

**Monadische Erweiterungen von
Monadic NP**

Dissertation
zur Erlangung des Grades
Doktor der Naturwissenschaften

am Fachbereich Mathematik und Informatik
der Johannes Gutenberg-Universität in Mainz

vorgelegt von

Sebastian Poloczek

geboren in Hindenburg

Mainz, im August 2003

Inhaltsverzeichnis

1	Einleitung	1
1.1	Inhalt und Ergebnisse dieser Arbeit	3
1.1.1	Monadic NP + built-in Relationen	4
1.1.2	Abschlüsse monadischer Sprachen	5
2	Sätze, Definitionen und Notationen	7
2.1	Endliche Relationsstrukturen	7
2.2	Deskriptive Komplexitätstheorie	8
2.3	Built-in Relationen	10
2.4	Ehrenfeucht-Fraïsse Spiele	12
2.5	Monadische Erweiterungen, Präfixe und Spiele	19
2.6	Spieltypen	21
3	Verpackbare built-in Relationen	24
3.1	Relationenklassen	25
3.2	Verpackbare Relationen	27
3.3	Beweis von Satz 3.8	30
3.4	Anwendungen	38
3.4.1	Ordnungen	39
3.4.2	Äquivalenzrelationen	39
3.4.3	Relationen mit beschränkter Baumweite	39
3.5	Verpackbare Relationen bezüglich Padding	48
4	Unendlich teilbare built-in-Relationen	50
4.1	Unendlich teilbare built-in Relationen	50
4.2	Monadic NP + unendlich teilbare built-in Relationen	52
4.3	Anwendungen	56
4.3.1	x-built-in Relationen	57
4.3.2	Built-in Relationen mit verbotenen Minor K_k	60

4.3.3	Lokale Eigenschaften	69
4.4	Unendlich teilbare built-in Relationen bezüglich Padding	72
4.5	Zusammenfassung und Bemerkungen zu Kapitel 3 und 4	72
5	Closed Monadic NP	76
5.1	Der Beweis von Marcinkowski	77
5.2	Der Beweis von Ajtai-Fagin-Stockmeyer	80
6	Färbbarkeitsprobleme	92
6.1	Beweis von Satz 6.1 für $k = 4$	93
6.2	Der Fall $k > 4$	108
6.3	Der Fall $k = 3$	111
6.4	Zusammenfassung und Bemerkungen zu Kapitel 5 und 6	112
	Literaturverzeichnis	116

Kapitel 1

Einleitung

Die Komplexitätstheorie beschäftigt sich mit der *Komplexität* von algorithmischen Problemen. Es geht dabei um die Frage, welche Menge an Zeit- und/oder Platzressourcen notwendig sind, um ein gegebenes Problem auf einer Maschine (z.B. Turing Maschine) zu berechnen.

Eine besonders schwierige Problemstellung in diesem Bereich ist die Frage nach un-teren Zeit- und/oder Platzschranken.

Fagin hat in seiner bahnbrechenden Arbeit einen direkten Zusammenhang zwischen der wichtigen Komplexitätsklasse NP (die Menge der in nicht deterministischer poly-nomieller Zeit berechenbaren Probleme) und existentieller Second Order Logik fest-gestellt. In der Folgezeit konnten viele wichtige Komplexitätsklassen mittels logischer Sprachen beschrieben werden. Folgende Tabelle gibt einen knappen Überblick:

	Komplexitätsklasse	logische Sprache	Referenz
Immermann, Vardi Fagin	P	= FO + LFP + <	[15]
	NP	= \exists SO	[10]
	co-NP	= \forall SO	[10]
Stockmeyer	PH	= SO	[25]

Untere Zeit- und/oder Platzkomplexitätsschranken können daher auch durch den Nachweis von Ausdrucksschwäche bestimmter logischer Sprachen gefunden werden. Um z.B. zu zeigen, daß ein algorithmisches Problem nicht in NP liegt, genügt es zu zeigen, daß das Problem nicht in existentieller SO-Logik (auf endlichen Modellen) ausgedrückt werden kann.

Die anfängliche Hoffnung, auf diesem Wege *neue* untere Komplexitätsschranken aufzufinden, hat sich allerdings nicht erfüllt. Treffend stellte Fagin fest:

„*There is no free lunch*“.

Der Nachweis von Ausdrucksschwäche in voller existentieller SO-Logik *scheint extrem* schwer zu sein. *Vernünftiges* Arbeiten ist bisher nur auf einfachen Fragmenten von existentieller SO-Logik möglich. Ein solches einfaches Fragment erhält man, wenn man nur existentielle SO-Quantifizierung über einstellige Relationen zuläßt. Man bezeichnet diesen Teil von existentieller SO-Logik mit **monadic NP**. Monadic NP ist einerseits *deutlich* ausdruckschwächer als volle existentielle SO-Logik, andererseits liefert diese Tatsache die Möglichkeit *relativ einfache* Beweise für Ausdrucksschwäche zu führen. Wesentliche Voraussetzung ist dabei die Beschränkung auf monadische (einstellige) SO-Quantifizierung (schon auf binärem NP funktionieren alle hier verwendeten Methoden nicht mehr).

Fagin bezeichnete die Beschäftigung mit monadic NP, als

„...*training ground for attacking the problems in their full generality*“.

Wie unten angedeutet wird, hat er monadic NP damit ein wenig unterbewertet. Diese Arbeit beschäftigt sich mit der Ausdrucksstärke von monadischen Erweiterungen von monadic NP. Dabei werden die folgenden Arten von Erweiterungen betrachtet:

1. Monadic NP + built-in Relationen (Kapitel 3 und 4).
2. Abschlüsse von monadic NP:
 - a. PFO(monNP) = Positiver FO-Abschluß von monadic NP.
 - b. Closed monadic NP = Abschluß von monadic NP bzgl. FO-Quantifizierung und existentieller monadischer SO-Quantifizierung.

Erstaunlicherweise kann die Betrachtung dieser Erweiterungen von monadic NP zu *höchst* interessanten komplexitätstheoretischen Ergebnissen führen. Es gelten die folgenden Sätze:

Satz 1 (Lynch 82)

monadic NP + built-in Addition + polynomielles Padding = NP.

Polynomielles Padding bedeutet dabei eine Struktur Erweiterung um polynomiell viele, isolierte Knoten. Diese können im gewissen Sinne als *Speicherbits* interpretiert werden.

Satz 2 (Schweikardt, Schwentick)

- a) *zweite Stufe von closed monadic NP + zwei built-in Nachfolgerrelationen \supset monadic NP + built-in Addition.*

b) zweite Stufe von closed monadic NP + zwei built-in Nachfolgerrelationen + polynomielles Padding = NP.

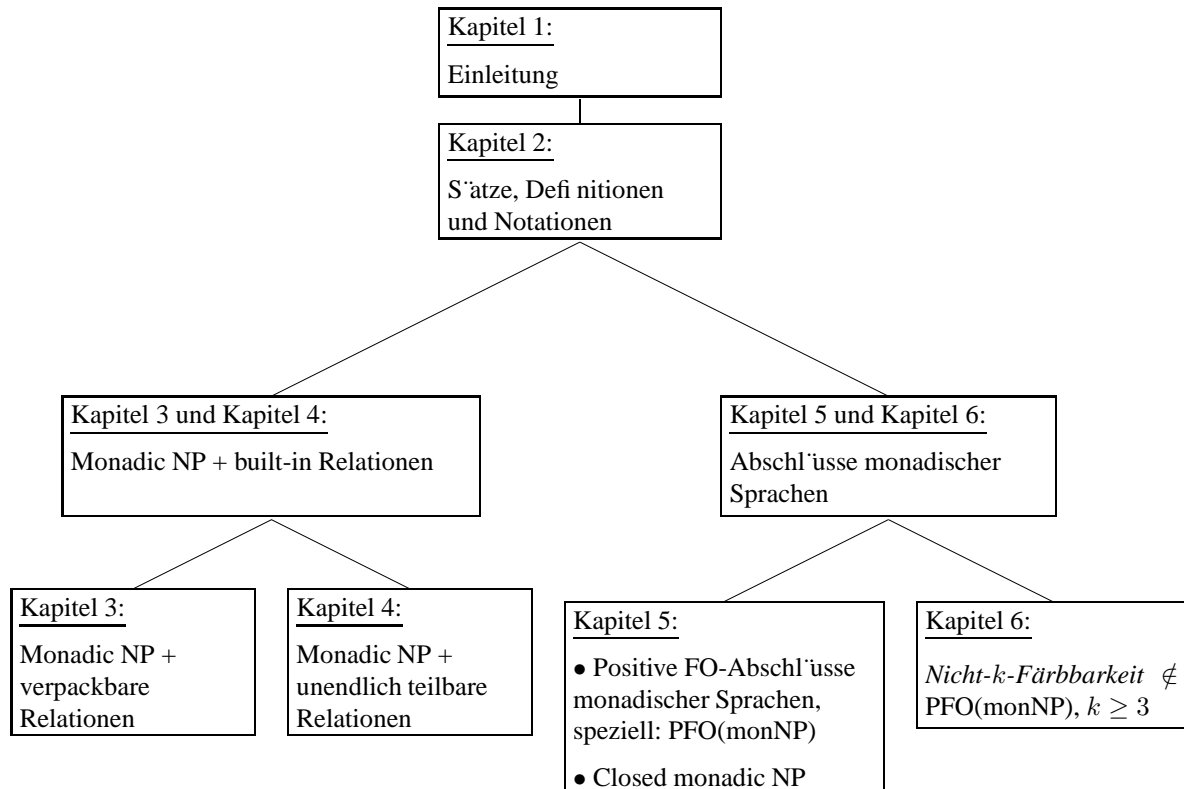
Monadic NP ist also nicht *bloß* eine Übungswiese, sondern kann im Zusammenhang mit bestimmten monadischen Erweiterungen durchaus zur Antwort der Frage NP $\stackrel{?}{\neq}$ co-NP führen.

1.1 Inhalt und Ergebnisse dieser Arbeit

Thema von Kapitel 3 und 4 ist monadic NP, erweitert um bestimmte Klassen von built-in Relationen. Eine built-in Relation kann dabei als eine *a priori* festgelegte *Strukturinformation* interpretiert werden (siehe Definition 2.7).

Kapitel 5 und 6 beschäftigen sich hauptsächlich mit dem Abschluß von monadic NP bzgl. FO-Quantifizierung (kurz: PFO(monNP)). Angeschnitten wird auch der Abschluß von monadic NP bzgl. FO- und existentieller monadischer SO-Quantifizierung (kurz: closed monadic NP) und der Abschluß einer beliebigen monadischen Sprache L bzgl. FO-Quantifizierung (kurz: PFO(L)).

Das folgende Schaubild beschreibt den Aufbau dieser Arbeit und gibt einen Überblick der vorkommenden Erweiterungen von monadic NP (kurz: monNP):



1.1.1 Monadic NP + built-in Relationen

Im Zusammenhang mit monadic NP können built-in Relationen in zwei Klassen aufgeteilt werden. Dazu die folgende Definition:

Definition 1

a) Eine built-in Relation \mathbf{B} heißt stark, falls

$$\text{monadic NP} + \mathbf{B} + \text{polynomielles Padding} = \text{NP}.$$

b) Eine built-in Relation heißt schwach, falls sie nicht stark ist.

Viele Arbeiten im Kontext von monadic NP + built-in Relationen beschäftigen sich mit der Grapheigenschaft *Zusammenhang von Graphen*. In den letzten drei Jahrzehnten wurde für eine Vielzahl von built-in Relationen \mathbf{B} gezeigt, daß

$$\text{Graphzusammenhang} \notin \text{monadic NP} + \mathbf{B}. \quad (*)$$

Dies hat zu der Vermutung geführt (Fagin, Stockmeyer, Vardi, siehe [13]), daß

$$\text{Graphzusammenhang} \notin \text{monadic NP} + \text{beliebige built-in Relationen}.$$

Ein Hauptergebnis des ersten Teils dieser Arbeit ist eine Klassifizierung aller built-in Relationen, von denen (bis 2002) (*) gezeigt werden konnte. Es ergibt sich eine (nicht disjunkte) Zerlegung in die beiden Klassen:

- verpackbare built-in Relationen (Kapitel 3) und
- unendlich teilbare built-in Relationen (Kapitel 4).

Mit diesen Konzepten kann (*) für neue Relationenklassen gezeigt werden. Dabei handelt es sich um:

- Äquivalenzrelationen
- x-built-in Relationen
- Relationen mit lokal beschränkter Baumweite
- Relationen mit lokal verbotenen vollständigen Minor
- Teilbarkeitsrelation
- Multiplikationsrelation

Desweiteren werden bekannte Ergebnisse mit neuen Methoden bewiesen. Die Beweise sind dadurch intuitiver, einfacher und kürzer geworden. Dabei wird (*) gezeigt für:

- Relationen mit beschränkter Baumweite (diese sind verpackbar) und
- Relationen mit verbotenen Minor K_k (diese sind unendlich teilbar).

Schließlich wird gezeigt, daß verpackbare und unendlich teilbare Relationen schwach im Sinne obiger Definition sind. Das bedeutet, daß alle built-in Relationen, für die (*) bisher gezeigt wurde, schwach sind.

(*) für eine starke built-in Relation zu zeigen, *scheint* sehr schwer zu sein. Daher ist es sinnvoll obige Vermutung abzuschwächen und zunächst die Frage zu untersuchen, ob

Graphzusammenhang \notin monadic NP + schwache built-in Relationen.

Dazu ist es wahrscheinlich notwendig eine *brauchbare* Klassifikation von schwachen built-in Relationen zu finden. Am Ende von Kapitel 4 wird die Frage aufgeworfen, ob eine Art Kombination der Eigenschaften verpackbar und unendlich teilbar eine solche Klassifikation liefern könnte.

1.1.2 Abschlüsse monadischer Sprachen

Wie oben erwähnt kann es lohnenswert sein, sich mit der zweiten Stufe von closed monadic NP zu beschäftigen. Erweitert man diese um zwei (*einfache*) Nachfolgerrelationen, so erhält man die volle Ausdrucksstärke von monadic NP + built-in Addition. Zusätzliches polynomielles Padding liefert sogar die volle Ausdrucksstärke von NP.

Im Kontext von closed monadic NP stellen sich insbesondere die folgenden zwei Fragen:

- 1) Ist die Hierarchie innerhalb von closed monadic NP strikt?
- 2) Gilt: closed monadic NP \neq MSO (monadische Second Order Logik)?

Bezogen auf monadische Eigenschaften (solche die in MSO ausdrückbar sind) *scheint* es jedoch schon auf der zweiten Stufe von closed monadic NP sehr schwer zu sein, Ausdrucksschwäche nachzuweisen.

Die bisherigen Ergebnisse auf diesem Gebiet beziehen sich daher *nur* auf PFO(monNP). Als erstes ist es Fagin et al. gelungen für eine monadische Eigenschaft zu zeigen, daß diese nicht in PFO(monNP) ausdrückbar ist (siehe [3]). In der Folgezeit konnte Marcinkowski nachweisen, daß die Eigenschaft *Erreichbarkeit in gerichteten Graphen* in PFO(monNP) nicht ausdrückbar ist (siehe [18]).

Im zweiten Teil dieser Arbeit werden die beiden Beweise gegenübergestellt. Es wird gezeigt, daß die zugrundeliegende Hauptidee in beiden Ansätzen identisch ist. Desweiteren wird der Beweis von Fagin et al. vereinfacht und verallgemeinert.

Das Hauptergebnis dieses Teils ist der Nachweis, daß

nicht-k-Färbbarkeit (kurz: $\text{non}k\text{C}$) \notin PFO(monNP), $k \geq 3$.

Aus der Annahme, daß $\text{NP} \neq \text{co-NP}$ folgt $\text{non}k\text{C} \notin$ closed monadic NP und da $\text{non}k\text{C}$ leicht in co-monadic NP beschrieben werden kann, folgt eine positive Antwort auf die zweite Frage.

Im Gegensatz zu den beiden obigen Eigenschaften (die beide in der zweiten Stufe von closed monadic NP beschreibbar sind) hat also die Beschäftigung mit $\text{non}k\text{C}$ das *Potential* eine positive Antwort auf die zweite Frage zu liefern.

Kapitel 2

Sätze, Definitionen und Notationen

2.1 Endliche Relationsstrukturen

Eine **Signatur** S ist eine nichtleere Menge $\{R_1, \dots, R_l, c_1, \dots, c_m\}$ von Relationensymbolen R_1, \dots, R_l mit endlicher Stelligkeit $\alpha(R_i), i = 1, \dots, l$ und Konstantensymbolen c_1, \dots, c_m . Eine **endliche Struktur** A über S ist ein Tupel $(\mathcal{U}^A, R_1^A, \dots, R_l^A, c_1^A, \dots, c_m^A)$, wobei \mathcal{U}^A , der **Grundraum** (auch **Universum** genannt), eine endliche, nichtleere Menge ist, $R_i^A \subseteq \times_{j=1}^{\alpha(R_i)} \mathcal{U}^A$ eine $\alpha(R_i)$ -stellige Relation ist und $c_i^A \in \mathcal{U}^A$ eine Konstante ist.

Soweit es aus dem Zusammenhang klar ist, wird in dieser Arbeit nicht zwischen Relationensymbolen und den zugehörigen Relationen unterschieden.

Beispiel 2.1

Eine spezielle, für diese Arbeit entscheidende, Signatur ist die zu Graphstrukturen gehörende Signatur $\{E\}$, wobei E eine zweistellige, symmetrische **Kantenrelation** ist. Graphstrukturen werden üblicherweise mit G bezeichnet, der Grundraum \mathcal{U}^G in diesem Falle oftmals mit $V(G)$ (V für Vertex Set).

Oftmals werden nicht reine Graphstrukturen, sondern um bestimmte Relationen und Konstanten erweiterte Graphstrukturen betrachtet. Das bedeutet Strukturen mit Signatur $S = \{E, R_1, \dots, R_l, c_1, \dots, c_m\}$, wobei R_1, \dots, R_l Relationen und c_1, \dots, c_m Konstanten sind. Auch dieser Typ von Strukturen wird in dieser Arbeit als Graphstruktur bezeichnet.

Bemerkung 2.2

Da in dieser Arbeit fast ausschließlich mit Graphstrukturen gearbeitet wird, beziehen sich die folgenden Betrachtungen auf Graphstrukturen. Die Verallgemeinerung auf beliebige Strukturen ist leicht durchzuführen.

Sei $G = (V(G), E, R_1, \dots, R_l, c_1, \dots, c_m)$ eine Graphstruktur mit zweistelligen Relationen R_1, \dots, R_l und Konstanten c_1, \dots, c_m . Zwei Knoten $x, y \in V(G)$ sind **R_i -benachbart**, wenn $R_i(x, y)$ (d.h. $(x, y) \in R_i$). Der **Grad** eines Knoten $x \in V(G)$

bzgl. R_i , kurz $\deg_{R_i}(x)$, ist die Anzahl der Knoten aus $V(G)$, die mit x R_i -benachbart sind. Oft schreibt man auch nur $\deg(x)$, falls R_i aus dem Zusammenhang klar ist.

Der **Grad von G** bzgl. R_i ist definiert durch

$$\deg_{R_i}(G) := \max_{x \in V(G)} (\deg_{R_i}(x)).$$

Der **Abstand** $\text{dist}_G(x, y)$ zweier Knoten $x, y \in V(G)$ ist die Länge eines kürzesten Weges von x nach y in G . Falls kein solcher Weg existiert, so ist der Abstand als ∞ definiert. Der Abstand $\text{dist}_G(x, U)$ eines Knoten $x \in V(G)$ zu einer Teilmenge $U \subset V(G)$ ist definiert durch

$$\text{dist}_G(x, U) := \min_{u \in U} (\text{dist}_G(x, u)).$$

Bezieht sich der Abstand nur auf eine bestimmte Relation R_i in G , so bezeichnet man das mit $\text{dist}_{R_i}(x, y)$.

Analog kann sowohl Abstand als auch Grad für eine Menge von Relationen (R_1, \dots, R_l) definiert werden. Dies wird mit $\deg_{(R_1, \dots, R_l)}(x)$ und $\text{dist}_{(R_1, \dots, R_l)}(x, y)$ bezeichnet.

Für $k \geq 0$ ist die k -Nachbarschaft $\mathcal{N}_k^G(x)$ eines Knoten $x \in V(G)$ als die folgende Unterstruktur von G definiert:

$$\mathcal{N}_k^G(x) := G \downarrow \{y \in V(G); \text{dist}_G(x, y) \leq k\}.$$

Dabei ist $G \downarrow U$ die Struktur mit Grundraum $U \subset V(G)$ und kanonisch reduzierten Relationen R_1^G, \dots, R_l^G .

Falls die zugrundeliegende Struktur G klar ist, wird auch oft die Bezeichnung \mathcal{N}_k verwendet. Manchmal ist es nötig die k -Nachbarschaft eines Knoten nicht bezüglich der *kanonischen* Distanzfunktion, sondern bezüglich einer bestimmten Distanzfunktion δ zu betrachten. Diese wird dann mit \mathcal{N}_k^δ bezeichnet.

2.2 Deskriptive Komplexitätstheorie

Als Fagin 1974 den folgenden Satz präsentierte, war das die Geburt der deskriptiven Komplexitätstheorie.

Satz 2.3 (Fagin 74)

Die Graphklasse \mathcal{P} ist genau dann in NP, wenn es eine Formel $\varphi \in \exists SO$ gibt, so daß für alle G gilt:

$$G \in \mathcal{P} \Leftrightarrow G \models \varphi.$$

Beweis: Siehe [10].

Anders geschrieben bedeutet obiger Satz $NP = \exists SO$, d.h. alle Graphklassen, die in NP liegen, können mittels eines existentiellen Second-Order Satzes beschrieben werden. In den folgenden Jahren konnten weitere Komplexitätsklassen mittels logischer Sprachen klassifiziert werden. Folgende Tabelle zeigt die Wichtigsten auf:

	Komplexitätsklasse	logische Sprache	Referenz
Immermann, Vardi	P	= FO + LFP + <	[15]
Fagin	NP	= $\exists SO$	[10]
	co-NP	= $\forall SO$	[10]
Stockmeyer	PH	= SO	[25]

Diese Klassifizierung eröffnet völlig neue Wege, um Komplexitätsklassen zu trennen. Um z.B. zu zeigen, daß $NP \neq \text{co-NP}$, würde es genügen für eine Graphklasse $\mathcal{P} \in NP$ zu zeigen, daß sie nicht in $\forall SO$ ausgedrückt werden kann. Es geht also nunmehr darum Ausdrucksschwäche von logischen Sprachen zu untersuchen.

Die in dieser Arbeit hauptsächlich verwendeten logischen Sprachen sind:

- First-Order Logic = FO
- monadic NP
- PFO(monadic NP)
- closed monadic NP
- Second-Order Logic = SO

Die zu FO und SO notwendigen Definitionen können in der Standardliteratur nachgelesen werden (z.B. [8]). Ebenso werden die Begriffe Formel, Satz, Quantorenrang hier nicht definiert. Die logischen Sprachen monadic NP, PFO(monadic NP) und closed monadic NP werden in den folgenden Definitionen erläutert.

Definition 2.4

- a) **Existentielle Second-Order Logik** ($\exists SO$) ist eine abgeschwächte Form von Second-Order Logik. Es sind nur existentielle SO-Quantoren erlaubt.

$$\exists SO := \{\exists R_1 \cdots \exists R_l \varphi; l \in \mathbb{N}, R_i \text{ Relation}, \varphi \in FO\}.$$

- b) **Monadic NP** (kurz *monNP*) ist eine Teilmenge von $\exists SO$. Es kann nur noch über einstellige Relationen quantifiziert werden.

$$\text{monadic NP} := \{\exists R_1 \cdots \exists R_l \varphi; l \in \mathbb{N}, R_i \text{ einstellige Relation}, \varphi \in FO\}.$$

- c) **PFO(monadic NP)**, kurz $PFO(monNP)$, ist der Abschluß von monadic NP bzgl. FO-Quantifizierung. Mittels regulärer Ausdrücke kann $PFO(monNP)$ wie folgt beschrieben werden:

$$PFO(monNP) := (\exists|\forall)^* (\exists^1)^* \varphi, \quad \varphi \in FO.$$

- d) **Closed monadic NP**, kurz $cmonNP$, ist der Abschluß von monNP bzgl. FO-Quantifizierung und existentieller monadischer SO-Quantifizierung. Mittels regulärer Ausdrücke kann $cmonNP$ wie folgt beschrieben werden:

$$cmonNP := (\exists|\forall|\exists^1)^* \varphi, \quad \varphi \in FO.$$

Bemerkung 2.5

Da im weiteren Verlauf fast ausschließlich monadische SO-Quantifizierungen verwendet werden, wird im Folgenden der Index 1 im Quantor \exists^1 weggelassen. Auf nicht monadische Quantifizierung wird jeweils explizit hingewiesen.

Ähnlich wie z.B. die polynomielle Hierarchie (PH) oder die monadische Hierarchie kann closed monadic NP formal in Stufen unterteilt werden. Diese Stufen unterscheiden sich in der Anzahl der geschachtelten SO-Quantorenblöcke. Folgende Definition präzisiert dieses.

Definition 2.6

- a) Die *i*-te Stufe von closed monadic NP (kurz i - $cmonNP$) besteht aus denjenigen Formeln aus $cmonNP$, deren Quantorenblock höchstens i (durch FO-Quantorenblöcke getrennte) existentielle SO-Quantorenblöcke enthält. Im Falle von genau i existentiellen SO-Quantorenblöcken beginnen die Formeln mit einem existentiellen SO-Quantorenblock.

Insbesondere entspricht monadic NP gerade der ersten Stufe von closed monadic NP.

- b) Als *i*-te Zwischenstufe von closed monadic NP bezeichnet man den Abschluß von i - $cmonNP$ bzgl. FO-Quantifizierung.

Ebenso wie in der polynomiellen Hierarchie ist die Frage nach der Striktheit dieser Hierarchie offen. Ein Zusammenfall von $cmonNP$ auf die zweite Stufe ist möglich.

Für die monadische Hierarchie konnten Matz, Schweikardt und Thomas die Striktheit nachweisen (siehe [20],[19]). Eine direkte Anwendung der dort verwendeten Methoden, für die Hierarchie in closed monadic NP, scheint nicht möglich zu sein.

2.3 Built-in Relationen

Bei der Beschäftigung mit der logischen Sprache monadic NP konnte im Laufe der Zeit von einigen Grapheigenschaften gezeigt werden, daß sie nicht ausdrückbar sind in monadic NP. Folgende Tabelle gibt einen knappen Überblick:

Eigenschaft	Referenz
Hamilton Kreis	[27]
Graphzusammenhang	[11]
Erreichbarkeit in gerichteten Graphen	[2]
nicht-3-Färbbarkeit	[5]

Viele dieser Eigenschaften sind in polynomieller Zeit entscheidbar, d.h. monadic NP ist in gewissem Sinne ausdruckschwach. Auf der anderen Seite beruhen die obigen Nichtausdruckschwachbeweise ganz wesentlich darauf, daß nur monadisch quantifiziert wird. Schon das Zulassen von binärer SO-Quantifizierung macht alle bisherigen Beweismethoden unmöglich.

Will man also monadic NP einerseits verstärken, andererseits bekannte Methoden in Nichtausdruckschwachbeweisen verwenden, so sollte man nicht auf die Eigenschaft *monadisch* verzichten. Eine Möglichkeit (diese wird in den Kapiteln 3 und 4 betrachtet) ist die Hinzunahme von built-in Relationen. Dazu folgende Definition:

Definition 2.7

Eine Folge $\mathbf{B} = (B_n)_{n \in \mathbb{N}}$ heißt **built-in Relation**, falls:

1. $B_i, i \in \mathbb{N}$, ist eine $\alpha(B_i)$ -stellige Relation.
2. $\alpha(B_i) = \alpha(B_j)$ für alle $i, j \in \mathbb{N}$, d.h. alle Relationen haben dieselbe Stelligkeit.
3. Der Grundraum, auf dem B_n definiert ist, hat Mächtigkeit n . Es reicht als Grundraum das Fragment $[n] := \{1, \dots, n\}$ der natürlichen Zahlen zu verwenden (Isomorphieabgeschlossenheit).

Bemerkung 2.8

1. Falls $\alpha(B_i) = 2$, so nennt man \mathbf{B} eine **binäre built-in Relation**. In dieser Arbeit werden hauptsächlich binäre built-in Relationen untersucht.
2. Wichtige Beispiele für built-in Relationen kann man der folgenden Tabelle entnehmen:

<i>succ</i>	$\text{succ}(x, y) \Leftrightarrow y = x + 1$	<i>Nachfolger-Relation</i>
<i><</i>	$<(x, y) \Leftrightarrow x < y$	<i>Ordnungs-Relation</i>
<i>add</i>	$\text{add}(x, y, z) \Leftrightarrow z = x + y$	<i>Additions-Relation</i>
<i>BIT</i>	$\text{BIT}(x, y) \Leftrightarrow \text{Bit } y \text{ in } \text{bin}(x) \text{ ist } 1$	<i>BIT-Relation</i>

Die Erweiterung einer logischen Sprache L um eine built-in Relation \mathbf{B} bedeutet das Hinzufügen von \mathbf{B} zur festgelegten Signatur S . Im Falle von Graphstrukturen betrachtet man also die Signatur $S=(E,\mathbf{B})$. Dabei ist die built-in Relation \mathbf{B} fixiert, hängt also nicht von der Graphstruktur G , sondern nur von deren Größe $|V_G|$ ab. Folgende Definition beschreibt die Ausdruckskraft von $\text{monNP} + \mathbf{B}$:

Definition 2.9

Eine Grapheigenschaft \mathcal{P} ist in $\text{monNP} + \mathbf{B}$ genau dann definierbar, wenn es eine monadic NP-Formel $\varphi^{\mathbf{B}}$ über der Signatur (E,\mathbf{B}) gibt, so daß für jeden Graph $G=(V(G),E(G))$ gilt:

$$(G, B_{|V(G)|}) \models \varphi^{\mathbf{B}} \Leftrightarrow G \in \mathcal{P}.$$

Bemerkung 2.10

- a) Built-in Relationen dürfen nicht mit existentieller SO-Quantifizierung über Relationen verwechselt werden. Die letzteren können beliebig interpretiert werden, wohingegen built-in Relationen a-priori festgelegt sind. Im Allgemeinen ist existentielle SO-Quantifizierung viel mächtiger, als das Hinzufügen einer z.B. zweistelligen built-in Relation.

Zum Beispiel läßt sich zeigen (siehe [7]), daß Graphzusammenhang ausdrückbar ist in $\exists\text{Succ FO}$. Dies ist das Fragment von binärem NP, in dem nur binäres Quantifizieren über Nachfolgerrelationen erlaubt ist.

- b) Im Allgemeinen wird eine logische Sprache durch Hinzunahme einer built-in Relation stärker. Z.B. läßt sich leicht zeigen, daß die Eigenschaft Parität des Universums in monadic NP nicht ausdrückbar ist, aber sehr wohl in $\text{monadic NP} + \text{Succ}$ ausdrückbar ist.
- c) Natürlich gibt es auch Unterschiede zwischen verschiedenen built-in Relationen. Z.B. ist $\text{monadic NP} + \text{eine Ordnungsrelation}$ ausdrucksstärker, als $\text{monadic NP} + \text{eine Nachfolgerrelation}$ (siehe dazu[21]).

2.4 Ehrenfeucht-Fraïsse Spiele

Die durch Fagin eingeleitete Klassifizierung von Komplexitätsklassen mittels logischer Sprachen eröffnet völlig neue Wege zur Trennung von Komplexitätsklassen. Gezeigt werden muß eine Ausdrucksschwäche von logischen Sprachen. Da alle diese Klassifizierungen nur auf endliche Modelle bezogen sind, muß diese Ausdrucksschwäche auf endlichen Modellen nachgewiesen werden. An dieser Stelle versagen die meisten Methoden der allgemeinen Modelltheorie, die zum Nachweis von Ausdrucksschwäche logischer Sprachen dienen (z.B. Kompaktheitssatz, Satz von Löwenheim-Skolem). Diese beruhen wesentlich auf der Möglichkeit unendliche Modelle in Betracht zu ziehen. Eine Methode, die auf endlichen Modellen verwendet werden kann, ist die der Ehrenfeucht-Fraïsse Spiele, kurz EF-Spiele.

Mittels E(hrenfeucht)-F(raïsse)-Spielen kann also Ausdrucksschwäche von logischen Sprachen auf endlichen Modellen bewiesen werden. Konkret bedeutet das zu zeigen, daß eine bestimmte Grapheigenschaft \mathcal{P} in einer bestimmten logischen Sprache L nicht ausdrückbar ist. Dabei hängen die Spielregeln des EF-Spiels von der Sprache L ab. Die Regeln zu den in dieser Arbeit hauptsächlich betrachteten logischen Sprachen werden im Folgenden dargestellt. Eine ausführliche Darstellung mit Beispielen und Beweisen der in diesem Zusammenhang auftretenden Sätze findet man z.B. in [15],[9].

Allgemein wird das EF-Spiel von den zwei Spielern **Spoiler** (bad guy) und **Duplikator** (good guy) bestritten, die entgegengesetzte Ziele haben. Als *Spielbretter* dienen zwei Strukturen $G_0 \in \mathcal{P}$ und $G_1 \notin \mathcal{P}$. Spoilers Ziel ist es, den Unterschied der beiden Strukturen G_0, G_1 nachzuweisen, wogegen Duplikator versucht diesen zu vertuschen.

Das FO-Spiel

Sei \mathcal{P} eine Grapheigenschaft. Die Spielregeln des FO-Spiels zur Eigenschaft \mathcal{P} lauten wie folgt:

0. Spoiler wählt eine Rundenzahl $r \in \mathbb{N}$.
1. Duplikator wählt eine Struktur $G_0 \in \mathcal{P}$.
2. Duplikator wählt eine Struktur $G_1 \notin \mathcal{P}$.
3. Für $i \leq r$ wählt (dies wird auch oft mit **spielt** bezeichnet) Spoiler einen beliebigen Knoten in einer der beiden Strukturen. Im Anschluß wählt Duplikator einen Knoten aus der anderen Struktur. Damit ist die i -te Spielrunde abgeschlossen.
Die in der i -ten Runde gewählten Knoten werden mit $a_i \in V(G_0)$ und $b_i \in V(G_1)$ bezeichnet.
4. Duplikator gewinnt das Spiel, falls nach der r -ten Runde

$$(G_0 \downarrow \{a_1, \dots, a_r\}, a_1, \dots, a_r) \cong (G_1 \downarrow \{b_1, \dots, b_r\}, b_1, \dots, b_r).$$

Andernfalls gewinnt Spoiler.

Insbesondere kann Duplikator leicht gewinnen, wenn die Strukturen G_0 und G_1 isomorph sind.

Bemerkung 2.11

Spoilers Ziel ist es also, im Laufe des Spieles, den Unterschied der beiden Strukturen G_0, G_1 zu zeigen. Dazu wird er versuchen seine Knoten so zu spielen, daß Duplikator keine isomorphe Antwort finden kann. Duplikator hingegen hat das gegenteilige Ziel, so zu spielen, daß er immer eine isomorphe Antwort auf Spoilers Spielzug finden kann.

Kann ein Spieler so spielen, daß er auf jeden Fall gewinnt, unabhängig davon, wie der andere Spieler sich verhält, so sagt man, daß dieser Spieler eine **Gewinnstrategie** hat. Die Bedeutung von EF-Spielen liegt in dem folgenden, auf Ehrenfeucht und Fraïsse zurückgehenden Satz.

Satz 2.12

Die Grapheigenschaft \mathcal{P} ist genau dann in FO definiierbar, wenn Spoiler eine Gewinnstrategie im FO EF-Spiel bzgl. \mathcal{P} hat.

Das folgende Korollar weist den Weg, der in den folgenden Kapiteln bestritten wird.

Korollar 2.13

Falls Duplikator eine Gewinnstrategie im FO EF-Spiel bzgl. \mathcal{P} hat, so ist \mathcal{P} nicht definiierbar in FO.

Bemerkung 2.14

1. Eine genaue Betrachtung des Beweises von Korollar 2.13 ergibt, daß man einzelne Spielrunden mit einzelnen FO-Quantoren identifizieren kann. Setzt man $r \in \mathbb{N}$ im Schritt 0 fest und spielt weiter mit diesem r , so nennt man dieses Spiel **r -Runden FO-Spiel**. Es gilt nun folgender Zusammenhang zwischen Quantorentiefe von FO-Formeln und Rundenzahl r im r -Runden FO-Spiel.

Satz 2.15

Falls Duplikator das r -Runden FO-Spiel bzgl. \mathcal{P} gewinnt, so ist \mathcal{P} nicht ausdrückbar mittels einer Formel $\varphi \in \text{FO}$ mit Quantorenrang $\leq r$.

2. Ziel der folgenden Kapitel wird es daher oft sein für beliebiges $r \in \mathbb{N}$ Strukturen $G_0 \in \mathcal{P}, G_1 \notin \mathcal{P}$ zu finden, so daß Duplikator das r -Runden Spiel auf G_0, G_1 gewinnt.

Das monadic NP Spiel

Nach obiger Bemerkung kann man eine Spielrunde im FO-Spiel gerade mit einem FO-Quantor einer Formel $\varphi \in \text{FO}$ in Bezug setzen. Insbesondere kann eine Graph-eigenschaft \mathcal{P} nicht mittels einer FO-Formel φ mit Quantorenrang r definiert werden, falls Duplikator eine Gewinnstrategie im r -Runden FO-Spiel bzgl. \mathcal{P} hat.

Die in monadic NP zusätzlich auftretenden monadischen SO-Quantoren drücken sich im monadic NP-Spiel durch zusätzliche sogenannte **Färbungsrunden** aus. Hier die Regeln des monadic NP-Spiels bzgl. \mathcal{P} :

0. Spoiler wählt Parameter $c, r \in \mathbb{N}$.
1. Duplikator wählt eine Struktur $G_0 \in \mathcal{P}$.
2. Duplikator wählt eine Struktur $G_1 \notin \mathcal{P}$.
3. Spoiler wählt c Mengen $C_1, \dots, C_c \subset V(G_0)$.
4. Duplikator wählt c Mengen D_1, \dots, D_c in $V(G_1)$.

Bemerkung: Die Schritte 3 und 4 bezeichnet man auch als **Färbungsrunde**, was damit begründet ist, daß jeder Knoten aus $V(G_0)$ (und $V(G_1)$) mit den Farben C_1, \dots, C_c (D_1, \dots, D_c) *gefärbt* wird.

5. Für $i \leq r$ wählt Spoiler einen beliebigen Knoten in einer der beiden Strukturen G_0, G_1 , danach wählt Duplikator einen beliebigen Knoten in der anderen Struktur.

Die in der i -ten Runde gewählten Knoten werden mit $a_i \in V(G_0)$ und $b_i \in V(G_1)$ bezeichnet.

6. Duplikator gewinnt das Spiel, falls

$$(G_0 \downarrow \{a_1, \dots, a_r\}, C_1, \dots, C_c, a_1, \dots, a_r) \cong (G_1 \downarrow \{b_1, \dots, b_r\}, D_1, \dots, D_c, b_1, \dots, b_r).$$

Andernfalls gewinnt Spoiler.

Dieses Spiel charakterisiert monadic NP in derselben Weise, wie das FO-Spiel FO-Logik charakterisiert.

Aufgrund der nicht vorkommenden Alternationen von Second Order All- und Existenzquantoren in monadic NP können die Spielregeln des monadic NP Spiels *zugunsten* des Duplikators verändert werden.

Dieses von Ajtai und Fagin ([2]) modifizierte Spiel heißt **Ajtai-Fagin Spiel** und wird in dieser Arbeit auch oft monNP-Spiel genannt.

Im *gewöhnlichen* monadic NP-Spiel ist es für Duplikator im allgemeinen viel schwerer zu gewinnen. Da eine Gewinnstrategie für Duplikator eine Ausdrucksschwäche für

monadic NP bedeutet, wird in dieser Arbeit ausschließlich das Ajtai-Fagin Spiel im Zusammenhang mit monadic NP verwendet.

Es folgen die Regeln des Ajtai-Fagin Spiels bzgl. \mathcal{P} :

0. Spoiler wählt Parameter $c, r \in \mathbb{N}$.
1. Duplikator wählt eine Struktur $G_0 \in \mathcal{P}$.
2. Spoiler wählt c Mengen $C_1, \dots, C_c \subset V(G_0)$.
3. Duplikator wählt eine Struktur $G_1 \notin \mathcal{P}$.
4. Duplikator wählt c Mengen $D_1, \dots, D_c \subset V(G_1)$.
5. Für $i \leq r$ wählt Spoiler einen beliebigen Knoten in einer der beiden Strukturen G_0, G_1 , danach wählt Duplikator einen beliebigen Knoten in der anderen Struktur.

Die in der i -ten Runde gewählten Knoten werden mit $a_i \in V(G_0)$ und $b_i \in V(G_1)$ bezeichnet.

6. Duplikator gewinnt das Spiel, falls

$$(G_0 \downarrow \{a_1, \dots, a_r\}, C_1, \dots, C_c, a_1, \dots, a_r) \cong (G_1 \downarrow \{b_1, \dots, b_r\}, D_1, \dots, D_c, b_1, \dots, b_r).$$

Andernfalls gewinnt Spoiler.

Im Unterschied zum gewöhnlichen monadic NP-Spiel muß sich Duplikator im Ajtai-Fagin Spiel erst nachdem Spoiler G_0 *gefärbt* hat, auf die Wahl von G_1 festlegen. Dies macht zwar, auf der einen Seite Duplikator nicht *mächtiger*, in dem Sinne, daß er Gewinnstrategien bzgl. einer Grapheigenschaft \mathcal{P} hätte, die er nicht im gewöhnlichen monadic NP Spiel hat, auf der anderen Seite erleichtert es aber das Auffinden von Gewinnstrategien im hohen Maße. Eine genauere Betrachtung zum Verhältnis dieser beiden Spiele findet man in [12].

Es gilt nun folgende zu Satz 2.12 analoge Aussage:

Satz 2.16 (Ajtai-Fagin)

Eine Grapheigenschaft \mathcal{P} ist genau dann in monadic NP definiert, wenn Spoiler eine Gewinnstrategie im Ajtai-Fagin-Spiel bzgl. \mathcal{P} hat.

Es folgt sofort folgendes Korollar:

Korollar 2.17

Falls Duplikator eine Gewinnstrategie im Ajtai-Fagin-Spiel bzgl. \mathcal{P} hat, so ist die Grapheigenschaft \mathcal{P} in monadic NP nicht definiert.

Das PFO(monNP)-Spiel

Nun wird das zur Logik PFO(monNP) gehörende Spiel vorgestellt. PFO(monNP)-Formeln unterscheiden sich von monadic NP-Formeln durch einen vorangestellten FO-Quantorenblock. Im PFO(monNP)-Spiel, dessen Regeln nun folgen, drückt sich das in vorangestellten FO-Spielrunden aus.

0. Spoiler wählt Parameter $r_1, c, r_2 \in \mathbb{N}$.
1. Duplikator wählt eine Struktur $G_0 \in \mathcal{P}$.
2. Duplikator wählt eine Struktur $G_1 \notin \mathcal{P}$.
3. Für $i \leq r_1$ wählt Spoiler einen beliebigen Knoten in einer der beiden Strukturen G_0, G_1 , danach wählt Duplikator einen beliebigen Knoten in der anderen Struktur.
Die in der i -ten Runde gewählten Knoten werden mit $a_i \in V(G_0)$ und $b_i \in V(G_1)$ bezeichnet.
4. Spoiler wählt c Mengen $C_1, \dots, C_c \subset V(G_0)$.
5. Duplikator wählt c Mengen D_1, \dots, D_c in $V(G_1)$.
6. Für $i \leq r_2$ wählt Spoiler einen beliebigen Knoten in einer der beiden Strukturen G_0, G_1 , danach wählt Duplikator einen beliebigen Knoten in der anderen Struktur.
Die in der i -ten Runde gewählten Knoten werden mit $\bar{a}_i \in V(G_0)$ und $\bar{b}_i \in V(G_1)$ bezeichnet.
7. Duplikator gewinnt das Spiel, falls

$$\begin{aligned} & (G_0 \downarrow \{a_1, \dots, a_{r_1}, \bar{a}_1, \dots, \bar{a}_{r_2}\}, C_1, \dots, C_c, a_1, \dots, a_{r_1}, \bar{a}_1, \dots, \bar{a}_{r_2}) \\ & \cong \\ & (G_1 \downarrow \{b_1, \dots, b_{r_1}, \bar{b}_1, \dots, \bar{b}_{r_2}\}, D_1, \dots, D_c, b_1, \dots, b_{r_1}, \bar{b}_1, \dots, \bar{b}_{r_2}). \end{aligned}$$

Andernfalls gewinnt Spoiler.

Auch hier gilt eine zu Satz 2.12 analoge Aussage:

Satz 2.18

Eine Grapheigenschaft \mathcal{P} ist genau dann in PFO(monNP) defi nierbar, falls Spoiler eine Gewinnstrategie im PFO(monNP)-Spiel bzgl. \mathcal{P} hat.

Korollar 2.19

Falls Duplikator eine Gewinnstrategie im PFO(monNP)-Spiel bzgl. \mathcal{P} hat, so ist die Grapheigenschaft \mathcal{P} in PFO(monNP) nicht defi nierbar.

Bemerkung 2.20

Eine Vereinfachung des PFO(monNP)-Spiels wie im Falle des monadic NP-Spiels ist nicht möglich. Diese Vereinfachung beruhte ganz wesentlich auf der Tatsache, daß monadic NP Formeln mit einem Block existentieller, monadischer SO-Quantoren beginnen. Im Spiel entspricht das der Wahl von c Farben in beiden Strukturen. Dabei läßt man zunächst Spoiler c Farben in G_0 wählen und kann sich im Anschluß (!) an diese Färbung auf eine Struktur G_1 festlegen. Da im PFO(monNP)-Spiel vor den Färbungsrunden noch FO-Runden gespielt werden, müssen beide Strukturen G_0, G_1 vor Beginn der Färbungsrunden feststehen.

Der Schritt 0 in den vorangegangenen Ehrenfeucht-Fraïsse Spielen kommt in der Standardliteratur so nicht vor. Stattdessen gibt es zu jeder Parameterwahl ein *spezielles* Spiel. Diese werden mit

- r -Runden FO-Spiel
- (c, r) -monNP Spiel
- (r_1, c, r_2) -PFO(monNP) Spiel

bezeichnet. Die Sätze 2.12-2.18 müssen mit diesen Bezeichnungen wie folgt umformuliert werden:

Satz 2.21

Eine Grapheigenschaft \mathcal{P} ist genau dann in FO (monNP, PFO(monNP)) definierbar, wenn es eine Parameterwahl $r \in \mathbb{N}$ ($(k, r) \in \mathbb{N}^2$, $(r_1, k, r_2) \in \mathbb{N}^3$) gibt, so daß Spoiler eine Gewinnstrategie im speziellen FO (monNP, PFO(monNP)) Spiel bzgl. dieser Parameterwahl und \mathcal{P} hat.

Analog die Umformulierung der Korollare:

Korollar 2.22

Sei \mathcal{P} eine Grapheigenschaft. Falls Duplikator für jede Parameterwahl $r \in \mathbb{N}$ ($(k, r) \in \mathbb{N}^2$, $(r_1, k, r_2) \in \mathbb{N}^3$) eine Gewinnstrategie im speziellen FO (monNP, PFO(monNP)) Spiel bzgl. dieser Parameterwahl und \mathcal{P} besitzt, so ist die Eigenschaft \mathcal{P} nicht ausdrückbar in FO (monNP, PFO(monNP)).

Auch spricht man oft vom r -Runden $((c, r)$ -monNP, (r_1, c, r_2) -PFO(monNP)) Spiel auf G_0, G_1 . In diesem Fall sind sowohl die Parameter als auch die Spielstrukturen G_0, G_1 festgelegt. Das Spiel wird bzgl. dieser Parameter und Spielstrukturen wie im allgemeinen Fall fortgesetzt.

Eine spezielle, in den Kapiteln 5 und 6 benötigte Art des PFO(monNP) Spiels wird mit $(\exists\forall)^{r_1}(c, r_2)$ -PFO(monNP) Spiel bezeichnet. Es handelt sich dabei um eine Form des $(2r_1, k, r_2)$ -PFO(monNP) Spieles, in dem in den ersten $2r_1$ Spielrunden Spoiler jeweils abwechselnd einen Knoten aus der Struktur G_0 und der Struktur G_1 auswählt.

Dies entspricht der oben angedeuteten Korrespondenz von FO-Spielrunden mit FO-Quantoren. Insbesondere entspricht ein Allquantor einer Spielrunde, in der Spoiler einen Knoten aus G_1 auswählen muß, hingegen ein Existenzquantor einer Spielrunde, in der Spoiler einen Knoten aus G_0 auswählt.

Folgendes Korollar ergibt sich direkt aus obigen Korollar für das (r_1, c, r_2) -PFO(monNP) Spiel:

Korollar 2.23

Gewinnt Duplikator für jede Parameterwahl $(r_1, c, r_2) \in \mathbb{N}^3$ das $(\exists\forall)^{r_1}(c, r_2)$ -PFO(monNP) Spiel bzgl. \mathcal{P} , so ist die Eigenschaft \mathcal{P} nicht in PFO(monNP) ausdrückbar.

2.5 Monadische Erweiterungen, Präfixe und Spiele

Definition 2.24

Sei L eine logische Sprache.

- a) L heißt **monadisch**, falls $L \subset SO$ und L nur monadische SO-Quantoren verwendet.
- b) Eine Erweiterung \bar{L} von L heißt **monadische Erweiterung**, falls \bar{L} eine monadische Sprache ist.

Insbesondere ist die Formelmenge einer monadischen Sprache immer eine Teilmenge von $MSO := (\exists|\forall|\exists|\forall)^* \varphi$, wobei $\varphi \in FO$.

Bemerkung 2.25

In dieser Arbeit wird die Ausdrucksschwäche von monadischen Erweiterungen von monadic NP untersucht.

Um Ausdrucksschwäche einer monadischen Sprache L mit EF-Spielen nachzuweisen, wird man zunächst wiederum bestimmte Parameter festlegen und bzgl. dieser Parameter das Spiel spielen.

Eine solche festgelegte Parameterwahl kann durch monadische Präfixe beschrieben werden. Dazu die folgende Definition.

Definition 2.26

- a) Ein **monadisches Präfix** W ist eine Teilmenge aus der Menge $(\exists^*|\forall^*|\exists^*|\forall^*)^*$.
- b) Das zu W **komplementäre monadische Präfix** \bar{W} entsteht, indem man für alle $w \in W$ folgende Transformation der Symbole durchführt:

$$\begin{array}{lcl} \exists^* & \rightarrow & \forall^* \\ \forall^* & \rightarrow & \exists^* \\ \exists^* & \rightarrow & \forall^* \\ \forall^* & \rightarrow & \exists^* \end{array}$$

Analog kann zu jedem Element $w \in W$ das komplementäre Element $\bar{w} \in \bar{W}$ definiert werden.

- c) Jedes monadische Präfix x W definiert eine monadische Sprache $\mathbf{L}(W)$. Die Formelmenge von $L(W)$ ist gegeben durch

$$L(W) = \{w\varphi; w \in W, \varphi \in FO, \text{Quantorentiefe von } \varphi = 0\}.$$

Analog definiert jedes Element $w \in W$ eines monadischen Präfixes eine monadische Sprache $\mathbf{L}(w)$ mit Formelmenge

$$L(w) = \{w\varphi; \varphi \in FO, \text{Quantorentiefe von } \varphi = 0\}.$$

- d) Mit $\oplus\mathbf{L}(W)$ wird der Boolesche Abschluß von $L(W)$ bezeichnet.
 e) Mit $\mathbf{PFO}(W)$ wird der Abschluß von $L(W)$ bzgl. FO-Quantifizierung bezeichnet.
 f) Sei W ein monadisches Präfix und $w \in W$. Die **Präfixdarstellung** von w ist eine Kodierung von w mittels folgender bijektiven Abbildung:

$$P : (\exists|\forall|\exists|\forall)^* \rightarrow (\{\exists, \forall, \exists, \forall\} \times \mathbb{N})^*.$$

Sei $w = w_1 \dots w_l \in W$ und (l_1, \dots, l_r) derart gewählt, daß $w_i = \dots = w_{l_{i+1}-1} \neq w_{l_{i+1}}$, dann ist

$$P((w_1, \dots, w_l)) := ((w_{l_1}, l_1), (w_{l_2}, l_2), \dots, (w_{l_r}, l_r)).$$

Die Präfixdarstellung beschreibt also ein $w \in W$ durch Angabe der Länge von Blöcken identischer Quantoren. Die Länge l eines SO-Quantorenblockes nennt man auch **Färbungszahl**.

Das w - und $\oplus w$ -Spiel

Sei W ein monadisches Präfix und $w \in W$. Jedem solchen *klein* w ($\oplus w$) läßt sich ein spezielles EF-Spiel zuordnen. Die Regeln dieses w -Spiels ($\oplus w$ -Spiels) werden im Folgenden beschrieben. Sie ergeben sich direkt aus den oben mehrfach erwähnten Korrespondenzen zwischen bestimmten Spielrunden und bestimmten Quantorentypen. Sei $w = (w_1, \dots, w_n)$ mit $w_i \in \{\exists, \forall, \exists, \forall\}$.

1. Duplikator wählt $G_0 \in \mathcal{P}$.
- 2a. Duplikator wählt $G_1 \notin \mathcal{P}$.
- (2b. Im Falle des $\oplus w$ -Spiels entscheidet Spoiler nun, ob der weitere Spielverlauf mit dem monadischen Präfix w oder dem monadischen Präfix \bar{w} fortgesetzt wird.)

3. Für $i \leq n$:

$w_i = \exists$: Spoiler wählt einen Knoten $a_i \in G_0$. Duplikator antwortet mit einem Knoten $b_i \in G_1$.

$w_i = \forall$: Spoiler wählt einen Knoten $b_i \in G_1$. Duplikator antwortet mit einem Knoten $a_i \in G_0$.

$w_i = \exists$: Spoiler färbt G_0 mit einer Farbe A_i . Duplikator antwortet mit einer Färbung B_i von G_1 .

$w_i = \forall$: Spoiler färbt G_1 mit einer Farbe B_i . Duplikator antwortet mit einer Färbung A_i von G_0 .

4. Seien $A_1, \dots, A_l, a_1, \dots, a_r$ und $B_1, \dots, B_l, b_1, \dots, b_r$ die im Laufe des Spieles auf G_0 und G_1 gewählten Farben und Knoten (die Indizes müssen hierbei geeignet unnummeriert werden). Duplikator gewinnt das w -Spiel ($\oplus w$ -Spiel), falls

$$\begin{aligned} (G_0 \downarrow \{a_1, \dots, a_r\}, A_1, \dots, A_l, a_1, \dots, a_r) \\ \cong \\ (G_1 \downarrow \{b_1, \dots, b_r\}, B_1, \dots, B_l, b_1, \dots, b_r). \end{aligned}$$

In völliger Analogie zu den obigen Spielen ergeben sich folgender Satz und Korollar.

Satz 2.27

Eine Grapheigenschaft \mathcal{P} ist genau dann in $L(W)$ ($\oplus L(W)$) ausdrückbar, wenn es ein $w \in W$ gibt, so daß Spoiler eine Gewinnstrategie im w -Spiel ($\oplus w$ -Spiel) bzgl. \mathcal{P} hat.

Korollar 2.28

Falls Duplikator für alle $w \in W$ eine Gewinnstrategie im w -Spiel ($\oplus w$ -Spiel) bzgl. der Grapheigenschaft \mathcal{P} hat, so ist diese nicht in $L(W)$ ($\oplus L(W)$) definierbar.

2.6 Spieltypen

Wir betrachten die durch

$$G \sim_I H \Leftrightarrow G \cong H$$

definierte Isomorphierelation \sim_I auf Graphstrukturen mit Signatur $S = \{E, R_1, \dots, R_l, c_1, \dots, c_m\}$. Es ist leicht zu zeigen, daß \sim_I eine Äquivalenzrelation ist. Ebenso sieht man leicht, daß die Menge der Äquivalenzklassen von \sim_I abhängig ist von der Größe des Universums $V(G)$. Die Anzahl der Äquivalenzklassen kann dabei beliebig groß werden.

Im vorherigen Abschnitt wurden EF-Spiele für bestimmte logische Sprachen L definiert. Durch

$$G \sim_L H :\Leftrightarrow \text{Duplikator gewinnt das } L \text{ EF-Spiel auf } G \text{ und } H$$

wird eine weitere Relation auf Graphstrukturen definiert. Diese ist reflexiv und transitiv, aber nicht notwendig symmetrisch. Für *vielen* logische Sprachen L ist \sim_L jedoch auch symmetrisch und damit eine Äquivalenzrelation. Insbesondere ist die zum r -Runden Spiel gehörende Spielrelation \sim_r eine Äquivalenzrelation. Diese hat die *angenehme* Eigenschaft, daß die Menge der Äquivalenzklassen nur von der Signatur S , nicht aber von der Größe des Universums abhängt. Insbesondere gibt es nur eine feste (endliche) Zahl von Äquivalenzklassen. Diese Äquivalenzklassen werden **Spieltypen** zum Spiel L genannt.

Bezeichnungen:

Für einige wichtige logische Sprachen L gibt es spezielle Bezeichnungen für die Relation \sim_L :

$$\begin{aligned} G \sim_r H & : \iff \text{Duplikator gewinnt das } r\text{-Runden Spiel auf } G, H \\ G \sim_{c,r} H & : \iff \text{Duplikator gewinnt das } (c, r)\text{-monNP Spiel auf } G, H \\ G \sim_w H & : \iff \text{Duplikator gewinnt das } w\text{-Spiel auf } G, H \end{aligned}$$

Für den wichtigen Fall des r -Runden Spiels werden Spieltypen auch als **r -Typen** bezeichnet. Der r -Typ einer Graphstruktur G wird auch mit τ_r^G bezeichnet. Im Folgenden wird eine induktive *Definition* von τ_r^G vorgestellt.

Definition 2.29

Seien $x_1, \dots, x_l \in V(G)$.

- $\tau_0^G(x_1, \dots, x_l)$ ist die S -Isomorphieklasse von $G \downarrow \{x_1, \dots, x_l\}$.
- $\tau_{r+1}^G(x_1, \dots, x_l) := \{\tau_r^G(x_1, \dots, x_l, x) ; x \in V(G)\}$.
- $\tau_r^G := \tau_r^G(\emptyset)$.

Folgendes Lemma zeigt, daß damit tatsächlich der Spieltyp des r -Runden Spiels definiert wird.

Lemma 2.30

Seien G_0, G_1 Graphstrukturen. Es gilt folgende Äquivalenz:

$$\tau_r^{G_0} = \tau_r^{G_1} \iff \text{Duplikator gewinnt das } r\text{-Runden Spiel auf } G_0, G_1.$$

Die Unabhängigkeit der Anzahl der Spieltypen von der Größe des Universums ist Aussage des folgenden Lemmas.

Lemma 2.31

Für jedes $r \in \mathbb{N}$ gibt es ein $\mathcal{N}_r \in \mathbb{N}$ derart, daß es nur \mathcal{N}_r verschiedene r -Typen gibt.

Beweise für die obigen Lemmata findet man zum Beispiel in [21].

Kapitel 3

Verpackbare built-in Relationen

Wie in der Einleitung beschrieben, kann man diese Arbeit grob in zwei Teile aufteilen. Dabei werden zwei unterschiedliche monadische Arten der Erweiterung von monadic NP betrachtet. In diesem und dem folgenden Kapitel wird monadic NP um bestimmte built-in Relationen erweitert. Diese werden jeweils auf Ausdrucksschwäche untersucht. Dazu wird gezeigt, daß sie monadic NP nicht ausreichend verstärken, um Graphzusammenhang auszudrücken.

Allgemein kann man die Menge der built-in Relationen mittels folgender Definition in die beiden Klassen **starke** und **schwache** built-in Relationen aufteilen (siehe dazu auch [24]).

Definition 3.1

a) Eine built-in Relation **B** auf Graphstrukturen heißt **stark**, falls

$$\text{monadic NP} + \mathbf{B} + \text{polynomielles Padding} = \text{NP}.$$

b) Eine built-in Relation **B** heißt **schwach**, wenn sie nicht stark ist.

Polynomielles Padding bedeutet dabei, daß die Graphstruktur um polynomiell viele, bzgl. der Kantenrelation E , isolierte Knoten erweitert wird. Diese zusätzlichen Knoten können gewissermaßen als Speicherbits verwendet werden. Eine formale Definition von polynomiellen Padding findet man in [24].

Bemerkung 3.2

Die Relationen *BIT* und *ADD* sind Beispiele für starke built-in Relationen.

Mit Methoden des folgenden Kapitels kann gezeigt werden, daß die Multiplikationsrelation *MULT* unendlich teilbar ist und daher eine schwache built-in Relation ist.

Was diese Aufteilung von built-in Relationen angeht, gibt es zum jetzigen Zeitpunkt eine klare Trennung hinsichtlich Spielbeweisen. Für eine sehr reichhaltige Zahl von schwachen built-in Relationen (die in dieser Arbeit erweitert wird) konnte mittels Ajtai-Fagin Spielen gezeigt werden, daß sie die Ausdrucksstärke von monadic

NP nicht ausreichend verstärken, um Graphzusammenhang auszudrücken (siehe z.B. [22],[23],[17]). Daher stellt sich die Frage, ob dies vielleicht sogar für alle schwachen built-in Relationen gilt.

Ganz im Gegensatz dazu war es bisher nicht möglich, einen derartigen Spielbeweis auf Graphstrukturen mit starken built-in Relationen zu führen. Es ist sogar noch viel gravierender, als diese Aussage vermuten läßt. Bisher ist es noch nicht gelungen, ein irgendwie geartetes Ajtai-Fagin Spiel auf nicht isomorphen Graphstrukturen zu gewinnen, wenn die Strukturen starke built-in Relationen enthalten.

Die Definition von starken built-in Relationen liefert auch ein plausibles Argument für die scheinbare Härte von Spielbeweisen auf Graphstrukturen mit starken built-in Relationen. Ein solcher Spielbeweis wäre der erste der folgenden zwei Schritte zum Beweis von $\text{co-NP} \neq \text{NP}$:

1. Schritt: Zeige, daß ein co-NP vollständiges Problem \mathcal{P} nicht ausdrückbar ist in monadic NP + starke built-in Relation.
2. Schritt: Zeige, daß auch polynomielles Padding nicht ausreicht, um das Problem \mathcal{P} zu beschreiben.

Fagin, Stockmeyer und Vardi haben in [13] folgende Vermutung geäußert:

Vermutung 3.3

Graphzusammenhang ist nicht ausdrückbar in monadic NP + beliebige built-in Relationen.

Aufgrund der oben geschilderten *deprimierenden* Situation mit starken built-in Relationen macht es vielleicht Sinn, das Blickfeld zunächst auf schwache built-in Relationen einzuschränken und zu versuchen, folgende, anhand bisheriger Erfahrungen mit schwachen built-in Relationen, naheliegende Vermutung zu beweisen.

Vermutung 3.4

Graphzusammenhang ist nicht ausdrückbar in monadic NP + beliebige schwache built-in Relation.

Dazu ist es vermutlich zunächst notwendig, eine *brauchbare* Klassifizierung der Menge der schwachen built-in Relationen zu finden.

Ob allerdings die bisherige Praxis (wie es auch im ersten Teil dieser Arbeit geschieht) für ganz bestimmte Klassen von schwachen built-in Relationen zu zeigen, daß sie in Bezug auf Graphzusammenhang keine Hilfe leisten, zum Beweis der zweiten Vermutung führen kann, erscheint mir eher fraglich.

3.1 Relationenklassen

In diesem und dem folgenden Kapitel werden zwei *neue* Klassen von built-in Relationen eingeführt. Dabei handelt es sich um:

- verpackbare built-in Relationen (Kapitel 3) und
- unendlich teilbare built-in Relationen (Kapitel 4).

Beide Klassen werden mittels *Strukturbedingungen* definiert.

Bemerkung 3.5

Eine weitere Struktureigenschaft von Relationen ist die von Kreidler eingeführte (k, j) -Zerlegbarkeit.

Definition:

Sei $\mathbf{B} = (B_n)_{n \in \mathbb{N}}$ eine built-in Relation. Sei v_n ein Knoten mit maximalem Grad in B_n und M_n die Menge aller Nachbarn von v_n .

- a) \mathbf{B} heißt (k, j) -**zerlegbar**, wenn es für alle $l \in \mathbb{N}$ ein $n \in \mathbb{N}$ gibt, so daß eine Menge $I_n \subset M_n$ existiert mit $|I_n| \geq l$ und eine Menge S_n existiert mit $|S_n| \leq k$, $v_n \in S_n$, $I_n \cap S_n = \emptyset$, derart daß für alle $x \neq y \in I_n$ gilt:

$$\mathcal{N}_j^{B_n - S_n}(x) \cap \mathcal{N}_j^{B_n - S_n}(y) = \emptyset.$$

- b) \mathbf{B} heißt **zerlegbar**, wenn für alle $j \in \mathbb{N}$ ein $k(j) \in \mathbb{N}$ existiert, so daß \mathbf{B} $(k(j), j)$ -zerlegbar ist.

Kreidler hat in [17] gezeigt, daß Graphzusammenhang nicht in $\text{monNP} + \text{zerlegbare Relation ausdrückbar}$ ist. Außerdem konnte er von folgenden Relationenklassen zeigen, daß sie zerlegbar sind:

- Wälder
- Planare Graphen
- Graphen mit beschränkter Baumweite
- Graphen mit verbotenen Minor K_k .

Im folgenden Kapitel wird gezeigt, daß zerlegbare Relationen unendlich teilbar sind. Die Klasse der zerlegbaren built-in Relationen ist also eine Untermenge der unendlich teilbaren Relationen.

Bemerkung 3.6

Die Vereinigung von verpackbaren und unendlich teilbaren built-in Relationen enthält zahlreiche (schwache) built-in Relationen. Beispiele für verpackbare built-in Relationen sind:

- Ordnung
- Äquivalenzrelationen

- Relationen mit beschränkter Baumweite

Beispiele für unendlich teilbare built-in Relationen sind:

- maximaler Grad $n^{o(1)}$
- x -built-in Relationen (siehe Kapitel 4)
- Relationen mit lokal beschränkter Baumweite
- Relationen mit lokal verbotenen Minor K_k
- Teilbarkeitsrelation
- Multiplikationsrelation

Insbesondere umfaßt die Vereinigung von verpackbaren und unendlich teilbaren Relationen alle built-in Relationen, von denen bisher Ausdrucksschwäche im Bezug auf monadic NP und Graphzusammenhang nachgewiesen wurde. Dazu folgende Definition

Definition

Eine built-in Relation \mathbf{B} für welche die Aussage

$$\text{Graphzusammenhang} \notin \text{monadic NP} + \mathbf{B}$$

bewiesen ist (bis 2002), heißt **sehr schwach**.

Die Vereinigung von verpackbaren und unendlich teilbaren Relationen enthält alle sehr schwachen built-in Relationen. Es stellt sich daher die folgende **Frage**:

Gibt es schwache built-in Relationen, die weder verpackbar, noch unendlich teilbar sind, noch eine Kombination dieser beiden Struktureigenschaften besitzen?

Was dabei eine Kombination dieser beiden Struktureigenschaften bedeutet, kann erst in der Zusammenfassung von Kapitel 3 und 4, im Anschluß an Kapitel 4, erläutert werden.

3.2 Verpackbare Relationen

Definition 3.7

Sei $\mathbf{B} = (B_n)_{n \in \mathbb{N}}$ eine built-in Relation.

a) Eine Teilmenge $P \subset [N]$ ($N \in \mathbb{N}$) heißt **Paket**, wenn

$$\forall x, \bar{x} \in P, \forall y \in P^c : B_N xy \iff B_N \bar{x}y \text{ und } B_N yx \iff B_N y\bar{x},$$

d.h. alle Knoten aus P haben bzgl. der Außenwelt P^c dieselben Kanteneigenschaften.

b) Eine Teilmenge $P \subset [N]$ heißt **l -Paket**, wenn eine Ausnahmemenge A mit $|A| \leq l$ existiert, so daß

$$\forall x, \bar{x} \in P \setminus A, \forall y \in P^c : B_N xy \iff B_N \bar{x}y \text{ und } B_N yx \iff B_N y\bar{x},$$

d.h. bis auf die maximal l Ausnahmeknoten aus A haben alle Knoten aus P bzgl. der Außenwelt dieselben Kanteneigenschaften.

c) **B heißt verpackbar**, falls ein $g, l \in \mathbb{N}$ existiert, so daß für alle $L, n \in \mathbb{N}$ ein $N_0 \in \mathbb{N}$ existiert derart, daß für alle $N > N_0$ auf $[N]$ L disjunkte l -Pakete P_i existieren mit

$$\frac{n}{g} \leq |P_i| \leq n \quad \forall i = 1 \dots L.$$

Grobgesprochen bedeutet das, daß man in einer verpackbaren built-in Relation beliebig viele disjunkte l -Pakete finden kann.

Im Laufe dieses Kapitels werden folgende Sätze bewiesen:

Satz 3.8

Graphzusammenhang ist nicht ausdrückbar in monadic NP + **eine** verpackbare built-in Relation.

Satz 3.9

a) Äquivalenzrelationen mit $\limsup(\maxgrad(B_n)) = \infty$ sind verpackbar. Dabei ist $\maxgrad(B_n)$ der maximale Grad von B_n .

b) Relationen mit beschränkter Baumweite sind verpackbar.

Satz 3.10

Verpackbare Relationen sind schwach im Sinne von Definition 3.1.

Zuerst wird Satz 3.8 bewiesen. Dazu wird die im Folgenden beschriebene Eigenschaft von verpackbaren Relationen benötigt.

Definition 3.11

Sei B eine Relation.

a) Eine Menge $\mathbf{P} = \{P_1, \dots, P_k\}$, deren Elemente P_i disjunkte l -Pakete sind, heißt **l -Paketmenge**.

Die Größe einer l -Paketmenge ist $\|\mathbf{P}\| = |\cup_{i=1}^k P_i| = \sum_{i=1}^k |P_i|$.

b) Seien $\mathbf{P} = \{P_1, \dots, P_m\}$ und $\mathbf{Q} = \{Q_1, \dots, Q_n\}$ l -Paketmengen. P und Q sind **paketisomorph**, wenn:

- 1) Es existiert ein B -Isomorphismus $f : \cup_{i=1}^m P_i \rightarrow \cup_{i=1}^n Q_i$.
- 2) $f|_{P_i}$ ist ein B -Isomorphismus von P_i nach Q_i .
- 3) $f|_{P_i}$ bildet die Ausnahmeknoten von P_i auf die Ausnahmeknoten von Q_i ab.

Inbesondere ist $m=n$.

Der folgende Satz besagt, daß man in verpackbaren built-in Relationen nicht nur beliebig viele l -Pakete (und damit auch l -Paketmengen) finden kann, sondern auch *beliebig* viele paarweise paketisomorphe l -Paketmengen *beliebiger* Größe.

Wesentlich für den Beweis ist dabei die Tatsache, daß es nur endlich viele Paketisomorphietypen für l -Paketmengen der Größe $\leq K$ gibt.

Satz 3.12

Sei \mathbf{B} eine verpackbare built-in Relation. Es existiert ein $g, l \in \mathbb{N}$ so, daß für alle $K, L, n \in \mathbb{N}$ ein $N_0 \in \mathbb{N}$ existiert derart, daß für alle $N > N_0$ auf $[N]$ L disjunkte, paarweise paketisomorphe l -Paketmengen $\mathbf{P}_i = (P_{i1}, \dots, P_{im})$ existieren mit

$$K \leq \|\mathbf{P}_i\| \leq K + n \quad \forall i = 1, \dots, L \text{ und}$$

$$\frac{n}{g} \leq |P_{ij}| \leq n \quad \forall i = 1, \dots, L, j = 1, \dots, m.$$

Beweis:

Seien $K, L, n \in \mathbb{N}$ fest. Da \mathbf{B} verpackbar ist, existieren feste $g, l \in \mathbb{N}$, welche die Verpackbarkeitsbedingung aus Definition 3.11 erfüllen. Die folgenden Punkte liefern den Beweis des Satzes:

- Es gibt nur endlich viele Paketisomorphietypen von Paketmengen der Größe $\leq K + n$. Sei E die Anzahl dieser Typen. Dies kann direkt durch *Abzählen* gezeigt werden.
- Da \mathbf{B} verpackbar ist, existieren $g, l \in \mathbb{N}$ derart, daß für $\bar{L} = L \cdot E$ ein $N_0 \in \mathbb{N}$ existiert so, daß für alle $N \geq N_0$ auf $[N]$ \bar{L} disjunkte l -Paketmengen \mathbf{P}_i der Größe $K \leq \|\mathbf{P}_i\| \leq K + n$ existieren. Da es nur maximal E verschiedene Paketisomorphietypen auf l -Paketmengen dieser Größe gibt, müssen mindestens L davon paarweise paketisomorph sein. \square

Nach diesen Vorbereitungen kann nun im folgenden Abschnitt der Beweis von Satz 3.8 geführt werden.

3.3 Beweis von Satz 3.8

Der Beweis von Satz 3.8 basiert auf Ideen von Schwentick, mit denen er bewiesen hat, daß Graphzusammenhang nicht ausdrückbar ist in monadic NP, auch bei Hinzunahme einer built-in Ordnung (siehe [23]). Insbesondere ist Ordnung eine verpackbare built-in Relation.

Wesentliches Hilfsmittel ist der starke Erweiterungssatz von Schwentick, der Bedingungen liefert unter denen Duplikator das FO-Spiel auf zwei (gefärbten) Strukturen A, B gewinnen kann. Dieser Satz wird hier leicht verändert dargestellt, so daß er den weiter unten stehenden Gegebenheiten genügen wird.

Satz 3.13 (starker Erweiterungssatz von Schwentick)

Sei $r > 0$ und $S = (E, B, C_1, \dots, C_c)$ eine Signatur mit zweistelligen Relationen E, B und einstelligen Relationen C_1, \dots, C_c . Dabei sei E eine symmetrische Relation.

Seien A, \bar{A} endliche S -Strukturen und $\delta, \bar{\delta}$ Distanzfunktionen auf A, \bar{A} .

Seien H_1, \dots, H_l und $\bar{H}_1, \dots, \bar{H}_l$ Teilmengen von \mathfrak{U}^A bzw. $\mathfrak{U}^{\bar{A}}$ derart, daß

$$\mathcal{N}_{2^r}^\delta(H_i) \cap \mathcal{N}_{2^r}^\delta(H_j) = \emptyset \text{ und } \mathcal{N}_{2^r}^{\bar{\delta}}(\bar{H}_i) \cap \mathcal{N}_{2^r}^{\bar{\delta}}(\bar{H}_j) = \emptyset, \quad i \neq j.$$

Sei $\delta_j(x) := \delta(x, H_j)$ und $\bar{\delta}_j(x) := \bar{\delta}(x, \bar{H}_j)$. Duplikator hat eine Gewinnstrategie im r -Runden FO-Spiel auf A, \bar{A} , falls folgende Bedingungen gelten:

- (i) Für alle $j \leq l$ hat Duplikator eine Gewinnstrategie im r -Runden FO-Spiel auf $(A \downarrow \mathcal{N}_{2^k}^\delta(H_j), \delta_j)$ und $(\bar{A} \downarrow \mathcal{N}_{2^k}^{\bar{\delta}}(\bar{H}_j), \bar{\delta}_j)$.
- (ii) Es gibt einen S -Isomorphismus α von $A \downarrow (\mathfrak{U}^A \setminus (H_1 \cup \dots \cup H_l))$ nach $\bar{A} \downarrow (\mathfrak{U}^{\bar{A}} \setminus (\bar{H}_1 \cup \dots \cup \bar{H}_l))$ derart, daß für alle $x \in \mathfrak{U}^A \setminus (H_1 \cup \dots \cup H_l)$ und alle $j \leq l$ gilt:

$$\delta_j(x) = \bar{\delta}_j(\alpha(x)) \quad \text{oder} \quad (\delta_j(x) > 2^r \text{ und } \bar{\delta}_j(\alpha(x)) > 2^r).$$

- (iii) Für alle $j \leq l$ und alle $x \in A, \bar{x} \in \bar{A}$ mit

- $\delta_j(x) = \bar{\delta}_j(\bar{x}) \leq 2^r$
- x, \bar{x} sind bzgl. C_1, \dots, C_c identisch gefärbt,

und alle $y \in A$ mit $\delta_j(y) > \delta_j(x) + 1$ gilt:

$$Bxy \iff B\bar{x}\alpha(y), \quad Byx \iff B\alpha(y)\bar{x} \quad \text{und}$$

$$Exy \iff E\bar{x}\alpha(y).$$

Beweis: siehe [23]

Weiterhin wird eine kombinatorische Eigenschaft von Permutationsgruppen benötigt. Dazu zunächst eine kurze Einleitung:

S_n sei die symmetrische Gruppe auf $\{1, \dots, n\}$. Ein k -Zyklus wird durch $(i_1, \dots, i_k), i_j \neq i_h$ für $j \neq h$ bezeichnet und ist eine Permutation, die i_1 auf i_2, i_2 auf i_3, \dots, i_k auf i_1 abbildet. Jede Permutation läßt sich als Produkt von elementfremden Zyklen schreiben. $g \in S_n$ heißt n -**Zyklus**, falls $g = (i_1, \dots, i_n)$.

Sei $H \leq S_n$ eine Untergruppe von S_n und g ein n -Zyklus. H heißt **zykluserhaltend** bzgl. g und n , falls für alle $h \in H$ das Produkt gh wiederum ein n -Zyklus ist. Folgender Satz von Coppersmith wird im Folgenden eine geeignete, nicht zykluserhaltende Untergruppe H garantieren.

Satz 3.14 (Coppersmith 94)

Sei n groß genug. Sei g ein n -Zyklus und $H \leq S_n$ eine zykluserhaltende Untergruppe bzgl. g und n . Dann gilt:

$$|H| \leq n! \left(\frac{6}{\log n} \right)^n.$$

Beweis: siehe [4].

Nun sind die Vorbereitungen abgeschlossen. Im Anschluß an die folgende Bemerkung folgt der Beweis von Satz 3.8.

Bemerkung 3.15

Schwentick hat in [23] gezeigt, daß Graphzusammenhang nicht ausdrückbar ist in monadic NP + eine lineare Ordnung. Eine genaue Betrachtung des Beweises liefert die wesentliche Eigenschaft der Ordnungsrelation, die im Beweis benötigt wird. Diese ist, daß alle Knoten in einem Intervall $I = \{o, o + 1, \dots, o + n\}, o \in \mathbb{N}$ sich bezüglich aller Knoten außerhalb des Intervalles I identisch verhalten. Diese Beobachtung war der Ausgangspunkt, welcher zur Einführung des Begriffes verpackbar geführt hat.

Der nachfolgende Beweis orientiert sich daher stark an Schwenticks Methoden. Die wesentliche neue Beweisschwierigkeit liegt in der Art und Weise, wie die built-in Relation B_n in die Graphstruktur eingebettet wird.

Beweis von Satz 3.8:

Es muß gezeigt werden, daß für alle $c, r \in \mathbb{N}$ Duplikator eine Gewinnstrategie im (c, r) -monNP Spiel bzgl. der Menge der zusammenhängenden Graphen mit verpackbarer built-in Relation \mathbf{B} hat. Seien $c, r \in \mathbb{N}$ fest.

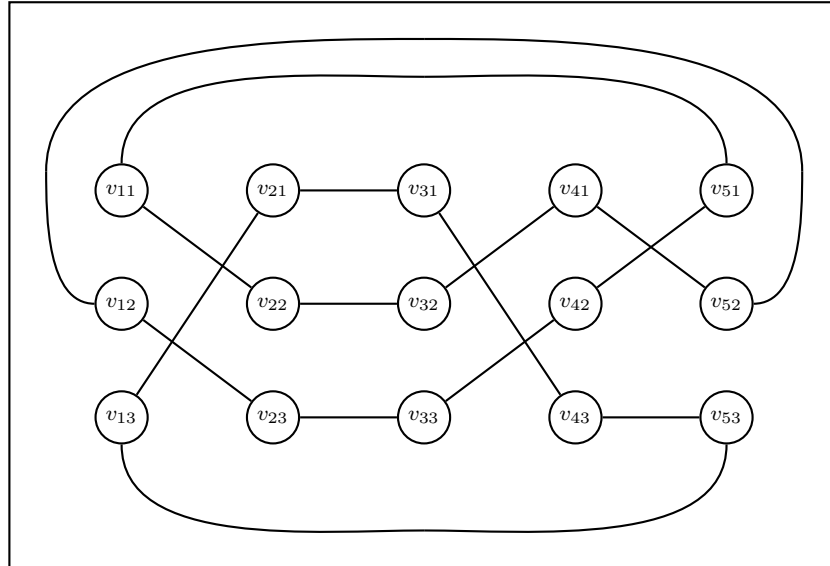
Zunächst wird der Spielgraph G_0 konstruiert.

Für $n > 0$ und eine Folge $\Pi = \pi_1, \dots, \pi_l$ von Permutationen auf $\{1, \dots, n\}$ wird wie folgt ein Graph G_Π definiert:

Die Knoten von G_Π seien $v_{ij}, i = 1, \dots, l + 1, j = 1, \dots, n$. Es gibt eine Kante von v_{ij} nach $v_{\hat{i}\hat{j}}$, falls:

- $\hat{i} = i + 1$ und $\hat{j} = \pi_i(j)$ oder
- $i = 1, \hat{i} = l + 1$ und $j = \hat{j}$.

Folgende Skizze zeigt G_{Π} für $n = 3$ und $\Pi = \pi_1, \pi_2, \pi_3, \pi_4$ mit $\pi_1 = (123)$, $\pi_2 = \text{id}$, $\pi_3 = (132)$ und $\pi_4 = (12)$:



Es gilt nun folgendes, entscheidendes Lemma:

Lemma 3.16

G_{Π} ist zusammenhängend $\iff \prod_{i=1}^l \pi_i$ ist ein n -Zyklus.

Auf den *leicht* zu führenden Beweis dieses Lemmas wird hier verzichtet.

Der weitere Beweisablauf orientiert sich an folgenden Ideen:

- Für hinreichend großes n wählt Duplikator $G_0 = G_{\Pi}$. Dabei ist Π eine geeignete Permutationsfolge, die aus vielen identischen Teilfolgen bestehen wird.
- Unabhängig davon wie Spoiler G_0 färbt, existieren in Π viele Permutationen, die durch andere Permutationen ersetzt werden können ($\Pi \rightarrow \bar{\Pi}$), so daß Spoiler den Unterschied zwischen G_{Π} und $G_{\bar{\Pi}}$ (mit identischer Färbung) in r Runden nicht nachweisen kann.
- Es gibt eine Kombination derartiger Ersetzungen, so daß $\bar{\Pi}$ keinen n -Zyklus liefert.

Schritt 1: Konstruktion des Spielgraphen G_0 .

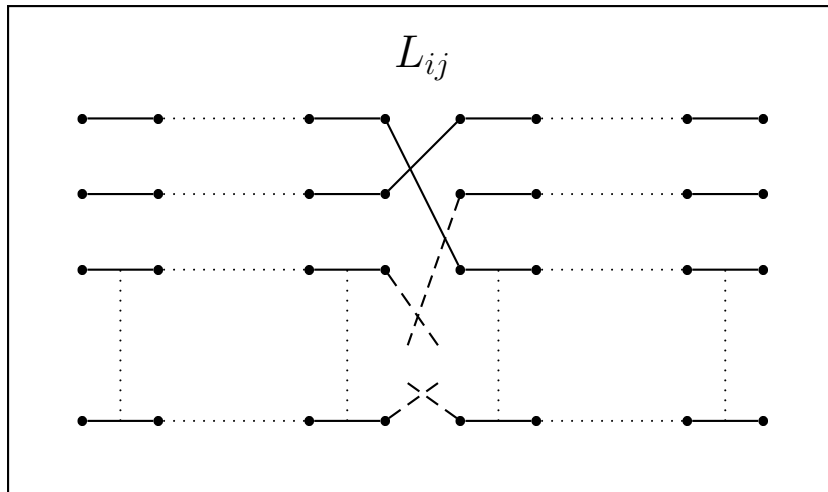
Sei $\sigma_1, \dots, \sigma_{n!}$ eine Aufzählung von S_n und sei

$$\varphi := \left(\prod_{i=1}^{n!} \sigma_i \right)^{-1}.$$

Seien $Q_i = \text{id}^{2^{r+1}}, \sigma_i, \text{id}^{2^{r+1}}$ und $R = \varphi, Q_1, \dots, Q_{n!}$. Dabei bedeutet σ^n das Hintereinanderschreiben von n Permutationen σ . Sei nun schließlich $\Pi = (123 \dots n)R^{n!2^{n!}} (= (123 \dots n)R_1 \dots R_{n!2^{n!}}, \forall i : R_i = R)$. Da das Produkt aller Permutationen aus R gerade die identische Abbildung ist, ergibt das Produkt aller Permutationen aus Π gerade $(123 \dots n)$. Der Graph G_Π ist also nach Lemma 3.16 zusammenhängend. Sei K_i der Untergraph von G_0 , der R_i entspricht und L_{ij} der Untergraph von K_i , der Q_j entspricht.

Folgende Skizze verdeutlicht die Struktur von Π und L_{ij} :

$$\begin{array}{c} \overbrace{\Pi} \\ (12 \dots n), R_1, \dots, R_i, \dots, R_{n!2^{n!}} \\ \underbrace{\varphi, Q_1, \dots, Q_j, \dots, Q_{n!}} \\ \underbrace{id, \dots, id, \sigma_j, id, \dots, id} \end{array}$$



Es wird nun die built-in Relation \mathbf{B} in G_Π eingebettet.

Es existieren $n!2^{2n!}$ verschiedene L_{ij} ($i = 1, \dots, n!2^{n!}, j = 1, \dots, n!$). Jedes L_{ij} enthält genau $K := (2^{r+2} + 2)n$ Knoten. Da \mathbf{B} verpackbar ist, existiert nach Satz 3.12 ein $N_0, g, l \in \mathbb{N}$ derart, daß für alle $N > N_0$ auf $[N]$ $n!2^{2n!}$ disjunkte, paarweise paketisomorphe l -Paketmengen $P_i = \{P_{i1}, \dots, P_{im}\}$ existieren mit

$$K \leq ||P_i|| \leq K + n \quad \forall i = 1, \dots, n!2^{2n!} \text{ und}$$

$$\frac{n}{g} \leq |P_{ih}| \leq n \quad \forall i = 1, \dots, n!2^{2n!}, h = 1, \dots, m. \quad (*)$$

Das *Einbetten* wird nun so durchgeführt, daß folgende wichtige Bedingungen gewährleistet sind:

- (B1) Eine feste l -Paketmenge P_k wird dazu dienen, einen festen Untergraph L_{ij} aufzufüllen.
- (B2) Die Strukturen $L_{ij}, i = 1, \dots, n!2^{n!}, j = 1, \dots, n!$ sind paarweise **B**-isomorph.
- (B3) Ein einzelnes l -Paket wird maximal in 2 Spalten von L_{ij} eingebettet.

Sei $P_k = \{P_{k1}, \dots, P_{km}\}$. L_{ij} wird von links oben nach rechts unten spaltenweise mit den l -Paketen aus P_k aufgefüllt. Dabei werden die l -Pakete der Reihe nach vollständig ausgeschöpft, bis L_{ij} vollständig aufgefüllt ist (damit ist B1 erfüllt). Dies ist möglich, da $\|P_k\| \geq K$, so daß eventuell am Ende einige Knoten aus P_k übrig bleiben. Wegen (*) folgt insbesondere, daß ein festes l -Paket nur in maximal 2 Spalten zum Zuge kommt (siehe B3).

Nun wird L_{ij} mit $ml + (g + 1)$ Farben $(M_1, \dots, M_{ml}, C_1, \dots, C_{g+1})$ gefärbt. Dabei werden die ersten ml Farben verwendet, um die maximal ml Ausnahmeknoten aus P_k (jeweils maximal l Ausnahmeknoten in jedem l -Paket P_{ih}) unterschiedlich zu markieren und die verbleibenden $(g + 1)$ Farben um jeweils $(g + 1)$ aufeinanderfolgende l -Pakete unterscheidbar zu machen (entscheidend hierbei ist, daß wegen (*) in einer Spalte maximal $(g + 1)$ verschiedene l -Pakete zum Auffüllen benötigt werden).

Folgende Skizze verdeutlicht das Vorgehen beim Auffüllen von L_{ij} für folgende Parameter:

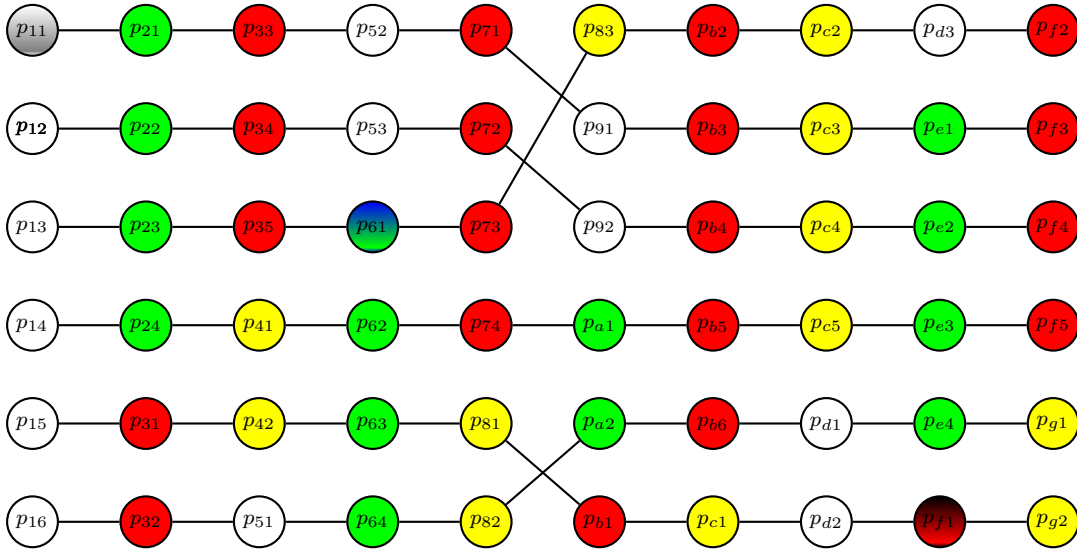
- $n = 6, r = 1$
- $\sigma_j = (123)(56)$
- $g = 3, l = 1$
- $\mathbf{P} = \{P_1, \dots, P_{16}\}$

$$\begin{array}{llll}
 P_1 = \{p_{11}, \dots, p_{16}\} & P_2 = \{p_{21}, \dots, p_{24}\} & P_3 = \{p_{31}, \dots, p_{35}\} & P_4 = \{p_{41}, p_{42}\} \\
 P_5 = \{p_{51}, \dots, p_{53}\} & P_6 = \{p_{61}, \dots, p_{64}\} & P_7 = \{p_{71}, \dots, p_{74}\} & P_8 = \{p_{81}, \dots, p_{83}\} \\
 P_9 = \{p_{91}, p_{92}\} & P_{10} = \{p_{a1}, p_{a2}\} & P_{11} = \{p_{b1}, \dots, p_{b6}\} & P_{12} = \{p_{c1}, \dots, p_{c5}\} \\
 P_{13} = \{p_{d1}, \dots, p_{d3}\} & P_{14} = \{p_{e1}, \dots, p_{e4}\} & P_{15} = \{p_{f1}, \dots, p_{f5}\} & P_{16} = \{p_{g1}, \dots, p_{g6}\}
 \end{array}$$

Die Markierungsfarben für Pakete sind: weiß, grün, rot, gelb.

- Ausnahmemengen: $M_1 = \{p_{11}\}, M_6 = \{p_{61}\}, M_{15} = \{p_{f1}\}, M_i = \emptyset$ für alle anderen 1-Pakete.

Die Markierungsfarben der Ausnahmeknoten sind: grau, blau, schwarz.



Seien $V_{L_{ij}} = \{v_{zs} : z = 1, \dots, n, s = 1, \dots, K\}$, $V_{L_{\hat{ij}}} = \{w_{zs} : z = 1, \dots, n, s = 1, \dots, K\}$ die Knotenmengen der Teilstrukturen L_{ij} und $L_{\hat{ij}}$. Dabei sei z der Zeilenindex und s der Spaltenindex.

Sei

$$f_{ij, \hat{ij}} : L_{ij} \rightarrow L_{\hat{ij}}, \quad v_{zs} \rightarrow w_{zs}$$

eine Abbildung von L_{ij} nach $L_{\hat{ij}}$.

Da die l -Paketmengen P_i paarweise paketisomorph sind, kann das oben beschriebene *Auffüllen* auf allen L_{ij} derart durchgeführt werden, daß alle L_{ij} paarweise $(B, C_1, \dots, C_{g+1}, M_1, \dots, M_{ml})$ -isomorph sind und die Abbildung $f_{ij, \hat{ij}}$ ein Isomorphismus von L_{ij} nach $L_{\hat{ij}}$ ist (siehe B2).

Bemerkung 3.17

Im weiteren Verlauf des Beweises wird gezeigt, daß Duplikator für alle i eine große Menge A_i von Indexen finden kann derart, daß er eine spaltenrespektierende r -Runden Gewinnstrategie auf L_{ij} und $\hat{L}_{\hat{ij}}$ hat. Dabei ist $\hat{j} \in A_i$ und $\hat{L}_{\hat{ij}}$ der Teilgraph, der entsteht, indem man die innerste Permutation von L_{ij} durch $\sigma_{\hat{j}}$ ersetzt. L_{ij} und $\hat{L}_{\hat{ij}}$ unterscheiden sich also nur bzgl. der Kantenrelation in der innersten Permutation.

Damit diese r -Runden Strategie auch global wirksam ist, muß gewährleistet werden, daß gespielte Knoten sich bezüglich der Knoten außerhalb von L_{ij} identisch verhalten.

Die ml Farben (M_1, \dots, M_{ml}) sorgen hierbei dafür, daß Ausnahmeknoten (die sich bezüglich der Außenwelt beliebig verhalten können) identisch gespielt werden.

Die $g + 1$ Farben (C_1, \dots, C_{g+1}) sorgen dafür, daß Duplikator jeweils nur mit Knoten aus einem festen l -Paket antwortet. Diese haben die Eigenschaft sich bezüglich der Außenwelt identisch zu verhalten.

Betrachtet man die Situation soweit, so hat man einen Graphen G_Π mit \bar{N} Knoten und eine built-in Relation B_N auf N Knoten (dieses N muß gemäß Satz 3.12 geeignet gewählt worden sein). Dabei ist i.a. $\bar{N} < N$. Der Spielgraph G_0 muß nun so gewählt werden, daß B_N vollständig auf ihn abgebildet werden kann. Dazu wird G_Π um $N - \bar{N}$ Knoten erweitert, die in einer Kette miteinander verbunden werden und an **genau einer** markierten Stelle mit dem Graph G_Π verbunden werden. Der so erhaltene Graph ist natürlich weiterhin zusammenhängend und entspricht bezüglich der Kantenrelation E schon vollständig dem Spielgraphen G_0 . Die Einbettung von B_N wurde auf dem Teilgraphen G_Π oben definiert. Die Fortsetzung dieser Einbettung auf G_0 kann nun beliebig geschehen. Damit ist die Konstruktion von G_0 abgeschlossen. Spoiler färbt nun G_0 mit c Farben.

Bemerkung 3.18

Um die Notation zu vereinfachen, steht im Folgenden c für die Anzahl der insgesamt verwendeten Farben. Das heißt:

$$c := c + ml + (g + 1).$$

Schritt 2: Konstruktion von G_1 .

- (i) In jedem K_i findet Duplikator eine große Menge A_i von Indizes, derart daß er eine Gewinnstrategie auf allen Paaren $L_{ij}, L_{i\bar{j}}, j, \bar{j} \in A_i$, hat:

Jedes L_{ij} besteht aus $K := (2^{r+2} + 2)n$ Knoten. Sei $\delta_{ij}(x)$ der Abstand von $x \in L_{ij}$ zu den beiden innersten Spalten in L_{ij} bzgl. der Kantenrelation E . Es existieren 2^{cK} verschiedene Färbungen von L_{ij} . Daher gibt es eine Menge \bar{A}_i von Indizes mit $|\bar{A}_i| \geq \frac{n!}{2^{cK}}$ derart, daß alle $L_{ij}, j \in \bar{A}_i$, identisch gefärbt sind. Es gibt nur \mathcal{N} viele r -Spieltypen auf derart identisch gefärbten L_{ij} Strukturen mit Distanzfunktion. \mathcal{N} hängt dabei nur von c, r (damit auch von $ml + (g + 1)$) und dem Bildbereich von δ_{ij} ab, nicht aber von n . Es gibt also eine Teilmenge $A_i \subset \bar{A}_i$ mit $|A_i| \geq \frac{n!}{2^{cK\mathcal{N}}}$ derart, daß für alle $j, \bar{j} \in A_i$ Duplikator eine Gewinnstrategie im r -Runden FO-Spiel auf L_{ij} und $L_{i\bar{j}}$ hat, die δ_{ij} und $\delta_{i\bar{j}}$ respektiert. Auch sind L_{ij} und $L_{i\bar{j}}$ in dem Fall identisch gefärbt.

- (ii) Die möglichen Änderungen von Π zu $\bar{\Pi}$ können so gewählt werden, daß $G_{\bar{\Pi}}$ nicht zusammenhängend ist:

Sei $j_i \in A_i$ fest. $R_i(j)$ sei die Permutationsfolge, die entsteht, wenn man in R_i die Permutation σ_{j_i} durch σ_j ersetzt. Das Produkt der Permutationen aus $R_i(j)$ sei π_{ij} . Sei weiterhin

$$X_i := \{\pi_{ij}; j \in A_i\}.$$

X_i ist also eine Menge von Permutationen, die man durch Veränderungen von L_{ij} erhält, welche vom Spoiler nicht nachgewiesen werden können. Es gibt $2^{n!}$ Teilmengen von S_n . Π besteht aus $n!2^{n!}$ vielen Kopien von R . Daher existiert ein $Y \subset S_n$, welches mindestens $n!$ -mal unter den X_i vorkommt.

Sei $U \leq S_n$ das Erzeugnis von Y . Es gilt:

$$|U| \geq |Y| \geq \frac{n!}{2^{cK\mathcal{N}}}.$$

Da \mathcal{N} nicht von n abhängt und K linear in n ist, existiert ein $n_0 \in \mathbb{N}$, so daß für alle $n \geq n_0$:

$$\frac{n!}{2^{cK\mathcal{N}}} > n! \left(\frac{6}{\log n} \right)^n.$$

Nach Coppersmith existiert daher ein $g \in U$ derart, daß $(123 \cdots n)g$ kein n -Zyklus ist. Da U von Y erzeugt wird und $|S_n| \leq n!$, existieren ein $m \leq n!$ und $y_1, \dots, y_m \in Y$ mit $g = y_1 \cdots y_m$. Da Y mindestens $n!$ -mal unter den X_i vorkommt, existieren paarweise verschiedene Indizes i_1, \dots, i_m und j_1, \dots, j_m derart, daß für alle $q \leq m$ gilt:

- $j_q \in A_{i_q}$
- $\pi_{i_q j_q} = y_q$.

Sei nun G_1 der Graph, der entsteht, indem man in G_0 $L_{i_q j_{i_q}}$ durch $L_{i_q j_q}$ ersetzt ($q \leq m$). Das Produkt der Permutationen von G_1 ist dann $(123 \cdots n)y_1 \cdots y_m = (123 \cdots n)g$, was kein n -Zyklus ist. Folglich ist G_1 nicht zusammenhängend.

Duplikator färbt nun G_1 identisch zu G_0 .

Schritt 3: Duplikator hat eine Gewinnstrategie auf G_0 und G_1 :

Für $q \leq m$ sei H_q der Teilgraph, der aus den innersten beiden Spalten von $L_{i_q j_{i_q}}$ in G_0 besteht und $\delta_q := \delta_{i_q j_{i_q}}$. Analog sind \overline{H}_q und $\overline{\delta}_q$ auf G_1 definiert. Es wird nun gezeigt, daß G_0, H_1, \dots, H_m und $G_1, \overline{H}_1, \dots, \overline{H}_m$ die Bedingungen des starken Erweiterungssatzes erfüllen.

Zunächst gilt:

$$\mathcal{N}_{2^r}^{\delta_i}(H_i) \cap \mathcal{N}_{2^r}^{\delta_j}(H_j) = \emptyset = \mathcal{N}_{2^r}^{\overline{\delta}_i}(\overline{H}_i) \cap \mathcal{N}_{2^r}^{\overline{\delta}_j}(\overline{H}_j) \quad \text{für } i \neq j.$$

zu (i): Dies ergibt sich direkt aus der Konstruktion.

zu (ii): Setze $\alpha(x) = x$ für $x \notin H_1 \cup \dots \cup H_m$.

zu (iii): E : Folgt direkt aus der Tatsache, daß zwei Punkte, deren Abstand zueinander größer als 1 ist, nicht miteinander verbunden sind.

B: Seien $x \in G_0$ und $\bar{x} \in G_1$ und $j \leq m$ mit

- $\delta_j(x) = \overline{\delta}_j(\bar{x}) \leq 2^r$

- x, \bar{x} sind bzgl. Spoilers Färbung und den $ml+(g+1)$ Hilfsfarben identisch gefärbt,

dann müssen x, \bar{x} Knoten aus einem festen l -Paket $P_{x\bar{x}}$ sein. Dies folgt aus der Art und Weise, wie B_N in G_0 eingebettet wurde und der Hilfsfärbung mit $(g+1) + ml$ Farben (gleichgefärbte Knoten innerhalb einer Spalte sind aus einem festen l -Paket). Es gibt nun zwei Fälle:

1. Fall: $x \notin$ Ausnahmemenge von $P_{x\bar{x}}$. Damit ist x mit keiner der ml Ausnahmefarben gefärbt und somit \bar{x} auch nicht. Die Bedingung (iii) folgt aus den Eigenschaften von Knoten eines l -Paketes.
2. Fall: $x \in$ Ausnahmemenge von $P_{x\bar{x}}$. Damit ist x der einzige Knoten in $L_{i_q j_{i_q}}$ mit einer bestimmten Ausnahmefarbe M_i . Selbiges gilt nun aber für \bar{x} und nach Konstruktion gilt $x = \bar{x}$, woraus (iii) folgt.

Damit kann der starke Erweiterungssatz angewendet werden und der Satz ist bewiesen. \square

Bemerkung 3.19

Wichtig für den Beweis ist, daß nur **eine** verpackbare built-in Relation vorkommt. Für den Fall von zwei verpackbaren built-in Relationen funktionieren obige Methoden nicht.

Nach einem unveröffentlichtem Ergebnis von Schweikardt und Schwentick kann man zwei built-in Ordnungen \leq_1, \leq_2 derart finden, daß

$$\text{monNP}(\leq_1, \leq_2) = \text{monNP}(+).$$

Da built-in Ordnungen verpackbar sind, würde ein Beweis von Satz 3.8 für zwei verpackbare built-in Relationen zeigen, daß Graphzusammenhang nicht ausdrückbar ist in monadic NP + built-in Addition.

3.4 Anwendungen

In diesem Abschnitt wird gezeigt, daß built-in Ordnungen, built-in Äquivalenzrelationen (unter bestimmten Bedingungen) und built-in Relationen mit beschränkter Baumweite verpackbar sind.

Damit ergibt sich direkt, daß Graphzusammenhang nicht ausdrückbar ist in monadic NP + eine built-in Relation aus diesen Klassen.

Im Falle von built-in Ordnungen und built-in Relationen mit beschränkter Baumweite handelt es sich um bekannte Ausdrucksschwäche-Resultate.

3.4.1 Ordnungen

Satz 3.20

Built-in Ordnungen sind verpackbar.

Beweis:

Ergibt sich direkt aus der Beobachtung, daß jedes Intervall ein Paket ist. \square

3.4.2 Äquivalenzrelationen

Satz 3.21

Graphzusammenhang ist nicht ausdrückbar in monadic NP + beliebige Äquivalenzrelation A.

Beweis:

Sei $\mathbf{A} = (A_n)_{n \in \mathbb{N}}$ eine beliebige built-in Äquivalenzrelation. Es müssen zwei Fälle unterschieden werden:

1. Fall: Es existiert ein $n_0 \in \mathbb{N}$, so daß für alle $n \geq n_0$ A_n Maximalgrad $n^{o(1)}$ hat. Dann folgt die Behauptung aus Satz 4.7.
2. Fall: Falls Fall 1 nicht eintritt, existiert ein $\epsilon > 0$, so daß für jedes $N_0 \in \mathbb{N}$ ein $N \geq N_0$ existiert derart, daß A_N Maximalgrad größer als N^ϵ hat. Insbesondere gibt es in A_N eine Äquivalenzklasse $\ddot{\text{ÄK}}$ mit $|\ddot{\text{ÄK}}| \geq N^\epsilon$. Da jeder Knoten aus $\ddot{\text{ÄK}}$ genau mit allen anderen Knoten aus $\ddot{\text{ÄK}}$ verbunden ist, kann man aus dieser Äquivalenzklasse *beliebig* viele, *beliebig* große 1-Pakete bilden. Damit ist in diesem Falle die built-in Relation \mathbf{A} verpackbar und die Behauptung folgt aus Satz 3.8. \square

Bemerkung 3.22

Satz 3.21 läßt sich auch leicht direkt beweisen, indem man das bekannte ein Kreis vs. zwei Kreise Spiel auf einer großen Äquivalenzklasse spielt und beachtet, daß sich dann alle Kreisknoten bzgl. der Außenwelt identisch verhalten.

3.4.3 Relationen mit beschränkter Baumweite

Auch built-in Relationen mit beschränkter Baumweite sind verpackbar. Damit kann Graphzusammenhang nicht ausgedrückt werden in monadic NP + eine built-in Relation mit beschränkter Baumweite. Dieses Ergebnis konnte auch schon von Kreidler in [17] (mit völlig anderen Methoden) gezeigt werden.

Kreidler zeigt, daß built-in Relationen mit beschränkter Baumweite zerlegbar sind (siehe Bemerkung 3.5). Dabei ist der Nachweis der Zerlegbarkeit (aus meiner Sicht) deutlich aufwendiger und weit weniger intuitiv, als der Nachweis von Verpackbarkeit.

Zunächst einige Definitionen:

Definition 3.23

Sei G ein Graph, T ein Baum mit Wurzel $\text{root}(T)$ und $\mathcal{V} = (\mathcal{V}_t)_{t \in T}$ eine Familie von Knotenmengen $\mathcal{V}_t \subset V(G)$. Wir nennen das Paar (T, \mathcal{V}) eine Baumzerlegung von G , wenn die folgenden drei Bedingungen erfüllt sind:

- (T1) $V(G) = \cup_{t \in T} \mathcal{V}_t$.
- (T2) Zu jeder Kante e von G gibt es ein $t \in T$, so daß \mathcal{V}_t beide Endknoten von e enthält.
- (T3) Falls t_1, t_2, t_3 Knoten in $V(T)$ sind und t_2 auf einem T -Pfad von t_1 nach t_3 liegt, dann gilt $\mathcal{V}_{t_1} \cap \mathcal{V}_{t_3} \subset \mathcal{V}_{t_2}$.

Definition 3.24

- a) Sei (T, \mathcal{V}) eine Baumzerlegung eines Graphen G . Durch

$$W((T, \mathcal{V})) := \max\{|\mathcal{V}_t| - 1; t \in T\}$$

ist die Weite der Baumzerlegung (T, \mathcal{V}) definiert. Die kleinste Weite einer Baumzerlegung von G heißt **Baumweite** $\text{tw}(G)$ von G .

- b) Sei $\mathbf{B} = (B_n)_{n \in \mathbb{N}}$ eine built-in Relation. \mathbf{B} hat **beschränkte Baumweite**, falls ein $k \in \mathbb{N}$ existiert, so daß $\text{tw}(B_n) \leq k$ für alle $n \in \mathbb{N}$.

Folgende Definition wird die Beweisführung des folgenden Satzes übersichtlicher gestalten:

Definition 3.25

Sei (T, \mathcal{V}) eine Baumzerlegung eines Graphen G . Sei $t \in V(T)$.

- a) Der an t hängende Teilbaum von T wird mit T_t bezeichnet. t ist die Wurzel von T_t .
- b) $V_{T_t} := \cup_{t \in T_t} \mathcal{V}_t$ ist die Menge aller Knoten aus G , die in der Teilbaumzerlegung (T_t, \mathcal{V}) vorkommen.
- c) Seien t_1, \dots, t_l die Kinder von t . t_{\max} sei dasjenige Kind t_i von t , für das $|\mathcal{V}_{T_{t_i}}|$ maximal wird. Existieren mehrere solche Kinder, so sei t_{\max} ein beliebiges aus dieser Menge.

Satz 3.26

Built-in Relationen mit beschränkter Baumweite sind verpackbar. Der Parameter g kann dabei jede beliebige natürliche Zahl ungleich 1 sein.

Beweis:

Sei \mathbf{B} eine built-in Relation mit beschränkter Baumweite. Sei $k := \max_{n \in \mathbb{N}} \text{tw}(B_n)$ die Baumweite von \mathbf{B} .

Zu zeigen ist:

Es existiert ein $l \in \mathbb{N}$ derart, daß für alle $L, n, g \in \mathbb{N}, g \geq 2$ ein $N_0 \in \mathbb{N}$ existiert so, daß für alle $N \geq N_0$, L disjunkte l -Pakete P_i mit $\frac{n}{g} \leq |P_i| \leq n$ existieren.

Es wird bewiesen, daß $l = 2k$ gewählt werden kann.

Sei (T^0, \mathcal{V}) eine Baumzerlegung von B_N mit minimaler Baumweite.

Der folgende Algorithmus liefert bei Eingabe einer *markierten* Baumzerlegung (T^{i-1}, \mathcal{V}) und der Menge $\mathbf{P} = \cup_{j=1}^{i-1} P_j$ aller Knoten, die in vorherigen l -Paketen verwendet werden, das i -te Paket P_i .

Weiterhin liefert der Algorithmus eine *markierte* Baumzerlegung (T^i, \mathcal{V}) , mit deren Hilfe das nächste l -Paket P_{i+1} berechnet werden kann.

Markierte Baumzerlegung bedeutet hierbei, daß einige Knoten in T^{i-1} markiert sein können.

Der Algorithmus ist also *induktiv* zu verwenden. Beginnend mit der Eingabe (T^0, \mathcal{V}) und $\mathbf{P} = \emptyset$ wird P_1 und (T^1, \mathcal{V}) berechnet. Mit (T^1, \mathcal{V}) und $\mathbf{P} = P_1$ wird P_2 und (T^2, \mathcal{V}) berechnet usw.

Algorithmus

Eingabe: markierte Baumzerlegung (T^{i-1}, \mathcal{V}) und $\mathbf{P} = \cup_{j=1}^{i-1} P_j$

Ausgabe: l -Paket P_i und markierte Baumzerlegung (T^i, \mathcal{V})

```

01   $P_i := \emptyset; \quad T := T^{i-1};$ 
02  if (Es existiert kein markierter Punkt in  $T$  ) then
    // dies passiert nur bei Berechnung von  $P_1$ .
03       $t_0 = \text{root}(T^{i-1});$ 
04  else
    // falls es markierte Knoten gibt
05a       $t_0 :=$  Ein beliebiger markierter Knoten derart,
           daß in  $T_{t_0}$  keine markierten Knoten vorkommen;
           // Suche einen Knoten, dessen Unterbaum  $T_{t_0}$  keine
           // weiteren markierten Knoten enthält.
05b       $t_h := t_0$  // Hilfsknoten für Beweis von Satz 3.31
06  if ( $t_0 == \text{root}(T^{i-1})$ ) or ( $|V_{T_{t_0}} \setminus \mathbf{P}| \geq \frac{n}{g}$ ) then
    // Falls kein markierter Knoten existiert oder ausreichend viele
    // neue Knoten (außerhalb von  $\mathbf{P}$ ) in  $V_{T_{t_0}}$  liegen

```

```

07     repeat                                     // Baum runterlaufen bis zum ersten
08          $t_0 := (t_0)_{\max};$                  // ein Unterbaum weniger als
09     until  $|V_{T_{t_0}} \setminus \mathbf{P}| \leq n;$  //  $n$  neue Knoten enthält.
10      $P_i = V_{T_{t_0}} \setminus \mathbf{P};$     $T := T \setminus T_{t_0};$ 
11     while  $(|P_i| < \frac{n}{g})$  do{           // Falls  $T_{t_0}$  zu wenige Knoten enthält
12          $t_0 = (\text{vater}(t_0))_{\max};$  // müssen weitere Unterbäume von
13          $P_i := P_i \cup V_{T_{t_0}} \setminus \mathbf{P};$  // vater( $t_0$ ) ergänzt werden.
14          $T := T \setminus T_{t_0};$  // Diese werden aus T entfernt.
15     markiere vater( $t_0$ ); // Vater von  $t_0$  muß markiert werden,
                                // er eventuell Ausnahmeknoten enthält
16 else {
    // falls  $V_{T_{t_0}}$  zuwenig neue Knoten enthält, verzichte auf diese.
17      $T := T \setminus T_{t_0};$ 
18     repeat
19         repeat
    // Den Baum aufwärts laufend, wird der erste Knoten gesucht
    // der entweder markiert ist oder genügend neue Knoten in
    // seinem Teilbaum enthält
20          $t_1 := t_0;$  // Hilfsknoten: Merkt sich den Sohnknoten
21                     // von dem man gekommen ist
22          $t_0 := \text{vater}(t_0);$ 
23     until  $(|V_{T_{t_0}} \setminus \{\mathbf{P} \cup V_{T_{t_1}}\}| \geq \frac{n}{g})$  or  $(|V_{T_{t_1}} \setminus \mathbf{P}| \geq \frac{n}{g})$  or  $(t_0$  ist markiert)
24     if  $|V_{T_{t_0}} \setminus \{\mathbf{P} \cup V_{T_{t_1}}\}| \geq \frac{n}{g}$  then {
    // Falls  $V_{T_{t_0}} \setminus \{\mathbf{P} \cup V_{T_{t_1}}\}$  ausreichend viele neue Knoten für  $P_i$ 
25     markiere  $t_0;$  //  $t_0$  könnte Ausnahmeknoten enthalten.
26      $T := T \setminus T_{t_1};$  // Verzichte auf die Knoten aus  $T_{t_1}$  (max.  $2\frac{n}{g}$ 
                                // Diese würden weitere Ausnahmeknoten liefern)
27     wiederhole Zeilen 7-15; Ende; }
    //  $(T_{t_0}, \mathcal{V})$  ist nun eine Baumzerlegung ohne markierten Knoten
    // aber ausreichend vielen neuen Knoten. Um  $P_i$  zu bestimmen
    // ebenso vorgegangen werden, wie bei der Bestimmung von  $P_i$ 
28 else if  $|V_{T_{t_1}} \setminus \mathbf{P}| \geq \frac{n}{g}$  then {
    // falls  $V_{T_{t_1}} \setminus \mathbf{P}$  ausreichend viele neue Knoten für  $P_i$  enthält
    // Bem.:  $\frac{n}{g} \leq |V_{T_{t_1}} \setminus \mathbf{P}| \leq 2\frac{n}{g}$ 
29      $P_i := V_{T_{t_1}} \setminus \mathbf{P};$  // max. 2k Ausnahmeknoten.
30      $T := T \setminus T_{t_1};$ 
31     markiere  $t_0;$  Ende; }

```

```

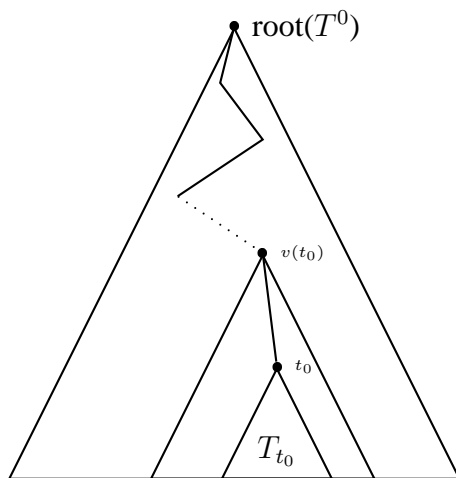
32         else
           // Falls auf dem Weg bis zum nächsten markierten Knoten nicht
           // ausreichend viele neue Knoten gesammelt werden können,
           // verzichte auf  $T_{t_0} \setminus \{t_0\}$ . Das sind max.  $2^{\frac{n}{g}}$  viele Knoten.
33a          $T := T \setminus \{T_{t_0} \setminus t_0\}$ ;
33b          $t_h := t_0$  // Hilfsknoten für Beweis von Satz 3.31
34     until True = False; // Endlosschleife
    }

```

Die Grundidee, die hinter dem Algorithmus steckt, ist die Tatsache, daß die Knotenmenge $V_{T_{t_0}}$ eines Teilbaumes T_{t_0} nur über die Knoten aus \mathcal{V}_{t_0} mit Knoten außerhalb von $V_{T_{t_0}}$ verbunden sein können. Daher bilden die Knoten aus $V_{T_{t_0}}$ für $|V_{t_0}| \leq k$ ein k -Paket.

Das folgende Schema beschreibt den Programmablauf. $v(t)$ steht dabei für den Vaterknoten von t .

$i = 1$: Berechnung von P_1 .

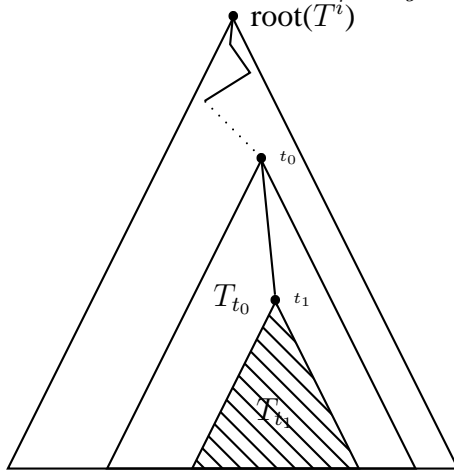


- $|V_{T_{v(t_0)}}| > n$
- $|V_{T_{t_0}}| \leq n$
- Zeilen 10-15 liefern P_1 .
- Knoten in $\mathcal{V}_{v(t_0)}$ können im weiteren Verlauf Ausnahmeknoten werden. Daher wird $v(t_0)$ markiert.

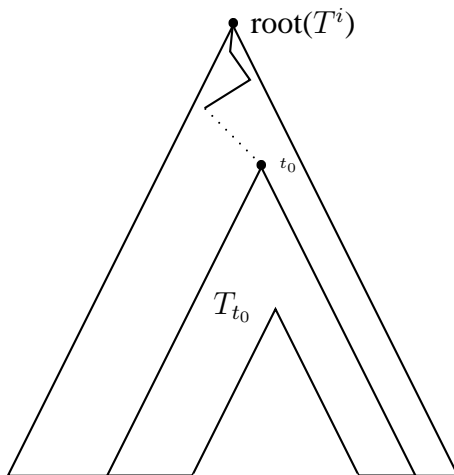
$i \rightarrow i + 1$: Sei t_0 ein markierter Knoten in T^i derart, daß in T_{t_0} keine weiteren Knoten markiert sind. Falls in T_{t_0} ausreichend viele neue Knoten vorkommen, das bedeutet $|V_{T_{t_0}} \setminus \mathbf{P}| \geq \frac{n}{g}$, kann analog zum Fall $i = 1$ verfahren werden.

Falls $|V_{T_{t_0}} \setminus \mathbf{P}| < \frac{n}{g}$ wird zunächst T_{t_0} aus T^i gelöscht. Dabei *verzichtet* man auf weniger als $\frac{n}{g}$ Knoten. Im Anschluß wird der Baum sukzessive nach oben abgelaufen. Sei dabei t_0 der aktuelle Knoten und t_1 der Sohnknoten, von dem man gekommen ist. In jedem Schritt müssen vier Fälle unterschieden werden:

1. $|V_{T_{t_0}} \setminus \{\mathbf{P} \cup V_{T_{t_1}}\}| \geq \frac{n}{g}$:

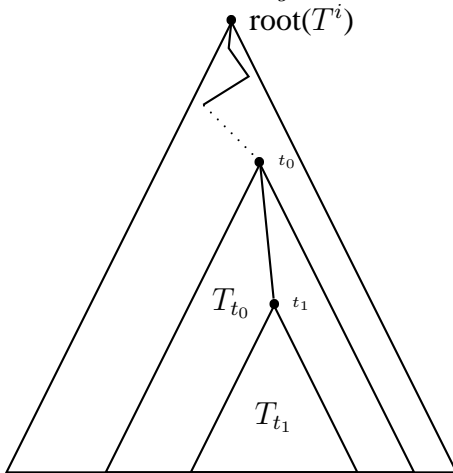


- $|V_{T_{t_0}} \setminus \{\mathbf{P} \cup V_{T_{t_1}}\}| \geq \frac{n}{g}$
- $|V_{T_{t_1}}| < 2\frac{n}{g}$, sonst wäre die repeat-Schleife aus Zeile 19 schon vorher abgebrochen worden.
- Lösche T_{t_1} aus T^i .



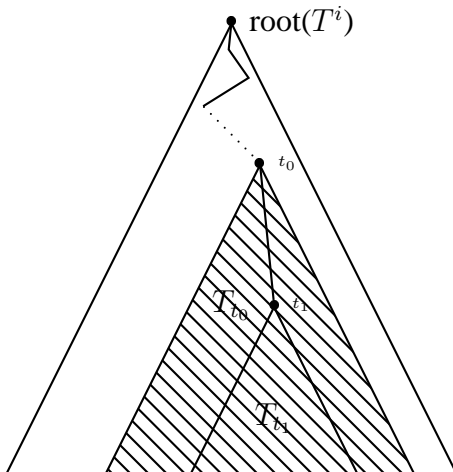
- Markiere t_0 .
- $|V_{T_{t_0}} \setminus \{\mathbf{P}\}| \geq \frac{n}{g}$
- T_{t_0} enthält ausreichend viele neue Knoten und nur der Knoten t_0 ist markiert. Auf T_{t_0} kann also wie im Fall $i = 1$ verfahren werden.

2. $|V_{T_{t_1}} \setminus \mathbf{P}| > \frac{n}{g}$:



- $|V_{T_{t_1}} \setminus \mathbf{P}| \geq \frac{n}{g}$
- $|V_{T_{t_1}} \setminus \mathbf{P}| \leq 2\frac{n}{g}$, sonst wäre die repeat-Schleife aus Zeile 19 schon vorher abgebrochen worden.
- In T_{t_1} sind keine markierten Knoten. Die Menge $V_{T_{t_1}} \setminus \mathbf{P}$ mit Ausnahmemenge \mathcal{V}_{t_1} ist ein k -Paket geeigneter Größe. Setze $P_{i+1} = V_{T_{t_1}} \setminus \mathbf{P}$.

3. t_0 ist markiert:



- $|V_{T_{t_0}} \setminus \{\mathbf{P} \cup V_{T_{t_1}}\}| < \frac{n}{g}$
- $|V_{T_{t_1}} \setminus \mathbf{P}| < \frac{n}{g}$
- Lösche T_{t_0} . Das sind weniger als $2\frac{n}{g}$ Knoten.
- Setze Aufwärtslaufen fort.

4. Keiner der drei Fälle 1.-3. tritt ein: Setze Aufwärtslaufen fort.

Korrektheit des Algorithmus:

Induktion über i .

$i = 1$: Die Eingabe bei Berechnung von P_1 besteht aus der Baumzerlegung (T^0, \mathcal{V}) von B_N (T^0 enthält keine markierten Knoten) und der Menge $\mathbf{P} = \emptyset$.

Da T^0 keine markierten Knoten enthält, wird P_1 in den Zeilen 07-15 berechnet. Man sieht leicht, daß der Algorithmus in diesem Falle eine Menge P_1 mit

$n \geq |P_1| > \frac{n}{g}$ zurückliefert (siehe Zeile 09 + 11). P_1 ist eine Vereinigung von $V_{T_{t_0}}, \dots, V_{T_{t_r}}$, wobei t_0, \dots, t_r jeweils denselben Vaterknoten t in T^0 haben (Zeilen 12-14). Aufgrund der Definition von Baumzerlegung sind die Knoten aus $P_1 \cap V_t$ die einzigen Knoten aus P_1 , die mit Knoten außerhalb von P_1 verbunden sein können. Da $|V_t| \leq k$ ist P_1 ein k -Paket.

$i \rightarrow i + 1$: Um die Korrektheit für beliebiges $i > 1$ zu beweisen, zunächst einige Hilfslemmata:

Lemma 3.27

In einem Durchlauf des Algorithmus werden maximal zwei Knoten markiert.

Beweis:

Nur in den Zeilen 15, 25 und 31 wird ein Knoten markiert. Diese Zeilen werden im Laufe des Algorithmus höchstens einmal bearbeitet. Dabei ist es nicht möglich, daß alle drei Zeilen bearbeitet werden. \square

Lemma 3.28

In L Durchläufen des Algorithmus wird maximal auf $2 \cdot L \cdot \frac{n}{g}$ Knoten von B_N verzichtet.

Beweis:

Dies folgt sofort aus Lemma 3.27, da nur aufgrund eines markierten Knotens von T auf maximal $2 \frac{n}{g}$ Knoten verzichtet wird (Zeilen 17, 26 und 33a). \square

Bemerkung: Um zu garantieren, daß man L disjunkte k -Pakete finden kann, muß wegen des letzten Lemmas die Größe N des Universums mindestens $L \cdot 4 \cdot \frac{n}{g} + Ln$ betragen.

Lemma 3.29

Ein Wurzelweg in einem Baum T ist ein Weg ($\text{root}(T) =: w_0, w_1, \dots, w_n$) in T mit $\text{vater}(w_{i+1}) = w_i$. Ein Wurzelweg ist also ein Weg mit Abwärtsrichtung und Startknoten $\text{root}(T)$.

Für alle $j = 0, \dots, L$ liegen alle markierten Knoten aus T^j auf einem Wurzelweg in T^j .

Beweis:

Induktion nach j .

$j = 0$: In T^0 ist kein Knoten markiert.

$j \rightarrow j + 1$:

Nach Voraussetzung existiert ein Wurzelweg $W^j := (w_0^j, \dots, w_n^j = t_0)$ in T^j . Es werden nun die verschiedenen Programmabläufe untersucht.

1. $|V_{T_{t_0}} \setminus \mathbf{P}| \geq \frac{n}{g}$ (Zeilen 06-15):
In Zeile 15 wird ein Knoten in T_{t_0} markiert. Es ist leicht einen geeigneten Wurzelweg in T^{j+1} zu finden.
2. 1. trifft nicht zu. T_{t_0} wird gelöscht (Zeile 17). Von t_0 aus wird der Baum T^j (der Wurzelweg W^j) aufwärts abgelaufen, bis einer der folgenden Fälle eintritt (Zeilen 18-34):
 - 2a. $|V_{T_{t_0}} \setminus \{\mathbf{P} \cup V_{T_{t_1}}\}| \geq \frac{n}{g}$ (Zeilen 24-27):
Da $t_0 \in W^j$ existiert $\hat{n} < n$ mit $t_0 = w_{\hat{n}}^j$. In Zeile 26 wird T_{t_1} gelöscht und damit die Knoten $w_{\hat{n}+1}^j, \dots, w_{\hat{n}}^j$. t_0 wird markiert (Zeile 25) und im weiteren Verlauf wird nur noch ein Knoten in T_{t_0} markiert (Zeile 27). Ein geeigneter Wurzelweg für T^{j+1} kann leicht bestimmt werden.
 - 2b. $|V_{T_{t_1}} \setminus \mathbf{P}| \geq \frac{n}{g}$ (Zeile 28-31):
Da $t_1 \in W^j$ existiert $\hat{n} < n$ mit $t_0 = \text{vater}(t_1) = w_{\hat{n}}^j$. In Zeile 30 wird T_{t_1} gelöscht. In Zeile 31 t_0 markiert und der Algorithmus terminiert. $W^{j+1} := (w_0^j, \dots, w_{\hat{n}}^j)$ ist ein geeigneter Wurzelweg in T^{j+1} .
 - 2c. Weder 2a. noch 2b. In Zeile 33 wird $T_{t_0} \setminus \{t_0\}$ gelöscht. Dann wird 2. wiederholt.

□

Lemma 3.30*Es gilt*

- a) $T^i \subset T^j$, $i < j$.
- b) T^i , $i > 0$ unterscheidet sich von T^0 nur unterhalb von markierten Knoten.

Beweis: a) Der Eingabebaum wird im Algorithmus nicht erweitert.b) Immer wenn ein Teil des Eingabebaumes T^i gelöscht wird, wird der Vaterknoten dieses Teiles markiert. □**Satz 3.31***Der Algorithmus liefert für alle $i = 2, \dots, M$ ein $2k$ -Paket P_i . Diese sind paarweise disjunkt und es gilt*

$$\frac{n}{c} \leq |P_i| \leq n.$$

Beweis:Das erste und zweite Lemma garantieren, daß bei ausreichend großem Grundraum $[N]$ immer genügend Knoten in (T^i, \mathcal{V}) übrig bleiben, um weiter $2k$ -Pakete zu konstruieren.Der Algorithmus liefert für alle $i = 2, \dots, L$ eine Menge P_i . Für diese gilt:

$$(1) \frac{n}{g} \leq |P_i| \leq n:$$

Dies wird garantiert durch die Zeilen(paare) (9,11), 24 und 28.

$$(2) \text{ Es gibt maximal } 2k \text{ Knoten in } P_i, \text{ die mit Knoten au\ss}erhalb \text{ von } P_i \text{ verbunden sind:}$$

Hier mu\ss unterschieden werden, ob der Algorithmus in Zeile 15,27 oder 31 terminiert:

15: Die Begr\u00fcndung entspricht der obigen f\u00fcr den Fall $i = 1$.

27: Aufgrund von Lemma 3.29 ist im Teilbaum T_{t_0} nach Entfernung von T_{t_1} nur noch der Knoten t_0 markiert. Damit entspricht die Situation derjenigen f\u00fcr den Fall $i = 1$.

31: Aufgrund der Definition von beschr\u00e4nkter Baumweite, der Tatsache, da\ss nur die Knoten t_1 und t_h in T_{t_1} markiert sind und Lemma 3.30 b), kann die Knotenmenge $V_{T_{t_1}}$ nur \u00fcber Knoten aus \mathcal{V}_{t_1} oder \mathcal{V}_{t_h} mit Knoten au\sserhalb von $V_{T_{t_1}}$ verbunden sein. Diese beiden Mengen enthalten maximal $2k$ Knoten.

Damit ist die Behauptung bewiesen. □

3.5 Verpackbare Relationen bez\u00fcglich Padding

Mit dem bisher Gezeigten ist es *leicht* zu sehen, da\ss Graphzusammenhang auch nicht in

monadic NP + eine verpackbare built-in Relation + beliebiges Padding

ausdr\u00fcckbar ist.

Im Beweis von Satz 3.8 wurde eine *Restmenge* zu einer Kette verbunden und an den dort konstruierten Graphen G_{Π} geh\u00e4ngt. Diese Restmenge kann durch Vergr\u00f6\sserung des Startuniversums *beliebig* gro\ss werden. Die Knoten dieser Restmenge werden im Spielverlauf des Beweises konstant gespielt. Man kann also *beliebig* viele *Padding-knoten* aus dieser Restmenge sch\u00f6pfen. Damit ist auch Satz 3.10 bewiesen.

Kapitel 4

Unendlich teilbare built-in-Relationen

Im vorherigen Kapitel wurde die Klasse der verpackbaren Relationen definiert und bewiesen, daß es sich dabei um schwache built-in Relationen handelt. Desweiteren wurde gezeigt, daß Graphzusammenhang nicht ausdrückbar ist in $\text{monNP} + \text{verpackbare built-in Relation}$. Verpackbarkeit ist eine Struktureigenschaft von built-in Relationen mit *globalem* Charakter.

In diesem Kapitel wird die Klasse der unendlich teilbaren built-in Relationen eingeführt.

Unendliche Teilbarkeit ist eine Struktureigenschaft von built-in Relationen mit *lokalem* Charakter. Es wird gezeigt, daß Graphzusammenhang nicht ausdrückbar ist in $\text{monadicNP} + \text{unendlich teilbare built-in Relation}$. Im Anwendungsabschnitt wird gezeigt, daß eine große Menge von built-in Relationen unendlich teilbar sind.

4.1 Unendlich teilbare built-in Relationen

Definition 4.1

Eine built-in Relation \mathbf{B} heißt **unendlich teilbar**, falls für alle $d \in \mathbb{N}$ ein $l_d \in \mathbb{N}$ existiert, so daß für alle $L \in \mathbb{N}$ ein $n \in \mathbb{N}$ existiert derart, daß eine Menge $A_n \subset [n]$ mit $|A_n| \leq l_d$ und eine Menge $X = \{x_1, \dots, x_L\} \subset [n]$ existieren mit

$$\mathcal{N}_d^{B_n - A_n}(x_i) \cap \mathcal{N}_d^{B_n - A_n}(x_j) = \emptyset \text{ für alle } i \neq j \in \{1, \dots, L\}.$$

Eine built-in Relation ist also unendlich teilbar, wenn man durch Löschung von maximal l_d Knoten beliebig viele disjunkte d -Umgebungen finden kann.

Definition 4.2

- a) Eine built-in Relation \mathbf{B} heißt **endlos**, falls ein $l \in \mathbb{N}$ existiert, so daß für alle $D \in \mathbb{N}$ ein $n \in \mathbb{N}$ und eine Menge $A_n \subset [n]$ mit $|A_n| \leq l$ existieren derart, daß

$$\text{diam}(B_n - A_n) \geq D.$$

Dabei ist $\text{diam}(G)$ der Durchmesser eines Graphen G , d.h. der größte endliche Abstand zweier Knoten in G .

Eine built-in Relation \mathbf{B} ist also endlos, wenn der Durchmesser von B_n durch Entfernung von maximal l Knoten beliebig groß werden kann.

- b) Eine built-in Relation \mathbf{B} heißt **trennbar**, falls ein $l \in \mathbb{N}$ existiert, so daß für alle $N \in \mathbb{N}$ ein $n \in \mathbb{N}$ und eine Menge $A_n \subset [n]$ mit $|A_n| \leq l$ existieren derart, daß

$$|\{\text{Zusammenhangskomponenten in } B_n - A_n\}| \geq N.$$

Eine built-in Relation \mathbf{B} ist also trennbar, wenn B_n durch Entfernen von maximal l Knoten in beliebig viele Zusammenhangskomponenten getrennt wird.

Eine ganze Reihe von built-in Relationen sind unendlich teilbar. Dazu der folgende Satz.

Satz 4.3

- a) x -built-in Relationen sind unendlich teilbar.
- b) Zerlegbare Relationen sind unendlich teilbar.
- c) Built-in Relationen mit beschränkter Baumweite sind unendlich teilbar.
- d) Built-in Relationen mit lokal beschränkter Baumweite sind unendlich teilbar.
- e) Built-in Relationen mit verbotenen Minor K_k sind unendlich teilbar.
- f) Built-in Relationen mit lokal verbotenen Minor K_k sind unendlich teilbar.
- g) Endlose built-in Relationen sind unendlich teilbar.
- h) Trennbare built-in Relationen sind unendlich teilbar.

Beweis:

zu a) siehe Satz 4.10.

zu b) folgt direkt aus der Definition von Zerlegbarkeit.

zu c) folgt aus b) und Ergebnissen von Kreidler (siehe [17]).

zu d) wird in diesem Kapitel bewiesen.

zu e) folgt aus b) und Ergebnissen von Kreidler (siehe [17]). Direkter Beweis im Abschnitt 4.3.2.

zu f) wird in diesem Kapitel bewiesen.

zu g) Gibt es in $B_n - A_n$ einen Weg der Länge $2dL$, so können *leicht* L Knoten auf diesem Weg gefunden werden, die paarweise disjunkte d -Umgebungen in $B_n - A_n$ haben.

zu h) leicht zu zeigen.

Bemerkung 4.4

Im weiteren Verlauf dieses Kapitels wird unter anderem gezeigt, daß unendlich teilbare built-in Relationen schwach sind. Insbesondere sind endlose und trennbare built-in Relationen schwach. Man sieht, daß starke built-in Relationen in gewisser Weise dicht sein müssen. Das bedeutet, daß in einer starken built-in Relation \mathbf{B}

1. Knoten innerhalb einer Zusammenhangskomponente Maximalabstand $d_{\mathbf{B}}$ haben. Dabei hängt $d_{\mathbf{B}}$ nur von \mathbf{B} , aber nicht von n ab.
2. maximal $z_{\mathbf{B}}$ viele Zusammenhangskomponenten existieren. Dabei hängt $z_{\mathbf{B}}$ wiederum nur von \mathbf{B} und nicht von n ab.

4.2 Monadic NP + unendlich teilbare built-in Relationen

In diesem Abschnitt wird mit Hilfe einer Verallgemeinerung von Schwentick's Erweiterungssatz gezeigt, daß Graphzusammenhang nicht in monadicNP + unendlich teilbare built-in Relation ausgedrückt werden kann. Die Verallgemeinerung besteht darin eine **konstante** Anzahl von Knoten konstant zu spielen (unten werden dies genau die Ausnahmeknoten A_n aus der Definition von unendlich teilbar sein). Ansonsten entspricht der Beweis dem von Schwentick geführten *Kreis vs. Doppelkreis* Beweis für built-in Relationen mit Grad $n^{o(1)}$ (siehe [22]).

Satz 4.5 (schwacher Erweiterungssatz, Schwentick)

Seien

- $r > 0$ und $S=(E,B)$ eine Signatur.
- $\mathcal{A}, \bar{\mathcal{A}}$ endliche S -Strukturen mit gleichem Universum $[n]$.
- H bzw. \bar{H} Teilstrukturen von \mathcal{A} bzw. $\bar{\mathcal{A}}$.
- $\delta = \bar{\delta} : [n] \times [n] \rightarrow \mathbb{N}$ Distanzfunktionen.
- $\delta_H := \delta(x, H), \quad \delta_{\bar{H}} := \delta(x, \bar{H})$ und schließlich
- $A := \{a_1, \dots, a_l\} \subset [n]$ die Menge der Knoten mit

$$\begin{aligned}
 & (\delta(a_i, H) > 2^r \text{ und } \text{dist}_{\mathcal{A}}(a_i, H) \leq 2^r) \\
 & \text{oder} \\
 & (\delta(a_i, \bar{H}) > 2^r \text{ und } \text{dist}_{\bar{\mathcal{A}}}(a_i, \bar{H}) \leq 2^r).
 \end{aligned}$$

Dabei ist $\text{dist}_{\mathcal{A}}$ ($\text{dist}_{\bar{\mathcal{A}}}$) die kanonische Distanzfunktion auf \mathcal{A} ($\bar{\mathcal{A}}$).

Duplikator hat eine Gewinnstrategie im r -Runden FO-Spiel auf \mathcal{A} , $\bar{\mathcal{A}}$, wenn:

- (i) Die Strukturen $\mathcal{A} \setminus H$ und $\bar{\mathcal{A}} \setminus \bar{H}$ identisch sind und
- (ii) Duplikator eine Gewinnstrategie im r -Runden FO-Spiel auf

$$(\mathcal{A} \downarrow \mathcal{N}_{2^r}^{\delta}(H), \delta_H, a_1, \dots, a_l) \text{ und } (\bar{\mathcal{A}} \downarrow \mathcal{N}_{2^r}^{\bar{\delta}}(\bar{H}), \delta_{\bar{H}}, a_1, \dots, a_l)$$

hat.

Beweis: siehe [21].

Satz 4.6

Graphzusammenhang ist nicht ausdrückbar in monadic NP + unendlich teilbare built-in Relation.

Beweis:

Seien $c, r \in \mathbb{N}$ und \mathbf{B} eine unendlich teilbare built-in Relation.

Seien $A_n \subset \{1, \dots, n\}$, $|A_n| \leq l_{2^r}$ und $\{x_1, \dots, x_L\} \subset \{1, \dots, n\}$ eine Knotenmenge mit paarweise disjunkten 2^r -Umgebungen in $B_n - A_n$. Diese Menge existiert da \mathbf{B} unendlich teilbar ist.

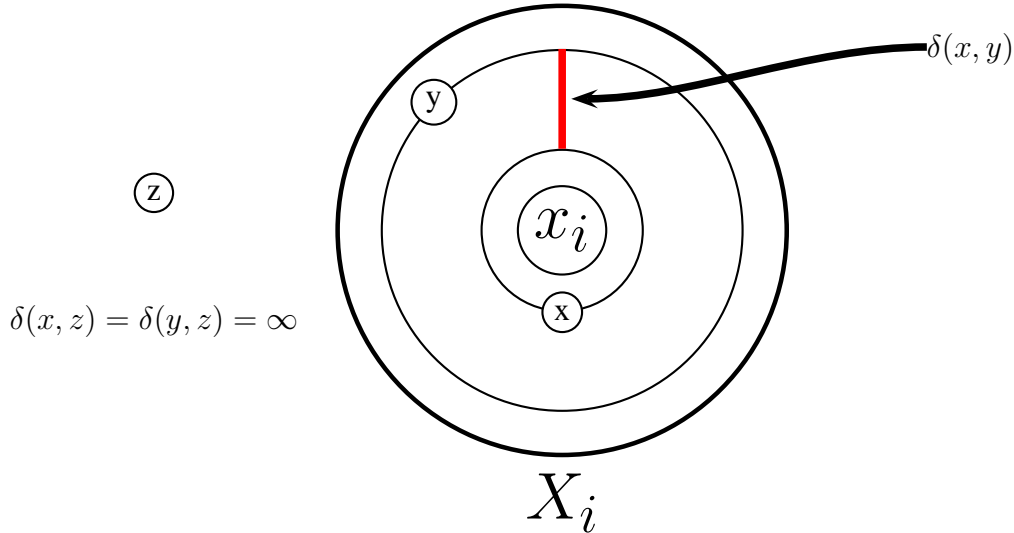
Konstruktion von G_0 :

Sei $X_i := \mathcal{N}_{2^r}^{B_n - A_n}(x_i)$.

Durch

$$\delta(x, y) := \begin{cases} |\text{dist}_{B_n - A_n}(x_i, x) - \text{dist}_{B_n - A_n}(x_i, y)|, \\ \text{falls } x, y \in X_i \text{ für } i \in \{1, \dots, L\} \\ \infty, \text{ sonst} \end{cases}$$

wird eine Distanzfunktion auf $[n]$ definiert.



Mit \mathcal{N}^δ werden im Folgenden Umgebungen in Bezug auf die Distanzfunktion δ bezeichnet.

Ketten:

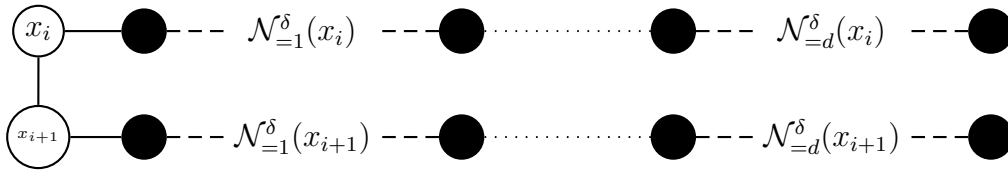
Es werden nun (analog zum Beweis von Schwentick) zunächst die Knoten der d -Umgebungen X_i zu Ketten *aufgereiht*, welche im Anschluß zu einem Kreis miteinander verbunden werden.

Mit $\mathcal{N}_{=d}^\delta(x_i) := \mathcal{N}_d^\delta(x_i) \setminus \mathcal{N}_{d-1}^\delta(x_i)$ wird die Menge der Knoten mit δ -Abstand d von x_i bezeichnet.

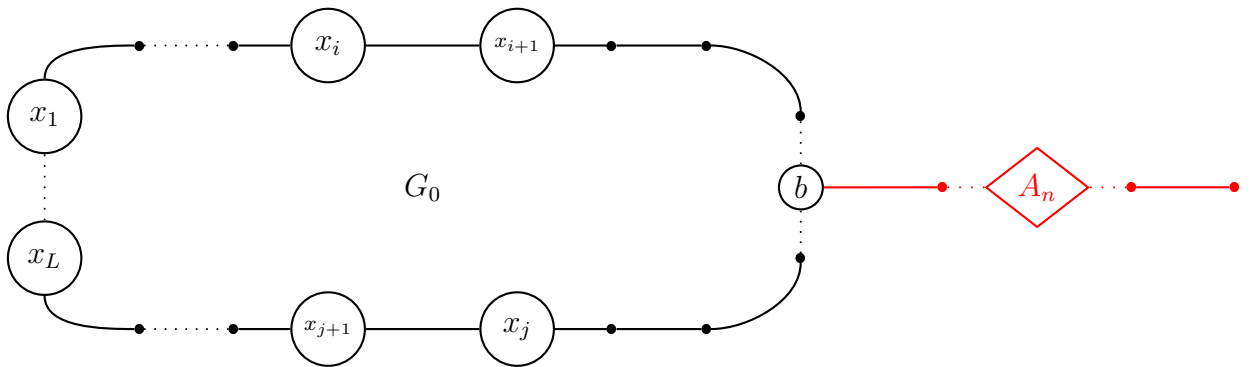
Die Kette K_i entsteht, indem der Reihe nach die Knoten $\mathcal{N}_{=1}^\delta(x_i), \mathcal{N}_{=2}^\delta(x_i), \dots, \mathcal{N}_{=2r}^\delta(x_i)$ an den Knoten x_i angehängt werden.



Dadurch erhält man L Ketten K_1, \dots, K_L . Jeweils zwei aufeinanderfolgende Ketten K_i, K_{i+1} werden zu Doppelketten $dK_{i,i+1}$ verbunden, indem der Knoten x_i mit dem Knoten x_{i+1} verbunden wird (dabei sei oBdA L gerade).



G_0 entsteht indem man alle Doppelketten zu einem Kreis miteinander verbindet und die verbleibenden Knoten (die in keiner 2^r -Umgebung X_i liegen) zu einer Kette aufreht und mit einem festen Knoten b verbindet.



Offensichtlich ist G_0 zusammenhängend. Spoiler färbt nun G_0 mit c Farben F_1^0, \dots, F_c^0 .

Konstruktion von G_1 :

Sei $S = (E, B, \delta, F_1, \dots, F_c, x, c_1, \dots, c_l)$ eine Signatur.

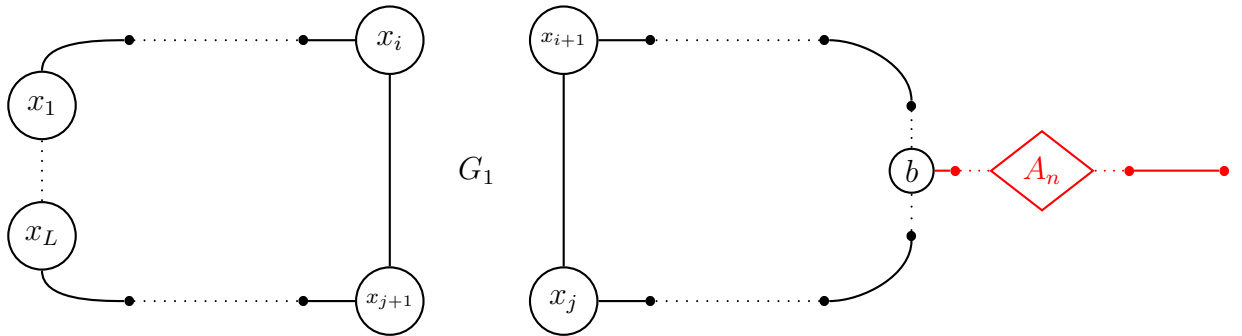
Es existieren nur M verschiedene r -Typen bzgl. der Signatur S .

Sei $X_m^+ := (G_0 \downarrow X_m \cup A_n, F_1^0, \dots, F_c^0, x_m, a_1, \dots, a_l)$ die 2^r -Umgebung von x_m erweitert um Spoilers Färbung F_1^0, \dots, F_c^0 und die Knoten x_m und $A_n = \{a_1, \dots, a_l\}$ als Konstanten.

Falls L groß genug gewählt wird, existieren $i \neq j$ mit

$$(\tau^r(X_i^+), \tau^r(X_{i+1}^+)) = (\tau^r(X_j^+), \tau^r(X_{j+1}^+)) \quad (\#)$$

G_1 entsteht aus G_0 indem die Kanten $(x_i, x_{i+1}), (x_j, x_{j+1})$ gelöscht werden und die Kanten $(x_i, x_{j+1}), (x_j, x_{i+1})$ hinzugefügt werden.



Offensichtlich ist G_1 nicht zusammenhängend.

Duplikator färbt G_1 identisch zur Spoilers Färbung von G_0 .

Mittels des schwachen Erweiterungssatzes 4.5 sieht man, daß Duplikator das r -Runden Spiel auf G_0, G_1 gewinnt.

Sei dazu

- $\mathcal{A} = G_0, \bar{\mathcal{A}} = G_1,$
- $H = \{x_i, x_{i+1}, x_j, x_{j+1}\} = \bar{H}$ (dabei enthält H die entsprechenden Knoten aus G_0, \bar{H} die Knoten aus G_1),
- $\delta = \bar{\delta}$ wie oben definiert und
- $A = A_n.$

Bedingung (i) ist offensichtlich. Bedingung (ii) folgt leicht aus (#). □

4.3 Anwendungen

In diesem Abschnitt wird bewiesen, daß folgende built-in Relationen unendlich teilbar sind:

- x-built-in Relationen
- Relationen mit verbotenen Minor K_k
- Relationen mit lokal verbotenen vollständigen Minor
- Relationen mit lokal beschränkter Baumweite

4.3.1 x-built-in Relationen

Zunächst wird eine Klasse von built-in Relationen betrachtet, die durch *Gradbedingungen* charakterisiert ist. Das bisher *stärkste* Ergebnis in dieser Richtung ist der folgende Satz von Schwentick:

Satz 4.7 (Schwentick 95)

Sei $f : \mathbb{N} \rightarrow \mathbb{N}$ mit $f(n) = n^{o(1)}$. Graphzusammenhang ist nicht ausdrückbar in monadic NP + endlich viele built-in Relationen mit Maximalgrad $f(n)$.

Bemerkung: Die Beweisidee dieses Satzes entspricht der Beweisidee von Satz 4.2. Im Verlauf des Beweises wird insbesondere gezeigt, daß built-in Relationen mit obiger Gradbedingung unendlich teilbar sind.

Eine genaue Betrachtung des Beweises zeigt, daß obige Gradbedingung abgeschwächt werden kann. Dazu die folgende Definition:

Definition 4.8

Eine built-in Relation $\mathbf{B} = (B_n)_{n \in \mathbb{N}}$ heißt *x-built-in Relation*, falls sie der folgenden Bedingung genügt:

Es existiert ein $\gamma < 1$ und ein $n_0 \in \mathbb{N}$, so daß für alle $n \geq n_0$ ein ϵ_n, δ_n mit $\epsilon_n + \delta_n < \gamma$ existiert derart, daß folgendes gilt:

1. Es existiert ein $f : \mathbb{N} \rightarrow \mathbb{N}$ mit $f(n) = n^{o(1)}$ so, daß mindestens $n - n^{\delta_n}$ Knoten von B_n Maximalgrad $f(n)$ haben.

Dieses f wird im Folgenden **Gradfunktion** von \mathbf{B} genannt.

2. Alle Knoten in B_n haben Grad $\leq n^{\epsilon_n}$.

Bemerkung 4.9

Es ist ein ungelöstes Problem, ob Graphzusammenhang ausdrückbar ist in monadic NP + BIT. Insbesondere ist BIT eine starke built-in Relation. Eine Betrachtung der Gradstruktur von BIT_n (BIT Relation auf $[n]$) liefert folgende Aufteilung (OBdA sei $n = 2^m$):

- Die Knoten $\{1, \dots, \lg n\}$ haben Grad $\frac{n}{2}$.
- Die Knoten $\{\lg n + 1, \dots, n\}$ haben Grad $\leq \lg n$.

Definiert man δ_n und ϵ_n mittels

- $\lg n = n^{\delta_n} (\Rightarrow \delta_n \downarrow 0)$
- $\frac{n}{2} = n^{\epsilon_n} (\Rightarrow \epsilon_n \uparrow 1)$

so sieht man, daß in BIT_n (maximal) n^{δ_n} Knoten mit $\text{Grad} \geq n^{\delta_n}$ und $\leq n^{\epsilon_n}$ existieren. Alle anderen Knoten haben $\text{Grad} \leq \lg n$. Man sieht leicht, daß $\delta_n + \epsilon_n \downarrow 1$.

Bezüglich Gradbetrachtungen liegen starke built-in Relationen (BIT) und schwache built-in Relationen (x-built-in) also sehr dicht beieinander.

Es wird nun gezeigt, daß x-built-in Relationen unendlich teilbar sind.

Satz 4.10

Eine Menge B^1, \dots, B^m von x-built-in Relationen ist unendlich teilbar.

Dabei ist eine Relationenmenge R^1, \dots, R^m unendlich teilbar, wenn die durch

$$\forall n \in \mathbb{N}, \forall x, y \in [n] \quad R_n xy \iff \exists i \in \{1, \dots, m\} \text{ mit } R_n^i xy$$

definierte built-in Relation R unendlich teilbar ist.

Daraus ergibt sich direkt:

Korollar 4.11

Graphzusammenhang ist nicht ausdrückbar in monadic NP + endlich viele x-built-in Relationen.

Beweis von Satz 4.10:

Seien $B^1, \dots, B^m = \{(B_n^1)_{n \in \mathbb{N}}, \dots, (B_n^m)_{n \in \mathbb{N}}\}$ beliebige x-built-in Relationen.

Sei $f_i : \mathbb{N} \rightarrow \mathbb{N}$ die Gradfunktion von B^i . Dann ist durch $f := m \cdot \max(f_1, \dots, f_m)$ eine universelle Gradfunktion für B^1, \dots, B^m definiert. Insbesondere gilt $f = n^{o(1)}$.

Die Idee des Beweises ist zu zeigen, daß es eine *ausreichend* große Knotenmenge \mathcal{M} gibt, deren Knoten bezüglich B_n^1, \dots, B_n^m *ausreichend* weit weg von allen Knoten *großen* Grades sind. Innerhalb dieser Menge ist es *leicht* eine Knotenmenge zu finden, deren Knoten paarweise disjunkte d -Umgebungen haben. Dazu folgende Definition:

Definition 4.12

Ein Knoten $x \in [n]$ hat **großen Grad** bezüglich B_n^1, \dots, B_n^m , falls ein $i \in \{1, \dots, m\}$ existiert, so daß der B_n^i -Grad von x größer als $f(n)$ ist.

Folgendes Lemma präzisiert obige Beweisidee.

Lemma 4.13

Sei \mathcal{J}^n die Menge aller Knoten aus $[n]$ mit großen Grad bezüglich B_n^1, \dots, B_n^m . Dann gilt für alle $d \in \mathbb{N}$:

$$|\mathcal{N}_d(\mathcal{J}^n)| \leq n^{\gamma_n} \cdot f(n)^d \quad \text{mit } \gamma_n < 1.$$

Dabei ist $\mathcal{N}_d(\mathcal{J}^n) = \cup_{x \in \mathcal{J}^n} \mathcal{N}_d(x)$ die d -Umgebung von \mathcal{J}^n .

Beweis des Lemmas:

Induktion nach m .

$m = 1$: Nach Definition von x -built-in Relation existiert für alle $n \geq n_0$ ein ϵ_n, δ_n mit $\epsilon_n + \delta_n < 1$ derart, daß alle Knoten Grad kleiner n^{ϵ_n} haben und nur maximal n^{δ_n} Knoten mit Grad größer als $f(n)$ existieren. \mathcal{J}^n ist die Menge dieser Knoten. Da die d -Umgebung von \mathcal{J}^n maximal wird, falls die d -Umgebungen aller Punkte aus \mathcal{J}^n nicht überlappen, kann man für die Abschätzung annehmen, daß für alle $x, y \in \mathcal{J}^n, x \neq y$ gilt:

$$\mathcal{N}_d(x) \cap \mathcal{N}_d(y) = \emptyset.$$

Wie man leicht sieht, gilt dann:

$$|\mathcal{N}_d(x)| \leq 1 + n^{\epsilon_n}(1 + f(n) + f(n)^2 + \dots + f(n)^{d-1}) \leq n^{\epsilon_n} f(n)^d.$$

Damit folgt:

$$|\mathcal{N}_d(\mathcal{J}^n)| = |\cup_{x \in \mathcal{J}^n} \mathcal{N}_d(x)| \leq n^{\delta_n} n^{\epsilon_n} f(n)^d.$$

Dies beweist den Fall $m = 1$.

$m \rightarrow m + 1$: Sei $\mathcal{J}_{1, \dots, m-1}^n$ die Menge aller großen Knoten bezüglich $(B_1^n, \dots, B_{m-1}^n)$. Sei \mathcal{J}_m^n die Menge aller großen Knoten bezüglich B_m^n . Offensichtlich gilt:

$$\mathcal{J}^n = \mathcal{J}_{1, \dots, m-1}^n \cup \mathcal{J}_m^n$$

und $|\mathcal{N}_d(\mathcal{J}^n)|$ wird maximal, wenn $\mathcal{N}_d(\mathcal{J}_{1, \dots, m-1}^n) \cap \mathcal{N}_d(\mathcal{J}_m^n) = \emptyset$. In diesem Fall gilt:

$$\begin{aligned} |\mathcal{N}_d(\mathcal{J}^n)| &= |\mathcal{N}_d(\mathcal{J}_{1, \dots, m-1}^n)| + |\mathcal{N}_d(\mathcal{J}_m^n)| \stackrel{I.V.}{\leq} n^{\gamma_n^{m-1}} f(n)^d + n^{\delta_n} n^{\epsilon_n} f(n)^d \leq \\ &\leq 2n^{\max(\gamma_n^{m-1}, \epsilon_n + \delta_n)} f(n)^d \leq n^{\gamma_n} f(n)^d, \end{aligned}$$

wobei $\epsilon_n + \delta_n < 1$ und $\max(\gamma_n^{m-1}, \epsilon_n + \delta_n) < \gamma_n < 1$.

□

Da $f(n) = n^{o(1)}$, existieren ein $n_0 \in \mathbb{N}$ und $\alpha < 1$ derart, daß für alle $n \geq n_0$ gilt:

$$|\mathcal{N}_d(\mathcal{J}^n)| \leq n^\alpha.$$

Das bedeutet, daß mindestens $n - n^\alpha$ Knoten aus $[n]$ Mindestabstand d von allen Knoten mit großem Grad haben (also alle Knoten aus \mathcal{J}^n). Sei \mathcal{M} die Menge dieser Knoten. Für \mathcal{M} gilt also:

$$\mathcal{N}_{d-1}(\mathcal{M}) \cap \mathcal{J}^n = \emptyset.$$

Insbesondere haben alle Knoten aus $\mathcal{N}_{d-1}(\mathcal{M})$ Grad $\leq f(n)$.

Es wird nun induktiv gezeigt, daß in der Menge \mathcal{M} beliebig viele Knoten mit paarweise disjunkter $d_2 := \frac{d-1}{2}$ -Umgebung (bzgl. (B_n^1, \dots, B_n^m)) gefunden werden können. Da d beliebig gewählt werden kann, ist der Satz bewiesen.

Seien also x_1, \dots, x_L Knoten aus \mathcal{M} mit paarweiser disjunkter d_2 -Umgebung. Nach Definition von \mathcal{M} gilt:

$$|\mathcal{N}_{2d_2}(x_1, \dots, x_L)| \leq L \cdot (1 + f(n) + \dots + f(n)^{2d_2}) \leq L \cdot f(n)^{2 \cdot d_2 + 1}.$$

Falls also

$$L \cdot f(n)^{2d_2+1} \leq n - n^\alpha \leq |\mathcal{M}|, \quad (*)$$

existiert ein $x_{L+1} \in \mathcal{M}$ derart, daß die Knoten x_1, \dots, x_{L+1} paarweise disjunkte d_2 Umgebungen haben. Da $f(n) = n^{o(1)}$, gilt (*) für ausreichend großes n . \square

Bemerkung 4.14

Die entscheidende Eigenschaft von x -built-in Relationen, welche im obigen Beweis verwendet wurde, ist die Existenz einer ausreichend großen Knotenmenge \mathcal{M} derart, daß alle Knoten aus $\mathcal{N}_d(\mathcal{M})$ Maximalgrad $f(n) = n^{o(1)}$ haben.

Dies legt folgende Definition nahe:

Definition 4.15

Sei $\mathbf{B} = (B_n)_{n \in \mathbb{N}}$ eine built-in Relation.

- a) \mathbf{B} heißt (d, N) -**f-beschränkt**, falls ein $n_0 \in \mathbb{N}$ existiert, so daß für alle $n > n_0$ eine Teilmenge $\mathcal{M} \subset [n]$ existiert mit
 - $|\mathcal{M}| \geq N$.
 - Für alle $x \in \mathcal{N}_d(\mathcal{M})$ gilt $\text{grad}_{B_n}(x) \leq f(n)$.
- b) \mathbf{B} heißt **f-beschränkt**, wenn \mathbf{B} für alle $d, N \in \mathbb{N}$, (d, N) -f-beschränkt ist.

Folgende Sätze ergeben sich direkt aus dem Beweis von Satz 4.10 oder können leicht analog zu diesem gezeigt werden:

Satz 4.16

Sei $f(n) = n^{o(1)}$. Graphzusammenhang ist nicht ausdrückbar in monadic NP + f-beschränkte built-in Relationen.

Satz 4.17

- a) x -built-Relationen sind f-beschränkt, mit $f(n) = n^{o(1)}$.
- b) Seien B_1, \dots, B_m x -built-in Relationen. Dann ist das Relationentupel (B_1, \dots, B_m) f-beschränkt mit $f(n) = n^{o(1)}$.

4.3.2 Built-in Relationen mit verbotenem Minor K_k

In diesem Abschnitt wird gezeigt, daß built-in Relationen mit verbotenem Minor K_k unendlich teilbar sind. Damit folgt direkt, daß diese built-in Relationen monadic NP nicht ausreichend verstärken, um Graphzusammenhang auszudrücken. Dieses Ergebnis konnte Kreidler schon in [17] beweisen. Der dort vorgestellte Beweis ist *äußerst*

kompliziert und sehr lang. Die Kürze und Eleganz des folgenden Beweises ist ein Indiz für die Stärke der Eigenschaft *unendlich teilbar*.

Zunächst die notwendigen Definitionen im Zusammenhang mit verbotenen Minoren.

Definition 4.18

Sei G ein Graph mit Kantenrelation E .

- a) Der Graph G_{-x} entsteht aus G , indem der Knoten x gelöscht wird.
- b) Seien $x, y \in V_G$ mit Exy . Der Graph $G_{x=y}$ entsteht durch **Kontraktion** der Knoten x mit y . Das bedeutet, daß der Knoten y aus G gelöscht und der Knoten x zusätzlich mit allen Knoten verbunden wird, mit denen y verbunden war. Man bezeichnet diese Kontraktion auch mit $K(y \rightarrow x)$.

Definition 4.19

Sei G ein Graph.

- a) G hat einen **Minor** K_k , falls durch eine Folge von Kontraktionen und Löschungen ein Graph \overline{G} gebildet werden kann, der einen K_k als Teilgraph hat.
- b) Sei $X := \{x_1, \dots, x_k\} \subset V_G$ eine Knotenmenge aus G . G hat einen Minor K_k mit **Minorknoten** X , falls eine Folge von Kontraktionen (K_i) existiert derart, daß
 - Keine Kontraktion K_i kontrahiert zwei Knoten aus X .
 - Für jede Kontraktion K_i , die einen Knoten $x \in X$ mit einem Knoten $y \in V_G \setminus X$ kontrahiert, gilt $K_i = K(y \rightarrow x)$.
 - Nach Durchführung der Kontraktionen K_i bilden die Knoten aus X einen K_k .

Die Menge X nennt man eine **Minormenge** von G .

Nun noch einige bekannte Lemmata aus der Graphentheorie (auf die Beweise wird hier verzichtet):

Lemma 4.20

Der vollständige bipartite Graph $K_{k-1, k-1}$ hat einen Minor K_k .

Lemma 4.21 (Ramsey)

Für jedes $N \in \mathbb{N}$ existiert ein $n(N)$ derart, daß jeder Graph G mit $|V_G| \geq n(N)$ einen K_N oder einen \overline{K}_N als Teilgraphen enthält.

Der Beweis des folgenden Satzes ist Inhalt dieses Abschnittes.

Satz 4.22

Built-in Relationen mit verbotenem Minor K_k ($k \in \mathbb{N}$) sind unendlich teilbar.

Damit ergibt sich sofort

Korollar 4.23

Graphzusammenhang ist nicht ausdrückbar in monadic NP + eine built-in Relation mit verbotenem K_k .

Bemerkung 4.24

Satz 4.22 läßt sich auch als graphentheoretische Aussage formulieren. Sei dazu G ein Graph mit verbotenem Minor K_k . Für $d \in \mathbb{N}$ findet man ein $l_d \in \mathbb{N}$ derart, daß für alle $L \in \mathbb{N}$ ein $n(L)$ existiert, so daß falls $|V_G| \geq n(L)$, man eine Menge A_G mit $|A_G| \leq l_d$ und Knoten $x_1, \dots, x_L \in V_G$ mit paarweise disjunkten d -Umgebungen in $G - A_G$ findet.

Beweis von Satz 4.22:

Induktion über die Umgebungsweite d .

$d = 0$: klar.

$d \rightarrow d + 1$: Gesucht ist ein l_{d+1} , so daß für beliebiges $L \in \mathbb{N}$ (mindestens) ein $n \in \mathbb{N}$, eine Menge $A_n^{d+1} \subset [n]$ mit $|A_n^{d+1}| \leq l_{d+1}$ und eine Menge $X_{d+1} = \{x_1, \dots, x_L\}$ existieren derart, daß in $B_n - A_n^{d+1}$ die Knoten aus X_{d+1} paarweise disjunkte $(d + 1)$ -Umgebungen haben.

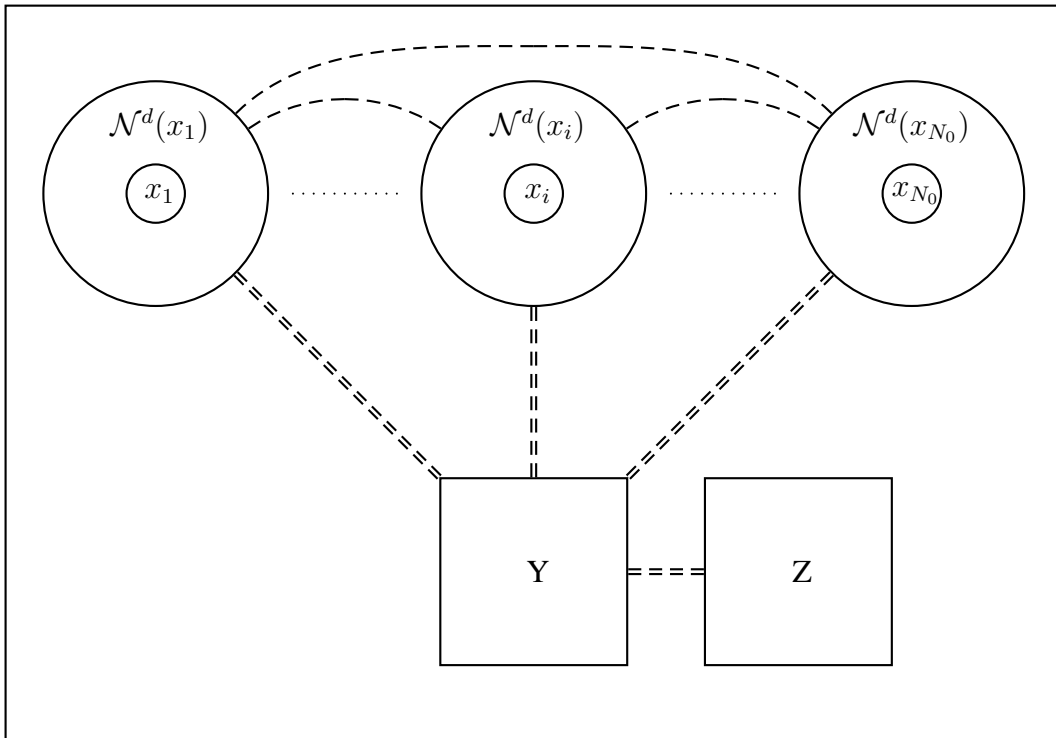
Nach Induktionsvoraussetzung existiert ein l_d , so daß für $N_0 \in \mathbb{N}$ beliebig, (mindestens) ein $n \in \mathbb{N}$, eine Menge $A_n^d \subset [n]$ mit $|A_n^d| \leq l_d$ und eine Menge $X_d = \{x_1, \dots, x_{N_0}\} \subset [n]$ existieren derart, daß in $B_n - A_n^d$ die Knoten aus X_d paarweise disjunkte d -Umgebung haben.

Die Knotenmenge von $B_n - A_n^d$ kann durch

$$V_{B_n - A_n^d} = \mathcal{N}_d^{B_n - A_n^d}(X_d) \dot{\cup} Y \dot{\cup} Z$$

disjunkt zerlegt werden. Dabei sei Y die Menge der Knoten, die Abstand $d + 1$ zu mindestens einem Knoten aus X_d haben, aber in keiner d -Umgebung der Knoten aus X_d liegen.

Folgende Skizze verdeutlicht die Struktur von $B_n - A_n^d$. Gestrichelte Linien deuten mögliche Verbindungen zwischen Knotenmengen an.



Im Verlauf des Beweises wird gezeigt, daß es eine große Teilmenge $X_{d+1} \subset X_d$ gibt, mit $|X_{d+1}| \geq L$ derart, daß die Knoten aus X_{d+1} paarweise disjunkte $(d+1)$ -Umgebungen in $B_n - A_n^{d+1}$ haben. Dabei ist A_n^{d+1} eine Obermenge von A_n^d mit $|A_n^{d+1}| = |A_n^d| + (k-1)$. Mit $l_{d+1} = l_d + (k-1)$ ist der Satz bewiesen.

Es stellt sich daher die Frage, unter welchen Bedingungen zwei Knoten x_i, x_j aus X_d überlappende $(d+1)$ -Umgebungen in $B_n - A_n^d$ haben. Dazu die folgende Definition und das folgende Lemma:

Definition 4.25

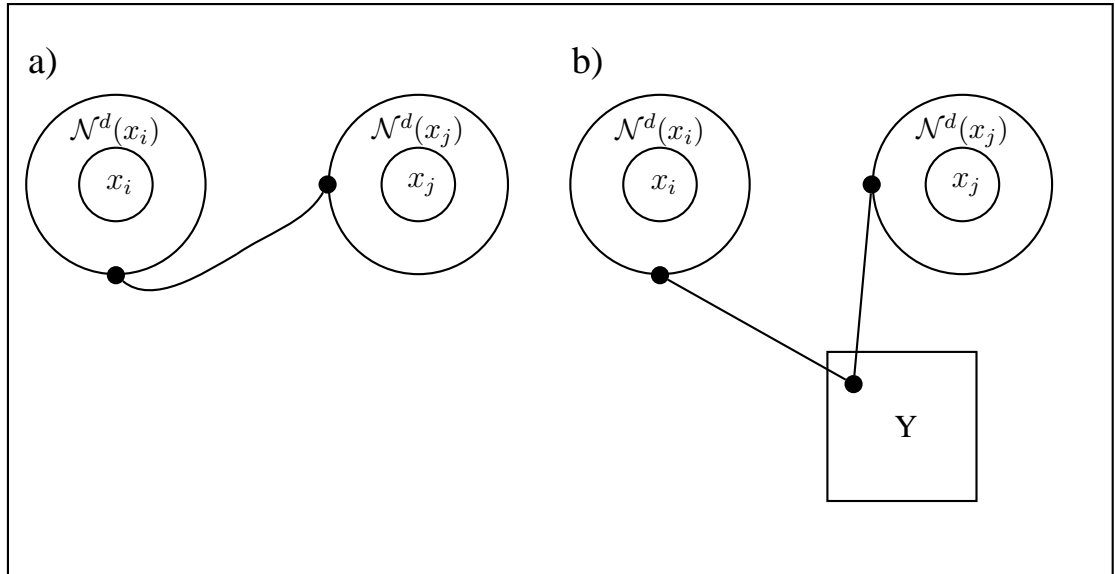
Zwei Knoten $x_i, x_j \in X_d$ sind **schwach verbunden**, wenn x_i, x_j in $B_n - A_n^d$ überlappende $(d+1)$ -Umgebungen haben.

Die Knoten sind **stark getrennt**, wenn sie nicht schwach verbunden sind.

Lemma 4.26

Zwei Knoten $x_i, x_j \in X_d$ sind genau dann schwach verbunden, wenn eine der beiden folgenden Bedingungen erfüllt ist:

- a) Zwei Randpunkte aus $\mathcal{N}_d(x_i)$ und $\mathcal{N}_d(x_j)$ sind miteinander verbunden.
- b) Zwei Randpunkte aus $\mathcal{N}_d(x_i)$ und $\mathcal{N}_d(x_j)$ sind mit einem (gemeinsamen) Punkt $y \in Y$ verbunden.



In fünf Schritten wird der Graph $B_n - A_n^d$ vereinfacht ohne dabei *relevante* Informationen zu verlieren:

- 1.Schritt: Für alle $i = 1, \dots, N$ wird die d -Umgebung von x_i in den Knoten x_i kontrahiert.
- 2.Schritt: Alle Knoten, deren Abstand zu allen $x_i \in X_d$ größer als 1 ist (dies sind gerade die Knoten aus Z), werden gelöscht.

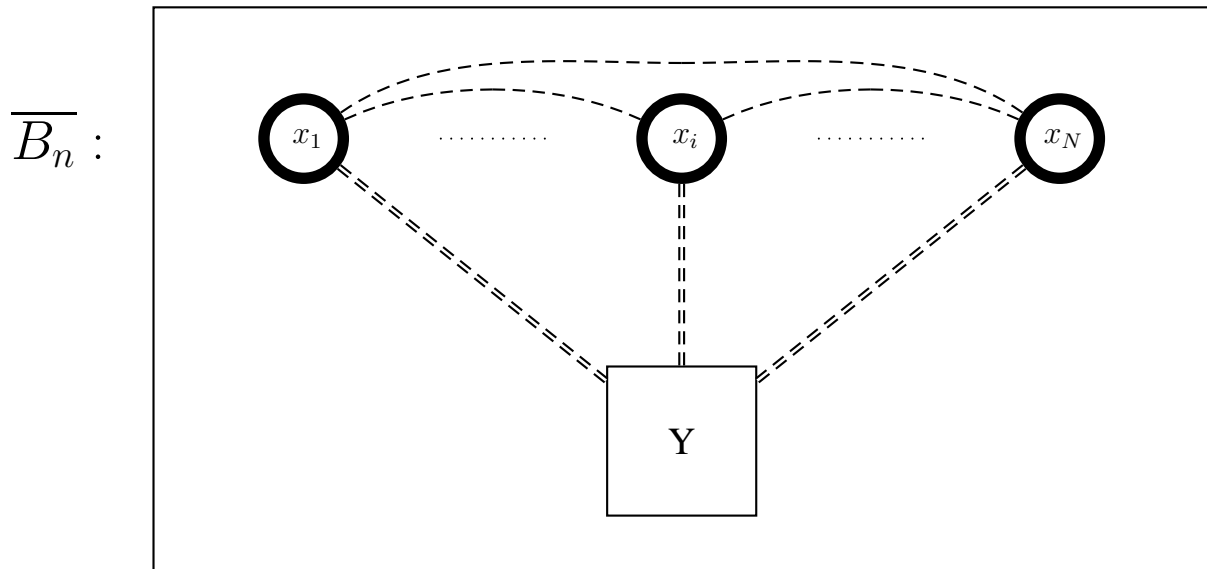
Bemerkung: Diese gelöschten Knoten haben keinen Einfluß auf die $(d + 1)$ -Umgebung der Knoten aus X_d .

- 3.Schritt: Alle Kanten zwischen Knoten in Y werden gelöscht.

Bemerkung: Der bisher erzeugte Graph wird mit $\overline{B_n}$ bezeichnet. Die Knotenmenge von $\overline{B_n}$ wird durch

$$V_{\overline{B_n}} = X_d \dot{\cup} Y$$

disjunkt zerlegt. Dabei ist Y in $\overline{B_n}$ eine unabhängige Menge, d.h. es gibt keine Kanten zwischen Knoten in Y .

**Lemma 4.27**

Falls \overline{B}_n einen Minor K_k hat, dann auch B_n .

Beweis:

\overline{B}_n entsteht aus B_n durch Löschung von Kanten und Knoten und durch Kontraktion von Knoten. □

Folgendes Lemma folgt direkt aus der Konstruktion und Lemma 4.26:

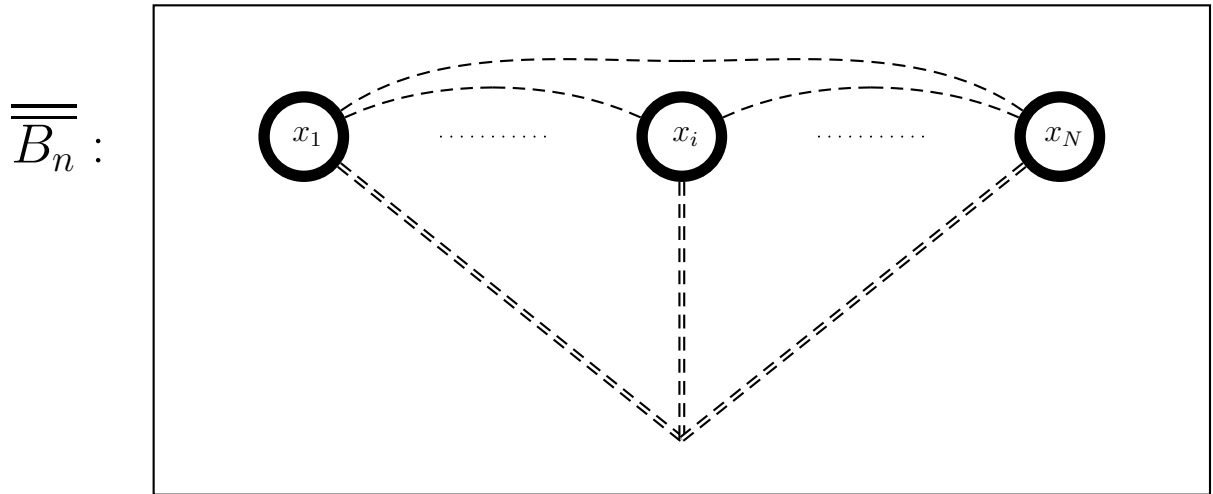
Lemma 4.28

Zwei Knoten $x_i, x_j \in X_d$ sind genau dann schwach verbunden, wenn

- x_i und x_j in \overline{B}_n verbunden sind, oder
- ein $y \in Y$ existiert, mit $\overline{B}_n(x_i, y)$ und $\overline{B}_n(y, x_j)$.

4.Schritt: Die Knoten $x_i, x_j \in X_d$ werden miteinander verbunden, falls sie über einen Knoten aus Y schwach verbunden sind.

5.Schritt: Alle Knoten aus Y werden gelöscht. Der so erzeugte Graph wird mit $\overline{\overline{B}_n}$ bezeichnet. Die Knotenmenge dieses Graphen ist X_d .



Folgendes Lemma folgt direkt aus der Konstruktion:

Lemma 4.29

Zwei Knoten $x_i, x_j \in X_d$ sind genau dann schwach verbunden, wenn sie in $\overline{\overline{B_n}}$ miteinander verbunden sind.

Nach Lemma 4.21 existiert bei geeigneter Wahl von N_0 , für $N_1 \in \mathbb{N}$ beliebig, in $\overline{\overline{B_n}}$ ein Teilgraph K_{N_1} oder ein Teilgraph $\overline{K_{N_1}}$. Diese beiden Fälle werden nun getrennt betrachtet:

1.Fall: Es existiert ein $\overline{K_{N_1}}$.

Nach Lemma 4.29 sind dann mindestens N_1 Knoten aus X_d paarweise nicht schwach verbunden (stark getrennt). Das bedeutet, daß diese Knoten paarweise disjunkte $(d+1)$ -Umgebungen in $B_n - A_n^d$ haben. Mit $N_1 = L$ ist der Schritt gezeigt.

2.Fall: Es existiert ein K_{N_1} .

Sei $X_d^1 := \{x_{11}, \dots, x_{1N_1}\} \subset X_d$ eine Knotenmenge, die in $\overline{\overline{B_n}}$ einen K_{N_1} bildet.

Die Knoten in X_d^1 sind also paarweise schwach verbunden. Es wird sich aber zeigen, daß durch Löschung weniger Knoten aus Y eine große Menge von Knoten in X_d^1 paarweise stark getrennt werden kann.

Sei $Y^1 \subset Y$ die Menge der Knoten aus Y , die in $\overline{\overline{B_n}}$ mit mindestens einem Knoten aus X_d^1 benachbart sind. Sei $B_1 := \overline{\overline{B_n}} \downarrow (X_d^1 \cup Y^1)$.

Der Kern des Beweises besteht in der Konstruktion von absteigenden Folgen $X_d^1 \supset X_d^2 \supset \dots \supset X_d^L$, $Y^1 \supset Y^2 \supset \dots \supset Y^L$ und Mengen

$M^X = \{m_1, \dots, m_L\}$, $M^Y = M_1 \cup \dots \cup M_L$ mit den im folgendem Schaubild dargestellten Eigenschaften (dabei werden Verbindungen in $\overline{B_n}$ betrachtet).

X_d^1	Y^1	
\cup	\cup	
X_d^2	Y^2	$m_1 \in X_d^1 \setminus X_d^2 \quad M_1 \subset Y^1$
\vdots	\vdots	1) Jeder Knoten $x \in X_d^2$ ist nicht mit m_1 verbunden.
\vdots	\vdots	2) Jeder Knoten $x \in X_d^2$ ist nur über Knoten aus M_1
\cup	\cup	direkt mit m_1 verbunden.
\vdots	\vdots	3) Jeder Knoten $x \in X_d^2$ ist mit jedem Knoten
\vdots	\vdots	$y \in M_1$ verbunden.
X_d^i	Y^i	$m_{i-1} \in X_d^{i-1} \setminus X_d^i \quad M_{i-1} \subset Y^{i-1}$
		0) $m_{i-1} \notin \{m_1, \dots, m_{i-2}\}$
\vdots	\vdots	1) Jeder Knoten $x \in X_d^i$ ist nicht mit Knoten aus
\vdots	\vdots	$\{m_1, \dots, m_{i-1}\}$ verbunden.
		2) Jeder Knoten $x \in X_d^i$ ist nur über Knoten aus $M_1 \cup \dots \cup M_{i-1}$
\downarrow	L-mal	direkt mit Knoten aus m_1, \dots, m_{i-1} verbunden.
		3) Jeder Knoten $x \in X_d^i$ ist mit allen Knoten aus
		$y \in M_1 \cup \dots \cup M_{i-1}$ verbunden.

Diese Prozedur liefert folgende Mengen:

- $M^X = \{m_1, \dots, m_L\}$
- $M^Y = M_1 \cup \dots \cup M_L$
- X_d^{L+1} mit $|X_d^{L+1}| \geq k - 1$

mit Eigenschaften

- (E0) $|M^X| = L$
 (E1) Die Menge M^X ist eine unabhängige Menge in $\overline{B_n}$.
 (E2) Die Knoten aus M^X sind nur über Knoten aus M^Y schwach verbunden.

(E3) Alle Knoten in X_d^{L+1} sind mit allen Knoten aus M^Y verbunden.

Bemerkung 4.30

- 1) Die Bedingung $|X_d^{L+1}| \geq k - 1$ liefert rekursiv den Mindestwert für N_0 .
- 2) Die Eigenschaften E0-E3 ergeben sich direkt aus den Eigenschaften, die dem obigen Schaubild entnommen werden können.

Nun kann der Beweis abgeschlossen werden. Es müssen die folgenden zwei Fälle betrachtet werden:

1. $|M^Y| \geq k - 1$: Die Knoten aus M^Y sind nach E3 mit allen Knoten aus X_d^{L+1} verbunden. Da $|M^Y|, |X_d^{L+1}| \geq k - 1$, folgt die Existenz eines $K_{k-1, k-1}$ in $\overline{B_n}$. Widerspruch.
2. $|M^Y| \leq k - 1$: Indem man die $\leq k - 1$ Knoten aus M^Y entfernt, werden die Knoten aus M^X paarweise stark getrennt.
Mit $l_{d+1} := l_d + k - 1$ folgt die Behauptung. \square

Abschließend wird die Existenz der Mengen aus obigen Schaubild bewiesen.

Die Mengen X_d^1 und Y^1 wurden oben definiert. Die Mengen X_d^2, Y^2, M_1 und der Knoten m_1 ergeben sich aus dem folgenden Konstruktionsschritt. Alle weiteren Mengen ergeben sich durch L -faches Wiederholen dieser Konstruktion.

Sei $Min_1^X = \{m_{11}, \dots, m_{1l}\} \subset X_d^1$ eine maximale Minormenge in $\overline{B_n}$ derart, daß eine Kontraktionsfolge $K = (K_i)$ existiert, mit folgenden Eigenschaften:

- Jede Kontraktion K_i kontrahiert einen Knoten $m \in Min_1^X$ mit einem Knoten $y \in Y^1$.
- Jedes $m \in Min_1^X$ kommt in höchstens $(l - 1)$ Kontraktionen vor.

Sei $Min_1^Y \subset Y^1$ die Menge der Knoten aus Y^1 , welche in der Kontraktionsfolge (K_i) vorkommen. Wegen obigen Bedingungen gilt $|Min_1^Y| \leq l(l - 1)$.

Da $\overline{B_n}$ verbotenen Minor K_k hat, folgt sofort $l < k$.

Es gilt nun das folgende Lemma:

Lemma 4.31

Jedem $x \in X_d^1 \setminus Min_1^X$ kann ein Tupel $T(x) = (m_x, M_x)$ zugeordnet werden, mit

- a) $m_x \in Min_1^X, M_x \subset Min_1^Y$.
- b) x und m_x sind in B_1 nicht miteinander verbunden.
- c) x ist genau über die Knoten aus M_x direkt mit m_x verbunden.

Beweis:

Falls ein $x \in X_d^1 \setminus \text{Min}_1^X$ existiert, dem kein derartiges Tupel zugeordnet werden kann, dann gilt für alle $\bar{m} \in \text{Min}_1^X$ (da alle Knoten aus X_d^1 schwach miteinander verbunden sind)

- a) x und \bar{m} sind in B_1 miteinander verbunden, oder
- b) es existiert ein $y_{\bar{m}} \in Y^1 \setminus \text{Min}_1^Y$, welches x und \bar{m} miteinander verbindet.

Damit hat die Struktur $B_1 \downarrow (\text{Min}_1^X \cup \{x\} \cup \text{Min}_1^Y \cup_{\bar{m} \in \text{Min}_1^X} y_{\bar{m}})$ einen Minor K_{l+1} mit Minorknoten $\text{Min}_1^X \cup \{x\}$. Dies ist ein Widerspruch zur Maximalität von Min_1^X . \square

Jedem Knoten $x \in X_d^1 \setminus \text{Min}_1^X$ kann also ein derartiges Tupel $T(x) = (m_x, M_x)$ zugeordnet werden (diese Zuordnung muß nicht eindeutig sein). Da es nur maximal $l \cdot 2^{l(l-1)}$ verschiedene solche Tupel gibt, findet man für jedes $N_2 \in \mathbb{N}$, bei entsprechender Wahl von N_1 , eine Teilmenge $X_d^2 \subset (X_d^1 \setminus \text{Min}_1^X)$ mit

- Für $x_1, x_2 \in X_d^2$ gilt $T(x_1) = T(x_2) = (m_1, M_1)$.
- $|X_d^2| \geq N_2$.

Sei schließlich $Y^2 \subset Y^1$ die Menge der Knoten aus Y^1 , die in B_1 mit mindestens einem Knoten aus X_d^2 verbunden sind. Man sieht leicht, daß

1. die so konstruierten X_d^2, Y_d^2, M_1, m_1 die notwendigen Eigenschaften aus obigen Schaubild haben.
2. dieses Verfahren sich induktiv fortsetzen läßt.

Damit ist der Beweis von Satz 4.22 abgeschlossen. \square

Bemerkung 4.32

Da Graphen mit Baumweite k verbotenen Minor K_{k+2} haben (siehe [6]), folgt aus Satz 4.22 insbesondere, daß built-in Relationen mit beschränkter Baumweite unendlich teilbar sind.

4.3.3 Lokale Eigenschaften

Knapp beschrieben ist eine built-in Relation \mathbf{B} unendlich teilbar, wenn man in einem *ausreichend* großen Universum $[n]$ in $B_n - A_n^d$, L Knoten mit paarweiser disjunkter d -Umgebung findet. Die Struktur außerhalb dieser d -Umgebungen ist in Bezug auf die Eigenschaft *unendlich teilbar* irrelevant. Aus dieser Perspektive kann die Eigenschaft *unendlich teilbar* als lokale Struktureigenschaft gesehen werden.

In den vorherigen Abschnitten wurde für zahlreiche Klassen von built-in Relationen die Eigenschaft *unendlich teilbar* nachgewiesen. In diesem Abschnitt wird gezeigt, daß auch die *lokalen Versionen* dieser Klassen unendlich teilbar sind.

Dazu zunächst folgende Definition:

Definition 4.33

Sei $\mathcal{P} = \{\mathcal{P}_k; k \in \mathbb{N}\}$ eine Klasse von Grapheigenschaften. Die built-in Relation \mathbf{B} hat lokal die Eigenschaft \mathcal{P} , falls eine Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$ existiert mit

$$\forall d \forall n \in \mathbb{N}, \forall x \in [n] : B_n \downarrow \mathcal{N}_d(x) \in \mathcal{P}_{f(d)},$$

d.h. die d -Umgebung eines jeden Punktes in B_n hat die Eigenschaft $\mathcal{P}_{f(d)}$.

Die Funktion f nennt man die **Parameterfunktion** von \mathbf{B} .

Beispiel 4.34

a) $\mathcal{P} = \{\text{Graph hat Baumweite } k; k \in \mathbb{N}\}$.

Die built-in Relation hat **lokal beschränkte Baumweite**, falls ein $f : \mathbb{N} \rightarrow \mathbb{N}$ existiert mit

$$\forall d, \forall n \in \mathbb{N}, \forall x \in [n] : B_n \downarrow \mathcal{N}_d(x) \text{ hat Baumweite } f(d).$$

b) $\mathcal{P} = \{\text{Graph hat verbotenen Minor } K_k; k \in \mathbb{N}\}$.

Die built-in Relation hat **lokal verbotenen vollständigen Minor**, falls ein $f : \mathbb{N} \rightarrow \mathbb{N}$ existiert mit

$$\forall d, \forall n \in \mathbb{N}, \forall x \in [n] : B_n \downarrow \mathcal{N}_d(x) \text{ hat verbotenen Minor } K_{f(d)}.$$

Die folgende Definition beschreibt die Eigenschaft *unendlich teilbar* für beliebige Graphenklassen. Sie dient an dieser Stelle nur dazu technischen Schwierigkeiten im Beweis des folgenden Satz aus dem Weg zu gehen.

Definition 4.35

Eine Graphenklasse $\mathcal{G} = \{G_k : k \in I\}$ heißt **unendlich teilbar**, falls für alle $d \in \mathbb{N}$ ein $l_d \in \mathbb{N}$ existiert, so daß für alle $L \in \mathbb{N}$ ein $n_{d,L} \in \mathbb{N}$ existiert derart, daß für alle Graphen G aus \mathcal{G} mit $|V_G| \geq n_{d,L}$ eine Menge $A_n^d \subset V_G$ mit $|A_n^d| \leq l_d$ und eine Menge $X = \{x_1, \dots, x_L\} \subset V_G$ existieren mit

$$\mathcal{N}_d^{G-A_n^d}(x_i) \cap \mathcal{N}_d^{G-A_n^d}(x_j) = \emptyset \quad \text{für alle } i \neq j.$$

Bemerkung 4.36

a) Eine built-in Relation $\mathbf{B} = (B_n)_{n \in \mathbb{N}}$ kann als Graphenklasse mit Graphen $\{B_n : n \in \mathbb{N}\}$ interpretiert werden. Falls diese Graphenklasse unendlich teilbar ist, so auch die built-in Relation \mathbf{B} .

b) Jeder Grapheigenschaft \mathcal{P} kann die Graphenklasse $\mathcal{G}_{\mathcal{P}}$, der Graphen mit Eigenschaft \mathcal{P} zugeordnet werden. Eine Grapheigenschaft \mathcal{P} heißt **teilbarkeitserzeugend**, falls $\mathcal{G}_{\mathcal{P}}$ unendlich teilbar ist.

c) Der Beweis des Satzes 4.22 zeigt, daß die Grapheigenschaft \mathcal{P}_k =verbotener Minor K_k für alle $k \in \mathbb{N}$ teilbarkeitserzeugend ist.

Insbesondere sind die Grapheigenschaften \mathcal{B}_k = beschränkte Baumweite k teilbarkeitserzeugend (siehe Bemerkung 4.32).

Der folgende Satz zeigt, daß lokale Versionen von teilbarkeitserzeugenden Grapheigenschaften, im Falle von built-in Relationen, ebenfalls teilbarkeitserzeugend sind. Der Satz kann natürlich auch für allgemeine Graphklassen formuliert werden.

Satz 4.37

Sei $\mathcal{P} = \{\mathcal{P}_k : k \in \mathbb{N}\}$ eine Klasse von teilbarkeitserzeugenden Grapheigenschaften. Eine built-in Relation \mathbf{B} , die lokal die Eigenschaft \mathcal{P} mit Parameterfunktion f hat, ist unendlich teilbar.

Beweis:

Sei im Folgenden $ZK(x)$ die Zusammenhangskomponente einer Graphstruktur, die x enthält.

Falls \mathbf{B} endlos (beliebig lange Wege) oder trennbar (beliebig viele Zusammenhangskomponenten) ist, dann nach Satz 4.3 auch unendlich teilbar.

Sei also \mathbf{B} weder endlos, noch trennbar. Dann existieren $d, N \in \mathbb{N}$ mit

- (i) $\forall n \in \mathbb{N}, \forall x \in [n]$ gilt $\mathcal{N}_d(x) = ZK(x)$.
- (ii) $\forall n \in \mathbb{N}$ enthält B_n weniger als N Zusammenhangskomponenten.

Nach (i) und den Voraussetzungen haben Zusammenhangskomponenten von B_n (n beliebig) Eigenschaft $\mathcal{P}_{f(d)}$. Nach (ii) kann man beliebig große Zusammenhangskomponenten finden. Da $\mathcal{P}_{f(d)}$ teilbarkeitserzeugend ist, kann man auf einer großen Zusammenhangskomponente (diese hat Eigenschaft $\mathcal{P}_{f(d)}$) die notwendigen Bedingungen für unendlich teilbare built-in Relationen erfüllen. □

Korollar 4.38

- a) Graphzusammenhang ist nicht ausdrückbar in monadic NP + built-in Relation mit lokal beschränkter Baumweite.
- b) Graphzusammenhang ist nicht ausdrückbar in monadic NP + built-in Relation mit lokal verbotenen vollständigen Minor.

Bemerkung 4.39

Die Klasse der built-in Relationen mit lokal beschränkter Baumweite schließt wiederum viele weitere Klassen von built-in Relationen ein, welche monadic NP nicht ausreichend verstärken um Graphzusammenhang auszudrücken. Darunter fällt z.B. die Klasse der planaren Graphen und die Klasse der Graphen mit $K_{3,n}$, $n \in \mathbb{N}$, als verbotenen Minor (siehe z.B. [14]).

4.4 Unendlich teilbare built-in Relationen bezüglich Padding

Auch unendlich teilbare built-in Relationen sind schwach im Sinne von Definition 3.1. Dies ist die Aussage des folgenden Satzes, dessen Beweis *kurz* dargestellt wird.

Satz 4.40

Graphzusammenhang \notin monadic NP + unendlich teilbare built-in Relation + polynomielles Padding.

Beweis:

Im Beweis von Satz 4.2 benötigt man zur Konstruktion von G_0 und G_1 *nur* eine von c und r abhängige feste Anzahl $n(c, r)$ von Knoten mit paarweise disjunkter (dabei wird eine *kleine* Ausnahmemenge A ignoriert) 2^r - \mathbf{B} -Umgebung. Eine unendlich teilbare built-in Relation ermöglicht es aber, *beliebig viele* Knoten mit paarweise disjunkter 2^r - \mathbf{B} -Umgebung zu finden. Man findet also ein $n \in \mathbb{N}$ derart, daß in B_n mindestens $n(c, r)^k + n(c, r)$, $k \in \mathbb{N}$ Knoten mit paarweise disjunkter 2^r -Umgebung existieren. Die $n(c, r)$ Knoten, deren 2^r -Umgebungen am kleinsten sind (gemessen an der Anzahl der Knoten in der Umgebung), werden für die Konstruktion von G_0 und G_1 verwendet. Alle anderen Knoten bleiben bezüglich der Kantenrelation E *isoliert*, sind also Paddingknoten. Zum einen sind dies mindestens $n(c, r)^k$ Knoten mitsamt ihren 2^r -Umgebungen, also k -polynomiell viele in der Anzahl der zur Konstruktion verwendeten Knoten. Zum andern sieht man *leicht*, daß Duplikator das (c, r) -monNP Spiel auf diesen um Paddingknoten erweiterten Strukturen gewinnen kann. Dazu spielt er auf G_0 und G_1 bezüglich seiner Strategie aus Satz 4.2 und auf den Paddingknoten konstant. \square

4.5 Zusammenfassung und Bemerkungen zu Kapitel 3 und 4

Gegenstand dieser Arbeit ist die Ausdruckstärke von monadischen Erweiterungen von monadic NP. Im ersten Teil wurden Erweiterungen um gewisse built-in Relationen betrachtet. Speziell ging es darum, für bestimmte Klassen von built-in Relationen \mathbf{B} zu zeigen, daß

$$\text{Graphzusammenhang} \notin \text{monadic NP} + \mathbf{B}. \quad (*)$$

Die Vielfalt an unterschiedlichen Ergebnissen und Konzepten in diesem Bereich konnte durch die Einführung zweier (nicht disjunkter) Klassen von built-in Relationen zusammengefaßt werden. Dabei handelt es sich um:

- a) verpackbare built-in Relationen (Kapitel 3) und

b) unendlich teilbare built-in Relationen (Kapitel 4).

Alle built-in Relationen, von denen bisher gezeigt werden konnte, daß (*) gilt, sind entweder verpackbar oder unendlich teilbar. Umgekehrt gilt (*) für alle built-in Relationen, die verpackbar oder unendlich teilbar sind. Außerdem wurde gezeigt, daß verpackbare und unendlich teilbare built-in Relationen schwach im Sinne von Definition 3.1 sind. Insbesondere sind damit alle Relationen, für die (*) bisher nachgewiesen werden konnte, schwach.

Beispiele für verpackbare built-in Relationen sind:

- Äquivalenzrelationen
- Ordnungsrelationen
- Relationen mit beschränkter Baumweite

Beispiele für unendlich teilbare built-in Relationen sind:

- Relationen mit Maximalgrad $n^{o(1)}$
- x-built-in Relationen
- Relationen mit verbotenem Minor K_k
- Relationen mit lokal beschränkter Baumweite
- Relationen mit lokal verbotenem vollständigen Minor
- Multiplikationsrelation
- Teilbarkeitsrelation

Bei den grün markierten built-in Relationen handelt es sich um neue Ergebnisse (im Zusammenhang mit (*)). Für die blau markierte built-in Relationen ist (*) kein neues Ergebnis. Allerdings handelt es sich um neue Beweismethoden, wodurch die bisherigen Beweise sowohl verkürzt, als auch vereinfacht werden konnten.

Der Nachweis von (*) für starke built-in Relationen *scheint* sehr schwer zu sein. Auch wäre das, wie in der Einleitung von Kapitel 3 beschrieben, ein möglicher erster Schritt zum Beweis von $\text{NP} \neq \text{co-NP}$. Es macht daher Sinn, die von Fagin, Stockmeyer und Vardi in [13] aufgestellte Vermutung, daß

Graphzusammenhang \notin monadic NP + beliebige built-in Relationen

zunächst in der abgeschwächten Form

Graphzusammenhang \notin monadic NP + schwache built-in Relationen (**)

zu untersuchen. Die Tatsache, daß für eine Vielzahl von schwachen built-in Relationen (*) gezeigt werden konnte, spricht für diese Vermutung. Um (**) zu beweisen, wird es vermutlich notwendig sein, eine *brauchbare* Klassifikation für schwache built-in Relationen zu finden. Da die Konzepte von verpackbaren und unendlich teilbaren built-in Relationen alle Relationen, für die (*) gezeigt werden konnte, subsumieren, stellt sich die Frage, ob mit diesen Konzepten eine solche Klassifikation möglich ist. Dazu zunächst eine Definition:

Definition 4.41

Eine built-in Relation \mathbf{B} ist halb verpackbar/halb teilbar, wenn ein $l \in \mathbb{N}$ existiert, so daß für alle $d, L \in \mathbb{N}$ ein $n(d, L) \in \mathbb{N}$ existiert derart, daß für alle $n > n(d, L)$ ein A_n^d mit $|A_n^d| \leq l$ existiert und eine Partition von $B_n = B_n^u \dot{\cup} B_n^v$ derart existiert, daß man L Knoten $\{x_1, \dots, x_L\}$ aus $B_n^u \setminus A_n^d$ findet mit

- paarweise disjunkter d -Umgebung in B_n^u (Teilbarkeitsanteil)
- alle Knoten aus $\mathcal{N}_d^{B_n^u \setminus A_n^d}(x_i)$ verhalten sich bezüglich allen Knoten aus B_n^v identisch (Verpackbarkeitsanteil).

Mit den Methoden aus den beiden vorangegangenen Kapiteln (speziell die aus dem Beweis von Satz 4.2) kann man zeigen, daß halb verpackbare/halb teilbare built-in Relationen schwach sind und (*) erfüllen.

Auch sieht man *leicht*, daß jede unendlich teilbare built-in Relation halb verpackbar/halb teilbar ist.

Ein möglicher Ansatz für eine Klassifikation von schwachen built-in Relationen ist Inhalt der folgenden **Frage**:

Ist jede schwache built-in Relation entweder

- verpackbar oder
- unendlich teilbar oder
- halb verpackbar/halb teilbar?

Eine positive Antwort auf diese Frage würde (**) beweisen.

Kapitel 5

Closed Monadic NP

Als Fagin in seiner Arbeit [11] die Klasse monadic NP eingeführt hat, mußte er sich der Erkenntnis beugen, daß Nichtausdrückbarkeitsbeweise oberhalb von monadischer Quantifizierung *nicht machbar schienen*. Schon die Möglichkeit der Verwendung von binären SO-Quantoren ermöglicht es dem Spoiler im EF-Spiel beliebige Knoten miteinander zu verbinden und macht so die Verwendung aller bisher bekannten Spielstrategien unmöglich. Diese *scheinbar extreme* Zunahme an Schwierigkeit im binären NP-Spiel, kann mit der Tatsache, daß die logische Sprache binäres NP sehr viel ausdrucksstärker als monadic NP ist, plausibel gemacht werden. Z.B. kann Graphzusammenhang in binären NP ausgedrückt werden. Es reicht dabei sogar nur binäre Quantifizierungen über Ordnungen zuzulassen (siehe [7]). Weiterhin gibt es die Vermutung, daß alle Grapheneigenschaften aus NP in binärem NP ausdrückbar sind.

Eingeschränkt auf monadic NP ist es sehr wohl möglich Nichtausdrückbarkeitsbeweise (speziell durch EF-Spiele) zu führen. Im weiteren Verlauf wurde monNP um verschiedene built-in Relationen erweitert (siehe Kapitel 3 und 4) und eine Weile lang war man bemüht, die Klasse der built-in Relationen, mit deren Hilfe Graphzusammenhang in monNP nicht ausdrückbar ist, stetig zu erweitern. Für viele schwache built-in Relationen ist dies gelungen. Die *Mauer* zu den starken built-in Relationen hingegen scheint sehr hoch zu sein.

Da die Betrachtung weiterer spezieller schwacher built-in Relationen *vermutlich* keinen Durchbruch dieser Mauer ermöglichen wird, allgemeine Ergebnisse für die Menge der schwachen built-in Relationen nicht greifbar scheinen, ist die Untersuchung von monNP + built-in Relationen ins Stocken geraten. Stattdessen hat man angefangen, andersartige monadische Erweiterungen von monNP zu untersuchen.

In der Arbeit [3] von Ajtai, Fagin und Stockmeyer betrachten und definieren die Autoren zwei monadische Erweiterungen von monNP. Es handelt sich dabei zum einen um den Abschluß von monNP bzgl. FO-Quantifizierung ($PFO(\text{monNP})$) und zum anderen um den Abschluß von monNP bzgl. FO-Quantifizierung und existentieller monadischer SO-Quantifizierung (closed monadic NP, kurz: cmonNP).

Die Autoren konnten folgende Hierarchie zeigen:

$$\text{monNP} \stackrel{1)}{\subsetneq} \text{PFO}(\text{monNP}) \stackrel{2)}{\subsetneq} \text{cmonNP} \stackrel{3)}{\subsetneq} \text{NP}.$$

Dabei ergibt sich 1) aus der Tatsache, daß Graphzusammenhang in $\text{PFO}(\text{monNP})$ ausdrückbar ist und 3) aus Betrachtungen von Turán in [27] (z.B. ist die Eigenschaft *Existenz eines Hamiltonkreises* nicht in cmonNP , sehr wohl aber in NP). Um 2) zu zeigen, haben die Autoren von einer sehr *künstlich* konstruierten Grapheigenschaft \mathcal{P}_g nachgewiesen, daß diese nicht in $\text{PFO}(\text{monNP})$ ausdrückbar ist, sehr wohl aber in cmonNP . Der vorgestellte Spielbeweis ist *lang* und *recht schwer*.

In [18] konnte Marcinkowski zeigen, daß die Grapheigenschaft *Erreichbarkeit in gerichteten Graphen* nicht in $\text{PFO}(\text{monNP})$ ausdrückbar ist. Zum einen ist dieser Beweis viel kürzer und anschaulicher, zum anderen die Eigenschaft sehr natürlich.

In diesem Kapitel wird zunächst ein kurzer Überblick der Methode von Marcinkowski gegeben. Anschließend wird die Methode von Fagin et al. ausführlich dargestellt, dabei verallgemeinert und vereinfacht.

Es wird gezeigt, daß obwohl beide Beweise auf den ersten Blick sehr verschieden wirken, sie doch auf derselben Grundidee aufbauen. Diese beruht darauf, die *vorangestellten* FO-Runden durch *Antworten in isomorphen Teilstrukturen* zu überstehen.

Zwei interessante Fragen in Bezug auf closed monadic NP sind bisher unbeantwortet geblieben. Zum einen kann man innerhalb von cmonNP eine Hierarchie definieren (siehe Kapitel 2) und die Frage nach deren Striktheit stellen. Zum anderen ist es unklar, ob es monadische Eigenschaften gibt, die nicht in cmonNP liegen. D.h. die Frage, ob $\text{cmonNP} \neq \text{MSO}$ ist. Falls $\text{NP} \neq \text{co-NP}$ gilt, kann die Eigenschaft *nicht-3-Färbbarkeit* nicht in NP und insbesondere nicht in closed monadic NP ausgedrückt werden. Da diese Eigenschaft offensichtlich in MSO ausdrückbar ist, findet die zweite Frage unter der Annahme $\text{NP} \neq \text{co-NP}$ eine positive Antwort.

Ebenso wie im Zusammenhang mit starken built-in Relationen scheint es, als wenn Spielbeweise zur Beantwortung dieser beiden Fragen sehr schwer sind. Eine mögliche Begründung dafür wird in den abschließenden Betrachtungen zu Kapitel 5 und 6, am Ende von Kapitel 6, gegeben.

5.1 Der Beweis von Marcinkowski

In diesem Abschnitt wird das Beweisverfahren von Marcinkowski knapp erläutert. Es wird dabei nur die Konstruktion der Spielstrukturen betrachtet und auf eine Beschreibung des Spiels verzichtet. In Kapitel 6 wird gezeigt, daß die Grapheigenschaft *nicht-k-Färbbarkeit*, $k \geq 3$, nicht ausdrückbar ist in $\text{PFO}(\text{monNP})$. Da die dort verwendeten Konstruktionen der Spielstrukturen und der Spielverlauf auf der Methode von Marcinkowski aufbauen, ist eine genaue Betrachtung an dieser Stelle nicht nötig. Eine ausführliche Betrachtung findet man in [18].

Ajtai und Fagin haben in [2] gezeigt, daß die Eigenschaft $\mathcal{E}_d := \text{Erreichbarkeit in gerichteten Graphen}$ nicht in monNP ausgedrückt werden kann.

Satz 5.1 (Ajtai, Fagin 90)

Die Eigenschaft $\mathcal{E}_d := \text{Erreichbarkeit in gerichteten Graphen}$ ist nicht ausdrückbar in monNP.

Das bedeutet: für jedes Parametertupel $(c, r) \in \mathbb{N}^2$ gibt es Graphen $G_0 \in \mathcal{E}_d$ und $G_1 \notin \mathcal{E}_d$ derart, daß Duplikator das (c, r) -monNP Spiel auf G_0, G_1 gewinnen kann.

Ajtai und Fagin konnten dabei *nur* die Existenz solcher Graphen G_0, G_1 nachweisen, diese aber nicht konstruieren (dies ist in der Folgezeit auch nicht gelungen). Die dabei verwendeten Methoden, speziell die Verwendung von Zufallsgraphen, sind für sich allein sehr interessant und waren im Zusammenhang mit Spielbeweisen bzgl. monNP ein völlig neuer Ansatz.

Marcinkowski verwendet die Graphen G_0, G_1 als *Bausteine* für seine Spielgraphen im PFO(monNP) Spiel.

Satz 5.2 (Marcinkowski 99)

Die Eigenschaft \mathcal{E}_d ist nicht ausdrückbar in PFO(monNP).

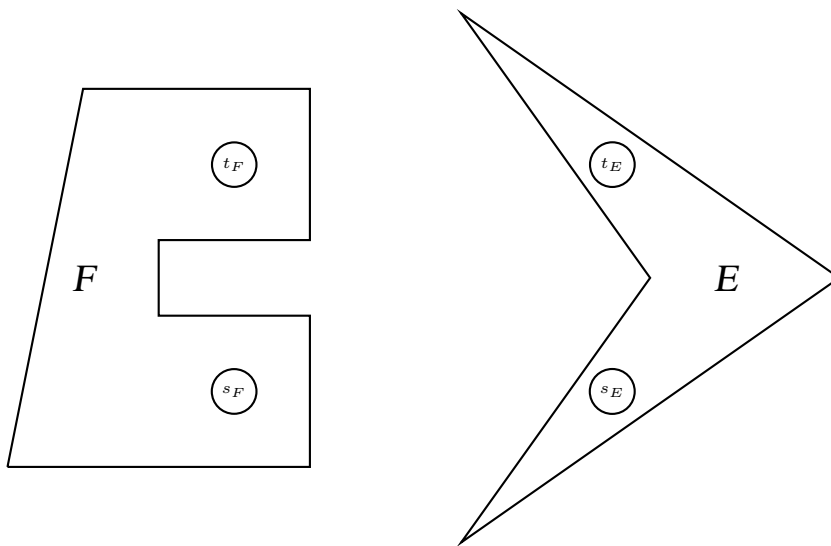
Beweisanfang:

Seien $r_1, c, r_2 \in \mathbb{N}$ fest gewählt. Gesucht sind Strukturen $S_0 \in \mathcal{E}_p, S_1 \notin \mathcal{E}_p$, so daß Duplikator das (r_1, c, r_2) -PFO(monNP) Spiel auf S_0, S_1 gewinnt.

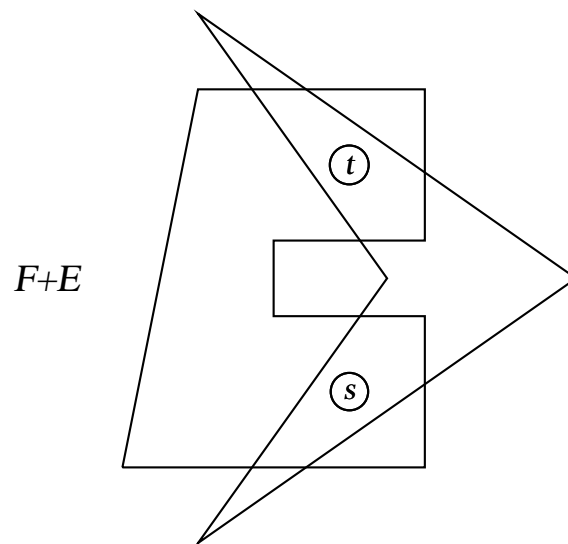
Seien $G_0 \in \mathcal{E}_d, G_1 \notin \mathcal{E}_d$ so gewählt, daß Duplikator das (c, r_2) -monNP Spiel auf G_0, G_1 gewinnt. S_0, S_1 werden aus einer Menge von Kopien von G_0, G_1 konstruiert, die *geschickt* miteinander verbunden werden. Dazu werden notwendige Verbindungsoperatoren für Graphstrukturen mit zwei ausgezeichneten Knoten s, t definiert.

Definition 5.3

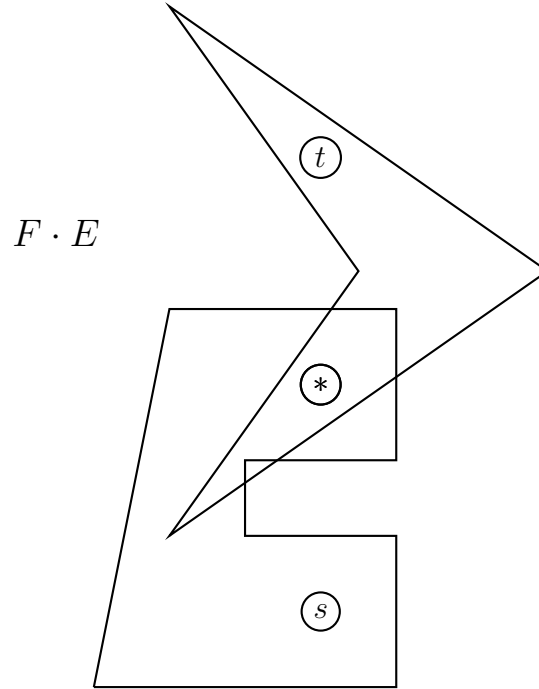
Seien F, E Graphstrukturen mit ausgezeichneten Knoten s, t ($s = \underline{\text{source}}, t = \underline{\text{target}}$).



- a) Der Graph $F + E$ entsteht aus einer Kopie von F und einer Kopie von E . Die Quelle und Senke von F, E werden zu einer gemeinsamen Quelle und Senke verschmolzen. Folgende Skizze verdeutlicht die Struktur von $F + E$:



- b) Der Graph $F \cdot E$ entsteht aus einer Kopie von F und einer Kopie von E . Die Senke t_F von F und die Quelle s_E von E werden miteinander zum Knoten * verschmolzen. Die Quelle s von $F \cdot E$ ist die Quelle s_F von F . Die Senke t von $F \cdot E$ ist die Senke t_E von E . Folgende Skizze verdeutlicht die Struktur von $F \cdot E$:



Es werden nun induktiv Strukturen $S_0^l \in \mathcal{E}_d, S_1^l \notin \mathcal{E}_d$ definiert, so daß Duplikator das $(\exists\forall)^l(c, r_2)$ -monNP Spiel auf S_0^l, S_1^l gewinnt.

$$l=0: S_0^0 := G_0, \quad S_1^0 := G_1$$

$l \rightarrow l + 1$:

$$S_0^{l+1} :=$$

$$[(S_0^l \cdot S_0^l + S_1^l \cdot S_1^l) \cdot (S_0^l \cdot S_0^l + S_1^l \cdot S_1^l)] + [(S_0^l \cdot S_1^l + S_1^l \cdot S_0^l) \cdot (S_0^l \cdot S_1^l + S_1^l \cdot S_0^l)]$$

$$S_1^{l+1} :=$$

$$[(S_0^l \cdot S_0^l + S_1^l \cdot S_1^l) \cdot (S_0^l \cdot S_1^l + S_1^l \cdot S_0^l)] + [(S_0^l \cdot S_1^l + S_1^l \cdot S_0^l) \cdot (S_0^l \cdot S_0^l + S_1^l \cdot S_1^l)]$$

Man sieht leicht, daß $S_0^l \in \mathcal{E}_d$ und $S_1^l \notin \mathcal{E}_d$.

Duplikators Gewinnstrategie ergibt sich induktiv. Der Induktionsschritt ergibt sich hauptsächlich aus der Tatsache, daß die Menge der *Klammerausdrücke* (z.B. $(S_0^l \cdot S_0^l + S_1^l \cdot S_1^l)$) in S_0^l und S_1^l identisch ist. Genauer dazu findet man in [18] und im folgenden Kapitel.

5.2 Der Beweis von Ajtai-Fagin-Stockmeyer

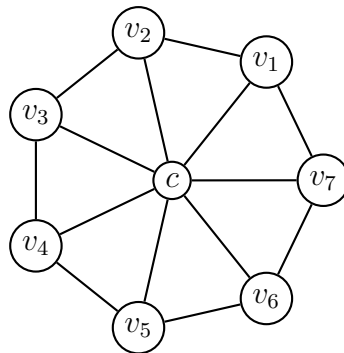
In diesem Abschnitt wird die Konstruktion und Spielidee von Fagin et al. skizziert. Dabei werden abweichend zu [3] folgende Veränderungen durchgeführt:

- Die betrachtete Eigenschaft wird leicht verändert und damit leichter beschreibbar.
- Das Spiel wird auf den PFO-Abschluß eines beliebigen monadischen Präfixes erweitert.
- Die Beweise der benötigten Lemmata werden durch Verwendung der Tatsache *PARITY ist nicht schwach ausdrückbar in FO* und des Satzes von Büchi stark vereinfacht.

Zunächst einige Definitionen:

Definition 5.4

- a) Ein Graph $G = (V, E, c)$ heißt ein **Rad**, falls G aus einem Kreis und einem ausgezeichneten Knoten c besteht, der mit allen Kreispunkten verbunden ist. Im Folgenden sei immer $V = c \cup \{v_1, \dots, v_n\}$, wobei die Nummerierung der Knoten v_i deren Reihenfolge (gegen den Uhrzeigersinn) auf dem Kreis widerspiegelt.



- b) Sei R ein Rad mit Radmittelpunkt c und Kreispunkten (v_1, \dots, v_n) . Seien H_1, \dots, H_n beliebige Graphen. Der Graph $\mathbf{R}(H_1, \dots, H_n)$ entsteht aus R durch Erweiterung um Kopien von H_1, \dots, H_n . Dabei wird der Knoten v_i ($i = 1 \dots, n$) mit allen Knoten aus H_i verbunden. Formal:

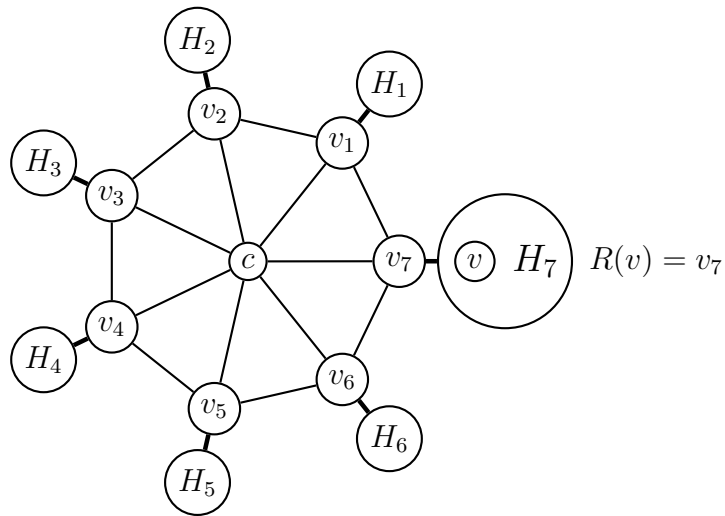
$$R(H_1, \dots, H_n) = (c \cup \{v_1, \dots, v_n\} \cup \{V_{H_1}, \dots, V_{H_n}\}, E, c) \text{ mit}$$

$$E = \{(c, v_i); i = 1, \dots, n\} \cup \{(v_i, v_{i+1}); i = 1, \dots, n\} \cup \{(v_1, v_n)\} \cup \{(v_i, x); x \in V_{H_i}\}$$

- c) Seien R ein Rad und H_1, \dots, H_n Graphen. Seien $G = \mathbf{R}(H_1, \dots, H_n)$ und $v \neq c$ ein Knoten von G . Der **Radknoten** von v (kurz: $R(v)$) wird durch

$$R(v) = v_i \iff (v \in H_i \vee v = v_i)$$

definiert.

**Definition 5.5**

Sei \mathcal{P} eine beliebige Grapheigenschaft. Mittels \mathcal{P} wird eine neue Grapheigenschaft \mathcal{P}_g definiert:

$$G \in \mathcal{P}_g \iff G = (V, E, c) \text{ und es existiert ein Rad } R = (c \cup \{v_1, \dots, v_n\}, E_R, c) \text{ und Graphen } H_1, \dots, H_n, \text{ so da\ss:}$$

- 1) $G = R(H_1, \dots, H_n)$.
- 2) $\#\{H_i \in \mathcal{P}; i = 1, \dots, n\}$ ist gerade.

Es gilt nun folgender Satz:

Satz 5.6

- a) Sei W ein beliebiges monadisches Prafix und \mathcal{P} eine Grapheigenschaft mit $\mathcal{P} \notin \oplus L(W)$, aber $\mathcal{P} \in PFO(W)$. Dann gilt: $\mathcal{P}_g \notin PFO(W)$.
- b) Falls $\mathcal{P}, \overline{\mathcal{P}} \in cmonNP$, dann $\mathcal{P}_g \in cmonNP$.

Bemerkung 5.7

- 1) Die Bedingung $\mathcal{P} \notin \oplus L(W)$ scheint sehr streng. Marcinkowski hat in [16] eine Eigenschaft $bool(\mathcal{P})$ definiert und folgendes Lemma bewiesen:

Lemma 5.8

Sei $\mathcal{P} \notin L(W)$, aber $\overline{\mathcal{P}}, \mathcal{P} \in L(V)$. Dann ist $bool(\mathcal{P}) \notin \oplus L(W)$, aber $bool(\mathcal{P}) \in L(\exists \exists V)$.

Die Eigenschaft $bool(\mathcal{P})$ erfullt also die Voraussetzungen von Satz 5.6 a), falls $\mathcal{P} \notin L(W)$, aber $\mathcal{P} \in L(\overline{W}) \cap PFO(W)$.

- 2) Marcinkowski konnte in [16] folgenden Satz zeigen:

Satz 5.9

Sei \mathcal{P} eine Grapheigenschaft für die gilt $\mathcal{P} \notin L(W)$, aber $\mathcal{P} \in PFO(W)$, dann existiert eine Grapheigenschaft $\text{reach}(\mathcal{P})$ für die gilt $\text{reach}(\mathcal{P}) \notin PFO(W)$, aber $\text{reach}(\mathcal{P}) \in \exists PFO(W)$.

Insbesondere folgt daraus, daß falls die Hierarchie innerhalb von closed monadic NP nicht strikt ist, diese zumindest nicht auf einer Zwischenstufe zusammenfällt.

Aus Satz 5.6 erhält man folgende leicht schwächere Aussage:

Korollar 5.10

Sei \mathcal{P} eine Grapheigenschaft für die gilt $\mathcal{P} \notin \oplus L(W)$, aber $\mathcal{P} \in PFO(W)$, dann gilt:

- a) $\mathcal{P}_g \notin PFO(W)$.
- b) $\mathcal{P}, \overline{\mathcal{P}} \in \text{cmonNP} \implies \mathcal{P}_g \in \text{cmonNP}$.
- c) $\overline{\mathcal{P}} \notin \text{cmonNP} \implies \text{cmonNP} \subsetneq \text{MSO}$.

Bemerkung 5.11

Auf der einen Seite gewinnt Duplikator das $PFO(\text{monNP})$ -Spiel für \mathcal{P}_g ebenso wie bei Marcinkowski, indem er die vorangestellten FO-Runden durch Spielen in isomorphen Bausteinen überwindet. Auf der anderen Seite ist die Eigenschaft \mathcal{P}_g von Fagin et al. derart gewählt, daß es notwendig ist, sowohl \mathcal{P} als auch $\overline{\mathcal{P}}$ in cmonNP beschreiben zu können, um \mathcal{P}_g in cmonNP ausdrücken zu können. Dies ist der wesentliche Nachteil von der Konstruktion von Fagin et al. und gleichzeitig der Grund für die schwächere Aussage des obigen Korollars.

Der Beweis von Satz 5.6 verläuft in mehreren Schritten. Zunächst benötigt man ein spezielles Spiel auf schwarz/weiß Rädern und zwei Lemmata. Ein **schwarz/weiß Rad** ist ein Rad, in dem jeder Kreisknoten entweder schwarz oder weiß gefärbt ist.

Definition 5.12 ((r, t, M)-schwarz/weiß Spiel)

Das (r, t, M)-schwarz/weiß Spiel auf schwarz/weiß Rädern A, B verläuft in r Runden. In jeder Runde i wählt Spoiler einen Punkt $a_i(b_i)$ aus $A(B)$. Duplikator antwortet mit einem Punkt der anderen Struktur. Duplikator gewinnt das Spiel unter folgenden Bedingungen:

1. Für alle i hat a_i dieselbe Farbe (kurz: $F(a_i)$) wie b_i .
2. $a_i = a_j \iff b_i = b_j$.
3. Die Folgen a_1, \dots, a_r und b_1, \dots, b_r sind bzgl. dem Uhrzeigersinn auf den jeweiligen Kreisrädern gleich geordnet.
4. Sei $\pi \in S(r)$ ordnungsdefinierend für a_1, \dots, a_r , d.h. $a_{\pi(1)} \leq \dots \leq a_{\pi(r)}$ bzgl. der Uhrzeigersinnordnung. Sei $l_i^a(l_i^b)$ der Abstand von $a_{\pi(i)}(b_{\pi(i)})$ nach $a_{\pi(i+1)}(b_{\pi(i+1)})$ (im Uhrzeigersinn), dann gilt:

- a) $l_i^a = l_i^b$ oder
 b) $l_i^a, l_i^b > t$ und $l_i^a \equiv_M l_i^b$.

Es werden nun zwei *wichtige* Lemmata bewiesen. Im Gegensatz zur Originalarbeit sind die Beweise hier nicht konstruktiv und dadurch deutlich kürzer und einfacher. Desweiteren ist die Aussage des zweiten Lemmas auf beliebige monadische Präfixe erweitert, ein entscheidender Schritt zum Beweis von Satz 5.6. Den Beweis des zweiten Lemmas verdanke ich Frau Dr. Nicole Schweikardt.

Lemma 5.13

Seien $r, t, M \in \mathbb{N}$. Dann existieren beliebig große schwarz/weiß Räder A, B mit $|A| = |B|$, so daß eine gerade (ungerade) Anzahl von Knoten von A (B) weiß gefärbt ist und Duplikator eine Gewinnstrategie im (r, t, M) -schwarz/weiß Spiel auf A, B hat.

Beweis:

Sei Q eine einstellige Relation. In [1] hat Ajtai gezeigt, daß die Eigenschaft

$$\mathbf{P} = |Q| \text{ ist gerade}$$

nicht schwach in FO ist, d.h. die Eigenschaft \mathbf{P} ist nicht ausdrückbar in FO+beliebige built-in Relationen. Sei $S = (Q, B_1, B_2, B_3)$ eine Signatur. Die built-in Relationen B_1, B_2, B_3 sind wie folgt definiert:

- B_1^n ist zweistellig und definiert die Kantenrelation eines Rades auf n Punkten.
- B_2^n ist zweistellig und definiert die Uhrzeigersinnordnung bzgl. B_1^n . Der Radmittelpunkt sei dabei zusätzlich der maximale Punkt bzgl. dieser Ordnung.
- B_3^n ist unär und definiert durch

$$v_i \in B_3^n \iff M \mid i.$$

Dabei sei $V = (v_1, \dots, v_n)$ die Knotenmenge sortiert bzgl. B_2^n .

Da \mathbf{P} nicht schwach in FO ist, existieren für alle $\tilde{r} \in \mathbb{N}$ S-Strukturen A, B mit $|A| = |B|, |Q_A|$ gerade und $|Q_B|$ ungerade, so daß Duplikator das \tilde{r} -Runden FO-Spiel auf A, B gewinnt. Falls $\tilde{r} > r$ groß genug gewählt ist, gewinnt Duplikator auch das (r, t, M) -Spiel auf A, B , wobei Q_A, Q_B jeweils die Menge der weißen Punkte sei. Die dritte Gewinnbedingung wird dabei durch B_2 garantiert. Die vierte durch B_3 und ausreichend viele (abhängig von t und M) FO-Runden im Anschluß an die ersten r Runden. \square

Lemma 5.14

Sei W ein beliebiges monadisches Präfix x . Sei $w \in W$ beliebig. Es existieren $t, M \in \mathbb{N}$, so daß für alle Ketten A, B mit $|A|, |B| > t$ und $|A| \equiv_M |B|$ Duplikator das w -Spiel gewinnt. OBdA werden dabei die Endpunkte konstant gespielt.

Beweis:

Es existieren nur endlich viele nicht äquivalente $\varphi \in L(w)$. Sei Φ die Menge aller paarweise nicht äquivalenten $\varphi \in L(w)$. Der Grundraum sei dabei die Menge der Anfangsstücke von \mathbb{N} , versehen mit der natürlichen Ordnung, d.h.

$$\text{Mod}(\varphi) \subseteq \{(\{1, \dots, n\}, <); n \in \mathbb{N}\}.$$

Nach dem Satz von Büchi ist $\text{Mod}(\varphi)$ eine reguläre Sprache über einem Alphabet mit einem Buchstaben $\{a\}$. Mit Hilfe des Pumping Lemmas erhält man leicht folgende Aussage (Erläuterungen zum Satz von Büchi, dem Pumping-Lemma und regulären Sprachen findet man z.B. in [26]):

$$\exists t_\varphi, M_\varphi \forall n \geq t_\varphi : a^n \in \text{Mod}(\varphi) \iff a^{n+\mu M_\varphi} \in \text{Mod}(\varphi) \quad \forall \mu \geq 1.$$

Sei nun $t := \max_{\varphi \in \Phi} t_\varphi$ und $M := \text{kgV}\{M_\varphi\}$ dann gilt

$$\forall n > t : a^n \sim_w a^{n+\mu M} \quad \forall \mu \in \mathbb{N}.$$

Eine leichte Reduktion liefert die Aussage des Lemmas. □

Nach diesen Vorbereitungen folgt nun der Beweis von Satz 5.6 a):

Beweis:

Zunächst eine Definition.

Definition 5.15

Sei G ein beliebiger Graph. Der Graph G^{+1} entsteht indem man G um einen Knoten erweitert und diesen mit allen Knoten aus G verbindet.

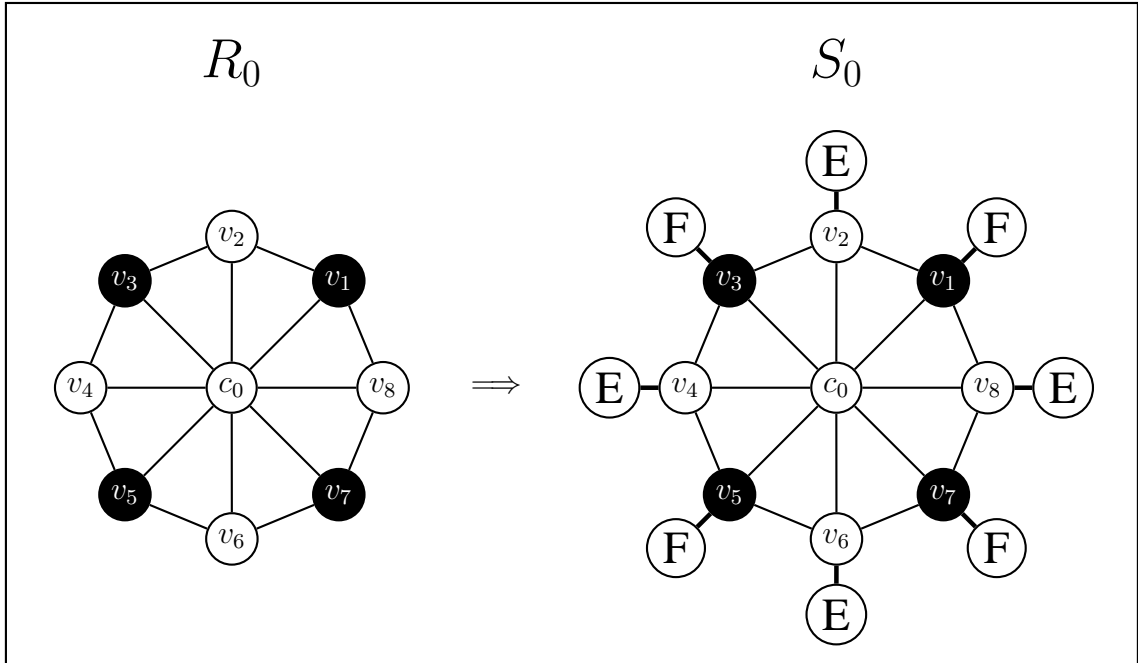
a) Sei W ein beliebiges monadisches Präfix und \mathcal{P} eine Grapheigenschaft mit $\mathcal{P} \notin \oplus L(W)$, aber $\mathcal{P} \in \text{PFO}(W)$. Sei $w \in W$ beliebig. Seien $E \in \mathcal{P}$ und $F \notin \mathcal{P}$ Graphen mit $E \sim_{\oplus w} F$.

Für $c \in \mathbb{N}$ sei $c(E^{+1}, F^{+1}) := c(E^{+1}) + c(F^{+1})$. Dabei ist $c(G)$ die Anzahl der Färbungen von G mit c Farben. Es wird nun ein weiteres monadisches Präfix \tilde{w} über seine Präfixdarstellung (siehe Definition 2.26) definiert. Die Präfixdarstellung von \tilde{w} erhält man aus der Präfixdarstellung von w , indem man die Färbungszahlen c_i durch $c_i(E^{+1}, F^{+1})$ ersetzt.

Sei nun $u \in (\exists|\forall)^* w$ beliebig. r sei die Anzahl der vordersten FO-Quantoren in u . Um zu zeigen, daß $\mathcal{P}_g \notin \text{PFO}(W)$ liegt, muß man Strukturen $S_0 \in \mathcal{P}_g, S_1 \notin \mathcal{P}_g$ finden mit $S_0 \sim_u S_1$. $t, M \in \mathbb{N}$ seien so gewählt, daß sie bezüglich \tilde{w} der Aussage von Lemma 5.14 genügen. Lemma 5.13 liefert bezüglich r, t, M zwei schwarz/weiß Räder R_0, R_1 , so daß R_0 gerade und R_1 ungerade viele weiße Punkte enthält und Duplikator das (r, t, M) -schwarz/weiß Spiel auf R_0, R_1 gewinnt. Seien $V_0 = c_0 \cup \{v_1, \dots, v_n\}, V_1 = c_1 \cup \{w_1, \dots, w_n\}$ die Knotenmengen von R_0 und R_1 . S_0, S_1 sind wie folgt definiert:

$$S_0 = R_0(C_1, \dots, C_n) \quad \text{mit} \quad C_i = \begin{cases} E & : v_i \text{ weiß} \\ F & : v_i \text{ schwarz} \end{cases}$$

$$S_1 = R_1(D_1, \dots, D_n) \quad \text{mit} \quad D_i = \begin{cases} E & : w_i \text{ weiß} \\ F & : w_i \text{ schwarz} \end{cases}$$



Spielstrategie im u -Spiel:

r FO-Runden: Diese vorangestellten FO-Runden übersteht Duplikator, indem er seine Strategie im (r, t, M) -Spiel auf R_0 und R_1 ausnutzt. Wählt Spoiler z.B. einen Knoten v ($\neq c_0$) aus S_0 , welcher oBdA kein Radknoten ist, dann übersetzt Duplikator v zunächst in einen Radknoten mittels $R(v)$. Die Strategie im (r, t, M) -Spiel liefert einen Radknoten w aus S_1 , an dem derselbe Baustein hängt wie an v . Duplikator antwortet nun indem er in diesem Baustein isomorph spielt.

Bemerkung 5.16

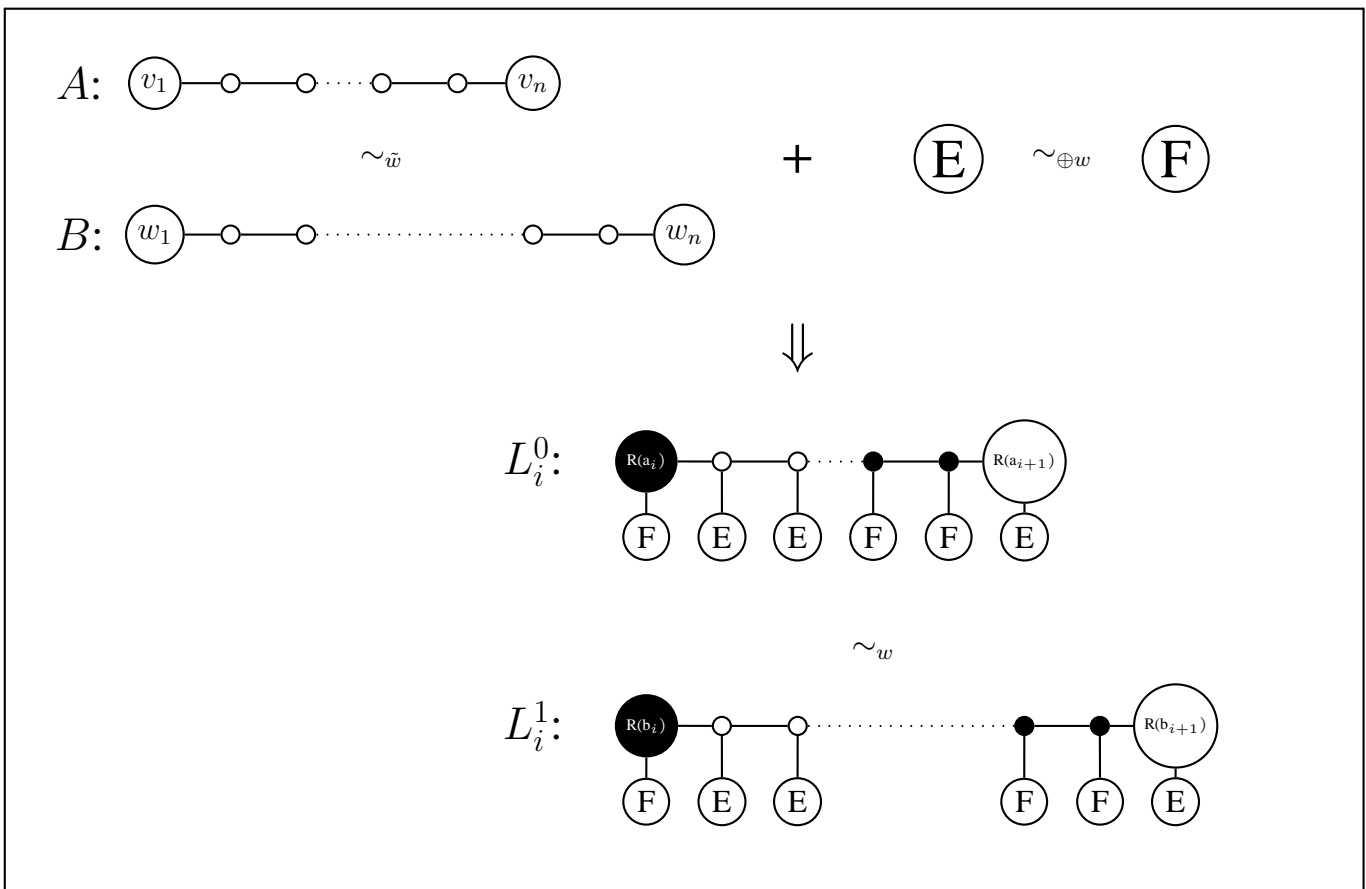
Dieser Spielabschnitt beinhaltet die oben angesprochene Basisidee des PFO(W)-Spieles: Die zusätzlichen FO-Runden übersteht Duplikator durch isomorphes Spielen auf isomorphen Teilstrukturen.

w -Spiel: Seien a_1, \dots, a_r und b_1, \dots, b_r die im ersten Spielabschnitt gewählten Knoten. OBdA sei

- $R(a_i) \neq R(a_j)$ für $a_i \neq a_j$
- $a_i \neq c \forall i$ und
- $R(a_1) \prec \dots \prec R(a_r)$, wobei \prec eine feste Uhrzeigersinnordnung auf R_0 ist.

Sei l_i^0 die Distanz zwischen $R(a_i)$ und $R(a_{i+1})$ im Uhrzeigersinn (analoge Definition von l_i^1). Aufgrund der Gewinnbedingungen im (r, t, M) -schwarz/weiß Spiel gilt $l_i^0 \equiv_M l_i^1 \forall i$. Der Teilgraph L_i^0 von S_0 besteht aus dem Intervall $[R(a_i), R(a_{i+1})]$ inklusive der an den Radknoten hängenden Bausteine E, F (analoge Definition für L_i^1).

Duplikator spielt das w -Spiel nun jeweils *separat* auf den Teilgraphen $L_i^0, L_i^1, i = 1, \dots, r$. Seine Strategie dafür ($L_i^0 \sim_w L_i^1$) ist eine Kombination aus seiner Strategie im $\oplus w$ -Spiel auf E, F und seiner Strategie im \tilde{w} -Spiel auf Ketten A, B mit $|A| \equiv_M |B|$.



w -Spiel auf L_i^0, L_i^1 : Seien $v_1(= a_i), v_2, \dots, v_n(= a_{i+1})$ bzw. $w_1(= b_i), w_2, \dots, w_m(= b_{i+1})$ die Radknoten von L_i^0 bzw. L_i^1 . Nach Voraussetzung gilt $n \equiv_M m$. Seien weiter A, B Ketten der Länge n bzw. m mit Knotenmengen $\{\bar{v}_1, \dots, \bar{v}_n\}$ bzw. $\{\bar{w}_1, \dots, \bar{w}_m\}$.

Definition 5.17

a) Sei G ein Graph. Eine $c \sim$ **F**ärbung von G ist eine Knotenfärbung von G mit c Farben.

Bemerkung: Im folgenden Kapitel wird der Begriff c -Färbung von G verwendet. Dabei bedeutet dort c -Färbung eine Knotenfärbungen von G mit c Farben und den zusätzlichen Bedingung:

- Jeder Knoten aus G wird mit genau einer der c Farben gefärbt.
- Benachbarte Knoten sind unterschiedlich gefärbt.

Die hier betrachteten $c \sim$ Färbungen unterliegen keinen Einschränkungen.

b) $c(G)$ bezeichnet die Menge aller $c \sim$ Färbungen von G .

c) Eine (c, M) -**Farbkodierung** ist eine injektive Abbildung $K : M \rightarrow c(\mathbb{E})$, dabei ist M eine beliebige Menge, $c \in \mathbb{N}$ und \mathbb{E} der Graph, der aus einem einzelnen Knoten besteht.

OBdA beginne w mit c monadisch existentiellen SO-Quantoren. Spoiler färbt also S_0 (und damit L_i^0) mit c Farben. Sei \mathfrak{C}_0 die von Spoiler gewählte Färbung. Duplikator übersetzt diese Färbung zunächst in eine Färbung von A mit $k := c(E^{+1}, F^{+1})$ Farben.

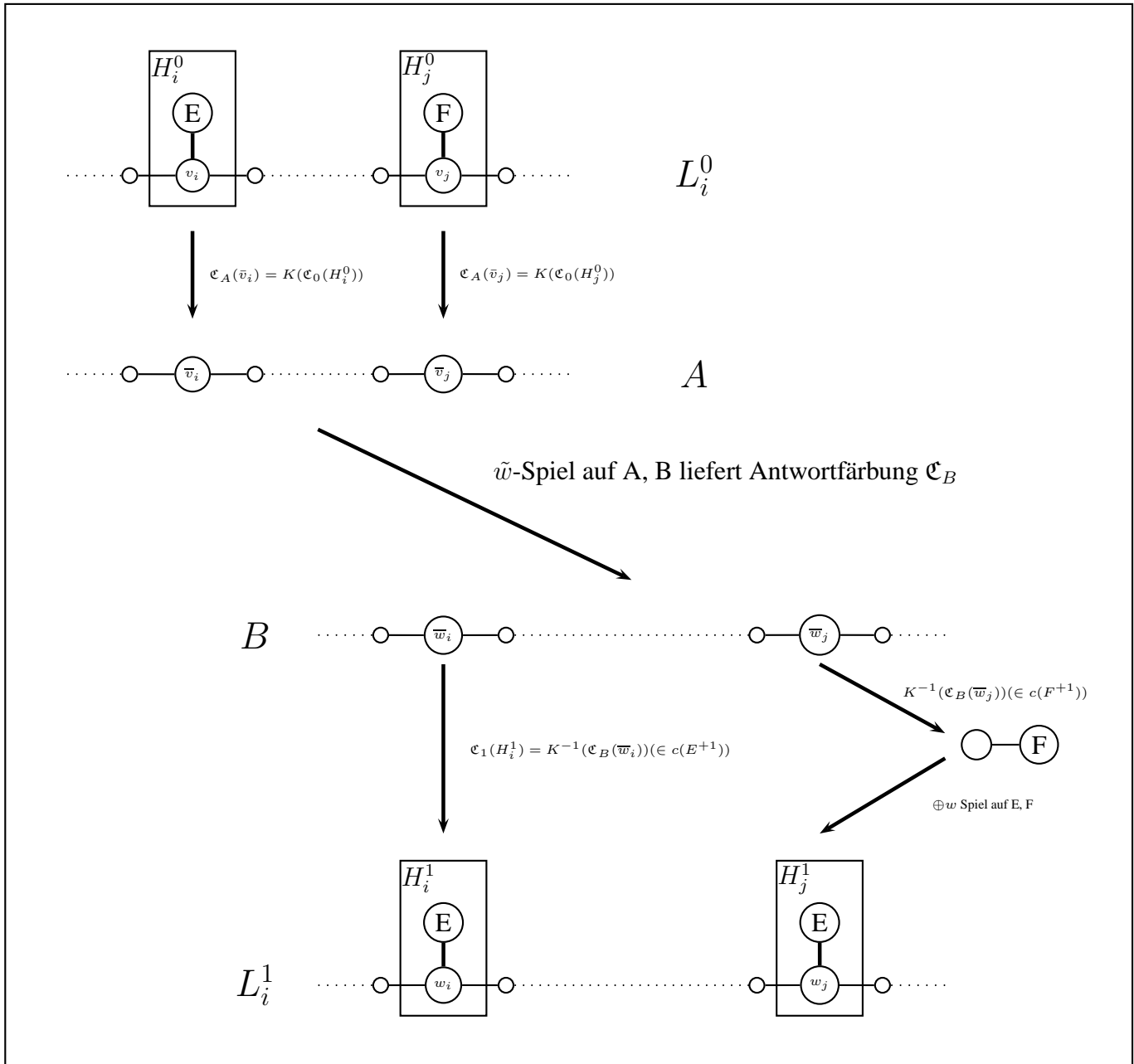
k ist so gewählt, daß eine $(k, c(E^{+1}) \cup c(F^{+1}))$ -Farbkodierung $K : c(E^{+1}) \cup c(F^{+1}) \rightarrow k(\mathbb{E})$ existiert. Insbesondere gilt $K(c_1(E^{+1})) \cap K(c(F^{+1})) = \emptyset$. Sei $H_i^0(H_i^1)$ der Teilgraph von $L_i^0(L_i^1)$, der aus dem Knoten v_i und dem an v_i angehängten Baustein besteht. Duplikator färbt nun $\bar{v}_i \in A$ mit $K(\mathfrak{C}_0(H_i^0))$. Da Duplikator das $\oplus \tilde{w}$ -Spiel auf A, B gewinnt, findet er für diese spezielle Färbung \mathfrak{C}_A von A eine Antwortfärbung \mathfrak{C}_B für B . Diese liefert ihm eine Antwortfärbung \mathfrak{C}_1 für L_i^1 . Die Teilgraphen H_i^1 werden dabei wie folgt gefärbt (oBdA sei $H_i^1 \cong E^{+1}$):

1. Fall: $K^{-1}(\mathfrak{C}_B(\bar{w}_i)) \in c(E^{+1}) \Rightarrow \mathfrak{C}_1(H_i^1) = K^{-1}(\mathfrak{C}_B(\bar{w}_i))$.

D.h. falls die Farbe von \bar{w}_i eine Färbung eines E^{+1} Graphen kodiert, so färbt Duplikator H_i^1 mit dieser Farbe.

2. Fall: $K^{-1}(\mathfrak{C}_B(\bar{w}_i)) \in c(F^{+1})$. Die Färbung von \bar{w}_i kodiert eine $c \sim$ Färbung von F^{+1} . H_i^1 ist aber (oBdA) ein E^{+1} Komponente. Duplikators $\oplus \tilde{W}$ Gewinnstrategie auf E, F liefert nun eine Gewinnfärbung von E bzgl. der durch \bar{w}_i kodierten Färbung von F . Diese nimmt Duplikator zum Färben von H_i^1 .

Duplikators Strategie für die erste Färbungsphase ist nun vollständig beschrieben. Die folgenden Färbungsrunden bestreitet Duplikator ganz analog. Folgende Skizze verdeutlicht Duplikators Färbungsstrategie.



Die FO-Runden verlaufen nach demselben Prinzip. Wählt etwa Spoiler einen Knoten v aus L_i^0 und sei $v_l = R(v)$ der Radknoten von v , dann wählt Duplikator den Knoten \bar{v}_l in A und findet einen Antwortknoten \bar{w}_p in B . Nun antwortet er in L_i^1 mit einem Knoten aus H_p^1 . Dabei muß er folgende Fälle unterscheiden (oBdA sei H_l^0 ein E^{+1}

Teilgraph):

1. Fall: H_p^1 ist ein E^{+1} Teilgraph \rightarrow Duplikator spielt isomorph auf H_p^1 .
2. Fall: H_p^1 ist ein F^{+1} Teilgraph \rightarrow Duplikator verwendet seine $\oplus w$ Strategie auf E, F , um einen Antwortknoten auf H_p^1 zu bestimmen.

Es ist *leicht* zu sehen, daß damit eine w -Gewinnstrategie auf L_i^0, L_i^1 beschrieben ist. Diese *addiert* sich zu einer Gewinnstrategie auf S_0 und S_1 (wichtig ist dabei, daß Duplikators Gewinnstrategie auf A, B endpunktrespektierend ist). Damit ist Teil a) des Satzes bewiesen. \square

Beweis von Satz 5.6 b):

Zu zeigen ist, daß falls \mathcal{P} und $\overline{\mathcal{P}}$ in cmonNP ausdrückbar sind, auch \mathcal{P}_g in cmonNP ausdrückbar ist. Der für den Spezialfall $W = \text{monNP}$ von Fagin et al. in [3] präsentierte Beweis kann hier direkt übernommen werden. \square

Kapitel 6

Färbbarkeitsprobleme

Auf der Suche nach einem monadischen Problem, das nicht in der zweiten Stufe von closed monadic NP ausdrückbar ist, kann man zwei Wege beschreiten. J.Marcinkowski hat gezeigt, daß Erreichbarkeit in gerichteten Graphen nicht in PFO(monNP) ausdrückbar ist. Es ist aber nicht schwer zu zeigen, daß dieses Problem in der zweiten Stufe von cmonNP liegt. Dennoch kann man natürlich versuchen, aufbauend auf das Problem *Erreichbarkeit in gerichteten Graphen* ein neues komplexeres Problem zu beschreiben und aus *intuitiven* Gründen die Vermutung äußern, daß dieses nicht in der zweiten Stufe von cmonNP liegt. Ein solches Problem wurde mir von J.Marcinkowski geschildert, und ich bin nicht abgeneigt mich seiner Intuition anzuschließen. Nur ist es bisher nicht gelungen, diese Vermutung zu beweisen.

Eine zweite Möglichkeit ein geeignetes monadisches Problem zu finden, das mit *großer Sicherheit* nicht in der zweiten Stufe von cmonNP (in Wahrheit sogar mit *großer Sicherheit* nicht in cmonNP) liegt, ist sich auf komplexitätstheoretische Vermutungen zu stützen. Die Tatsache, daß $NP \neq co-NP$ ist zwar nicht bewiesen, aber *weitläufig akzeptiert*. Daher wird man kein NP-vollständiges Problem finden, dessen Komplement wiederum in NP liegt. Bekanntermaßen ist k -Färbbarkeit (kC) von Graphen für $k > 2$ NP-vollständig. Es ist also zu erwarten, daß nicht- k -Färbbarkeit ($nonkC$) von Graphen ($k > 2$) nicht in NP liegt. Da schließlich $cmonNP \subset NP$, ist zu erwarten, daß nicht- k -Färbbarkeit nicht in cmonNP ausdrückbar ist. Andererseits ist $nonkC$ für alle $k \in \mathbb{N}$ leicht in co-monNP zu beschreiben und daher monadisch beschreibbar.

Ich habe mich für diesen Weg entschieden und gezeigt, daß nicht- k -Färbbarkeit ($k > 2$) nicht in PFO(monNP) ausgedrückt werden kann. Im Unterschied zu den bisherigen Ergebnissen bzgl. PFO(monNP) von Fagin und Marcinkowski ([3] und [16]), die jeweils Probleme betrachtet haben, die sofort in der zweiten Stufe von cmonNP liegen, kann (wie oben beschrieben) erwartet werden, daß $nonkC$ nicht in cmonNP ausdrückbar ist. Vielleicht ist es mit weiteren Ideen, unter Verwendung der bisherigen, möglich dies auch zu zeigen.

Am Ende dieses Kapitels wird der folgende Satz bewiesen sein.

Satz 6.1

Nicht- k -Färbbarkeit ($k \geq 3$) ist nicht ausdrückbar in PFO(monNP).

Satz 6.1 wird zunächst für den Spezialfall $k = 4$ bewiesen. Anschließend wird die Verallgemeinerung für $k > 4$ grob skizziert und der Fall $k = 3$ durch Reduktion erledigt.

6.1 Beweis von Satz 6.1 für $k = 4$

Eine der Hauptideen, die im Beweis von Satz 6.1 verwendet wird, besteht darin zwei Spielstrukturen S_0, S_1 zu konstruieren, die jeweils aus vielen Kopien zweier unterschiedlicher Teilstrukturen (hier: G_0, G_1) bestehen. Dadurch hat Duplikator die Möglichkeit, den ersten Spielabschnitt (r FO-Runden) so zu bestreiten, daß er sich in der jeweiligen Struktur eine geeignete Kopie von G_0 oder G_1 sucht und in dieser denselben Knoten auswählt, wie ihn vorher Spoiler in der anderen Struktur gewählt hat. Diese Idee ist schon in den Beweisen von Marcinkowski und Fagin et al. zu finden.

Die hier benötigten Strukturen G_0, G_1 werden folgende Eigenschaften haben:

1. Duplikator gewinnt das (c, \bar{r}) -monNP Spiel auf G_0, G_1 . Dabei kann $(c, \bar{r}) \in \mathbb{N}^2$ beliebig festgelegt werden.
2. $G_0 \in \text{non3C}$, $G_1 \in \text{3C}$.
3. $G_0 \in \text{4C}$ und G_0 enthält ein Dreieck.

Der Beweis der Existenz solcher Strukturen wird im ersten der folgenden vier Konstruktionschritte der Spielstrukturen S_0, S_1 gegeben:

- 1.Schritt: Beweis der Existenz von G_0 und G_1 mit obigen Eigenschaften.
- 2.Schritt: G_0 und G_1 werden um jeweils vier Knoten und geeignete Kanten erweitert. Diese vier Knoten dienen als Schnittstellen zum Verbinden zweier Bausteine.
- 3.Schritt: Es werden drei Verbindungsstrukturen (Adapter) definiert, durch die jeweils die Schnittstellen von zwei Bausteinen verbunden werden können.
- 4.Schritt: Es werden induktiv Strukturen S_0^r, S_1^r konstruiert mit folgenden Eigenschaften:
 - $S_0^r \in \text{non4C}$, $S_1^r \in \text{4C}$.
 - Duplikator gewinnt das $(\exists \forall)^r(c-4, \bar{r})$ -PFO(monNP) Spiel auf S_0^r, S_1^r . Dabei sind c, \bar{r} die oben gewählten Parameter.

Satz 6.1 für dann Fall $k = 4$ folgt dann mit Korollar 2.23.

1.Schritt: Existenz von G_0, G_1

Der Existenzbeweis für G_0, G_1 wird durch eine direkte Reduktion 1. Ordnung des Graphproblems

$$\mathcal{Z} = \text{Graph } G \text{ ist zusammenhängend}$$

auf Graphstrukturen mit einer eingebauten built-in Nachfolgerrelation auf das Problem

$$\mathcal{C} = \text{Der Graph } G \text{ ist 4-färbbar, aber nicht 3-färbbar}$$

gegeben. Dazu zunächst die Definition für Reduktion 1. Ordnung in der Form, wie sie hier benötigt wird (für die allgemeine Definition, siehe z.B. Schwentick [21]).

Definition 6.2

Seien $S = (E_S, Succ), T = E_T$ Signaturen, wobei E_S, E_T zweistellige, symmetrische Kantenrelationen und $Succ$ eine Nachfolgerrelation bezeichnet. Sei \mathcal{Z} eine Menge von S -Strukturen und \mathcal{C} eine Menge von T -Strukturen.

Eine **FO-Reduktion 1.Ordnung** von \mathcal{Z} nach \mathcal{C} ist ein Tupel $(k, \overline{\varphi^0}, \overline{\varphi})$, wobei

- $k \in \mathbb{N}$,
- $\overline{\varphi^0} = (\varphi_1^0, \dots, \varphi_k^0)$ ist ein Tupel von FO-Formeln über S mit einer freien Variable, und
- $\overline{\varphi} = (\varphi_{11}, \dots, \varphi_{kk})$ ist ein Tupel von FO-Formeln über S mit zwei freien Variablen,

so daß für jede S -Struktur \mathcal{A} wie folgt eine T -Struktur \mathcal{B} definiert wird:

- Der Grundraum (Universum) von \mathcal{B} ist die Menge aller Paare $(x, j), x \in \mathcal{U}^{\mathcal{A}}, j \leq k$, für die gilt $\langle \mathcal{A}, x \rangle \models \varphi_j^0$.
- Für alle Tupel $(x_1, x_2) \in (U^{\mathcal{A}})^2$ und $(i, j) \in \{1, \dots, k\}^2$ gilt

$$E_T((x_1, i), (x_2, j)) \iff \langle \mathcal{A}, x_1, x_2 \rangle \models \varphi_{ij}.$$

- $\mathcal{A} \in \mathcal{Z} \iff \mathcal{B} \in \mathcal{C}$.

Im Folgenden bezeichnet **Red**(\mathcal{A}), die durch Reduktion von \mathcal{A} erhaltene Struktur \mathcal{B} .

Folgende Abschlußeigenschaft ist entscheidend für die Nützlichkeit von Reduktionen 1. Ordnung im Zusammenhang mit monNP:

Satz 6.3 (Cosmadakis 93)

MonNP ist abgeschlossen unter FO-Reduktionen 1.Ordnung.

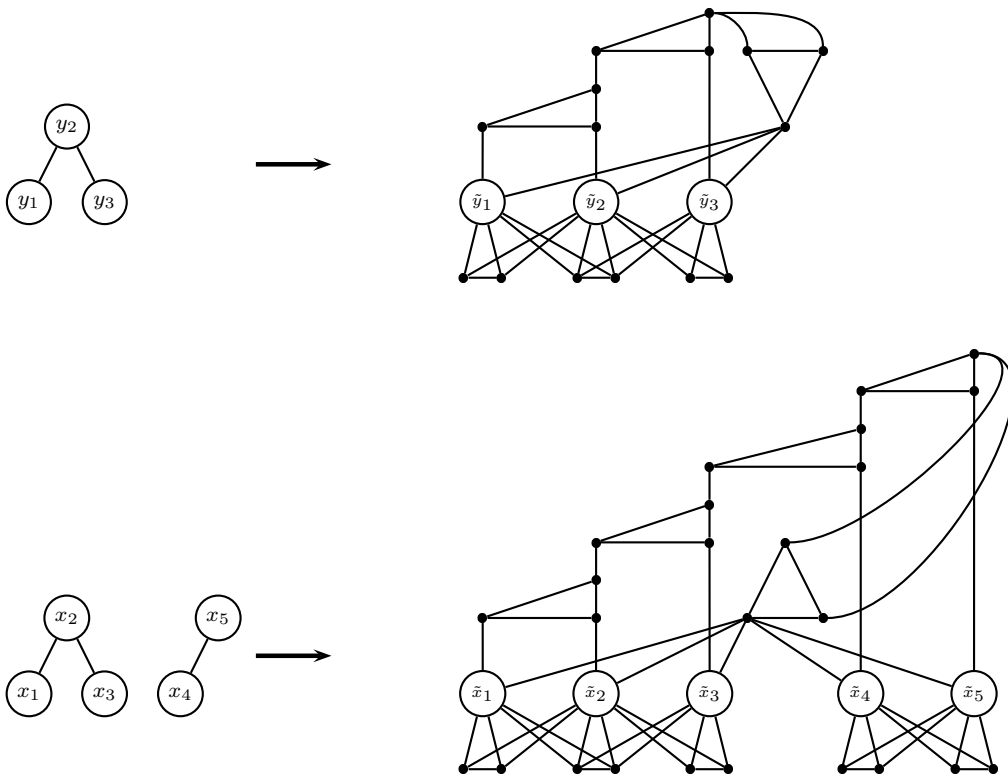
Beweis: siehe [5].

Es ist bekannt, daß die Eigenschaft \mathcal{Z} in monNP+Succ nicht ausdrückbar ist. Im weiteren Verlauf dieses Abschnittes wird eine Reduktion von Graphen beschrieben, derart daß folgende Äquivalenz gilt

$$G \in \mathcal{Z} \iff \text{Red}(G) \in \mathcal{C}. \quad (6.1)$$

D.h. der Graph G ist genau dann zusammenhängend, wenn das Redukt von G nicht 3-färbbar ist, aber 4-färbbar. Daraus folgt, daß die Eigenschaft \mathcal{C} in monNP nicht ausgedrückt werden kann.

Die folgende Skizze beschreibt zum einen die Struktur des Reduktes einer Kette mit drei Punkten, zum anderen die eines Graphen, der aus einer dreipunktigen Kette und zwei miteinander verbundenen Punkten besteht.



Die Äquivalenz (6.1) ergibt sich daraus, daß $\text{Red}(G)$ so konstruiert ist, daß die zu einer Zusammenhangskomponente des Ursprungsgraphen G gehörenden Tildeknoten in einer 3-Färbung von $\text{Red}(G)$ identisch gefärbt werden müssen (dies wird garantiert durch die Knotenreihe unter den Tildeknoten). $\text{Red}(G)$ ist aber nur dann 3-färbbar, wenn man zwei Tildeknoten verschieden färben kann (dies wird garantiert durch die Knoten über den Tildeknoten). Ansonsten ist $\text{Red}(G)$ auf jeden Fall 4-färbbar.

3-Färbbarkeit von $\text{Red}(G)$ erzwingt also mindestens zwei Zusammenhangskomponen-

ten im ursprünglichen Graphen G .

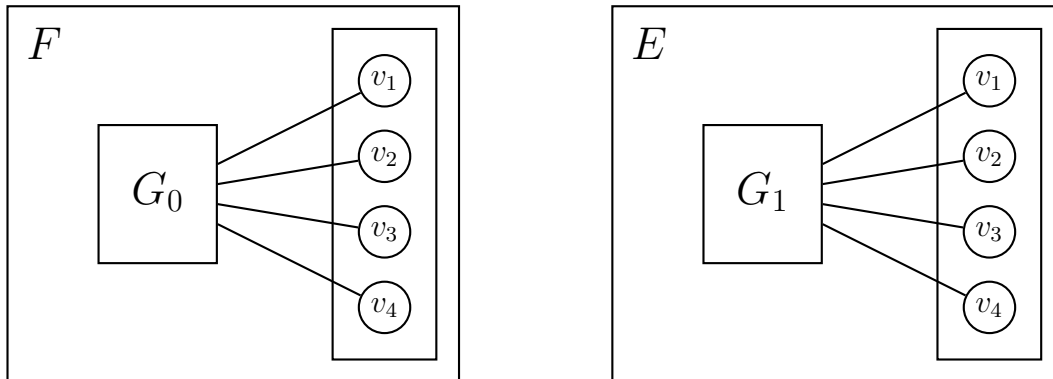
Die folgende Tabelle gibt abschließend eine formale Definition der hier verwendeten Reduktion:

- $k = 6$
- $\varphi_i^0 \equiv (x = x) \forall i \in \{1, \dots, 6\}$

$(i, j) \in \{1, \dots, 6\}^2$	$\varphi_{ij}(x, y)$
(1,2),(2,1),(1,3),(3,1)	$(x = y) \vee E_S(x, y)$
(1,4), (4,1)	$(x = y) \wedge (x \neq \text{Min})$
(1,5), (5,1)	$(x \neq x)$
(1,6)	$y = \text{Min}$
(6,1)	$x = \text{Min}$
(2,3), (3,2)	$x = y$
(2,4),(4,2),(2,5),(5,2),(2,6),(6,2)	$x \neq x$
(3,4),(4,3),(3,5),(5,3),(3,6),(6,3)	$x \neq x$
(4,5), (5,4)	$x = y$
(4,6)	$(x = y) \vee (x = \text{Min} \wedge y = \text{Max})$
(6,4)	$(x = y) \vee (x = \text{Max} \wedge y = \text{Min})$
(5,6)	$(x = y) \vee (x = \text{Min} \wedge y = \text{Max}) \vee (x = \text{Succ}(y))$
(6,5)	$(x = y) \vee (x = \text{Max} \wedge y = \text{Min}) \vee (y = \text{Succ}(x))$

2.Schritt: Schnittstellen

Zur Konstruktion der Strukturen S_0^r, S_1^r wird es notwendig sein, Graphen (z.B. G_0 mit G_1) zu verbinden. Dazu werden die Graphen G_0, G_1 um jeweils vier Knoten v_1, v_2, v_3, v_4 erweitert. Diese vier Knoten dienen dann als Schnittstelle und können mit Hilfe eines Adapters mit einer anderen Schnittstelle verbunden werden. Zunächst müssen aber die vier Knoten mit dem Ursprungsgraphen G_0 oder G_1 verbunden werden. Die so konstruierten Graphen bekommen die Namen F und E .



Bevor die Art der Verbindungen der Graphen G_0, G_1 mit der Schnittstelle (v_1, v_2, v_3, v_4) beschrieben wird, werden zunächst die *benötigten* Eigenschaften von F und E festgehalten. Dazu folgende Definitionen:

Definition 6.4

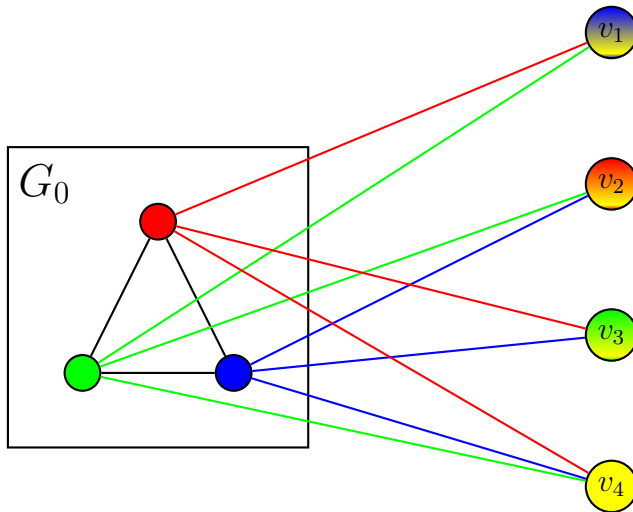
- (i) Sei H ein Graph mit vier ausgezeichneten Knoten (v_1, v_2, v_3, v_4) . Nenne H einen **Graph mit Schnittstelle** und die vier Knoten (v_1, v_2, v_3, v_4) **Schnittstelle** von H , kurz **S(H)**. Der Teilgraph von H , der entsteht, wenn man aus H die Schnittstelle entfernt, sei das **Modul** von H , kurz **M(H)**.
- (ii) Sei H ein Graph mit Schnittstelle $S(H)$. $S(H)$ nennt man **farbenreich**, wenn in jeder 4- F - \bar{a} rbung von H die Knoten aus $S(H)$ paarweise verschieden gef \bar{a} rbt sind und mindestens eine 4- F - \bar{a} rbung von H existiert.
- (iii) Sei H ein Graph mit Schnittstelle $S(H)$. $S(H)$ nennt man **einfarbig**, wenn es mindestens eine 4- F - \bar{a} rbung von H gibt, so da \bar{a} die Knoten aus $S(H)$ die gleiche Farbe haben und mindestens eine 4- F - \bar{a} rbung von H , so da \bar{a} die vier Schnittstellenknoten paarweise verschieden gef \bar{a} rbt sind.

F und E werden so konstruiert, da \bar{a} folgendes gilt:

- 0. $F \in 4C \setminus 3C, E \in 4C$.
- 1. $S(F)$ ist farbenreich.
- 2. $S(E)$ ist einfarbig.
- 3. Duplikator gewinnt das $(c - 4, \bar{r})$ -monNP Spiel auf F und E .

Konstruktion von \bar{F} :

Sei $G_0 = (V_0, E_0)$. Eine der Voraussetzungen an G_0 war, daß G_0 mindestens ein Dreieck enthält. Es wird nun ein beliebiges Dreieck aus G_0 gemäß folgender Skizze mit den Knoten (v_1, v_2, v_3, v_4) verbunden:



Seien blau, grün, rot und gelb die zur Verfügung stehenden Farben und das Dreieck aus G_0 sei oBdA wie oben gefärbt. Sei $\bar{F} = \{V_0 \cup \{v_1, v_2, v_3, v_4\}, \bar{E}_0\}$, wobei \bar{E}_0 alle Kanten aus E_0 plus die Kanten aus obiger Skizze enthält. Es ist klar, daß $\bar{F} \in 4C \setminus 3C$ und man sieht leicht ein, daß für jede beliebige 4-Färbung von \bar{F} folgende Bedingungen gelten:

- v_1 ist entweder blau oder gelb gefärbt.
- v_2 ist entweder rot oder gelb gefärbt.
- v_3 ist entweder grün oder gelb gefärbt.
- v_4 ist gelb gefärbt.

$S(\bar{F})$ ist also (noch) nicht farbenreich. Es sind weitere Kanten nötig, um dies sicher zu stellen.

Sei C_4 eine beliebige 4-Färbung von G_0 , welche die Knoten des Dreiecks gemäß obiger Skizze färbt. F entsteht aus \bar{F} durch Hinzufügen folgender Kanten:

- v_1 wird mit allen Knoten verbunden, welche in C_4 nicht blau gefärbt sind.
- v_2 wird mit allen Knoten verbunden, welche in C_4 nicht rot gefärbt sind.
- v_3 wird mit allen Knoten verbunden, welche in C_4 nicht grün gefärbt sind.

- v_4 wird mit allen Knoten verbunden welche in C_4 nicht gelb gefärbt sind.

Es wird nun gezeigt, daß $\overline{C}_4 := C_4 \cup \{v_1 = \text{blau}, v_2 = \text{rot}, v_3 = \text{grün}, v_4 = \text{gelb}\}$ die einzige 4-Färbung von F ist. Nach Konstruktion genügt es zu zeigen, daß in jeder 4-Färbung von F die Knoten v_1, v_2, v_3, v_4 paarweise verschieden gefärbt werden müssen (damit ist $S(F)$ farbenreich). Sei v_4 oBdA gelb gefärbt. Es reicht zu zeigen, daß keiner der Knoten v_1, v_2, v_3 gelb gefärbt sein kann.

Jeder Knoten aus $F \setminus \{v_1, v_2, v_3, v_4\}$, der nicht mit v_4 verbunden ist, ist nach Konstruktion mit v_1, v_2 und v_3 verbunden. Da G_0 nach Voraussetzung nicht-3-färbbar ist, muß einer dieser Knoten mit der Farbe gelb gefärbt werden. Damit kann keiner der Knoten v_1, v_2, v_3 gelb gefärbt werden.

Insbesondere ist $F \in 4C \setminus 3C$ und $S(F)$ farbenreich.

Konstruktion von E :

Sei $G_1 := (V_1, E_1)$. Auch E entsteht aus G_1 durch Erweiterung um eine Schnittstelle (w_1, w_2, w_3, w_4) . Im Gegensatz zur Konstruktion von F liegt in diesem Fall die Schwierigkeit weniger in der Schnittstellenbedingung, als darin geeignete Verbindungskanten zu definieren, so daß Duplikator das $(c - 4, \bar{r})$ -monNP Spiel auf F und E gewinnt. Dies wird durch einen kleinen Trick realisiert.

Eine Voraussetzung von G_0, G_1 war, daß Duplikator das (c, \bar{r}) -monNP Spiel auf diesen Strukturen gewinnen kann. In diesem Spiel werden nun vier Färbungsrunden gespielt. G_0 wird mit den Farben F_1, F_2, F_3, F_4 gefärbt. Ein Knoten $x \in V_0$ wird genau dann mit der Farbe F_i gefärbt, wenn F die Kante (x, v_i) enthält. Diese Färbung beschreibt die Verbindung von G_0 mit der Schnittstelle $S(F)$ in eindeutiger Weise. Duplikator besitzt nun eine Gewinnfärbung auf G_1 . Diese beschreibt nun in eben dieser eindeutigen Weise die Verbindungen von G_1 mit der Schnittstelle $S(E)$. Formal bedeutet das:

$$E = (V_1 \cup \{w_1, w_2, w_3, w_4\}, E_E) \text{ mit}$$

$$\begin{aligned} E_E = E_1 \quad \cup \quad & \{(x, w_1), (w_1, x); x \in F_1\} \\ & \cup \quad \{(x, w_2), (w_2, x); x \in F_2\} \\ & \cup \quad \{(x, w_3), (w_3, x); x \in F_3\} \\ & \cup \quad \{(x, w_4), (w_4, x); x \in F_4\}. \end{aligned}$$

Das so konstruierte E erfüllt die notwendige Spieleigenschaft. Dieses wird im folgenden Satz präzisiert:

Satz 6.5

Die oben konstruierten Graphen F, E mit Schnittstellen besitzen die folgenden Eigenschaften:

- (i) $F \in 4C \setminus 3C$, $S(F)$ ist farbenreich.

(ii) Duplikator gewinnt das $(c - 4, \bar{r})$ -monNP Spiel auf F und E . Duplikators Gewinnstrategie läßt dabei die Schnittstellenknoten konstant.

(iii) $E \in 4C$, $S(E)$ ist einfarbig.

Beweis:

Seien E, F die oben konstruierten Graphen.

zu (i): Folgt direkt aus der Konstruktion von F .

zu (ii): Es wird Duplikators Gewinnstrategie im $(c - 4, \bar{r})$ -monNP Spiel beschrieben.

Bevor das eigentliche Spiel beginnt, färbt Duplikator das Modul $M(F)$ von F mit den Farben F_1, F_2, F_3, F_4 wie oben beschrieben gemäß den Verbindungen von $M(F)$ mit der Schnittstelle $S(F)$. Diese Färbung beantwortet er sich selbst auf dem Modul $M(E)$ von E gemäß seiner Gewinnstrategie im (c, \bar{r}) -monNP Spiel auf G_0, G_1 . Die Situation ist nun die, daß die Module von F und E mit vier Farben gefärbt sind und Duplikator weiterhin das $(c - 4, \bar{r})$ -monNP Spiel auf diesen Modulen gewinnen kann.

Nun beginnt das eigentliche $(c - 4, \bar{r})$ -monNP Spiel. Hier verfolgt Duplikator auf den Modulen und den Schnittstellen jeweils unterschiedliche Strategien. Auf den Modulen spielt er gemäß seiner $(c - 4, \bar{r})$ -monNP Gewinnstrategie. Auf den Schnittstellen spielt er konstant. Es ist *leicht* zu sehen, daß dies eine Gewinnstrategie auf F, E ist. Die einzige Schwierigkeit sind die Verbindungen von Modulknoten zu Schnittstellenknoten. Durch die anfängliche 4-Färbung wird garantiert, daß Duplikator immer mit einem Modulknoten antworten kann, der sich bezüglich aller Schnittstellenknoten so verhält, wie der von Spoiler gewählt. \square

zu (iii): Man sieht leicht, daß es eine 4-Färbung von E gibt, so daß die Schnittstelle $S(E)$ mit einer Farbe gefärbt ist (G_1 ist 3-färbbar). Die zweite Bedingung für eine einfarbige Schnittstelle ist die Existenz einer 4-Färbung von E , so daß die vier Schnittstellenknoten paarweise verschieden gefärbt sind. Diese folgt aus der Gewinnstrategie auf F, E (siehe (ii)): Spoiler färbt F in vier Runden gemäß einer 4-Färbung. Hierbei sind die Schnittstellenknoten paarweise verschieden gefärbt. Duplikator findet darauf eine Antwort auf E . Es ist klar, daß diese Antwort ebenfalls eine 4-Färbung sein muß. Da Duplikators Gewinnstrategie die Schnittstellenknoten konstant läßt, muß diese 4-Färbung von E die Schnittstellenknoten von E paarweise verschieden färben.

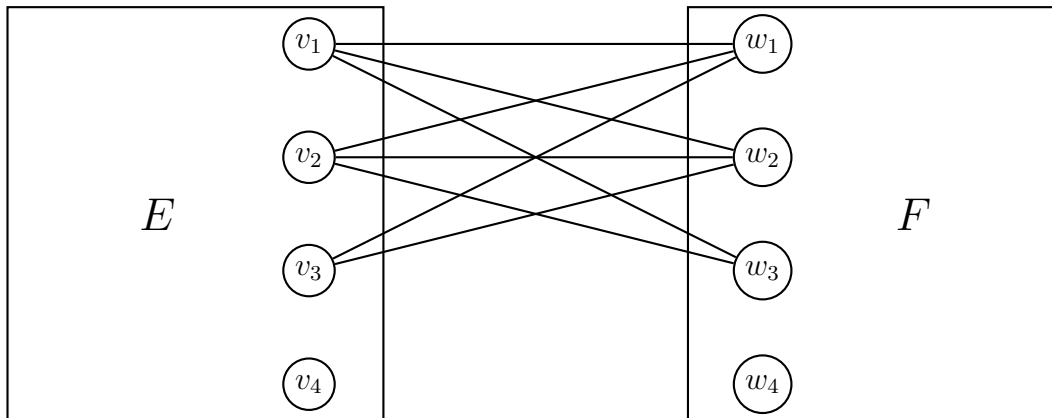
3.Schritt: Adapter

Die Konstruktion der Strukturen S_0^r, S_1^r , die als Spielstrukturen im PFO(monNP)-Spiel verwendet werden, erfordert es, zwei Graphen E und F mit Schnittstellen miteinander zu verbinden. Der neue verbundene Graph $E \star F$ soll wiederum ein Graph mit

Schnittstelle sein. In diesem Abschnitt werden drei verschiedene Verbindungsadapter definiert, die jeweils verschiedene Eigenschaften der Schnittstelle von $E \star F$ liefern.

Adapter A

Folgende Skizze beschreibt die Struktur von Adapter A:



Das Zeichen \star_X steht für die Verknüpfung zweier Graphen E, F mit Schnittstelle mittels Adapter X , d.h. $E \star_A F$ ist der Graph, den man erhält, wenn man E mittels Adapter A mit F verbindet. Aufgrund der Symmetrie von Adapter A ist der Operator \star_A kommutativ. Alle hier betrachteten Adapter werden kommutativ sein.

Folgende Eigenschaften des Operators \star_A werden wichtig sein und sind leicht zu zeigen:

Lemma 6.6

Seien E, F zwei Graphen mit Schnittstelle. Dann gilt:

(A1) Sind $S(E)$ und $S(F)$ farbenreich, so ist $E \star_A F$ nicht-4-färbbar.

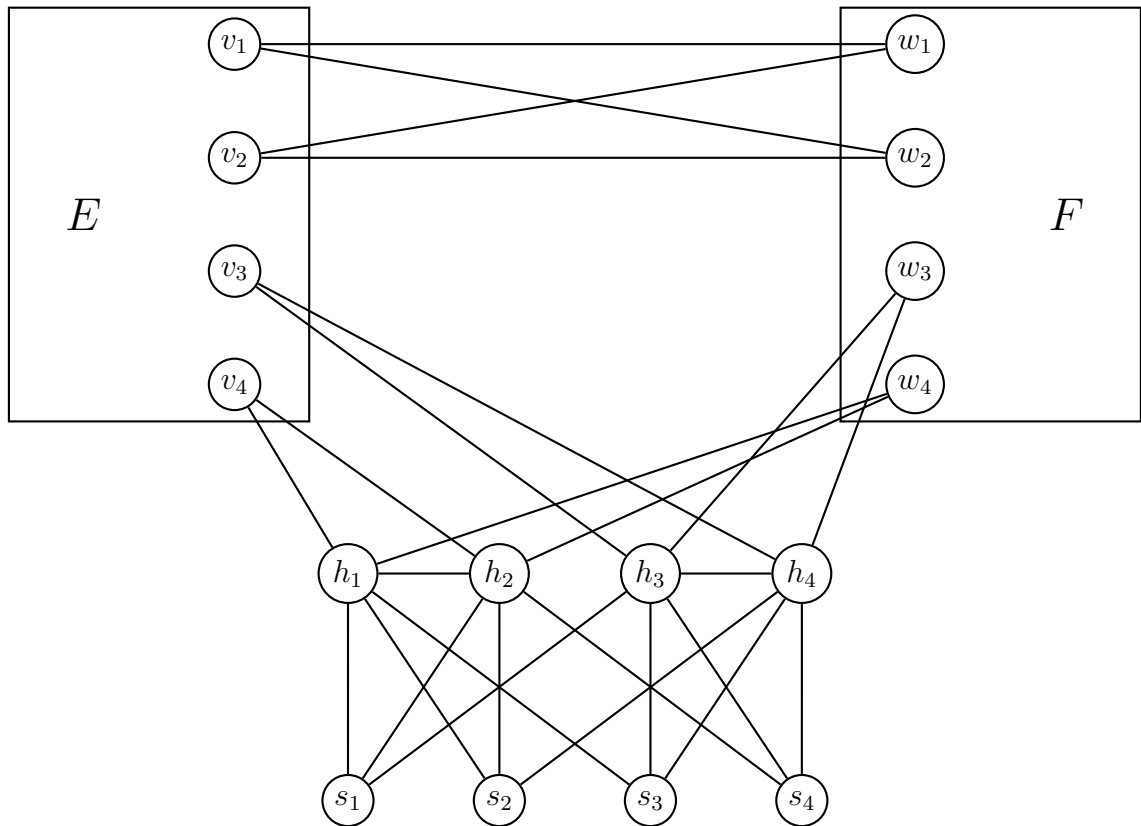
(A2) Ist $S(E)$ oder $S(F)$ einfarbig, so ist $E \star_A F$ 4-färbbar.

Bemerkung 6.7

Im weiteren Verlauf wird es nicht notwendig sein, daß der Graph $E \star_A F$ eine Schnittstelle hat. Ohne Einschränkung kann aber die Schnittstelle des linken Graphen (E) als Schnittstelle von $E \star_A F$ definiert werden.

Adapter B

Folgende Skizze beschreibt die Struktur von Adapter B :



Adapter B erweitert die verknüpften Graphen um acht Knoten. Diese werden **Adapterknoten** genannt. Die neuen Knoten (s_1, s_2, s_3, s_4) bilden die Schnittstelle von $E \star_B F$. Die vier Knoten h_1, h_2, h_3, h_4 dienen lediglich als Hilfsknoten, um folgende an Adapter B geforderten Eigenschaften zu erreichen:

Lemma 6.8

Seien E, F zwei Graphen mit Schnittstelle. Dann gilt:

(B1) Sind $S(E)$ und $S(F)$ farbenreich, so ist $S(E \star_B F)$ ebenfalls farbenreich.

(B2) Ist $S(E)$ oder $S(F)$ einfarbig, so ist auch $S(E \star_B F)$ einfarbig.

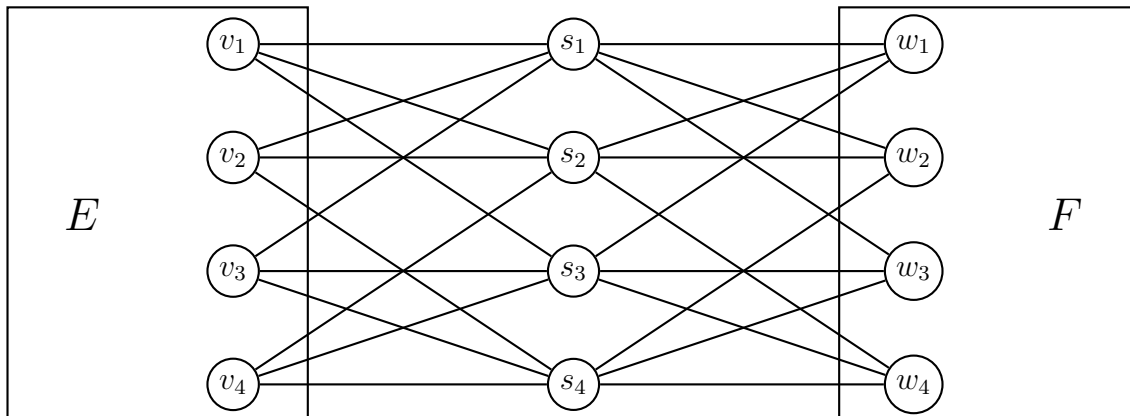
Beweisidee: (B1) Ist sowohl E als auch F farbenreich erzwingen die Verbindungen der Knoten v_1, v_2, w_1, w_2 , daß die Knoten v_3, v_4, w_3, w_4 in einer 4-Färbung paarweise verschieden gefärbt werden. Diese Eigenschaft vererbt sich auf die Knoten h_1, h_2, h_3, h_4 und dann weiter auf die Schnittstellenknoten s_1, s_2, s_3, s_4 .

(B2) Zunächst folgt wie in (B1), daß die Knoten so gefärbt werden können, daß s_1, s_2, s_3, s_4 paarweise verschieden gefärbt sind.

Sei oBdA E einfarbig. Man sieht leicht, daß es eine 4-Färbung gibt, die für die Knoten v_3, v_4, w_3, w_4 nur drei Farben benötigt. Dies vererbt sich auf die Knoten h_1, h_2, h_3, h_4 . Mit der noch freien Farbe können dann die Knoten s_1, s_2, s_3, s_4 gefärbt werden.

Adapter C

Folgende Skizze beschreibt die Struktur von Adapter C :



Die neuen Knoten (s_1, s_2, s_3, s_4) bilden die Schnittstelle von $E \star_C F$.

Folgende Eigenschaften des Operators \star_C werden wichtig sein und sind leicht zu zeigen:

Lemma 6.9

Seien E, F zwei Graphen mit Schnittstelle. Dann gilt:

- (C1) Ist $S(E)$ oder $S(F)$ farbenreich, so ist $S(E \star_C F)$ ebenfalls farbenreich.
- (C2) Sind $S(E)$ und $S(F)$ einfarbig, so ist auch $S(E \star_C F)$ einfarbig.

4.Schritt: Konstruktion von S_0^r und S_1^r

In diesem Abschnitt werden die Spielgraphen S_0^r und S_1^r induktiv konstruiert. Diese Spielgraphen erfüllen die im folgenden Satz beschriebenen Eigenschaften.

Satz 6.10

- (i) Für alle $r \in \mathbb{N}$ ist \mathfrak{S} nicht-4-färbbar.
- (ii) Für alle $r \in \mathbb{N}$ ist \mathfrak{S} 4-färbbar.
- (iii) Für alle $r \in \mathbb{N}$ gewinnt Duplikator das $(\exists \forall)^*(c - 4, \bar{r})$ -PFO(monNP) Spiel auf S_0^r, S_1^r .

Da $(c, \bar{r}) \in \mathbb{N}^2$ zu Beginn beliebig gewählt werden konnte, folgt sofort, daß die Grapheneigenschaft nicht-4-Färbbarkeit in PFO(monNP) nicht ausdrückbar ist. Teil (i) und (ii) des Satzes wird sich direkt aus der Konstruktion und den Eigenschaften der \star Operatoren ergeben.

Zunächst wird ein weiterer Operator $\star_{X \rightarrow Y}$ benötigt. Dazu folgende Definition:

Definition 6.11

Seien $K_i, i \in \mathbb{N}$, Graphen mit Schnittstelle. Sei G ein Graph, der als Produkt einer endlichen Teilmenge von $\{K_i; i \in \mathbb{N}\}$ darstellbar ist. Dabei sind die Operatoren $\star_A, \star_B, \star_C$ als Produktoperator zulässig. Dann ist $\star_{X \rightarrow Y} G$ derjenige Graph, der entsteht, wenn man in der Produktdarstellung von G den Operator \star_X durch den Operator \star_Y ersetzt.

Beispiel: Seien K_1, K_2 Graphen mit Schnittstelle. G sei definiert durch

$$G := (K_1 \star_C K_2) \star_A (K_1 \star_C K_2).$$

Dann gilt:

$$\star_{A \rightarrow B} G := (K_1 \star_C K_2) \star_B (K_1 \star_C K_2).$$

Konstruktion von S_0^r und S_1^r :

Für $r = 0$ sei $S_0^0 = F \star_A F$ und $S_1^0 = E \star_A E$.

Seien nun S_0^r und S_1^r schon konstruiert. Dann seien

$$S_0^{r+1} := ((\star_{A \rightarrow B} S_0^r) \star_C (\star_{A \rightarrow B} S_1^r)) \star_A ((\star_{A \rightarrow B} S_0^r) \star_C (\star_{A \rightarrow B} S_1^r))$$

und

$$S_1^{r+1} := ((\star_{A \rightarrow B} S_0^r) \star_C (\star_{A \rightarrow B} S_1^r)) \star_A ((\star_{A \rightarrow B} S_1^r) \star_C (\star_{A \rightarrow B} S_0^r)).$$

Folgende Skizze erleichtert das Verständnis dieser Definitionen:

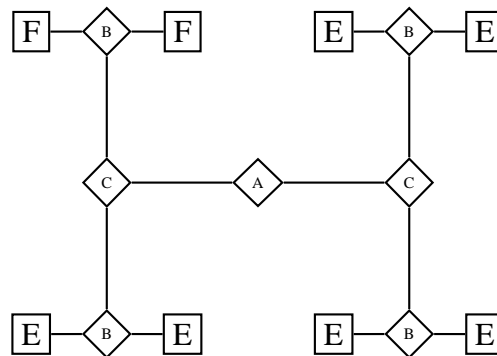
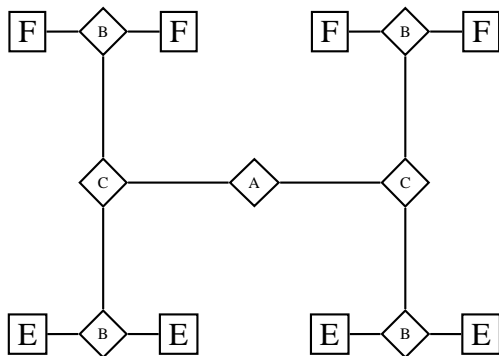
S_0^r

S_1^r

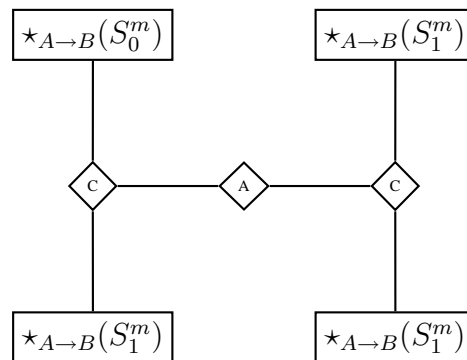
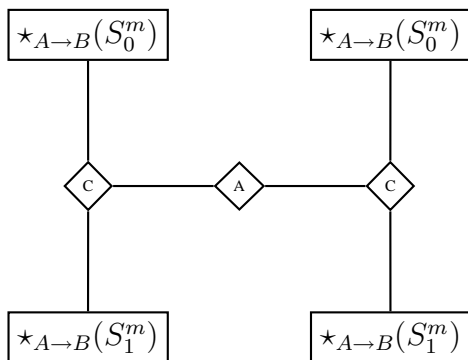
$r = 0 :$



$r = 1 :$



$r = m + 1 :$



Beweis von Satz 6.10:

zu (i),(ii): Aus folgendem Lemma über die Schnittstellen der Spielgraphen S_0^r, S_1^r ergibt sich die Behauptung *leicht*.

Lemma 6.12

- (1) Für alle $r \in \mathbb{N}$ ist $S(*_{A \rightarrow B} S_0^r)$ farbenreich.
- (2) Für alle $r \in \mathbb{N}$ ist $S(*_{A \rightarrow B} S_1^r)$ einfarbig.

Beweis:

(1) $r = 0$: Da $S(F)$ farbenreich ist, folgt aus Eigenschaft (B1) von \star_B sofort, daß $S(S_0^0) = S(F \star_B F)$ farbenreich ist.

$r \rightarrow r + 1$: Nach Definition ist

$$\star_{A \rightarrow B} S_0^{r+1} := ((\star_{A \rightarrow B} S_0^r) \star_C (\star_{A \rightarrow B} S_1^r)) \star_B ((\star_{A \rightarrow B} S_0^r) \star_C (\star_{A \rightarrow B} S_1^r)).$$

Nach (B1) ist S_0^{r+1} farbenreich, wenn $((\star_{A \rightarrow B} S_0^r) \star_C (\star_{A \rightarrow B} S_1^r))$ farbenreich ist. Nach (C1) gilt das, sobald entweder $(\star_{A \rightarrow B} S_0^r)$ oder $(\star_{A \rightarrow B} S_1^r)$ farbenreich ist. $(\star_{A \rightarrow B} S_0^r)$ ist nach Induktionsannahme farbenreich.

Der Beweis von (2) läuft analog. □

Da

$$S_0^{r+1} := ((\star_{A \rightarrow B} S_0^r) \star_C (\star_{A \rightarrow B} S_1^r)) \star_A ((\star_{A \rightarrow B} S_0^r) \star_C (\star_{A \rightarrow B} S_1^r)).$$

folgt nun nach (A1), daß S_0^{r+1} nicht 4-färbbar ist, falls $((\star_{A \rightarrow B} S_0^r) \star_C (\star_{A \rightarrow B} S_1^r))$ farbenreich ist. Nach (C1) ist das dann der Fall, wenn entweder $(\star_{A \rightarrow B} S_0^r)$ oder $(\star_{A \rightarrow B} S_1^r)$ farbenreich ist. $(\star_{A \rightarrow B} S_0^r)$ ist aber nach obigem Lemma farbenreich. Dies beweist (i).

Der Beweis von (ii) läuft analog.

zu (iii): Es wird gezeigt, daß Duplikator für alle $r \in \mathbb{N}$ eine Gewinnstrategie im $(\exists \forall)^r (c-4, \bar{r})$ -PFO(monNP) Spiel auf S_0^r, S_1^r hat, welche die beiden Schnittstellen, die durch den inneren A-Adapter verbunden sind, entweder konstant läßt oder miteinander vertauscht.

$r = 0$: Duplikator hat im $(c-4, \bar{r})$ -monNP-Spiel eine Gewinnstrategie auf F, E , welche die Schnittstellen konstant läßt (siehe Satz 6.5). Daraus ergibt sich sofort eine Gewinnstrategie auf $S_0^0 = (F \star_A F), S_1^0 = (E \star_A E)$, welche beide Schnittstellen konstant läßt.

$r \rightarrow r + 1$: Für den Induktionsschritt wird folgendes Lemma benötigt:

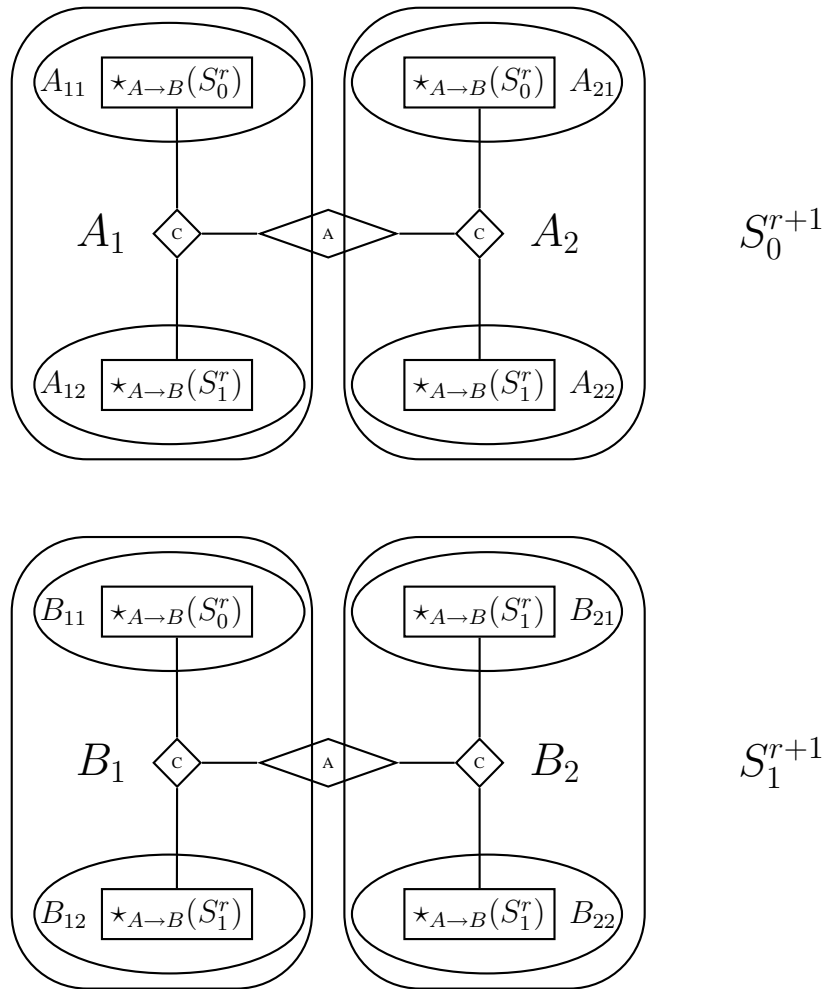
Lemma 6.13

Duplikator hat eine Gewinnstrategie im $(\exists \forall)^r (c-4, \bar{r})$ -PFO(monNP) Spiel auf $(\star_{A \rightarrow B} S_0^r), (\star_{A \rightarrow B} S_1^r)$, welche die Adapterknoten des neuen (innersten) B Adapters konstant läßt.

Beweis:

Der einzige Unterschied zwischen S_0^r, S_1^r und $(\star_{A \rightarrow B} S_0^r), (\star_{A \rightarrow B} S_1^r)$ ist der innere A-Adapter, der durch einen B-Adapter ersetzt wird. Da die Adapterknoten dieses B-Adapters symmetrisch zu beiden Schnittstellen sind, kann Duplikator sein Spiel auf S_0^r, S_1^r übernehmen und auf den Adapterknoten konstant spielen. □

Folgende Skizze ist hilfreich, um die ersten beiden Runden im $(\exists\forall)^{r+1}(c-4, \bar{r})$ -PFO(monNP) Spiel zu verstehen:



Die ersten zwei Spielrunden auf S_0^{r+1}, S_1^{r+1} spielt Duplikator nach folgender Strategie:

Runde 1: Spoiler wählt einen Knoten aus S_0^{r+1} . Wegen der Symmetrie kann dieser oBdA aus A_2 angenommen werden. Duplikator antwortet mit demselben Knoten aus B_1 ($A_2 \cong B_1$). Im weiteren Verlauf des Spiels spielt Duplikator auf A_2, B_1 immer konstant.

Runde 2: Spoiler wählt einen Knoten aus S_1^{r+1} .

Fall 1: Dieser ist aus B_1 . Dann antwortet Duplikator konstant auf A_2 (siehe oben).

Fall 2: Dieser ist ein Adapterknoten des (innersten) C -Adapters in B_2 . Dann antwortet Duplikator konstant (isomorph) auf A_1 .

Fall 3: Dieser ist oBdA aus B_{21} . Dann antwortet Duplikator mit demselben Knoten aus A_{12} ($A_{12} \cong B_{21}$) und im weiteren Verlauf des Spiels spielt Duplikator auf A_{12}, B_{21} konstant.

Der Rest des Spieles ist auf den Bereichen A_2, A_{12}, B_1, B_{21} geklärt. Auf $A_{11} = \star_{A \rightarrow B}(S_0^r)$ und $B_{22} = \star_{A \rightarrow B}(S_1^r)$ spielt Duplikator seine Gewinnstrategie aus Lemma 6.13 aus.

Man sieht leicht, daß damit eine $(\exists\forall)^{r+1}(c-4, \bar{r})$ -PFO(monNP) Gewinnstrategie auf S_0^{r+1}, S_1^{r+1} gegeben ist, die die Schnittstellenknoten des A -Adapters entweder konstant läßt oder vertauscht. \square

6.2 Der Fall $k > 4$

Es ist nicht schwer zu zeigen, daß die Methoden für den Fall $k = 4$ auf den allgemeinen Fall $k \geq 4$ erweitert werden können. Diese Erweiterung wird im Folgenden knapp beschrieben.

Schritt 1: Im allgemeinen Fall, werden Graphstrukturen G_0^k, G_1^k mit folgenden Eigenschaften benötigt:

1. Duplikator gewinnt das (c, \bar{r}) -monNP Spiel auf G_0^k, G_1^k . Dabei kann $(c, \bar{r}) \in \mathbb{N}^2$ beliebig festgelegt werden.
2. $G_0^k \in \text{non}(k-1)\mathbf{C}$, $G_1^k \in (k-1)\mathbf{C}$
3. $G_0^k \in k\mathbf{C}$ und G_0^k enthält einen K_{k-1} .

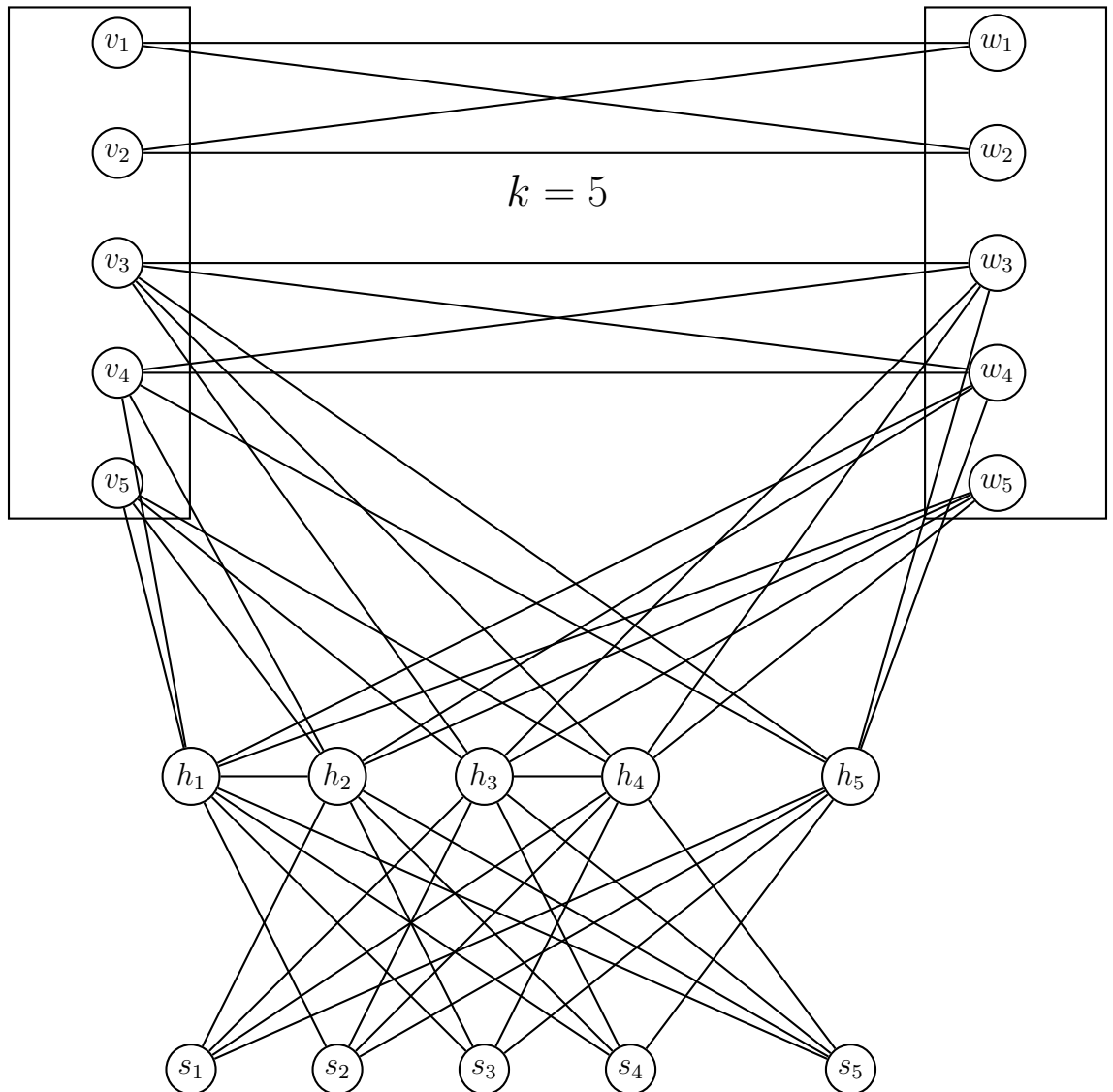
Diese lassen sich aus den im Falle $k=4$ verwendeten Graphen G_0, G_1 konstruieren. Dazu werden diese um jeweils einen Punkt erweitert, der mit allen anderen verbunden wird. Dadurch verändert sich nichts an den Spieleigenschaften, sehr wohl wird aber die notwendige Farbenzahl um eins erhöht. Dies kann induktiv fortgesetzt werden.

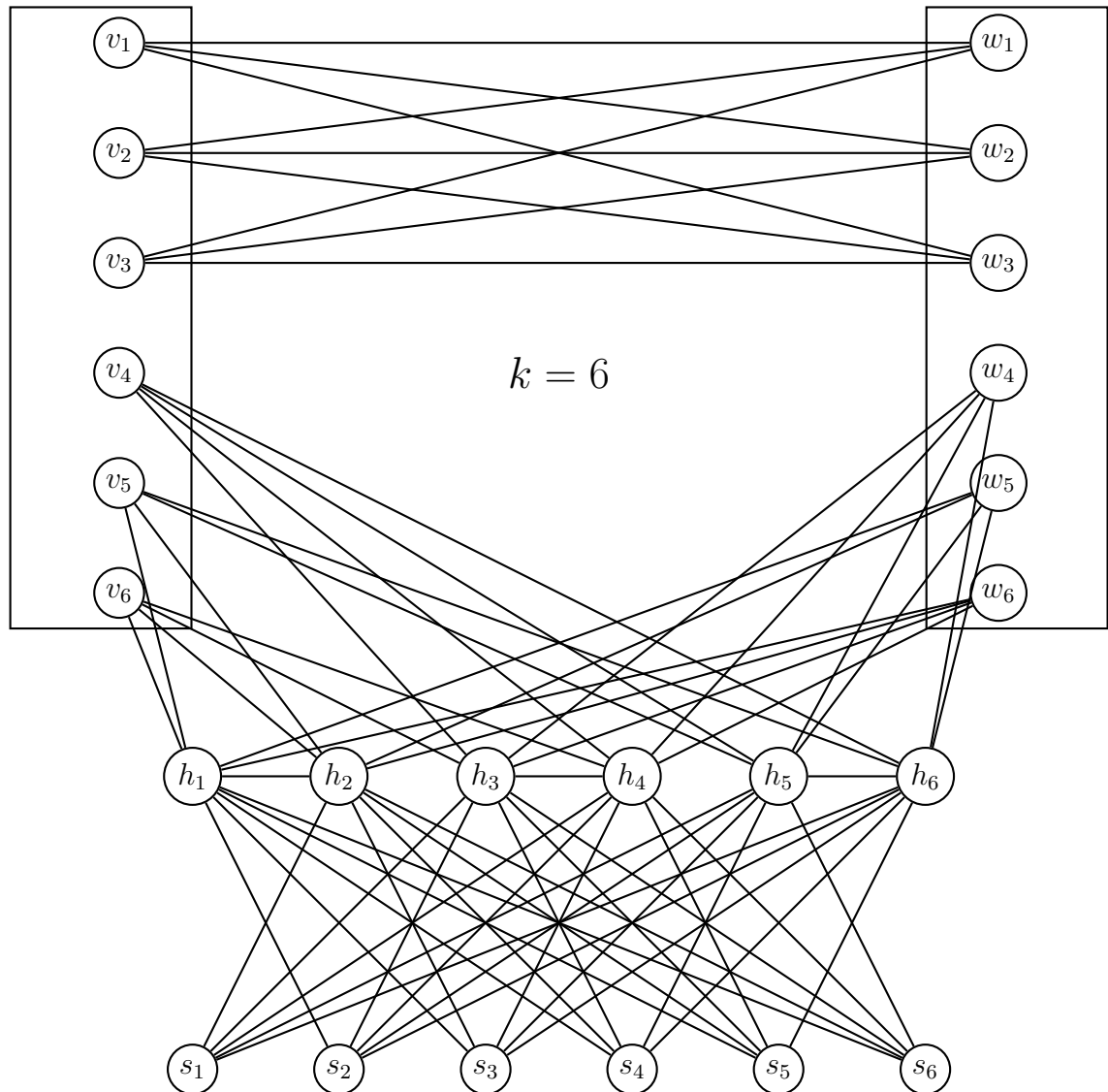
Schritt 2: G_0^k, G_1^k werden um jeweils k Knoten und geeignete Kanten erweitert. Diese Erweiterung ist völlig analog zum Fall $k = 4$. Die so konstruierten Graphen kriegen die Namen F_k, E_k . Mit einer völlig analogen Definition von Schnittstelle, farbenreich und einfarbig ergeben sich folgende Eigenschaften von F_k, E_k :

0. $F_k \in k\mathbf{C} \setminus (k-1)\mathbf{C}$, $E_k \in k\mathbf{C}$.
1. $S(F)$ ist farbenreich.
2. $S(E)$ ist einfarbig.

3. Duplikator gewinnt das $(c - k, \bar{r})$ -monNP Spiel auf F_k, E_k .

Schritt 3: Ebenso wie im Fall $k=4$ werden drei Adapter A, B, C mit analogen Eigenschaften definiert. Die Erweiterung der Adapter A und C ist leicht zu sehen. Die Erweiterung von Adapter B ein wenig schwieriger. Man muß in dem Fall zwischen geradem und ungeradem k unterscheiden. Da eine formale Definition des Adapters B für allgemeines k sehr umständlich ist, zeigt die folgende Abbildung die Struktur für $k=5$ und $k=6$. Darauf aufbauend ist die Erweiterung für allgemeines k leicht zu sehen.





Schritt 4: Es werden schließlich völlig analog zum Fall $k = 4$ Strukturen S_0^r, S_1^r mit folgenden Eigenschaften konstruiert:

1. $S_0^r \in \text{non}k\mathbf{C}, S_1^r \in k\mathbf{C}$.
2. Duplikator gewinnt das $(\exists\forall)^r(c - k, \bar{r})$ -PFO(monNP) Spiel auf S_0^r, S_1^r .

6.3 Der Fall $k = 3$

Wie sieht es mit der hier verwendeten Methode im Fall $k = 3$ aus? Um analog zum Fall $k = 4$ argumentieren zu können, müßte *nicht-2-Färbbarkeit* nicht in monNP ausdrückbar sein. Die folgenden beiden Sätze zeigen aber, daß dies nicht so ist:

Satz 6.14

Ein Graph G ist genau dann nicht-2-färbbar, wenn er einen Kreis mit ungerader Knotenzahl enthält.

Beweis: Leicht zu zeigen.

Satz 6.15

Nicht-2-Färbbarkeit \in monNP.

Beweis:

Es wird gezeigt, daß die Eigenschaft

$$UK := G \text{ enthält einen ungeraden Kreis}$$

in monNP ausgedrückt werden kann.

Jeder Graph mit einem ungeraden Kreis enthält auch einen minimalen ungeraden Kreis C . Die Knoten von C haben bis auf die Kreisverbindungskanten keine weiteren Kanten innerhalb von C , denn sonst wäre C nicht minimal. Die Existenz eines solchen *reinen* ungeraden Kreises wird durch folgende monNP-Formel beschrieben:

$$\varphi_{n2F} := \exists UK \text{ (ungerader Kreis)} \exists B \text{ (blau)} \exists R \text{ (rot) mit}$$

+ jeder Punkt aus UK hat genau zwei Nachbarn in UK .

+ jeder Punkt aus UK ist entweder blau oder rot.

+ es gibt genau zwei benachbarte Punkte in UK mit derselben Farbe. □

Auch wenn der Fall $k = 3$ nicht mit derselben Methode wie der Fall $k \geq 4$ erledigt werden kann, gilt das Ergebnis auch für $k = 3$. Dies läßt sich durch eine Reduktion auf den Fall $k = 4$ erledigen. Die entscheidende Idee zu dieser Reduktion verdanke ich Herrn Konrad Zdanowski.

Notwendig ist eine FO-Reduktion erster Ordnung des Problems *nicht-4-Färbbarkeit* auf das Problem *nicht-3-Färbbarkeit*. Desweiteren braucht man eine zu Satz 6.3 analoge Abschlußeigenschaft von PFO(monNP) bzgl. solcher FO-Reduktionen 1. Ordnung. Diese läßt sich analog zu Satz 6.3 beweisen.

6.4 Zusammenfassung und Bemerkungen zu Kapitel 5 und 6

Im zweiten Teil dieser Arbeit wurden *monadische Abschlüsse* von monadic NP untersucht. Hauptsächlich ging es um den positiven FO-Abschluß von monadic NP (kurz:

PFO(monNP)). Im weiteren Kontext wurde der Abschluß von monadic NP bzgl. FO-Quantifizierung und existentieller monadischer SO-Quantifizierung (closed monadic NP, kurz: cmonNP) betrachtet.

Die Tatsache, daß Graphzusammenhang in $\forall\forall\text{monNP} \subset \text{PFO}(\text{monNP})$ ausdrückbar ist, zeigt, daß PFO(monNP) eine *interessante* und *vernünftige* Erweiterung von monadic NP ist. Man kann sogar zeigen, daß es für jedes $k \in \mathbb{N}$ in PFO(monNP) beschreibbare Eigenschaften gibt, die nicht in k -stelliger existentieller SO-Logik ausgedrückt werden können (siehe [1]). PFO(monNP) ist also durchaus ausdrucksstärker als monadic NP.

Ajtai, Fagin und Stockmeyer waren die ersten, denen es gelungen ist, eine in PFO(monNP) nicht ausdrückbare monadische Eigenschaft \mathcal{P}_g zu finden. Dabei ist eine Eigenschaft monadisch, falls sie in MSO ausdrückbar ist. In der Folgezeit konnte Marcinkowski zeigen, daß die Eigenschaft $\mathcal{E} = \text{Erreichbarkeit in gerichteten Graphen}$ nicht in PFO(monNP) ausdrückbar ist.

In Kapitel 5 wurde gezeigt, daß beide Beweise auf derselben Grundidee beruhen. Grobgesprochen werden die vorangestellten FO-Runden durch Spielen auf isomorphen Teilstrukturen *bewältigt*.

Desweiteren wurde der Beweis von Fagin et al. vereinfacht und verallgemeinert.

Im Hinblick auf closed monadic NP sind folgende Fragen offen:

- Ist die Hierarchie innerhalb von closed monadic NP strikt (siehe Abschnitt 2.2)?
- Gibt es monadische Eigenschaften, die nicht in closed monadic NP ausdrückbar sind. Gilt also $\text{cmonNP} \neq \text{MSO}$?

Sowohl \mathcal{P}_g , als auch \mathcal{E} sind in der zweiten Stufe von cmonNP beschreibbar. Beide Eigenschaften sind also nicht geeignet, um Antworten auf obige Fragen zu finden.

In Kapitel 6 wurde gezeigt, daß

$$\text{nicht-}k\text{-Färbbarkeit (kurz: non}k\text{C)} \notin \text{PFO}(\text{monNP}) \text{ für } k \geq 3 \quad (*).$$

Dies ist das Hauptergebnis des zweiten Teils dieser Arbeit. Unter der Annahme, daß $\text{NP} \neq \text{co-NP}$, gilt

$$\text{non}k\text{C} \notin \text{closed monadic NP.} \quad (**)$$

Da non k C leicht in co-monadic NP beschreibbar ist, würde (**) die zweite Frage positiv beantworten. Das Ergebnis (*) ist ein erster Schritt in diese Richtung.

Es ist jedoch nicht zu erwarten, daß eine *leichte* Verallgemeinerung der hier verwendeten Methoden zum Beweis von (**) führen kann. Wesentlicher Bestandteil im Beweis von (*) war eine Reduktion der Eigenschaft *Graphzusammenhang* auf das Problem non k C. Duplikators Spielstrategie im PFO(monNP)-Spiel beruht entscheidend auf seiner Spielstrategie im monadic NP-Spiel bzgl. der Eigenschaft *Graphzusammenhang*. Diese ist aber schon in $\forall\forall\text{monNP}$ ausdrückbar.

Aus meiner Sicht könnte ein direkter (nicht auf Reduktion beruhender) Beweis von

$\text{non}kC \notin \text{monadic NP}$

zu großen Fortschritten in Bezug auf (**) führen.

Was die erste der obigen Fragen angeht, ist die Situation ähnlich *deprimierend* wie im Kontext von *Graphzusammenhang* und monadic NP + starke built-in Relationen. Es ist bisher nicht gelungen, eine Eigenschaft *nicht-2te-Stufe* zu finden mit

$$\text{nicht-2te-Stufe} \notin 2\text{te Stufe von cmonNP} \wedge \text{nicht-2te-Stufe} \in \text{cmonNP} \quad (***)$$

Auch das Anwenden von *plausiblen* komplexitätstheoretischen Annahmen liefert keine solche Eigenschaft.

Es könnte also sein, daß die Hierarchie auf der zweiten Stufe zusammenfällt. Es war mir lange Zeit unklar, warum es (scheinbar) so schwer ist, Nichtausdrückbarkeitsbeweise für monadische Eigenschaften bzgl. der zweiten Stufe von cmonNP zu führen. Für die *scheinbar* schwierigere monadische Hierarchie konnte Matz und Thomas (siehe [20]) die Striktheit beweisen. Dies wurde im wesentlichen durch Zählen von Spieltypen erreicht. Ein analoges Vorgehen im Kontext von closed monadic NP ist nicht möglich, da die Anzahl von Spieltypen stark von der Anzahl der Alternationen von All- und Existenzquantoren (unabhängig ob FO- oder SO-Quantoren) abhängt. Es ist auch *leicht* zu zeigen, daß alle von Matz und Thomas verwendeten Eigenschaften in der zweiten Stufe von closed monadic NP ausdrückbar sind.

Der folgende (unveröffentlichte) Satz von Schweikardt und Schwentick liefert eine mögliche Begründung für die Schwierigkeit von (***):

Satz 6.16

Es gibt zwei Ordnungsrelationen $<_1, <_2$, so daß folgende Implikation gilt:

$$\text{monNP}(+) \subseteq \text{monNP}(<_1, <_2).$$

Bemerkung 6.17

Unter bestimmten Bedingungen an die beiden Ordnungsrelationen gilt auch die umgekehrte Inklusion.

Für die zweite Stufe von closed monadic NP (kurz: 2cmonNP) folgt das folgende Korollar.

Korollar 6.18

Es gibt zwei Nachfolgerrelationen S_1, S_2 , so daß folgende Implikation gilt:

$$\text{monNP}(+) \subseteq 2\text{cmonNP}(S_1, S_2).$$

Erweitert man also 2cmonNP um zwei *einfache* Nachfolgerrelationen, so erhält man die volle Ausdrucksstärke von monNP(+). Einerseits liefert dieses Korollar ein *intuitives* Verständnis dafür, daß es schwierig ist, Eigenschaften mit (***) zu finden.

Andererseits könnte es Ausdrucksschwächeergebnisse im Kontext mit $\text{monNP}(+)$ liefern. Dazu braucht man eine Eigenschaft \mathcal{E} mit $\mathcal{E} \notin 2\text{cmonNP}(S_1, S_2)$, wobei S_1, S_2 beliebige Nachfolgerrelationen sind.

Überraschenderweise ist es *leicht*, nicht monadische Eigenschaften zu finden. Insbesondere sind diese Eigenschaften nicht in cmonNP ausdrückbar. Ein Beispiel dafür ist die Eigenschaft *Hamiltonkreis* (siehe Turan [27]). Der *einfache* Beweis kann ohne Schwierigkeiten zum Beweis des folgenden Satzes erweitert werden.

Satz 6.19

Hamiltonkreis $\notin \text{MSO}(S_1)$, wobei S_1 eine beliebige Nachfolgerrelation ist.

Allerdings *scheinen* die Methoden nicht mehr zu funktionieren, sobald man eine zweite Nachfolgerrelation zulässt.

Literaturverzeichnis

- [1] M. Ajtai. Σ_1^1 -formulae on finite structures. *Ann. Pure Appl. Logic*, 24(1):1–48, 1983.
- [2] Miklós Ajtai and Ronald Fagin. Reachability is harder for directed than for undirected finite graphs. *J. Symbolic Logic*, 55(1):113–150, 1990.
- [3] Miklos Ajtai, Ronald Fagin, and Larry Stockmeyer. The closure of monadic NP. *J. Comput. System Sci.*, 60(3):660–716, 2000. 30th Annual ACM Symposium on Theory of Computing (Dallas, TX, 1998).
- [4] D. Coppersmith. A left coset composed of n-cycles. Research report rc 19511. IBM, 1994.
- [5] Stavros S. Cosmadakis. Logical reducibility and monadic NP. In *34th Annual Symposium on Foundations of Computer Science (Palo Alto, CA, 1993)*, pages 52–61. IEEE Comput. Soc. Press, Los Alamitos, CA, 1993.
- [6] Reinhard Diestel. *Graphentheorie*. Springer-Lehrbuch. [Springer Textbook]. Springer-Verlag, Berlin, 1996.
- [7] Arnaud Durand, Clemens Lautemann, and Thomas Schwentick. Subclasses of binary NP. *J. Logic Comput.*, 8(2):189–207, 1998.
- [8] H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical logic*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, second edition, 1994. Translated from the German by Margit Meßmer.
- [9] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, second edition, 1999.
- [10] Ronald Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In *Complexity of computation (Proc. SIAM-AMS Sympos. Appl. Math., New York, 1973)*, pages 43–73. SIAM-AMS Proc., Vol. VII. Amer. Math. Soc., Providence, R.I., 1974.
- [11] Ronald Fagin. Monadic generalized spectra. *Z. Math. Logik Grundlagen Math.*, 21:89–96, 1975.

- [12] Ronald Fagin. Comparing the power of games on graphs. *Math. Logic Quart.*, 43(4):431–455, 1997.
- [13] Ronald Fagin, Larry Stockmeyer, and Moshe Y. Vardi. On monadic NP vs. monadic co-NP (extended abstract). In *Proceedings of the Eighth Annual Structure in Complexity Theory Conference (San Diego, CA, 1993)*, pages 19–30, Los Alamitos, CA, 1993. IEEE Comput. Soc. Press.
- [14] Markus Frick and Martin Grohe. Deciding first-order properties of locally tree-decomposable graphs. In *Automata, languages and programming (Prague, 1999)*, volume 1644 of *Lecture Notes in Comput. Sci.*, pages 331–340. Springer, Berlin, 1999.
- [15] Neil Immerman. *Descriptive complexity*. Graduate Texts in Computer Science. Springer-Verlag, New York, 1999.
- [16] David Janin and Jerzy Marcinkowski. A toolkit for first order extensions of monadic games. In *STACS 2001 (Dresden)*, volume 2010 of *Lecture Notes in Comput. Sci.*, pages 353–364. Springer, Berlin, 2001.
- [17] Martin Kreidler and Detlef Seese. Monadic NP and graph minors. In *Computer science logic (Brno, 1998)*, volume 1584 of *Lecture Notes in Comput. Sci.*, pages 126–141. Springer, Berlin, 1999.
- [18] Jerzy Marcinkowski. Directed reachability: from Ajtai-Fagin to Ehrenfeucht-Fraïssé games. In *Computer science logic (Madrid, 1999)*, volume 1683 of *Lecture Notes in Comput. Sci.*, pages 338–349. Springer, Berlin, 1999.
- [19] Oliver Matz, Nicole Schweikardt, and Wolfgang Thomas. The monadic quantifier alternation hierarchy over grids and graphs. *Inform. and Comput.*, 179(2):356–383, 2002. LICS’97 (Warsaw).
- [20] Oliver Matz and Wolfgang Thomas. The monadic quantifier alternation hierarchy over graphs is infinite. In *Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 236–244. Warsaw, Poland, 1997.
- [21] Thomas Schwentick. *On winning Ehrenfeucht games and monadic NP*. PhD thesis, Johannes Gutenberg-Universität Mainz, 1994.
- [22] Thomas Schwentick. Graph connectivity, monadic NP and built-in relations of moderate degree. In *Automata, languages and programming (Szeged, 1995)*, volume 944 of *Lecture Notes in Comput. Sci.*, pages 405–416. Springer, Berlin, 1995.
- [23] Thomas Schwentick. On winning Ehrenfeucht games and monadic NP. *Ann. Pure Appl. Logic*, 79(1):61–92, 1996.

- [24] Thomas Schwentick. Padding and the expressive power of existential second-order logics. In *Computer science logic (Aarhus, 1997)*, volume 1414 of *Lecture Notes in Comput. Sci.*, pages 461–477. Springer, Berlin, 1998.
- [25] Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoret. Comput. Sci.*, 3(1):1–22 (1977), 1976.
- [26] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of formal languages, Vol. 3*, pages 389–455. Springer, Berlin, 1997.
- [27] György Turán. On the definability of properties of finite graphs. *Discrete Math.*, 49(3):291–302, 1984.

