OBSTRUCTIONS TO TRIGRAPH HOMOMORPHISMS

by

Wing Xie
Hon. B.Sc., University of Toronto, 2003

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Fall 2006

APPROVAL

Name: Wing Xie

Degree: Master of Science

Title of thesis: Obstructions to Trigraph Homomorphisms

Examining Committee: Dr. Joseph G. Peters
Chair

____

Dr. Pavol Hell
Professor, Computing Science
Senior Supervisor

____

Dr. Ramesh Krishnamurti
Professor, Computing Science
Supervisor

____

Dr. Arthur L. Liestman
Professor, Computing Science
Supervisor

____

Dr. Ladislav Stacho
Assistant Professor, Mathematics
Simon Fraser University
SFU Examiner

Date Approved: November 30, 2006

Abstract

Many graph partition problems seek a partition into parts with certain internal constraints on each part, and similar external constraints between the parts. Such problems have been traditionally modeled using matrices, as the so-called M-partition problems. More recently, they have also been modeled as trigraph homomorphism problems. This thesis consists of two parts. In the first part, we survey the literature dealing with both general and restricted versions of these problems. Most existing results attempt to classify these problems as NP-complete or polynomial time solvable. In the second part of the thesis, we investigate which of these problems can be characterized by a finite set of forbidden induced subgraphs. We develop new tools and use them to find all such partition problems with up to five parts. We also observe that these problems are automatically polynomial time solvable.

Keywords: matrix partitions; forbidden subgraphs; minimal obstructions; generalized colourings; trigraph homomorphisms

Dedication

To my parents, for their support and encouragement.

向我的父母致谢，感谢你们对我的支持与鼓励。

Acknowledgments

The completion of this thesis would not be possible without the direction and support of Dr. Pavol Hell. He has been a dedicated and encouraging mentor and I am indebted to him for all that he has taught me.

I would like to thank my committee members Dr. Ramesh Krishnamurti, Dr. Arthur Liestman, Dr. Joseph Peters and Dr. Ladislav Stacho for their editorial comments on this thesis. I would also like to thank Dr. Derek Corneil for introducing to me the world of graph theory.

Finally, I would like to thank my family and friends for giving me the courage and confidence to pursue my dreams, and encouragement to see them to the end.

Contents

Approval ii

Abstract iii

Dedication iv

Acknowledgments v

Contents vi

List of Figures ix

List of Tables xi

1 Introduction 1

1.1 Homomorphism 2

1.2 M-partition 4

1.3 Obstructions 5

1.4 Summary of results 7

2 A survey of existing results 9

2.1 Terminology 9

2.1.1 Variations of trigraph homomorphisms 10

2.1.2 Graph families 12

2.1.3 Special classes of trigraphs 13

2.2 Problems trigraph homomorphisms can model 14

2.2.1 m-colouring 15

2.2.2 H-colouring . . . 16
2.2.3 Split graphs . . . 17
2.2.4 (a, b)-graphs . . . 18
2.2.5 Cutsets . . . 19
2.2.6 The Winkler problem . . . 22
2.3 Tools . . . 23
2.3.1 Retraction and domination . . . 23
2.3.2 2-SAT . . . 25
2.3.3 Odd girth and chromatic number . . . 26
2.3.4 Labeled and unlabeled graphs . . . 27
2.4 General results . . . 27
2.4.1 Small matrices . . . 28
2.4.2 Full homomorphisms . . . 29
2.4.3 Directed graphs . . . 29
2.5 Homomorphisms of graph families . . . 31
2.5.1 Perfect graphs . . . 31
2.5.2 Cographs . . . 32
2.5.3 Chordal graphs . . . 32
2.6 Table of known results . . . 34

3 Infinitely many minimal obstructions **36**
3.1 Messy trigraphs have IMMO . . . 37

4 Finitely many minimal obstructions **41**
4.1 Split-friendly trigraphs . . . 41
4.2 Labeled trigraphs . . . 44
4.2.1 Similarity . . . 44
4.2.2 Nice trigraphs . . . 47

5 Trigraphs with up to five vertices **51**
5.1 Trigraphs with one vertex . . . 52
5.2 Trigraphs with two vertices . . . 52
5.3 Trigraphs with three vertices . . . 57
5.4 Trigraphs with four vertices . . . 58

5.5 Trigraphs with five vertices 60

6 An exceptional trigraph with IMMO **66**

6.1 Clam graphs 67

6.2 Every clam graph is an obstruction to H 67

6.3 Every clam graph is a minimal obstruction to H 70

7 Conclusion **72**

7.1 Future work 73

Bibliography **74**

List of Figures

1.1 A trigraph and its adjacency matrix 1
1.2 A graph and the result of applying convention 1 and 2 3
1.3 A trigraph and a M-partition 5
1.4 Trigraphs corresponding to a 2-colouring and 3-colouring 7
1.5 A trigraph B and its complement $\overline{B}$ 8

2.1 Two constant trigraphs 14
2.2 Two normal trigraphs 15
2.3 Trigraphs used to characterize 2-colourability and 3-colourability 16
2.4 Forbidden induced subgraphs for split graphs 17
2.5 A trigraph used to characterize split graphs 18
2.6 Trigraphs used to characterize (1,2)-graphs and (2,2)-graphs 19
2.7 Trigraphs used to model cutset problems 21
2.8 Two trigraphs used to model new cutset problems 22
2.9 A trigraph used to model the Winkler problem 23
2.10 Three homomorphically equivalent trigraphs 24
2.11 Trigraph that models the 'stubborn problem' 28
2.12 NP-complete directed trigraphs 30

3.1 A trigraph B and its complement $\overline{B}$ 37

4.1 Split-friendly trigraphs 43
4.2 A trigraph used to characterize split graphs 44
4.3 A trigraph H and a graph G 46
4.4 Two nice trigraphs 48

5.1 Trigraphs with exactly one vertex. 52
5.2 Trigraphs with two 0-vertices 53
5.3 Obstructions to H_5 54
5.4 Trigraphs with two 1-vertices 55
5.5 Trigraphs with one 0-vertex and one 1-vertex 55
5.6 Obstructions to H_{10} 57
5.7 The general structure of an exceptional trigraph 61
5.8 Similarity classes for the general structure of an exceptional trigraph 61

6.1 An exceptional trigraph and its corresponding partition 66
6.2 A clam graph 67

List of Tables

2.1 Table of known results 35

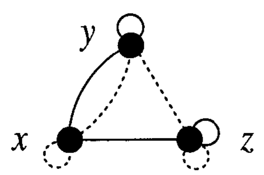
Chapter 1

Introduction

A *trigraph* H consists of a set of vertices $V(H)$ and two symmetric binary relations $E(H)$ and $N(H)$ such that $E(H) \cup N(H) = V(H) \times V(H)$. Let $m = |V(H)|$. The adjacency matrix of a trigraph $adj(H)$ is defined to be a symmetric $m \times m$ matrix M_H (with each vertex $v \in V(H)$ corresponding to one row and one column of M_H over $\{0, 1, *\}$ such that M_H has the entries:

- $M_H(u, v) = 0$ if $uv \in N(H) - E(H)$
- $M_H(u, v) = 1$ if $uv \in E(H) - N(H)$
- $M_H(u, v) = *$ if $uv \in N(H) \cap E(H)$

Figure 1.1 illustrates a trigraph and its corresponding adjacency matrix. In this figure (and all future figures of trigraphs), we use solid lines to indicate pairs in the relation $E(H)$ and dotted lines to indicate pairs in the relation $N(H)$.



	x	y	z
x	0	*	1
y	*	1	0
z	1	0	*

Figure 1.1: A trigraph and its adjacency matrix

We will refer to the pair uv as a 0-edge, 1-edge or $*$-edge when $M_H(u,v) = 0$, $M_H(u,v) = 1$ and $M_H(u,v) = *$ respectively. Likewise a vertex v is referred to as a 0-vertex, 1-vertex or $*$-vertex when $M_H(v,v) = 0$, $M_H(v,v) = 1$ and $M_H(v,v) = *$ respectively. Note that v is a 0-vertex if and only if the loop vv is a 0-edge, v is a 1-vertex if and only if the loop vv is a 1-edge, and v is a $*$-vertex if and only if the loop vv is a $*$-edge.

A *graph* G (in the usual meaning of graphs without loops and multiple edges) consists of a set of vertices $V(G)$ and one binary relation $E(G)$ on $V(G)$. A pair uv in the relation $E(G)$ is commonly referred to as an *edge* in G. Note that there are two ways (conventions) in which a graph G can be viewed as a trigraph H.

Convention 1: G can be viewed as a trigraph H with

- $V(H) = V(G)$
- $E(H) = E(G)$
- $N(H) = E(\overline{G})$

Convention 2: G can be viewed as a trigraph H with

- $V(H) = V(G)$
- $E(H) = E(G)$
- $N(H) = V(H) \times V(H)$

Let u, v be two vertices in $V(G)$. The first convention adds a pair uv in $N(H)$ when $uv \notin E(G)$, and the second convention adds every possible pair uv to $N(H)$. Thus, graphs are viewed as a subset of trigraphs. Both conventions are useful depending on the situation in which the graph is being used. The precise details of when each convention is used will be addressed later in this chapter. To illustrate these two conventions consider graph G and trigraphs H_1, H_2 in Figure 1.2. Applying convention 1 to graph G results in the trigraph H_1 and applying convention 2 to graph G results in the trigraph H_2.

1.1 Homomorphism

A *homomorphism* of a trigraph G to a trigraph H is a mapping : $V(G) \to V(H)$ such that the following properties hold:

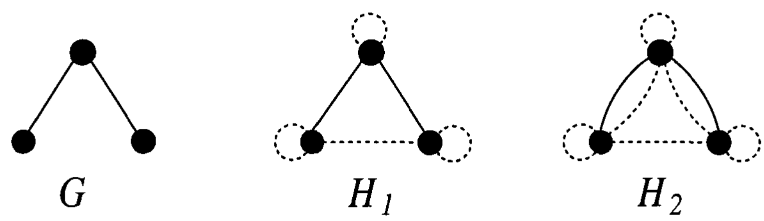


Figure 1.2: A graph and the result of applying convention 1 and 2

- $uv \in E(G)$ implies $f(u)f(v) \in E(H)$
- $uv \in N(G)$ implies $f(u)f(v) \in N(H)$

A trigraph G that has a homomorphism to trigraph H is said to be *H-colourable* or admit an *H-colouring*. The *Trigraph Homomorphism Problem* is defined for a fixed trigraph H as follows.

Instance: A trigraph G.
Problem: Is there a homomorphism of G to H? (i.e., is G H-colourable?)

Homomorphism applies to graphs as well. A *homomorphism* of a graph G to a graph H is a mapping : $V(G) \to V(H)$ such that $uv \in E(G)$ implies $f(u)f(v) \in E(H)$. A homomorphism of a graph G to a graph H can be restated as a trigraph homomorphism if we apply the conventions stated earlier. Let G' be the resulting trigraph from applying convention 1 to graph G and H' be the resulting trigraph from applying convention 2 to graph H. It follows from the construction of G' and H' that there is a (graph) homomorphism of G to H if and only if there is a (trigraph) homomorphism of G' to H'. Note that when converting a graph to a trigraph, convention 1 is always used for the input graph G, and convention 2 is always used for the fixed graph H.

In this thesis we focus on situations in which G is a graph and H a trigraph. A homomorphism of a graph G to a trigraph H is a mapping $f : V(G) \to V(H)$ such that the following holds.

- $uv \in E(G)$ implies $f(u)f(v) \in E(H)$

- $uv \notin E(G)$ and $u \neq v$ implies $f(u)f(v) \in N(H)$

The *Basic Trigraph Homomorphism Problem* is defined for a fixed trigraph H as follows.

Instance: A graph G.
Problem: Is there a homomorphism f of G to H?

A homomorphism f can also be viewed as a partition of the set of vertices $V(G)$ in G, into sets $f^{-1}(x)$, $x \in V(H)$. In other words, for each $x \in V(H)$, consider $f^{-1}(x) = \{v \in V(G) : f(v) = x\}$ to be the sets that partition $V(G)$.

Other types of homomorphisms that we will review in Chapter 2 are surjective homomorphisms and list homomorphisms. Let G be a graph and H a trigraph. A *surjective homomorphism* of G to H is a homomorphism f of G to H such that for all $x \in V(H)$, the set $f^{-1}(x)$ is non-empty. Now let G be a graph with lists $L_v \subseteq V(H)$ for all $v \in V(G)$. A *list homomorphism* of G to H is a homomorphism f of G to H such that for all $v \in G$, we have $f(v) \in L_v$.

1.2 M-partition

Let G be any (input) graph and M be a fixed symmetric $m \times m$ matrix over $\{0, 1, *\}$. The *M-partition problem* first introduced in [18] is defined for a fixed $m \times m$ matrix M over $\{0, 1, *\}$ as follows:

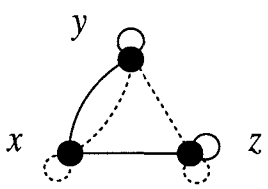
Instance: A trigraph G.
Problem: Is there a partition of the vertices $V(G)$ into m parts $P_1, \ldots, P_m$ such that if $M(i, j) = 0$ every vertex placed in P_i is non-adjacent with every vertex placed in P_j, and if $M(i, j) = 1$ then every vertex placed in P_i is adjacent to every vertex placed in P_j. Note that if $M(i, j) = *$ then there are no restrictions on edges between P_i and P_j.

Note that when $M(i, i) = 0$, the vertices in P_i induce an independent set in G, and when $M(i, i) = 1$ the vertices in P_i induce a clique in G.

Let H be a trigraph and M_H its adjacency matrix. The M_H-partition problem corresponds to the trigraph homomorphism problem on trigraph H. Since H and M_H define an identical structure, the trigraph homomorphism problem and the matrix partitions problem are identical problems. It is easy to check that the following terminology is used to describe the same fact:

- There is a homomorphism of G to H
- G admits an H-colouring
- G is H-colourable
- G is M_H-partitionable
- G admits an M_H-partition

The M-partition problem was originally defined graphically by 3 kinds of circles and 3 kinds of edges [18]. An empty circle represents a 0-vertex $x \in V(H)$, a crosshatched circle represents a 1-vertex $y \in V(H)$ and a hatched circle represents a $*$-vertex $z \in V(H)$. Similarly a regular (non-bold) edge xy represents a $*$-edge, a bold edge represents a 1-edge and a non-edge represents a 0-edge. Pictorially, Figure 1.3 shows a graphical representation of a M-partition as it was originally defined, and a trigraph that represents the identical structure.



	x	y	z
x	0	*	1
y	*	1	0
z	1	0	*

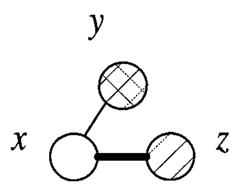


Figure 1.3: A trigraph and a M-partition

1.3 Obstructions

An *obstruction* to a trigraph H is a graph G which does not admit an H-colouring. A *minimal obstruction* to a trigraph H is an obstruction G to H such that for any vertex $v \in V(G)$, the subgraph $G - v$ admits an H-colouring. Thus, any graph G that does not admit an H-colouring must have an induced subgraph that is a minimal obstruction.

Clearly there is no homomorphism of G to H if and only if G contains an induced subgraph isomorphic to a minimal obstruction to H. Minimal obstructions to trigraph H can be viewed as *forbidden subgraphs* that characterize which graphs G admit a homomorphism to H.

Let H be a trigraph. The *complement* $\overline{H}$ of H has $V(\overline{H}) = V(H)$, $E(\overline{H}) = N(H)$ and $N(\overline{H}) = E(H)$. Thus, the adjacency matrix $adj(\overline{H})$ of $\overline{H}$ is obtained from the adjacency matrix $adj(H)$ of H by exchanging 0's for 1's and 1's for 0's. Since the relations E and N are complementary in H and $\overline{H}$, we have the following result.

Proposition 1.3.1 *G admits an H-colouring if and only if $\overline{G}$ admits an $\overline{H}$-colouring.* □

It follows that G is an obstruction to H if and only if $\overline{G}$ is an obstruction to $\overline{H}$.

The focus of this thesis is to find whether a fixed trigraph H has an infinite or a finite set of minimal obstructions σ. That is, whether there is a finite and complete set of forbidden induced subgraphs that characterize whether graph G admits an H-colouring. Note that having a finite set of minimal obstructions implies that there is a bound on the size of a minimal obstruction while an infinite set of minimal obstructions implies that the size of minimal obstructions is unbounded. Let σ be a set of minimal obstructions to trigraph H. We denote by $\overline{\sigma}$ the minimal obstructions to $\overline{H}$ consisting of obstructions $\overline{O}$ for all $O \in \sigma$.

Theorem 1.3.2 *If H is a trigraph which has finitely many minimal obstructions, then the trigraph homomorphism problem to H can be solved in polynomial time.*

Proof. For a trigraph H that has a finite set of minimal obstructions we can apply the algorithm below to check whether a graph G admits an H-colouring.

Algorithm: Generic Polynomial Time Recognition Algorithm

Input: A graph G and a finite set $S = \{O_1, O_2, \ldots, O_p\}$ of minimal obstructions to trigraph H and $t = max\{|V(O)| : O \in S\}$.

Action: Check whether each subgraph G' of G consisting of $1 \leq r \leq t$ vertices is isomorphic to O_i in time $O(n^r)$. If it is, then G does not admit a H-colouring since G' is an obstruction to H and does not admit an H-colouring. Since we have a finite set of minimal obstructions for H, we can simply check every obstruction in time $O(n^t)$. □

The trigraph homomorphism problem for a fixed trigraph H having a minimal obstruction with at most t vertices can be solved in time $O(n^t)$. We note however that more efficient algorithms are sometimes possible [37, 40].

Although a finite set of minimal obstructions for H implies there is a polynomial time algorithm for the trigraph homomorphism problem on H, having an infinite set of minimal obstructions to H neither implies the trigraph homomorphism problem on H is polynomial time solvable nor that it is NP-complete. For example, both the 2-colouring and 3-colouring problem modeled by trigraphs H_1 and H_2 in Figure 1.4 respectively have an infinite set of minimal obstructions. For the 2-colouring problem it is the set $\sigma_1 = \{C_{2n+1} : n > 0\}$, and for the 3-colouring problem σ_2 contains the set $\{C_{2n+1} + v : n > 0\}$. The 2-colouring problem is known to be polynomial time solvable, while the 3-colouring problem is known to be NP-complete [27].

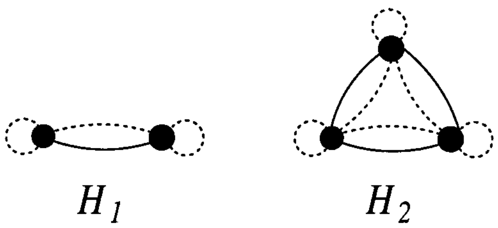


Figure 1.4: Trigraphs corresponding to a 2-colouring and 3-colouring

1.4 Summary of results

In this thesis we look at undirected graphs and trigraphs, that is, graphs and trigraphs with symmetric relations $N(H), E(H)$ and symmetric adjacency matrices. Relaxing these restrictions leads to the study of digraph partitions and directed trigraph homomorphisms [21]. It is obvious that for trigraphs H with a $*$-vertex, an H-colouring of a graph G is always possible. Thus, in our results and discussions from Chapters 3 to 7, we shall assume that trigraphs do not have any $*$-vertices.

We will first review some known results and tools in Chapter 2. In Chapter 3, we will prove that all trigraphs with an induced subtrigraph isomorphic to B or $\overline{B}$ in Figure 1.5

have infinitely many minimal obstructions.

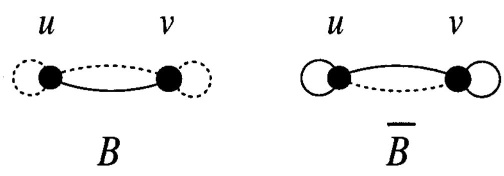


Figure 1.5: A trigraph B and its complement $\overline{B}$

In Chapter 4, we define two trigraph families that have finitely many minimal obstructions. In Chapter 5, we show the precise minimal obstructions for all trigraphs with up to two vertices. For trigraphs with up to five vertices, we show that the only trigraphs with infinitely many minimal obstructions have an induced subtrigraph isomorphic to B or $\overline{B}$, and all other trigraphs have finitely many minimal obstructions. Finally, in Chapter 6, we will give an example of a trigraph with six vertices that does not have an induced subtrigraph isomorphic to B or $\overline{B}$, but still has infinitely many minimal obstructions.

Chapter 2

A survey of existing results

Since its introduction by Feder et. al. in [18], the M-partition problem, later modeled as a trigraph homomorphism problem, has been studied by many [14, 15, 17, 18, 19, 20, 21]. In this chapter, we give an overview to a selection of known results in this area and introduce some terminology and techniques commonly used in solving these types of problems. In addition we will show that trigraph homomorphisms can model a variety of interesting graph theoretic concepts that have been studied in the past. The study of trigraph homomorphisms is motivated in part by the study of these different concepts since trigraph homomorphism model, unifies, and in many cases generalizes, these concepts. A majority of the terminology and techniques that are given in this chapter are adapted from [14, 15, 17, 18, 19, 20, 21], to correspond to the trigraph homomorphism terminology.

2.1 Terminology

We first introduce some terminology that will come up in our literature review and in various parts of the thesis. Four common types of trigraph homomorphism problems will be introduced. In addition, three well-known graph families, and two trigraph families will also be defined. We will later review some general results as well as some special cases pertaining to these different families.

2.1.1 Variations of trigraph homomorphisms

There are four common variations of trigraph homomorphisms that we consider in our review: an (ordinary) homomorphism (formulated in Chapter 1), a surjective homomorphism, a labeled homomorphism and a list homomorphism.

The (ordinary) homomorphism variation is perhaps the most natural way to think of the trigraph homomorphism problems. Let G be a graph and H a trigraph. In this case, one simply seeks a homomorphism f of G to H without additional restrictions. The *Basic Trigraph Homomorphism Problem* is defined for a fixed trigraph H as follows:

Instance: A graph G.
Problem: Is there a homomorphism f of G to H?

As we will see, all other variations of the trigraph homomorphism problem that we consider adds restrictions on where vertices of G may map to.

The *Surjective Homomorphism Problem* is defined for a fixed trigraph H as follows:

Instance: A graph G.
Problem: Is there a homomorphism f of G to H such that $|f^{-1}(x)| > 0$ for all $x \in V(H)$?

Recall that a surjective homomorphism is a homomorphism with the added restriction that f maps to every vertex of H at least one vertex of G. Surjective homomorphisms are used to model many different kinds of partition problems. They include certain cutset problems which will be explained in more detail in a later section.

The *Labeled Homomorphism Problem* is defined for a fixed trigraph H as follows:

Instance: A graph G with vertices $v \in V(G)$ labeled A or B.
Problem: Is there a homomorphism f of G to H such that $f(v)$ is a 0-vertex for each A-labeled vertex v and $f(v)$ is a 1-vertex for each B-labeled vertex v.

Finally, the *List Homomorphism Problem* is defined for a fixed trigraph H as follows:

Instance: A graph G and a list $L_v \subseteq V(H)$ for all $v \in V(G)$.
Problem: Is there a homomorphism f of G to H such that $f(v) \in L_v$ for all $v \in V(G)$?

The list trigraph homomorphism problem is the most general of all the variations mentioned. In particular, a polynomial time algorithm for the list trigraph homomorphism problem implies a polynomial time algorithm for the other three. For example, an instance G of the basic trigraph homomorphism problem for a trigraph H can be described as an instance for the list trigraph homomorphism problem for H, if we set all lists $L_v = V(H)$ for vertices $v \in V(G)$. Thus, we can formulate the following theorem.

Theorem 2.1.1 *[18] A mapping f is a homomorphism of G to H if and only if it is a list homomorphism with respect to the list L_v.* □

Below, we describe a polynomial time algorithm that uses a solution to list homomorphism to solve the surjective trigraph homomorphism problem.

Algorithm: Solving the surjective trigraph homomorphism problem with a list homomorphism algorithm.

Let H be a fixed trigraph with vertices $x_1, \ldots, x_m$.

Input: A graph G.

Action: For all tuples $(v_1, \ldots v_m)$ of vertices of G we proceed as follows: First, assign each v_i to x_i. In other words, we assign lists $L_{v_i} = \{x_i\}$ and lists $L_u = V(H)$ for all vertices $u \neq v_i$ in G. We then apply our list homomorphism algorithm to H and G with respect to our constructed lists L_{v_i} and L_u. If there is a list homomorphism of G to H, then we have a surjective homomorphism of G to H. Otherwise, we continue to the next tuple $(v_1, \ldots, v_m)$. Over all tuples of m vertices of G, if none admits a list homomorphism of G to H, then G does not admit a surjective homomorphism to H.

Theorem 2.1.2 *[18]There exists a surjective homomorphism of G to H if and only if there exists a tuple $(v_1, \ldots v_m)$ of vertices of G and a list homomorphism of G to H with respect to the lists $L_{v_i} = \{x_i\}$ for $i \leq m$, and $L_u = V(H)$ for all $u \neq v_i$ in G.* □

For each tuple of m vertices in G, the assignment of lists to these vertices guarantees that a list homomorphism f of G to H maps at least one vertex v_i of G to each vertex x_i of H. Suppose list homomorphism for trigraph H can be solved in time $O(T)$. By checking all

m-tuples in G, we exhaust every possible surjective homomorphism of G to H. Checking all m-tuples can be done in time $O(n^m)$. For each assignment we use list homomorphism, thus surjective homomorphism can be solved in time $O(n^m T)$. Therefore, a polynomial time algorithm for list homomorphism implies a polynomial time algorithm for surjective homomorphism.

It is not hard to see that the labeled homomorphism problem is also a special case of the list homomorphism problem. An instance G of the labeled homomorphism problem for a trigraph H can be described as an instance of the list homomorphism problem for H, if we replace each A-labeled vertex $v \in V(G)$ and B-labeled vertex $u \in V(G)$ with list $L_u = \{x : x \text{ is a 0-vertex in } H\}$ and list $L_v = \{x : x \text{ is a 1-vertex in } H\}$ respectively. Thus we can formulate the following theorem.

Theorem 2.1.3 *[17] A mapping f is a labeled homomorphism of G to H if and only if it is a list homomorphism with respect to lists L_u and L_v.*

Labeled homomorphisms are especially useful in this thesis as they play a critical role in one of the main proofs in Chapter 4.

2.1.2 Graph families

In addition to the variations of the trigraph homomorphism problems, there are also results to other kinds of trigraph homomorphism problems in which the input graph G is restricted to the family of cographs, chordal graphs, or perfect graphs. In some cases, the complexity of certain trigraph homomorphism problems that are NP-complete in the general case, become polynomial time solvable in the restricted cases. In other restricted cases, the bounds on the size of a minimal obstruction may be lowered, improving the efficiency of the generic algorithm given in the first chapter.

Let G be a graph and S a subset of the vertices in G. We denote by $G[S]$ the subgraph of G induced by the vertices in set S. A graph G is a *cograph* if it has no induced path P_4 with four vertices [9]. Equivalently [9], G is a cograph if G is a single vertex, or the vertices of G (or its complement $\overline{G}$) can be partitioned into two sets X and Y such that the subgraphs $G[X]$ and $G[Y]$ are cographs and every vertex in X is non-adjacent to every vertex of Y. Cographs can be recognized in linear time [9].

Let G be a graph and n the number of vertices in G. Graph G is *chordal* if it has no induced cycle C_t with $t > 3$ [28]. Equivalently [28], G is chordal if there is an enumeration of the n vertices $v_1, \ldots, v_n$ of G, such that for $1 \leq i < j < k \leq n$ the pair $v_j v_k$ is an edge in G whenever the pairs $v_i v_j$ and $v_i v_k$ are edges in G. Again, chordal graphs can be recognized in linear time [28].

There has been much attention given to the study of perfect graphs. Indeed, both cographs and chordal graphs are members of the perfect graph family [9, 28]. Let G be a graph. Recall, the *chromatic number* of G, denoted by $\chi(G)$, is the fewest number of colours needed to colour the vertices of G. Recall again, the *clique number* of G, denoted by $\omega(G)$, is the size of the largest clique in G. Graph G is perfect [28] if the chromatic number is equal to the size of the largest clique for the graph itself and all of its induced subgraphs. The *strong perfect graph theorem*, originally conjectured by Berge in the early 1960's, and finally proven by Chudnovsky, Robertson, Seymour, and Thomas [10] in 2003, states that a graph G is perfect if and only if it has no induced cycle C_{2n+1} with $n > 1$ or its complement $\overline{C}_{2n+1}$ with $n > 1$ [10]. Perfect graphs are of interest since they include many well studied graph families, strictly including cographs and chordal graphs. The problem of finding the chromatic number and clique number for perfect graph is polynomial time solvable [30]. Thus, the chromatic number and clique number for cographs and chordal graphs can be found in polynomial time as well; however for these special cases, more efficient algorithms are possible [28].

2.1.3 Special classes of trigraphs

Let H be a trigraph. We define the sets S_A, S_B, S_C, S_D to be a partition of the pairs $V(H) \times V(H) = S_A \cup S_B \cup S_C \cup S_D$ of H as follows.

- $S_A = \{uv : u, v \text{ are distinct 0-vertices in } V(H)\}$
- $S_B = \{uv : u, v \text{ are distinct 1-vertices in } V(H)\}$
- $S_C = \{uv : u \text{ is a 0-vertex and } v \text{ is a 1-vertex in } V(H)\}$
- $S_D = \{uv : u = v \text{ in } V(H)\}$

Note that sets S_A, S_B, S_C all contain edges, and set S_D contain all loops or what we have referred to as 0-vertices and 1-vertices. Recall that there are no $*$-vertices in H by our main assumption in Chapter 1. A trigraph H is *constant* if the sets S_A, S_B, S_C each contain only 0-edges, 1-edges or $*$-edges. In other words, in a constant trigraph, all pairs of distinct vertices u, v incident to the same kind of vertices have the same kind of edges. Figure 2.1 illustrates two constant trigraphs H_1 and H_2 on four and six vertices respectively. For trigraph H_1, vertices a, b are 1-vertices and vertices c, d are 0-vertices. The set $S_A = \{ab\}$ contains only 1-edges, $S_B = \{cd\}$ contains only 0-edges and $S_C = \{ac, ad, bc, bd\}$ contains only $*$-edges. A constant trigraph H is also defined in terms of its adjacency matrix M_H and was originally introduced in [17].

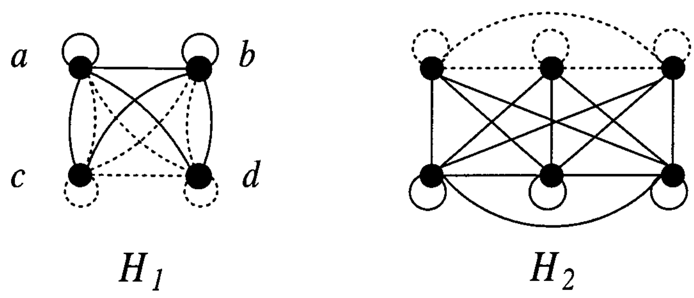


Figure 2.1: Two constant trigraphs

A trigraph H is *normal* if the sets S_A, S_B, S_C each contains only $*$-edges or no $*$-edges (i.e. only 0-edges and 1-edges). In other words, in a normal trigraph, all pairs of distinct vertices u, v incident to the same kind of vertices are either all $*$-edges or no $*$-edges. Figure 2.2 illustrates two normal trigraphs H_1 and H_2 on four and six vertices respectively. For trigraph H_1, vertices a, b are 1-vertices and vertices c, d are 0-vertices. The set $S_A = \{ab\}$ contains no $*$-edges, $S_B = \{cd\}$ contains no $*$-edges and $S_C = \{ac, ad, bc, bd\}$ contains only $*$-edges. Again, a normal trigraph H is also defined in terms of its adjacency matrix M_H and was originally introduced in [19].

2.2 Problems trigraph homomorphisms can model

We now give an overview of some problems trigraph homomorphisms can model which include many popular partition and characterization problems. We define a selection of these problems and we give a detailed explanation of how trigraphs can model them. Some

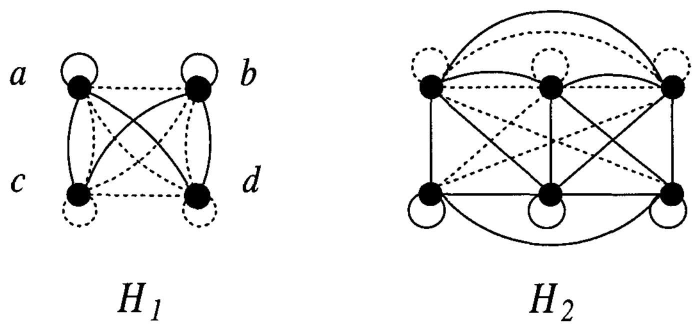


Figure 2.2: Two normal trigraphs

of the problems presented are NP-complete in the general case, but when restricted to certain families of graphs, some problems become polynomial time solvable. These restricted forms of trigraph homomorphisms will be presented in a later section.

2.2.1 m-colouring

An *m-colouring* [28] of a graph G is a partition of the vertices $V(G)$ into sets $V(G) = V_1 \cup \ldots \cup V_m$ such that all vertices in each set V_i are independent in G. The following dichotomy is known for m-colouring.

Theorem 2.2.1 *[27] Finding an m-colouring is polynomial time solvable if and only if $m \leq 2$ and NP-complete otherwise.* □

Finding a colouring or finding the chromatic number of a graph is useful for many graph theoretic problems. Many types of scheduling problems can be modeled as colouring problems. Some common examples are the scheduling of final exams for a university, or the scheduling of departure times of planes for an airport terminal. These and many other examples can be found in most graph theory texts [8, 26, 29, 45].

Let G be a graph and H a trigraph on m 0-vertices with only $*$-edges. It is easy to see that a homomorphism of the trigraph H models m-colouring. All vertices mapped to the same vertex in H must be independent and there are no restrictions among vertices mapped to different vertices in H.

Proposition 2.2.2 *[33] There is an m-colouring of G if and only if G admits an H-colouring.* □

For example, consider the two trigraphs H_1, H_2 illustrated in Figure 2.3. A graph G is 2-colourable if and only if there is a homomorphism of G to H_1, and a graph G is 3-colourable if and only if there is a homomorphism of G to H_2.

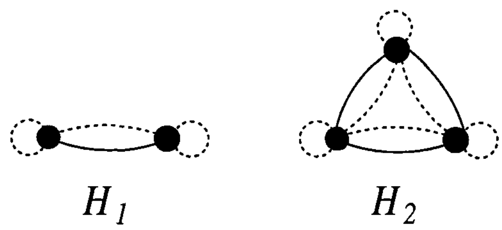


Figure 2.3: Trigraphs used to characterize 2-colourability and 3-colourability

2.2.2 H-colouring

Let H be a graph with m vertices. Recall from Chapter 1 that an H-colouring of a graph G is a homomorphism f of G to H. The study of graph homomorphism began in the early 1960's, and was pioneered by Sabidussi, Hedrlin, and Pultr [41, 35]. In 1990, the complexity of the graph homomorphism problems was completely classified by Hell and Nesetril.

Theorem 2.2.3 *[32] Let H be a fixed graph. If H is bipartite or contains a loop, then the H-colouring problem can be solved in polynomial time; otherwise the problem is NP-complete.* □

Graph homomorphisms are a generalization of graph colouring and have gained much interest over the past four decades. Graph homomorphisms have been used for the recognition of certain graph families. They arise in other areas of computer science and mathematics such as complexity theory, artificial intelligence, telecommunications and statistical physics [33].

Let G and H be graphs and H' be the trigraph that results from applying convention 2 (from Chapter 1) to H.

Proposition 2.2.4 *Graph G admits an H-colouring if and only if it admits an H'-colouring.* □

A detailed survey of graphs and homomorphisms can be found in [32].

2.2.3 Split graphs

A graph G is *split* [28] if its vertices can be partitioned into two sets $V(G) = V_A \cup V_B$ such that vertices in V_A are independent in G and vertices in V_B induce a clique in G. Split graphs are well studied and have many interesting graph theoretic properties. They are chordal, and the complement $\overline{G}$ of a split graph G is also a split graph. They can also be characterized by three forbidden subgraphs as illustrated in Figures 2.4.

Theorem 2.2.5 *[25] A graph G is split if and only if it does not contain $2K_2$, C_4 and C_5 as an induced subgraph.* □

Recognition and many optimization problems can be done efficiently for split graphs. Note that the recognition of spit graphs can be done in linear time $O(n + m)$ [25]; a great improvement to the generic algorithm described in Chapter 1. Later in Chapter 4, we will classify a family of graphs that closely resembles split graphs and show that they too have a finite set of minimal obstructions.

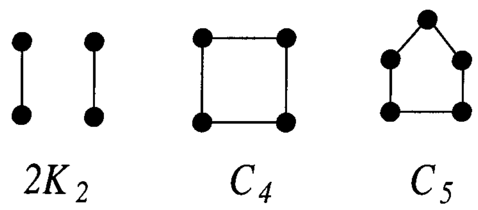


Figure 2.4: Forbidden induced subgraphs for split graphs

The question of whether a graph G is split is modeled by trigraph homomorphisms as well. Consider trigraph H in Figure 2.5. By definition a graph G is split if it can be partitioned into a clique and an independent set. In a homomorphism of G to H, vertices mapped to a must be independent and vertices mapped to b must induce a clique. Thus, we have the following proposition.

Proposition 2.2.6 *[17] Let G be a graph and H the trigraph in Figure 2.5 . Graph G is split if and only if it admits an H-colouring* □

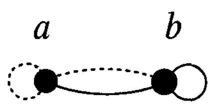


Figure 2.5: A trigraph used to characterize split graphs

2.2.4 (a, b)-graphs

A graph G is an *(a,b)-graph* [2] if its vertices can be partitioned into sets $V(G) = V_1 \cup \ldots \cup V_a \cup V_{a+1} \cup \ldots \cup V_{a+b}$ such that for $1 \leq i \leq a$, vertices in set V_i are independent in G and for $a < j \leq a + b$, vertices in set V_j induce a clique in G. In other words, G can be partitioned into a independent sets and b cliques. It is not difficult to see that (a, b)-graphs are a generalization of split graphs. By definition, a $(1, 1)$-graph is precisely a split graph. When a is greater than 2 or when b is greater than 2, the complexity of recognizing an (a, b)-graph is NP-complete and can be be reduced to 3-colouring [27]. In [5], Brandstadt et. al. investigated a special case of (a, b)-graphs. They showed that the recognition of $(2, 1)$-graphs and $(1, 2)$-graphs are polynomial time solvable. Later in [18], Feder et. al. showed more generally that partition problems of a certain type are polynomial time solvable. Their original proof implies the following.

Theorem 2.2.7 *[18] Let G be a graph and $a, b \neq 3$ with $a + b \leq 4$. There is a polynomial time algorithm to decide whether G is an (a, b)-graph.* □

The problem becomes more interesting when we restrict graph G to be chordal. Feder et. al. showed that there is exactly one forbidden subgraph for a chordal (a, b)-graph.

Theorem 2.2.8 *[20] A chordal graph is an (a, b)-graph if and only if it does not contain $(b+1)K_{a+1}$ as an induced subgraph.* □

They also presented a simple $O(n(m+n))$ algorithm for the recognition of chordal (a, b)-graphs. Again, this is a significant improvement over the generic algorithm from Chapter 1.

Let H be a trigraph with a 0-vertices and b 1-vertices and only $*$-edges.

Proposition 2.2.9 *Let G be a graph. Graph G is an (a, b)-graph if and only if it admits an H-colouring.* □

For example, consider trigraphs H_1, H_2 in Figure 2.6. A graph G is a $(1, 2)$-graph if and only if there is a homomorphism of G to H_1 and a graph G is a $(2, 2)$-graph if and only if there is a homomorphism of G to H_2.

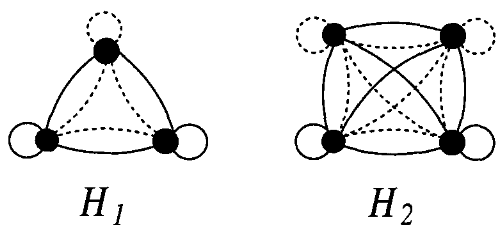


Figure 2.6: Trigraphs used to characterize (1,2)-graphs and (2,2)-graphs

2.2.5 Cutsets

A *cutset* in a connected graph G is a set of vertices $C \subset V(G)$ such that $G - C$ is disconnected. Some of these problems require $G[C]$ to be a clique, stable set, or complete bipartite. Cutset problems are of interest because certain types come up in solutions to many optimization problems [42] and in the recognition of perfect graphs [10].

A *clique cutset* in a connected graph G is a cutset C such that the subgraph $G[C]$ is a clique. Equivalently [46], G has a clique cutset if and only if it can be partitioned into three non-empty sets $V(G) = V_1 \cup V_2 \cup V_3$ such that vertices of V_3 induce a clique in G and every vertex of V_1 is non-adjacent to every vertex of V_2 in G.

Theorem 2.2.10 *[46] Let G be a graph. A clique cutset in G can be found in polynomial time.* □

A *stable cutset* in a connected graph G is a cutset C such that the subgraph $G[C]$ is an independent set of vertices. Equivalently [43], G has a stable cutset if and only if it can be partitioned into three non-empty sets $V(G) = V_1 \cup V_2 \cup V_3$ such that vertices of V_3 are independent in G and every vertex if V_1 is non-adjacent to every vertex of V_2 in G.

Theorem 2.2.11 *[23] Let G be a graph. Finding a stable cutset in G is NP-complete.* □

A *skew cutset* in a connected graph G is a cutset C such that the subgraph $G[C]$ can be partitioned into two non-empty sets U and V such that each vertex of U is adjacent to each vertex of V. Equivalently [18], G has a skew cutset if and only if it can be partitioned into four non-empty sets $V(G) = V_1 \cup V_2 \cup V_3 \cup V_4$ such that every vertex of V_2 is adjacent to every vertex of V_4 in G and every vertex of V_1 is non-adjacent to every vertex of V_3 in G. In [18], a quasi-polynomial time algorithm, of complexity ($n^{O(log\ n)}$), is given for finding a skew cutset in a graph. It was later shown by du Figueiredo et. al. to be polynomial time solvable.

Theorem 2.2.12 *[24] Let G be a graph. A skew cutset in G can be found in polynomial time.* □

All three of the cutset problems described can be modeled as a surjective trigraph homomorphism.

Proposition 2.2.13 *Let G be a graph and H_1 the trigraph in Figure 2.7. There is a clique cutset in G if and only if there is a surjective homomorphism f of G to H_1* □

Note that the vertices of a connected graph G mapped to vertex c in H_1 is a clique cutset of G.

Proposition 2.2.14 *Let G be a graph and H_2 the trigraph in Figure 2.7. There is a stable cutset in G if and only if there is a surjective homomorphism f of G to H_2* □

Note that the vertices of a connected graph G mapped to vertex c in H_2 is a stable cutset of G.

Proposition 2.2.15 *Let G be a graph and H_3 the trigraph in Figure 2.7. There is a skew cutset in G if and only if there is a surjective homomorphism f of G to H_3* □

Note that the vertices of a connected graph G mapped to vertex c or d in H_3 is a skew cutset of G.

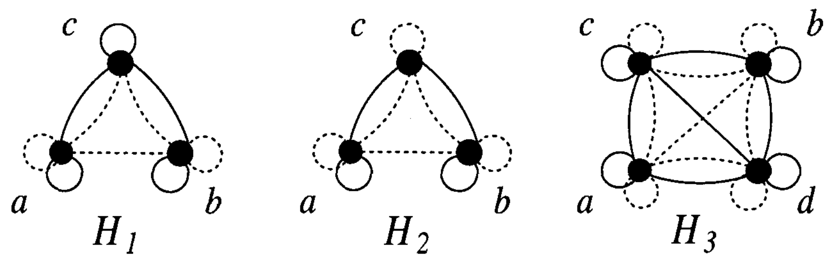


Figure 2.7: Trigraphs used to model cutset problems

Notice that surjective trigraph homomorphisms generalize many cutset problems. Indeed it is possible to describe many other types of cutset problems in addition to the ones given above. For example, a *split graph cutset*, in a connected graph G is a cutset C of G such that the subgraph $G[C]$ is a split graph. Finding a split graph cutset can be modeled by trigraph H_1 illustrated in Figure 2.8 in the same way as the previous cutset problems. A *2-independent clique cutset* in a connected graph is a cutset C such that the subgraph $G[C]$ consists of two non-adjacent cliques. Again, finding such a cutset can be modeled by trigraph H_2 in Figure 2.8. By using trigraphs to model cutset problems, we can restrict the cutset to have any property that can be modeled by a trigraph. For example, we may restrict the cutset to be an (a, b)-graph, n-cliques, or n-independent clique cutset.

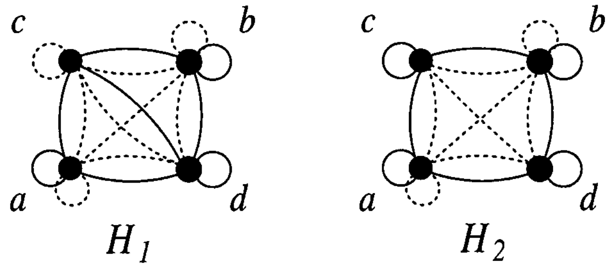


Figure 2.8: Two trigraphs used to model new cutset problems

2.2.6 The Winkler problem

The *Winkler Problem* proposed by Winkler in 1998 asks whether a graph G can be partitioned into four non-empty sets $V(G) = V_1 \cup V_2 \cup V_3 \cup V_4$ with the following restrictions:

- every vertex in V_1 is non-adjacent to every vertex in V_3
- every vertex in V_2 is non-adjacent to every vertex in V_4
- at least one adjacency between vertices in V_1 and V_2; V_2 and V_3; V_3 and V_4; and V_4 and V_1.

Equivalently, the problem can be stated as a homomorphism of a graph G to a cycle on four vertices C_4 with the third restriction. The problem was shown to be NP-complete by Vikas a year later in [44].

Let G be a graph and H a trigraph with four $*$-vertices a, b, c, d such that ab, cd are 0-edges and the remaining pairs ac, ad, bc, bd are $*$-edges as illustrated in Figure 2.9. A surjective trigraph homomorphism f of G to H with the added restriction that for each $*$-edge xy in H, some vertex in $f^{-1}(x)$ is adjacent to some vertex in $f^{-1}(y)$ corresponds to the Winkler Problem.

As we have shown, trigraph homomorphisms of varying types model many different kinds of well-known partition and characterization problems. Truly, trigraphs are very versatile in modeling these and other types of problems which gives us further motivation to study them.

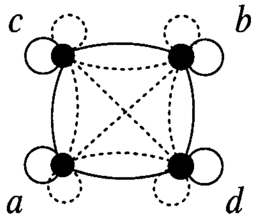


Figure 2.9: A trigraph used to model the Winkler problem

2.3 Tools

Recall from Chapter 1 that the trigraph homomorphism problem is very similar to the graph homomorphism problem. In this section we introduce some tools that will be helpful in proving some of the results in this thesis. Some of the results and tools have been taken from the study of graph homomorphisms and adapted for trigraphs, others have been taken and adapted from the study of M-partition problems and some are specific to trigraphs.

2.3.1 Retraction and domination

Let H be a trigraph and H' an induced subtrigraph of H. A *retraction* f of H to H' is the trigraph homomorphism f of H to H' such that $f(x) = x$ for all $x \in V(H')$. If there is a retraction of H to H' we say H *retracts* to H', and H' is a *retract* of H. Alternatively, retractions can be expressed in terms of their adjacency matrices. Recall the adjacency matrix of a trigraph H is denoted by M_H. There is a retraction of H to H' if the following conditions hold:

- $f(x) = x$ for all $x \in V(H')$
- $M_H(u, v) = M_{H'}(f(u), f(v))$ or $M_{H'}(f(u), f(v)) = *$

Since retracts are not unique, there may be many induced subgraphs of H that trigraph H retracts to. The *core* of a trigraph is an induced subtrigraph H' of H such that H' is a retract of H and there is no proper induced subtrigraph H'' of H' that is a retract of H. Let H_1, H_2 be two trigraphs with cores H_1', H_2' respectively. Trigraph H_1 is *homomorphically equivalent* to H_2 if and only if H_1' is isomorphic to H_2'. We have adapted the preceding terminology which was originally introduced in [32] for graph homomorphism to apply to trigraph homomorphism as well. Figure 2.10 illustrates three homomorphically equivalent trigraphs H_1, H_2, H_3, with a core isomorphic to H_3.

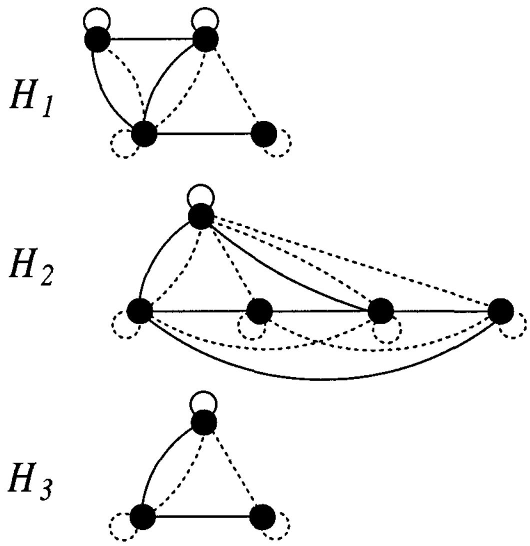


Figure 2.10: Three homomorphically equivalent trigraphs

Let G be any graph and H a trigraph with adjacency matrix M_H. Let u, v be two distinct vertices in H corresponding to two rows and two columns in M_H. A row (column) u *dominates* row (column) v in matrix M_H if for any $w \in V(H)$ we have $M(u,w) = M(v,w)$ or $M(u,w) = *$. Domination was introduced by Feder et. al. in [18] as a tool for solving M-partition problems. It was proposed that if u dominates v, then there is a homomorphism of G to H if and only if there is a homomorphism of G to $H - v$. Let H_q be the subtrigraph of H after repeated deletion of q dominating vertices in H (we may assume $H_0 = H$). Notice that for $0 \leq j \leq i \leq q$, H_i is a retract of H_j. Thus, if we maximize q and remove all dominating vertices from H, the resulting subtrigraph H_q is the core of H. We present the following two facts regarding retracts and homomorphically equivalent trigraphs.

Proposition 2.3.1 *Let r be a retraction of trigraph H to core R. Trigraph G admits a homomorphism f of G to H if and only if G admits a homomorphism f' of G to R.*

Proof. (To prove from left to right). Suppose G admits a homomorphism f of G to H. Since there is a retraction r of H to R any vertices placed in vertex $x \in V(H)$ can be placed in $r(x)$ instead. Thus, if there is a homomorphism f of G to H, then there is a homomorphism $f' = r \circ f$ of G to R. (To prove from right to left). Suppose G admits a homomorphism f' of G to R. Since R is a core of H, it is isomorphic to some induced subgraph of H. Thus, there is a homomorphism $f = f'$ of G to H. □

Corollary 2.3.2 *Let H_1, H_2 be two homomorphically equivalent trigraphs. Then the complete set of minimal obstructions σ_{H_1} to H_1 is the same as the complete set of minimal obstructions σ_{H_2} to H_2.*

Proof. Since H_1 and H_2 are homomorphically equivalent, they have an isomorphic core R. Thus by Proposition 2.3.1 $\sigma_{H_1} = \sigma_{H_2}$. □

2.3.2 2-SAT

A basic technique for solving 'simple' list trigraph homomorphism problems is to reduce the problem to the 2-satisfiability problem [18] known to be polynomial time solvable. The 2-satisfiability problem is defined as follows [27]:

Instance: A set S of boolean variables and a collection C of clauses over S with each clause in C having at most two literals.

Problem: Is there a satisfying truth assignment for C?

We will now describe a polynomial time reduction from an instance of the list trigraph homomorphism problem to the 2-satisfiability problem.

Let H be a trigraph with two vertices and G a graph with list L. Suppose each vertex $v \in V(G)$ is assigned an ordered set (list) $L_v \subseteq V(H)$ containing two elements $L_v = \{a, b\}$ or one element $L_v = \{a\}$. Let x_v be the literal corresponding to vertex $v \in V(G)$ and $S = \{x_v : v \in V(G)\}$ the set of variables. We interpret the truth assignment $x_v = 1$ to mean v maps to a and an assignment $x_v = 0$ to mean v does not map to a. We now construct clauses for the 2-satisfiability problem from the trigraph homomorphism problem. Note that there are two types of clauses. Each type imposes a restriction that corresponds to some constraint dictated by the trigraph.

The first set of clauses corresponds to constraints that are imposed by each vertex $b \in V(H)$. Suppose b is a 0-vertex. By definition, any homomorphism f of G to H cannot map adjacent vertices $u, v \in G$ to b. Thus, for every pair of adjacent vertices u, v in G with $b \in L_u$ and $b \in L_v$, we construct a clause that restricts no more than one of u or v to be mapped to b. For example, if $L_u = \{a, b\}$ and $L_v = \{b, c\}$, we construct the clause $(x_u \vee \overline{x_v})$. Similar clauses are constructed for the case when b is a 1-vertex. For example, again, if $L_u = \{a, b\}$ and $L_v = \{b, c\}$, we construct both the clauses $(\overline{x_u} \vee x_v)$ and $(x_u \vee x_v)$.

The second set of clauses corresponds to constraints that are imposed by the 0-edges and 1-edges in H. Suppose zz' is a 0-edge in H. Again, by definition, any homomorphism f of G to H cannot map adjacent vertices $u, v \in G$ to z, z' respectively. Thus, for every pair of adjacent vertices u, v in G with $z \in L_u$ and $z' \in L_v$, we construct a clause that prevents both u from mapping to z and v from mapping to z' simultaneously. For example, if $L_u = \{w, z\}$ and $L_v = \{w, z'\}$, we construct the clause $(x_u \vee x_v)$. Again, similar clauses are constructed for the case when zz' is a 1-edge. The collection of all clauses C (from the first and second set) become the input to the 2-satisfiability problem.

The same technique also applies as long as each list L_i has at most two vertices. We can now formulate the following theorem.

Theorem 2.3.3 *[18] There is a polynomial time algorithm which solves any list M-partition problem restricted to the instances in which the list of every vertex of the input graph has at most two vertices.* □

2.3.3 Odd girth and chromatic number

Erdos [12] first showed that there exist graphs with arbitrarily large chromatic number and girth. The construction of these graphs is difficult. The following slightly weaker statement is easy to prove.

Theorem 2.3.4 *[33] Let g and k be positive integers, $g \geq 3$ odd. Then there exists a graph $S(g, k)$ with odd girth greater than g and chromatic number greater than k.* □

The only difference between the two theorems is that the weaker one allows short even cycles. This result will be used in the next chapter to show the construction of infinitely many minimal obstructions to certain types of homomorphisms.

2.3.4 Labeled and unlabeled graphs

A graph G is *labeled* if each vertex $v \in V(G)$ is assigned an A or B label. Let G be a labeled graph and H be a trigraph. Recall a labeled homomorphism of G to H is a homomorphism of G to H such that A labeled vertices map to 0-vertices and B labeled vertices map to 1-vertices. As with the (unlabeled) homomorphism problems, a *labeled obstruction* G is a labeled graph that does not admit a labeled homomorphism of G to H. A *minimal labeled obstruction* is a labeled obstruction such that for any vertex $v \in V(G)$, the subgraph $G - v$ admits a labeled homomorphism of G to H.

Let H_0 and H_1 be the induced subgraphs of H on 0-vertices and 1-vertices respectively. A graph G that admits both a homomorphism to H_0 and a homomorphism to H_1 has at most $r = |V(H_0)| \times |V(H_1)|$ vertices [18]. Feder and Hell [15] proved that a bound on the size of a minimal labeled obstruction to H implies a bound on the size of a minimal (unlabeled) obstructions to H. In other words, if H has finitely many minimal labeled obstructions then H has finitely many minimal (unlabeled) obstructions.

Theorem 2.3.5 *[15] If every minimal labeled obstruction to H has at most t vertices, then an minimal (unlabeled) obstruction to H has at most $2t^{2r+1}$ vertices.* □

This theorem will be used to help us prove finitely many minimal obstructions in some of the more difficult cases of trigraphs. Examples of these difficult trigraphs will come up in Chapters 4 and 5. In addition, in Chapter 5 we will prove the converse of Theorem 2.3.5 stated below.

Theorem 2.3.6 *Let H be a trigraph. If every minimal unlabeled obstruction to H has at most t vertices, then a minimal labeled obstruction to H has at most t vertices.* □

2.4 General results

We now review how the tools discussed above are used to derive general results on M-partitions of arbitrary graphs. These have been focused on small matrices (when the size of matrix M is at most four) [7, 18, 24], and on matrices without $*$ entries [14]. In trigraph terminology, these problems focus on trigraphs H with at most four vertices, and trigraphs H without $*$-edges.

2.4.1 Small matrices

Feder et. al. investigated in [18] the basic trigraph homomorphism problem on general graphs, as well as the more general list trigraph homomorphism problem on general graphs. Their results again focused on relatively small trigraphs of no more than four vertices. In particular, recall from Theorem 2.3.3 that all list trigraph homomorphism problems for trigraphs with two vertices are reducible to 2-SAT, and hence polynomial time solvable. They further classified the list trigraph homomorphism problems for all trigraphs on three vertices.

Theorem 2.4.1 *[18] Let H be a trigraph with at most three vertices. Then the list trigraph homomorphism problem on H is NP-complete when H corresponds to the 3-colouring problem, or to the stable cutset problem, and is polynomial time solvable otherwise.* □

When the trigraph H has four vertices, they have only shown that the list trigraph homomorphism is NP-complete when H has an induced subgraph isomorphic to the trigraphs corresponding to the two stated problems (3-colouring and stable cutset). For all other trigraphs with four vertices, they have shown the list trigraph homomorphism problem to have a quasi-polynomial time algorithm [18]. Cameron et. al. [7] extended the results and showed that up to four vertices, with the exception of the trigraph named the 'stubborn problem' illustrated in Figure 2.11 (and its complement), all other list trigraph homomorphism problems not classified by Theorem 2.4.2 are indeed polynomial time solvable. Whether there is a polynomial time algorithm for stubborn problem still remains open.

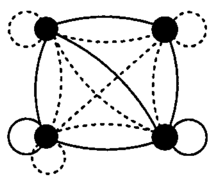


Figure 2.11: Trigraph that models the 'stubborn problem'

Theorem 2.4.2 *[18] Let H be a trigraph with at most four vertices with no $*$-vertex. If H contains an induced subgraph corresponding to the 3-colouring or stable cutset problem then* *the list homomorphism problem on H is NP-complete; and it is polynomial time solvable otherwise.* □

Note that the above theorem implies the classification of all basic trigraph homomorphism problems for trigraphs with four vertices.

2.4.2 Full homomorphisms

In this section, we will be discussing M-partitions in the case when the matrix M has no $*$ entries; we will do this in the language of full homomorphisms. Let G and H be graphs. A *full homomorphism* of G to H is a mapping $f : V(G) \to V(H)$ such that uv is an edge of G if and only if $f(u)f(v)$ is an edge in H. Again, full homomorphisms can be modeled by trigraph homomorphisms. Let G' and H' be the trigraphs resulting from applying convention 1 to graphs G and H respectively. There is a full homomorphism of G to H if and only if there is a trigraph homomorphism of G' to H'. In [14], Feder and Hell proved that determining whether G admits a full homomorphism to H is characterized by a finite set of minimal obstructions, and therefore polynomial time solvable. They showed that each minimal obstruction O to a full homomorphism of graph G to graph H is bounded by at most $(k+1)(l+1)$ vertices such that k is the number of vertices with loops and l is the number of vertices without loops in H. A different proof can be found in [6]. We also provide a simple proof that the size of a minimal obstruction to a full homomorphism is bounded in Chapter 4.1.

Theorem 2.4.3 *[14] Let H be a trigraph with l 0-vertices and k 1-vertices. Then every minimal obstruction to H has at most $(l+1)(k+1)$ vertices, and there is at most one minimal obstruction with exactly $(l+1)(k+1)$ vertices.* □

2.4.3 Directed graphs

Trigraph Homomorphisms can be further generalized if we consider directed graphs and directed trigraphs. The adjacency matrix M_H of a directed trigraph is not necessarily symmetric. In [21], Feder, Hell and Tucker-Nally gave a complete classification of the list trigraph homomorphisms for directed trigraphs H with at most three vertices

Theorem 2.4.4 *[21] Let G be a directed graph and H a directed trigraph with at most three vertices. Then the list trigraph homomorphism problem for H is NP-complete if H is isomorphic to one of the trigraphs in Figure 2.12 and polynomial otherwise.* □

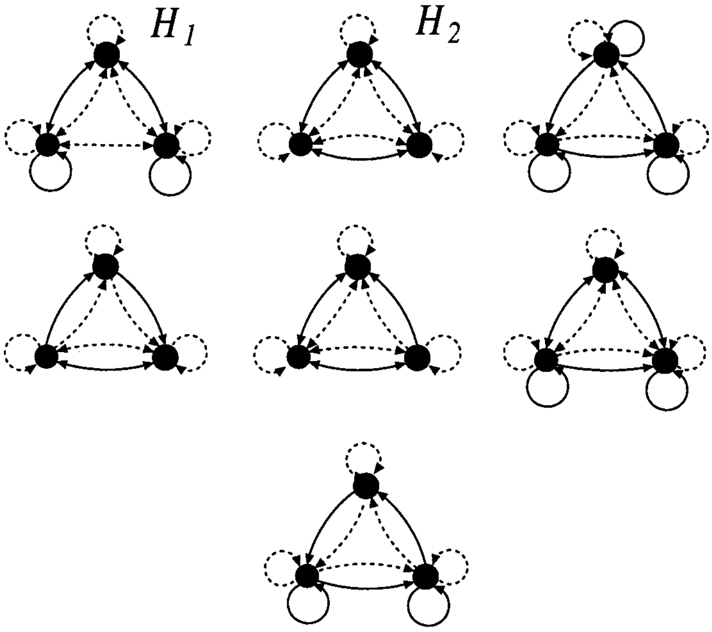


Figure 2.12: NP-complete directed trigraphs

Note that the first two trigraphs H_1, H_2 in Figure 2.12 correspond to the 3-colouring problem and the stable cutset problem which are well known to be NP-complete. Recall that with the exception of the trigraph corresponding to the stubborn problem (and its complement), (undirected) list trigraph homomorphisms problems have been classified as NP-complete or polynomial time solvable for trigraphs with up to four vertices. Since a polynomial time algorithm for the stubborn problem is seemingly difficult, it is likely difficult to classify all list trigraph homomorphism problems for all directed trigraphs with four vertices.

2.5 Homomorphisms of graph families

So far, the trigraph homomorphism problems that we have discussed applies to general graphs. In [16], Feder and Hell showed that all list trigraph homomorphism problems are NP-complete or solvable in quasi-polynomial time ($n^{O(log\ n)}$). Although some quasi-polynomial time algorithms have been improved to polynomial time, a true dichotomy of which list trigraph homomorphism problems are NP-complete and which are polynomial is still not known. Again, an example of this is the trigraph illustrated in Figure 2.11, which corresponds to the 'stubborn problem'. Note that even for the easier variation, basic trigraph homomorphism, a dichotomy is not yet known.

As with many graph theoretic problems, a natural question is to ask the complexity of a problem when restricted to certain families of graphs. Indeed, there are results to the trigraph homomorphism problem when the input graph is restricted to the family of perfect graphs [15], cographs [17], and chordal graphs [19, 20]. Many trigraph homomorphism problems, when restricted to one of these three families, become polynomial time solvable when their counterpart, general graphs, are NP-complete. We note, however, that there are still problems that remain NP-complete even in these restricted cases. As well, there are some problems where the complexity remains unknown or became unknown in the restricted cases. Thus, for certain families of graphs, dichotomy for trigraph homomorphisms (with or without lists), may not be known. We will review the main results for each of these three graph families below.

2.5.1 Perfect graphs

Dichotomy for trigraph homomorphisms (without list) is not known even for perfect graphs. The results that are known only apply to restricted families of trigraphs. In [15], Feder and Hell showed that when restricted to normal trigraphs, the trigraph homomorphism problem is polynomial time solvable. Recall again that l is the number of 0-vertices in H and k is the number of 1-vertices in H.

Theorem 2.5.1 *[15] Let H be a normal trigraph with $l \leq k$. A minimal (perfect) obstruction to H has at most $2(k+1)^{2lk+1}$ vertices* □

They also included an improved bound of $(k+1)(l+1)$ vertices for three subsets of normal trigraphs. Recall S_A is the set of edges incident to two 0-vertices, S_B is the set of edges incident to two 1-vertices, and S_C is the set of edges incident to one 0-vertex and one 1-vertex. The first of these subsets is a normal trigraph with the property that sets S_A, S_B, S_C all contain no $*$-edges. Note that this has already been proven for general graphs by Theorem 2.4.3. The second subset of normal trigraphs restricts vertices in S_B to contain only 1-edges, and the third restricts vertices in S_B to contain only 0-edges.

2.5.2 Cographs

Feder, Hell and Hochstattler [17] were the first to consider trigraph homomorphisms for cographs. They showed that when restricted to cographs, there is a bound on the size of a minimal obstruction to every trigraph homomorphism problem.

Theorem 2.5.2 *[17] Let H be a trigraph with l 0-vertices and k 1-vertices. A minimal (cograph) obstruction to H has at most* $O(8^{l+k}/\sqrt{l+k})$ *vertices.* □

This implies that there are finitely many minimal (cograph) obstructions to the trigraph homomorphisms restricted to cographs. They also proved a similar result for the more general homomorphism variation, list trigraph homomorphism, again showing that there are finitely many minimal (cograph) obstructions (with lists) to list trigraph homomorphisms restricted to cographs. For constant trigraphs, bounds for (non-list) homomorphisms are greatly improved.

Theorem 2.5.3 *[17] Let H be a constant trigraph. A minimal (cograph) obstruction to H has at most* $(k+1)(l+1)$ *vertices.* □

Cographs are the largest family of graphs that have been shown to have bounded minimal obstruction sizes for the list trigraph homomorphism problem. Unfortunately, the other two families that we review do not share this property.

2.5.3 Chordal graphs

Recall that chordal graphs are a subclass of perfect graphs, thus all of the results for perfect graphs apply to chordal graphs as well. However, for chordal graphs, Feder et. al. [19]

showed that the bound on minimal (chordal) obstructions could be improved upon when normal trigraphs are further restricted.

Let H be a normal trigraph and S_A, S_B, S_C, S_D a partition of its pairs $V(H) \times V(H)$ as defined in Section 2.1.3. The improved bound on minimal (chordal) obstructions are as follows. Note that the case when S_A, S_B, and S_C all contain $*$-edges has been discussed earlier in Section 2.2.4.

- If S_A contains no $*$-edges and S_C contains all only $*$-edges then the a minimal (chordal) obstruction to H has at most $2^{(6k+3)l+1}l^l$ vertices.
- If S_A and S_C contains only $*$-edges then a minimal (chordal) obstruction to H has at most $2(l+1)^{(4l+2)k+2}$ vertices.

By imposing stricter restrictions on H, these bounds can be further improved as follows:

- If S_A contains only 1-edges and S_B, S_C contain only $*$-edges then the a minimal (chordal) obstruction to H has at most $2(2k+2)^l$ vertices.
- If S_A contains only 1-edges, S_B contains only 0-edges and S_C contains only $*$-edges then the a minimal (chordal) obstruction to H has at most $2(8k^2+25k+5)^l$ vertices.
- If S_A, S_C contains only $*$-edges and S_B contains only 0-edges then the a minimal (chordal) obstruction to H has at most $2(l+1)^{(l+2)k+1}$ vertices.

Surprisingly, there are certain kinds of the basic trigraph homomorphism problem that are NP-complete, even when restricted to chordal graphs.

Theorem 2.5.4 *[19] There exists a trigraph H for which the basic trigraph homomorphism problem restricted to chordal graphs is NP-complete.* □

However, for certain kinds of trigraphs H, the basic and list trigraph homomorphism problems for chordal graphs can be solved in polynomial time. In particular, these are trigraphs H for which the adjacency matrix M_H has no restrictions on S_A, S_B, but has restrictions on S_C.

Theorem 2.5.5 *[19] Let H be a trigraph. If S_C consists of only $*$-edges, or has no $*$-edges, then the list trigraph homomorphism problem for H, restricted to chordal graphs, is polynomial time solvable.* □

2.6 Table of known results

Below we give a table detailing the results that have been reviewed and some of our new results. Each row in the table details a particular trigraph homomorphism and its known results. The first column specifies restrictions on the input graph G. In this column, any rows marked by the superscript Φ indicate that the result is new and will be presented in later chapters. The second column specified restrictions on the fixed trigraph H with l 0-vertices and k 1-vertices. The shorthand 'E' is used to mean edge(s) and 'V' to mean vertex(ices). Some list homomorphism problems marked by the superscript Ψ indicate that the results hold even when $*$-vertices are allowed. The third column specifies whether the problem has a polynomial time algorithm. An entry of FMMO indicates the problem was classified polynomial by finitely many minimal obstructions. For relatively smaller obstructions, the bound on the number of vertices is given. The fourth column indicates whether the problem has infinitely many minimal obstructions (IMMO). This column is primarily for results new to this thesis. The last column indicates what type of dichotomy is known, if any, for the given problem.

Table 2.1: Table of known results

Graph G	*Trigraph H*	*Polynomial*	*IMMO*	*Dichotomy*
List Digraph	Any	-	-	open
List Digraph	≤ 3 vertices$^{\Psi}$	-	-	P/NPc
Digraph	Any	-	-	open
List General	Any	-	-	Quasi-P/NPc
List General	≤ 2 vertices	O(n)	-	-
General	Any	-	-	Quasi-P/NPc
General	no $*$-E	$(k+1)(l+1)$	-	-
General	≤ 4 vertices	-	-	P/NPc
General$^{\Phi}$	≤ 5 vertices	-	-	FMMO/IMMO
General$^{\Phi}$	Messy	-	IMMO	-
General$^{\Phi}$	Nice	FMMO	-	-
List Perfect	Any	-	-	open
Perfect	Any	-	-	open
Perfect	Normal	FMMO	-	-
Perfect	Normal, S_C all 0-E	$(k+1)(l+1)$	-	-
Perfect	Normal, S_C all 1-E	$(k+1)(l+1)$	-	-
List Cograph	Any$^{\Psi}$	FMMO	-	-
Cograph	Any	FMMO	-	-
Cograph	Constant	$(k+1)(l+1)$	-	-
List Chordal	Any	-	-	open
List Chordal	no $*$-E in S_A	-	-	Quasi-P/NPc
List Chordal	no $*$-E in S_B	-	-	Quasi-P/NPc
List Chordal	no $*$-E in S_A, no 1-E in S_C	-	-	P/NPc
List Chordal	no $*$-E in S_A, no 0-E in S_C	-	-	P/NPc
List Chordal	only 0-V	$O(nl(2l)^l)$	-	-
Chordal	Any	-	-	open
Chordal	only $*$-E	$O(n(n+m))$	-	-

Chapter 3

Infinitely many minimal obstructions

We shall assume from now on that all trigraphs do not have any $*$-vertices.

We start our investigation into minimal obstructions to trigraph homomorphism by dealing with a family of trigraphs that have infinitely many minimal obstructions (IMMO). A trigraph H is *messy* if it has two vertices u, v such that both u, v are 0-vertices or both u, v are 1-vertices and the pair uv is a $*$-edge. Graphically, trigraph H is messy if and only if it has an induced subtrigraph of H isomorphic to B or $\overline{B}$ in Figure 3.1. For convenience, the trigraphs B and $\overline{B}$ will refer to the trigraphs in Figure 3.1 for the remainder of the thesis. We prove in this chapter that every messy trigraph H has infinitely many minimal obstructions. It is interesting to note at this point that we will later provide an example trigraph H that has infinitely many minimal obstructions, but is not messy. We will devote all of Chapter 6 to this trigraph and the obstructions to it.

Before we begin the main proofs in this chapter, we first need to define some terminology and review a classical result for 2-colouring. Let G be a graph. The *odd circumference* $c(G)$ of G is the length of the longest odd cycle in G; if G has no cycles, then we define $c(G)$ to be infinite. The *odd girth* of $g(G)$ of G is the length of the shortest odd cycle in G; if G does not contain an odd cycle, then we again define $g(G)$ to be zero. Let H be a

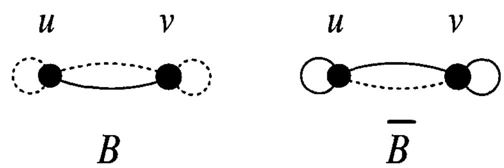


Figure 3.1: A trigraph B and its complement $\overline{B}$

trigraph. A set of minimal obstructions $\sigma = \{G_1, G_2, \ldots\}$ to H is *useful* if σ is infinite and $g(G_i) \leq c(G_i) < g(G_{i+1})$ for $i \geq 1$.

In [36], Konig first showed that a graph G admits a 2-colouring if and only if G has no odd cycle. In our terminology this result can be formulated as follows.

Lemma 3.0.1 *A graph G admits a B-colouring if and only if G has no odd cycle.* □

Corollary 3.0.2 *Both B and $\overline{B}$ have infinitely many minimal obstructions.*

Proof. Let C be an odd cycle. By Lemma 3.0.1, C does not admit a B-colouring, hence it is an obstruction to B. Let v be a vertex of C. The graph $C-v$ is a path, and again by Lemma 3.0.1, $C - v$ admits a B-colouring. Thus, C is a minimal obstruction to B. By Proposition 1.3.1, since C is a minimal obstruction to B, the graph $\overline{C}$ is a minimal obstruction to $\overline{B}$. Since there are infinitely many odd cycles, there are infinitely many minimal obstructions to B and $\overline{B}$. □

3.1 Messy trigraphs have IMMO

In this section we will extend the result in Corollary 3.0.2 to all messy trigraphs H. The proof is broken into three parts. The first part shows that a messy trigraph without any 1-vertices has infinitely many minimal obstructions. The actual statement will be stronger, and will also be used in the second and third parts. The second and third parts, when applied recursively, will result in infinitely many minimal obstructions to any messy trigraph.

Theorem 3.1.1 *Let H be a messy trigraph with no 1-vertices. Then H has a useful set of minimal obstructions.*

Proof. Since H has no 1-vertices, it must have an induced subgraph isomorphic to B. Let $m = |V(H)|$ and $g = 4$. By Theorem 2.3.4 there exists a graph G_1 with odd girth g_1 greater than g and chromatic number greater than m. In particular, G_1 does not admit an m-colouring. We note that G_1 is an obstruction to H; indeed any homomorphism of G_1 to H would be an m-colouring of G_1.

Let G'_1 be an induced subgraph of G_1 such that G'_1 is a minimal obstruction to H. By Lemma 3.0.1, G'_1 must contain an odd cycle, otherwise G'_1 would admit a B-colouring and thus it would also admit an H-colouring since H contains B. The graph G'_1, of non-zero odd girth $g_1 = g(G'_1)$ and finite odd circumference $c_1 = c(G'_1)$, will be our first obstruction to H. We can now recursively construct infinitely many additional minimal obstructions as follows.

Let G_i be a minimal obstruction to H with odd circumference c_i and odd girth g_i. By Lemma 2.3.4 there exists a graph G_{i+1} with odd girth g_{i+1} greater than c_i and chromatic number greater than m. As above, G_{i+1} does not admit an H-colouring, and we let G'_{i+1} be an induced subgraph of G_{i+1} such that G'_{i+1} is a minimal obstruction to H. Again, G'_{i+1} must contain an odd cycle, and hence an odd cycle of length at least g_{i+1} vertices. The odd girth g'_{i+1} of G'_{i+1} is strictly larger than the odd circumference c'_i of G'_i. Note that G'_{i+1} is not isomorphic to G_i or any G_j with $j < i$. Applying this technique recursively, we construct a useful set σ_H containing infinitely many minimal obstructions to H. □

We now consider messy trigraphs H with both 0-vertices and 1-vertices, and distinguish whether H contains B or $\overline{B}$.

Theorem 3.1.2 *Assume H contains B and x is a 1-vertex of H. If $H - x$ admits a useful set of minimal obstructions σ, then H also admits a useful set of minimal obstructions σ'.*

Proof. Let O_i be any obstruction in σ and let the graph $G_i = 2O_i$ be 2 disjoint and non-adjacent copies of O_i. Suppose there is a homomorphism f of G_i to H. Since O_i is an obstruction to $H - x$ there is at least one vertex $v \in O_i$ mapped to x by f. Since G_i consists of 2 disjoint and non-adjacent copies of O, there are two independent vertices v_1, v_2

in G_i (taken from each copy of O_i in G_i) that are mapped to x by f. Vertex x is a 1-vertex, thus vertices v_1 and v_2 must be adjacent and we have a contradiction. Therefore G_i is an obstruction to H.

Let O_i' be an induced subtrigraph of G_i such that O_i' is a minimal obstruction to H. Trigraph H contains B, thus by Theorem 3.0.1, O_i' must contain an odd cycle. Recall σ is useful and $g(O_i) \leq c(O_i) < g(O_{i+1}) \leq c(O_{i+1})$. Since O_i' is an induced subgraph of $2O_i$ we have $g(O_i) \leq g(O_i') \leq c(O_i') \leq c(O_i)$. It follows that $g(O_i') \leq c(O_i') < g(O_{i+1}') \leq c(O_{i+1}')$. Thus, applying this technique to every obstruction O_i in the useful set σ we construct a useful set σ'. □

Let H be a messy trigraph with an induced subgraph isomorphic to B and let H' be the induced subgraph of H consisting of 0-vertices in H. Applying Theorem 3.1.1 we can get an useful set of minimal obstructions $\sigma_{H'}$ to H'. We can recursively add 1-vertices to H' and apply Theorem 3.1.2 to find an useful set of minimal obstruction σ_H to H. Thus, we have the following fact.

Corollary 3.1.3 *Let trigraph H contain B . Then H has infinitely many minimal obstructions.* □

For trigraphs H with an induced subgraph isomorphic to $\overline{B}$, we can use apply the same theorems on the complement $\overline{H}$ of H. Note that $\overline{H}$ has an induced subgraph isomorphic to B.

Theorem 3.1.4 *Let trigraph H contain $\overline{B}$. Then H has infinitely many minimal obstructions.*

Proof. Let $\overline{H}$ be the complement of trigraph H. By Theorems 3.1.1 and 3.1.2, there is a useful set of minimal obstructions σ' to $\overline{H}$. Let $G \in \sigma'$ be any minimal obstruction to $\overline{H}$. By Proposition 1.3.1, $\overline{G}$ is a minimal obstruction to H. The set σ' contains infinitely many minimal obstructions to $\overline{H}$. Therefore, there are infinitely many minimal obstructions to H. □

A messy trigraph H has an induced subgraph isomorphic to B or $\overline{B}$. By Corollary 3.1.3 a trigraph containing B has infinitely many minimal obstructions. By Theorem 3.1.4, a trigraph containing $\overline{B}$ has infinitely many minimal obstructions. Thus, we have the following fact.

Corollary 3.1.5 *Let H be a messy trigraph. Then H has infinitely many minimal obstructions.* □

Chapter 4

Finitely many minimal obstructions

In the previous chapter, we introduced the family of messy trigraphs. A messy trigraph H has an induced subgraph isomorphic to B or $\overline{B}$ in Figure 3.1. Recall that messy trigraphs have infinitely many minimal obstructions. A trigraph is *clean* if it is not messy. In other words, a clean trigraph has no induced subgraph isomorphic to B or $\overline{B}$. In this chapter, we will only consider clean trigraphs. We also introduce two trigraph families that have finitely many minimal obstructions. In the next chapter, we will use the results from Chapters 2, 3, and 4 to complete our classification of all trigraphs up to five vertices.

4.1 Split-friendly trigraphs

We first provide our simple proof of Theorem 2.4.3, that the size of a minimal obstruction to a full homomorphism is bounded. Recall l is the number of 0-vertices in H, k is the number of 1-vertices in H and $m = k + l$. Although our bound of $m(2max(k, l) + 2)$ vertices is not as tight as the one detailed in [14], it still implies there are only finitely many minimal obstructions.

In a graph G, two vertices $u, v \in V(G)$ are *similar* if u and v have the same set of neighbours other than u and v. That is $N(u) - v = N(v) - u$. (Note that the definition applies both when u is adjacent to v, and when u is non-adjacent to v). Without loss of generality, we assume $l \leq k$. (We can ensure this by complementation).

Theorem 4.1.1 *Let H be a trigraph with $l \leq k$ and without $*$-edges. Then every minimal obstruction to H has at most $m(2k+2)+1$ vertices.*

Proof. Suppose a graph G is a minimal obstruction to H with more than $m(2k+2)+1$ vertices. Thus G has at least $m(2k+2)+2$ vertices. Let v be any vertex in G. Since G is a minimal obstruction, the subgraph $G-v$ must admit an H-colouring f. Since $G-v$ has at least $m(2k+2)+1$ vertices and H has m vertices, by the pigeonhole principle, there exists a vertex x in H such that the set S of vertices mapped to x by f has at least $2k+3$ vertices. Since there are no $*$-vertices and no $*$-edges in H, all vertices in the set S are similar in $G-v$. Suppose x is a 0-vertex. (An analogous argument can be made if x is a 1-vertex.) Each pair vu, $u \in S$, is either an edge or non-edge in G, hence there is a set $T \subset S$ with $k+2$ vertices that are similar in G.

Therefore G contains an independent set T of $k+2$ similar vertices. Let $t \in T$. The subgraph $G-t$ must again admit an H-colouring g since G is minimal. Since $T-t$ has $k+1$ independent vertices, g must map at least one of them, say $t' \neq t$, to a 0-vertex y in H since $l \leq k$. Since t' is similar to t in G, and the pair tt' is a 0-edge, t may be mapped to y as well, and we have a contradiction. Therefore every minimal obstruction G has at most $m(2k+2)+1$ vertices. □

A trigraph H is *split-friendly* if every pair of 0-vertices uu' in H is a 0-edge and every pair of 1-vertices vv' in H is a 1-edge. In other words, S_A has only 0-vertices and S_B has only 1-vertices. Figure 4.1.2 illustrates 3 split-friendly trigraphs H_1, H_2 and H_3. In trigraph H_3, we have $S_A = \{xy, xz, yz\}$ and $S_B = \{uv, uw, vw\}$. Note that a split-friendly trigraph is clean. Let G be a graph and H be a split-friendly trigraph. Recall from Section 2.2.3 that if G admits an H-colouring, then G must be a split graph. However, not every split graph admits an H-colouring.

Theorem 4.1.2 *A split-friendly trigraph has finitely many minimal obstructions.*

Proof. We will consider two cases in our proof; when the split-friendly graph H has no $*$-edges and when it has at least one $*$-edge. First, suppose H has no $*$-edges. By Theorem 4.1.1, every minimal obstruction to H has at most $m(2k+2)+1$ vertices. In other words, H has finitely many minimal obstructions.

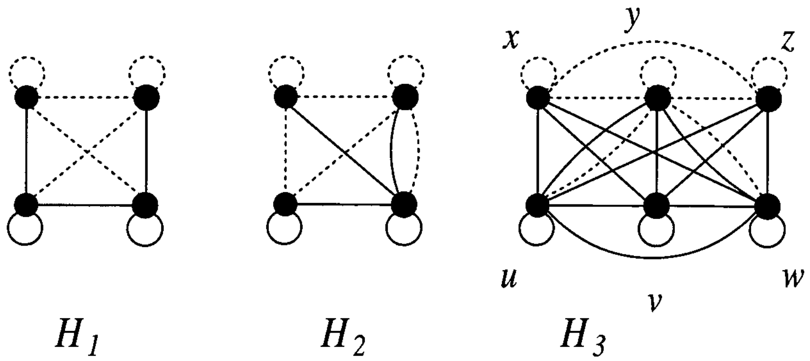


Figure 4.1: Split-friendly trigraphs

Now suppose H has a $*$-edge uv. Since H is split-friendly, the edge uv must be incident to a 0-vertex and a 1-vertex. Without loss of generality, let u be a 0-vertex in H and v be a 1-vertex in H. Let H' be a subtrigraph of H induced by only the vertices u, v. Note that H' is isomorphic to Figure 4.2. We will define a retraction f from H to H' as follows.

- $f(x) = u$ whenever x is a 0-vertex in H.
- $f(x) = v$ whenever x is a 1-vertex in H.

Now we show that f is a retraction. For this, verify that:

- If pair $xy \in E(H)$ then $f(x)f(y) \in E(H')$.
- If pair $xy \in N(H)$ then $f(x)f(y) \in N(H')$.

If the pair $xy \in E(H)$ then xy is not a 0-edge. In a split-friendly graph, this means that vertices x and y are not both 0-vertices, so either both x and y are 1-vertices, or one is a 0-vertex and the other a 1-vertex. Hence $f(x)f(y) = vv$ or uv, and both $vv \in E(H')$ and $uv \in E(H')$. An analogous argument can be made when we have the pair $xy \in N(H)$.

By Corollary 2.3.2, the complete set of minimal obstructions σ_H to H is the same as the complete set of minimal obstructions $\sigma_{H'}$ to H'. It is well known [28] that $\sigma_{H'} = \{C_4, C_5, 2K_2\}$, thus H also has the same set of minimal obstructions. Therefore H has finitely many minimal obstructions. □

Figure 4.2: A trigraph used to characterize split graphs

4.2 Labeled trigraphs

It is often easier to prove that there are only finitely many minimal obstructions for labeled graphs than for unlabeled graphs. Let H be a trigraph and G a labeled graph. A *labeled homomorphism* f of G to H is a homomorphism from G to H such that A-labeled and B-labeled vertices in G map to 0-vertices and 1-vertices in H respectively. A labeled graph G is a *labeled obstruction* to H if there is no labeled homomorphism of G to H. A labeled graph G is a *minimal labeled obstruction* to a trigraph H if G is a is a labeled obstruction such that for any vertex $v \in V(G)$, the subgraph $G - v$ (with the same labels) admits a labeled homomorphism of G to H.

Let G' be the unlabeled graph underlying the labeled graph G. (In other words G' is obtained from G by deleting all labels.) Recall l and k are the number of 0-vertices and 1-vertices in H respectively and $r = l \times k$. By Theorem 2.3.5, if a labeled obstruction to H has at most t vertices, then an unlabeled obstruction to H has at most $2t^{2r+1}$ vertices. Thus, if H has finitely many minimal labeled obstructions then H also has finitely many minimal unlabeled obstructions. In this section, we only consider labeled graphs G; however, by Theorem 2.3.5, these results will apply to unlabeled graphs G as well.

4.2.1 Similarity

For this chapter, we will use a weaker definition of similarity than the one used in Section 4.1. Let G be a labeled graph. A pair of A-labeled vertices $u, v \in V(G)$ is *similar* if u and v are non-adjacent and have the same A-labeled neighbours. A pair of B-labeled vertices $u, v \in V(G)$ is *similar* if uv and v are adjacent and have the same B-labeled neighbours other than u and v. Note that an A-labeled vertex is never similar to a B-labeled vertex.

Similarity is defined for trigraphs as well. Let H be a trigraph. Recall that in this chapter we assume that trigraph H is clean. A pair of 0-vertices $u, v \in V(H)$ is *similar* if for every 0-vertex w, the following holds:

- the pair wu is a 0-edge if and only if the pair wv is a 0-edge.
- the pair wu is a 1-edge if and only if the pair wv is a 1-edge.

Note that similar 0-vertices u, v have a 0-edge uv. This is evident when we choose w to be either u or v. Since trigraph H is clean, it does not have an induced subgraph isomorphic to B or $\overline{B}$ and so, wu and wv cannot be a $*$-edge. Similar 1-vertices are defined analogously. Note again that a 0-vertex is never similar to a 1-vertex.

It is easy to check that similarity is an equivalence relation on the vertices of both a graph and a trigraph. Thus we obtain a partition of the vertices into equivalence classes which we shall call the *similarity classes.*

Identifying similarity classes is helpful in solving the trigraph homomorphism problems. We first prove some facts involving similarity classes. These results will be used in the last theorem in this chapter to prove that our second trigraph family has finitely many minimal obstructions.

Lemma 4.2.1 *Let H be a clean trigraph, G a labeled graph, and f a labeled homomorphism of G to H. If u and v are non-similar vertices of G, then $f(u)$ and $f(v)$ are non-similar vertices of H.*

Proof. Suppose u and v are both A-labeled vertices. Since u, v are non-similar, there exists an A-labeled vertex $w \in V(G)$ that distinguishes u and v. In other words, one of wu, wv is an edge and the other a non-edge. Note that neither wu nor wv can be a $*$-edge since H is clean.

Now consider the vertices $f(u)$ and $f(v)$. Both are 0-vertices since u and v are A-labeled vertices and f a labeled homomorphism. Without loss of generality, suppose w is adjacent to u and non-adjacent to v. The relation $f(w)f(u)$ cannot be a 0-edge and the relation $f(w)f(v)$ cannot be a 1-edge. Since H is clean, it follows that $f(w)f(u)$ must be a 1-edge and $f(w)f(v)$ must be a 0-edge. The vertex $f(w)$ distinguishes $f(u)$ and $f(v)$, therefore vertices $f(u), f(v)$ are non-similar in H. □

Let N^A, N^B be the number of similarity classes of A-labeled vertices and B-labeled vertices in G, respectively. Let N^0, N^1 be the number of similarity classes of 0-vertices and 1-vertices in H, respectively. The following fact involving the number of similarity classes in G and H is derived directly from the previous lemma.

Corollary 4.2.2 *Let H be a clean trigraph, and G a labeled graph. If there exists a labeled homomorphism of G to H, then $N^A \leq N^0$ and $N^B \leq N^1$.*

Proof. Suppose $N^A > N^0$, (analogously $N^B > N^1$). Let $v_1, \ldots, v_{N^A}$ be non-similar vertices taken one from each similarity class of A-labeled vertices in G, and let f be a homomorphism of G to H. By Lemma 4.2.1, the vertices $f(v_1), \ldots, f(v_{N^A})$ are not similar in G. There are only N^0 similarity classes in H, so by the pigeonhole principle, at least two distinct vertices v_i, v_j have $f(v_i) = f(v_j)$, and we have a contradiction. Therefore we have $N^A \leq N^0$ and $N^B \leq N^1$. □

Note that the converse of Lemma 4.2.1 does not hold. Let H be the trigraph and G the graph illustrated in Figure 4.3. Trigraph H has 4 similarity classes (each containing one vertex). Graph G has two similarity classes: one class contains vertex b and the other vertices a and c. In a homomorphism f of G to H, vertices a and c may map to non-similar vertices of H. For example, the homomorphism f maps vertices a, b, c to vertices w, x, y respectively. However, note that no vertex of G was mapped by f to z.

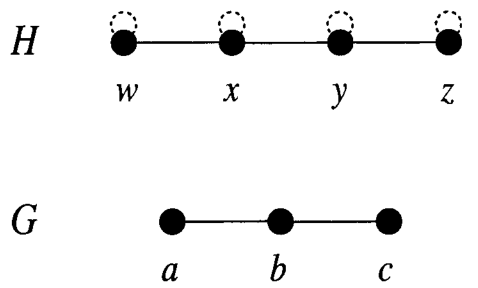


Figure 4.3: A trigraph H and a graph G

A labeled homomorphism f of labeled graph G to trigraph H is *sim-surjective* if for each similarity class C of H there exists a vertex $x \in V(G)$ with $f(x) \in C$.

Lemma 4.2.3 *Let H be a clean trigraph, G a labeled graph, and f a sim-surjective homomorphism of G to H. If u is similar to v in G, then $f(u)$ is similar to $f(v)$.*

Proof. Let u, v be A-labeled vertices, and $f(u), f(v)$ non-similar 0-vertices. (We proceed analogously if u, v are B-labeled vertices and $f(u), f(v)$ are non-similar 1-vertices). Since $f(u), f(v)$ are non-similar, by definition there exists a 0-vertex z adjacent to one of $f(u), f(v)$ and non-adjacent to the other. In other words, one of the pairs $zf(u)$ or $zf(v)$ is a 0-edge and the other is a 1-edge. Let C be the similarity class containing z. By assumption, there exists a 0-vertex z' in C that has an A-labeled vertex x of G mapped to z' by f. Thus, x must be adjacent to one of u, v but not the other, and we have a contradiction to the fact that u, v are similar. □

Corollary 4.2.4 *Let H be a clean trigraph and G a labeled graph. If $N^A < N^0$ or $N^B < N^1$ then no labeled homomorphism f of G to H is sim-surjective.*

Proof. Let $N^A < N^0$. (We proceed analogously if $N^B < N^1$). Suppose f is sim-surjective. By Lemma 4.2.3, similar vertices u, v are mapped by f to similar vertices $f(u), f(v)$. Thus, vertices of one similarity class in G are mapped by f to vertices of one similarity class in H. Since the number of similarity classes involving A-labeled vertices is less than the number of similarity classes involving 0-vertices, there exists a similarity class C of 0-vertices with $f(x) \notin C$ for all $x \in V(G)$. Therefore, f is not sim-surjective. □

4.2.2 Nice trigraphs

A trigraph H is *nice* if it is clean and has no similar 0-vertices or has no similar 1-vertices. Figure 4.4 illustrates two nice trigraphs H_1 and H_2. Trigraph H_1 has no similar 0-vertices and trigraph H_2 has no similar 1-vertices.

Let H be a trigraph, G a labeled graph and G^A, G^B denote the induced subgraphs of G on the A-labeled and B-labeled vertices respectively. Let H^0, H^1 denote the induced subtrigraphs of H on the set of 0-vertices and 1-vertices respectively. An *AB-mapping* is a mapping $s : V(G) \to V(H)$ such that s is a homomorphism of both G^A to H^0 and of G^B to H^1. We denote by $MAP(G, H)$ the set of all AB-mappings $s : V(G) \to V(H)$, and

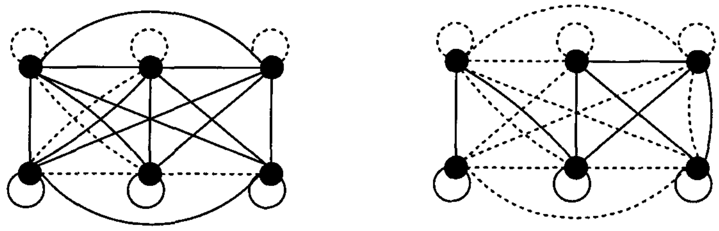


Figure 4.4: Two nice trigraphs

$HOM(G, H)$ the set of all homomorphisms $f : V(G) \to V(H)$. Recall that in a homomorphism of G to H, all A-labeled vertices are mapped to 0-vertices and all B-labeled vertices are mapped to 1-vertices. Thus, it is not hard to see that $HOM(G, H) \subseteq MAP(G, H)$. In other words, f is a homomorphism of G to H only if f is also an AB-mapping of G to H. Note that the only difference between a mapping in $MAP(G, H)$ and a mapping in $HOM(G, H)$ is that a mapping in $HOM(G, H)$ guarantees edges incident to both an A-labeled vertex and a B-labeled vertex is preserved, while a mapping in $MAP(G, H)$ does not guarantee this.

Let H be a clean trigraph and $\Re_H$ the set containing all proper induced subgraphs of H obtained from H by deleting one similarity class; let also $\Re_H^0$ (respectively $\Re_H^1$) denote the subset of $\Re_H$ obtained by deleting a similarity class of 0-vertices (respectively of 1-vertices). Recall again that l denotes the number of 0-vertices in H, and k denotes the number of 1-vertices in H. Our main result in this chapter can be formulated as follows.

Theorem 4.2.5 *Let H be a nice trigraph. Suppose for each H' in $\Re_H$, all minimal labeled obstructions to H' have at most t vertices. Then every minimal labeled obstruction to H has at most*

$$max(N^0t,\ N^1t,\ N^0+1,\ N^1+1,\ k,\ l,\ N^0!N^1!(k+1))$$

vertices.

Proof. We will assume H has no similar 0-vertices. (We proceed analogously if H has no similar 1-vertices). Our proof is broken into three cases. Each case will address a different property a minimal labeled obstruction may have. In the first case we consider the case when $N^A < N^0$ or $N^B < N^1$, in the second case we consider when $N^A > N^0$ or $N^B > N^1$, and in the third case we consider when $N^A = N^0$ and $N^B = N^1$. For each case, we show a bound for a minimal labeled obstruction to H.

Case 1: Consider a min. labeled obst. G with $N^A < N^0$ (analogously with $N^B < N^1$).

Since G does not admit a labeled homomorphism to H, it does not admit a labeled homomorphism to any $H' \in \Re_H$. There are exactly N^0 trigraphs H' in $\Re^0_H$. Each such H' identifies a subset S of at most t vertices of G such that $G[S]$ does not admit an H'-colouring. Let G' be the subgraph of G induced by the union of all such sets S. Then G' has at most $N^0 t$ vertices. Note that G' also satisfies $N^A < N^0$. Thus, Corollary 4.2.4 implies that there is no sim-surjective homomorphism of G' to H. In other words, for any homomorphism f of G' to H there is a class C in H with $f(x) \notin C$ for all $x \in V(G')$. We know from the proof of Corollary 4.2.4 that class C contains only 0-vertices of G'. Now the definition of G' ensures that G' does not have any homomorphism to H. Thus the minimality of G implies that $G = G'$, i.e., that G has at most $N^0 t$ vertices.

Case 2: Consider a min. labeled obst. G with $N^A > N^0$ (analogously with $N^B > N^1$).

Let G' be an induced subgraph of G with $N^0 + 1$ non-similar A-labeled vertices. By Corollary 4.2.2, G' is a labeled obstruction to H. Thus, the minimality of G implies that $G = G'$ and hence has at most $N^0 + 1$ vertices.

Case 3: Consider a minimal labeled obstruction G with $N^A = N^0$ and $N^B = N^1$

We now may assume that $MAP(G, H) \neq \emptyset$, else the minimal labeled obstruction G has at most k or l vertices. We claim that G has at most $N^0! N^1! (k + 1)$ vertices.

Let $f \in MAP(G, H)$ be any AB-mapping. By definition, f is a homomorphism of both G^A to H^0 and of G^B to H^1. By Lemma 4.2.1, non-similar vertices u, v in G have $f(u), f(v)$ non-similar in H. Since $N^A = N^0$ and $N^B = N^1$, a homomorphism of G^A to H^0 maps vertices of one similarity class in G^A only to vertices of one similarity class in H^0. (Analogously for a homomorphism of G^B to H^1). In other words, f assigns to each similarity class in G, a unique similarity class in H. Let A_i, B_j be the similarity classes of A-labeled and B-labeled vertices in G respectively and Y_i, Z_j be the similarity classes of 0-vertices and 1-vertices in H respectively, with $1 \leq i \leq N^0$ and $1 \leq j \leq N^1$. By assumption, each class Y_i consists of a single vertex since H has no similar 0-vertices.

For each bijective assignment of A_i to Y_i and B_j to Z_j we shall identify $k+1$ vertices used to ensure that $f \notin HOM(G,H)$ for any f corresponding to this assignment. Without loss of generality, let each A_i be assigned to Y_i and each B_j be assigned to Z_j. Since each Y_i has a single vertex, any f corresponding to this assignment maps all vertices of A_i to this vertex. If each B-labeled vertex v, say $v \in B_j$, could be mapped to some vertex in the corresponding Z_j so that the edges and non-edges to the A-labeled vertices are preserved by f, we would have $f \in HOM(G,H)$. Thus there exists a B-labeled vertex $v \in B_j$ for which each choice of $u \in Z_j$ has an A-labeled vertex w preventing v from mapping to u. Since there are at most k vertices in any B_j, we obtain at most k of these vertices w, which together with v induce a labeled subgraph of G with at most $k+1$ vertices that does not admit a homomorphism to H.

Recall that there are N^0 similarity classes of 0-vertices and N^1 similarity classes of 1-vertices, thus there are $N^0!N^1!$ possible assignments. For each of the assignments, a minimal labeled obstruction has at most $k+1$ vertices, thus a labeled obstruction for any possible assignment has at most $N^0!N^1!(k+1)$ vertices. Therefore, the minimal labeled obstruction G (for this case) has at most $max(k,\ l,\ N^0!N^1!(k+1))$ vertices.

Every minimal labeled obstruction G falls under one of these cases, thus a minimal labeled obstruction to H has at most

$$max(N^0t,\ N^1t,\ N^0+1,\ N^1+1,\ k,\ l,\ N^0!N^1!(k+1))$$

vertices.

□

Chapter 5

Trigraphs with up to five vertices

We have now built up enough tools in the previous three chapters to complete our classification of finitely or infinitely many minimal obstructions to all trigraphs of up to five vertices. To summarize, any messy trigraph H with an induced subgraph isomorphic to B or $\overline{B}$ in Figure 3.1 has infinitely many minimal obstructions, we will apply previous results to deduce that all other trigraphs of up to five vertices have finitely many minimal obstructions. Recall that we made the basic assumption that trigraphs do not have $*$-vertices.

In this chapter, all minimal obstructions for trigraphs with one and two vertices will be provided. For trigraphs with more than two vertices, the precise set of minimal obstructions to a homomorphism of a graph G to a trigraph H is difficult and/or tedious to prove in many instances. However, we will show that all trigraphs with three and four vertices are covered by theorems presented in previous chapters, thus classifying them as having infinitely or finitely many minimal obstructions. Of the remaining trigraphs with five vertices, we will show that most are covered by theorems in the previous chapters as well. A detailed proof of the remaining exceptional cases of trigraphs with five vertices that are not covered will be given in the last section of this chapter. A clean trigraph H is *exceptional* when it has a pair of similar 0-vertices, a pair of similar 1-vertices and it is not split-friendly. In other words, exceptional trigraphs are clean and non-nice and not split-friendly. We will show that all exceptional trigraphs with five vertices have finitely many minimal obstructions.

5.1 Trigraphs with one vertex

There are only three trigraphs H_1, H_2 and H_3 with exactly one vertex as illustrated in Figure 5.1. We included H_3 for completeness; however note that it is not considered since it is a $*$-vertex. Trigraph H_1 is simply a 0-vertex, and trigraph H_2 is simply a 1-vertex. It is easy to verify that each has exactly one minimal obstruction.

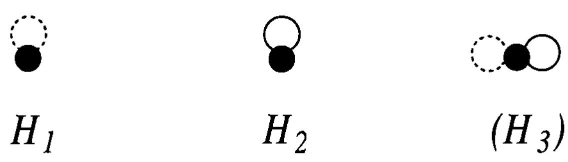


Figure 5.1: Trigraphs with exactly one vertex.

Theorem 5.1.1 *There is only one minimal obstruction to H_1, namely, K_2.*

Proof. Clearly a graph G without edges admits a homomorphism to H_1; and if G contains K_2 then it does not. □

Theorem 5.1.2 *There is only one minimal obstruction to H_2, namely $\overline{K_2}$.*

Proof. By Theorem 5.1.1, the only minimal obstruction to H_1 is K_2. Trigraph H_2 is the complement of trigraph H_1, thus by Proposition 1.3.1, the graph $\overline{K_2}$ is the only minimal obstruction to H_2. □

5.2 Trigraphs with two vertices

Excluding trigraphs with a $*$-vertex, there are only 9 non-isomorphic trigraphs with two vertices. The first three trigraphs H_4, H_5, H_6 are illustrated in Figure 5.2. The common trait shared by these three trigraphs is that they consist of only 0-vertices. Since there are only three kinds of (non-loop) edges and each trigraph has one edge, there are only three trigraphs with this trait. Trigraphs H_4 has a 0-edge, H_5 has a 1-edge and H_6 has a $*$-edge.

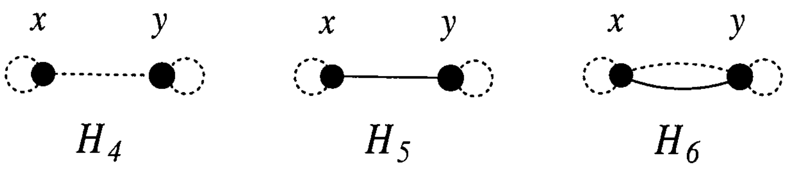


Figure 5.2: Trigraphs with two 0-vertices

Trigraph H_4 consists of two 0-vertices u, v with 0-edge uv. There is an obvious retraction of H_4 to an induced subgraph isomorphic to H_1. In other words, H_4 and H_1 are homomorphically equivalent. Thus, by Corollary 2.3.2, they have the same set of minimal obstructions.

Theorem 5.2.1 *There is only one minimal obstruction to H_4, namely K_2.* □

A graph G that admits an H_5-colouring can be partitioned into two sets $V(G) = V_x \cup V_y$ such that both V_x, V_y are independent sets and every vertex of V_x is adjacent to every vertex of V_y. In other words, G is a complete bipartite graph.

Theorem 5.2.2 *There are only two minimal obstructions to H_5, namely $\overline{P_3}$ and K_3.*

Proof. Graphs $\overline{P_3}$ and K_3 are illustrated in Figure 5.3. It is easy to verify that graphs $\overline{P_3}$ and K_3 are obstructions to H_5. We claim that these are the only minimal obstructions to H_5. Let G be a minimal obstruction to H_5 that is non-isomorphic to $\overline{P_3}$ and K_3. Let v be any vertex in G and x, y vertices in H_5. Since G is minimal, the subgraph $G - v$ admits an H-colouring f. The mapping f partitions subgraph $G - v$ into two sets $V(G) = V_x \cup V_y$. Recall V_x and V_y are independent sets in G, and every vertex of V_x is adjacent to every vertex of V_y. We now consider vertex v and its adjacencies in G.

If v is adjacent to no vertices in G and V_x (or analogously V_y) is empty, then v may be placed in V_y (or V_y), and we have a contradiction.

If v is adjacent to no vertices in G, and sets V_x and V_y are non-empty, then G has an induced subgraph isomorphic to $\overline{P_3}$ and we have a contradiction.

If v is adjacent to vertex v' in V_x, and non-adjacent to vertex v'' in V_x, then the subgraph induced by vertices v, v', v'' is isomorphic to $\overline{P_3}$ and we have a contradiction. A similar argument applies to set V_y.

If v is adjacent to vertex v' in V_x and adjacent to vertex v'' in V_y, then the subgraph induced by vertices v, v', v'' is isomorphic to K_3, and we have a contradiction.

If v is adjacent to every vertex in V_x and non-adjacent to every vertex in V_y, then v may be placed in V_y, and we have a contradiction. An analogous argument applies when v is adjacent to every vertex in V_y and non-adjacent to every vertex in V_x. Therefore, the only minimal obstructions to H_5 are $\overline{P_3}$ and K_3. □

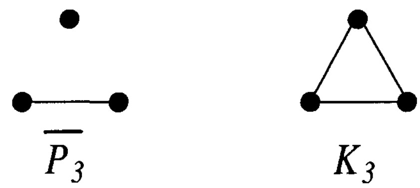


Figure 5.3: Obstructions to H_5

Trigraph H_6 is a messy trigraph, so by Corollary 3.1.5, it has infinitely many minimal obstructions. A graph admits a H_6-colouring if and only if it is bipartite. In other words, the trigraph homomorphism problem for trigraph H_6 models the problem of deciding whether a graph is bipartite or not. Forbidden subgraph characterization for bipartite graphs are well known. These subgraphs are the minimal obstructions to H_6.

Theorem 5.2.3 *[28] The minimal obstructions for H_6 are exactly the odd cycle C_n, $n \geq 3$.* □

The next three non-isomorphic trigraphs with two vertices, H_7, H_8, H_9, are illustrated in Figure 5.4. The common trait for these three trigraphs is that every vertex in H_7, H_8, H_9 is a 1-vertex. Trigraphs H_7 has a single 1-edge, H_8 a single 0-edge and H_9 a single $*$-edge.

It is easy to see that trigraphs H_7, H_8, and H_9 are the complements of H_4, H_5, and H_6. Since obstructions are known for trigraphs H_4 and H_5, by Proposition 1.3.1, they are

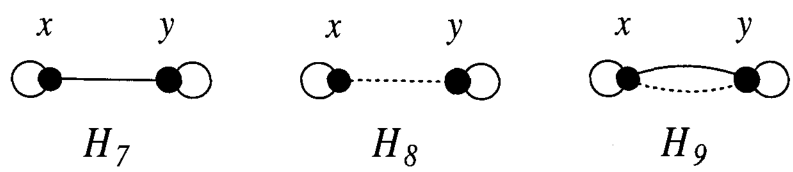


Figure 5.4: Trigraphs with two 1-vertices

known for H_7 and H_8. Trigraph H_9 is messy, so by Corollary 3.1.5 it has infinitely many minimal obstruction. In particular, these obstructions are the complements of odd cycles.

Theorem 5.2.4 *There is only one minimal obstruction to H_7, namely $\overline{K_2}$.* □

Theorem 5.2.5 *There are only two minimal obstructions to H_8, namely P_3 and $\overline{K_3}$.* □

Theorem 5.2.6 *[28] The minimal obstructions for H_9 are exactly the complements of odd cycle $\overline{C_n}$, $n \geq 3$.* □

The final three non-isomorphic trigraphs H_{10}, H_{11}, H_{12} with two vertices are illustrated in Figure 5.5. Again, the common trait for the three trigraphs here is that each trigraph has exactly one 0-vertex and one 1-vertex. Trigraphs H_{10} has a 0-edge, H_{11} has a 1-edge and H_{12} has a $*$-edge.

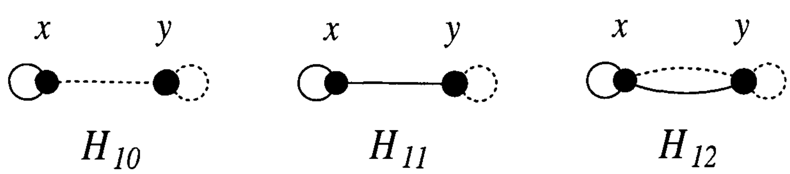


Figure 5.5: Trigraphs with one 0-vertex and one 1-vertex

Theorem 5.2.7 *There are only two minimal obstructions to H_{10}, namely $2K_2$ and P_3.*

Proof. Graphs $2K_2$ and P_3 are illustrated in Figure 5.6. It is easy to verify that they are minimal obstructions to H_{10}. We claim that these are the only minimal obstructions to H_{10}. Let G be a minimal obstruction to H_{10} that is non-isomorphic to $2K_2$ and P_3. Let v be any vertex in G and x, y vertices in H_{10}. Since G is minimal, the subgraph $G - v$ admits an H-colouring f. The mapping f partitions subgraph $G - v$ into two sets $V(G) = V_x \cup V_y$. We shall assume $|V_x| \neq 1$. In other words, if $G - v$ is an independent set, f maps all vertices of subgraph $G - v$ to y. The sets V_x induces a clique in G, V_y an independent set in G, and every vertex of V_x is non-adjacent to every vertex of V_y. We now consider vertex v and its adjacencies in G.

If v is adjacent to no vertex in G then v may be placed in V_y and we have a contradiction.

If v is adjacent to every vertex in V_x and no vertex in V_y then v may be placed in V_x and we have a contradiction.

If v is adjacent to two vertices v', v'' in V_y, then the subgraph induced by vertices v, v', v'' is isomorphic to P_3.

If v is adjacent to vertex v' in V_y and V_x is empty, then both v, v' may be place in V_x and we have a contradiction.

If v is adjacent to vertex v' in V_y and V_x has at least two vertices u, u', then the subgraph induced by vertices v, v', u, u' is isomorphic to $2K_2$.

If v is adjacent to vertex v' in V_x and non-adjacent to vertex v'' in V_x, then the subgraph induced by vertices v, v', v'' is isomorphic to P_3. Therefore, the only minimal obstructions to H_{10} are $2K_2$ and P_3. □

It is easy to see that trigraph H_{11} is the complement of trigraph H_{10}. So by Corollary 3.1.5, the only minimal obstructions to H_{11} are C_4 and $\overline{P_3}$.

Theorem 5.2.8 *There are only two minimal obstructions to H_{11}, namely C_4 and $\overline{P_3}$.* □

Trigraph H_{12} is isomorphic to the trigraph illustrated in Figure 2.5. In other words, every graph that admits a H_{12}-colouring is a split graph. Recall that the only minimal obstructions to H_{12} are $2K_2$, C_4, and C_5.

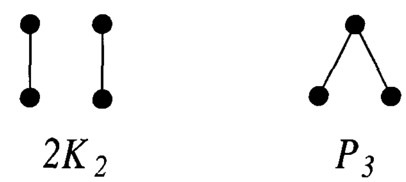


Figure 5.6: Obstructions to H_{10}

Theorem 5.2.9 *There are only three minimal obstructions to H_{12}, namely $2K_2$, C_4, and C_5.* □

Note that trigraphs H_6, and H_9 are the only two with infinitely many minimal obstructions since they are isomorphic to either B or $\overline{B}$ in Figure 3.1. The other trigraphs in this section are in fact covered by applying Theorems 4.1.1 and 2.3.5. In other words, they have both finitely many minimal labeled obstructions and finitely many minimal unlabeled obstructions.

5.3 Trigraphs with three vertices

As we alluded to earlier, obtaining the precise obstructions to trigraphs with three or more vertices is tedious and in some cases difficult to prove. Instead, we will show that our previous results from Chapters 2, 3 and 4 can be applied to prove that there are only finitely many minimal obstructions. In the previous section, we showed that trigraphs H_4, H_5, and H_{10} all have finitely many minimal obstructions. By applying Proposition 1.3.1, we showed that the complements to H_4, H_5, and H_{10}, namely trigraphs H_7, H_8, and H_{11} respectively have finitely many minimal obstructions. In this, and the following sections, we will consider up to complementation, non-isomorphic trigraphs that are clean.

Let $\zeta(a, b)$ denote the family of clean trigraphs with a 0-vertices and b 1-vertices. Let H be a clean trigraph with exactly three vertices. All clean trigraphs with three vertices belong to one of the 4 families $\zeta(0, 3)$, $\zeta(1, 2)$, $\zeta(2, 1)$, and $\zeta(3, 0)$. Up to complementation, families $\zeta(0, 3)$ and $\zeta(1, 2)$ are equivalent to families $\zeta(3, 0)$ and $\zeta(2, 1)$ respectively. Therefore, up to complementation, all non-isomorphic, clean trigraphs with three vertices belong to one of the following families.

- $\varsigma(3,0)$
- $\varsigma(2,1)$

Recall that by Corollary 3.1.5, messy trigraphs have infinitely many minimal obstructions. It is easy to see that a trigraph H in the family $\varsigma(3,0)$ has no $*$-edges. Thus, by Theorem 4.1.1, every $\varsigma(3,0)$ trigraph has finitely many minimal obstructions.

Trigraph H of the family $\varsigma(2,1)$ has exactly one 1-vertex, thus it is impossible to have two similar 1-vertices. So by Theorem 4.2.5 every $\varsigma(2,1)$ trigraph has finitely many minimal labeled obstructions and by Theorem 2.3.5 every $\varsigma(2,1)$ trigraph has finitely many minimal unlabeled obstructions.

5.4 Trigraphs with four vertices

In the previous sections, we have used Theorem 2.3.5 to show a trigraph with finitely many minimal labeled obstruction has finitely many minimal unlabeled obstructions. We now formulate the following theorem for the converse.

Theorem 5.4.1 *Let H be a trigraph with m vertices. If every minimal unlabeled obstruction to H has at most t vertices, then any minimal labeled obstruction to H has at most t vertices.*

Proof. Suppose G is a minimal labeled obstruction with more than t vertices. Thus, G has at least $t+1$ vertices. Let unlabeled graph G' be the result of replacing each A-labeled vertex u in G with an independent set I_u of size $m+1$ and each B-labeled vertex v in G with a clique K_v of size $m+1$. Graph G' is an obstruction to H since at least one vertex from each independent set I_u must map to a 0-vertex and at least one vertex from each clique K_v must map to a 1-vertex. Thus, there is a minimal induced subgraph G'' of G' that is an obstruction to H. By assumption, G'' has at most t vertices, so all vertices of G'' must come from no more than t distinct sets I_u or K_v. Since there are at least $t+1$ such sets, at least one set S may be removed from G' such that the subgraph $G'-S$ is an obstruction to H.

Let x be the vertex in G corresponding to set S. Since G is minimal, the subgraph $G-x$ admits a homomorphism f to H. Let unlabeled graph G^* be the result of replacing each

A-labeled vertex u in $G - x$ with a independent set I_u of size $m + 1$ and each B-labeled vertex v in $G - x$ with a clique K_v of size $m + 1$. Note that graph G^* is isomorphic to subgraph $G' - S$. Recall that A-labeled vertices maps to 0-vertices and B-labeled vertices map to 1-vertices. There is a homomorphism of G^* to H if we map all vertices in I_u to $f(u)$ and all vertices in C_v to $f(v)$, and we have a contradiction. Therefore, a minimal labeled obstruction to H has at most t vertices. □

Let H be a clean trigraph with exactly four vertices. Again, each vertex must either be a 0-vertex or a 1-vertex. Thus all clean trigraphs with four vertices belongs to one of the five families $\zeta(0,4)$, $\zeta(1,3)$, $\zeta(2,2)$, $\zeta(3,1)$, and $\zeta(4,0)$. Up to complementation, clean trigraphs with four vertices belong to one of the following families.

- $\zeta(4,0)$
- $\zeta(3,1)$
- $\zeta(2,2)$

Similar to the previous section, all trigraphs in the $\zeta(4,0)$ family have no $*$-edges, and can be resolved by Theorem 4.1.1. All trigraphs in the $\zeta(3,1)$ have no similar pair of 1-vertices and can be resolved by Theorem 4.2.5.

We now consider the $\zeta(2,2)$ family. Let H be a $\zeta(2,2)$ trigraph with two 0-vertices a, b, and two 1-vertices c, d. Trigraphs of the $\zeta(2,2)$ family can be further divided into the following subfamilies with respect to the pairs ab and cd.

1. The pairs ab and cd are 0-edges; or the pairs ab and cd are 1-edges.
2. The pair ab is a 1-edge and the pair cd a 0-edge.
3. The pair ab is a 0-edge and the pair cd a 1-edge.

Let H be a trigraph from the first subfamily. In trigraph H, either the pair ab is a 1-edge, or the pair cd is a 0-edge. In both cases, H has either no similar 0-vertices or no similar 1-vertices. Thus, by Theorems 4.2.5 and 2.3.5, trigraphs of the first subfamily have finitely many minimal labeled and unlabeled obstructions.

Let H be a trigraph from the second subfamily. It is easy to see that H has no similar 0-vertices and no similar 1-vertices, and we may apply Theorems 4.2.5 and 2.3.5, again to get finitely many minimal labeled and unlabeled obstructions.

It is obvious that all trigraphs from the third subfamily are split-friendly. Thus, by Theorem 4.1.2 they have finitely many minimal obstructions, and by Theorem 5.4.1 they also have finitely many minimal labeled obstructions.

5.5 Trigraphs with five vertices

Let H be a clean trigraph with exactly five vertices. Up to complementation, all clean trigraphs with five vertices belongs to one of the following three families.

1. $\zeta(5, 0)$
2. $\zeta(4, 1)$
3. $\zeta(3, 2)$

Similar to techniques used in the previous sections, we can show that trigraphs in the $\zeta(5, 0)$ and $\zeta(4, 1)$ family have finitely many minimal obstructions. Trigraphs from the $\zeta(3, 2)$ can be further divided into two subfamilies, namely those trigraphs that are nice those that are not nice. We may apply Theorems 4.2.5 and 2.3.5 to nice trigraphs and have finitely many minimal obstructions. Thus, we need only consider $\zeta(3, 2)$ trigraphs that are clean and non-nice. In other words, exceptional $\zeta(3, 2)$ trigraphs.

Let H be an exceptional $\zeta(3, 2)$ trigraph with three 0-vertices a, b, c and two 1-vertices d, e. By definition, H must have two similar 0-vertices and two similar 1-vertices. Since there are only two 1-vertices d, e, the pair de must be a 1-edge. Among the 0-vertices, there are exactly two possible ways in which there are similar 0-vertices. The first possibility is when all pairs ab, bc, ac are 0-edges. If this occurs, then we observe that H is split-friendly and we may apply Theorem 4.1.2 to show that H has finitely many minimal obstructions. The second possibility is when two of the pairs ab, bc, ac are 1-edges, and the other is a 0-edge. Without loss of generality, suppose ab, bc are 1-edges and ac is a 0-edge. This situation defines a general structure of what the remaining exceptional trigraph looks like as illustrated in Figure 5.7. Note that in this figure, the remaining six pairs of vertices

ad, ae, bd, be, cd, ce are not shown. These pairs may be any combination of 0-edges, 1-edges and $*$-edges. Since each pair has the choice of being any of the 3 kinds of edges, there are 3^6 exceptional trigraphs with five vertices. Even up to isomorphism, the number is high, so we only illustrate the general structure and not the precise trigraphs.

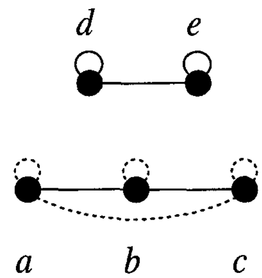


Figure 5.7: The general structure of an exceptional trigraph

We now examine all exceptional trigraphs with five vertices. The only similar vertices in H are between vertices a, c and between vertices d, e. Thus, there are three similarity classes in H. Let X, Y, Z be the similarity classes in H, with $a, c \in X$, $b \in Y$, and $d, e \in Z$ as illustrated in Figure 5.8.

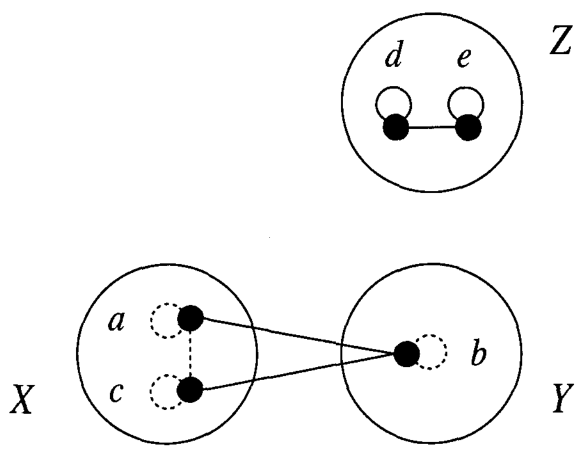


Figure 5.8: Similarity classes for the general structure of an exceptional trigraph

Let H be a clean trigraph and $\Re_H$ the set containing all proper induced subgraphs of H obtained from H by deleting one similarity class. Recall that l denotes the number of 0-vertices in H, and k denotes the number of 1-vertices in H. Recall again that N^A and N^B denote the number of similarity classes involving A-labeled and B-labeled vertices respectively, and N^0 and N^1 denote the number of similarity classes involving 0-vertices and 1-vertices respectively. Our main result in this chapter is formulated below. Note that clean trigraphs are hereditary. That is, every induced subgraph of a clean trigraph H is also clean.

Theorem 5.5.1 *Let H be an exceptional trigraph with five vertices. Suppose for each proper induced subgraph H' of H, all minimal labeled obstructions to H' have at most t vertices. Then every minimal labeled obstruction to H has at most*

$$max(N^0t,\ N^1t,\ N^0+1,\ N^1+1,\ 18,\ 2t)$$

vertices.

Proof. Our proof is broken into three cases. Each case will address a different property an arbitrary input graph G may have. In the first case we consider the case when $N^A < N^0$ or $N^B < N^1$, in the second case we consider when $N^A > N^0$ or $N^B > N^1$, and in the third case we consider when $N^A = N^0$ and $N^B = N^1$. For each case, we show a bound for a minimal labeled obstruction to H. Note that each trigraph H' in $\Re_H$ is a proper induced subgraph of H, so by assumption all minimal labeled obstructions to H' has at most t vertices.

<u>Case 1: Consider a labeled graph G with $N^A < N^0$, (analogously with $N^B < N^1$)</u>

Identical to case 1 from the proof of Theorem 4.2.5. Thus, a minimal labeled obstruction for this case has at most $max(N^0t, N^1t)$ vertices.

<u>Case 2: Consider a labeled graph G with $N^A > N^0$, (analogously with $N^B > N^1$)</u>

Identical to case 2 from the proof of Theorem 4.2.5. Thus, a minimal labeled obstruction for this case has at most $max(N^0+1, N^1+1)$ vertices.

<u>Case 3: We now assume a labeled graph G has $N^A = N^0$ and $N^B = N^1$</u>

Since the number of similarity classes in H is the same as the number of similarity classes in G, we can assign each similarity class of G to a similarity class of H. Recall, X and Y are similarity classes of 0-vertices in H and Z is the similarity class of 1-vertices in H. Let

U, V be the similarity classes of A-labeled vertices in G and let W be the similarity class of B-labeled vertices in G. Without loss of generality, let U, V, W be assigned to X, Y, Z respectively. We first find a bound on a minimal labeled obstruction to H for this assignment, and then compute the bound on a minimal labeled obstruction for any assignment.

We assume every vertex of U is adjacent to every vertex of V, else the minimal labeled obstruction G has 2 vertices; one vertex from similarity class U and one vertex from similarity class V.

Suppose pairs bd and be are both 0-edges (or analogously when they are both 1-edges). If some pair vw with $v \in V$ and $w \in W$ is an edge in G, then it is a minimal labeled obstruction to this assignment. If every pair vw with $v \in V$ and $w \in W$ is a non-edge then every vertex in V is non-adjacent to every vertex in W. In other words, any minimal labeled obstruction to H does not involve any vertices from the similarity class V. Let $H' = H - Y$. Recall that $H' \in \Re_H$. In this scenario, since we have assigned V to Y, a graph G admits a H-colouring if and only if $G - V$ admits an H'-colouring. Thus, by assumption, a minimal labeled obstruction in this subcase has at most t vertices.

Suppose one of the pairs bd, be is a 0-edge and the other is a 1-edge. Without loss of generality, let bd be a 0-edge and be a 1-edge. Any vertex $v \in V$ distinguishes vertices in W as being mapped to d or e. In other words, vertices in W are partitioned into two parts $W = W_d \cup W_e$ such that every vertex in V is non-adjacent to every vertex in W_d and adjacent to every vertex in W_e. If no such partition exists then there is a minimal labeled obstruction consisting of a vertex w and any two vertices $v, v' \in V$ with wv an edge and wv' a non-edge. We now assume that a partition $W = W_d \cup W_e$ exits. If vertex $u \in U$ cannot map to vertex a in H it is due to a some vertex w_1 in W and if u cannot map to vertex c in H it is due to some vertex w_2 in W. Thus, a minimal labeled obstruction for this case has at most 4 vertices, namely vertices v, u, w_1, and w_2.

Suppose pairs bd and be are both $*$-edges. Then there are no minimal labeled obstructions involving vertices in V. Let $H' = H - Y$. Recall again that $H' \in \Re_H$. As with the second subcase, a graph G admits an H-colouring if and only if $G - V$ admits an H'-colouring. Thus, a minimal labeled obstruction to this subcase has at most t vertices.

Suppose exactly one of bd, be is a $*$-edge. Without loss of generality, let bd be a $*$-edge and be a 1-edge. We know that the graph G consists of one clique W and a complete bipartite graph with independent sets U and V. Since the pair be is a 1-edge, any vertex w of W which is non-adjacent to a vertex of V must map to d. Let $H' = H - Y$. If such a vertex w does not exist, then G admits an H-colouring if and only if $G - V$ admits an H'-colouring, and we have a minimal labeled obstruction of size at most t vertices for this subcase. Thus suppose that such a vertex w does exist. Let W_d be the set of all such vertices w (non-adjacent to some vertex of V). If the pair ad (or similarly for cd) is a $*$-edge, then there are no labeled minimal obstructions to this subcase. Vertices in similarity class U are mapped to a (or similarity c), vertices in similarity class V are mapped to b, and vertices in similarity class W are mapped to d. Thus, we assume neither ad nor cd is a $*$-edge. In other words, we have one of the following scenarios.

- ad, cd are both 0-edges
- ad, cd are both 1-edges
- one of ad, cd is a 0-edge and the other is a 1-edge

Consider the first case. (We proceed analogously for the second case). Recall the set W_d from above. Let W_e be the set of all vertices of W adjacent to some vertex of U. Such vertices must map to vertex e. If $W_d \cap W_e \neq \emptyset$ then we obtain a minimal labeled obstruction with three vertices. Let w_1, w_2 be any two vertices in W_e adjacent to vertices u_1, u_2 in U respectively. If w_1, w_2 force a different mapping of vertex u_3 in U because of their connections to u_3, then we have a minimal labeled obstruction with five vertices. If neither of above occurs, then G is not a labeled obstruction to H.

Now consider the third scenario. Recall vertex w of W from above. It now separates U into two sets $U = U_a \cup U_c$ with vertices in U_a mapping to a and vertices in U_c mapping to c, otherwise we have a minimal labeled obstruction on four vertices. For each vertex $w' \neq w$ in W, there is a vertex from each of U_a, U_c, V preventing w' from mapping to d or e. Thus, a minimal labeled obstruction for this scenario prevents w' from mapping to either d or mapping to e and has at most 9 vertices.

We have now shown that a minimal labeled obstruction for this assignment has at most $max(9, t)$ vertices. Recall there is 1 similarity class of 1-vertices, $N^1 = 1$, and 2 similarity classes 0-vertices, $N^0 = 2$, thus there are only 2 possible assignments. For each assignment a labeled obstruction has at most $max(9, t)$ vertices, thus a minimal labeled obstruction for this case has at most $2max(9, t)$ vertices.

Every graph G falls under one of these cases, thus a minimal labeled obstruction to H has at most

$$max(N^0t,\ N^1t,\ N^0 + 1,\ N^1 + 1,\ 18,\ 2t)$$

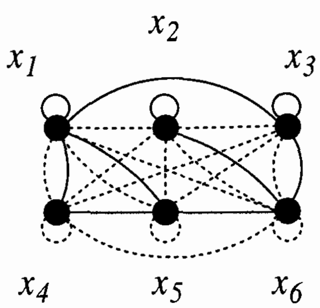
vertices. □

We have shown that all exceptional trigraphs with five vertices have finitely many minimal labeled obstructions, thus by Theorem 2.3.5 they have finitely many unlabeled obstructions. Although we are fortunate to show this for all exceptional trigraphs with five vertices, this fact is not true in general. We will show in the next chapter that there exists a clean, non-nice (i.e. exceptional) trigraph H with infinitely many minimal obstructions.

Chapter 6

An exceptional trigraph with IMMO

In the previous chapter we have seen a complete classification of trigraph homomorphisms (as having FMMO/IMMO) for trigraphs up to five vertices. It was shown that all clean trigraphs H with at most five vertices have infinitely many minimal obstructions. This classification, however, is not true for all trigraphs with more than five vertices. In this chapter, we will show that there is a clean trigraph H with six vertices, illustrated in Figure 6.1, which has infinitely many minimal obstructions. Note that H is an exceptional trigraph. Although exceptional trigraphs with five vertices have finitely many minimal obstructions, H does not.



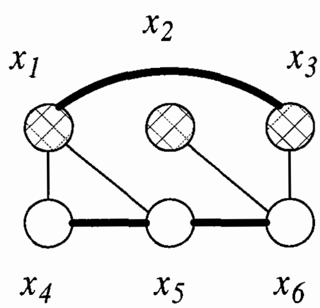


Figure 6.1: An exceptional trigraph and its corresponding partition

6.1 Clam graphs

We now define an infinite family of graphs that do not admit a homomorphism to H. Let $n \geq 5$. A graph G with $2n$ vertices $V(G) = v_1, \ldots, v_{2n}$ is a *clam graph* if all its edges are as follows.

- $v_a v_{a+1}$ for $a < 2n$
- $v_1 v_{2b+1}$ for $b < n$
- $v_{2c} v_{2d}$ for $c, d < n$ and $c \neq d$

In other words, G is a clam graph if $v_1, \ldots, v_{2n}$ is a path, vertex v_1 is adjacent to all odd numbered vertices greater than 1, all even numbered vertices less than $2n$ induce a clique, and there are no other edges. Figure 6.2 illustrates a clam graph on 10 vertices (i.e., with $n = 5$). In some instances, Figure 6.2 will be used to illustrate some of the processes for the proofs in this chapter. It will be easy to see how the proofs can be applied to larger clam graphs. Note that since clam graphs are defined for $n \geq 5$, the family of clam graphs is infinite.

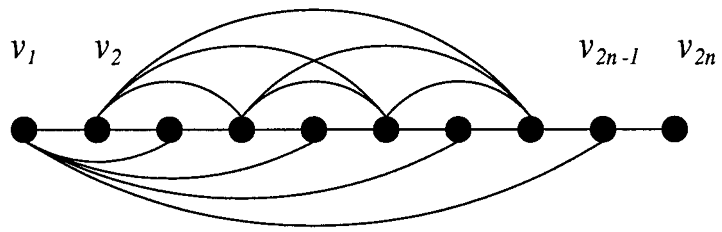


Figure 6.2: A clam graph

6.2 Every clam graph is an obstruction to H

We will first prove that every clam graph is an obstruction to H. In the next section we will prove that every clam graph is also a minimal obstruction to H.

Theorem 6.2.1 *Let G be a clam graph and H be the trigraph in Figure 6.1. Then G is an obstruction to H.*

Proof. Suppose there exists a homomorphism f of G to H. We will prove by contradiction that vertex v_1 cannot be mapped by f to any of the six vertices in H. The proof is broken into 6 cases. Each case will make the assumption that v_1 is mapped by f to one of the six vertices in H.

Case 1. Suppose v_1 is mapped to x_1 by f

There are three independent vertices v_3, v_5, v_7 adjacent to v_1. Since x_1 is a 1-vertex, at most one of these three vertices may be mapped to x_1 by f, thus two must be mapped to the same vertex x_4 (or x_5) by f. Without loss of generality, suppose v_5, v_7 are mapped to x_4 by f. The vertices v_4, v_6 are non-adjacent to v_1, thus they cannot be mapped to x_1. Since they are both adjacent to at least one of v_5 or v_7, they both cannot be mapped to x_2, x_3, x_4, x_6. Since they are adjacent to each other, they cannot be mapped to x_5 and we have a contradiction.

Case 2. Suppose v_1 is mapped to x_2 by f

The vertices v_1, v_2, v_3 induce a clique in G, thus at least one of v_2, v_3 is mapped to x_2 by f. Without loss of generality, let $f(v_2) = x_2$. Vertex v_5 is adjacent to v_1 but non-adjacent to v_2 (or v_3) and so v_5 must be mapped to x_6 by f. Now consider vertex v_4. It is adjacent to v_2 (or v_3) but not v_1 so it must be mapped to x_6. But v_4 is adjacent to v_5 and x_6 is a 0-vertex, so we have a contradiction.

Case 3. Suppose v_1 is mapped to x_3 by f

The vertices v_1, v_2, v_3 induce a clique in G, thus one of v_2 or v_3 is mapped to x_3 by f or both v_2 and v_3 are mapped to x_1 by f. We begin by addressing the first scenario. Without loss of generality, suppose $f(v_2) = x_3$. Vertex v_5 is adjacent to v_1 but non-adjacent to v_2 and so v_5 must be mapped to x_6 by f. Now consider vertex v_4. It is adjacent to v_2 but not v_1 so it must be mapped to x_6. But v_4 is adjacent to v_5 and x_6 is a 0-vertex, so we have a contradiction. Now in the second scenario, we assume both v_2 and v_3 are mapped to x_1 by f. Vertices v_5, v_7, v_9 are all adjacent to v_1 but non-adjacent to vertices v_2, v_3, thus they must be mapped to x_6. Consider vertex v_4. It is adjacent to vertices v_2, v_3, v_5 so it must map to x_5, but it is not adjacent to v_7, so we have a contradiction.

Case 4. Suppose v_1 is mapped to x_4 by f

The vertices v_1, v_2, v_3 induce a clique in G, thus at least one of v_2, v_3 must be mapped to x_1 by f. The vertices v_{2n-1}, v_{2n-3} are independent and adjacent to v_1. Since both are also non-adjacent to v_2, v_3, both must be mapped to x_5 by f. Now consider vertex v_{2n}. It is non-adjacent to v_2, v_3 so it cannot be mapped to either x_1 or x_3 by f. It is non-adjacent to v_{2n-3} so it cannot be mapped to either x_4 or x_6 by f. It is adjacent to v_{2n-1} so it cannot be mapped to either x_2 or x_5 by f and we have a contradiction.

Case 5. Suppose v_1 is mapped to x_5 by f

The vertices v_1, v_2, v_3 induce a clique in G, thus at least one of v_2, v_3 is mapped to x_1 by f. More precisely, v_2, v_3 may be mapped by f in exactly three different ways:

- $f(v_2) = f(v_3) = x_1$
- $f(v_2) = x_1,\ f(v_3) = x_4$
- $f(v_2) = x_4,\ f(v_3) = x_1$

Suppose we have the first scenario. Vertex v_6 is adjacent to v_2 and non-adjacent to v_1, v_3 so it must be mapped to x_5 by f. Vertex v_9 is adjacent to v_1 but not v_6, so it must be mapped to x_1. But x_1 is a 1-vertex and v_9 is non-adjacent to vertices v_2, v_3 and we have a contradiction.

Suppose we have the second scenario. Vertex v_9 is adjacent to v_1, v_{10}, and vertex v_{10} is only adjacent to v_9. Thus, v_9 must be mapped to x_6 and v_{10} must be mapped to x_2. Vertex v_8 is adjacent to v_2, v_9, but not v_1, so it must map to x_3. Vertex v_7 is adjacent to v_1, v_8, but not adjacent to v_2, so it must map to x_6. By the same logic, we force v_6 to x_3 and v_5 to x_6. Now v_4 must map to x_5 since it is adjacent to both v_3, v_5, but is it also adjacent to x_6, x_8 and the pair x_3x_5 is a 0-edge, so we have a contradiction.

Suppose we have the third scenario. We arrive at the same contradiction if we apply the same proof as in the second scenario.

Case 6. Suppose v_1 is mapped to x_6 by f

The vertices v_1, v_2, v_3 induced a clique in G, thus either both v_2, v_3 are mapped to x_2 or they are both mapped to x_3 by f. Vertices v_6, v_8 are adjacent to v_2 but not v_3, so they must map to x_6 by f. But x_6 is a 0-vertex and v_6 is adjacent to v_8, so both v_6, v_8 cannot map to x_6 and we have a contradiction.

We have now shown that v_1 cannot be mapped to any vertex in H. Thus, there is no homomorphism from G to H and G is an obstruction to H. □

6.3 Every clam graph is a minimal obstruction to H

Now that we have established that every clam graph G is an obstruction, we will prove that every clam graph G is also a minimal obstruction.

Theorem 6.3.1 *Let G a clam graph and H be the trigraph in Figure 6.1. Then G is a minimal obstruction to H.*

Proof. By Theorem 6.2.1, we have shown G to be a obstruction to H. We will now show that G is a minimal obstruction to H. Let v_j be any vertex in G. In particular, we will show that the graph $G - v_j$ admits an H-colouring f. Our proof is broken into 3 cases. For each case, we will define a homomorphism f from G to H.

Case 1. suppose $v_j = v_1$

Let $i \leq n-1$. We define $f(v_{2i}) = x_3$, $f(v_{2i+1}) = x_6$ and $f(v_{2n}) = x_2$. Vertices mapped to x_3 induce a clique in G, vertices mapped to x_6 are independent in G, and the vertex mapped to x_2 is non-adjacent to any vertices mapped to x_3.

Case 2. suppose $v_j = v_{2n}$

Let $i \leq n-1$. We define $f(v_1) = x_5$, $f(v_{2i}) = x_1$ and $f(v_{2i+1}) = x_4$. Vertices mapped to x_1 induce a clique in G, vertices mapped to x_4 are independent in G, and the vertex mapped to x_5 is adjacent to all vertices mapped to x_4.

Case 3. suppose $v_j \neq 1, v_{2n}$

Let $1 \leq a \leq \lfloor j/2 \rfloor$ and $\lceil j/2 \rceil \leq b \leq n-1$. We define

- $f(v_1) = x_5$
- $f(v_{2n}) = x_2$.
- $f(v_{2a}) = x_4$ for $2a \neq j$
- $f(v_{2a-1}) = x_1$ for $2a - 1 \neq 1$

- $f(v_{2a}) = x_3$ for $2a \neq j$
- $f(v_{2a+1}) = x_6$ for $2a + 1 \neq n$

In other words, we define f such that vertex v_1 maps to x_5, vertex v_n maps to x_2, all even labeled vertices less than j map to x_1, all odd labeled vertices less than j map to x_4, all even labeled vertices greater than j map to x_3 and all odd labeled vertices greater than j map to x_6.

It is easy to confirm that all vertices mapped to x_1, x_3 induce a clique in $G - v$, all vertices mapped to x_3, x_6 are independent in $G - v$ and are all adjacent to v_1, all vertices mapped to x_1 are not adjacent to any vertex mapped to x_6, all vertices mapped to x_3 are not adjacent to any vertex mapped to x_4, and v_n is only adjacent to vertices mapped to x_6.

We have shown a homomorphism of $G - v$ to H, and we know by Theorem 6.2.1 that G is an obstruction, therefore G is a minimal obstruction to H. □

We have shown that every clam graph is a minimal obstruction to H. Since the family of clam graphs is infinite, there are infinitely many minimal obstructions to H.

Chapter 7

Conclusion

In Chapter 2, we reviewed some general tools, such as retractions and 2-SAT, that have been used in solving partition problems. These, along with other techniques, were used in the classification of almost all 'small' list trigraph homomorphism problems as NP-complete or polynomial time solvable [7, 18]. In particular, all trigraphs having up to four vertices, with the exception of the trigraph corresponding to the 'stubborn problem' (and its complement) has been classified.

In our original contribution, we focused our attention on small trigraphs and obstructions. In particular, we investigated all trigraphs having up to five vertices and ask whether they have finitely many minimal obstructions (FMMO) or infinitely many minimal obstructions (IMMO). Recall that a trigraph having FMMO is automatically polynomial time solvable. We developed tools in Chapters 3 and 4 to aid in proving that up to five vertices, there is an easy classification of these trigraphs. Namely, those trigraphs that are messy have IMMO, and those trigraphs that are clean have FMMO. However, this simple dichotomy of trigraphs does not apply in general. In particular, we gave an example of a clean trigraph on six vertices in Chapter 6, and proved that it has IMMO.

From the tools that were developed, we are also able to define two infinite families of trigraphs that have FMMO. The *split-friendly family* is defined to contain precisely those trigraphs with only 0-edges in S_A and only 1-edges in S_B. Trigraphs in this family are all split-friendly, and thus have FMMO. A trigraph H is *very nice* if it has only 1-edges in S_A or only 0-edges in S_B. Very nice trigraphs have the hereditary property of being nice, and thus they also have FMMO. The trigraphs in Figure 4.4 are in fact very nice.

7.1 Future work

It would be nice to give a necessary and sufficient condition for when a trigraph H yields a trigraph homomorphism problem with FMMO. Our distinction of clean versus messy trigraphs was based on the observation that the structures $B, \overline{B}$ were the only causes for IMMO for the trigraph homomorphism problems for small trigraphs H. In Chapter 6, we found a second structure H (and its complement $\overline{H}$) that yielded IMMO for the trigraph homomorphism problem. It may be possible to show that structures H, $\overline{H}$ and perhaps finitely many others structures are the only minimal trigraphs that cause IMMO.

Of those trigraphs classified as having IMMO, it would be interesting to identify whether the basic trigraph homomorphism problem is NP-complete or polynomial time solvable. It is clear that any messy trigraph with an induced subtrigraph corresponding to the 3-colouring problem is NP-complete. However, little is known about other messy trigraphs and in general, other trigraphs with IMMO.

As we have extended the results from [7, 18] by considering the IMMO/FMMO view for undirected trigraphs, it is also interesting to consider this same IMMO/FMMO view for directed trigraphs. A complete classification of NP-complete or polynomial time solvable was achieved for all directed trigraphs with up to three vertices in [21], however the classification of these directed trigraphs in terms of their obstructions has yet to be investigated.

Bibliography

[1] B. Aspvall, F. Plass and R.E. Tarjan, A linear time algorithm for testing the truth of certain quantified Boolean formulas, Information Processing Letters 8 (1979) 121-123.

[2] A. Brandstadt, Partitions of graphs into one or two stable sets and cliques, Discrete Mathematics 152 (1996) 47-54.

[3] C. Berge and C. Chvatal, Topics on Perfect Graphs, Annals of Discrete Math. 21, 1984.

[4] C. Berge, Farbung von Graphen deren samtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung), Wiss. Zeit. Der Martin-Luther Universitat Halle-Wittenberg, Math. Natur. Reihe 10 (1961) 114.

[5] A. Brandstadt, V.B. Le, T. Szymczak, The complexity of some problems related to graph 3-colourability, Discrete Applied Mathematics 89 (1998) 59-73.

[6] R.N. Ball, J. Nesetril, and A. Pulr, Dualities in full homomorphisms, manuscript 2006.

[7] K. Cameron, E. Eschen, C. Hoang, and R. Sritharan, The list partition problems for graphs, SODA (2004) 391-399.

[8] G. Chartrand, Graphs as mathematical models, Prindle, Weber and Schmidt Incorporated, Boston (1977).

[9] D.G. Corneil, Y. Perl, and L.K. Stewart, A linear recognition algorithm for cographs, SIAM J. on Computing. 14 (1985) 926-934.

[10] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, The strong perfect graph theorem, Annals of Mathematics 164, (2006), 51-229.

[11] V.Chvatal and N. Sbihi, Bull-free Berge graphs are perfect, Graphs and Combinatorics 3 (1987) 127-139.

[12] P. Erdos, Graph theory and probability, Canad. J. Math 11 (1959) 34-38.

[13] H. Everett, S. Klein, and B. Reed, An algorithm for finding clique-cross partitions, Congressus Numerantium 135 (1998) 171-177.

[14] T. Feder and P. Hell, On realizations of point determining graphs and obstructions to full homomorphisms, manuscript 2004.

[15] T. Feder and P. Hell, Matrix partitions of perfect graphs, Discrete Math., in press.

[16] T. Feder and P. Hell, Full constraint satisfaction problems, SIAM J. on Computing, in press.

[17] T. Feder, P. Hell, and Hochstattler, Generalized colourings (matrix partitions) of cographs, Graph Theory, (J. Ramirez ed.) Volume in Memory of Claude Berge, Trend in Mathematics, Birkhauser Verlag, Basel (2006) 149-167.

[18] T. Feder, P. Hell, S. Klein, and R. Motwani, Complexity of graph partition problems, Proc. 31st Annual ACM Symposium on Theory of Computing, Atlanta GA, May 1999, pp. 464-472.

[19] T. Feder, P. Hell, S. Klein, L. Tito Nogueira, and F. Protti, List matrix partitions of chordal graphs, Theoretical Computer Science 349 (2005) 52–66.

[20] T. Feder, P. Hell, S. Klein, L. Tito Nogueira, and F. Protti, Partitioning chordal graphs into independent sets and cliques, Discrete Applied Mathematics 141 (2004) 185–194.

[21] T. Feder, P. Hell, and K. Tucker-Nally, Digraph matrix partitions and trigraph homomorphisms, Discrete Applied Mathematics 154 (2006) 2458-2469.

[22] T. Feder, and M. Vardi, The computational structure of monotone monadic SNP and constraint statisfaction: a study throught Datalog and group theory, SIAM J. Computing 28 (1998) 236–250.

[23] C.M.H. de Figueiredo and S. Klein, The NP-completeness of multipartitie cutset testing, Congressus Numerantium 119 (1996) 217-222.

[24] C.M.H. de Figueiredo, S. Klein, Y. Kohayakawa and B. Reed, Finding skew partitions efficiently, J. Algorithms 37 (2000) 505-521.

[25] S. Foldes, P. Hammer, Split graphs, Congr. Numer. 19 (1977) 311-315.

[26] L.R. Foulds, Graph theory applications, Springer-Verlag, New York (1992).

[27] M.R. Garey and D.S. Johnson, Computers and intractability, W.H. Freeman and Company, San Francisco, 1979.

[28] M.C. Golumbic, Algorithmic graph theory and perfect graphs, Academic Press, New York, 1980.

[29] J.L Gross and J. Yellen, Graph theory and its applications, CRC Press, 1998.

[30] M. Grotschel, L. Lovasz and A. Schrijver, Polynomial algorithms for perfect graphs, Ann. Discrete Math. 21 (1984) 325-356.

[31] W.Gutjahr, E. Welzl and G. Woeginger, Polynomial graph-colorings, Discrete Applied Mathematics 35 (1992) 29-45.

[32] P.Hell and J. Nesetril, On the complexity of H-colouring, Journal of Combinatorial Theory, Series B 48 (1990) 92-110.

[33] P.Hell and J. Nesetril, Graphs and Homomorphisms, Oxford University Press, Oxford, 2004.

[34] C.T. Hoang and V.B. Le, Recognizing perfect 2-split graphs, SIAM J. on Discrete Math. 13 (2000) 48-55.

[35] Z. Hedrlin and A. Pultr, Relations (graphs) with given finitely generated semigroups, Monatsh. Math. 68 (1964) 213-217.

[36] D. Konig, Theorie der endlichen und unendlichen Graphen, Akademische Verlagsgesellschaft (1936) (reprinted Chelsea 1950).

[37] T. Kloks, D. Kratsch, and H. Muller, Finding and counting small induced subgraphs efficiently, Information Processing Letters 74 (2000) 115-121.

[38] L. Lovasz, Normal hypergraphs and the perfect graph conjecture, Discrete Math. 2 (1972) 253-267.

[39] G. MacGillivray, and M.L. Yu, Generalized parttions of graphs, Discrete Applied Mathematics 91 (1999) 143-153.

[40] J. Nestril and S. Poljak, On the complexity of the subgraph problem Commentationes Mathematicae Universitatis Carolinae, 14 (1985) no.2 415-419

[41] G. Sabidussi, Graph derivatives, Math. Z. 76 (1961) 385-401.

[42] R.E. Tarjan, Decomposition by clique separators, Discrete Mathematics 55 (1985) 221-232.

[43] A. Tucker, Coloring graphs with stable sets, Journal of Combinatorial Theory, series B 34 (1983) 258-267.

[44] N. Vikas, Computational complexity of graph compaction, Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, SODA 1999.

[45] D.B. West, Introduction to graph theory, Prentice Hall, New Jersey (2001).

[46] S. Whitesides, An algorithm for finding clique cutsets, Information Processing Letters 12 (1981) 31-32.

[47] S. Whitesides, A method for solving certain graph recognition and optimization problems, with applications to perfect graphs, Annals of Discrete Mathematics 21 (1984) 281-297.