Edge Colourings, Strong Edge Colourings, and Matchings in Graphs

Master's Thesis

Graph Theory

May 26th 2017

Frederik Vestergaard Kjemtrup

Department of Mathematical Sciences

Fredrik Bajers Vej 7G • 9220 Aalborg Ø • Tlf. 99409940

Danish abstract

Dette kandidatprojekt i matematik beskæftiger sig med grafteori, specifikt teorien vedrørende kantfarvninger, stærke kantfarvninger og parringer samt faktoriseringer af grafer.

Kantfarvning af en graf består i at tildele hver kant en farve, sådan at intet par af tilstødende kanter tildeles samme farve. Selvfølgelig kan dette opnås ved at tildele en særskilt farve til hver kant i en given graf, men ofte kan man have interesse i at benytte så få farver som muligt.

Efter en kort introduktion til kantfarvningsterminologi, indeholder Kapitel 2 et bevis for Vizings sætning, som angiver en generel øvre grænse for grafers kromatiske indeks, som er det mindste antal farver, der behøves for at opnå en gyldig farvning af alle grafens kanter.

Herefter betragtes i Kapitel 3 kantfarvninger fra et andet perspektiv, nemlig parringer. Parringer vises at være grundlæggende forbundet med faktorer i grafer, hvorefter flere resultater vedrørende parringer og faktorer bevises.

Dernæst skærpes begrebet kantfarvninger i Kapitel 4 til det specialtilfælde, som stærke kantfarvninger udgør. En kantfarvning af en graf kaldes stærk, hvis der om enhver kant i grafen gælder, at alle dennes tilstødende kanter er tildelt særskilte farver.

Det bevises, at en velkendt formodning om en øvre grænse for det stærke kromatiske indeks vil være skarp, såfremt formodningen viser sig at være sand. Denne formodning går i grove træk ud på, at en graf med højeste grad Δ højst kan have stærkt kromatisk indeks $\frac{5}{4}\Delta^2$.

Dernæst bekræftes et særtilfælde af formodningen som kun betragter særlige todelte grafer. Resultatet lyder, at for en todelt graf, hvori den ene delgraf har højeste grad 2, vil det stærke kromatiske indeks højst være lig det dobbelte af grafens højeste grad.

Endelig indeholder Kapitel 5 et bevis for en sætning, som for kubiske grafer bekræfter den omtalte formodning om en øvre grænse for det stærke kromatiske indeks.

AALBORG UNIVERSITY
STUDENT REPORT

Department of Mathematical Sciences
Fredrik Bajers Vej 7G
Telephone 99409940
http://math.aau.dk

Title:

Edge Colourings, Strong Edge Colourings, and Matchings in Graphs

Project period:

Spring semester 2017
10'th semester

Project group:

G2-106a

Group members:

Frederik Vestergaard Kjemtrup

Supervisor:

Leif Kjær Jørgensen

Number of copies: 2

Page number: 54 (57 with appendices)

Finished: May 26$^{\text{th}}$ 2017

Synopsis:

This master's thesis deals with graph theory, specifically the theory of edge colourings, strong edge colourings, matchings, and factorisations of graphs.
After a brief introduction to edge-colouring terminology, Chapter 2 contains a proof of Vizing's theorem, bounding the chromatic index of graphs in a general way.
Chapter 3 considers edge colouring graphs from a different perspective, that of matchings. Matchings are shown to be inherently linked to factors of graphs, whereupon several results on both matchings and factors are proven throughout the chapter.
In Chapter 4, a special case of edge colouring is considered, namely that of strong edge colourings. It is shown that a well-known conjecture about an upper bound on the strong chromatic index would be sharp if the conjecture were true. Then, a special case of the conjecture is proven for bipartite graphs.
Finally, Chapter 5 contains a proof of a theorem which confirms the conjectured upper bound on the strong chromatic index, however only in the special case of cubic graphs.

Contents

Chapter 1
Introduction

Throughout the thesis, we consider simple, finite graphs unless otherwise specified. We will occasionally need to consider directed graphs or multigraphs to gain a better understanding of a given theory or to establish a result. When this occurs, it will be clearly noted in the preceding text.

A thorough introduction to graph theory would be of great aid to the reader before reading this thesis, and for this purpose the book [Gross et al., 2006] must be recommended for its intuitive composition and clear language.

Likewise, the thesis makes liberal use of basic definitions, terminology, and notation as it is introduced throughout the book [Gross et al., 2006]. Should another perspective on the subject matter be desired, [Chartrand et al., 2009] provides a different, yet equally complete treatment of chromatic graph theory.

Edge colourings CHAPTER 2

The most famous theorem of graph theory is, undoubtedly, the Four-Colour Theorem, one reason for its fame being its reasonably simple statement, and another reason being the remarkable span of time from its first statement to its confirmative proof published in [Appel et al., 1989]. The Four-Colour Theorem, in its conjectured form, asked whether it was possible to colour the vertices of a simple, planar graph with four colours, such that no two vertices of the same colour were connected by an edge. A notion closely related to this question of vertex colouring is the notion of *edge* colouring a graph. There is no edge-colouring theorem equivalent to the Four-Colour Theorem, but edge colouring theory is nonetheless a rich theoretical field.

2.1 Basic edge colouring

Before we can prove any statements about edge-colouring graphs, we need to formalise what it means to colour edges.

2.1.1 Definition ((Proper) Edge colouring):
Let G be a graph. An edge-colouring of the graph is a map $c : E(G) \to S$, which maps each edge to an element of the finite set S. We refer to these elements as colours *and say that an edge e is* assigned *the colour $c(e)$ in S. It is tacitly assumed that $S = \{1, 2, \ldots, k\}$ for some finite integer k. This has the effect of imposing a natural order on the elements of S. An edge colouring is called proper if no two edges joined by a single vertex are assigned the same colour.*

2.1.2 Definition (Edge colour class):
An edge colour class is defined for an edge-colouring as the set of all the edges in $E(G)$ that are mapped to the same colour in S.

The above definition implies that, for a given edge-colouring $c : E(G) \to S$, at most $|S|$ edge colour classes can exist.

Two edges are said to be *adjacent* if they have a common end vertex. Similarly, two vertices in a graph, say v and w, are said to be adjacent if there exists an edge in the graph with v and w as its end vertices.

2.1.3 Definition (Edge-k-colourable):
A graph G is said to be edge k-colourable if it has a proper edge k-colouring.

If a graph has a proper edge k-colouring, it is said to be properly edge k-colourable.

2.1.4 Definition (Chromatic index):
For a graph G, the chromatic index $\chi'(G)$ is defined as the smallest integer k for which G is properly edge k-colourable. The graph G is then said to be edge k-chromatic.

Perhaps the most immediate question arising from these definitions, is how one might find a proper edge-colouring of a given graph. One method is contained in the following algorithm.

Algorithm 1 Greedy edge-colouring

Input: A graph G of size m with edges listed as $e_1, e_2, \ldots, e_m$.
Output: A proper edge-colouring $c : E(G) \to S$, where $S = \{1, 2, \ldots, k\}$

for $i = 1$ **to** m **do**
    Assign to e_i the smallest colour not assigned to any edge adjacent to e_i
return The edge-colouring c determined by the above assignments.

2.2 Chromatic index of bipartite graphs

A graph G is said to be bipartite if its vertex set can be partitioned into two subsets $U \subseteq V(G)$ and $W \subseteq V(G)$, such that no two vertices of either subset are adjacent. That is, the only edges in G are edges between some vertex in U and some other vertex in W. A graph is bipartite, if and only if it does not contain any cycles of odd length. A proof of this may be found in [Chartrand et al., 2009; Theorem 1.10, p. 40]. Bipartite graphs find applications in the modelling of many practical problems in the areas of time scheduling and job assignments. The chromatic index of such graphs are the focus of this section. The section is inspired by [Gross et al., 2006; Section 9.3]

Given a graph G with edge colouring c, we say that $c(e)$ is an *incident edge-colour* of v if v is an endpoint of e. If v is not an endpoint of e, the colour $c(e)$ is said to be an *absent edge-colour* of v.

2.2.1 Definition (Chromatic incidence):
The chromatic incidence of a vertex v in a graph G is defined as the number of distinct colours assigned by a given edge colouring c to the edges adjacent to v. It is denoted by $eci_v(c)$ to represent the edge chromatic incidence at v with respect to the colouring c.

2.2.2 Definition (Total chromatic incidence):
The total chromatic incidence of an edge colouring c of a given graph G is defined by

$$eci(c) = \sum_{v \in V(G)} eci_v(c).$$

It is easily observed, that $eci_v(c) \leq \deg(v)$, since at most $\deg(v)$ colours can be assigned to the edges adjacent to v. The following result is similarly obvious.

2.2.3 Proposition:
The graph G is properly edge coloured by the map $c : E(G) \rightarrow S$, if and only if $eci_v(c) = \deg(v)$ for every $v \in V(G)$.

Proposition 2.2.3 readily extends to the following.

2.2.4 Proposition:
The graph G is properly edge coloured by the map $c : E(G) \rightarrow S$, if and only if

$$\sum_{v \in V(G)} eci_v(c) = \sum_{v \in V(G)} \deg(v).$$

The above propositions double as lemmata for a theorem which we will later state. To state and prove the following lemma, we first recall some definitions.

2.2.5 Definition (Eulerian circuit and eulerian graph):
An eulerian circuit of a graph G is a circuit that contains every edge of G exactly once. A graph is said to be eulerian if every component of it contains an eulerian circuit.

The statement of the lemma and its subsequent proof follows the strategy contained in [Gross et al., 2006; Chapter 9.4].

2.2.6 Lemma:
Let G be a connected graph that is not an odd cycle, and let G contain at least two edges. Then G permits an edge 2-coloring such that every vertex of degree at least 2 has both colours as incident edge-colours.

Proof:
The proof is handled by distinct cases.
Case 1:
If G is a cycle, it must be even by assumption. Assigning the colour 1 to an arbitrarily chosen edge forces the choice of colour 2 to be assigned to the two edges adjacent to the first one. The continued use of this argument creates a cycle of two alternating colours in which every vertex of degree at least 2 has two incident edge-colours.
Case 2:
Let G be an eulerian graph that is not a cycle. Then it must contain an eulerian circuit that contains a vertex of degree at least 4, since every vertex of an eulerian graph has even degree. For a proof of this rather simple fact, see [Chartrand et al., 2009; Theorem 3.1, p. 73]. Construct an edge 2-colouring of this circuit by assigning the colour 1 to the edges that occur as odd terms and the colour 2 to the edges that occur as even terms in the edge sequence describing the circuit. By this construction, the colours 1 and 2 are both incident edge-colours to every vertex in the eulerian circuit of degree at least 2. Since some vertex

had degree at least 4, then it occurs on the cycle as an internal vertex, implying that it will also have both 1 and 2 as incident edge-colours.

Case 3:
Assume that G is not eulerian. Construct a new graph $\tilde{G}$ by adding a vertex v to $V(G)$ and joining v to every vertex in $V(G)$ of odd degree by an edge. Since every edge of a graph contributes to the degree of both endpoints, it is easy to see that $\sum_{v\in V(G)} \deg(v) = 2\,|E(G)|$. It follows from this that every graph contains an even number of vertices of odd degree. This implies that the degree of v is even, which again implies that $\tilde{G}$ is eulerian. Since $\tilde{G}$ is eulerian, Case 2 above applies, and we need only argue that the restriction of the edge colouring c to the original graph G is still an edge 2-colouring of G that meet the requirements. This is true, since the removal of v does not restrict the colours incident to the neighbours of v. Hence, the colours incident to any vertex adjacent to v can have its incident colours assigned in such a way as to preserve the required quality. ■

2.2.7 Definition (Kempe $i-j$ edge-chain):
A Kempe $i-j$ edge-chain of a graph G that has been edge coloured by c is a component of the subgraph of G induced by all edges coloured by either i or j.

Note that an edge coloured graph can contain several Kempe $i-j$ edge-chains.

2.2.8 Lemma:
Let c be an edge k-colouring of a graph G, such that c has the largest possible edge chromatic incidence $eci(c)$. Further, let v be a vertex in $V(G)$ to whom i is an incident edge-colour at least twice, and to which j is an absent edge-colour. Finally, denote by K the Kempe $i-j$ edge-chain containing v. Then, K is an odd cycle.

Proof:
Lemma 2.2.6 states that, if K is not an odd cycle, the colours i and j could be reassigned to edges in the component K in such a way that the chromatic incidence of every vertex in K would be at least 2. Such an altered edge-colouring $\tilde{c}$ would increase the chromatic incidence of v from 1 to 2, since i contributes to $eci_v(c) = 1$, while both i and j contribute to the sum $eci_v(\tilde{c}) = 2$. Additionally, every vertex in K apart from v would have at least the same chromatic incidence with respect to $\tilde{c}$ as it would with respect to c. This would contradict the assumption that c was an edge-colouring with maximal total chromatic incidence $eci(c)$, implying that K must be an odd cycle. ■

We may now state and prove the main theorem of this section, first published by Denes König in 1916, [König, 1916].

2.2.9 Theorem:
Let G be a bipartite graph. Then $\chi'(G) = \Delta(G)$.

Proof:
Assume for contradiction that $\chi'(G) \neq \Delta(G)$. The inequality $\chi'(G) \geq \Delta(G)$ holds since, for any vertex $v \in V(G)$, the chromatic incidence of v with respect to a proper edge colouring, $eci_v(c)$, must be at least $\deg(v)$. This implies that $\chi'(G) > \Delta(G)$.

Now, let c be an edge $\Delta(G)$-colouring of G with maximal total chromatic incidence $eci(c)$. The edge colouring c is not proper, so by Proposition 2.2.3 there must be at least one vertex $v \in V(G)$ for which $eci_v(c) < \deg(v)$. The pigeonhole principle implies that at least one of the $\Delta(G)$ available colours must be incident on v at least twice, leaving the colours of at most $\Delta(G) - 2$ edges unaccounted for. However, there are $\Delta(G) - 1$ colours left with which to colour the at most $\Delta(G) - 2$ remaining edges. Lemma 2.2.8 now implies that G contains an odd cycle, contradicting the assumption that G is bipartite. Hence, a graph G cannot be both bipartite and obey $\chi'(G) \neq \Delta(G)$. ■

2.3 Vizing's theorem

While a lower bound on the chromatic index is found in $\Delta(G) \leq \chi'(G)$, we may prove a surprisingly sharp general upper bound on the chromatic index, first proven by Vadim G. Vizing in 1964.

2.3.1 Lemma:
Let i and j be two colours used to properly edge colour a graph G. Then every Kempe $i - j$ edge-chain in G is a path.

Proof:
Any vertex in a Kempe $i - j$ edge-chain in a properly edge 2-coloured graph has degree at most 2. By definition, a Kempe $i - j$ edge-chain is a connected subgraph, so it must be a path. ■

Note that the path described by a Kempe $i - j$ edge-chain is not required to be open. It may be a cycle.

2.3.2 Theorem (Vizing's Theorem):
Let G be a graph. Then there exists a proper edge $\Delta(G) + 1$-colouring of G.

The proof follows the strategy of [Greene, 2013].

Proof:
The theorem is proven by induction on $|E(G)|$. If G is a trivial graph, then the assertion that G can be properly edge $\Delta(G) + 1$-coloured is also trivial.

Thus, suppose that $|E(G)| > 0$ and that the theorem holds for any graph with fewer than $|E(G)|$ edges. Choose an edge $e \in E(G)$ with endpoints v and v_0. By assumption, $G - e$ can be properly edge $\Delta(G)$-coloured, and since $\deg(v) \leq \Delta(G)$, at least one colour, say c_0, is not incident to e at v. From this, it is seen that the following conditions are met for $k = 0$:

Statement $P(k)$**:**

(i) v has k distinct neighbours labelled $v_0, v_1, \ldots, v_k$.

(ii) The graph $G - vv_k$ admits a proper edge $\Delta(G)$-colouring with colours $S = \{c_0, \ldots, c_{\Delta(G)}\}$.

(iii) The colour c_0 is absent to e at v.

(iv) For $j = 1, \ldots, k$, the colour c_j must appear on the edge vv_{j-1}.

(v) For $j = 1, \ldots, k$, the Kempe $c_0 - c_j$ edge-chain on which v appears is a path with endpoints v and v_j.

The latter two of these conditions are vacuously met for $k = 0$. Since $P(k)$ implies that $k+1 \leq \deg(v)$, which is again less than $\Delta(G)$, the theorem will follow by showing that $P(k)$ implies that either G can be properly edge $\Delta(G)+1$-coloured, or $P(k+1)$ is true. Indeed, if no k is encountered for which G can be properly edge $\Delta(G)$-coloured, then G can be properly $\Delta(G)+1$-coloured, since (ii) implies the existence of a proper $\Delta(G)$-colouring.

Assume that $P(k)$ holds. Then, since $\deg(v_k) \leq \Delta(G)$ and no colour has been assigned to the edge vv_k, some colour $c_{k+1} \neq c_k$ must be absent at v_k. Since this colour still belongs to the set $S = \{c_0, \ldots, c_{\Delta(G)}\}$, so consider the Kempe $c_0 - c_k$ edge-chain that has v_k as an endpoint. We denote this chain by K.

If K does not have v as an endpoint, the colours along it may be interchanged to produce a proper edge $\Delta(G)+1$-colouring of the graph $G - vv_k$, in which c_0 is absent to e at both v and v_k. Assigning c_0 to the edge e with endpoints v and v_k now yields the desired colouring.

If K does, in fact, reach the vertex v, then it arrives at this vertex along some edge vv_{k+1} to which the colour c_{k+1} has been assigned. This colour c_{k+1} cannot be the same as c_j for all values assumed by $j = 1, \ldots, k$, because if it were, then K would be the Kempe $c_0 - c_j$ edge-chain with endpoints v and v_j. From this it would follow that $v_j = v_k$, implying that $c_{k+1} = c_j = c_k$, a contradiction in the choice of c_{k+1}. Since $c_{k+1} \neq c_j$ for all values of $j = 1, \ldots, k$, it follows that $v_{k+1} \neq v_j$ for $j = 1, \ldots, k$. Since the colour c_{k+1} remains absent at v_k, the edge vv_{k+1} can have its colour, c_{k+1}, reassigned to the edge vv_k. This implies that $P(k+1)$ holds, and the theorem follows. ■

Clearly, for any graph G we have either $\chi'(G) = \Delta(G)$ or $\chi'(G) = \Delta(G) + 1$.

CHAPTER 3
Matching and factorisation

To understand a mathematical theory well, one may often benefit from the study of theories related to it. Matchings in graphs constitutes a concept closely related to that of edge colourings.

3.1 Matchings in graphs

Edge colouring graphs can be seen as the description of an edge colouring map $c : E(G) \to S$, but may also be viewed as a partitioning of the edges of a graph into independent subsets. That is, disjoint subsets of the edge set which contain all edges to which some single colour is assigned. Under this perspective, the study of matchings are inherently connected to the study of edge colourings, and deserves thorough mention.

3.1.1 Definition (Matching):
A matching in a graph G is a subset M of $E(G)$ such that no pair of edges $x, y \in M$ are adjacent.

3.1.2 Definition (Maximal matching):
A matching M in a graph G is called maximal if it is not a proper subset of any other matching in G.

3.1.3 Definition (Maximum matching):
A maximum matching in G is a matching M whose cardinality is maximal.

Clearly, a maximum matching is a maximal matching. A maximal matching need not be a maximum matching, however.

One connection between matchings and proper edge colourings is displayed in the following algorithm.

Algorithm 2 Proper edge-colouring using maximum matchings

Input: A graph G

Output: A proper edge-colouring $c : E(G) \to S$, where $S = \{1, 2, \ldots, k\}$

```
1: k = 0
2: while E(G) ≠ ∅ do
3:     k = k + 1
4:     Find a maximum matching M in G
5:     for e ∈ M do
6:         Assign to e the colour k
7:     G = G − M
8: return The edge-colouring c determined by the above assignments.
```

Notice that the algorithm makes no comment on how to find the required maximum matchings, nor is the colouring guaranteed to use a minimal number of colours. It remains, however, a proper edge colouring.

The final special case of matchings that we shall consider are the so-called perfect matchings.

3.1.4 Definition (Perfect matching):
A matching M in a graph G is said to be perfect, if the induced graph $G[M]$ is a spanning subgraph of G.

In particular, these matchings are related to the notion of graph factorisations.

3.2 Factorisations of graphs

A spanning subgraph of a graph G is also known as a *factor* of G. A *factorisation* of G is a set of factors whose edge sets constitute a partition of the edge set $E(G)$. A factor is called a k*-factor*, if it is k-regular, while a k*-factorisation* is a factorisation into k-factors. From the above terminology, it is clear that a perfect matching in a graph is a 1-factor of that graph. If two vertices are connected by an edge in a perfect matching, the two vertices are said to be *matched*.

3.3 Tutte's 1-factor theorem

In [Tutte, 1947], William S. Tutte published a theorem that characterises the set of all nontrivial graphs that have 1-factors. To state it, we require the following definitions.

3.3.1 Definition (Odd component):
An odd component of a graph G is a component of G that has an odd number of vertices.

The number of odd components of a graph G is denoted by $oc(G)$.

3.3.2 Definition (Tutte's condition):
Tutte's condition is said to apply for a graph G if, for every subset $W \subseteq V(G)$, it holds that $oc(G - W) \leq |W|$.

3.3.3 Definition (Symmetric difference of graphs):
Let M and N be two spanning subgraphs of G. The symmetric difference of M and N is denoted by $M \triangle N$ and defined as the spanning subgraph of G whose edge set is defined by $(E(M) \cup E(N)) \setminus (E(M) \cap E(N))$.

We may now characterise all graphs that contain 1-factors.

3.3.4 Theorem (Tutte's 1-factor Theorem):
A nontrivial graph has a 1-factor, if and only if it obeys Tutte's condition of Definition 3.3.2.

The proof follows the strategy of [Czygrinow, 2014].

Proof:
Suppose first that G has a 1-factor M, and let W be some subset of $V(G)$. Then every odd component of $G - W$ has at least one vertex matched, with respect to M, to another vertex that must be in W, since all vertices are matched pairwise. Since the existence of an odd component in $G - W$ implies the existence of a vertex in W, it must be true that $oc(G - W) \leq W$.

Now suppose for contradiction that G has no 1-factor, but obeys Tutte's condition. First, let $\tilde{G} = G + e$ by the graph constructed by adding any edge e to G. If $\tilde{G}$ does not have a 1-factor, then neither does G. Additionally, $oc(\tilde{G} - W)) \leq oc(G - W)$, since the addition of an edge cannot increase the number of odd components. Because of this, we may assume that G is edge-maximal.

Construct $U = \{v \in V(G) \mid \deg(v) = |V(G)| - 1\}$.

If $G - U$ consists of disjoint complete graphs, then clearly $oc(G - U) \leq |U|$. Note that $G - U$ may be a single complete graph. Each of the even components of $G - U$ have a 1-factor, while, for every odd component, a matching can be found that leaves only one vertex of $G - U$ unmatched, since these are complete subgraphs by assumption. Each of these unmatched vertices in $G - U$ can be matched to a vertex in U since, by construction, every vertex of U is connected to every other vertex of G.

It remains to be shown that the unmatched vertices of U can be perfectly matched to construct a 1-factor of G. Since U induces a complete subgraph, it will contain a 1-factor if an even number of vertices in U are left unmatched. Since an even number of vertices have already been matched in G, U will have an even number of vertices, and thus a 1-factor, if $|V(G)|$ is even. Tutte's condition holds by assumption, and we may take W to be empty. Then Tutte's condition implies that G contains no odd components. Hence, G contains only even components, and thus an even number of vertices. This implies that G contains a 1-factor.

Now suppose that $G - U$ does not consist entirely of disjoint complete graphs. Then $G - U$ must contain two non-adjacent vertices u and v such that there exists a single vertex x adjacent to both u and v. This is true since not every component is a complete graph, hence, at least one component must contain a shortest path between two vertices of length at least 2. Choose two vertices on this path such that they are separated by a vertex. There must also exist some vertex $y \in V(G - U)$ that is not adjacent to x, since x was not in U.

Denote by e_1 the edge between u and v, and denote by e_2 the edge between x and y. Now, neither e_1 or e_2 is contained in $E(G)$, and by the assumption that G was edge-maximal under the restriction that G did not contain a 1-factor, both $G_1 = G + e_1$ and $G_2 = G + e_2$ contain 1-factors, denoted M_1 and M_2, respectively.

Denote by F the symmetric difference $M_1 \triangle M_2$. Definition 3.3.3 implies that F contains both e_1 and e_2. Since every vertex in M_1 and M_2 has degree 1, every vertex in F must have degree 0 or 2. Thus, every component of F is either an isolated vertex or a vertex in a cycle. Since both M_1 and M_2 are 1-factors, any cycle created by taking the symmetric difference of these matchings must have edges of alternating origin, exactly because they are both perfect matchings, implying that any such cycle must be of even length.

Denote by C the component of F containing e_1. If C does not contain e_2, then it is easy to construct a 1-factor of the even cycle C that does not include e_1, and it is equally easy to construct a 1-factor of the even cycle that contains e_2, but does not contain e_1.

Suppose now that C is a cycle containing both e_1 and e_2. Because u, v, and x are in the same component, there exist paths P_1 connecting u to x and P_2 connecting v to x. A cycle C can be created by connecting these two paths by the edge e_1, that is, $C = P_1 e_1 P_2$. Suppose without loss of generality that y is a vertex in the path P_2, and consider the matchings $\tilde{M}_1 = E(P_2) \cap M_1$ and $\tilde{M}_2 = E(P_1) \cap M_2$. Finally, with e' denoting the edge between x and v, set $\tilde{M} = \tilde{M}_1 \cup \tilde{M}_2 \cup e'$. Now $\tilde{M} \cup (M_1 \setminus E(C))$ constitutes a 1-factor of G. ■

Petersen's 1-factor theorem

While Tutte's 1-factor theorem constitutes one criteria by which a graph can be characterised as containing a 1-factor or not, several other criteria imply the existence of factors in graphs. One example of such a result is the following.

3.3.5 Theorem (Petersen's 1-factor theorem):
If a 3-regular graph G is bridgeless, then G has a 1-factor.

To ease reading in the following, denote by $OC(G)$ the subgraph of G consisting exactly of the odd components of G.

It may be easier to discern whether a given cubic graph contains a bridge than it would be to decide whether the graph obeys Tutte's condition.

Proof:
By Theorem 3.3.4, we need only argue that a graph being bridgeless and 3-regular implies that it obeys Tutte's condition as in Definition 3.3.2. Choose some arbitrary subset $W \subseteq V(G)$, and denote by k the number of edges joining any vertex in W to any vertex in $OC(G-W)$. Since G is 3-regular, we must have $k \leq 3|W|$, because it generally holds that no more than $\Delta(G)$ edges could join any vertex in W to any vertex in $G-W$, and, certainly, any vertex in $OC(G-W)$. For any component H of $OC(G-W)$, denote by k_H the number of edges between a vertex in H and a vertex in the set W. This constitutes a single summand in the sum

$$\sum_{H \subseteq OC(G-W)} k_H = k.$$

The sum of all vertex degrees in the odd component H is equal to $3|V(H)| - k_H$. It is well-known that the degree sum of a graph is an even number, and since H was taken as an odd component, the number $|V(H)|$ is odd, hence $3|V(H)|$ is also odd. Because $3|V(H)| - k_H$ was even, it follows from the above that k_H must be odd.

Since G contains no bridges, k_H is at least 2 for every odd component H chosen in $OC(G-W)$. Since k_H must be odd, we have $k_H \geq 3$. This implies that

$$3oc(G-W) \leq \sum_{H \subseteq OC(G-W)} k_H = k \leq 3|W|.$$

This further implies that $oc(G-W) \leq |W|$ for any subset $W \subseteq V(G)$, also known as Tutte's condition of Definition 3.3.2. ■

As a corollary of Petersen's 1-factor theorem, every bridgeless 3-regular graph must also have a 2-factor, namely the edge-complement of the 1-factor guaranteed to exist by the theorem.

3.4 Basic flow theory as a technical tool

The purpose of this section is to provide results on the existence of certain matchings and factors in particular graphs. However, our approach to proving these results require the use of an alternative perspective on the concept of matching. We turn our attention to the theory of flows in directed graphs. The following definitions and results are immediately necessary throughout the chapter, and are quoted from [Gross et al., 2006; Chapter 13] to recall notation and introduce terminology. A more thorough treatment of the subject may be found in the referenced chapter.

Consider a digraph G on at least two distinguished vertices s and t, respectively called the source and the sink of the digraph G. Assign to each arc e a nonnegative real number, a so-called capacity, denoted by $cap(e)$, of the edge e. Such a digraph is called a network. Let W and U be subsets of $V(G)$. We denote by (W,U) the set of arcs that begin in a vertex of W and end in a vertex of U. Denote by $E^+(v)$ and $E^-(v)$ the set of arcs that begin, respectively end, in the vertex v. Now, a flow on G is defined as a mapping $f : E(G) \to \mathbb{R}^+$ that obeys two conditions:

- $f(e) \leq cap(e)$ for every arc $e \in E(G)$
- $\sum_{e\in E^+(v)} f(e) = \sum_{e\in E^-(v)} f(e)$ for every vertex $v \in V(G)$ that is neither a source nor sink of G.

The source, s, of G obeys that $\sum_{e\in E^+(s)} f(e) \geq \sum_{e\in E^-(s)} f(e)$, while the sink, t, obeys that $\sum_{e\in E^+(t)} f(e) \leq \sum_{e\in E^-(t)} f(e)$. The value of a flow f on a network G is denoted by $val(f)$ and defined by

$$val(f) = \sum_{e\in E^+(s)} f(e) - \sum_{e\in E^-(s)} f(e).$$

A maximum flow f_{max} in a network G is a flow which obeys $val(f_{max}) \geq val(f)$ for every other flow f in the network G. Finding such a maximum flow has obvious practical applications, and one well-known algorithm for finding a maximum flow is the Ford-Fulkerson algorithm. A popular and efficient modification of this algorithm is known as the Edmonds-Karp algorithm. A thorough treatment of both algorithms may be found in [Cormen et al., 2009; Chapter 26]. We state without proof the fact that a maximum flow in a network of integer capacities must also have an integer value. Finally, a cut in a network G is an arc set (V_s, V_t) constructed such that V_s and V_t form a partition of $V(G)$, where $s \in V_s$ and $t \in V_t$. The capacity of a cut (V_s, V_t), denoted $cap((V_s, V_t))$, is given by the sum of the capacities of the arcs in the set (V_s, V_t). That is, $cap((V_s, V_t)) = \sum_{e\in(V_s,V_t)} cap(e)$. A minimum cut $(V_s, V_t)_{min}$ in a network G is a cut whose value is minimal among all other cuts of G.

We now state, without proof, the famed Max-flow Min-cut Theorem before moving on to results on the existence of factors of certain graphs.

3.4.1 Theorem (Max-flow Min-cut Theorem):
Given a network G, the value of a maximum flow is equal to the capacity of a minimum cut. That is,

$$val(f_{max}) = cap((V_s, V_t)_{min}).$$

3.5 Hall's theorem

Consider first a bipartite graph G with its vertex set $V(G)$ partitioned into the sets X and Y. Construct a digraph, $\tilde{G}$ whose vertex set is $V(\tilde{G}) = V(G) \cup \{s, t\}$, where s and t in the following will be referred to as the source, respectively the sink, of $\tilde{G}$. The arc set of $\tilde{G}$ is constructed as follows:
For each edge between a vertex in X and one in Y, orient the edge such that it begins at the vertex in X and ends at the vertex in Y. Now join each vertex of X to the source s by an arc that ends at the vertex located in X, and join each vertex of Y to the sink t by an arc ending at the sink t. Finally, assign to each arc $e \in E(\tilde{G})$ the capacity 1. Finding a maximum matching in the graph G now amounts to the problem of finding a maximum flow in the constructed digraph $\tilde{G}$. A proof of the above statement is aided by the following.

3.5.1 Proposition (Correspondence between flows and matchings):
Let G be a bipartite graph, and let $\tilde{G}$ be a digraph constructed as in the above text. Then there is a one-to-one correspondence between the integral flows of $\tilde{G}$ and the matchings in G.

Proof:
First, let f be an integer flow in the digraph $\tilde{G}$, with $\tilde{G}$ constructed as above using G. Since the capacity of every arc $e \in E(\tilde{G})$ is 1, the contribution of that arc to the value of the flow is either 0 or 1. Denote by M the set of arcs e in $\tilde{G}$ for which $f(e) = 1$. Since every arc in $\tilde{G}$ has capacity 1, the flow property of f can only be preserved if at most one arc is directed away from any given vertex in X, and a similar argument applies to the vertices of Y. That is, at most one arc can end in a vertex of Y. Hence, the set M constitutes a matching in G.

Consider a bipartite graph G with a matching M. Construct from G a digraph $\tilde{G}$ as described above, and define the function $f : E(e) \to \{0, 1\}$ by the following rule:

$$f(e) = \begin{cases} 1 \text{ if the arc } e \text{ corresponds to an edge } e' \in M \\ 1 \text{ if the arc } e \text{ is adjacent to an edge } e' \in M \\ 0 \text{ otherwise.} \end{cases}$$

Consider a vertex $v \in G$, and assume without loss of generality that $v \in X$. Since M is a matching, at most one edge of M is adjacent to v. The first case of the definition of f ensures that the arc corresponding to this edge contribues the value 1 to the flow. Further, since $v \in X$, a unique arc will begin in s and end in v. The second case of the definition of f ensures that this arc is also assigned the value 1. Since all other arcs in $\tilde{G}$ are assigned the value 0, the flow property is preserved, and f is seen to be a flow in $\tilde{G}$. ■

3.5.2 Corollary (Maximum flows and maximum matchings correspond):
Let G be a bipartite graph with a matching M, and let f be a flow in the digraph $\tilde{G}$ constructed as above using G. Then the value of the flow f, $val(f)$, is equal to $|M|$, and f is a maximum flow if and only if M is a maximum matching.

The proof of this corollary follows directly from Proposition 3.5.1. Having thus established the correspondence between matchings and flows, we turn our attention to a particular kind of matching.

3.5.3 Definition (Saturating matching):
A matching M in a bipartite graph G with vertex partition $V(G) = X \cup Y$ is said to be an X-saturating or Y-saturating matching, if every vertex of X, respectively Y, is the end of an edge $e \in M$.

Note that any saturating matching in G must be a maximum matching, since no edge adjacent to a vertex of the saturated subset of $V(G)$ can be added to the matching.

Consider a bipartite graph G with vertex partition $V(G) = X \cup Y$, and let W be a subset of X. In the following, we denote by $N(W)$ the neighbourhood of the vertex set

W, in the sense that $N(W)$ is the subset of Y containing exactly those vertices who are adjacent to at least one vertex in W. The following result was first published in [Hall, 1935].

3.5.4 Theorem (Hall's Theorem for Bipartite Graphs):
Let G be a bipartite graph with vertex partition $V(G) = X \cup Y$. The graph G has an X-saturating matching, if and only if it obeys Hall's condition, that is, if for every subset W of X, the inequality $|W| \leq |N(W)|$ holds.

The proof of the theorem follows the strategy shown in [Gross et al., 2006; Section 13.4].

Proof:
Let M be an X-saturating matching in G, and let W be an arbitrarily chosen subset of X. Then each vertex of W is matched to a single vertex of Y, implying that $|W| \leq |N(W)|$.

Now let G be a bipartite graph that obeys Hall's condition. Using G, construct the digraph $\tilde{G}$ as described in Section 3.5. By Corollary 3.5.2, an X-saturating matching in G corresponds to a maximum flow f in $\tilde{G}$ with the value $val(f) = |X|$. By the Max-flow Min-cut theorem referenced in the beginning of this chapter, Theorem 3.4.1, the flow f can be shown to be a maximum flow if a minimum s-t cut can be shown to have capacity $|X|$.

One such cut of capacity $|X|$ would be $(\{s\}, X \cup Y \cup \{t\})$, so we need only show that any other s-t cut of $\tilde{G}$ has at least capacity $|X|$. Denote by (V_s, V_t) an s-t cut in $\tilde{G}$, and set $W = V_s \cap X$. The cut (V_s, V_t) may now be represented by the union of three disjoint cuts:

$$(V_s, V_t) = (\{s\}, V_t \cap X) \cup (W, V_t \cap Y) \cup (V_s \cap Y, \{t\}).$$

Since every arc of $\tilde{G}$ has capacity 1, we have that

$$cap((V_s, V_t)) = |(\{s\}, V_t \cap X)| + |(W, V_t \cap Y)| + |(V_s \cap Y, \{t\})|,$$

where the addition of the cardinalities is due to the arc subsets being disjoint. By the construction of $\tilde{G}$, we also have that

$$|(\{s\}, V_t \cap X)| + |(W, V_t \cap Y)| + |(V_s \cap Y, \{t\})| = |X \setminus W| + |(W, V_t \cap Y)| + |V_s \cap Y|.$$

By the definition of $N(W)$, we now have

$$X \setminus W + |(W, V_t \cap Y)| + |V_s \cap Y| \geq |X \setminus W| + |V_t \cap N(W)| + |V_s \cap Y|.$$

It is also true that $|V_t \cap N(W)| = |N(W)| - |V_s \cap N(W)|$, and since $N(W) \subseteq Y$, we must also have that

$$\begin{aligned} |X \setminus W| + |V_t \cap N(W)| + |V_s \cap Y| &\geq |X \setminus W| + |N(W)| - |V_s \cap Y| + |V_s \cap Y| \\ &= |X \setminus W| + |N(W)|. \end{aligned}$$

Finally, by the assumption that G obeyed Hall's condition, we have that $|X \setminus W| + |N(W)| \geq |X \setminus W| + |W|$, which is certainly equal to $|X|$. Hence, the capacity of an artbitrarily chosen s-t cut in $\tilde{G}$ is at least $|X|$, completing the proof that Hall's condition guarantees the existence of an X-saturating matching. ■

Returning now to the object of interest in Section 3.2, we give two results on factorisation whose proofs rely on Hall's Theorem, 3.5.4.

3.5.5 Theorem (König's 1-factorisation Theorem):
A nontrivial r-regular bipartite graph G has a 1-factorisation.

Proof:
Suppose that G has vertex partition $V(G) = X \cup Y$, and let W be a subset of X. Since G is r-regular, we have that $r\,|W|$ edges join vertices in W to vertices in Y. At most r edges can be incident on any single vertex of Y, so by the pigeonhole principle, the inequality

$$|N(W)| \geq \left\lceil \frac{r\,|W|}{r} \right\rceil = |W|$$

holds. This is Hall's condition as described in Theorem 3.5.4. Hence, the graph G has an X-saturating matching. Since G is regular, we have that $|X| = |Y|$, which implies that a saturating matching, necessarily also a maximum matching, constitutes a 1-factor of G. Deleting the edges contained in such a 1-factor leaves behind an $(r-1)$-regular bipartite graph to which similar arguments apply, whereupon the result follows by induction. ■

3.5.6 Theorem (Petersen's 2-factorisation Theorem):
Every r-regular graph G has a 2-factorisation if r is even.

Proof:
The graph G can be considered to be connected, since a factorisation of a graph requires the factorisation of each of its components. Let G be such a connected graph on vertices listed as $v_1, v_2, \ldots, v_n$, and assume that G is r-regular, with r being an even natural number. Then G must contain an eulerian circuit, here denoted by C. Define a bipartite graph H with vertex partition $V(H) = U \cup W$, where the vertices of U and W are denoted by $u_1, u_2, \ldots, u_n$ and $w_1, w_2, \ldots, w_n$, respectively. Specifically, construct H such that the vertex partition of $V(H) = U \cup W$ obeys the condition that u_i is adjacent to w_j if the eulerian circuit C contains the edge between v_i and v_j.

Now, the bipartite graph H is $\frac{r}{2}$-regular, since the eulerian circuit C enters and leaves each vertex in G a combined number of r times. The above theorem, Theorem 3.5.5, now grants us the existence of a 1-factor of H. By construction, the edge joining u_i to w_j corresponds to the edge joining v_i and v_j, which, taken with the fact that each vertex appears only once in the set of vertices connected by the 1-factor of H, implies that the edges in G corresponding to the edges in the 1-factor of H constitute a 2-factor of G.

Finally, deleting the edges of this 2-factor of G from G leaves behind a $(2r-2)$-regular graph, and similar arguments apply to this reduced graph. Then, by induction, the result follows. ■

As we have now treated results on edge colourings, matchings, and factorisations, we turn our attention to a more narrow notion of edge colouring in the following chapter.

Chapter 4
Strong edge colouring

A strengthening of the definition of a proper edge colouring is the notion of a strong edge colouring.

4.0.1 Definition (Strong edge colouring):
A proper edge colouring of a graph G is called a strong edge colouring if no edge $e \in E(G)$ is adjacent to two edges of the same colour.

It is seen from the definition that, given a strong edge colouring of some graph G, every path in G of length 3 must be edge coloured with three distinct colours.

4.0.2 Definition (Strong chromatic index):
The strong chromatic index of a graph G is the least integer k for which G is strongly edge k-colourable. The strong chromatic index of a graph G is denoted $s'(G)$.

Naturally, it would be interesting to bound the strong chromatic index of graphs in as general a way as possible. Before specifying any such bound, we introduce the terminology that an edge $e_1 = xy$ in a graph G is *joined* to another edge $e_2 = vw$ if any of the following edges exist in $E(G)$: xv, xw, yv, or yw.

4.0.3 Proposition:
Let G be a graph. Then the strong chromatic index of G obeys

$$s'(G) \leq 2\Delta(G)^2 - 2\Delta(G) + 1.$$

Proof:
Modify Algorithm 2.1 so that, at the i'th iteration of step 2, instead of assigning to e_i the least colour that is not assigned to any adjacent edges, the algorithm should assign to e_i the least colour that is not assigned to any edges joined to e_i. This obviously produces a strong edge colouring. Since at the i'th iteration of step 2, the edge e_i can have at most $2\Delta(G) - 2$ adjacent vertices, and since each of these vertices may be incident with at most $\Delta(G)$ edges, we have that $2\Delta(G)^2 - 2\Delta(G)$ colours are sufficient to strongly colour the graph $G - e_i$, implying that $2\Delta(G)^2 - 2\Delta(G) + 1$ colours are sufficient to strongly colour G. ■

According to [Faudree et al., 1989], a general bound was conjectured by Paul Erdős and Jaroslav Nesetril at a seminar in Prague in 1985.

4.0.4 Conjecture:
Let G be a graph. Then

$$s'(G) \leq \begin{cases} \frac{5}{4}\Delta(G)^2 \text{ if } \Delta(G) \text{ is even} \\ \frac{5}{4}\Delta(G)^2 - \frac{2\Delta(G)-1}{4} \text{ if } \Delta(G) \text{ is odd.} \end{cases}$$

We show why this bound is the least possible.

4.0.5 Proposition:
Let $\Delta(G)$ be a natural number. Then there exists a graph G such that

$$s'(G) = \begin{cases} \frac{5}{4}\Delta(G)^2 \text{ if } \Delta(G) \text{ is even} \\ \frac{5}{4}\Delta^2 + \frac{1-2\Delta}{4} \text{ if } \Delta(G) \text{ is odd.} \end{cases}$$

Graphs that achieve this bound are constructed using a strategy outlined in [Dębski, 2015; Section 2.3].

Proof:
Assume that Δ is an even natural number. We construct G such that $\Delta(G) = \Delta$, and such that $s'(G) = \frac{5}{4}\Delta(G)^2$. Let

$$V(G) = \{(i,j) \mid i \in \{1,2,3,4,5\}, j \in \{1,2,\ldots,\frac{\Delta}{2}\}\}.$$

Let $v_1 = (i_1, j_1)$ and $v_2 = (i_2, j_2)$ be two vertices in $V(G)$. Connect these by an edge if $i_1 \equiv i_2 \pm 1 \mod 5$. From this, it is seen that G is Δ-regular. To show that G constructed in this fashion obeys $s'(G) = \frac{5}{4}\Delta(G)^2$, we show that $|E(G)| = \frac{5}{4}\Delta(G)^2$, and that no two vertices exist such that the shortest path between them contains three edges. If this is the case, then $s'(G) = |E(G)|$. The Δ-regular graph G contains $5\frac{\Delta}{2}$ vertices, which implies that the degree sum is of the graph is given by $\frac{5}{4}\Delta^2$. Every edge contributes to this degree sum twice, so dividing the degree sum by 2 equals to the number of edges: $\frac{5}{4}\Delta(G)^2$.

To see that $s'(G) = |E(G)|$, assume that the shortest path between two vertices, say $v_1 = (i_1, j_1)$ and $v_4 = (i_4, j_4)$, has length 3. This would imply the existence of two additional vertices, say $v_2 = (i_2, j_2)$ and $v_3 = (i_3, j_3)$, such that i_1, i_2, i_3, i_4 all pairwise obey the congruence $i_n \equiv i_{n+1} \pm 1 \mod 5$, while at the same time obeying the incongruences $i_n \not\equiv i_{n+2} \pm 1 \mod 5$ and $i_n \not\equiv i_{n+3} \pm 1 \mod 5$ This is clearly impossible, so there is no shortest path of length 3 between any two vertices of $V(G)$.

Now assume that Δ is an odd natural number. We construct G' with respect to $\Delta(G') = \Delta$ and $s'(G') = \frac{5}{4}\Delta^2 + \frac{1-2\Delta}{4}$, using the graph constructed with respect to the even number $\Delta - 1$. Construct this graph as above, and add to $V(G)$ the vertices $v_{1,2}$ and $v_{3,4}$, that is, $V(G') = V(G) \cup \{v_{1,2}, v_{3,4}\}$. Connect the vertices $v_{1,2}$ and $v_{3,4}$ by an edge, and add to $E(G)$ every edge on the form $(v_{1,2}, (1,j))$, $(v_{1,2}, (2,j))$, $(v_{3,4}, (3,j))$, and $(v_{3,4}, (4,j))$. This constructs $E(G')$.

By the attributes of G, we need only show that the graph G' constructed in this fashion obeys $|E(G')| = \frac{5}{4}\Delta^2 + \frac{1-2\Delta}{4}$, and that the shortest path between any two vertices of G' remains at most of length 2. The latter fact is the most immediate. Indeed, choose some vertex $v \in V(G')$. If v is not adjacent to $v_{1,2}$ or $v_{3,4}$, then v must be on the form $(0, j)$. Regardless of the value of j, both $v_{1,2}$ and $v_{3,4}$ are adjacent to a vertex that is again adjacent to v, implying the existence of a shortest path of length 2.

The edge set $E(G')$ contains the edges in $E(G)$ along with the $2(\Delta - 1) + 1 = 2\Delta - 1$ edges added to construct $E(G')$. By the construction of G, this is equal to

$$\frac{5}{4}(\Delta - 1)^2 + 2\Delta - 1 = \frac{5}{4}\Delta^2 + \frac{1 - 2\Delta}{4}. \qquad \blacksquare$$

4.1 The strong chromatic index of bipartite graphs

The strong chromatic index of bipartite graphs is a special case of Conjecture 4.0.4. Faudree et al. conjectured in [Faudree et al., 1989] that any bipartite graph G obeys $s'(G) \leq \Delta(G)^2$. It can be shown that $s'(K_{n,n}) = n^2$, implying that the conjectured bound, if true, would also be sharp. Four years later, [Steger et al., 1993] proved the conjectured result for $\Delta(G) \leq 3$ and gave their own version of the conjecture which stated that, given a bipartite graph G with vertex partition $V(G) = X \cup Y$, the strong chromatic index of G would satisfy the bound $s'(G) \leq \Delta(X)\Delta(Y)$, with $\Delta(X)$ and $\Delta(Y)$ naturally denoting the maximal degree of any vertex in X, respectively Y. This was affirmed by [Nakprasit, 2008] for the case of $\Delta(X) = 2$. We give the proof of this result here. Note that the case of $\Delta(X) = 1$ is trivial, since $\Delta(Y)$ colours would suffice.

4.1.1 Theorem:
Let G be a bipartite graph with vertex partition $V(G) = X \cup Y$, and assume that $\Delta(X) = 2$. To ease notation, we denote by Δ the maximal degree among vertices of Y. Then

$$s'(G) \leq 2\Delta.$$

Proof:
Let G be a bipartite graph with vertex partition $V(G) = X \cup Y$ and let $\Delta(X) = 2$. We may assume that $\delta(X) = 2$. Let $x \in X$ have degree 1. Then the edge joining x to some vertex $y \in Y$ can be removed from G, and $G - xy$ can still be colored using at most $2(\Delta - 1)$ colours. This leaves an additional colour available to be assigned to the edge xy, so we may assume that $\deg(x) = 2$ for any vertex $x \in X$.

The proof relies on manipulation of an array of dimensions $|Y| \times \Delta$, from here on denoted A, containing edges of $E(G)$. Index each row by an element of Y and insert in each column of the row corresponding to, say, y_i, those edges which are adjacent to the vertex y_i. Clearly, if $\deg(y_i) < \Delta$, at least one entry of the i'th row corresponding to y_i will be empty. Exchanging two edges in the same row or moving an edge from one column in its given row to another column, will not change the association between the vertices and the rows. We now introduce the terminology that a vertex in X is *good* if both of its incident edges are in the same column, and call the same vertex *bad* whenever this is not the case.

Now, exchange and move edges in A until the number of good vertices is maximal, and observe the following claim.

Claim: If a vertex $v \in X$ is bad, there exists a cycle C in G obeying the following:

(i) $v \in C$

(ii) $\frac{|C|}{2}$ is odd

(iii) All vertices of $C \cap X$ other than v are good ones

(iv) All edges incident to the vertices of $C \cap X$ are contained in two columns of A.

Proof: Assume that x is a bad vertex of degree 2 and denote by e_1 and e_2 its incident edges. We may assume without loss of generality, that e_1 is in column 1 and e_2 is in column 2 of the array A. If no edge occupies the first column of the row in which e_2 lies, the edge e_2 may be moved here. Hence, we assume that an edge e_3 is contained in the first column of the row that contains e_1. Now, if e_3 is incident to a bad vertex, the edges e_2 and e_3 may be exchanged to produce one good vertex without incurring another bad one. Hence, we may assume that e_3 is incident to a good vertex, say v. Denote by e_4 the other edge that is incident to v. This implies that e_4 is in the same column as e_3, that is, the first column.

If there is no edge in the second column of the row containing e_4, then e_4 may be moved to the second column. Hence, we may suppose that an edge e_5 occupies this second column of the same row that e_4 occupies. If e_5 is incident to a bad vertex, the exchange operation may be employed to swap e_2 with e_3, and e_4 with e_5. Once again, this constructs at least one more good vertex and creates no additional bad ones. Repeating the argument as before, we may again assume the existence of an edge e_6 that is incident to the same good vertex as e_5. Since both e_5 and e_6 are incident to a good vertex, they occupy the same column, that is, the second column. Now $e_1e_2e_3e_4e_5$ forms a path by the construction of the array A and the implied *goodness* of the connected vertices.

The cycle C mentioned in the statement of the claim can now be constructed. Since G is finite, the above arguments cannot be repeated indefinitely. This implies the existence of an edge e_t such that the path $e_1e_2\cdots e_t$ cannot be extended any further by the same arguments. Now, as above, e_{2i} and e_{2i+1} are incident to the same good vertex, while e_j and e_{j+1} are in the same column for every j. Since the argument constructing the path terminates at e_t, this edge must be incident to e_1. We can write $t = 2k+2$ for some integer k, and by the above, $e_1e_2\cdots e_{2k+2}$ must constitute a cycle, C. ■

A cycle C constructed in this way is called *vicious*. By the above arguments, each bad vertex corresponds to exactly one vicious cycle. By the construction of vicious cycles, these are pairwise disjoint, so we can consider the edge colouring of these independently. Define a colouring $c : E(G) \to S$ by $c : e \mapsto (i, j)$ if e is an edge located in the j'th column of the array A. Naturally, i may be either 1 or 2, and the assignment is decided as follows:

When colouring vicious cycles:

If e is incident to some bad vertex x, then $i = 1$. For each vertex $y \in C \cap Y$ that is not adjacent to x, we assign to the incident edges of y in C the first colour coordinate $i = 1$.

For each good vertex in the cycle C, and to every vertex adjacent to x, different colours are assigned to the edges connecting these.

When colouring the remainder of $E(G)$:
After colouring vicious cycles, only good vertices remain in G. For every edge incident to such vertices, the first coordinate of its colour may be chosen arbitrarily.

What remains is to show that the colouring constructed in this fashion is a strong edge 2Δ-colouring of G. We begin by showing that the described edge colouring is proper. Suppose that e and $\tilde{e}$ are two edges incident to the same vertex in Y or to a bad vertex in X. Then, by the above construction, e and $\tilde{e}$ are moved to different columns, implying that the second coordinates of $c(e)$ and $c(\tilde{e})$ differ. If, on the other hand, e and $\tilde{e}$ are both incident to a good vertex of X, then $c(e)$ and $c(\tilde{e})$ have distinct first coordinates. This is an immediate consequence of the described colour assignments.

It remains to be shown that this proper edge colouring is also a strong edge colouring of G. Let $e = (x, y)$ and $e' = (x', y')$ be two edges that constitute a matching in G, where the edges e and e' are not joined by any single edge of G. Further, assume that $x, x' \in X$ and $y, y' \in Y$. Assume without loss of generality that (x, y') is an edge in $E(G)$. If the edges e and e' are stored in distinct columns of the array A, then the second coordinates of $c(e)$ and $c(e')$ also differ. If, on the other hand, e and e' are stored in the same column, the existence of the edge (x, y') implies that (x, y') and e' are in distinct columns of A. This implies that x is a bad vertex and that (x, y') is an edge in the vicious cycle corresponding to x. Then, by the construction of c, the first coordinates of $c(e)$ and $c(e')$ must differ. This implies that c is a strong edge 2Δ-colouring. ■

4.2 The strong chromatic index of planar graphs

Planar graphs are a class of graphs whose strong chromatic index is quite easy to present a different bound on. The section is inspired by [Hudák et al., 2013], who attributes the original result to [Faudree et al., 1990].

4.2.1 Proposition (Strong chromatic index of planar graphs):
Let G be a planar graph. Then

$$s'(G) \leq 4\Delta(G) + 4.$$

Proof:
By Vizing's Theorem, Theorem 2.3.2, the graph G is $\Delta(G) + 1$-edge colourable. Let c be an edge colouring of G and denote by $A_i \subset E(G)$ the edge colour class $A_i = \{e \in E(G) \mid c(e) = i\}$. Further, denote by $G(A_i)$ the graph G in which every edge contained in A_i is contracted. Note that edges in G that are joined by an edge in A_i will be adjacent in the contracted graph $G(A_i)$. Edge contraction does not change the planarity of G, so when G is assumed to be planar, the graph $G(A_i)$ will also be planar for any edge colour class A_i. Since $G(A_i)$ is planar, the celebrated Four-Colour Theorem published in [Appel et al., 1989] implies that the graph $G(A_i)$ can be properly vertex coloured using four colours, irrespective of the choice of A_i.

Every vertex corresponding to an edge in $E(G)$, that is adjacent to a pair of edges in A_i, is assigned a distinct colour among the four colours needed to colour the vertices of $G(A_i)$. Vertex colouring the at most $\chi'(G)$ graphs on the form $G(A_i)$, where $i \in \{1, 2, \ldots, \chi'(G)\}$, yields a strong edge colouring of G using at most $4(\Delta(G) + 1) = 4\Delta(G) + 4$ colours. ■

The general language of Conjecture 4.0.4 has seen few meaningful attacks. One, however, dates back to 1990, and is covered in the following chapter.

Chapter 5

The strong chromatic index of cubic graphs

In 1990, Lars D. Andersen confirmed the statement of Conjecture 4.0.4 for the case of $\Delta(G) = 3$. The result was published in [Andersen, 1992], and while the treatment in the original publication focuses heavily on the construction of a linear-time algorithm for strongly edge 10-colouring cubic graphs, it is the purpose of this chapter to present the proof of the more simple fact that, for a graph G, it is true that $\Delta(G) = 3$ implies $s'(G) \leq 10$.

While we are not preoccupied with the running time of the algorithm proposed in the article, we will still rely on some of its defining properties as theoretical tools. Specifically, many of our results are proven constructively by the use of some proposed algorithm. Despite this, we still follow [Andersen, 1992].

Throughout the chapter, we broaden our previous definition of graph to possibly include multiple edges between vertex pairs.

5.1 Preliminaries

Given an edge e in some graph G, define $N(e)$ as the set of edges in G that are either adjacent to e or joined to e by a path of two edges. For a cubic graph G, we would naturally have $N(e) \leq 12$ for any $e \in E(G)$. Denote by $F(e)$ the set of colours assigned to any edge contained in the set $N(e)$. Clearly, we have $F(e) \leq N(e) \leq 12$ for any edge e in a graph where $\Delta(G) \leq 3$.

Throughout the chapter, we modify the usual terminology of an algorithm being *greedy*. Instead, we call an edge-colouring algorithm greedy as long as it operates sequentially with respect to some ordering of $E(G)$ such that, whenever it encounters an edge e, any colour not in $F(e)$ can be arbitrarily chosen, and at least one such colour will always be available to be assigned to e. This ensures that the algorithm never has to retrace its steps and possibly re-assign a different colour to an already coloured edge, and in this sense, it is greedy. Constructing an ordering of $E(G)$ that guarantees that a sequential algorithm produces a satisfying strong edge colouring, however, proves difficult.

We also require the following notion of a *distance class*. Choose some vertex v in a connected graph G. Denote by D_i the i'th distance class of v, defined by the set of vertices connected to v by a shortest path of length i, with $i = 1, 2, \ldots$. Then every vertex of D_{i+1} is adjacent to at least one vertex of D_i.

For any edge $e \in E(G)$ and vertex $v \in V(G)$, denote by $d(e)$ the shortest distance between v and an end-vertex of e. We call an ordering of the edges in $E(G)$ *compatible* with the distance classes of v if an edge e only comes before $\tilde{e}$ when $d(e) \leq d(\tilde{e})$.

5.1.1 Lemma:
Let G be a graph with $\Delta(G) \leq 3$ and let $v \in V(G)$. A greedy edge colouring algorithm that colours the edges of $E(G)$ in the reverse order of an edge ordering compatible with the distance classes of v will produce a partial strong edge 10-colouring that can only leave edges incident to v uncoloured.

Proof:
Let $e \in E(G)$ be any edge that is not incident to v. Then the end vertices of e will be in the distance classes $D_{d(e)}$ and $D_{d(e)-1}$. A greedy algorithm colouring the edges of G in the reverse of an order compatible with the distance classes of v, will not have assigned a colour to the edges of $D_{d(e)-1}$ at the point of assigning a colour to e. This implies that $|F(e)| \leq 9$, which must leave a colour available to be assigned to e. ■

This short lemma actually grants us considerable power of reduction. In the following results, we will often reduce a strong edge colouring problem to only consider some local neighbourhood of a particular vertex.

5.2 Vertices of degree less than 3

5.2.1 Lemma:
Let G be a connected graph with $\Delta(G) \leq 3$. If there exists $v \in V(G)$ of degree 1, then there exists a greedy edge colouring algorithm that strongly edge-colours G using at most 10 colours.

Proof:
Let v be a vertex of degree 1 and consider an ordering of $E(G)$ compatible with the distance classes of v. Colour the edges in the reverse of this order. By Lemma 5.1.1, only the edge e incident to v may be left uncoloured. But by the construction of G, we have that $|N(e)| \leq 6$, so surely, $|F(e)| \leq 6$, leaving at least 4 colours available to be assigned to e. ■

5.2.2 Lemma:
Let G be a connected graph with $\Delta(G) \leq 3$. If there exists $v \in V(G)$ of degree 2, then there exists a greedy edge colouring algorithm that strongly edge-colours G using at most 10 colours.

Proof:
As in the proof of the previous lemma, consider the vertex $v \in V(G)$ of degree 2, and denote by e and e' the edges incident to v, with e being ordered before e'. Colouring the edges in the reverse order, we have again by Lemma 5.1.1 that $|F(e')| \leq 9$ and $|F(e)| \leq 8$. In both cases, this leaves a colour available to be assigned to the edge in question. ■

5.3 Graphs containing n-gons with $n \leq 5$

We define an n-gon as a circuit graph on n vertices. To say that a graph contains an n-gon is to say that it contains a circuit graph on n vertices as a subgraph.

5.3.1 Lemma (Graphs containing multiple edges):
Let G be a connected graph with $\Delta(G) \leq 3$. If G contains a multiple edge, that is, a 2-gon, then there exists a greedy edge colouring algorithm that strongly edge-colours G using at most 10 colours.

Proof:
Let v be a vertex incident with a multiple edge. Order the edges of $E(G)$ in a way compatible with the distance classes of v, and colour the edges in the reverse of this order. Again we have by Lemma 5.1.1 that if e and e' are the two multiple edges incident to v that constitute a 2-gon, we may colour the first, say e, because $|F(e)| \leq 6$, and the second, say e', because we will have $|F(e')| \leq 7$ after colouring e. ■

5.3.2 Lemma:
Let G be a connected graph with $\Delta(G) \leq 3$. If G contains a 3-gon, then there exists a greedy edge colouring algorithm that strongly edge-colours G using at most 10 colours.

Proof:
Let v be a vertex in a 3-gon of G. By Lemma 5.2.2, we are done if $\deg(v) = 2$, so we may assume that $\deg(v) = 3$.

If we choose an ordering of $E(G)$ compatible with the distance classes of v, such that the edges incident to v, say e_1, e_2, and e_3, are the first ones with respect to the order. If we colour the edges in the reverse of this order as in the previous lemma, we will have $|F(e_3)| \leq 9$, $|F(e_2)| \leq 8$, and $|F(e_1)| \leq 9$. This leaves a colour available to assign to each edge at the necessary stage of the algorithm. ■

As we expand the results to also apply for graphs containing n-gons where $n \geq 4$, we need to also introduce the notion that an edge-colouring algorithm can be *greedy except for k edges*. This property is said to apply to a greedy edge-colouring algorithm that produces a partial strong edge-colouring of G which colours at least $|E(G)| - k$ edges, leaving at most k edges uncoloured.

5.3.3 Lemma:
Let G be a connected graph with $\Delta(G) \leq 3$. If G contains a 4-gon, then it may be strongly edge 10-coloured by some edge-colouring algorithm that is greedy except for 8 edges.

Proof:
Denote by C a 4-gon in a graph G, and denote by v_1, v_2, v_3, and v_4 the consecutive vertices appearing in G. Lemma 5.2.2 tells us that we are done if any of these have degree 2, so we assume that none of them do. If any pair of the listed vertices are additionally joined by an edge that is not adjacent to any other vertex of the 4-gon C, Lemmata 5.3.1 and 5.3.2

ensure that the result also holds. Hence, we assume that neither of these two conditions are met, allowing us to denote by f_i the edge incident to v_i that is not part of the 4-gon C. Further, we denote by e_i the edge between vertices v_i and v_{i+1}, with e_4 being the edge between v_4 and v_1.

For each index $i \in \{1, 2, 3, 4\}$, denote by w_i the end vertex of f_i that is not v_i.

It follows from Lemmata 5.2.1 and 5.2.2 that the reduced graph $G - V(C)$ can be strongly edge 10-coloured by a greedy algorithm, since removing the 4-gon C from G would leave each vertex w_i with degree at most 2, with each component of $G - V(C)$ containing a vertex on the form w_i.

Because of this, we may denote by U the set $\{e_1, e_2, e_3, e_4, f_1, f_2, f_3, f_4\}$ and assume the existence of a partial strong edge 10-colouring of $G - U$. We have only to argue that these eight edges can be assigned colours without incurring the need for an 11'th colour in order to preserve the strong edge colouring of G.

By construction of G and the above assumptions, we must have $|F(f_i)| \leq 6$ and $|F(e_i)| \leq 4$ for $i \in \{1, 2, 3, 4\}$.

Denote by $A(e)$ the edges available for assignment to the edge e. Then $|A(f_i)| = 4$ and $|A(e_i)| = 6$ for $i \in \{1, 2, 3, 4\}$.

If we can assign to each edge contained in U a distinct colour among the 10 available ones, we have done nothing to violate the condition that the proper 10-edge colouring should be strong. If this is not possible, that is, if some two edges of U must be assigned the same colour, then Theorem 3.5.4 implies the existence of a nonempty subset W of U for which it holds that

$$\left| \bigcup_{e \in W} A(e) \right| < |W| .$$

We already know that $|A(e)| \geq 4$ for any $e \in U$, so we must have $|W| \geq 5$. If this is true, then the set W contains at least one edge e_i, given that there are only four distinct edges on the form f_i in the set U. Since, for every edge e_i, it holds that $|e_i| \geq 6$, we must have $|W| \geq 7$.

This leaves us with only three cases to consider.

Case 1: $\left|\bigcup_{e \in W} A(e)\right| = 6$**:**

If $W = U$, then w_1 and w_3 are distinct and not adjacent, and so are w_2 and w_4. This is because either equality between the two pairs of vertices would imply that $|A(e_1)| \geq 7$ for the case of w_1 and w_3, or $|A(e_2)| \geq 7$ for the case of w_2 and w_4. If either vertex pair were pairwise adjacent, we would have $|A(e_1) \cup A(e_2)| \geq 7$ for the case of w_1 and w_3, with a similar argument showing that w_2 and w_4 cannot be adjacent. If a multiple edge joins either pair, Lemma 5.3.1 applies.

This leaves the following number of available colours:

$$|A(f_1) \cap A(f_3)| \geq 2$$
$$|A(f_2) \cap A(f_4)| \geq 2.$$

In either case, we take the intersection of two sets of cardinality 4, the union of which can at most contain 6 elements. Because of this, we may assign one colour to f_1 and another to f_3, and likewise for the edges f_2 and f_4. This removes at most two elements from $A(e_i)$ for $i \in \{1,2,3,4\}$, so each edge on the form e_i can be coloured with a distinct colour.

Case 2: $\left|\bigcup_{e \in W} A(e)\right| = 6$**,** $W \neq U$**,** $|W| = 7$**:**
Assume that W contains both f_1 and f_3. The above arguments imply that a single colour can be assigned to both edges. Now the remaining edges of U can still be assigned distinct colours, as both $A(f_2)$ and $A(f_4)$ contain at least three elements. This leaves at least 5 colours in any set of the form $A(e_i)$, permitting the assignment of a distinct colour to each edge on the form e_i. Finally, the single edge contained in U but not in W may be assigned any of the colours that do no appear in $\bigcup_{e \in W} A(e)$.

Case 3: $\left|\bigcup_{e \in W} A(e)\right| = 7$**:**
If this is the case, we need only consider $W = U$. Suppose that $w_1 = w_3$. Now, assuming $w_2 = w_4$ would imply that $|F(e_1)| \leq 2$, leaving the set $A(e_1)$ with at least 8 distinct elements. This cannot be true, so we cannot have both equalities simultaneously.

Hence, assume without loss of generality that $w_2 \neq w_4$. If w_2 and w_4 were adjacent, we would have the inequality $|A(e_1) \cup A(e_3)| \geq 8$, another contradiction. As in the first case, if they are joined by a multiple edge, we apply Lemma 5.3.1.

The final possible subcase is that all four vertices are distinct, with both pairs w_1 and w_3, and w_2 and w_4 being adjacent. This would imply that either a multiple edge connects one of the two pairs, or we would have

$$|A(e_1) \cup A(e_2) \cup A(e_3) \cup A(e_4)| \geq 8,$$

constituting yet another contradiction in the assumed cardinality of $\bigcup_{e \in W} A(e)$. We may assume that w_1 and w_3 is the vertex pair that is not adjacent. Since we have $|A(f_1) \cap A(f_3)| \geq 3$, both edges can be assigned the same colour. This leaves at least 3 colours available for assignment to the edges f_2 and f_4, yet again leaving at least 5 colours available to be assigned to edges on the form e_i. From this it follows that the remaining six edges contained in U can be given distinct colours after first assigning to f_1 and f_3 the same one.

This can only fail if the union of all remaining sets of the form $A(e)$ contains 5 elements. However, if this equality were obtained, then w_2 and w_4 must be distinct and non-adjacent vertices, implying that the edges f_2 and f_4 can be coloured using the same colour without violating the strong edge 10-colouring of G that we wanted. In doing so, we are left with 4 available colours with which to colour the four edges e_1, e_2, e_3, and e_4. ∎

5.3.4 Lemma:

Let G be a connected graph with $\Delta(G) \leq 3$. If G contains a 5-gon, then it may be strongly edge 10-coloured by some edge-colouring algorithm that is greedy except for 10 edges.

The strategy of the following proof is remarkably similar to that of the previous one. The details, however, differ enough to make the proof nontrivial.

Proof:
Let G be a graph with $\Delta(G) \leq 3$, and let G contain a 5-gon denoted C. Consecutively denote the vertices of C by v_1, v_2, v_3, v_4, and v_5, with the edges between these denoted by $e_i = e_i e_{i+1}$ for $i \in \{1, 2, 3, 4\}$, and the edge e_5 connecting the vertices v_5 and v_1. As in the proof of the above lemma, we apply Lemma 5.2.2 in case any of these vertices have degree 2, or Lemmata 5.3.1 and 5.3.2 if any pair of vertices in C are joined by an edge not in the circuit described by C.

Hence, we may once again assume that no such internal edge of the 5-gon exists, and again denote by f_1, f_2, f_3, f_4, and f_5 the edges outside of C that are incident to v_1, v_2, v_3, v_4, and v_5, respectively. Again as in the proof of the above lemma, denote for $i \in \{1, 2, 3, 4, 5\}$ by w_i the endpoint of edge f_i that is not v_i.

If these are not all distinct, they imply the existence of a 3-gon or 4-gon in C, causing Lemmata 5.3.2 and 5.3.3 to apply. Hence, we assume that all five vertices on the form w_i are distinct.

By Lemmata 5.2.1 and 5.2.2, we may greedily strongly edge 10-colour $G - V(C)$, since every vertex in the set $\{w_1, w_2, w_3, w_4, w_5\}$ has degree 2 in $G - U$, and since each component of $G - U$ contains at least one vertex in $\{w_1, w_2, w_3, w_4, w_5\}$.

Hence, we consider a partial strong edge 10-colouring of $G - V(C)$, leaving only the subset of $E(G)$ denoted by $U = \{e_1, e_2, e_3, e_4, e_5, f_1, f_2, f_3, f_4, f_5\}$ as yet uncoloured.

By construction of G, we have both $|F(f_i)| \leq 6$ and $|F(e_i)| \leq 4$ for every $i \in \{1, 2, 3, 4, 5\}$, implying both $|A(f_i)| \geq 4$ and $|A(e_i)| \geq 6$ for $i \in \{1, 2, 3, 4, 5\}$, where $A(e)$ is once again understood to be the set of colours left available for assignment to the edge e, without violating any properties of the strong edge 10-colouring that we wish to construct.

If each edge in U can have assigned to it a distinct colour among the 10 available, we are done.

Assume that this is not possible. We apply Theorem 3.5.4, which implies the existence of some nonempty subset W of U for which

$$\left| \bigcup_{e \in W} A(e) \right| < |W|.$$

The least value of $|A(e)|$ for any edge $e \in U$ is 4, so we must have $|W| \geq 5$. If $|W| = 5$, then W must contain all edges f_i with $i \in \{1, 2, 3, 4, 5\}$, as the union with any set $A(e_i)$ would force W to contain at least six elements. For ease of reading in the following, we name the colours contained in $A(f_i)$ as $\{a, b, c, d\}$. This imposes no immediate ordering on them, but none is required in the proof.

No vertices on the form w_i are adjacent, since this would imply that the edge f_i corresponding to w_i would have at least 5 colours available for assignment. Additionally, by construction of G, we have $A(f_i) \subseteq A(e_i)$ for every $i \in \{1, 2, 3, 4, 5\}$. This allows us to make the following assignments:

- a is assigned to f_1 and f_3
- b is assigned to f_2 and e_4
- c is assigned to f_4 and e_1
- d is assigned to f_5 and e_2

We only need to assign colours to e_3 and e_5. However, we have used the four colours a, b, c, and d, and for each e_i, we had six available colours, that is, $|A(e_i)| \geq 6$ for $i \in \{1, 2, 3, 4, 5\}$. Because this would finish the proof, we assume in the following that $A(f_i) \geq 5$ for $i \in \{1, 2, 3, 4, 5\}$.

Now we must have $|W| \geq 6$, implying that W contains some edge on the form e_i, further implying that $|W| \geq 7$.

Throughout the remainder of the proof, we alleviate notation by introducing $x = \left|\bigcup_{e \in W} A(e)\right|$.

If $x \leq 8$, the edge pair f_i and e_{i+2} are neither adjacent nor joined, hence they may be assigned the same colour. Such an edge pair will be referred to as a pair of *opposite edges*.

For each such pair contained in W, both have at least $10 - x$ colours available, where $10 - x \geq 2$. By Hall's Condition contained in Theorem 3.5.4, we have $|W| > x$, with W containing at least $(x + 1) - 5 = x - 4 \geq 2$ pairs of opposite edges.

We now show that two such pairs can always have two colours assigned in a pairwise fashion, leaving three colours still available to be assigned to the remaining three edges on the form f_i. Since Theorem 3.5.4 implied the existence of at least two pairs of opposite edges in W, we may always choose two such pairs and assign to them each a colour, say a and b. Assume that this leaves the colours c and d available for assignment to the remaining edges on the form f_i.

Denote by $\tilde{a}$ some colour distinct from all four colours a, b, c, and d. The set $A(f_j)$ must contain some such $\tilde{a}$ for some $j \in \{1, 2, 3, 4, 5\}$, because the union $\bigcup_{i=1}^{5} A(f_i)$ was assumed to contain at least five elements.

Choose two pairs of opposite edges contained in the set W and assign to the edges of one pair the colour a before assigning the colour b to the other edge pair. Having done this, we modify our choice of pairs in the following way:

- If $x = 6$, the colour $\tilde{a}$ must be contained in the set $A(e_{j+2})$, implying that $\tilde{a}$ is available for assignment to one pair.
- If $x = 7$ or $x = 8$, one remaining edge on the form f_i belongs to a pair of opposite edges in W. Removing the colour assigned to f_j and e_{j+2} before colouring the remaining opposite edge pair yields the configuration necessary for colouring two opposite edge pairs.

Having thus extracted four edges from U and coloured them without violating the properties demanded of the edge colouring, we need to assign colours to the remaining six edges of U. If it is possible to assign distinct colours to all of them, we are done.

Hence, assume that it is not. Then we once again return to an implication of Theorem 3.5.4. Denote by $\tilde{A}(e)$ the set of edges available for assignment to the edge e, and denote by $\tilde{W}$ some subset of the remaining six edges in U which obeys the condition that

$$\left|\bigcup_{e \in \tilde{W}} \tilde{A}(e)\right| < \left|\tilde{W}\right|.$$

We must have $\left|\tilde{W}\right| \geq 4$, implying that $\tilde{W}$ contains an edge on the form e_i, again implying that $\tilde{W}$ contains five or six elements. This implies that $\tilde{W}$ contains at least two opposite edge pairs.

Case 1: $\left|\bigcup_{e\in\tilde{W}} \tilde{A}(e)\right| = 4$**:**

Whenever this holds true, each pair of opposite edges in $\tilde{W}$ has two colours available for their assignment. Colouring two such pairs with distinct colours leaves, as argued above, a colour available for assignment to whichever edge on the form f_i that still remains to be coloured.

Case 2: $\left|\bigcup_{e\in\tilde{W}} \tilde{A}(e)\right| = 5$**:**

If this is true, $\tilde{W}$ contains all remaining uncoloured edges of G. It is still the case that each opposite edge pair of $\tilde{W}$ can have a distinct colour assigned to both edges, and the two edges on the form f_i that are left uncoloured by this procedure both have at least two colours available for assignment, that is, we still have $A(f_i) \geq 2$ for the uncoloured edges f_i. The configuration is finally resolved by assigning a distinct colour to each remaining edge of U. If this is not possible, each opposite edge pair can be assigned a single colour, freeing one up.

The final case we must consider is when $\left|\bigcup_{e\in W} A(e)\right| = 9$. When this occurs, each opposite edge pair of W has at least one colour available for assignment to both edges of the pair. We have argued that one such pair may be coloured in a way that leaves four colours free for assignment to whichever edges on the form f_i remain uncoloured. Having assigned to an opposite edge pair one such colour, we may attempt to assign distinct colours to the remaining eight edges.

If this proves impossible, Theorem 3.5.4 again implies the existence of some subset W' of $E(G)$ for which it holds that

$$\left|\bigcup_{e\in W'} \tilde{A}(e)\right| < |W'|,$$

with W' containing either 6, 7, or 8 edges. An argument similar to the ones made above once again leads us to conclude that when there are still uncoloured edges, they can either be assigned distinct colours, or Theorem 3.5.4 implies that at least one opposite edge pair can be coloured in such a way as to leave distinct colours available for the remaining edges on the form f_i. ■

5.4 Graphs containing edge cuts of cardinality less than 4

Having now established the result for graphs containing n-gons with $n \leq 5$, we turn our attention to graphs that contain edge cuts of small cardinality. We define an edge cut of a graph G as a subset P of $E(G)$ such that $G - P = G_1 \cup G_2$, with G_1 and G_2 being disjoint subgraphs of G. Note the possible clash in terminology here: The definition of an edge cut often includes the condition that both disjoint subgraphs must be connected. We impose no such condition on the subgraphs G_1 and G_2 of G.

We also introduce the term *bigreedy except for k edges* to apply to a greedy algorithm which independently and greedily colours two disjoint subgraphs G_1 and G_2 of a graph, G, before permuting the colours assigned to one of these subgraphs, colouring G in this fashion, and finally colouring or re-colouring at most k edges to obtain the desired strong edge-colouring of G. If an algorithm is bigreedy except for 0 edges, we will just call it bigreedy. Considering G as the union of an empty graph with G itself shows that an algorithm which is greedy except for k edges is also bigreedy except for k edges.

5.4.1 Lemma:
Let G be a connected graph with $\Delta(G) \leq 3$. If G contains a bridge, then there exists a bigreedy edge colouring algorithm that strongly edge-colours G using at most 10 colours.

Proof:
Let $e \in E(G)$ be a bridge with end vertices v_1 and v_2. Denote the components of $G - e$ by G_1 and G_2, with $v_1 \in V(G_1)$. Lemmata 5.2.1 and 5.2.2 shows that both G_1 and G_2 can be strongly edge 10-coloured, and, moreover, that $\tilde{G}_1 = G_1 \cup \{v_2, e\}$ can be strongly edge 10-coloured. If necessary, permute the colours assigned to each colour class of G_2 so that the colour assigned to $e \in V(\tilde{G}_1)$ occurs nowhere in the set $N(e) \cap G_2$, and such that no colours assigned to edges incident to $v_1 \in V(G_1)$ are incident to $v_2 \in V(G_2)$. This yields a strong edge 10-colouring of G. ■

5.4.2 Lemma:
Let G be a connected graph with $\Delta(G) \leq 3$. If G contains an edge cut consisting of 2 edges, then there exists a bigreedy edge colouring algorithm that strongly edge-colours G using at most 10 colours.

Proof:
Assume that an edge cut of G contains two edges, e_1 and e_2, and denote by x_1, x_2 and y_1, y_2 the end vertices of e_1 and e_2, respectively. Since $\{e_1, e_2\}$ constitutes an edge cut, the graph $G - \{e_1, e_2\}$ consists of disjoint graphs, say G_1 and G_2 containing the vertices x_1, x_2 and y_1, y_2, respectively.

Now construct $\tilde{G}_1 = G_1 + \{e_1, e_2, x_2, y_2\}$ and strongly edge 10-colour $\tilde{G}_1$. This is possible by Lemma 5.2.2.

Denote by a and b the two distinct colours assigned to e_1 and e_2, respectively. Further, denote by c_1 and c_2 the incident edge colours of x_1, and by d_1 and d_2 the incident edge colours of y_1. Certainly, neither a nor b is identical to any of these four colours.

Now construct $\tilde{G}_2$ using G_2 by adding a vertex, say z, that is adjacent to both x_2 and y_2. Since z now has degree 2, we may again colour the graph $\tilde{G}_2$ greedily using the same colours used to colour $\tilde{G}_1$.

By substituting the colour of the edge assigned to the edge between z and x_2 and the edge between z and y_2 with a and b, respectively, we may also ensure that none of the 4 incident colours to x_2 and y_2 occur among the set of colours $\{a, b, c_1, c_2, d_1, d_2\}$. We may use as many as 10 colours, so this is clearly feasible.

Combining the colouring thus obtained of both $\tilde{G}_1$ and G_2 constitutes a strong edge 10-colouring of G. ■

Having established $s'(G) \leq 10$ for any cubic graph G containing edge cuts of size 1 or 2, we turn our attention to edge cuts of size 3. We call such an edge cut *trivial* if it consists of every edge adjacent to a single vertex, and non-trivial if this is not the case.

5.4.3 Lemma:
Let G be a connected graph with $\Delta(G) \leq 3$. If G contains a non-trivial edge cut consisting of 3 edges, then a strong edge 10-colouring of G may be constructed by an algorithm that is bigreedy except for 3 edges.

Proof:
Let $U = \{e_1, e_2, e_3\}$ be a non-trivial edge cut of G and denote by u_1, u_2, v_1, v_2, and w_1, w_2 the end vertices of e_1, e_2, and e_3, respectively. Since U is a cut, we have that $G - U$ consists of two disjoint graphs. We denote by G_1 the one containing u_1, v_1, w_1, and by G_2 the one containing u_2, v_2, w_2.

If any of these vertices are not distinct, an edge could be removed from U without violating the condition that it was an edge cut of G, implying by Lemma 5.4.2 that G is strongly edge 10-colourable. Hence, assume that the six vertices are all distinct. We also assume that every end vertex of an edge in U has degree 3 in G, since Lemmata 5.2.1 or 5.2.2 would otherwise imply the existence of a strong edge 10-colouring of G. Finally, we may assume that neither G_1 nor G_2 is a 3-gon, as this would imply the strong edge 10-colourability of G by 5.3.2. This implies that if a path contains exactly the vertices u_1, v_1, w_1 or u_2, v_2, w_2, then G would contain an edge cut containing exactly the edges between the pairwise endpoints of these two paths. Because of this, at least one pair of vertices among u_1, v_1, w_1 and u_2, v_2, w_2 in either graph of G_1 and G_2 must be nonadjacent. Assume without loss of generality that u_1 is not adjacent to v_1, and that u_2 is not adjacent to v_2. However, denote by f_1 an edge between u_1 and v_1, and similarly denote by f_2 an edge between u_2 and v_2.

Now strongly edge-colour the graphs $\tilde{G}_1 = G_1 + f_1$ and $\tilde{G}_2 = G_2 + f_2$ with the same set of 10 colours. Since w_1 and w_2 both have degree 2 in $\tilde{G}_1$ and $\tilde{G}_2$, respectively, Lemma 5.2.2 tells us that this is possible by initiating a greedy edge-colouring algorithm with the vertices w_1 and w_2, respectively.

If, in the process of obtaining such a strong edge 10-colouring, we do not assign any colours to the edges incident to u_1 or v_1, it must be due to the fact that e_3 constituted a bridge of G. Had this been the case, Lemma 5.4.1 would tell us that G could be strongly edge 10-coloured. Hence, we assume that this is not the case.

Denote by a the colour assigned to f_1, and denote by c a colour distinct from a that is not an incident edge colour of u_1, while still being available for assignment to any edge adjacent to w_1. At least one such colour exists by the proof of Lemma 5.2.2. Further, denote by b_1 and b_2 the two incident edge-colours of u_1 when considered in the graph G_1, and denote by b_3 and c or b_3 and b_4 the two incident edge-colours of v_1. Whichever pair of colours to consider depends on whether or not the colour c is an incident edge-colour of v_1, which again will depend on the order imposed on $E(G_1)$ before initiating the edge-colouring algorithm.

Under these assumptions, we have that the six colours a, b_1, b_2, b_3, b_4, and c are distinct.

Note, however, that we may only need to consider the five distinct colours a, b_1, b_2, b_3, and c.

To consider a set of at most 10 edge-colours, we name the at most 4 remaining colours contained in the colour set S in the following way:

$$S = \{a, b_1, b_2, b_3, b_4, c, d_1, d_2, d_3, d_4\}.$$

In the colouring of $\tilde{G_2}$ that we obtained earlier, we denote the colour of f_2 by a and assign the colour name c to some colour that is available for w_2, and which it not an edge colour incident to v_2. Further, denote by d_1 and d_2 or d_1 and c the colours that are edge incident to u_2 in G_2, depending on whether or not an incident edge of u_2 has been assigned the colour c in the previously obtained colouring of G_2. Finally, denote the edge incident colours of v_2 with d_3 and d_4.

Our objective now is to construct a colouring of G using the established strong edge 10-colourings of G_1 and G_2. The obvious way of doing this is to assign to e_1, e_2, and e_3 the colours a, a, and c, respectively. Whenever this is impossible, it is so because the colour a is an incident edge colour of both w_1 in G_1 and w_2 in G_2. We consider the two possible colourings of G for which this occurs.

Case 1: The colour a is an incident edge colour of either w_1 or w_2:
Suppose that two colours x and y are incident edge colours of w_1 and w_2, with both of them distinct from the colour a. If $x = b_i$ and $y = b_j$ for some $i, j \in \{1, 2, 3, 4\}$, rename the colours assigned to edges in G_2 by swapping the names of b_i and b_j with those of the two remaining colours b_k and b_l, where l and k must both be distinct from i and j. If, on the other hand, $x = d_i$ and $y = d_j$, again change the names of colours assigned to edges of G_2 by interchanging the pair of colours d_i, d_j with the remaining pair of colours, d_k, d_l. As before, we have that both i and j are distinct from both k and l. The remaining option is for x to be equal to a colour on the form b_i, while y is equal to a colour on the form d_j. If this happens, we simply swap b_i assigned to edges in G_2 with a distinct colour b_k, and swap d_j with some other colour on the form d_l.

Now, if the two edge colours incident to w_1 are denoted by x and y, with the incident edge colours of w_2 being denoted by x and z, we suppose that y is distinct from z, and that x is distinct from the colour a that was assigned to both e_1 and e_2. If x is equal to a colour on the form b_i or d_j, swap the names of colours assigned to edges of G_2, so that $x = b_i$ is renamed as some other colour b_k which is distinct from x, y, and z, or alternatively, such that $x = d_j$ is renamed as some other colour d_l that is distinct from x, y, and z.

Case 2: The colour a is an incident edge colour of both w_1 and w_2:
If this happens, denote by y the other incident edge colour of w_1, and similarly denote by z the other incident edge colour of w_2. Considering the edge colouring of G_1, choose some colour d that is available for assignment to incident edges of v_1, and which is also distinct from a, y, and c. This forces d to be in the set $\{b_1, b_2, d_2, d_3, d_4\}$, and we may assume that d is equal to either b_1 or d_4.

Considering now the edge colouring of G_2, let x be a colour available for assignment to incident edges of u_2, where x is distinct from z, a, and c. As above, this again lets us assume that x coincides with b_1 or d_4. This leaves us with four possible cases, to be considered as subcases of the second case, in which a is an incident edge colour of both w_1 and w_2.

Case 2.1: $d = b_1$ **and** $x = b_1$**:**
Considering the colouring of G_2, swap the names of a and b_1. Prior to the swap, b_1 was available for assignment to incident edges of u_2, and this property is transferred to colour a by the interchange. Similarly, the colour b_1 is available for assignment to incident edges of v_2 after the swap between a and b_1. Additionally, c is available for assignment to incident edges of w_2 as c is distinct from both a and b_1. Assign to e_1 the colour a, assign to e_2 the colour b_1, and assign to e_3 the colour c. This does indeed constitute a strong edge 10-colouring of G, unless $y = z$, in which case we swap the names of z and b_j, where j is neither i nor 1, if z were equal to some colour on the form b_i. If, on the other hand, z were equal to some colour on the form d_k, we swap the colour z with any other colour on the form d_l, with l distinct from k.

Case 2.2: $d = b_1$ **and** $x = d_4$**:**
Consider again the colouring obtained of G_2. Swap the names of d_4 and b_1. The incident edge of w_2 with the colour z may now have to have another colour assigned. To rectify the situation, set $\tilde{z}$ equal to z if a change is unnecessary, and let $\tilde{z}$ be equal to d_4 if a change is necessary. In either case, $\tilde{z}$ must be distinct from b_1.

Swap the names of the colours a and b_1, and assign to e_1 the colour a before assigning to e_2 the colour b_1 and the colour c to the edge e_3. We have only violated the properties of a strong edge 10-colouring if y is now equal to $\tilde{z}$, but if $\tilde{z}$ is equal to a colour on the form b_i, we may swap the names of $\tilde{z}$ with any other colour b_j, where j is distinct from both i and 1. If, on the other hand, colour $\tilde{z}$ is equal to d_k for some k, we swap the name of $\tilde{z}$ with the name of some colour on the form d_l, where we only demand that l be distinct from k.

Case 2.3: $d = d_4$ **and** $x = b_1$**:**
Considering the colouring of G_2, swap the names of d_4 and b_1. Again denote by $\tilde{z}$ whichever colour that is now assigned to the one previously coloured with z. That is, $\tilde{z}$ can be equal to either z or b_1, but must be distinct from d_4.

Now assign to e_1 the colour a, to e_2 the colour d_4, and to e_3 the colour c. The only problem arises if $y = \tilde{z}$, but if $\tilde{z}$ is equal to some colour on the form b_i with $i \neq 2$, then we swap the name of $\tilde{z}$ with the name of b_2, or if $\tilde{z}$ is equal to b_2, interchange the names of $\tilde{z}$ and b_3. Finally, if $\tilde{z}$ is equal to some colour described by d_k for some k, then we may swap $\tilde{z}$ with any colour d_l, where l is neither k nor 4.

Case 2.4: $d = d_4$ **and** $x = d_4$**:**
In this final case, we once again consider only the colouring obtained for G_2. We swap the names of colours between a and d_4, and assign to e_1 the colour a before assigning to e_2 the colour d_4 and to e_3 the colour c.

If we should happen to encounter a colour z that is identical to y, then if z is equal to some colour on the form b_i, we swap the name of z with b_j for some j distinct from i. Alternatively, if z is equal to d_k for some index k, we swap the name of z with d_l with any

l distinct from both k and 4.

Whatever case or subcase thereof we encounter, we can thus always construct a strong edge 10-colouring of G using first a bigreedy algorithm before manually assigning or possibly reassigning colours to at most 3 edges of G. ■

5.5 Graphs containing no n-gons with $n \leq 5$

With the previous results, Lemmata 5.3.1, 5.3.2, 5.3.3, and 5.3.4 established, we turn our attention to graphs that contain none of the characteristic substructures contained in the referenced lemmata. That is, we consider in this section only connected cubic graphs containing no smaller n-gons than 6-gons, unless specifically noted otherwise.

Let G be exactly such a graph and let x be a vertex of G. Let the vertices adjacent to x be the three distinct vertices u, v, and w. Further, denote by u_1 and u_2 the two other vertices adjacent to u, and similarly, denote by v_1 and v_2 and w_1 and w_2 the vertices adjacent to v and w, respectively. Since G contains no n-gon for $n \leq 5$, all of the vertices here described must be distinct, and further, the vertices $u_1, u_2, v_1, v_2, w_1, w_2$ must all be pairwise independent, that is, none of them are pairwise adjacent.

In the pursuit of a strong edge 10-colouring of G, we successively colour the edges uu_1, vv_1 and ww_1 with one colour, before colouring the edges uu_2, vv_2 and ww_2 with another colour.

Now denote by q_1 and q_2 the two vertices other than w that are adjacent to w_2, and consider the following sequence of edges represented by their end vertices:

$$s_0 = xw, xv, xu, ww_2, w_2q_1.$$

Let s be a sequence of uncoloured edges in G constructed by the following criteria. The sequence s is initiated by the edges in s_0, and for any edge $e \in s \setminus s_0$, at least three edges in $N(e)$ must precede it in the sequence s. Finally, construct s to be edge-maximal under these conditions.

5.5.1 Lemma:
If the sequence s, as constructed above, contains all edges of G left uncoloured by a partial strong edge-colouring, then a strong edge 10-colouring can be obtained by colouring the edges of $s \setminus s_0$ greedily in the reverse order, possibly re-assigning a colour to the edge ww_1, before colouring the edges of s_0 greedily in reverse order.

Proof:
Greedily colouring the edges of G in the reverse ordering imposed by the sequence s implies that, for any edge e of $s \setminus s_0$, we have $|F(e)| \leq 9$, so $|A(e)| \geq 1$, leaving a colour available to assign to e.

If a colour is still available for the edge encountered between vertices w_2 and q_1, we proceed without any problem. Indeed, by the construction of s, we will always have a colour available. Assume then, that no colour is available for the edge w_2q_1. Then $|F(w_2q_1)| = 10$, and all edges of $N(w_2q_1)$ have been assigned distinct colours. Whichever colour has been assigned to the edge ww_1, we remove it and assign it instead to the edge w_2q_1. Following this, ww_1 is once again coloured. This can be done without violating the need for a strong edge 10-colouring, as we have $|F(ww_1)| \leq 8$. In doing this, we may proceed to re-assign colours to the edges of s_0 in the reverse of the order imposed by the sequence s, as we will always have at least one such colour available. ■

In the following, let H be the subgraph of G induced by the uncoloured edges of G that are not contained in s. Assume in the following that $H \neq \emptyset$. Were this not the case, Lemma 5.5.1 would apply.

5.5.2 Lemma:
With the subgraph H of G and the edge sequence s constructed as above, it holds that no two vertices of H are connected by a single edge contained in s.

Proof:
For edges contained in s_0, the result is trivial. Suppose then that the result does not hold for an edge between vertices yy' contained in the sequence $s \setminus s_0$. Because both vertices y and y' by assumption will belong to H, there must be two uncoloured edges, say zy and $z'y'$, not contained in s. However, by the construction of s, at least three edges of $N(yy')$ must be contained in s, and at least one of the two sets $N(zy)$ and $N(z'y')$ must contain two of these. Assume without loss of generality that $N(zy)$ contains such two edges. Since $N(zy)$ also contains yy', it contains at least 3 edges of the sequence s, implying that it could be added to s, contradicting the edge-maximal property by which s was constructed. ■

5.5.3 Lemma:
Denote by $F = (G - H, G)$ the subset of edges in G with one end vertex in H and another end vertex in $G - H$. Then any edge in the set F is incident with one of the vertices u_1, u_2, v_1, v_2, w_1, and no other vertices than u and v are incident with two edges in F.

Proof:
A pre-coloured edge will be incident with some vertex in the set $\{u_1, u_2, v_1, v_2, w_1\}$, so choose instead some uncoloured edge e with end vertices g in $G - H$ and h in H. By construction of H, we have that e occurs on the sequence s with some uncoloured edge between h and h' that is not in s. Hence, by construction of s, there must be an edge incident to g that is not in the sequence s. This shows that g has an incident edge that is already coloured. Since we assumed that g could not be u, v, or w, then g must be identical to some element in the set $\{u_1, u_2, v_1, v_2, w_1\}$.

To show that u and v are the only vertices which may be incident to more than one edge in F, we begin by assuming that some vertex z belonging to the set $\{u_1, u_2, v_1, v_2, w_1\}$ is incident with two edges of F. If any of these two edges are already coloured, we must have $z \in H$, implying that some uncoloured edge of F joins z to another vertex $\tilde{z} \in G - H$. This would again imply that all incident edges of $\tilde{z}$ would be contained in the sequence s, a contradiction in the assumption that the third incident edge of z would not be contained in s. Hence, the two uncoloured edges of z, here denoted by e_1 and e_2, must be contained in F. Then no incident edge of z is contained in H, implaying that z itself must be a vertex of $G-H$. Since we have that neither e_1 nor e_2 are contained in s_0, we may assume without restriction that e_1 is ordered before e_2 in the edge-maximal sequence s.

However, this would mean that three edges in the set $N(e_1)$ appears before e_1 in the sequence s. Considering one such edge of $N(e_1)$ along with the edges e_1 and e_2 shows that all three edges with an end-vertex distinct from z of either e_1 or e_2 must be in the sequence s, contradicting our previous assumption that both e_1 and e_2 must be edges contained in F.

Since w cannot be incident with two edges of F either, we are done if we can show that no vertex z that is not contained in $\{u_1, u_2, v_1, v_2, w_1\}$ can be incident with two edges in F.

Were this the case, both incident edges of z contained in F would join the vertex z to some vertex in the set $\{u_1, u_2, v_1, v_2, w_1\}$, implying that z should be a vertex of H. If this were true, the sequence s would not be edge-maximal, as it could be extended to contain whatever third edge that was incident to z. ■

We now introduce the term *small cut* to describe any non-trivial edge-cut of a graph that contains $1, 2$, or 3 edges.

5.5.4 Lemma:
Either the graph G contains a small cut, or it may be described by one of the fifteen illustrations included in Appendix A.

Proof:
If the edge set F constructed as in the above contains fewer than four elements, it is a small cut. Hence, we assume that $|F| \geq 4$, implying that F contains either 4 or 5 elements.

Case 1: Edges uu_1 and uu_2 are both contained in F

If this configuration occurs, the graph $H + \{u, u_1, u_2\}$ is joined to the remainder of the graph only by the edge set F and the single edge between x and u. Hence, a small cut exists unless F contains five edges that are incident with v_1, v_2, and w_1, beyond the edge joining u to u_1 and the edge joining u to u_2. If this is the case, we can exclude the possibility that vv_1, vv_2, or ww_1 belongs to F, as this would imply the existence of a small cut that would separate the remainder of the graph from the subgraph $G - (H + \{x, u, v, v_1, v_2, w, w_1\})$. The only possible remaining configuration in which F contains the edges uu_1 and uu_2, and where G does not contain a small cut, is depicted here:

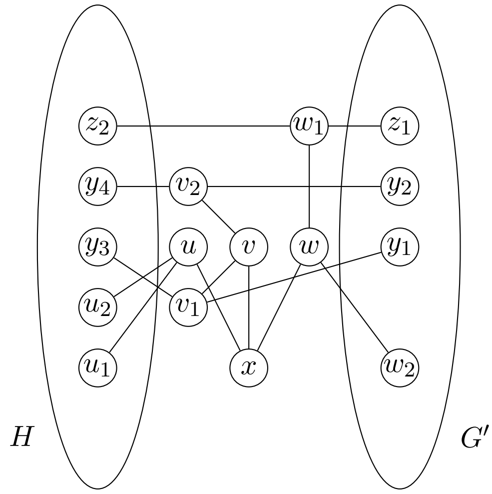

Figure 1. Case 1.

Because of this, we may now and for the remainder of the proof assume that F does not contain both uu_1 and uu_2, with a symmetrical argument showing that F cannot at the same time contain both vv_1 and vv_2.

Cases 2 and 3: Edges uu_1 and vv_1 are both contained in F
When this configuration arises, we have that u_2 or v_2 is incident to some edge contained in F. We suppose without loss of generality that u_2 is incident to some edge in F, and denote this edge by its end vertices u_2t_2. If F also contains v_2, the graph G contains a small cut whenever w_1 is not incident with some edge in F that is not ww_1. This would result in the following case:

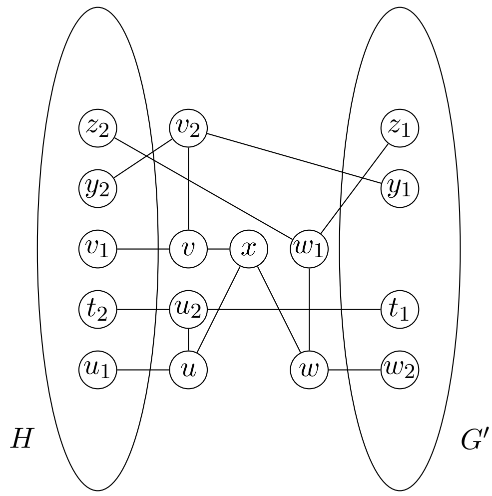

Figure 2. Case 2.

If, on the other hand, v_2 is not an incident edge of any edge in F, then w_1 must be incident to an edge in F that is not ww_1. If this is the case, we observe the third case:

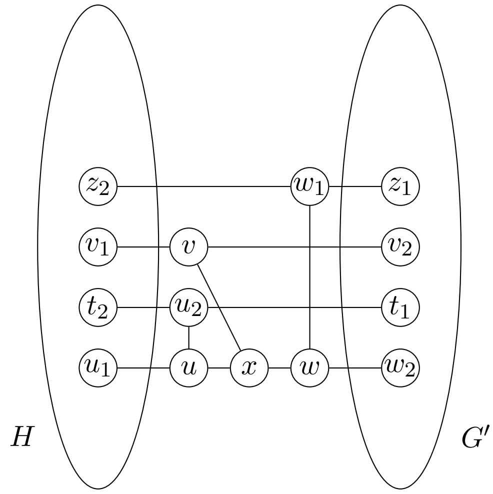

Figure 3. Case 3.

In the following, we assume that only one edge uu_1 of the set $\{uu_1, uu_2, vv_1, vv_2\}$ is contained in F.

Cases $4, 5$, and 6: Edges uu_1 and ww_1 are both contained in F
If F also contains ww_1, then F must also contain edges incident to at least two of the vertices u_2, v_1, v_2. If edges of F are incident with all three vertices, we obtain this description of G:

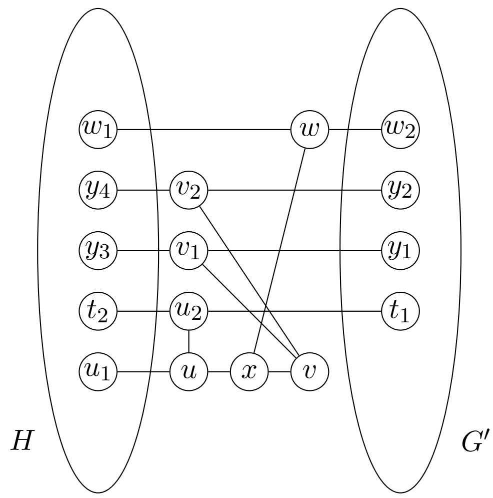

Figure 4. Case 4.

If no edge of F is incident to u_2, we conclude that it must be possible to depict G like this:

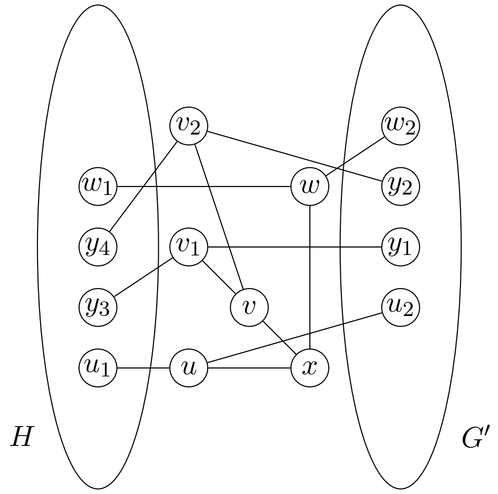

Figure 5. Case 5.

If, on the other hand, the set F contains no edges incident with v_1 or v_2, we are left with the following in the case of v_1:

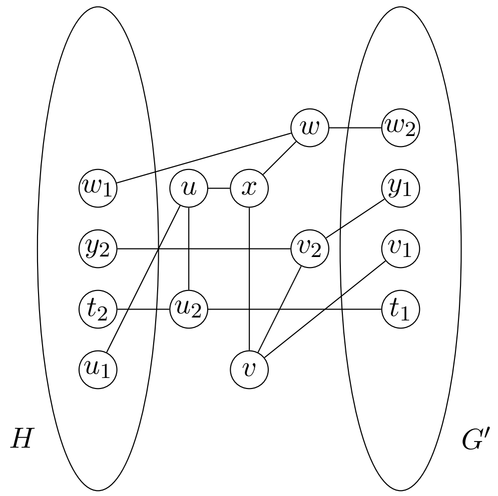

Figure 6. Case 6.

A similar description would be possible if F did not contain any edges incident with v_2.

Cases $7, 8, 9$, and 10: Edges uu_1 is in F, while ww_1 is not in F
If F contains exactly five elements, we can describe G as such:

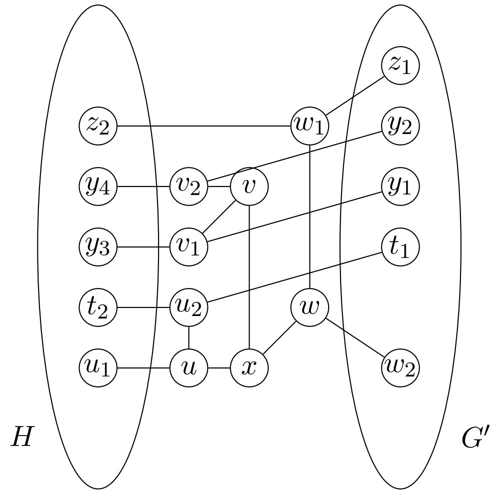

Figure 7. Case 7.

Alternatively, F contains four elements. When this happens, three possible descriptions of G arise when u_2, w_1, or v_1 or v_2 are not incident to any edge in F. These are listed respectively as the following figures, with only the case of v_1 not being incident to an edge of F:

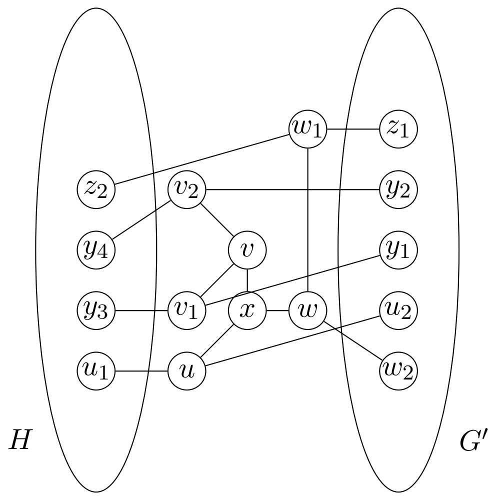

Figure 8. Case 8.

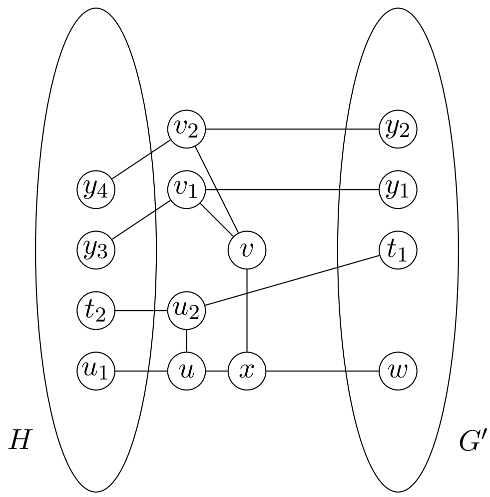

Figure 9. Case 9.

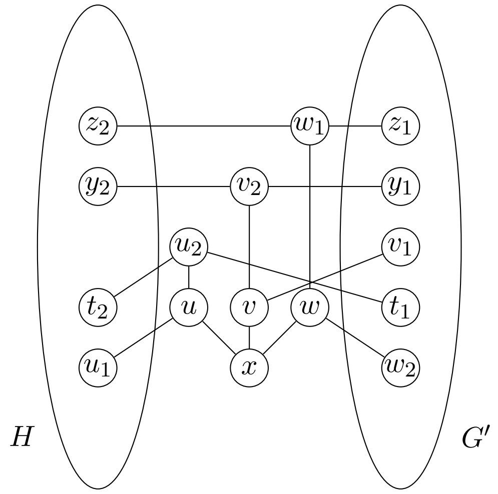

Figure 10. Case 10.

A description similar to that of the case when v_1 is not incident to any edge in F would apply if v_2 were the vertex not incident to any edge found in F. The description of G would be similar to that of Figure 10.

Cases 11 and 12: Edge ww_1 is in F, while uu_1 is not
When F contains exactly five elements, we arrive at this configuration:

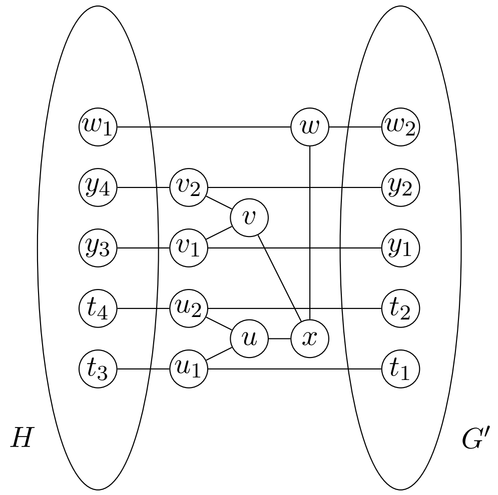

Figure 11. Case 11.

If this is not the case, F must contain only four elements. We can assume that u_1 is the vertex not incident to any edge of F, giving us the following description of G:

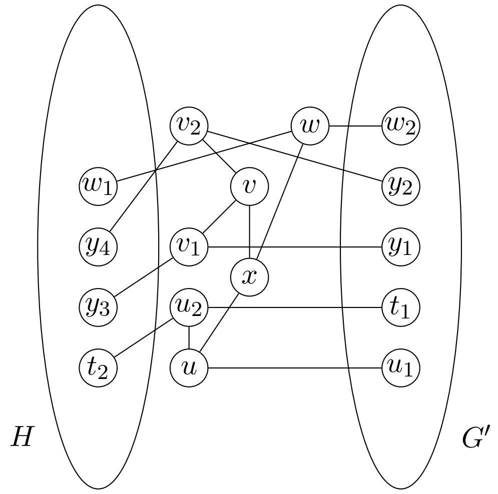

Figure 12. Case 12.

Cases $13, 14$**, and** 15**: No edges of** F **have already been coloured**
Two cases here, when $|F| = 5$, we have Case 13 as pictured here:

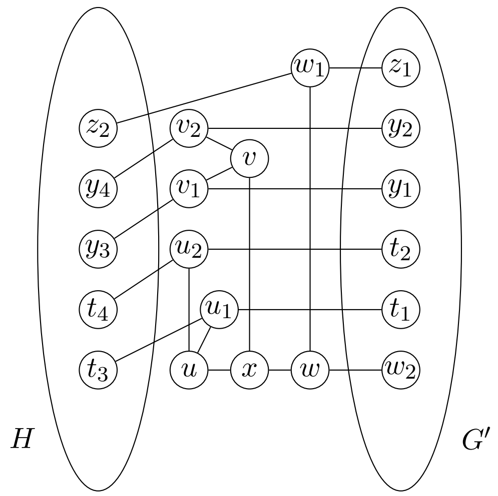

Figure 13. Case 13.

If, on the other hand, $|F| = 4$, we have only two asymmetrical distinct cases remaining:

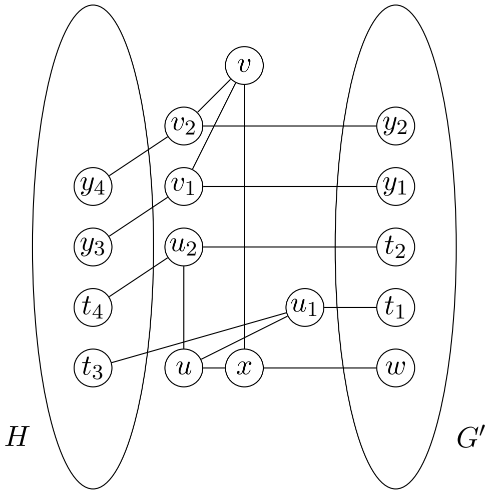

Figure 14. Case 14.

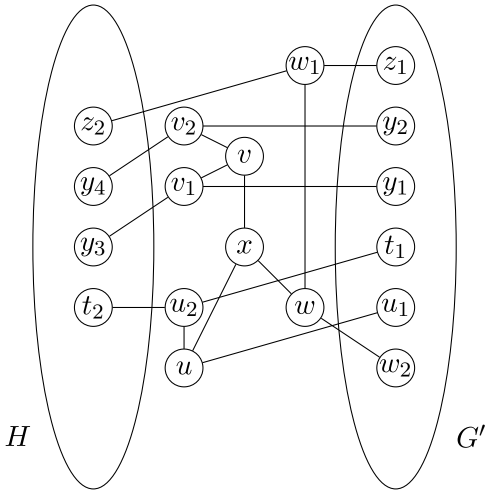

Figure 15. Case 15.

Having this exhausted the distinct compositions of the set F, we consider the proof complete. ■

5.6 The main result

The process of reducing a general problem to a finite set of distinct problem instances is no new strategy. To take one example, the strategy was also employed in the proof of the Four-Colour Theorem contained in [Appel et al., 1989]. While the fifteen cases of Lemma 5.5.4 may seem cumbersome to manually verify, it is still quite preferable to the 1476 distinct configurations encountered in the proof of the Four-Colour Theorem.

5.6.1 Theorem:
Let G be a graph with $\Delta(G) \leq 3$. Then a strong edge 10-colouring of G can be produced by an edge-colouring algorithm that is bigreedy except for 21 edges.

Proof:
If G contains a vertex of degree 1 or 2, Lemmata 5.2.1 and 5.2.2 tells us that we are done. Hence, assume that every vertex has degree 3.

If G contains a multiple edge pair between two vertices, or indeed a 3-gon, 4-gon, or 5-gon, Lemmata 5.3.1, 5.3.2, 5.3.3, and 5.3.4 again implies that we are done. Hence, assume that G contains no n-gon with $n \leq 5$.

If G contains a small cut, Lemmata 5.4.1, 5.4.2, and 5.4.3 again shows us that we are done.

Finally, Lemmata 5.5.1 through 5.5.4 establish the convenient fact that only fifteen distinct descriptions are possible for a cubic graph that is assumed to possess none of the previously mentioned properties. Hence, we will have proven the result if we can show how to strongly edge 10-colour the edge set of all fifteen distinct cases. We refer to the graph construction and notation described throughout Section 5.5, specifically referring to the cases in the order that they are described in Lemma 5.5.4.

Case 1**:**
In this case, the four vertices w, y_1, y_2, and z_1 are all distinct, and so we may construct a graph $G_g = G' + g$, where g is some vertex adjacent to vertices w_2, y_1, and y_2. Then, z_1 will have degree 2 in G_g, and by Lemma 5.2.2, we can strongly edge 10-colour G_g. Denote by a and b the colours assigned to edges gy_1 and gy_2, respectively. Then, if a is not an incident edge colour of z_1, the colour c distinct from both a and b must be incident to z.

In similar fashion, construct the graph $H_g = H + \{u, h, uu_1, uu_2\}$, and let h be a vertex adjacent to y_3, y_4, and y. Again, H_g may be strongly edge 10-coloured because u has degree 2.

Procedurally rename colours assigned to edges of H_g in such a way as to colour the edge hy_4 with a and hy_3 with b. Further, ensure that if a is not an incident edge colour of z_2, then the colour c must be.

Now, a strong edge 10-colouring of G may be obtained by choosing the colour assignments to G_g and H_g and applying them to the edges of G' and $H + \{u, uu_1, uu_2\}$, discarding the colours assigned to incident edges of g and h in G_g and H_g, respectively. We are not done, since we still need to assign the colour a to edges v_1y_1, v_2y_4, and xw, assign the colour b to edges v_2y_2 and v_1y_3, and assigning colours greedily to the remaining edges in the following sequential order:

$$w_1z_1, w_1z_2, ww_2, ww_1, xu, xv, xv_1, xv_2.$$

Every edge e of this sequence will have a colour available at the time of assignment, since we have either coloured at most nine edges of the set $N(e)$ or because we have repeated the colour assigned to some edge of $N(e)$ sufficiently often to leave it available. The algorithm is bigreedy on G_g and H_g, and leaves at most 13 edges uncoloured at the time of greedy termination.

Throughout the remaining cases, we assume that graphs constructed as G_g and H_g are both connected. Were they not, G would contain a small cut, in which case we would be done, as we have already argued.

Case 2**:**
Assume that the four vertices w_2, t_1, y_1, and z_1 are all distinct. Were some pair of these vertices identical, G would contain a small cut, a contradiction in our assumptions. Construct a subgraph $H_g = H + \{u_1z_2, t_2y_2\}$, and strongly edge 10-colour this greedily. Denote by a and b the colours that occur at z_2 in H. Here, b is chosen as some colour that is not an incident edge colour of u_1. Denote by c and d two distinct colours, both distinct from a and b, such that either b or c is an incident edge colour of t_2 in H, while at the same time either b or d is an incident edge colour of y_2 in H.

Construct $G_g = G' + \{t_1y_1\}$, and strongly edge 10-colour G_g. Name these colours such that both a and b are incident edge colours of z_1, and such that either b or c is an incident edge colour of t_1 in G', while either b or d is an incident edge colour of y_1, also in G'.

Now, we can strongly edge 10-colour G by considering the colourings assigned to H and G' and assigning to ux the colour b before greedily colouring the remaining edges in the following sequential order:

$$t_2u_2, t_1u_2, uu_1, uu_2, v_2y_2, v_2y_1, vv_1, vv_2, xv, xw, ww_2, w_1z_1, w_1z_2, ww_1.$$

Case 3:
Here, we again assume vertices to be distinct in order to avoid constructing a small cut. These are w_2, t_1, v_2, and z_1. Were they not so, we could construct a small cut by considering $t_1 = z_1$.

Construct again H_g, this time by adding some vertex h to H before adding the three edges hu_1, hv_1, and hz_2. This may be strongly edge 10-coloured greedily, and we choose colour names such that hu_1 is assigned the colour a, while hv_1 is assigned the colour b. Let some colour c be distinct from both a and b, and let c be such that either a or c is an incident edge colour of t_2. Further, let d, another colour distinct from both a, b, and c, be an incident edge colour of z_2 in H.

Construct $G_g = G' + \{w_2v_2\}$ and strongly edge 10-colour G_g greedily. Here, name or rename colours such that the edge w_2v_2 has been assigned the colour a, with a distinct colour b chosen to not be incident to v_2 or z_1, or indeed any vertices adjacent to z_1. Then, let c be some colour, also distinct from both a and b, such that either a or c is an incident edge colour of t_1, and let d be another colour distinct from both a and c, such that either a or d is an incident edge colour of z_1. By construction, we have that d is distinct from b, hence we can consider the colourings of H and G' thus obtained in order to strongly edge 10-colour G. We assign to uu_1, ww_2, and vv_2 the colour a, and assign to vv_1 and w_1z_1 the colour b. Finally, we greedily colour the remaining edges of G with respect to the following order:

$$w_1z_2, ww_1, u_2t_2, u_2t_1, uu_2, xu, xv, xw.$$

Case 4:
Again we assume the uniqueness of four vertices, given that we would otherwise obtain a small cut of G. In this case, we consider t_1, y_1, y_2, w_2 as distinct. Construct H_g by adding the vertex h to H before adding these three edges: hy_3, hy_4, hw_1. Now, strongly edge 10-colour H_g such that hy_3 is assigned the colour a, such that hy_4 is assigned the colour b which is distinct from a, and such that c is some colour distinct from both a and b where it holds that either a or c is an incident edge colour of u_1, while a fourth colour, d, is distinct from c and an incident edge colour of w_1 in H.

Construct G_g by adding to G' the vertex g and the edges gy_1, gy_2, and gw_2. Now, strongly edge 10-colour G_g and assign or possibly reassign names to the employed colours, such that gy_1 is assigned the colour b, while gy_2 is assigned the colour a. Further, name or possibly rename the employed colours so that c is some colour incident to t_1 if and only if a is not an incident edge colour of t_1. Further, let d be a colour distinct from c that is an incident edge colour of w_2 in G'.

We may now colour G by using the colour assignments obtained in the colourings of G' and H, before assigning to v_1y_3, v_2y_2, and xw the colour a, and assigning to v_1y_1 and v_2y_4 the colour b. Finally, we are left with edges which we may greedily colour with respect to the following order:

$$ww_1, ww_2, u_2t_2, u_2t_1, uu_1, uu_1, xu, vv_1, vv_2, xv.$$

Case 5**:**
Here, the vertices u_2, y_1, y_2, w_2 are all distinct, so we construct H_g from H by adding the vertex h and the edges $hy_3, hy_4,$ and hw_1. We can now strongly edge 10-colour H_g such that hy_3 is assigned the colour a, such that hy_4 is assigned the colour b, and such that c is a colour distinct from both a and b that c or a is an incident edge colour of u_1. Finally, denote by d some colour distinct from c and let d be an incident edge colour of w_1 in H.

Similarly, construct G_g by adding to G' the vertex g before adding the three edges $gy_1, gy_2,$ and gw_2. Here, similarly to the above, we assign the colour b to gy_1, the colour a to gy_2, and the colour a or c to an incident edge of u_2, with d finally being assigned to w_2 in G'. Now, we need only perform the following assignments: assign colour a to $v_1y_3, v_2y_2,$ and xw, colour b to v_1y_1 and v_2y_4, before greedily colouring the remaining edges in the following order:

$$ww_1, ww_2, uu_1, uu_2, xu, vv_1, vv_2, xv.$$

Case 6**:**
The construction of a strong edge 10-colouring of this graph is similar to that of Case 3 with the vertex v interchanged with w.

Case 7**:**
For this configuration, any number of the following equalities may occur: $t_1 = y_1, t_1 = y_2, t_1 = z_1, y_1 = z_1$, or $y_2 = z_1$. Assume without loss of generality that when any two of the equalities are met, t_1 is equal to either y_1 or y_2, while z_1 is equal to the other vertex among y_1 and y_2. Then, G' may also be assumed to take the shape of a path containing only the vertices $t_1, w_2,$ and z_1 in that order, since G would otherwise contain a small cut. If any two of these three vertices on the path G' are one and the same, we would have a vertex of degree 1, in which case a single colour would suffice to colour G'.

Several symmetrical cases arise, but we simply strongly edge 10-colour one of them. Construct H_g by adding to H the vertex h and the edges $hu_1, hy_3,$ and hy_4. This may be strongly edge 10-coloured, which we proceed to do with the following colour names: Colour a is assigned to hy_3, colour b is assigned to hy_4, and if either colour is an incident edge colour of t_2, we may assume that only colour a is. Then, let c be some colour distinct from a and assign c to an incident edge of z_2 if a is not already an incident edge colour of z_2. Further, let d be some colour distinct from both $a, b,$ and c, such that d is an incident edge colour of t_2 if a is not.

Now construct G_g from G' by adding the vertex g to G' before further adding the edges $gy_1, gy_2,$ and gz_1 if z_1 is distinct from both y_1 and y_2. Strongly edge 10-colour G_g, choosing colour names such that b is assigned to gy_1 and a is assigned to gy_2. Additionally, let d be some colour distinct from both a and b, such that d is an incident edge colour of t_1 in G' if a is not. Finally, if z_1 is not equal to either t_1 or y_2, let c be some colour distinct from d that is an incident edge colour of z_1 in G'. If z_1 were equal to y_1, and d was an incident edge colour of z_1, then that particular edge would have to be reassigned a colour c, distinct from both a, b, and d. This would be readily possible since at least four colours would be available for assignment to any edge incident to a vertex of degree 1. Additionally, if z_1 were equal to y_1, and z_1 were adjacent to t_1, we would have constructed a small cut of G, so we may in the following assume that the edge to which colour d was assigned can be recoloured without any effect on the edge colour incidence of d on t_1.

We can now construct a strong edge 10-colouring of G by taking the colours assigned to edges of G' and H, assigning to edges v_1y_3, v_2y_2 the colour a, and to edges v_2y_4, v_1y_1 the colour b, before finally assigning colours greedily to the remaining edges in the following order:

$$w_1z_1, w_1z_2, ww_1, ww_1, u_2t_2, u_2t_1, uu_1, uu_2, xw, vv_1, vv_2, xv.$$

The sequence consists of 17 edges to be coloured after the termination of the bigreedy algorithm.

Case 8**:**
We may encounter that z_1 equals y_1 or y_2. We are essentially in a situation identical to that of Case 4, only where H and G' have switched places. Since we could allow for t_2 to coincide with y_3 or y_4 in Case 4, symmetrical arguments tell us that we are done.

Case 9**:**
This is equivalent to Case 1, where we again must swap the subgraph G' with the subgraph H that has had the vertex u and the edges uu_1 and uu_2 added. In the arguments establishing the existence of a strong edge 10-colour of the graph in Case 1, we could allow for t_1 to coincide with y_1 or y_2, but doing so would create a small cut of G, so we have sufficiently argued in Case 1 for the strong edge 10-colourability of the graphs that this case contains.

Case 10**:**
Here, we may observe any of the following equalities: $t_1 = y_1$, $t_1 = z_1$, and $y_1 = z_1$. If none of the equalities are attained, we have reduced the configuration to that of Case 2 with H and G' swapped, implying that we are done.

Hence, assume that one of the equalities are attained. Since the first and second equality result in symmetrical descriptions of graphs, we need only consider the first and third equality being attained. Assume that $t_1 = y_1$. Then we can colour H with a greedy algorithm to obtain a strong edge 10-colouring. Name the colours assigned to $E(H)$ such that a and b are incident edge colours of z_2, letting c be some colour distinct from both a and b such that b or c is an incident edge colour of t_2.

Now construct $G_g = G' + \{v_1z_1\}$. Colour G_g greedily to obtain a strong edge 10-colouring, naming the assigned colours in such a way as to ensure that a and b are incident edge colours of z_1, while b is not an incident edge colour of v_1. Then, reassign colours such that c is an incident edge colour of t_1. Assign to the edge xv in G the colour b and greedily colour the remaining edges in the following order:

$$vv_1, v_2y_2, vv_2, t_1v_2, t_1u_2, u_2t_2, uu_1, uu_2, ux, xw, ww_2, w_1z_2, w_1z_1, ww_1.$$

If, on the other hand, we proceed with the assumption that y_1 coincides with z_1, we can greedily colour H and assign colour names to obtain a and b as incident edge colours of t_2, while letting b or c be an incident edge colour of y_2. Now, greedily colour G' with the edge t_1w_2 added to obtain a strong edge 10-colouring of this graph. Name or rename colours to obtain a and b as incident edge colours of t_1 in G', with b being an absent edge colour of w_2. Then, ensure that the incident edge colour of y_1 is named c. Finally, we colour G by considering the colours assigned to edges of G' and H, assigning the colour b to xw, before greedily colouring the remaining edges in the following order:

$$ww_2, w_1z_2, ww_1, w_1y_1, y_1v_2, v_2y_2, vv_1, vv_2, vx, xu, uu_1, u_2t_2, u_2t_1, uu_2.$$

Case 11**:**
Here, we must consider one of t_1 and t_2 as potentially coinciding with one of y_1 and y_2. If two pairs both coincide, the subgraph G' is a path graph, and as such, vertex w_2 in G' will have degree 2.

Construct H_g by adding to H the vertex h and the edges hy_3, hy_4, and hw_1. Colour H_g greedily to obtain a strong edge 10-colouring of H_g. Assign to hy_3 the colour a, assign to hy_4 the colour b, and denote by c some colour distinct from both a and b that is an incident edge colour of t_4 if a is not. Finally, let d be a colour distinct from c, such that d is an incident edge colour of w_1 in H.

Construct G_g by adding to G' the vertex g and the edges gy_1, gy_2, and gw_2. Colour G_g greedily to obtain a strong edge 10-colouring, and name the assigned colours such that gy_1 is coloured by b, such that gy_2 is coloured by a, and such that c, a colour distinct from both a and b, is an incident edge colour of t_2 in G' if a is not. Finally, let d be a colour distinct from c that is an incident edge colour of w_2 in G'.

Consider the edge colourings of H and G' as partial colourings of G, and assign to edges v_1y_3, v_2y_2, and xw the colour a, before assigning to edges v_1y_1 and v_2y_4 the colour b. Finally, colour the remaining edges of G greedily with respect to the following order:

$$u_1t_1, u_1t_3, u_2t_2, u_2t_4, uu_1, uu_2, ww_2, ww_1, xu, xv, vv_1, vv_2.$$

This single construction covers every possible combination of equalities in the vertex pairs y_1, y_2 and t_1, t_2.

Case 12**:**
We have already argued that this description of G should be strongly edge 10-colourable in Case 8.

Case 13**:**
Here we consider the possibilities that t_1 or t_2 coincide with one of y_1, y_2, or z_1, that z_1 coincides with y_1 or y_2, or the case where several of these equalities occur simultaneously. If three vertices coincide, we have a small cut of G, so we assume to the contrary that no three vertices coincide. Then, without loss of generality, we may assume that z_1 and t_2 are distinct, and argue for this case in a way that confirms the existence of a strong edge 10-colouring of G in any symmetrical case.

We begin by constructing G_g by adding to G' a vertex g and the edges gw_2, gwy_1, gy_2, and z_1t_2. Obtain a strong edge 10-colouring of G_g using a greedy algorithm, and denote by a the colour assigned to gy_2, denote by b the colour assigned to gy_1, and let c and d be two colours distinct from both a and b such that d is an incident edge colour of z_1 in G' if a is not, and such that c is an incident edge colour of t_2 in G' if a is not. The colours c and d may be chosen since both z_1 and t_2 are vertices of degree 2 in G', and if one of them has degree 1 in G', the edge incident to that vertex of degree 1 can be coloured by at least four distinct colours. This was argued in the proof of Lemma 5.2.1.

Now construct H_g by adding to H one vertex h and the edges hy_3, hy_4, and hz_2. Obtain a strong edge 10-colouring of H_g by a greedy algorithm and assign colour names such that hy_3 is assigned the colour a, such that hy_4 is assigned the colour b, and such that c is a colour distinct from both a and b such that a or c is an incident edge colour of t_4 in H. Finally, choose some colour d distinct from c, such that d is an incident edge colour of z_2 in H.

A partial edge 10-colouring of G may be obtained by considering the colours assigned to edges of G' and H, before assigning the colour a to the edges v_1y_3, v_2y_2, and xw, assigning the colour b to the edges v_1y_1 and v_2y_4, and finally colouring the remaining edges greedily with respect to the following order:

$$w_1z_1, w_1z_2, ww_2, ww_1, u_2t_2, u_2t_4, u_1t_1, u_1t_3, uu_1, uu_2, ux, vv_1, vv_2, xv.$$

This is the case that incurs the highest number of edges for which our bigreedy algorithm is excempt. Indeed, after the bigreedy algorithm has terminated, we still need to colour or recolour 21 edges.

Case 14**:**

Here, we consider only that t_1 or t_2 might coincide with y_1 or y_2. If two vertex pairs coincide, G' is a path graph containing only two edges, so we assume that no two vertex pairs coincide. Without loss of generality, we may assume that the equality thus attained is $t_1 = y_2$. Similar arguments apply to every other possible configuration.

Construct H_g by adding to H a vertex h and the edges ht_4, hy_3, and hy_4. Greedily obtain a strong edge 10-colouring of H_g, such that colour a is assigned to hy_4, such that colour b is assigned to hy_3, and such that c is a colour distinct from both a and b, where c is an incident edge colour of t_3 if a is not. Finally, denote by d some colour distinct from c that is an incident edge colour of t_4 in H.

Now construct G_g by adding to G' a vertex g and the edges gy_1 and gy_2. If t_2 and y_2 do not coincide, add the edge gt_2 to G' to finish the construction of G_g. Obtain a strong edge 10-colouring of G_g with a greedy algorithm, naming colours such that a is assigned to gy_1, such that b is assigned to gy_2, and such that a or c is an incident edge colour of t_1 in G'. Finally, ensure that d is an incident edge colour of t_2 in G'.

We obtain a strong edge 10-colouring of G by considering the colours assigned to edges of G' and H, assigning to v_2y_4, v_1y_1, and uu_2 the colour a, assigning to v_1y_3 and v_2y_2 the colour b, before finally colouring the remaining edges greedily in the following order:

$$u_2t_2, u_2t_4, u_1t_1, u_1t_3, uu_1, ux, xw, vv_1, vv_2, xv.$$

Case 15**:**

Here we have to consider the vertex t_1 coinciding with y_1, y_2, or z_1, and the vertex z_1

coinciding with y_1 or y_2. If three such vertices coincide, G contains a small cut, so we assume that at most two pairs coincide.

Construct H_g by adding to H a vertex h and the edges hy_3, hy_4, and hz_2. Obtain a strong edge 10-colouring of H_g by a greedy algorithm, naming colours to ensure that colour a is assigned to hy_4, that colour b is assigned to hy_3, that c is a colour distinct from a and b such that a or c is an incident edge colour of t_2, and such that d is a colour distinct from c such that d is an incident edge colour of z_2 in H.

Now construct G_g by adding to G the vertex g and the edges gw_2, gy_1, and gy_2. Further, if t_1 does not coincide with z_1, add to G' the edge t_1z_1 to finish constructing G_g. Obtain a strong edge 10-colouring of G_g using a greedy algorithm and ensure that colour a is assigned to gy_1, that colour b is assigned to gy_2, and that d and c are two colours such that a or c is an incident edge colour of t_1 in G', while a or d is an incident edge colour of z_1 in G'.

Obtain a strong edge 10-colouring of G by considering the colours assigned to edges of G' and H, assigning to edges v_1y_1, v_2y_4, and xw the colour a, assigning to edges v_1y_3 and v_2y_2 the colour b, before finally colouring the remaining edges greedily with respect to the following order:

$$w_1z_2, w_1z_1, ww_2, ww_1, u_2t_1, u_2t_2, uu_1, uu_2, ux, vv_1, vv_2, xv$$

As we have now constructed a strong edge 10-colouring of G for the fifteen possible descriptions of a connected, cubic graph that contains no n-gon for $n \leq 5$, the proof is finished. ■

Taking $\Delta = 3$ in the statement of Conjecture 4.0.4 shows that Theorem 5.6.1 confirms the conjecture for the case of graphs with $\Delta(G) = 3$.

Chapter 6
Potential future efforts

Having confirmed the statement of Conjecture 4.0.4 with $\Delta(G) = 3$, it would be interesting to see what strategy might confirm the conjecture for $\Delta(G) = 4$, if indeed the conjecture is even true for $\Delta(G) = 4$. Obviously, we need only find one integer for which the conjectured bound does not hold to show that the conjecture fails, but we choose to be optimistic in our speculation, and suggest that a strategy similar to that of [Andersen, 1992] could prove fruitful in confirming Conjecture 4.0.4. Quite like the fashion in which the author of that article confirmed the conjecture for $\Delta(G) = 3$, it may be possible to find some relatively small integer k where, for every integer less than or equal to k, the confirmation of Conjecture 4.0.4 is tedious and must be handled by individual cases, but for every integer greater than k, the conjecture would be confirmed by relying on smaller problem instances.

These smaller problem instances might themselves require considerable work, because, as we have seen, even the smallest non-trivial case required quite a bit of manual verification and construction. With recent progress in processing speeds of computers, it may at that point be prudent to attempt a brute-force electronical approach, if the problem of confirming Conjecture 4.0.4 for $\Delta(G) = 4, 5, \ldots$ could be reduced to a finite set of distinct configurations. Whatever the case, the conjecture remains open at the time of writing.

There would appear to be some progress on the problem of confirming or denying the conjectured bound of $s'(G) \leq \Delta(X)\Delta(Y)$ on the strong chromatic index of a bipartite graph G with vertex partition $V(G) = X \cup Y$. The result of [Nakprasit, 2008] was published as recently as 2008. As a measure of progress towards a confirmation of the general bipartite bound, it is not very different from the work contained in [Andersen, 1992]. Indeed, if we choose again to speculate optimistically, then a confirmation of the conjecture could come from first confirming the case for $\Delta(X) = 3$, $\Delta(X) = 4$, and so on, up to some fixed integer k, after which a general solution would build on the previously established results. We already have confirmation in the case of $\Delta(X) = 2$.

Bibliography

Andersen, **Lars Døvling** (1992). 'The strong chromatic index of a cubic graph is at most 10'. In: *Journal of Discrete Mathematics* 108, pp. 231–252.

Appel, **Kenneth and Wolfgang Haken** (1989). *Every Planar Map is Four Colorable*. American Mathematical Society. ISBN: 0-8218-5103-9.

Chartrand, **Gary and Ping Zhang** (2009). *Chromatic Graph Theory*. Chapman and Hall. ISBN: 978-1-58488-800-0.

Cormen, **Thomas H.** et al. (2009). *Introduction to Algorithms*. 3rd ed. The MIT Press. ISBN: 978-0-262-53305-8.

Czygrinow, **Andrzej** (2014). 'Tutte's Theorem'. In: URL: `https://math.la.asu.edu/~andrzej/teach/mat416/proofs3.pdf` (visited on 20/02/2017).

Dębski, **Michał** (2015). 'Strong Chromatic Index of Graphs'. English. PhD Dissertation. Faculty of Mathematics and Computer Science, Jagiellonian University.

Faudree, **R. J.** et al. (1989). 'Induced Matchings in Bipartite Graphs'. In: *Discrete Mathematics* 78, pp. 83–87.

Faudree, **R. J.** et al. (1990). 'The Strong Chromatic Index of Graphs'. In: *Ars Combinatoria* 29B, pp. 205–211.

Greene, **Joshua Evan** (2013). 'Vizing's Theorem'. In: URL: `https://www2.bc.edu/joshua-e-greene/MT445F13/Vizing.pdf` (visited on 16/02/2017).

Gross, **Jonathan L. and Jay Yellen** (2006). *Graph Theory and its Applications*. 2nd ed. Chapman & Hall. ISBN: 978-1-58488-505-4.

Hall, **Philip** (1935). 'On Representatives of Subsets'. In: *The Journal of the London Mathematical Society* 10, pp. 26–30.

Hudák, **Dávid** et al. (2013). 'Strong edge colorings of planar graphs'. In: URL: `https://arxiv.org/pdf/1303.4580v2.pdf` (visited on 24/03/2017).

König, **Dénes** (1916). 'Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre'. In: *Mathematische Annalen* 77, pp. 453–465. URL: http://gdz.sub.uni-goettingen.de/dms/resolveppn/?PPN=GDZPPN002266539 (visited on 09/02/2017).

Nakprasit, **K.** (2008). 'A note on the strong chromatic index of bipartite graphs'. In: *Discrete Mathematics* 308 (16), 3726–3728.

Steger, **Angelika and Min li Yu** (1993). 'On induced matchings'. In: *Discrete Mathematics* 120, pp. 291–295.

Tutte, **William S.** (1947). 'The factorization of linear graphs'. In: *The Journal of the London Mathematical Society* 22, pp. 107–111.

Appendix A

Descriptions of G in Lemma 5.5.4

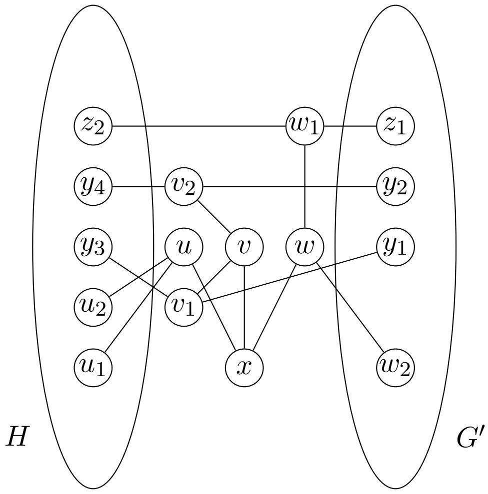

Figure 1. Case 1.

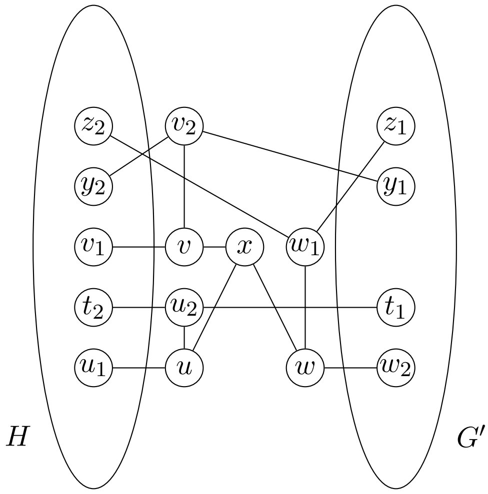

Figure 2. Case 2.

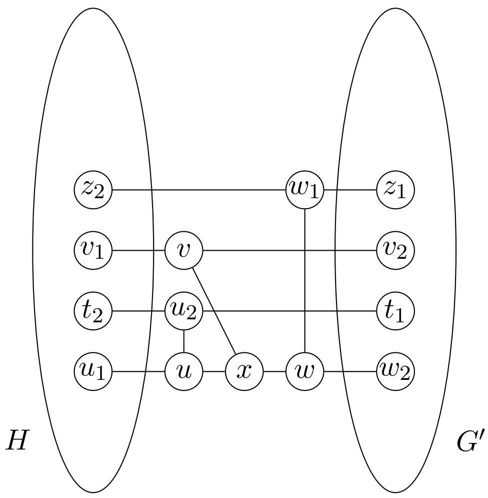

Figure 3. Case 3.

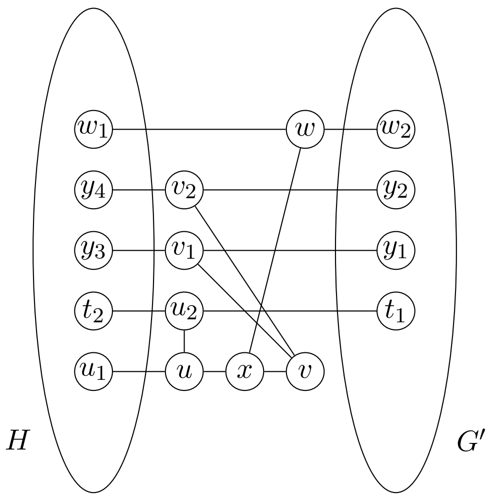

Figure 4. Case 4.

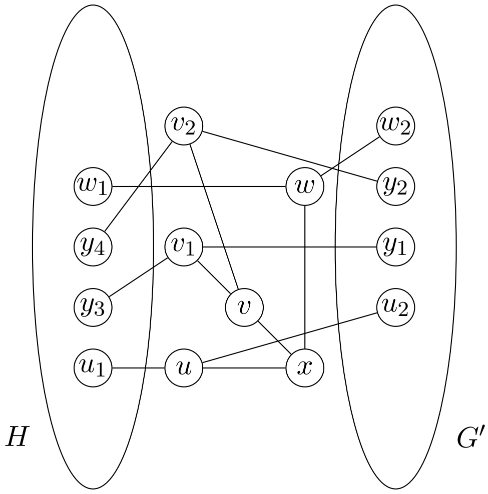

Figure 5. Case 5.

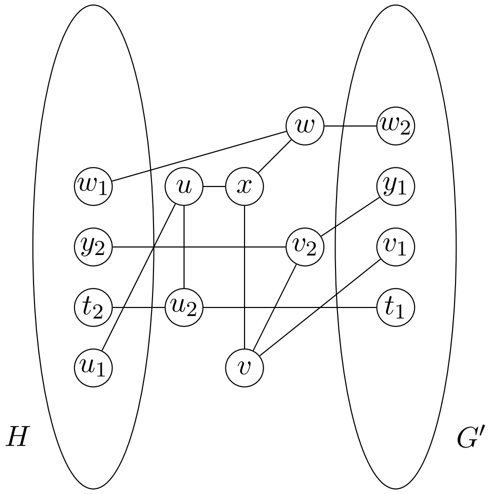

Figure 6. Case 6.

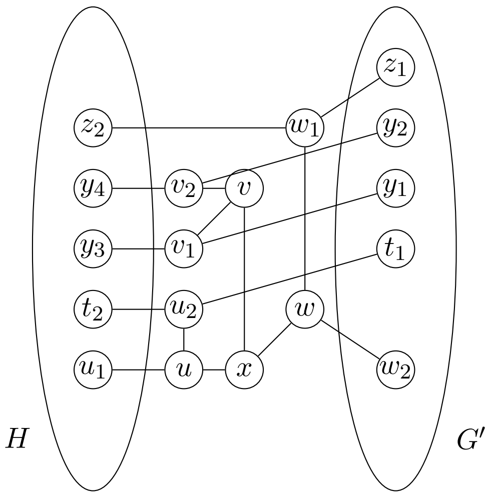

Figure 7. Case 7.

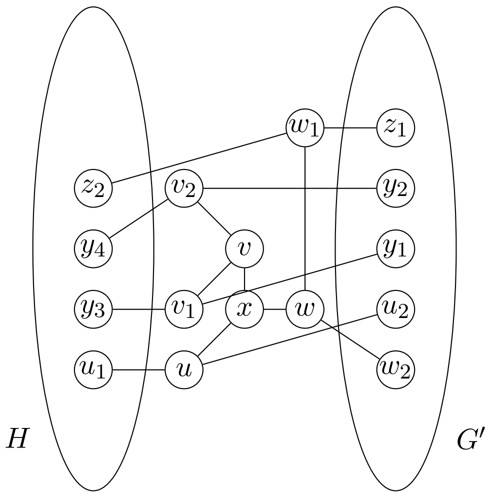

Figure 8. Case 8.

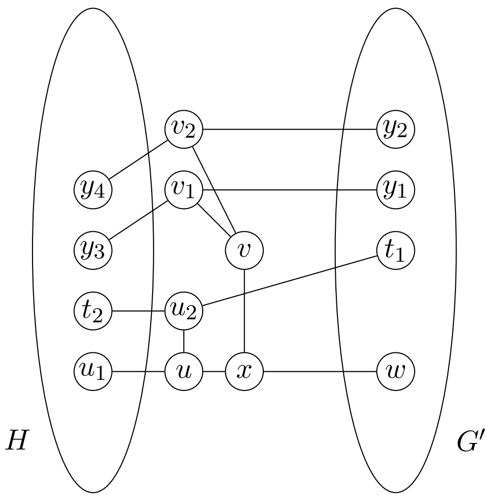

Figure 9. Case 9.

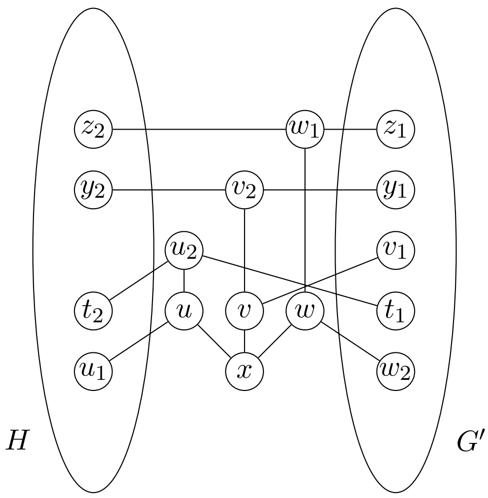

Figure 10. Case 10.

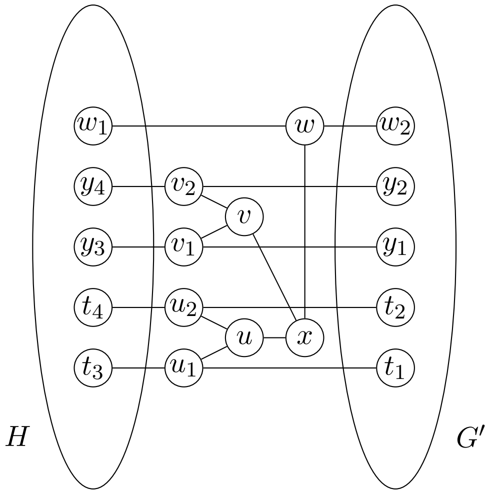

Figure 11. Case 11.

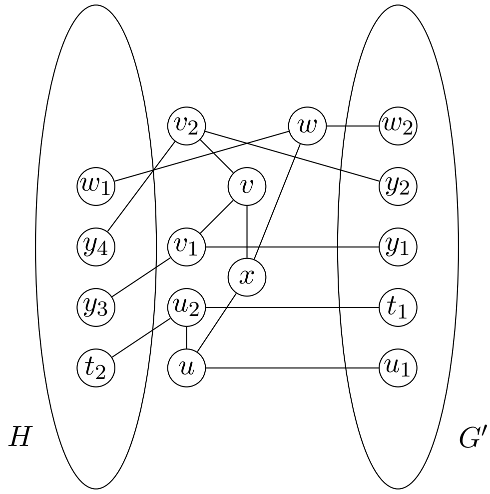

Figure 12. Case 12.

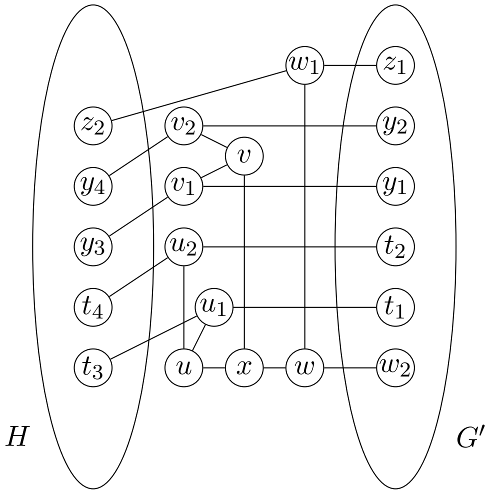

Figure 13. Case 13.

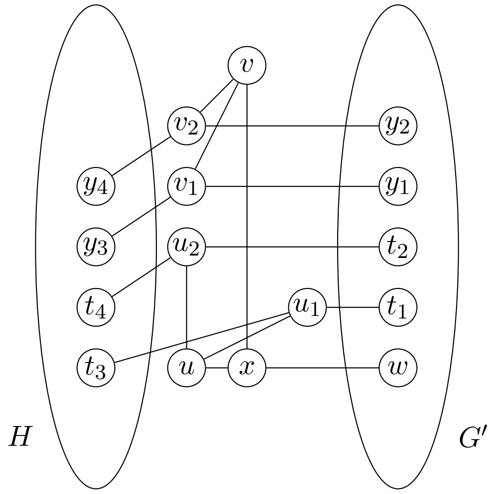

Figure 14. Case 14.

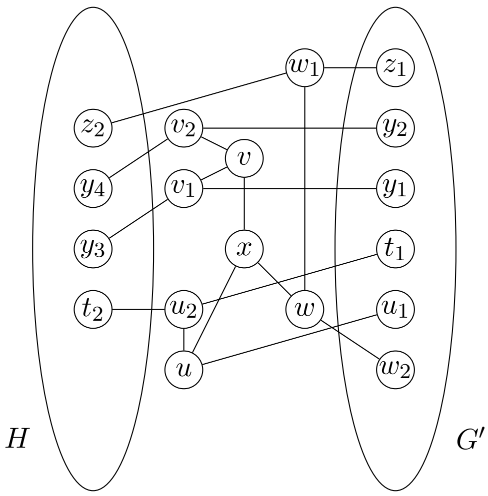

Figure 15. Case 15.