CHL⁺00] D.G. Corneil, M. Habib, J.M. Lanlignel, B. Reed, and U. Rotics. Polynomial time recognition of clique-width at most three graphs. In *Proceedings of Latin American Symposium on Theoretical Informatics (LATIN '2000)*, volume 1776 of *LNCS*. Springer-Verlag, 2000.
- [CLSB81] D.G. Corneil, H. Lerchs, and L. Stewart-Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3:163–174, 1981.

-
- [CM93] B. Courcelle and M. Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science*, 109:49–82, 1993.
- [CMR00] B. Courcelle, J.A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000.
- [CMR01] B. Courcelle, J.A. Makowsky, and U. Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second order logic. *Discrete Applied Mathematics*, 108(1-2):23–52, 2001.
- [CO00] B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101:77–114, 2000.
- [Cou90a] B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Methods and Semantics, pages 193–242. North-Holland, Amsterdam, 1990.
- [Cou90b] B. Courcelle. The monadic second-order logic of graphs I: Recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.
- [Cou92] B. Courcelle. The monadic second-order logic of graphs III: Tree-decompositions, minors and complexity issues. *Informatique Théorique et Applications*, 26:257–286, 1992.
- [Cou97] B. Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. In G. Rozenberg, editor, *Handbook of graph grammars and computing by graph transformations, Vol. 1 : Foundations*, chapter 5, pages 313–400. World Scientific, 1997.
- [Cou03] B. Courcelle. The monadic second-order logic of graphs XIV: Uniformly sparse graphs and edge set quantifications. *Theoretical Computer Science*, 299:1–36, 2003.
- [CPS84] D.G. Corneil, Y. Perl, and L.K. Stewart. Cographs: recognition, applications, and algorithms. *Congressus Numerantium*, 43:249–258, 1984.
- [CPS85] D.G. Corneil, Y. Perl, and L.K. Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14(4):926–934, 1985.
- [CR01] D.G. Corneil and U. Rotics. On the relationship between clique-width and tree-width. In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 2204 of *LNCS*, pages 78–90. Springer-Verlag, 2001.
- [EGW01a] W. Espelage, F. Gurski, and E. Wanke. Deciding clique-width for graphs of bounded tree-width. In *Proceedings of the Workshop on Algorithms and Data Structures*, volume 2125 of *LNCS*, pages 87–98. Springer-Verlag, 2001.
- [EGW01b] W. Espelage, F. Gurski, and E. Wanke. How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 2204 of *LNCS*, pages 117–128. Springer-Verlag, 2001.

-
- [EGW03] W. Espelage, F. Gurski, and E. Wanke. Deciding clique-width for graphs of bounded tree-width. *Journal of Graph Algorithms and Applications - Special Issue of JGAA on WADS 2001*, 7(2):141–180, 2003.
- [EIS76] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5:691–703, 1976.
- [Gal67] T. Gallai. Transitiv orientierbare Graphen. *Acta Math. Acad. Sci. Hung.*, 18:25–66, 1967.
- [GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, San Francisco, 1979.
- [GM03] A. Glikson and J.A. Makowsky. NCE graph grammars and clique-width. In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 2880 of *LNCS*, pages 237–248. Springer-Verlag, 2003.
- [Gö77] M. Götschel. Polyedrische Charakterisierungen kombinatorischer Optimierungsprobleme. *Mathematical Systems in Economics*, 36, 1977.
- [GR00] M.C. Golumbic and U. Rotics. On the clique-width of some perfect graph classes. *IJFCS: International Journal of Foundations of Computer Science*, 11(3):423–443, 2000.
- [GRT97] V. Giakoumakis, F. Roussel, and H. Thuillier. On P_4 -tidy graphs. *Discrete Mathematics and Theoretical Computer Science*, 1:17–41, 1997.
- [Gur98] F. Gurski. Algorithmische Charakterisierungen spezieller Graphklassen. Diplomarbeit, Heinrich-Heine-Universität, Düsseldorf, Germany, 1998.
- [GV97] V. Giakoumakis and J.-M. Vanherpe. On extended P_4 -reducible and extended P_4 -sparse graphs. *Theoretical Computer Science*, 180:269–286, 1997.
- [GW00] F. Gurski and E. Wanke. The tree-width of clique-width bounded graphs without $K_{n,n}$. In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 1938 of *LNCS*, pages 196–205. Springer-Verlag, 2000.
- [GW04] F. Gurski and E. Wanke. Vertex disjoint paths on clique-width bounded graph. In *Proceedings of Latin American Symposium on Theoretical Informatics (LATIN '2004)*, *LNCS*. Springer-Verlag, 2004. Appears.
- [Hab71] M. Habib. *Substitution de structures combinatoires, theorie et algorithmes*. PhD thesis, Universite Pierre et Marie Curie, Paris VI, 1971.
- [Har69] F. Harary. *Graph Theory*. Addison-Wesley Publishing Company, Massachusetts, 1969.
- [HM79] M. Habib and M.C. Maurer. On the X -join decomposition for undirected graphs. *Discrete Applied Mathematics*, 3:198–207, 1979.
- [Hoa85] C.T. Hoang. *Perfect Graphs*. Ph.D. thesis, Mc Gill University, Montreal, 1985.

-
- [ISG] <http://www.teo.informatik.uni-rostock.de/isgci/>.
- [JO89] B. Jamison and S. Olariu. P_4 reducible graphs a class of tree representable graphs. *Studies Appl. Math.*, 81:79–87, 1989.
- [JO92] B. Jamison and S. Olariu. A unique tree representation for P_4 sparse graphs. *Discrete Applied Mathematics*, 35:115–129, 1992.
- [Joh98] Ö. Johansson. Clique-decomposition, NLC-decomposition, and modular decomposition - relationships and results for random graphs. *Congressus Numerantium*, 132:39–60, 1998.
- [Joh00] Ö. Johansson. NLC₂-decomposition in polynomial time. *International Journal of Foundations of Computer Science*, 11(3):373–395, 2000.
- [Joh01] Ö. Johansson. log n -Approximative NLC_k-Decomposition in $O(n^{2k+1})$ Time. In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 2204 of *LNCS*, pages 229–240. Springer-Verlag, 2001.
- [KR01] D. Kobler and U. Rotics. Polynomial algorithms for partitioning problems on graphs with fixed clique-width. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 468–476. ACM-SIAM, 2001.
- [Lap98] D. Lapoire. Recognizability equals definability, for every set of graphs of bounded tree-width. In *Proceedings 15th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1373 of *LNCS*, pages 618–628. Springer-Verlag, 1998.
- [Ler71] H. Lerchs. On cliques and kernels. Technical report, Dept. of Comput. Sci, Univ. of Toronto, 1971.
- [LW93] T. Lengauer and E. Wanke. Efficient analysis of graph properties on context-free graph languages. *Journal of the ACM*, 40(2):368–393, 1993.
- [MP95] M. Middendorf and F. Pfeiffer. On the complexity of the disjoint paths problems. *Combinatorica*, 35(1):97–107, 1995.
- [MR99] J.A. Makowsky and U. Rotics. On the clique-width of graphs with few P_4 . *International Journal of Foundations of Computer Science*, 10:329–348, 1999.
- [MS94] R.M. McConnell and J. Sprinrad. Linear time modular decomposition and efficient transitive orientation of comparability graphs. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 536–545. ACM-SIAM, 1994.
- [NVZ01] T. Nishizeki, J. Vygen, and X. Zhou. The edge-disjoint paths problem is NP-complete for series-parallel graphs. *Discrete Applied Mathematics*, 115:177–186, 2001.
- [Pap94] C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, New York, 1994.
- [Ros74] D.J. Rose. On simple characterizations of k -trees. *Discrete Mathematics*, 7:317–322, 1974.

- [Roz97] G. Rozenberg. *Handbook of graph grammars and computing by graph transformations*. SIAM Monographs on Discrete Math. and Applications. World Scientific, New-Jersey, 1997.
- [RS86] N. Robertson and P.D. Seymour. Graph minors II. Algorithmic aspects of tree width. *Journal of Algorithms*, 7:309–322, 1986.
- [RS95] N. Robertson and P.D. Seymour. Graph minors XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.
- [Sch94] P. Scheffler. A practical linear time algorithm for vertex disjoint paths in graphs with bounded treewidth. Technical Report 396, Dept. of Mathematics, Technische Universität Berlin, 1994.
- [Sei92] J.J. Seidel. More about two-graphs, combinatorics, graphs and complexity. *Annals of Discrete Mathematics*, 51:297–308, 1992.
- [Sum71] P.D. Sumner. *Indecomposable Graphs*. PhD thesis, University of Massachusetts, Amherst, 1971.
- [Tod03] I. Todinca. Coloring powers of graphs of bounded clique-width. In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 2880 of *LNCS*, pages 370–382. Springer-Verlag, 2003.
- [Wan94a] E. Wanke. Bounded tree-width and LOGCFL. *Journal of Algorithms*, 16:470–491, 1994.
- [Wan94b] E. Wanke. The complexity of connectivity problems on context-free graph languages. *Journal of Computer and System Sciences*, 49(1):56–82, 1994.
- [Wan94c] E. Wanke. k -NLC graphs and polynomial algorithms. *Discr. Applied Mathematics*, 54:251–266, 1994. revised version: “<http://www.cs.uni-duesseldorf.de/~wanke>”.

Index

- (q, t) -Graph, 46
- 1.a-Weg, 74
- 1.b-Weg, 74
- 2.a-Weg, 76
- 2.b-Weg, 76
- 3.a-Weg, 79
- 3.b-Weg, 80
- $N_G(u)$, 7
- P_4 -reduzierbar Graphen, 41
- P_4 -sparse Graphen, 42
- P_4 -tidy Graphen, 45
- T_P , 71
- k -Ausdruck, 13, 17
 - äquivalent, 62
- k -Baum, 11
- k -markiert, 13
- k -regulär, 7
- l -sparse, 26
- l -terminaler Graph, 59
- 1-Weg, 74
- 4-Weg, 87

- adjazent, 7
- Algorithmus, 93

- Baum, 9
- Baumdekomposition, 10
- Baumweite, 10
- bipartite, 8
- Brückenknoten, 87
- bull, 36

- chair, 36
- chordal, 48
- Chromatic Index, 120
- Chromatic Number, 99
- Clique, 7
- Cliquenweite, 13
 - k -Ausdruck, 13
 - k -Ausdrucksbaum, 15

- co-chair, 36
- co-gem, 36
- Cobaum, 32
- Cograph, 32, 41
- CW_k , 13

- $\deg_G(v)$, 7
- Dekompositionsbaum, 11
- diamond, 36
- Distanz, 9
- distanzerhaltend, 43
- Durchmesser, 9

- effizient, 93
- Erkennungsproblem, 57
- Ersetzbarkeit, 59

- färbbar, 95
- Färbungszahl, 99
- forb $_P$, 78
- freie Knoten, 83
- freie Markierungen, 83

- gem, 36
- Gitter, 27
- Graph, 7
 - gerichtet, 10
 - prim, 32
- Grapheigenschaft, 59
- Graphklasse, 7

- Hamilton Kreis, 94
- Hamilton Weg, 111
- house, 36
- HR-System, 11
- Hyperkantenersetzungssystem, 11

- Intervallgraph, 48
- inzident, 7
- isomorph, 8

-
- Kanten, 7
 - Kanteneinfügeknoden, 15
 - Kanteneinfügeoperation, 14
 - Knoten, 7
 - knotendisjunkt, 112
 - Knotenersetzungssystem, 11
 - Komplementgraph, 8
 - Kreis, 9

 - lineare NLC-Weite, 28
 - Linearzeit, 93
 - Linegraph, 123

 - Matching, 108
 - maximal, 110
 - perfekt, 108
 - Minor, 8
 - Modul, 32
 - starkes, 32
 - triviales, 32
 - Modulare Dekomposition, 29
 - Multimenge, 98

 - Nachbarn, 7
 - Nachbarschaft, 7
 - NLC_k, 16
 - NLC-Weite, 16
 - k*-Ausdrucksbaum, 18
 - k*-Ausdruck, 17
 - Normalform, 62
 - NP-schwer, 94
 - NP-vollständig, 93
 - NR-System, 11

 - partieller *k*-Baum, 11
 - perfekte Eliminationsordnung, 123
 - Permutationsgraph, 48
 - planar, 26
 - polynomiell, 93
 - Polynomzeit, 93
 - prim, 32

 - Quotientgraph, 33

 - Rahmen, 74
 - regulär, 7

 - Sohn, 10
 - Spinne, 42

 - Splitgraph, 42
 - starker Verbindungsbaum, 67
 - starker Verbindungstyp, 68

 - Teilausdruck, 15, 19
 - Teilgraph, 8
 - induzierter, 8
 - Terminale, 59
 - TW_k, 10

 - Ummarkierungsknoten, 15, 19
 - Ummarkierungsoperation, 14, 17
 - unabhängig, 7
 - uniformly *l*-sparse, 26
 - unlab(*G*), 15

 - Vater, 10
 - Verbindungsbaum, 88
 - Verbindungstyp, 66, 88
 - Vereinigungsknoten, 15, 19
 - Vereinigungsoperation, 14, 17
 - vollständig, 7
 - vollständig bipartite, 8
 - vollständig unzusammenhängend, 7

 - Wald, 9
 - Weg, 8
 - Wurzel, 9, 10
 - Wurzelpfad, 25

 - zusammenhängend, 9
 - Zusammenhangskomponente, 9

Danksagung

Ich möchte an dieser Stelle Prof. Dr. Egon Wanke meinen besonderen Dank aussprechen. Seine kritische Betreuung hat zum Gelingen dieser Arbeit wesentlich beigetragen. Er stand mir in zahlreichen Diskussionen stets mit guten Anregungen zur Seite und er begleitete die Arbeit mit großem Interesse.

Mein Dank gilt weiterhin Prof. Dr. Jörg Rothe für seine Tätigkeit als Zweitgutachter. Ebenso danke ich meinen Kollegen Prof. Dr. Volker Aurich, Dr. Andreas Beck, Dr. Wolfgang Espelage und Prof. Dr. Jochen Reihmann für die gute Zusammenarbeit, die freundliche Atmosphäre und das gute Arbeitsklima.

Ferner danke ich der Deutschen Forschungsgemeinschaft für die finanzielle Unterstützung dieser Arbeit im Rahmen des DFG Projekts „Effiziente Algorithmen für Graphen mit beschränkter Cliquesweite“ (WA 674/9-2).