uOttawa

L'Université canadienne
Canada's university

The Vertex-Switching Reconstruction Problem

Patrick Niesink

Thesis Submitted to the Faculty of Graduate and Postdoctoral Studies
In partial fulfilment of the requirements for the degree of Master of Science in Mathematics [1]

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

[1] The M.Sc. program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Abstract

Switching on a vertex of a graph involves swapping the sets of neighbours and non-neighbours of the vertex. The resultant graph is called a switch card of the original graph. The switch deck of a graph is the collection of all of its switch cards. The vertex-switch reconstruction problem then asks which graphs (termed non-VSR graphs) cannot be uniquely determined from their switch decks. A review of the published knowledge about this problem is followed by an improved bound on the number of edges in a non-VSR graph, and a bound on the size of the automorphism group of a non-VSR graph. Finally, the results of a computer search are presented, showing that no non-VSR graphs of order 8 or 12 exist.

Acknowledgements

For many years, I worked on this problem as a hobby. Finally turning this hobby into something useful is the (partial) realization of a long-standing dream. It would not have been possible without the formal mathematical training provided by my professors, who are too numerous to mention here. I thank every one of them for the amazing amount of effort they put into sometimes seemingly thankless job of teaching each course. They have each made a difference.

Many thanks to my supervisors, Mateja Šajna and Lucia Moura, for their extremely helpful guidance, advice, collaboration, and financial support. Thanks also to Jérôme Lefebvre, Robert Bailey, Mike Newman, Jason Gao, Brendan McKay, Andrea Burgess, Shonda Gosselin, Elizabeth Maltais, Maria Lewanski, and Johanna Coutts for their comments, suggestions, inspiration, and moral support.

I also deeply thank Bill Kocay for changing my life by introducing me to graph theory as well as to the vertex-switching reconstruction problem.

Finally, I thank my parents for many things, but particularly for inculcating me with the worth of education and academic pursuits.

Dedication

For Jim Ferguson, who always set a good example.

Contents

List of Figures

List of Tables

Chapter 1

The Vertex-Switching Reconstruction Problem

Before we can describe the vertex-switching reconstruction problem, we need a few definitions of graph theory terms, and a few definitions of concepts specific to this problem.

1.1 Definitions

Definition 1.1.1 A *graph* G is a pair $(V(G), E(G))$, where $V(G)$ is a nonempty set of *vertices* and $E(G)$ is a (possibly empty) set of *edges*, and $E(G) \subseteq \{\{v, w\} : v \neq w \text{ and } v, w \in V(G)\}$. Note that in the literature this is typically termed a "simple graph". The *order* of a graph G, denoted ν_G, is $|V(G)|$. The *size* of a graph G, denoted ε_G, is $|E(G)|$. Two vertices v and w comprising an edge e of a graph G are *joined* by e, and are called the *endpoints* of e. Two vertices of a graph G are *adjacent* if they are joined by some edge of G. An edge e is *incident* with a vertex v if v is an endpoint of e. The *neighbours* of a vertex v of G are the vertices adjacent to v in G. The set of neighbours of v in G is denoted $N_G(v)$. The *non-neighbours* of a vertex v

of G are the elements of $\overline{N}_G(v) = V(G) \setminus (N_G(v) \cup \{v\})$. The *degree* of a vertex v of G (written $d_G(v)$) is the number of neighbours of v in G, i.e. $d_G(v) = |N_G(v)|$.

Definition 1.1.2 The *edge function* of a graph G is a map from $V(G) \times V(G)$ to $\{0, 1\}$, and is denoted e_G. The edge function maps a pair of vertices (v, w) to 1 if v and w are adjacent in G, and to 0 otherwise.

Where the graph in question is understood, ε may be written for ε_G, and ν may be written for ν_G. For brevity, we may refer to the edge $\{v, w\}$ as vw.

We now define some terms related to the action of mapping graphs onto other graphs.

Definition 1.1.3 If G and H are graphs, an *isomorphism* from G to H is a bijective mapping $\psi : V(G) \to V(H)$ such that for all $u, v \in V(G)$, we have that $e_G(u, v) = e_H(\psi(u), \psi(v))$. If an isomorphism exists between graphs G and H, then G and H are *isomorphic*, denoted $G \cong H$.

It should be noted that isomorphic graphs are not necessarily equal. If two graphs are equal, they have the same vertex set and the same edge set. If they are isomorphic, however, this means only that it is possible to relabel the vertices of one graph (and to relabel its edges correspondingly) so that the result is equal to the other graph. In other words, all graphs dealt with in this thesis are *labelled* graphs.

Definition 1.1.4 The *isomorphism class* of a graph G (denoted $\Xi(G)$) is the set of all graphs with vertex set $V(G)$ that are isomorphic to G.

Note that since graph isomorphism is an equivalence relation, the isomorphism classes of the set of all graphs of a given order partition this set.

Definition 1.1.5 An *automorphism* of a graph G is an isomorphism that maps G onto itself. The set of all automorphisms of a graph G forms a group under composition, called the *automorphism group* of G and denoted $\mathrm{Aut}(G)$.

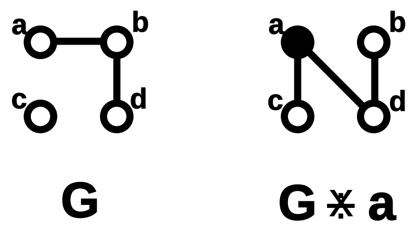

Figure 1.1: A graph and its switch card with respect to one vertex

Note that any graph G admits the identity mapping $\varphi : V(G) \to V(G)$, defined by $\varphi(v) = v$ for all $v \in V(G)$, as an automorphism.

Finally, we define some terms related to vertex switching, and then introduce the vertex-switching reconstruction problem.

Definition 1.1.6 A graph H is the *switch card* of a graph G with respect to the vertex v of G if $V(G) = V(H)$ and $E(H) = (E(G) \backslash \{ uv : u \in N_G(v) \}) \cup \{ uv : u \in \overline{N}_G(v) \}$. The switch card of G with respect to v is denoted $G \divideontimes v$. The operation of creating $G \divideontimes v$ is called *switching on v in G*.

In other words, switching on a vertex of a graph has the effect of removing all edges incident with the vertex and joining the vertex to all vertices to which it was formerly non-adjacent. For example, Figure 1.1 shows the effect of switching on vertex a in the given graph.

We now prove the intuitive result that switching on the same vertex twice has no effect.

Lemma 1.1.7 *Let G be a graph and let v be a vertex of G. Then $(G \divideontimes v) \divideontimes v = G$.*

Proof: Consider any pair of distinct vertices x, y of G. If $x = v$ or $y = v$ then $e_{G \divideontimes v}(x, y) = 1 - e_G(x, y)$, since switching on v removes edge xy if it exists in G, and adds it otherwise. Then, similarly, $e_{(G \divideontimes v) \divideontimes v}(x, y) = 1 - e_{G \divideontimes v}(x, y) = e_G(x, y)$. Now suppose $x \neq v$ and $y \neq v$. Then $e_{G \circledast v}(x,y) = e_G(x,y)$ since adjacency of two vertices neither of which is v is unaffected by switching on v. Similarly, $e_{(G \circledast v) \circledast v}(x,y) = e_{G \circledast v}(x,y) = e_G(x,y)$. Therefore in all cases, $e_{(G \circledast v) \circledast v}(x,y) = e_G(x,y)$, and so $G = (G \circledast v) \circledast v$. ∎

Definition 1.1.8 Two vertices v and w of a graph G are *vertex-switch similar* (or *VS-similar* for short) if $G \circledast v \cong G \circledast w$.

In other words, the switch cards of G with respect to two of its VS-similar vertices are isomorphic.

Next, we see that the result of switching on a set of vertices is independent of the order in which the vertices are switched on.

Lemma 1.1.9 *Let G be a graph and let $u, v \in V(G)$. Then $(G \circledast u) \circledast v = (G \circledast v) \circledast u$.*

Proof: First, suppose that $u = v$. Then by Lemma 1.1.7, $(G \circledast u) \circledast v = (G \circledast v) \circledast u$. Now suppose $u \neq v$, and let x and y be any two distinct vertices of G. Suppose that exactly one of x and y is in $\{u, v\}$. Without loss of generality, we may assume that $x = u$ and $y \neq v$. Then

$$e_{(G \circledast u) \circledast v}(x,y) = e_{G \circledast u}(x,y) = 1 - e_G(x,y),$$

and

$$e_{(G \circledast v) \circledast u}(x,y) = 1 - e_{G \circledast v}(x,y) = 1 - e_G(x,y).$$

If $x = u$ and $y = v$, then

$$e_{(G \circledast u) \circledast v}(x,y) = 1 - e_{G \circledast u}(x,y) = 1 - (1 - e_G(x,y)) = e_G(x,y),$$

and

$$e_{(G \circledast v) \circledast u}(x,y) = 1 - e_{G \circledast v}(x,y) = 1 - (1 - e_G(x,y)) = e_G(x,y).$$

Otherwise (when $x \notin \{u, v\}$ and $y \notin \{u, v\}$),

$$e_{(G \circledast u) \circledast v}(x, y) = e_{G \circledast u}(x, y) = e_G(x, y),$$

and

$$e_{(G \circledast v) \circledast u}(x, y) = e_{G \circledast v}(x, y) = e_G(x, y).$$

Therefore in all cases $e_{(G \circledast u) \circledast v}(x, y) = e_{(G \circledast v) \circledast u}(x, y)$, and so $(G \circledast u) \circledast v = (G \circledast v) \circledast u$. ∎

Definition 1.1.10 A *multiset* M is a set S_M together with a mapping $\mu_M : S_M \to \mathbb{N} \setminus \{0\}$. The elements x of the underlying set S_M are called the *elements* of M. If the elements of a multiset M are listed as $a_1, a_2, \ldots, a_n$ (where the number of occurrences of each element of M in the list is equal to its multiplicity in M, given by $\mu_M(\cdot)$), then we may denote M as $\langle a_1, a_2, \ldots, a_n \rangle$. The *cardinality* of a multiset M, denoted $|M|$, is the sum of the multiplicities of the elements of M. Two multisets A and B are *equal* if $S_A = S_B$ and $\mu_A(x) = \mu_B(x)$ for all $x \in S_A$. The *union* of two multisets A and B (denoted $A \uplus B$) is defined by $S_{A \uplus B} = S_A \cup S_B$, and

$$\mu_{A \uplus B}(x) = \begin{cases} \mu_A(x) + \mu_B(x) & \text{if } x \in S_A \cup S_B \\ \mu_A(x) & \text{if } x \in S_A \setminus S_B \\ \mu_B(x) & \text{if } x \in S_B \setminus S_A. \end{cases}$$

The *difference* of two multisets A and B (denoted $A \setminus B$) is defined by $S_{A \setminus B} = (S_A \setminus S_B) \cup \{\, x \in S_A \cap S_B : \mu_A(x) > \mu_B(x) \,\}$ and for all $x \in S_{A \setminus B}$, by

$$\mu_{A \setminus B}(x) = \begin{cases} \mu_A(x) - \mu_B(x) & \text{if } x \in S_B \\ \mu_A(x) & \text{otherwise.} \end{cases}$$

Lemma 1.1.11 *Let G be a graph and let M be a multiset whose elements are vertices of G. Then switching on all of the vertices of M in G produces the same graph regardless of the order in which the vertices are switched on in G.*

Proof: Let k be the cardinality of M, and let $A = (a_1, a_2, \ldots, a_k)$ be some ordering of the elements of M (with multiplicities). Note that the elements of A can be rearranged into any desired order by a succession of swaps of consecutive elements of A (i.e., by interchanging the positions of a_i and a_{i+1} for some $i \in \{1, 2, \ldots, k-1\}$). We therefore need only show that the effect of switching on A in G is not affected by a single such swap.

Let i be an integer with $1 \leq i < k$. Lemma 1.1.9 shows that $(((G \divideontimes a_1) \divideontimes a_2) \divideontimes \ldots \divideontimes a_i) \divideontimes a_{i+1} = (((G \divideontimes a_1) \divideontimes a_2) \divideontimes \ldots \divideontimes a_{i+1}) \divideontimes a_i$. It follows that $(((((G \divideontimes a_1) \divideontimes a_2) \divideontimes \ldots \divideontimes a_i) \divideontimes a_{i+1}) \divideontimes a_{i+2}) \divideontimes \ldots \divideontimes a_k = (((((G \divideontimes a_1) \divideontimes a_2) \divideontimes \ldots \divideontimes a_{i+1}) \divideontimes a_i) \divideontimes a_{i+2}) \divideontimes \ldots \divideontimes a_k$. This shows that swapping the order of two consecutive elements of an ordered multiset of vertices does not affect the result of switching on the multiset, and therefore the order in which the vertices of a multiset are switched on in a graph does not affect the graph which results from these switches. ∎

The result of the previous lemma means that for any multiset M of vertices of G we may economically write $G \divideontimes M$ to describe the graph produced by switching on all of the vertices of the multiset M in the graph G.

Lemma 1.1.12 *Let G be a graph, let M be a multiset whose elements are vertices of G, and let M' be a multiset with $S_{M'} = \{\, v : \mu_M(v) \equiv 1 \pmod 2 \,\}$ and $\mu_{M'}(v) = 1$ for all $v \in S_{M'}$. Then $G \divideontimes M = G \divideontimes M'$.*

Proof: Let v be some element of M with $\mu_M(v) > 1$. (If no such element exists, then $M = M'$ and we are done.) Then order the elements of M so that if $s \in M$ appears after a v, then $s = v$, i.e. so that the ordering is of the form $(s_1, s_2, \ldots, s_k, v, \ldots, v)$ where $s_i \neq v$ for all $i \in \{1, 2, \ldots, k\}$. Lemma 1.1.11 shows that ordering the elements of M has no effect on $G \divideontimes M$. Now let $M_v = (s_1, s_2, \ldots, s_k, v)$ if $\mu_M(v)$ is odd, or $(s_1, s_2, \ldots, s_k)$ if $\mu_M(v)$ is even. Repeated applications of Lemma 1.1.7 show that $G \divideontimes M = G \divideontimes M_v$. In this way, we can remove all pairs of elements of the form (v, v) from M, eventually producing M', with $S_{M'} = \{ v : \mu_M(v) \equiv 1 \pmod 2 \}$ and $\mu_{M'}(v) = 1$ for all $v \in V(G)$. ∎

Note that, as a consequence of the previous lemma, the set of graphs obtained by switching on all possible multisets of vertices of a graph of order n has maximum cardinality 2^n.

We now define a concept that is central to this thesis.

Definition 1.1.13 The *switch deck* of a graph G, denoted $\mathrm{SD}(G)$, is the multiset composed of all of the switch cards of G, i.e. $\mathrm{SD}(G) = \langle G \divideontimes v : v \in V(G) \rangle$. The switch deck of a multiset of graphs $\mathcal{G}$, denoted $\mathrm{SD}(\mathcal{G})$, is the multiset composed of all of the switch cards of all of the graphs which compose $\mathcal{G}$, that is, for all $X \in \mathrm{SD}(\mathcal{G})$, we have $\mu_{\mathrm{SD}(\mathcal{G})}(X) = \displaystyle\sum_{G : X \in \mathrm{SD}(G)} \mu_{\mathrm{SD}(G)}(X) \cdot \mu_{\mathcal{G}}(G)$. The switch decks of two graphs G_1 and G_2 are *isomorphic* (denoted $\mathrm{SD}(G_1) \cong \mathrm{SD}(G_2)$) if there is a bijective mapping $\theta : \mathrm{SD}(G_1) \to \mathrm{SD}(G_2)$ such that $\theta(C_1) \cong C_1$ for all $C_1 \in \mathrm{SD}(G_1)$. If k is a positive integer, then the *k-switch deck* of a multiset of graphs $\mathcal{G}$, denoted $\mathrm{SD}_k(\mathcal{G})$, is the multiset consisting of all graphs produced, for each $G \in S_{\mathcal{G}}$, and each k-vertex subset of $V(G)$, by switching on all of the vertices of this subset in G and taking $\mu_{\mathcal{G}}(G)$ copies of each such graph. Finally, we define $\mathrm{SD}_0(\mathcal{G}) = \mathcal{G}$ for all multisets $\mathcal{G}$ of graphs.

Definition 1.1.14 Two graphs G_1 and G_2 are *vertex-switch equivalent* (*VSE* for short) if $V(G_1) = V(G_2)$ and $\mathrm{SD}(G_1) \cong \mathrm{SD}(G_2)$. Furthermore G_1 and G_2 are *s-vertex-switch equivalent* (*s-VSE* for short) if $V(G_1) = V(G_2)$ and $\mathrm{SD}_s(G_1) \cong \mathrm{SD}_s(G_2)$.

Definition 1.1.15 A graph G is *vertex-switch reconstructible* (or *VSR* for short) if, for any graph H which is vertex-switch equivalent to G, we have $H \cong G$. A graph property P is *vertex-switch reconstructible* (VSR for short) if, for any pair of VSE graphs G and H, the property has the same value (i.e., $P(G) = P(H)$). A graph G is *s-vertex-switch reconstructible* if for any graph H which is s-VSE to G, we have

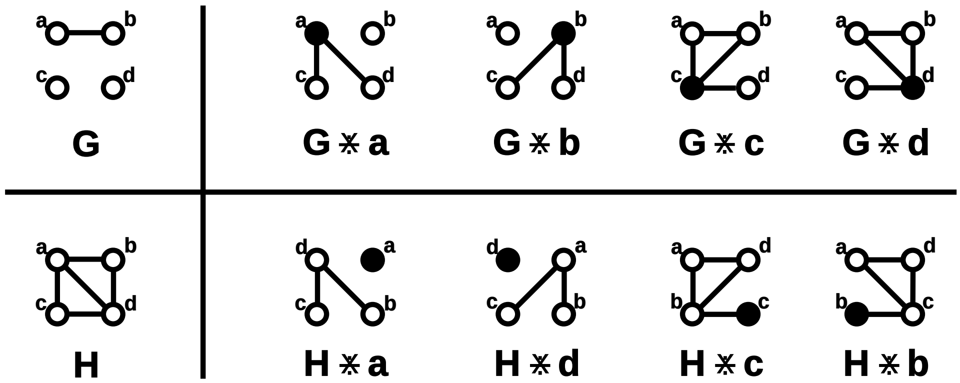

Figure 1.2: Two vertex-switch equivalent graphs

$H \cong G$.

This means that if a graph G is not VSR, there is some graph H with $V(G) = V(H)$ and $G \ncong H$ such that $\mathrm{SD}(G) \cong \mathrm{SD}(H)$. If a graph property P is not VSR, then there are two graphs G and H that are VSE but $P(G) \neq P(H)$.

Example 1.1.16 Figure 1.2 shows two graphs that have isomorphic switch decks. Note that, since $G \circledast a$ and $G \circledast b$ are isomorphic, vertices a and b of G are vertex-switch similar. The isomorphism between $\mathrm{SD}(G)$ and $\mathrm{SD}(H)$ is implied in Figure 1.2 by the ordering of the switch cards in $\mathrm{SD}(H)$, and by the labelling of the vertices of these switch cards. Since G and H have isomorphic switch decks, and $V(G) = V(H)$, they are therefore VSE, and since $G \ncong H$, neither G nor H is VSR.

1.2 Introduction to the Thesis

The vertex-switching reconstruction problem asks for a characterisation of the non-VSR graphs. Currently, the only known non-VSR graphs have four vertices. All 9 non-VSR graphs of order 4, as well as their switch decks, are listed in the first four rows of the table in Figure 1.3. Although many classes of graphs have been shown to be VSR, the general problem is still open. This thesis reviews the published knowledge about the vertex-switching reconstruction problem, and presents some new related results. A number of very different approaches to this problem have yielded a variety of results. The contents of this thesis are arranged according to the approach used in attacking the problem.

The question of which graphs are not VSR was first posed by Stanley in 1985 [14]. He showed that if a graph of order n is not VSR, then $n \equiv 0 \pmod 4$. In Chapter 2 we present the proof of this result, which uses an approach based on linear algebra.

In Chapter 3, we present a proof, due to Krasikov [6], showing that every vertex of a non-VSR graph G with more than 4 vertices must have an associated distinct vertex called an "H-switch partner", for any graph H that is VSE to G but not isomorphic to G. When we switch on both v and its H-switch partner in G, the result is a graph isomorphic to H. This then leads to some restrictions on the structure of a non-VSR graph G of order > 4; for example, G must be connected and cannot be regular.

Chapter 4 then establishes some properties of a non-VSR graph with more than 4 vertices. In particular, we present an upper and lower bound on the number of edges, an inequality relating the maximum and minimum degrees of the graph, and an upper bound on the order of the automorphism group of the graph.

This is followed, in Chapter 5, by a proof that any two graphs of order $n > 4$ with the same switch deck have the same number of induced subgraphs isomorphic to any given graph of order less than $\frac{n}{2}$. This result is then reshaped into a set of results that can be used to speed up an exhaustive search for a non-VSR graph.

Finally, in Chapter 6 we discuss the results of an exhaustive search for a non-VSE graph of the two smallest previously-unsolved orders, 8 and 12. This search reveals that all graphs of order 8 and 12 are vertex-switch reconstructible.

1.3 Some Simple Results

We begin our journey by establishing a few lemmas, some of which clarify what is actually going on when we switch on the vertices of a graph, and some of which are quite useful in more substantial proofs.

Lemma 1.3.1 *Vertex-switch similarity is an equivalence relation on the vertex set of a graph.*

Proof: We would like to show that vertex-switch similarity is reflexive, symmetric, and transitive. Let u, v and w be vertices of a graph G. Then u is VS-similar to itself, since $G \circledast u \cong G \circledast u$. Therefore, vertex-switch similarity is reflexive. If u is VS-similar to v, then $G \circledast u \cong G \circledast v$, and so $G \circledast v \cong G \circledast u$, which means v is VS-similar to u. This shows vertex-switch similarity to be symmetric. Finally, if u is VS-similar to v and v is VS-similar to w, then $G \circledast u \cong G \circledast v$ and $G \circledast v \cong G \circledast w$. Therefore $G \circledast u \cong G \circledast w$, and so u is VS-similar to w. It follows that VS-similarity is transitive. We conclude that VS-similarity is an equivalence relation. ∎

Definition 1.3.2 The equivalence classes on the vertex set of a graph G with respect to VS-similarity are called the *vertex-switch similarity classes* of G.

Note that the vertex-switch similarity classes of a graph G partition $V(G)$.

Lemma 1.3.3 *Vertex-switch equivalence is an equivalence relation on the set of all graphs.*

Proof: We would like to show that vertex-switch equivalence is reflexive, symmetric, and transitive. Let G be any graph. Then $\mathrm{SD}(G) \cong \mathrm{SD}(G)$ (and clearly $V(G) = V(G)$), so G is VSE to itself and hence vertex-switch equivalence is reflexive. Also, for any graphs G_1 and G_2, if $V(G_1) = V(G_2)$ and $\mathrm{SD}(G_1) \cong \mathrm{SD}(G_2)$, then

$V(G_2) = V(G_1)$ and $\mathrm{SD}(G_2) \cong \mathrm{SD}(G_1)$. Therefore, G_2 is VSE to G_1 if and only if G_1 is VSE to G_2. Therefore, vertex-switch equivalence is symmetric. Finally, if G_1, G_2 and G_3 are all graphs with $V(G_1) = V(G_2)$ and $V(G_2) = V(G_3)$, and if G_1 is VSE to G_2 and G_2 is VSE to G_3, then $V(G_1) = V(G_3)$, and $\mathrm{SD}(G_1) \cong \mathrm{SD}(G_2) \cong \mathrm{SD}(G_3)$ and so $\mathrm{SD}(G_1) \cong \mathrm{SD}(G_3)$. This shows that G_1 is VSE to G_3. Thus vertex-switch equivalence is also transitive. We conclude that vertex-switch equivalence is an equivalence relation. ∎

Now we show that a few simple properties of a graph are vertex-switch reconstructible. In particular, the number of vertices, number of edges, and degree sequence of a graph are all VSR. The latter means that if two graphs (on the same vertex set) have isomorphic switch decks, then they must have the same degree sequence (see Definition 1.3.6).

Lemma 1.3.4 *The order of a graph is a VSR property.*

Proof: Let G and H be any VSE graphs. Then, since $\mathrm{SD}(G) \cong \mathrm{SD}(H)$, it follows that $|\mathrm{SD}(G)| = |\mathrm{SD}(H)|$. However, for any graph K we have $|\mathrm{SD}(K)| = \nu_K$, and so $\nu_G = \nu_H$. Since any two VSE graphs have the same number of vertices, we conclude that the order of a graph is vertex-switch reconstructible. ∎

Lemma 1.3.5 *The size of a graph of order not equal to* 4 *is a VSR property.*

Proof: Let G and H be any VSE graphs, and let $v \in V(G)$. The number of edges in $G \circledast v$ is

$$\varepsilon_G - d_G(v) + ((\nu_G - 1) - d_G(v)) = \varepsilon_G - 2d_G(v) + \nu_G - 1.$$

Now since $\sum_{v \in V(G)} d_G(v) = 2\varepsilon_G$, summing the number of edges in $G \circledast v$ over all $v \in V(G)$ gives

$$\sum_{v \in V(G)} \big(\varepsilon_G - 2d_G(v) + \nu_G - 1\big) = \nu_G\varepsilon_G - 4\varepsilon_G + \nu_G^2 - \nu_G$$

$$= \nu_G^2 - \nu_G + (\nu_G - 4)\varepsilon_G.$$

Therefore the total number of edges in all of the graphs in $\mathrm{SD}(G)$ is $\nu_G^2 - \nu_G + (\nu_G - 4)\varepsilon_G$. Similarly, the total number of edges in all of the graphs in $\mathrm{SD}(H)$ is $\nu_H^2 - \nu_H + (\nu_H - 4)\varepsilon_H$. Since $\mathrm{SD}(G) \cong \mathrm{SD}(H)$, there is a bijection which maps each graph in $\mathrm{SD}(G)$ to an isomorphic graph in $\mathrm{SD}(H)$, and so the total number of edges in the graphs of $\mathrm{SD}(G)$ equals the total number of edges in the graphs of $\mathrm{SD}(H)$. Therefore,

$$\nu_G^2 - \nu_G + (\nu_G - 4)\varepsilon_G = \nu_H^2 - \nu_H + (\nu_H - 4)\varepsilon_H$$

We also note that $\nu_G = \nu_H$ from Lemma 1.3.4, and so

$$\nu_G^2 - \nu_G + (\nu_G - 4)\varepsilon_G = \nu_G^2 - \nu_G + (\nu_G - 4)\varepsilon_H$$
$$(\nu_G - 4)\varepsilon_G = (\nu_G - 4)\varepsilon_H$$

Therefore, if $\nu_G \neq 4$, then $\varepsilon_G = \varepsilon_H$. This means that any graph which is VSE to a graph with other than 4 vertices must have the same size as that graph, and so the size of a graph with other than 4 vertices is VSR. ∎

Currently, the only known non-VSR graphs all have order 4. The first column of Figure 1.3 consists of isomorphism classes of the graphs of order 4, grouped according to vertex-switch equivalance. The graphs in the second column represent the isomorphism classes of the graphs in the switch decks belonging to the graph(s) in

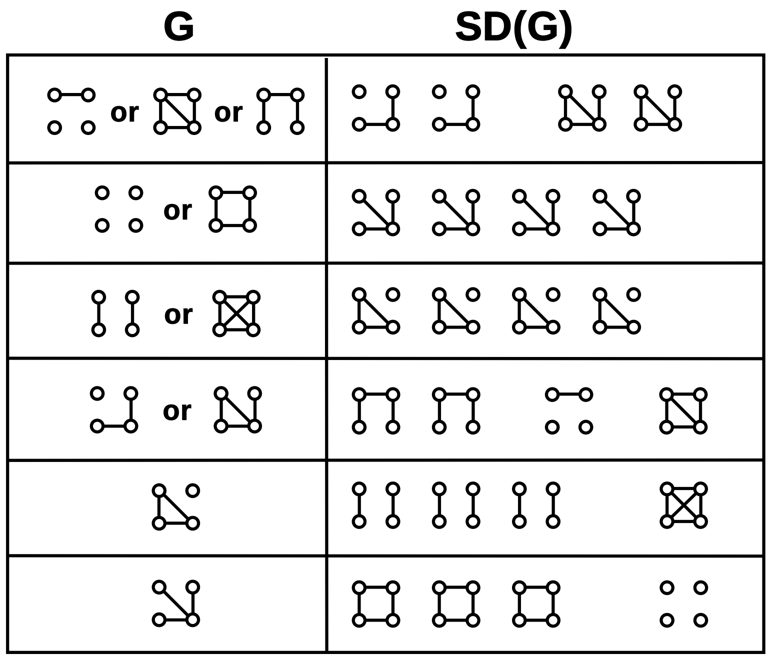

Figure 1.3: Switch decks of graphs on four vertices

the first column. The figure therefore demonstrates that all of the graphs in the first column of any chosen row are VSE, and that, since no two rows contain isomorphic switch decks in their second columns, graphs from different rows of the first column of the figure are not VSE.

Definition 1.3.6 Let G be a graph. Then the *degree sequence* of G is the multiset composed of the degrees of the vertices of G.

Lemma 1.3.7 *The degree sequence of a graph with order not equal to* 4 *is VSR.*

Proof: Let G be a graph with $\nu_G \neq 4$. Since $\nu_G \neq 4$, Lemma 1.3.5 shows $\varepsilon_G = |E(G)|$ is vertex-switch reconstructible. Let $\{v_1, v_2, \ldots, v_{\nu_G}\}$ be the vertex set of G. The switch deck of G consists of ν_G switch cards. For all $1 \leq i \leq \nu_G$, let

$M_i = G \divideontimes v_i$. Then $\varepsilon_{M_i} = \varepsilon_G + \nu_G - 2d_G(v_i) - 1$, that is,

$$d_G(v_i) = \tfrac{1}{2}(\varepsilon_G - \varepsilon_{M_i} + \nu_G - 1). \tag{1.3.1}$$

We conclude that the degree sequence of a graph of order not equal to 4 is VSR. ∎

1.4 Some Properties of Vertex Switching

We now establish a number of useful tools for working with vertex switching. We begin with some rules for simplifying certain expressions involving vertex switching.

Lemma 1.4.1 *Let G_1 and G_2 be two graphs with $V(G_1) = V(G_2)$, and let $v \in V(G_1)$. Then $G_1 \divideontimes v = G_2 \divideontimes v$ if and only if $G_1 = G_2$.*

Proof: Suppose $G_1 = G_2$. Then, clearly, $G_1 \divideontimes v = G_2 \divideontimes v$. Now suppose $G_1 \divideontimes v = G_2 \divideontimes v$. Then $(G_1 \divideontimes v) \divideontimes v = (G_2 \divideontimes v) \divideontimes v$, and so by Lemma 1.1.7, we have $G_1 = G_2$. ∎

Lemma 1.4.2 *Let G be a graph of order at least 3, and let $v, w \in V(G)$ such that $G \divideontimes v = G \divideontimes w$. Then $v = w$.*

Proof: Since $G \divideontimes v = G \divideontimes w$, we have $G \divideontimes v \divideontimes w = G \divideontimes w \divideontimes w$, which, by Lemma 1.1.7, equals G. Suppose $v \neq w$. Let $x \in V(G) \setminus \{v, w\}$. Now observe that $e_{G \divideontimes v \divideontimes w}(v, x) = e_{G \divideontimes v}(v, x) = 1 - e_G(v, x)$. Therefore $G \divideontimes v \divideontimes w \neq G$, a contradiction. Thus $v = w$. ∎

Lemma 1.4.3 *Let G be a graph on a vertex set V, and let $U, U' \subseteq V$. If $G \divideontimes U = G \divideontimes U'$, then either $U = U'$ or $V \setminus U = U'$.*

Proof: Let $U, U' \subseteq V$ such that $G \circledast U = G \circledast U'$. Let $A = U \cap U'$. Then

$$G \circledast (U \setminus A) = G \circledast U \circledast A = G \circledast U' \circledast A = G \circledast (U' \setminus A),$$

and hence

$$G \circledast ((U \cup U') \setminus A) = G \circledast (U \setminus A) \circledast (U' \setminus A) = G.$$

Now for any $v, w \in V$, we have $vw \in E(G)$ if and only if $vw \in E(G \circledast ((U \cup U') \setminus A))$. Consequently, v and w are either both in $(U \cup U') \setminus A$, or neither. Since this holds for any two vertices of V, we conclude that either $(U \cup U') \setminus A = V$ or $(U \cup U') \setminus A = \emptyset$. Hence either $U \cup U' = V$ and $A = \emptyset$, or else $U \cup U' = A$. The former yields $U' = V \setminus U$, and the latter $U = U'$, as claimed. ∎

The following result, which will become quite useful in Chapter 4, shows that switching on a set of vertices of a graph has the same effect as switching on all of the vertices not in the set. This implies that, given a graph, we can produce all of the graphs obtained by switching on each vertex subset, by considering only the subsets consisting of at most half of the vertices.

Lemma 1.4.4 *Let G be a graph, and let $U \subseteq V(G)$. Then $G \circledast U = G \circledast (V(G) \setminus U)$.*

Proof: Let $\overline{U} = V(G) \setminus U$. We will show that $e_{G \circledast U}(x, y) = e_{G \circledast \overline{U}}(x, y)$ for all pairs $x, y \in V(G)$. Let x and y be distinct elements of U. Since switching on any vertex other than x or y does not affect the mutual adjacency of x and y, we have $e_{G \circledast \overline{U}}(x, y) = e_G(x, y)$, and $e_{G \circledast U}(x, y) = e_{G \circledast x \circledast y}(x, y)$. But $e_{G \circledast x \circledast y}(x, y) = 1 - e_{G \circledast x}(x, y) = 1 - (1 - e_G(x, y)) = e_G(x, y)$. Therefore, $e_{G \circledast U}(x, y) = e_G(x, y)$, and so $e_{G \circledast U}(x, y) = e_{G \circledast \overline{U}}(x, y)$. Similarly, if x and y are distinct elements of $\overline{U}$, then $e_{G \circledast \overline{U}}(x, y) = e_G(x, y) = e_{G \circledast U}(x, y)$. The only other possibility is that one of x and y is in U and the other is in $\overline{U}$. In this case, $e_{G \circledast U}(x, y) = 1 - e_G(x, y) = e_{G \circledast \overline{U}}(x, y)$. Therefore $e_{G \circledast U}(x, y) = e_{G \circledast \overline{U}}(x, y)$ for all distinct x and y in $V(G)$, and so $G \circledast U =$

$G \circledast (U \setminus V(G))$. ∎

Another way of looking at the previous proof is to observe that switching on a set of vertices affects only the edges (and non-edges) that have a single endpoint in that set. This then leads us to the following simple result.

Corollary 1.4.5 *If G is a graph, then $G \circledast V(G) = G$.*

Proof: Let $U = \emptyset$. Then $V(G) \setminus U = V(G)$, and the result follows from Lemma 1.4.4. ∎

Now we examine the interaction between complementing a graph and switching on its vertices. We will see that the operations of vertex switching and complementation commute with one another, and that complementation preserves vertex-switch equivalence.

Definition 1.4.6 Let G be a graph. Then the *complement* of G, denoted $\overline{G}$, is the graph with $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{\, vw : v, w \in V(G),\ v \neq w, \text{ and } e_G(v, w) = 0 \,\}$.

Definition 1.4.7 Let G_1, G_2, and H_1 be graphs, with $V(H_1) \subseteq V(G_1)$. Let φ be an isomorphism that maps G_1 to G_2. Then φ induces an isomorphism that maps H_1 to the graph H_2, where $V(H_2) = \varphi(V(H_1))$ and $E(H_2) = \{\, \varphi(x)\varphi(y) : xy \in E(H_1) \,\}$. We then write $\varphi(H_1) = H_2$.

Lemma 1.4.8 *Let G and H be graphs, and let φ be an isomorphism which maps G to H. Then φ is an isomorphism from $\overline{G}$ to $\overline{H}$.*

Proof: Since $\varphi(G) = H$, we have $e_G(x, y) = e_H(\varphi(x), \varphi(y))$ for all $x, y \in V(G)$ with $x \neq y$. Then $e_{\overline{G}}(x, y) = 1 - e_G(x, y) = 1 - e_H(\varphi(x), \varphi(y)) = e_{\overline{H}}(\varphi(x), \varphi(y))$.

Thus $\varphi(\overline{G}) = \overline{H}$. ∎

Lemma 1.4.9 *Let G be a graph and let $v \in V(G)$. Then $\overline{G} \circledast v = \overline{G \circledast v}$.*

Proof: Let $x, y \in V(G)$ with $x \neq y$. Suppose $x = v$. Then $e_{\overline{G} \circledast v}(x, y) = 1 - e_{\overline{G}}(x, y) = 1 - (1 - e_G(x, y)) = 1 - (e_{G \circledast v}(x, y)) = e_{\overline{G \circledast v}}(x, y)$. Now suppose $x \neq v$ and $y \neq v$. Then $e_{\overline{G} \circledast v}(x, y) = e_{\overline{G}}(x, y) = 1 - e_G(x, y) = 1 - e_{G \circledast v}(x, y) = e_{\overline{G \circledast v}}(x, y)$. Thus in all cases, $e_{\overline{G} \circledast v}(x, y) = e_{\overline{G \circledast v}}(x, y)$, and so $\overline{G} \circledast v = \overline{G \circledast v}$. ∎

Lemma 1.4.10 *Let G and H be vertex-switch equivalent graphs. Then $\overline{G}$ and $\overline{H}$ are vertex-switch equivalent as well.*

Proof: Since $V(G) = V(H)$ implies $V(\overline{G}) = V(\overline{H})$, we need only show that there is a bijection $\psi_2 : \mathrm{SD}(\overline{G}) \to \mathrm{SD}(\overline{H})$ such that for all $C \in \mathrm{SD}(\overline{G})$, we have $\psi_2(C) \cong C$.

Since G and H are VSE, we have $\mathrm{SD}(G) \cong \mathrm{SD}(H)$. This means there is a bijective mapping $\psi : \mathrm{SD}(G) \to \mathrm{SD}(H)$ such that if $\psi(G \circledast v) = H \circledast w$ for some $w \in V(H)$, then $G \circledast v \cong H \circledast w$. Define a map $\psi_2 : \mathrm{SD}(\overline{G}) \to \mathrm{SD}(\overline{H})$ by $\psi_2(\overline{G} \circledast v) = \overline{H} \circledast w$ if $\psi(G \circledast v) = H \circledast w$ for some $w \in V(G)$, for all $v \in V(G)$. Since ψ is a bijection, ψ_2 is a bijection as well. Let $G \circledast v$ be any element of $\mathrm{SD}(G)$, and let $H \circledast w = \psi(G \circledast v)$. Then since $G \circledast v \cong H \circledast w$, there is some isomorphism φ from $G \circledast v$ to $H \circledast w$. Now we can apply Lemma 1.4.8 to see that φ is also an isomorphism from $\overline{G \circledast v}$ to $\overline{H \circledast w}$. Applying Lemma 1.4.9 then shows that φ is an isomorphism from $\overline{G} \circledast v$ to $\overline{H} \circledast w$, and therefore $\overline{G} \circledast v \cong \overline{H} \circledast w$. Thus $\mathrm{SD}(\overline{G}) \cong \mathrm{SD}(\overline{H})$, and so $\overline{G}$ and $\overline{H}$ are VSE. ∎

Corollary 1.4.11 *Let G be a graph. Then G is VSR if and only if $\overline{G}$ is VSR.*

Proof: Since $\overline{\overline{G}} = G$, it suffices to show that if G is VSR, then so is $\overline{G}$. Suppose G is VSR, and suppose further that $\overline{G}$ is not VSR. Then there is some graph H on $V(G)$ that is VSE to $\overline{G}$, such that $\overline{G} \ncong H$. Now by Lemma 1.4.10, we have that G and $\overline{H}$ are VSE. However, $\overline{G} \ncong H$ implies $G \ncong \overline{H}$, which contradicts our supposition that G is VSR. Thus $\overline{G}$ is VSR. Therefore, G is VSR if and only if $\overline{G}$ is VSR. ∎

Another useful result shows that, given an isomorphism that maps one graph to another, switching on a vertex in the first graph produces a graph which the isomorphism maps to the graph produced by switching on the image under the isomorphism of this vertex in the second graph.

Lemma 1.4.12 *Let G_1 and G_2 be graphs with $V(G_1) = V(G_2)$, let $u \in V(G_1)$, and let φ be an isomorphism that maps G_1 to G_2. Then φ is an isomorphism from $G_1 \circledast u$ to $G_2 \circledast \varphi(u)$.*

Proof: Let x and y be any two distinct vertices of $\varphi(G_1 \circledast u)$. Then $e_{\varphi(G_1 \circledast u)}(x, y) = e_{G_1 \circledast u}(\varphi^{-1}(x), \varphi^{-1}(y))$. Suppose $\varphi^{-1}(x) = u$. Then

$$\begin{aligned}
e_{\varphi(G_1 \circledast u)}(x, y) &= e_{G_1 \circledast u}(u, \varphi^{-1}(y)) \\
&= 1 - e_{G_1}(u, \varphi^{-1}(y)) \\
&= 1 - e_{G_2}(\varphi(u), y) \qquad \text{(since } \varphi \text{ is an isomorphism from } G_1 \text{ to } G_2\text{)} \\
&= 1 - e_{G_2}(x, y) \\
&= e_{G_2 \circledast x}(x, y) \\
&= e_{G_2 \circledast \varphi(u)}(x, y).
\end{aligned}$$

Now suppose $\varphi^{-1}(x) \neq u$ and $\varphi^{-1}(y) \neq u$. Then $x \neq \varphi(u)$ and $y \neq \varphi(u)$. Therefore,

$$\begin{aligned}
e_{\varphi(G_1 \circledast u)}(x, y) &= e_{G_1 \circledast u}(\varphi^{-1}(x), \varphi^{-1}(y)) \\
&= e_{G_1}(\varphi^{-1}(x), \varphi^{-1}(y))
\end{aligned}$$

$$
\begin{aligned}
&= e_{G_2}(x, y) && \text{(since } \varphi \text{ is an isomorphism from } G_1 \text{ to } G_2\text{)} \\
&= e_{G_2 \circledast \varphi(u)}(x, y) && \text{(since } x \neq \varphi(u) \text{ and } y \neq \varphi(u)\text{)}.
\end{aligned}
$$

Therefore, $e_{\varphi(G_1 \circledast u)}(x, y) = e_{G_2 \circledast \varphi(u)}(x, y)$ for all x and y in $V(G_1)$ with $x \neq y$, and so $\varphi(G_1 \circledast u) = G_2 \circledast \varphi(u)$. ∎

The notion of vertex similarity (defined below) is stronger than that of vertex-switch similarity, as we shall see in the following lemma.

Definition 1.4.13 Two vertices u and v of a graph G are *similar* if some automorphism of G maps u onto v.

Lemma 1.4.14 *Let G be a graph, and let u and v be similar vertices of G. Then u and v are vertex-switch similar.*

Proof: Since u and v are similar, there is some $\varphi \in \mathrm{Aut}(G)$ such that $\varphi(u) = v$. Then by Lemma 1.4.12, we have that φ is an isomorphism from $G \circledast u$ to $\varphi(G) \circledast \varphi(u)$, and $\varphi(G) \circledast \varphi(u) = G \circledast \varphi(u) = G \circledast v$. Therefore $G \circledast u \cong G \circledast v$, and so u and v are vertex-switch similar. ∎

The example below shows that the converse of the previous lemma is not true.

Definition 1.4.15 Two vertices u and v of a graph G are *vertex-switch pseudosimilar* if $G \circledast u \cong G \circledast v$ but u and v are not similar.

Example 1.4.16 Figure 1.4 shows an example of a graph with two vertex-switch pseudosimilar vertices, namely u and v. Note that u and v are not similar in G. (This is easily verified since there is a set $S = \{u, a, b\}$ of three vertices including u such that $G[S]$ has three edges, but there is no such set of three vertices which includes

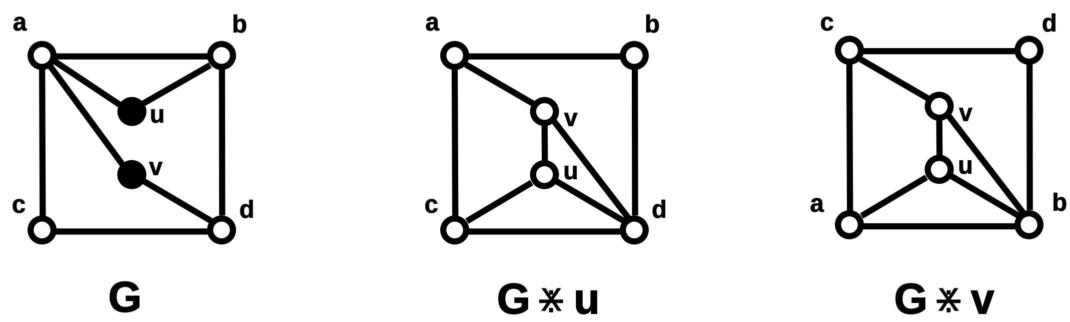

Figure 1.4: A graph with two vertex-switch pseudosimilar vertices

v, and therefore no automorphism of G can map u to v.) However, $G \circledast u \cong G \circledast v$. Therefore, u and v are vertex-switch pseudosimilar.

Ellingham [3] completely characterised all graphs in which vertex-switch similar vertices exist. Unfortunately this characterisation does not distinguish between similar pairs and vertex-switch pseudosimilar pairs. This means that his results cannot be used to find graphs with pairs of vertices which are vertex-switch pseudosimilar.

Chapter 2

Graphs of Order Not Divisible by 4

The first result about the vertex switching reconstruction problem was proven in the same paper that introduced the problem. In 1985, Stanley [14] showed that all graphs whose order is not divisible by 4 must be vertex-switch reconstructible. However, his proof does not show how to reconstruct such a graph from its switch deck. He also proved some results related to a more general problem, namely, the question of which graphs can be reconstructed from the multiset of graphs produced by switching on all s-vertex subsets of their vertices, for any fixed s. (Graphs which can be so reconstructed are termed s-*VSR*.) However, this more general question is outside of the scope of this thesis.

The proof of the $s = 1$ case is quite involved. Therefore an informal description of the proof might be helpful at this point. Stanley treats graphs as binary vectors, whose coordinates correspond to unordered pairs of distinct vertices. In this world, switching on a vertex v of a graph G is modelled by adding (modulo 2) the vector corresponding to the star graph with all possible edges incident with v to the vector corresponding to G. Then we move to the world of formal linear combinations of these vectors. From this viewpoint, we can create an operator (the "unlabelling" operator) which maps a graph to the formal sum of the vectors of the graphs in its isomorphism class. We can also define a transformation, called the switch-deck transformation, which maps a graph to the formal sum of the vectors of the graphs in its switch deck. Then, by use of these operators, we can express algebraically the notion of two graphs having isomorphic switch decks. The switch deck operator is then shown to be have a left inverse whenever a certain linear transformation (the Fourier transform of the characteristic function of the set of star graphs) never achieves the value of 0. Finally, the Fourier transform of the characteristic function of the set of star graphs is shown to take on the value 0 only when the number of vertices is divisible by 4.

Before we dive into all of this, we will define the Fourier transform. The version presented here is a special case of the usual definition.

2.1 The Discrete Fourier Transform

A formal definition of a graph vector is required before we can describe the Fourier transform.

Definition 2.1.1 Let G be a graph of order n with vertex set $V = \{v_1, v_2, \ldots, v_n\}$, and let $k = \binom{n}{2}$. Then the *graph vector* of G, denoted X_G, is the element of $\mathbb{Z}_2^k$ with coordinates indexed by $\{v_i, v_j\} \in V \times V$ such that $X_G[\{v_i, v_j\}] = e_G(v_i, v_j)$ (where $X[i]$ denotes the ith coordinate of the vector X). We also define $X_G \circledast v_i$ (where $v_i \in \nu_G$) to mean the graph vector $X_{G \circledast v_i}$.

We will use the above definitions of G, n, V, and k throughout this chapter. Note that the association between a particular coordinate of a graph vector and a pair of vertices of the graph associated with that graph vector is the same for all graphs on the same set of vertices. Therefore, given a set of graphs on the same vertex set, we may refer to the pair of vertices corresponding to a particular coordinate of a graph vector without specifying a particular graph vector.

Definition 2.1.2 Let $X, Y \in \mathbb{Z}_2^k$, and let $X = (a_1, a_2, \ldots, a_k)$ and $Y = (b_1, b_2, \ldots, b_k)$, where $a_i, b_i \in \mathbb{Z}_2$ for all $i \in \{1, 2, \ldots, k\}$. Then the *dot product* of X and Y, denoted $X \cdot Y$, is the scalar value $\displaystyle\sum_{i=1}^{k} a_i b_i$.

The following properties are an immediate consequence of the previous definition.

Lemma 2.1.3 *Let $X, Y, Z \in \mathbb{Z}_2^k$, and let $X = (a_1, a_2, \ldots, a_k)$, $Y = (b_1, b_2, \ldots, b_k)$, and $Z = (c_1, c_2, \ldots, c_k)$, where $a_i, b_i, c_i \in \mathbb{Z}_2$ for all $i \in \{1, 2, \ldots, k\}$. Then*

(i) $X \cdot Y = Y \cdot X$, *and*

(ii) $(X + Y) \cdot Z = (X \cdot Z) + (Y \cdot Z)$.

Lemma 2.1.4 *Let k be a positive integer, and let $X \in \mathbb{Z}_2^k$. Then (evaluated in $\mathbb{R}$)*

$$\sum_{Y \in \mathbb{Z}_2^k} (-1)^{(X \cdot Y)} = \begin{cases} 2^k, & \text{if } X = 0; \\ 0, & \text{otherwise.} \end{cases}$$

Proof: Let $e(X)$ be the number of 0 elements of $S_X = \left\langle X \cdot Y : Y \in \mathbb{Z}_2^k \right\rangle$. Then the number of 1 elements of S_X is $2^k - e(X)$. If $X \cdot Y$ is even, then $(-1)^{X \cdot Y} = 1$, and if $X \cdot Y$ is odd, then $(-1)^{X \cdot Y} = -1$. Therefore,

$$\sum_{Y \in \mathbb{Z}_2^k} (-1)^{(X \cdot Y)} = e(X) - (2^k - e(X)) = 2e(X) - 2^k.$$

If $X$is the zero vector, then $X \cdot Y$ is 0 for all $Y \in \mathbb{Z}_2^k$, and so $e(X) = 2^k$. In this case, $\displaystyle\sum_{Y \in \mathbb{Z}_2^k} (-1)^{(X \cdot Y)} = 2(2^k) - 2^k = 2^k$. Now suppose X has at least one non-zero coordinate. Let r be the position of the first 1 in X. For every $Y \in \mathbb{Z}_2^k$, there is a unique $Y' \in \mathbb{Z}_2^k$ which differs from Y only in position r. Then the set $\{X \cdot Y, X \cdot Y'\}$ contains one even and one odd element. Since $\mathbb{Z}_2^k$ can be partitioned into pairs of the form $\{Y, Y'\}$, exactly half of the elements of S_X are even, and so $e(X) = \frac{1}{2} 2^k$. Therefore, $\sum_{Y \in \mathbb{Z}_2^k} (-1)^{(X \cdot Y)} = 2(\frac{1}{2}) 2^k - 2^k = 2^k - 2^k = 0$. ∎

Definition 2.1.5 Define $\mathcal{F}_k$ as the set of all functions which map graph vectors with k coordinates to real numbers, i.e. $\mathcal{F}_k = \{\, f : \mathbb{Z}_2^k \to \mathbb{R} \,\}$.

Definition 2.1.6 Let k be a positive integer. The *Fourier transform* is a mapping from $\mathcal{F}_k$ to $\mathcal{F}_k$ defined by $f \mapsto \widehat{f}$ where, for all $X \in \mathbb{Z}_2^k$,

$$\widehat{f}(X) = \sum_{Y \in \mathbb{Z}_2^k} (-1)^{X \cdot Y} f(Y).$$

Lemma 2.1.7 *The Fourier transform has a left inverse, given by*

$$f(X) = \frac{1}{2^k} \sum_{Y \in \mathbb{Z}_2^k} (-1)^{X \cdot Y} \widehat{f}(Y), \textit{ for all } f \in \mathcal{F}_k, X \in \mathbb{Z}_2^k. \tag{2.1.1}$$

Proof: To show that the Fourier transform has a left inverse, it suffices to prove (2.1.1); that is, to show that

$$f(X) = \frac{1}{2^k} \sum_{Y \in \mathbb{Z}_2^k} (-1)^{X \cdot Y} \left(\sum_{Z \in \mathbb{Z}_2^k} (-1)^{Y \cdot Z} f(Z) \right) \tag{2.1.2}$$

for all $X \in \mathbb{Z}_2^k$. Fix $X \in \mathbb{Z}_2^k$. For any $Z \in \mathbb{Z}_2^k$, let $W = X + Z$. Then the right hand side of (2.1.2) is equal to

$$\begin{aligned}
&\frac{1}{2^k} \sum_{Y \in \mathbb{Z}_2^k} (-1)^{X \cdot Y} \left(\sum_{W \in \mathbb{Z}_2^k} (-1)^{Y \cdot (W - X)} f(W - X) \right) \\
&= \frac{1}{2^k} \sum_{W \in \mathbb{Z}_2^k} \sum_{Y \in \mathbb{Z}_2^k} \big((-1)^{X \cdot Y} (-1)^{Y \cdot (W - X)} f(W - X) \big) \\
&= \frac{1}{2^k} \sum_{W \in \mathbb{Z}_2^k} \sum_{Y \in \mathbb{Z}_2^k} \big((-1)^{X \cdot Y} (-1)^{(Y \cdot W - Y \cdot X)} f(W - X) \big) \\
&= \frac{1}{2^k} \sum_{W \in \mathbb{Z}_2^k} \sum_{Y \in \mathbb{Z}_2^k} \big((-1)^{(X \cdot Y + Y \cdot W - Y \cdot X)} f(W - X) \big) \\
&= \frac{1}{2^k} \sum_{W \in \mathbb{Z}_2^k} \sum_{Y \in \mathbb{Z}_2^k} \big((-1)^{(X \cdot Y + Y \cdot W - X \cdot Y)} f(W - X) \big) \\
&= \frac{1}{2^k} \sum_{W \in \mathbb{Z}_2^k} \sum_{Y \in \mathbb{Z}_2^k} \big((-1)^{Y \cdot W} f(W - X) \big)
\end{aligned}$$

$$= \frac{1}{2^k} \sum_{W \in \mathbb{Z}_2^k} \left(f(W - X) \sum_{Y \in \mathbb{Z}_2^k} (-1)^{Y \cdot W} \right). \tag{2.1.3}$$

Using Lemma 2.1.4, we know that $\sum_{Y \in \mathbb{Z}_2^k} (-1)^{Y \cdot W} = 0$ when $W \neq 0$. Therefore, (2.1.3) is equal to

$$\begin{aligned}
&\frac{1}{2^k} f(0 - X) \left(\sum_{Y \in \mathbb{Z}_2^k} (-1)^{Y \cdot 0} \right) \\
&= \frac{1}{2^k} f(0 - X)(2^k) && \text{(also by Lemma 2.1.4)} \\
&= f(0 - X) \\
&= f(X) && \text{(since } X = -X \text{ in } \mathbb{Z}_2^k\text{)}
\end{aligned}$$

as required. ∎

2.2 Unlabelling

We now formally define the vector space of linear combinations of graph vectors, which then allows us to define the unlabelling operator. This is a linear transformation which maps a graph vector to a (formal) sum of the graph vectors of all members of the isomorphism class of the original graph. This is used to express the notion of graphs being isomorphic—two graphs are isomorphic if and only if the their vectors have equal images under the unlabelling operator.

Throughout this section, we let $r = 2^k$, and let $\mathbb{Z}_2^k = \{X_1, X_2, \ldots, X_r\}$.

Definition 2.2.1 The *star graph* on vertex set V where $n = |V|$, with v as the centre (where $v \in V$) is the graph containing $n-1$ edges, each with one endpoint at v. This graph is denoted $C_n(V, v)$. Where V is understood, we may write simply $C_n(v)$.

Lemma 2.2.2 *Let G be a graph, and let $v \in V$. Then $X_{G \circledast v} = X_G + X_{C_n(v)}$ (where the addition is performed in $\mathbb{Z}_2^k$).*

Proof: Since $C_n(v)$ contains all possible edges incident with v, the graph vector $X_{C_n(v)}$ has a 1 in precisely those coordinates which correspond to edges incident with v.

Suppose $X_{C_n(v)}[\{a,b\}] = 0$ for some $a, b \in V$. Then $(X_G + X_{C_n(v)})[\{a,b\}] = X_G[\{a,b\}]$, and we also have $v \notin \{a,b\}$. Hence switching on v does not change the $\{a,b\}$-coordinate of the corresponding graph vector. We thus have $X_{G \circledast v}[\{a,b\}] = X_G[\{a,b\}]$. It follows that $X_{G \circledast v}[\{a,b\}] = (X_G + X_{C_n(v)})[\{a,b\}]$.

Now suppose $X_{C_n(v)}[\{a,b\}] = 1$ for some $a, b \in V$. Then $(X_G + X_{C_n(v)})[\{a,b\}] = X_G[\{a,b\}] + 1$, and we also have $v \in \{a,b\}$. Hence $X_{G \circledast v}[\{a,b\}] = X_G[\{a,b\}] + 1 = (X_G + X_{C_n(v)})[\{a,b\}]$. Therefore, since for all coordinates $\{a,b\}$ we have $X_{G \circledast v}[\{a,b\}] = (X_G + X_{C_n(v)})[\{a,b\}]$, this shows that $X_{G \circledast v} = X_G + X_{C_n(v)}$. ∎

Definition 2.2.3 Define $\mathcal{V}_n$ as the vector space of all formal linear combinations of elements of $\mathbb{Z}_2^k$ with coefficients in $\mathbb{R}$. That is, if $W \in \mathcal{V}_n$, then $W = c_1X_1 + c_2X_2 + \cdots + c_rX_r$, for $c_1, c_2, \ldots, c_r \in \mathbb{R}$.

Let Sym_n be the group of all permutations on n items. A permutation σ on a set V induces an action on the set of all graphs with vertex set V as follows: $\sigma(G) = (V, E')$, where $E' = \{\, \sigma(v_i)\sigma(v_j) : v_iv_j \in E(G)\,\}$. Hence $v_iv_j \in E'$ if and only if $\sigma^{-1}(v_i)\sigma^{-1}(v_j) \in E(G)$, and

$$X_{\sigma(G)}[\{v_i, v_j\}] = X_G[\{\sigma^{-1}(v_i), \sigma^{-1}(v_j)\}]. \tag{2.2.1}$$

The action of σ on $\mathbb{Z}_2^k$ is simply defined as $\sigma(X_G) = X_{\sigma(G)}$ for all graphs G. Now we extend the definition to all of $\mathcal{V}_n$ by

$$\sigma\left(\sum_{i=1}^{r} c_iX_i\right) = \sum_{i=1}^{r} c_i\sigma(X_i). \tag{2.2.2}$$

Next, we show that any $\sigma \in \mathrm{Sym}_n$, in its action on $\mathcal{V}_n$ as defined above, is a linear transformation.

Lemma 2.2.4 *Let $\sigma \in \mathrm{Sym}_n$, let $c \in \mathbb{R}$, and let $X = \sum_{i=1}^{r} a_i X_i$ and $Y = \sum_{i=1}^{r} b_i X_i$, where $a_i, b_i \in \mathbb{R}$ for all $i \in \{1, 2, \ldots, r\}$. Then*

(i) $\sigma(cX) = c\sigma(X)$, and

(ii) $\sigma(X + Y) = \sigma(X) + \sigma(Y)$.

Proof: (i) Observe that

$$\begin{aligned}
\sigma(cX) &= \sigma\left(c\sum_{i=1}^{r} a_i X_i\right) \\
&= \sigma\left(\sum_{i=1}^{r} c a_i X_i\right) \\
&= \sum_{i=1}^{r} c a_i \sigma(X_i) && \text{(by Equation (2.2.2))} \\
&= c\left(\sum_{i=1}^{r} a_i \sigma(X_i)\right) \\
&= c\sigma\left(\sum_{i=1}^{r} a_i X_i\right) && \text{(again by Equation (2.2.2))} \\
&= c\sigma(X).
\end{aligned}$$

(ii) Observe that

$$\begin{aligned}
\sigma(X + Y) &= \sigma\left(\sum_{i=1}^{r} a_i X_i + \sum_{i=1}^{r} b_i X_i\right) \\
&= \sigma\left(\sum_{i=1}^{r} (a_i + b_i) X_i\right) \\
&= \sum_{i=1}^{r} (a_i + b_i) \sigma(X_i) && \text{(by Equation (2.2.2))} \\
&= \sum_{i=1}^{r} a_i \sigma(X_i) + \sum_{i=1}^{r} b_i \sigma(X_i)
\end{aligned}$$

$$= \sigma\left(\sum_{i=1}^{r} a_i X_i\right) + \sigma\left(\sum_{i=1}^{r} b_i X_i\right) \qquad \text{(by Equation (2.2.2))}$$
$$= \sigma(X) + \sigma(Y).$$

∎

We now have two different ways of adding two graph vectors. Such an addition can occur either in $\mathbb{Z}_2^k$ or in $\mathcal{V}_n$. Unless otherwise specified, for the balance of this chapter, addition of graph vectors will occur in $\mathcal{V}_n$.

We are now ready to define the unlabeling operator, which can be used to describe the isomorphism class of a graph in terms of graph vectors.

Definition 2.2.5 The *unlabeling* on $\mathcal{V}_n$ is a mapping $\Upsilon : \mathcal{V}_n \to \mathcal{V}_n$ defined by

$$\Upsilon(W) = \sum_{\sigma \in \mathrm{Sym}_n} \sigma(W)$$

for all $W \in \mathcal{V}_n$.

Note that the unlabeling of a graph vector X_G is the formal sum of the vectors of the graphs produced by permuting the vertices of G in all possible ways. Note also that every graph isomorphic to G appears in $\Upsilon(X_G)$ with coefficient equal to the size of the automorphism group of the graph.

Lemma 2.2.6 *Unlabelling is a linear map.*

Proof: We require $\Upsilon(cX) = c\Upsilon(X)$ for all $c \in \mathbb{R}$ and all $X \in \mathcal{V}_n$, and $\Upsilon(X+Y) = \Upsilon(X) + \Upsilon(Y)$ for all $X, Y \in \mathcal{V}_n$. First,

$$\Upsilon(cX) = \sum_{\sigma \in \mathrm{Sym}_n} \sigma(cX)$$
$$= \sum_{\sigma \in \mathrm{Sym}_n} c\sigma(X) \qquad \text{(by Lemma 2.2.4)}$$

$$
\begin{aligned}
&= c \sum_{\sigma \in \mathrm{Sym}_n} \sigma(X) \\
&= c\Upsilon(X).
\end{aligned}
$$

Next,

$$
\begin{aligned}
\Upsilon(X+Y) &= \sum_{\sigma \in \mathrm{Sym}_n} \sigma(X+Y) \\
&= \sum_{\sigma \in \mathrm{Sym}_n} (\sigma(X) + \sigma(Y)) \text{ (by Lemma 2.2.4)} \\
&= \sum_{\sigma \in \mathrm{Sym}_n} \sigma(X) + \sum_{\sigma \in \mathrm{Sym}_n} \sigma(Y) \\
&= \Upsilon(X) + \Upsilon(Y).
\end{aligned}
$$

Therefore, Υ is a linear map. ∎

Lemma 2.2.7 *Let G, H be graphs on the same vertex set of size n. Then $\Upsilon(X_G) = \Upsilon(X_H)$ if and only if $G \cong H$.*

Proof: If $G \cong H$, then there is some $\tau \in \mathrm{Sym}_n$ such that $\tau(G) = H$. Now,

$$
\begin{aligned}
\Upsilon(X_H) &= \sum_{\sigma \in \mathrm{Sym}_n} \sigma(X_H) \\
&= \sum_{\sigma \in \mathrm{Sym}_n} X_{\sigma(H)} \\
&= \sum_{\sigma \in \mathrm{Sym}_n} X_{\sigma(\tau(G))} \\
&= \sum_{\sigma \in \mathrm{Sym}_n} X_{\sigma\tau(G)} \\
&= \sum_{\sigma \in \mathrm{Sym}_n} \sigma\tau(X_G).
\end{aligned}
$$

Since as σ runs through all elements of Sym_n, so does $\sigma\tau$ (since Sym_n is a group under composition), and so we have

$$\sum_{\sigma \in \mathrm{Sym}_n} (\sigma\tau)(X_G) = \sum_{\sigma \in \mathrm{Sym}_n} \sigma(X_G) = \Upsilon(X_G).$$

Therefore $G \cong H$ implies $\Upsilon(X_G) = \Upsilon(X_H)$.

Now assume $G \ncong H$. Then the coefficient of X_G in $\Upsilon(X_H)$ is 0. However, the coefficient of X_G in $\Upsilon(X_G)$ is nonzero. Therefore $\Upsilon(X_G) \neq \Upsilon(X_H)$, and so $\Upsilon(X_G) = \Upsilon(X_H)$ implies $G \cong H$. ∎

2.3 The Switch-Deck Transformation

In order to give the vertex-switching reconstruction problem a presence in the world of graph vectors, we define a linear transformation that maps a graph's vector to a formal sum of the vectors of the graphs of its switch deck. Then we will see that this transformation and the unlabelling transformation commute.

Definition 2.3.1 The *switch-deck transformation* is the mapping $\phi : \mathcal{V}_n \to \mathcal{V}_n$ defined by

$$\phi(W) = \sum_{j=1}^{n} \sum_{i=1}^{r} c_i (X_i + X_{C_n(v_j)})$$

for all $W = \sum_{i=1}^{r} c_i X_i$, where $c_i \in \mathbb{R}$ for all i, and where the sum $X_i + X_{C_n(v_j)}$ involves addition in $\mathbb{Z}_2^k$, i.e., $X_i + X_{C_n(v_j)} = X_{G \circledast v_j}$ if $X_i = X_G$. Note that for $W = X_i$ (for some i) we have $\phi(X_i) = \sum_{j=1}^{n} (X_i + X_{C_n(v_j)})$.

Lemma 2.3.2 *Let G and H be VSE graphs. Then $\Upsilon(\phi(X_G)) = \Upsilon(\phi(X_H))$.*

Proof: By Definition 2.3.1 and Lemma 2.2.2, we have

$$\phi(X_G) = \sum_{j=1}^{n}(X_G + X_{C_n(v_j)}) = \sum_{j=1}^{n} X_{G \circledast v_j}.$$

Since Υ is a linear transformation (by Lemma 2.2.6), we also have

$$\Upsilon(\phi(X_G)) = \Upsilon\left(\sum_{j=1}^{n} X_{G \circledast v_j}\right) = \sum_{j=1}^{n}\Upsilon(X_{G \circledast v_j}). \tag{2.3.1}$$

Now since G and H are VSE, there exists a bijection $\beta : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$ such that $G \circledast v_j \cong H \circledast v_{\beta(j)}$ for all $j \in \{1, 2, \ldots, n\}$. Hence by Lemma 2.2.7, $\Upsilon(X_{G \circledast v_j}) = \Upsilon(X_{H \circledast v_{\beta(j)}})$, for all $j \in \{1, 2, \ldots, n\}$. Thus

$$\begin{aligned}
\Upsilon(\phi(X_G)) &= \sum_{j=1}^{n} \Upsilon(X_{G \circledast v_j}) \\
&= \sum_{j=1}^{n} \Upsilon(X_{H \circledast v_{\beta(j)}}) \\
&= \sum_{j=1}^{n} \Upsilon(X_{H \circledast v_j}) && \text{(since } \beta \text{ is a bijection)} \\
&= \Upsilon(\phi(X_H)) && \text{(by Equation (2.3.1)).}
\end{aligned}$$

∎

Lemma 2.3.3 *The switch-deck transformation is a linear map.*

Proof: In order for ϕ to be a linear map we must have $\phi(X + Y) = \phi(X) + \phi(Y)$ for all $X, Y \in \mathcal{V}_n$, and $\phi(cX) = c\phi(X)$ for all $c \in \mathbb{R}$ and all $X \in \mathcal{V}_n$. Let $X, Y \in \mathcal{V}_n$ with $X = \sum_{i=1}^{r} a_i X_i$ and $Y = \sum_{i=1}^{r} b_i X_i$. Then $X + Y = \sum_{i=1}^{r}(a_i + b_i)X_i$ and so

$$\phi(X + Y) = \sum_{j=1}^{n}\sum_{i=1}^{r}(a_i + b_i)(X_i + X_{C_n(v_j)})$$

(Note that for the balance of this proof, $X_i + X_{C_n(v_j)}$ refers to addition in $\mathbb{Z}_2^k$.)

$$\begin{aligned}
&= \sum_{j=1}^{n}\sum_{i=1}^{r}(a_i(X_i + X_{C_n(v_j)}) + b_i(X_i + X_{C_n(v_j)})) \\
&= \left(\sum_{j=1}^{n}\sum_{i=1}^{r} a_i(X_i + X_{C_n(v_j)})\right) + \left(\sum_{j=1}^{n}\sum_{i=1}^{r} b_i(X_i + X_{C_n(v_j)})\right) \\
&= \phi(X) + \phi(Y).
\end{aligned}$$

Furthermore, let c be a real scalar. Then

$$\begin{aligned}
\phi(cX) &= \sum_{j=1}^{n}\sum_{i=1}^{r} ca_i(X_i + X_{C_n(v_j)}) \\
&= c\sum_{j=1}^{n}\sum_{i=1}^{r} a_i(X_i + X_{C_n(v_j)}) \\
&= c\phi(X)
\end{aligned}$$

Therefore, ϕ is a linear map. ∎

Lemma 2.3.4 *Let $\sigma \in \mathrm{Sym}_n$ and let $X, Y \in \mathbb{Z}_2^k$. Then $\sigma(X+Y) = \sigma(X) + \sigma(Y)$, where both additions occur in $\mathbb{Z}_2^k$.*

Proof: Let $Z \in \mathbb{Z}_2^k$ such that $Z = X + Y$. Then for all $v_i, v_j \in V$, $v_i \neq v_j$, we have $Z[\{v_i, v_j\}] = (X+Y)[\{v_i, v_j\}] = X[\{v_i, v_j\}] + Y[\{v_i, v_j\}]$. Therefore

$$\begin{aligned}
\sigma(Z)[\{v_i, v_j\}] &= \sigma(X+Y)[\{v_i, v_j\}] \\
&= (X+Y)[\{\sigma^{-1}(v_i), \sigma^{-1}(v_j)\}] && \text{(by Equation 2.2.1)} \\
&= X[\{\sigma^{-1}(v_i), \sigma^{-1}(v_j)\}] + Y[\{\sigma^{-1}(v_i), \sigma^{-1}(v_j)\}] \\
&= \sigma(X)[\{v_i, v_j\}] + \sigma(Y)[\{v_i, v_j\}] && \text{(by Equation 2.2.1).}
\end{aligned}$$

Thus, since $\sigma(X+Y)$ and $\sigma(X) + \sigma(Y)$ are equal in every coordinate, $\sigma(X+Y) = \sigma(X) + \sigma(Y)$. ∎

The following lemma and its corollaries were proven by Stanley [14].

Lemma 2.3.5 *Unlabelling and the switch-deck transformation commute. That is,* $\Upsilon\phi = \phi\Upsilon$.

Proof: Let G be a graph, and let X be its corresponding graph vector. Then

$$\begin{aligned}
\phi(\Upsilon(X)) &= \phi\left(\sum_{\sigma\in\mathrm{Sym}_n}\sigma(X)\right) \\
&= \sum_{\sigma\in\mathrm{Sym}_n}\phi(\sigma(X)) && \text{(since } \phi \text{ is linear)} \\
&= \sum_{\sigma\in\mathrm{Sym}_n}\sum_{v\in V}(\sigma(X)+X_{C_n(v)}) && \text{(since } \sigma(X)\in\mathbb{Z}_2^k \text{ for all } X\text{; } + \text{ is in } \mathbb{Z}_2^k\text{)} \\
&= \sum_{\sigma\in\mathrm{Sym}_n}\sum_{v\in V}(\sigma(X)+X_{C_n(\sigma(v))}) && \text{(since } \sigma(v) \text{ ranges over all } V \text{ as } v \text{ does)} \\
&= \sum_{\sigma\in\mathrm{Sym}_n}\sum_{v\in V}(\sigma(X)+\sigma(X_{C_n(v)})) \\
&= \sum_{\sigma\in\mathrm{Sym}_n}\sum_{v\in V}\sigma(X+X_{C_n(v)}) && \text{(by Lemma 2.3.4)} \\
&= \sum_{\sigma\in\mathrm{Sym}_n}\sigma\left(\sum_{v\in V}(X+X_{C_n(v)})\right) && \text{(by Lemma 2.2.4(ii))} \\
&= \sum_{\sigma\in\mathrm{Sym}_n}\sigma(\phi(X)) \\
&= \Upsilon(\phi(X))
\end{aligned}$$

∎

Corollary 2.3.6 *If* G *and* H *are VSE graphs, then* $\phi(\Upsilon(X_G)) = \phi(\Upsilon(X_H))$.

Proof: The result follows from the previous lemma, and Lemma 2.3.2. ∎

Now we are about to motivate the remainder of this chapter, by giving a sufficient condition for a graph of a given order to be VSR.

Corollary 2.3.7 *Assume the switch-deck transformation has a left inverse on* $\mathcal{V}_n$*. Then all graphs on* n *vertices are VSR.*

Proof: Let G be a graph on n vertices. Suppose G is not VSR. Then there exists a graph H such that $G \ncong H$ and G and H are VSE. Then $\phi(\Upsilon(X_G)) = \phi(\Upsilon(X_H))$ by Corollary 2.3.6. Since ϕ is assumed to have a left inverse, we thus have $\Upsilon(X_G) = \Upsilon(X_H)$. Then by Lemma 2.2.7, we have $G \cong H$, a contradiction. Therefore G is VSR. ∎

2.4 Another View of the Switch-Deck Transformation

Next we consider another linear mapping, which resembles the switch-deck transformation, but operates on functions which map graph vectors onto real numbers, rather than on linear combinations of graph vectors. We then describe a sufficient condition for this mapping to have a left inverse. From there it will be a simple matter to relate the existence of a left inverse of this mapping to the existence of a left inverse of the switch-deck transformation (though that particular result will have to wait until the next section).

Definition 2.4.1 Let Λ be a subset of some set S. Then the *characteristic function* of Λ is the mapping $\chi_\Lambda : S \to \mathbb{Z}_2$ where

$$\chi_\Lambda(X) = \begin{cases} 1 & \text{if } X \in \Lambda; \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2.4.2 Let Λ be some non-empty subset of $\mathbb{Z}_2^k$. Define the mapping $\Theta_\Lambda : \mathcal{F}_k \to \mathcal{F}_k$ by

$$\Theta_\Lambda(f)(X) = \sum_{Y \in \Lambda} f(X+Y) \text{ for all } X \in \mathbb{Z}_2^k.$$

Note that here, $X+Y$ refers to addition in $\mathbb{Z}_2^k$, and the sum over all elements of Λ is a sum of real numbers.

Lemma 2.4.3 *Let Λ be some non-empty subset of $\mathbb{Z}_2^k$. Then Θ_Λ is a linear transformation.*

Proof: We need to show that if $f, g \in \mathcal{F}_k$, and $X \in \mathbb{Z}_2^k$, and c is a scalar, then $\Theta_\Lambda(f+g)(X) = \Theta_\Lambda(f)(X) + \Theta_\Lambda(g)(X)$, and that $\Theta_\Lambda(cf)(X) = c\Theta_\Lambda(f)(X)$. First,

$$\begin{aligned} \Theta_\Lambda(f+g)(X) &= \sum_{Y \in \Lambda} (f+g)(X+Y) \\ &= \sum_{Y \in \Lambda} \left(f(X+Y) + g(X+Y)\right) \\ &= \left(\sum_{Y \in \Lambda} f(X+Y)\right) + \left(\sum_{Y \in \Lambda} g(X+Y)\right) \\ &= \Theta_\Lambda(f)(X) + \Theta_\Lambda(g)(X). \end{aligned}$$

Next,

$$\begin{aligned} \Theta_\Lambda(cf)(X) &= \sum_{Y \in \Lambda} (cf)(X+Y) \\ &= \sum_{Y \in \Lambda} cf(X+Y) \\ &= c \sum_{Y \in \Lambda} f(X+Y) \end{aligned}$$

$$= c\Theta_\Lambda(f)(X).$$

Therefore Θ_Λ is a linear transformation. ∎

We are now on the threshold of proving our main result, namely, that graphs with order not divisible by 4 are VSR. The following lemma is the key result of this chapter. With it, we specify a sufficient condition for the existence of a left inverse for Θ_Λ, and soon for the switch-deck transformation as well.

Lemma 2.4.4 (Diaconis and Graham [2]) *Let Γ be the set of elements of $\mathbb{Z}_2^k$ which are the graph vectors of star graphs. Then the mapping Θ_Γ has a left inverse whenever the Fourier transform of the characteristic function of Γ is never 0 (that is, $\widehat{\chi_\Gamma}(X) \neq 0$ for all $X \in \mathbb{Z}_2^k$).*

Proof: First, note that for each $X \in \mathbb{Z}_2^k$,

$$\widehat{\chi_\Gamma}(X) = \sum_{Y \in \mathbb{Z}_2^k} (-1)^{X \cdot Y} \chi_\Gamma(Y) = \sum_{Y \in \Gamma} (-1)^{X \cdot Y}.$$

Next observe that $\Theta_\Gamma(f)$, where $f \in \mathcal{F}_k$, acts on $\mathbb{Z}_2^k$ as follows:

$$\begin{aligned}
\Theta_\Gamma(f)(X) &= \sum_{Y \in \Gamma} f(X+Y) \\
&= \sum_{Y \in \Gamma} f(Y-X) \qquad\qquad (\text{since } Y - X = Y + X \text{ in } \mathbb{Z}_2^k\ ) \\
&= \sum_{Y \in \mathbb{Z}_2^k} f(Y-X)\chi_\Gamma(Y).
\end{aligned}$$

Then the Fourier transform of $\Theta_\Gamma(f)$ acts on $\mathbb{Z}_2^k$ as follows:

$$\begin{aligned}
\widehat{\Theta_\Gamma(f)}(X) &= \sum_{Y \in \mathbb{Z}_2^k} (-1)^{X \cdot Y} \Theta_\Gamma(f)(Y) \\
&= \sum_{Y \in \mathbb{Z}_2^k} (-1)^{X \cdot Y} \left(\sum_{Z \in \mathbb{Z}_2^k} f(Z-Y)\chi_\Gamma(Z) \right)
\end{aligned}$$

$$= \sum_{Y\in\mathbb{Z}_2^k} \sum_{Z\in\mathbb{Z}_2^k} \left((-1)^{X\cdot Y} f(Z-Y)\chi_\Gamma(Z)\right)$$
$$= \sum_{Z\in\mathbb{Z}_2^k} \sum_{Y\in\mathbb{Z}_2^k} \left((-1)^{X\cdot Y} f(Z-Y)\chi_\Gamma(Z)\right).$$

Substituting $W = Z - Y$ we obtain

$$\begin{aligned}
\widehat{\Theta_\Gamma(f)}(X) &= \sum_{Z\in\mathbb{Z}_2^k} \sum_{W\in\mathbb{Z}_2^k} \left((-1)^{X\cdot(Z+W)} f(W)\chi_\Gamma(Z)\right) && \text{(since } W=-W \text{ in } \mathbb{Z}_2^k\text{)}\\
&= \sum_{Z\in\mathbb{Z}_2^k} \left(\sum_{W\in\mathbb{Z}_2^k} (-1)^{X\cdot W} f(W)(-1)^{X\cdot Z}\chi_\Gamma(Z)\right) && \text{(from Lemma 2.1.3)}\\
&= \sum_{Z\in\mathbb{Z}_2^k} (-1)^{X\cdot Z}\chi_\Gamma(Z) \left(\sum_{W\in\mathbb{Z}_2^k} (-1)^{X\cdot W} f(W)\right)\\
&= \widehat{\chi_\Gamma}(X)\widehat{f}(X).
\end{aligned}$$

If $\widehat{\chi_\Gamma}(X) \neq 0$ for all $X \in \mathbb{Z}_2^k$, we can divide both sides by $\widehat{\chi_\Gamma}(X)$, giving

$$\widehat{f}(X) = \frac{\widehat{\Theta_\Gamma(f)}(X)}{\widehat{\chi_\Gamma}(X)},$$

and since the Fourier transform has a left inverse on $\mathcal{F}_k$ (by Lemma 2.1.7), we can therefore derive f from $\Theta_\Gamma(f)$ as follows:

$$\begin{aligned}
f(X) &= \frac{1}{2^k} \sum_{A\in\mathbb{Z}_2^k} (-1)^{X\cdot A}\widehat{f}(A)\\
&= \frac{1}{2^k} \sum_{A\in\mathbb{Z}_2^k} (-1)^{X\cdot A} \left(\frac{\widehat{\Theta_\Gamma(f)}(A)}{\widehat{\chi_\Gamma}(A)}\right)\\
&= \frac{1}{2^k} \sum_{A\in\mathbb{Z}_2^k} (-1)^{X\cdot A} \left(\frac{\sum_{Y\in\mathbb{Z}_2^k}(-1)^{A\cdot Y}\Theta_\Gamma(f)(Y)}{\widehat{\chi_\Gamma}(A)}\right). \qquad (2.4.1)
\end{aligned}$$

This shows that Θ_Γ has a left inverse if $\widehat{\chi_\Gamma}(X) \neq 0$ for all $X \in \mathbb{Z}_2^k$. ∎

We shall now explain the importance of Lemma 2.4.4 for the vertex-switching reconstruction problem. First we need a couple of preliminary lemmas.

Lemma 2.4.5 *Let G and H be two graphs on the vertex set V. Then $\mu_{\mathrm{SD}(G)}(H) = \mu_{\mathrm{SD}(H)}(G)$.*

Proof: For any $v \in V$, we have that $G \divideontimes v = H$ if and only if $H \divideontimes v = G$. Hence H occurs in $\mathrm{SD}(G)$ exactly as many times as G occurs in $\mathrm{SD}(H)$. ∎

Lemma 2.4.6 *Let G be a fixed graph on the vertex set V. Define $f_G \in \mathcal{F}_k$ by*

$$f_G(X) = \begin{cases} 1 & \text{if } X = X_G, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, let $\Gamma = \{ X_{C_n(v)} : v \in V \}$. Then, for any graph H on V, we have $\Theta_\Gamma(f_G)(X_H) = \mu_{\mathrm{SD}(G)}(H)$.

Proof: From the definition of Θ_Γ we have

$$\begin{aligned} \Theta_\Gamma(f_G)(X_H) &= \sum_{Y \in \Gamma} f_G(X_H + Y) \\ &= \sum_{v \in V} f_G(X_H + X_{C_n(v)}) \\ &= \sum_{v \in V} f_G(X_{H \divideontimes v}) \\ &= \mu_{\mathrm{SD}(H)}(G) \\ &= \mu_{\mathrm{SD}(G)}(H) \qquad \text{(by Lemma 2.4.5).} \end{aligned}$$

∎

Thus $\Theta_\Gamma(f_G)$ (with f_G defined as in Lemma 2.4.6) tells us how many copies of a given graph occur in the switch deck of G. Hence, $\Theta_\Gamma(f_G)(Y)$ can be found from $\mathrm{SD}(G)$ alone for every $Y \in \mathbb{Z}_2^k$. Because of this fact, if the left inverse of Θ_Γ exists then we can retrieve $f_G(X)$ for all $X \in \mathbb{Z}_2^k$, and thus the identity of G, from $\mathrm{SD}(G)$.

2.5 The Main Result

We now have essentially everything we need to prove the main result. One more detail is required first: a way to get from ϕ to Θ_Λ and back again.

Definition 2.5.1 Let $\Omega : \mathcal{F}_k \to \mathcal{V}_n$ be defined by $\Omega(f) = \sum_{i=1}^{r} f(X_i)X_i$, where $f \in \mathcal{F}_k$ and $\mathbb{Z}_2^k = \{X_1, X_2, \ldots, X_r\}$.

Lemma 2.5.2 *The mapping Ω is invertible.*

Proof: Define a mapping $A : \mathcal{V}_n \to \mathcal{F}_k$ by setting $A(W) = f_W$, where if $W = \sum_{i=1}^{r} c_i X_i$, then $f_W(X_i) = c_i$ for all $i \in \{1, 2, \ldots, r\}$. We will show that A is the inverse of Ω by showing that $\Omega(A(W)) = W$ for all $W \in \mathcal{V}_n$ and that $A(\Omega(f)) = f$ for all $f \in \mathcal{F}_k$.

Let f be any element of $\mathcal{F}_k$. Then $A(\Omega(f)) = A\left(\sum_{i=1}^{r} f(X_i)X_i\right) = f_2$ for some $f_2 \in \mathcal{F}_k$. By the definition of A, it follows that $f_2(X) = f(X)$ for all $X \in \mathbb{Z}_2^k$, and so $f_2 = f$. Therefore $A(\Omega(f)) = f$.

Now let $W = \sum_{i=1}^{r} c_i X_i \in \mathcal{V}_n$. Then $\Omega(A(W)) = \Omega(f)$, where $f \in \mathcal{F}_k$ and $f(X_i) = c_i$ for all $1 \leq i \leq r$. But $\Omega(f) = \sum_{i=1}^{r} f(X_i)X_i = \sum_{i=1}^{r} c_i X_i = W$, and so $\Omega(A(W)) = W$. Therefore $A = \Omega^{-1}$, and so Ω is invertible. ∎

And now, the main event at last.

Theorem 2.5.3 (Stanley [14]) *Let G be a graph with $\nu_G \not\equiv 0 \pmod 4$. Then G is VSR.*

Proof: First we shall show that ϕ has a left inverse whenever Θ_Γ does. We begin by noting that the mapping $\tau_C : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$, defined by $\tau_C(X) = X + C$ (for a fixed $C \in \mathbb{Z}_2^k$), is invertible, since $X + C + C = X$. Therefore, τ_C is bijective. Recall that Γ

is the set of graph vectors of all star graphs on V. Let $W \in \mathcal{V}_n$, and let $f = \Omega^{-1}(W)$. Then we observe the following.

$$\begin{aligned}
\phi(W) &= \phi\left(\sum_{X\in\mathbb{Z}_2^k} f(X)X\right) \\
&= \sum_{X\in\mathbb{Z}_2^k} f(X)\phi(X) && (\text{ since } \phi \text{ is linear}) \\
&= \sum_{X\in\mathbb{Z}_2^k} f(X)\left(\sum_{v\in V}(X + X_{C_n(v)})\right) && (\text{from the definition of } \phi) \\
&= \sum_{X\in\mathbb{Z}_2^k} f(X)\left(\sum_{C\in\Gamma}(X + C)\right) && (\text{where the sum } X + C \text{ is computed in } \mathbb{Z}_2^k) \\
&= \sum_{X\in\mathbb{Z}_2^k}\sum_{C\in\Gamma} f(X)(X + C) && (\text{distributive law, since } \mathcal{V}_n \text{ is a vector space})
\end{aligned}$$

Since τ_C is a bijection, $\tau_C(X)$ runs over all elements of $\mathbb{Z}_2^k$ as X does, and so we may replace X with $\tau_C(X)$, giving

$$\begin{aligned}
\phi(W) &= \sum_{X\in\mathbb{Z}_2^k}\sum_{C\in\Gamma} f(X + C)(X + C + C) \\
&= \sum_{X\in\mathbb{Z}_2^k}\sum_{C\in\Gamma} f(X + C)X && (\text{since } C + C = 0 \text{ in } \mathbb{Z}_2^k) \\
&= \sum_{X\in\mathbb{Z}_2^k}\left(\sum_{C\in\Gamma} f(X + C)\right)X \\
&= \sum_{X\in\mathbb{Z}_2^k}\Theta_\Gamma(f)(X)X.
\end{aligned}$$

Therefore, $\Omega^{-1}(\phi(W)) = \Theta_\Gamma(f)$, and so $\Omega^{-1}\phi = \Theta_\Gamma\Omega^{-1}$. Now assume Θ_Γ has a left inverse F. Then ϕ has a left inverse too, namely $\Omega F\Omega^{-1}$.

Next we show that ϕ, indeed, has a left inverse if $\nu_G \not\equiv 0 \pmod 4$. Let $n = \nu_G$, and assume $n \not\equiv 0 \pmod 4$. Suppose ϕ does not have a left inverse. Then, as shown above, Θ_Γ has no left inverse, and, by Lemma 2.4.4, $\widehat{\chi_\Gamma}(X) = 0$ for some $X \in \mathbb{Z}_2^k$. Recall from the proof of Lemma 2.4.4 that $\widehat{\chi_\Gamma}(X) = \sum_{Y \in \Gamma}(-1)^{X \cdot Y} = \sum_{v \in V}(-1)^{X \cdot C_n(v)}$. Since $X_G \cdot X_H$ counts the number of edges that graphs G and H have in common, we know that $X_G \cdot C_n(v_i) = d_G(v_i)$. Therefore $\widehat{\chi_\Gamma}(X_G) = \sum_{v \in V(G)} (-1)^{d_G(v)}$. For this expression to be 0, the number of even-degree vertices of G must equal the number of odd-degree vertices of G. Since the sum of the degrees of a graph is even, there must be an even number of vertices of odd degree, and therefore the same number of vertices of even degree, and so the order of G must be divisible by 4. This contradicts the assumption that $n \not\equiv 0 \pmod 4$. Therefore ϕ has a left inverse. Then by Corollary 2.3.7, all graphs on n vertices are VSR. ∎

2.6 Related Results

In a recent paper, Abatangelo and Dragomir [1, Corollary 2] show that if G and H are non-isomorphic VSE graphs, then there exists a graph vector X such that

$$|\operatorname{Aut}(G)| \sum_{G' \cong G} (-1)^{X \cdot X_{G'}} \neq |\operatorname{Aut}(H)| \sum_{H' \cong H} (-1)^{X \cdot X_{H'}}.$$

This appears to be, essentially, a computational test for the existence of an isomorphism which maps G to H. Abatangelo and Dragomir state that the utility of this result in the context of the vertex-switching reconstruction problem is unknown.

Much of the literature on vertex-switch reconstruction addresses the more general problem of reconstructing G from its s-switch deck, for all values of s. Stanley [14] extends the result of Theorem 2.5.3 to show that a graph G of order n is s-VSR if the Krawtchouk polynomial

$$p_s^n(x) = \sum_{i=0}^{s}(-1)^i \binom{x}{i}\binom{n-x}{s-i}$$

has no even integer zeros in the interval $[0, n]$. Note that for $s = 1$, $p_s^n(x) = n - 2x$, and so p_s^n has no even integer zeroes in $[0, n]$ if 4 does not divide n.

Krasikov and Roditty also give some results for $s \geq 4$ [10], and show that if $p_s^n(x)$ has one or two even integer roots sufficiently far from $n/2$, then any graph G of order n can be reconstructed from $\mathrm{SD}_s(G)$ [11]. In particular, they show that all graphs of order n are s-VSR if $n = 2s + k$, where $k = 0$, 1, or 3. Note that when $s = 1$ this result is subsumed by Theorem 2.5.3.

Chapter 3

Switch Partners

In this chapter, we see that if G and H are two non-VSR graphs of order not equal to 4 that are VSE but not isomorphic, then each vertex of G must have at least one H-switch partner (defined below). Switching on any vertex of G and one of its H-switch partners will produce a graph isomorphic to H. This result tells us quite a bit about the structure of such graphs, should any exist. Among other things, we get easily-computable necessary conditions when searching for non-VSR graphs. We also see that disconnected graphs and regular graphs are VSR.

3.1 Existence of Switch Partners

Definition 3.1.1 Let G and H be VSE graphs and let $v, w \in V(G)$. Then w is an *H-switch partner* of v in G if $G \circledast v \circledast w \cong H$. Note that if w is an H-switch partner of v in G, then v is an H-switch partner of w in G (since $G \circledast v \circledast w = G \circledast w \circledast v$), and in this case we say that v and w are an *H-switch pair* in G. When the graph H is understood or indeterminate, we may refer simply to a *switch pair* in G.

Krasikov [6] proved the following quite useful lemma.

Lemma 3.1.2 *Let G and H be VSE graphs on the same vertex set V with $\nu = |V| \neq 4$, and let $v \in V$. Then:*

(i) if $G \ncong H$, then v has an H-switch partner w in G such that $w \neq v$, and

(ii) for every H-switch partner w of v in G such that $w \neq v$, we have $d_G(v) + d_G(w) = \nu - 2 + 2e_G(v, w)$.

Proof:

(i) Since $\mathrm{SD}(G) \cong \mathrm{SD}(H)$, and $\mathrm{SD}(G)$ consists of graphs of the form $G \circledast x$, and $\mathrm{SD}(H)$ consists of graphs of the form $H \circledast x$, there must be some vertex $x \in V$ such that $G \circledast v \cong H \circledast x$. Then there exists an isomorphism $\varphi : H \circledast x \to G \circledast v$. By Lemma 1.4.12, $G \circledast v = \varphi(H \circledast x) = \varphi(H) \circledast \varphi(x)$. Therefore $G \circledast v \circledast \varphi(x) = \varphi(H)$, and so $G \circledast v \circledast \varphi(x) \cong H$, showing that $\varphi(x)$ is an H-switch partner of v in G. If $\varphi(x) = v$ then $H \cong G \circledast v \circledast \varphi(x) = G \circledast v \circledast v = G$, which contradicts the fact that $G \ncong H$. Therefore $\varphi(x) \neq v$.

(ii) Let w be an H-switch partner of v in G. Then switching on v in G removes $d_G(v)$ edges and adds $\nu - d_G(v) - 1$ new edges, and a subsequent switch on w in $G \circledast v$ removes $d_{G\circledast v}(w) = d_G(w) + 1 - 2e_G(v, w)$ edges and adds $\nu - d_{G\circledast v}(w) - 1 = \nu - d_G(w) + 2e_G(v, w) - 2$ new edges. Since $|E(G)| = |E(H)| = |E(G \circledast v \circledast w)|$ (by Lemma 1.3.5), we have

$$-d_G(v) + (\nu - d_G(v) - 1) - (d_G(w) + 1 - 2e_G(v, w)) + (\nu - d_G(w) + 2e(v, w) - 2) = 0.$$

Therefore,

$$2\nu - 2d_G(v) - 2d_G(w) + 4e_G(v, w) - 4 = 0,$$

that is, $d_G(v) + d_G(w) = \nu - 2 + 2e_G(v, w)$ as claimed.

∎

3.2 Neighbourhoods of Switch Pairs

Given an arbitrary switch pair in a non-VSR graph G, we can distinguish the vertices of G according to the number of neighbours they have within the switch pair. This outlook yields some interesting results that limit the structure of G.

Definition 3.2.1 Let G be a graph and let $v, w \in V(G)$. Then the *k-neighbourhood* of $\{v, w\}$ in G (written $N_G^k(\{v, w\})$) is the set of vertices in $V(G) \setminus \{v, w\}$ which are adjacent to exactly k members of $\{v, w\}$. For convenience, we may write $N_G^k(v, w)$ instead of $N_G^k(\{v, w\})$.

Note that for any G, v, and w as above,

$$\{\{v\}, \{w\}, N_G^0(v, w), N_G^1(v, w), N_G^2(v, w)\}$$

partitions $V(G)$.

We shall see below (Lemma 3.2.3) that switching on a switch pair has the effect of swapping the degrees of these two vertices and has no effect on the degree of any vertex that is adjacent to just one member of the switch pair. But first, we need a preliminary result.

Lemma 3.2.2 *Let G be a graph and let $v, w \in V(G)$ with $v \neq w$. Then $d_{G \circledast v \circledast w}(v) = \nu_G - d_G(v) + 2e_G(v, w) - 2$.*

Proof: Observe that $d_{G \circledast v}(v) = \nu_G - d_G(v) - 1$. Suppose $e_G(v, w) = 1$. Then $e_{G \circledast v}(v, w) = 0$, and so switching on w in $G \circledast v$ adds an edge that is incident with v. Thus $d_{G \circledast v \circledast w}(v) = d_{G \circledast v}(v) + 1 = \nu_G - d_G(v) = \nu_G - d_G(v) + 2e_G(v, w) - 2$. Now suppose $e_G(v, w) = 0$. Then $e_{G \circledast v}(v, w) = 1$, and so switching on w in $G \circledast v$ removes an edge that is incident with v. In this case, $d_{G \circledast v \circledast w}(v) = d_{G \circledast v}(v) - 1 = \nu_G - d_G(v) - 2 = \nu_G - d_G(v) + 2e_G(v, w) - 2$. Thus in all cases, $d_{G \circledast v \circledast w}(v) = \nu_G - d_G(v) + 2e_G(v, w) - 2$, as claimed. ∎

Lemma 3.2.3 *Let G and H be VSE graphs with $\nu_G \neq 4$. Let $v, w \in V(G)$ with $v \neq w$ such that $G \circledast v \circledast w \cong H$, and let $K = G \circledast v \circledast w$. Finally, let $r \in N_G^1(v, w)$. Then:*

(i) $d_K(v) = d_G(w)$ *(and* $d_K(w) = d_G(v)$*), and*

(ii) $d_K(r) = d_G(r)$.

Proof: Lemma 3.2.2 shows that $d_K(v) = \nu_G - d_G(v) + 2e_G(v, w) - 2$. Next, from Lemma 3.1.2, we know that $d_G(v) + d_G(w) = \nu - 2 + 2e_G(v, w)$. Therefore, $\nu - d_G(v) + 2e_G(v, w) - 2 = d_G(w)$, and so $d_K(v) = d_G(w)$. By symmetry, we have $d_K(w) = d_G(v)$, thus proving (i).

Since r is adjacent to exactly one of v and w, and since switching on both v and w in G results in deleting one edge incident with r and adjoining another, we have $d_G(r) = d_K(r)$, proving (ii). ∎

The following lemma, due to Ellingham and Royle [5], says that, if v and w are a switch pair of a non-VSR graph G, then the number of vertices adjacent to both v and w is the same as the number of vertices adjacent to neither v nor w. It also says that in this case there must be at least one vertex adjacent to both v and w (and therefore at least one vertex adjacent to neither v nor w). Both of these results are quite helpful in quickly eliminating potentially non-VSR graphs during an exhaustive search. Part (i) of this lemma was stated but not proven in [5].

Lemma 3.2.4 *Let G and H be VSE graphs with the same vertex set V. Let $v, w \in V$ be an H-switch pair in G with $v \neq w$. Then*

(i) $|N_G^0(v, w)| = |N_G^2(v, w)|$*, and*

(ii) if $G \ncong H$, then $|N_G^0(v,w)| > 0$.

Proof: We prove part (i) by observing that

$$\begin{aligned}
|N_G^0(v,w)| &= \sum_{x\in V\setminus\{v,w\}} (1-e_G(x,v))(1-e_G(x,w)) \\
&= \sum_{x\in V}(1-e_G(x,v))(1-e_G(x,w)) \\
&\qquad - (1-e_G(v,v))(1-e_G(v,w)) - (1-e_G(w,v))(1-e_G(w,w)) \\
&= \sum_{x\in V}(1-e_G(x,v)-e_G(x,w)+e_G(x,v)e_G(x,w)) - 2(1-e_G(v,w)) \\
&= \nu_G - d_G(v) - d_G(w) + \sum_{x\in V} e_G(x,v)e_G(x,w) - 2(1-e_G(v,w)) \\
&= \nu_G - (d_G(v)+d_G(w)) + \sum_{x\in V} e_G(x,v)e_G(x,w) - 2 + 2e_G(v,w),
\end{aligned}$$

which, since Lemma 3.1.2(ii) shows that $d_G(v)+d_G(w) = \nu_G - 2 + 2e_G(v,w)$, equals

$$\begin{aligned}
&\nu_G - (\nu_G - 2 + 2e_G(v,w)) + \sum_{x\in V} e_G(x,v)e_G(x,w) - 2 + 2e_G(v,w) \\
&= \sum_{x\in V} e_G(x,v)e_G(x,w) \\
&= \sum_{x\in V} e_G(x,v)e_G(x,w) - e_G(v,v)e_G(v,w) - e_G(w,v)e_G(w,w) \\
&= \sum_{x\in V\setminus\{v,w\}} e_G(x,v)e_G(x,w) \\
&= |N_G^2(v,w)|.
\end{aligned}$$

Next, we will show that $|N_G^0(v,w)| \neq 0$. Suppose $|N_G^0(v,w)| = 0$. Then (from the above) every vertex $x \in V\setminus\{v,w\}$ is adjacent to exactly one of v and w in G; that is, $e_G(x,v) = 1 - e_G(x,w)$. Define a permutation φ on V by $\varphi(v) = w$, $\varphi(w) = v$, and $\varphi(x) = x$ for all $x \in V\setminus\{v,w\}$, and let $K = G \circledast v \circledast w$. We shall show that φ is an isomorphism from K to G. Indeed,

$$e_{\varphi(K)}(v,x) = e_K(w,x) = 1 - e_{G\circledast v}(w,x) = 1 - e_G(w,x)$$

$$= e_G(v, x)$$

and, by symmetry,

$$e_{\varphi(K)}(w, x) = e_G(w, x)$$

for all $x \in V \setminus \{v, w\}$, and

$$e_{\varphi(K)}(v, w) = e_K(v, w) = 1 - e_{G \circledast v}(v, w) = 1 - (1 - e_G(v, w)) = e_G(v, w).$$

Hence $\varphi(K) = G$, i.e., φ is an isomorphism that maps K to G. Therefore $G \cong K$, and $K \cong H$ by the assumption of the lemma. Thus $G \cong H$, a contradiction. We conclude that $|N_G^0(v, w)| \neq 0$, proving part (ii). ∎

3.3 Neighbour Degrees

Since the degree sequence of a graph is VSR, switching on a switch pair of a non-VSR graph produces a graph with the same degree sequence. This can be used to place a condition on the degrees of the vertices which are adjacent to both or neither of the elements of the switch pair.

The next two rather technical lemmas will be used to prove a relationship between the sums of degrees of the vertices in $N_G^0(v, w)$ and $N_G^2(v, w)$, for a switch pair $\{v, w\}$.

Lemma 3.3.1 *Let X and Y be multisets of integers with $|X| = |Y|$. Let $X_+ = \langle x + 2 : x \in X \rangle$, and let $Y_- = \langle y - 2 : y \in Y \rangle$. If $X \uplus Y = X_+ \uplus Y_-$, then $X = Y_-$ and $Y = X_+$.*

Proof: The proof proceeds by induction on $|X|$. Suppose $|X| = 1$, and let $X = \langle x_1 \rangle$ and $Y = \langle y_1 \rangle$. Then $X_+ = \langle x_1 + 2 \rangle$ and $Y_- = \langle y_1 - 2 \rangle$. Furthermore, $X \uplus Y = \langle x_1, y_1 \rangle$ and $X_+ \uplus Y_- = \langle x_1 + 2, y_1 - 2 \rangle$. Now suppose $X \uplus Y = X_+ \uplus Y_-$.

Then since $x_1 \neq x_1+2$, we have $x_1 = y_1-2$. Therefore $X = Y_-$. Similarly, $y_1 = x_1+2$, and so $Y = X_+$. Therefore, the claim holds for $|X| = 1$.

Suppose that for all X, Y, X_+, and Y_- as above, when $|X| < n$, the assumption $X \uplus Y = X_+ \uplus Y_-$ implies $X = Y_-$ and $Y = X_+$. Now take any two multisets X and Y with $X \uplus Y = X_+ \uplus Y_-$ and $|X| = |Y| = n$. Let x be a minimum element of X. Since $X \uplus Y = X_+ \uplus Y_-$, we have $x \in X_+ \uplus Y_-$. Suppose $x \in X_+$. Then there exists some $x' \in X$ such that $x' + 2 = x$. This contradicts the fact that x is a minimum element of X. Therefore $x \in Y_-$, and so there is some $y \in Y$ such that $y - 2 = x$. Now let $X' = X \setminus \langle x \rangle$, $Y' = Y \setminus \langle y \rangle$, $X'_+ = X_+ \setminus \langle x+2 \rangle$, and $Y'_- = Y_- \setminus \langle y-2 \rangle$. It is not hard to see that $X' \uplus Y' = X'_+ \uplus Y'_-$. Since also $|X'| = |Y'| = n-1$, by the induction hypothesis we have that $X' = Y'_-$ and $Y' = X'_+$.

Now we note that

$$\begin{aligned} X &= X' \uplus \langle x \rangle = Y'_- \uplus \langle x \rangle && \text{(since } X' = Y'_-\text{)} \\ &= Y'_- \uplus \langle y - 2 \rangle \\ &= Y_-. \end{aligned}$$

Similarly,

$$Y = Y' \uplus \langle y \rangle = X'_+ \uplus \langle y \rangle = X'_+ \uplus \langle x+2 \rangle = X_+.$$

Therefore $X = Y_-$ and $Y = X_+$, as required, and the result follows by induction. ∎

Now we will see that, given a switch pair of a non-VSR graph, adding 2 to the degrees of each vertex that is adjacent to neither member of the switch pair produces the multiset of the degrees of the vertices adjacent to both members of the switch pair. This is another condition which fairly quickly can eliminate candidates from the search for a non-VSE graph.

Lemma 3.3.2 *Let G and H be non-isomorphic VSE graphs on the same set of vertices V with $|V| \neq 4$. Let $v, w \in V$ be an H-switch pair in G. Then there exists a bijection $\Pi : N_G^0(v,w) \to N_G^2(v,w)$ satisfying $\Pi(a) = b \Rightarrow d_G(b) = d_G(a) + 2$ for all $a \in N_G^0(v,w)$.*

Proof: Since $\nu_G \neq 4$, by Lemma 1.3.7, the degree sequences of G and H are equal. Let $K = G \circledast v \circledast w$. Since $K \cong H$, the degree sequences of G and K are equal as well.

Now let $X = \langle d_G(x) : x \in N_G^0(v,w) \rangle$, and let $Y = \langle d_G(y) : y \in N_G^2(v,w) \rangle$. Then let $X_K = \langle d_K(x) : x \in N_G^0(v,w) \rangle$, and let $Y_K = \langle d_K(y) : y \in N_G^2(v,w) \rangle$. Note that for all $x \in N_G^0(v,w)$, we have $d_K(x) = d_G(x) + 2$, and for all $y \in N_G^2(v,w)$, we have $d_K(y) = d_G(y) - 2$. Hence $X_K = \langle\ d + 2 : d \in X\ \rangle$ and $Y_K = \langle\ d - 2 : d \in Y\ \rangle$. Now recall that

$$\langle d_G(v) \rangle \uplus \langle d_G(w) \rangle \uplus \left\langle d_G(x) : x \in N_G^1(v,w) \right\rangle \uplus X \uplus Y = \langle d_G(x) : x \in V \rangle .$$

Therefore, since G and K have the same degree sequence (by Lemma 1.3.7),

$$\begin{aligned} &\langle d_G(v) \rangle \uplus \langle d_G(w) \rangle \uplus \left\langle d_G(x) : x \in N_G^1(v,w) \right\rangle \uplus X \uplus Y \\ &\quad = \langle d_K(v) \rangle \uplus \langle d_K(w) \rangle \uplus \left\langle d_K(x) : x \in N_G^1(v,w) \right\rangle \uplus X_+ \uplus Y_-. \end{aligned}$$

Lemma 3.2.3 shows that $d_G(x) = d_K(x)$ for all $x \in N_G^1(v,w)$, and so

$$\left\langle d_G(x) : x \in N_G^1(v,w) \right\rangle = \left\langle d_K(x) : x \in N_G^1(v,w) \right\rangle .$$

From the same lemma, we also know that $d_G(v) = d_K(w)$ and $d_K(v) = d_G(w)$. Therefore we are left with $X \uplus Y = X_+ \uplus Y_-$. Since $|X| = |Y|$ by Lemma 3.2.4(ii), Lemma 3.3.1 shows that $X = Y_-$. This means that for every element x of $N_G^0(v,w)$, there is a corresponding element y of $N_G^2(v,w)$ such that $d_G(x) = d_G(y) - 2$, and $\mu_X(d_G(x)) = \mu_Y(d_G(x) + 2)$.

We may now define a map $\Pi : N_G^0(v,w) \to N_G^2(v,w)$ as follows. For each $d \in X$, let $X_d = \{\, z : z \in N_G^0(v,w)$ and $d_G(z) = d\,\}$, and let $Y_d = \{\, z : z \in$

$N_G^2(v,w)$ and $d_G(z) = d+2\}$. Order the elements of X_d and Y_d arbitrarily, i.e., set $X_d = \{x_1, x_2, \ldots, x_k\}$ and $Y_d = \{y_1, y_2, \ldots, y_k\}$. Define $\Pi(x_i) = y_i$ for each $i \in \{1, 2, \ldots, k\}$. Then Π is defined on all of $N_G^0(v,w)$ and is a bijection satisfying $d_G(\Pi(x)) = d_G(x) + 2$ for all $x \in N_G^0(v,w)$. ∎

Corollary 3.3.3 *Let G and H be non-isomorphic VSE graphs with $\nu_G \neq 4$. Let $v \in V(G)$ and let w be an H-switch partner of v in G. Then*

$$\sum_{x \in N_G^2(v,w)} d_G(x) - \sum_{x \in N_G^0(v,w)} d_G(x) = 2|N_G^2(v,w)|.$$

Proof: Choose a bijection $\Pi : N_G^0(v,w) \to N_G^2(v,w)$ such that for all $a \in N_G^0(v,w)$, we have $d_G(a) = d_G(\Pi(a)) - 2$. Such a bijection exists by Lemma 3.3.2. Therefore,

$$\sum_{x \in N_G^0(v,w)} d_G(x) = \sum_{x \in N_G^0(v,w)} (d_G(\Pi(x)) - 2) = \sum_{x \in N_G^2(v,w)} (d_G(x) - 2),$$

and so

$$\begin{aligned}
\sum_{x \in N_G^2(v,w)} d_G(x) - \sum_{x \in N_G^0(v,w)} d_G(x) &= \sum_{x \in N_G^2(v,w)} d_G(x) - \sum_{x \in N_G^2(v,w)} (d_G(x) - 2) \\
&= \sum_{x \in N_G^2(v,w)} d_G(x) - \left(\sum_{x \in N_G^2(v,w)} d_G(x) - 2|N_G^2(v,w)| \right) \\
&= \sum_{x \in N_G^2(v,w)} (d_G(x) - d_G(x)) + 2|N_G^2(v,w)| \\
&= 2|N_G^2(v,w)|.
\end{aligned}$$

∎

We end this section by observing that when the vertices of a switch pair are switched on, the degrees of the neighbours of one member of the switch pair become the degrees of the neighbours of the other member of the switch pair.

Definition 3.3.4 Let G be a graph, and let $v \in V(G)$. Define

$$\mathrm{ND}_G(v) = \langle d_G(x) : x \in N_G(v) \rangle ,$$

i.e., $\mathrm{ND}_G(v)$ is the multiset composed of the degrees of the neighbours of v.

Lemma 3.3.5 *Let G and H be non-isomorphic VSE graphs with the same vertex set V of size $n \neq 4$. Let $v, w \in V$ such that $G \circledast v \circledast w \cong H$. Finally, let $K = G \circledast v \circledast w$. Then $\mathrm{ND}_G(v) = \mathrm{ND}_K(w)$.*

Proof: We begin by noting that $N_G(v) = N_G^2(v,w) \cup (N_G(v) \setminus N_G(w))$, and $N_K(w) = N_K^2(v,w) \cup (N_K(w) \setminus N_K(v))$. We will first show that the multiset of degrees of the vertices in $N_G^2(v,w)$ is equal to the multiset of degrees of the vertices in $N_K^2(v,w)$.

By Lemma 3.3.2, there is a bijection $\Pi : N_G^0(v,w) \to N_G^2(v,w)$ such that for all $a \in N_G^0(v,w)$, we have $d_G(a) + 2 = d_G(\Pi(a))$. Now let $a \in N_G^2(v,w)$. Then $d_G(a) = d_G(\Pi^{-1}(a)) + 2$. Since $\Pi^{-1}(a) \in N_G^0(v,w)$, we have $d_G(\Pi^{-1}(a)) = d_K(\Pi^{-1}(a)) - 2$. Thus $d_G(a) = d_K(\Pi^{-1}(a))$. Since Π^{-1} is a bijection, this shows that the multiset of degrees of the vertices in $N_G^2(v,w)$ is equal to the multiset of degrees of the vertices in $N_K^2(v,w)$.

Next, we will show that the multiset of degrees of the vertices in $N_G(v) \setminus N_G(w)$ is equal to the multiset of degrees of the vertices in $N_K(w) \setminus N_K(v)$. First, note that $(N_G(v) \setminus N_G(w)) \setminus \{w\} = (N_K(w) \setminus N_K(v)) \setminus \{v\}$, since a vertex (other than w) that is adjacent to v and not w in G must be adjacent to w and not v in K. Observe that for every vertex $x \in (N_G(v) \setminus N_G(w)) \setminus \{w\}$, we have $d_G(x) = d_{G \circledast v}(x) + 1 = d_K(x)$. If $w \in N_G(v) \setminus N_G(w)$ (i.e., if $vw \in E(G)$), then $v \in N_K(w) \setminus N_K(v)$. In this case, part (i) of Lemma 3.2.3 shows that $d_G(w) = d_K(v)$. Therefore, the multiset of degrees of the vertices in $N_G(v) \setminus N_G(w)$ is equal to the multiset of degrees of the vertices in $N_K(w) \setminus N_K(v)$. Thus, using the result of the previous paragraph, we may conclude that $\mathrm{ND}_G(v) = \mathrm{ND}_K(w)$. ∎

3.4 Some VSR Graphs

We conclude this chapter with some substantial results. Disconnected graphs and regular graphs are shown to be VSR.

Definition 3.4.1 A graph G is *connected* if for any partition of $V(G)$ into two non-empty subsets A and B there is at least one edge with one endpoint in A and the other endpoint in B. If a graph is not connected, then it is *disconnected*, i.e., there is a partition of $V(G)$ into two non-empty subsets A and B such that no edge of G has one endpoint in A and the other endpoint in B.

Definition 3.4.2 Let G be a graph and let $U \subseteq V(G)$. The subgraph of G *induced* by U (denoted $G[U]$) is the graph whose vertex set is U and whose edge set consists of every edge of G that has both endpoints in U.

Definition 3.4.3 If G is a graph, then C is a *connected component* of G if C is a connected graph, $C = G[S]$ for some $S \subseteq V(G)$, and there are no edges of G with one endpoint in S and the other endpoint in $V(G) \setminus S$.

Theorem 3.4.4 below was proven by Krasikov [6], but the following simpler proof, which relies on Lemma 3.2.4, appears in Ellingham and Royle [5].

Theorem 3.4.4 *Let G be a disconnected graph G with $\nu_G \neq 4$. Then G is VSR.*

Proof: Suppose that G is not VSR. Then there is some graph $H \ncong G$ such that H and G are VSE. Let $v \in V(G)$ be in a connected component of G with a minimum number of vertices, and let w be an H-switch partner of v. (Note that w exists and $w \neq v$, by Lemma 3.1.2.) Since part (ii) of Lemma 3.2.4 shows that $|N_G^2(v,w)| > 0$, there is some $x \in V(G)$ such that $vx, wx \in E(G)$. Therefore v and w must lie in the same connected component of G. Since, by part (ii) of Lemma 3.1.2, we have $d_G(v) + d_G(w) = \nu_G - 2 + 2e_G(v, w)$, there must be exactly $\nu_G - 2$ edges of G with exactly one endpoint in $\{v, w\}$. This means there are least $(\nu_G - 2)/2$ distinct vertices at the other end of these edges. These vertices must be in the same component of G that contains v and w. Therefore there are at least $\nu_G/2 + 1$ vertices in this component. This is impossible since the component containing v (being smallest) can have at most $\nu_G/2$ vertices. Therefore, G is VSR. ∎

Definition 3.4.5 A *regular graph* is a graph in which all vertices have the same degree. If $d_G(v) = k$ for all $v \in V(G)$, then we say that G is *k-regular*.

We now consider a simplified proof of a result due to Ellingham and Royle [5].

Theorem 3.4.6 *Let G be a regular graph on n vertices, where $n \neq 4$. Then G is VSR.*

Proof: Suppose G is not VSR. Then there is some graph H with $V(H) = V(G)$ such that $G \ncong H$ and such that G and H are VSE. Let $v \in V(G)$. By Lemma 3.1.2, there exists some $w \in V(G)$ such that $G \circledast v \circledast w \cong H$. Since $G \ncong H$, part (ii) of Lemma 3.2.4 shows that there is some $x \in V(G)$ such that $x \in N^0_G(v, w)$. This means that $d_{G \circledast v \circledast w}(x) = d_G(x) + 2$. Since $G \circledast v \circledast w \cong H$, this means that H has a vertex of degree $d_G(x) + 2$. However, Lemma 1.3.7 shows that the degree sequences of G and H are equal. Since G is regular, $d_G(y) = d_G(z)$ for all $y, z \in V(G)$, and consequently $d_H(x) = d_G(x)$ for all $x \in V(G)$. This contradicts the fact that H has a vertex of degree $d_G(x) + 2$. Therefore G is VSR. ∎

Chapter 4

Vertex-Switch Balance Equations

In this chapter, we review the known bounds on the number of edges and on the degrees of the vertices of a non-VSR graph, and then improve the published bounds on the number of edges. We also give an upper bound on the order of the automorphism group of a non-VSR graph. These results are obtained by examining a set of "balance equations", developed by (and named by) Krasikov and Roditty [9]. These equations in turn are developed by considering a pair of non-isomorphic VSE graphs G and H, and counting the number of graphs isomorphic to G or H that are obtained from the switch cards of G by switching on all possible k-subsets of vertices.

4.1 The Balance Equations

First, we develop an expression that describes the result of switching on all k-vertex subsets of the members of a graph's switch deck.

Lemma 4.1.1 (Krasikov and Roditty [9, Lemma 2.1])

Let G be a graph with n vertices, and let k be a positive integer. Then

$$\mathrm{SD}_k(\mathrm{SD}(G)) = \biguplus_{i=1}^{n-k+1} \mathrm{SD}_{k-1}(G) \uplus \left(\biguplus_{i=1}^{k+1} \mathrm{SD}_{k+1}(G) \right).$$

Proof: Every element of $\mathrm{SD}_k(\mathrm{SD}(G))$ is the result of switching on a single vertex of G and then switching on a set of k vertices. Therefore,

$$\begin{aligned}
\mathrm{SD}_k(\mathrm{SD}(G)) &= \biguplus_{\substack{S\subseteq V(G)\\|S|=k}} \left(\biguplus_{v\in V(G)} G \divideontimes v \divideontimes S \right) \\
&= \biguplus_{\substack{S\subseteq V(G)\\|S|=k}} \left(\left(\biguplus_{v\in S} G \divideontimes v \divideontimes S \right) \uplus \left(\biguplus_{v\in V(G)\setminus S} G \divideontimes v \divideontimes S \right) \right) \\
&= \biguplus_{\substack{S\subseteq V(G)\\|S|=k}} \left(\left(\biguplus_{v\in S} G \divideontimes (S\setminus\{v\}) \right) \uplus \left(\biguplus_{v\in V(G)\setminus S} G \divideontimes (S\cup\{v\}) \right) \right) \\
&= \biguplus_{\substack{S\subseteq V(G)\\|S|=k}} \left(\biguplus_{v\in S} G \divideontimes (S\setminus\{v\}) \right) \uplus \biguplus_{\substack{S\subseteq V(G)\\|S|=k}} \left(\biguplus_{v\in V(G)\setminus S} G \divideontimes (S\cup\{v\}) \right)
\end{aligned} \tag{4.1.1}$$

Now,

$$\begin{aligned}
\biguplus_{\substack{S\subseteq V(G)\\|S|=k}} \left(\biguplus_{v\in S} G \divideontimes (S\setminus\{v\}) \right) &= \biguplus_{\substack{T\subseteq V(G)\\|T|=k-1}} \left(\biguplus_{v\in V(G)\setminus T} G \divideontimes T \right) \\
&= \biguplus_{\substack{T\subseteq V(G)\\|T|=k-1}} \left(\biguplus_{i=1}^{n-(k-1)} G \divideontimes T \right) \\
&= \biguplus_{i=1}^{n-(k-1)} \left(\biguplus_{\substack{T\subseteq V(G)\\|T|=k-1}} G \divideontimes T \right) \\
&= \biguplus_{i=1}^{n-k+1} \mathrm{SD}_{k-1}(G),
\end{aligned}$$

and

$$\biguplus_{\substack{S\subseteq V(G)\\|S|=k}} \left(\biguplus_{v\in V(G)\setminus S} G \divideontimes (S\cup\{v\}) \right) = \biguplus_{\substack{T\subseteq V(G)\\|T|=k+1}} \left(\biguplus_{v\in T} G \divideontimes T \right)$$

$$= \biguplus_{\substack{T \subseteq V(G) \\ |T|=k+1}} \left(\biguplus_{i=1}^{k+1} G \circledast T \right)$$

$$= \biguplus_{i=1}^{k+1} \left(\biguplus_{\substack{T \subseteq V(G) \\ |T|=k+1}} G \circledast T \right)$$

$$= \biguplus_{i=1}^{k+1} \mathrm{SD}_{k+1}(G).$$

Then, by combining these last two results with Equation (4.1.1), we get

$$\mathrm{SD}_k(\mathrm{SD}(G)) = \biguplus_{i=1}^{n-k+1} \mathrm{SD}_{k-1}(G) \uplus \left(\biguplus_{i=1}^{k+1} \mathrm{SD}_{k+1}(G) \right).$$

∎

Next, we define notation for the number of graphs of a particular isomorphism class in the s-switch deck of a graph.

Definition 4.1.2 Let G and H be graphs with $V(G) = V(H)$, and let s be a non-negative integer. For $s \geq 1$, define $I_s(G \to H)$ as the number of graphs isomorphic to H which are obtained by switching on any set of s vertices in G. We define $I_0(G \to H) = 1$ if $G \cong H$, and 0 otherwise.

Now we may establish the balance equations, which relate the values of $I_s(G \to H) - I_s(H \to H)$ for various values of s. These equations eventually lead to bounds on the number of edges and on vertex degrees of a non-VSR graph.

Lemma 4.1.3 (Krasikov and Roditty [9, Theorem 2.2])

Let G and H be non-isomorphic VSE graphs on the same set of n vertices. Then for all $k \in \{2, 3, \ldots, n\}$, we have

$$I_k(H \to H) - I_k(G \to H) = -\frac{n-k+2}{k} \Big(I_{k-2}(H \to H) - I_{k-2}(G \to H) \Big).$$

Proof: Since G and H are VSE, we have $\mathrm{SD}(G) \cong \mathrm{SD}(H)$ and so $\mathrm{SD}_{k-1}(\mathrm{SD}(G)) \cong \mathrm{SD}_{k-1}(\mathrm{SD}(H))$. Therefore the sum of the multiplicities in $\mathrm{SD}_{k-1}(\mathrm{SD}(G))$ of all graphs isomorphic to H is the same as the sum of the multiplicities in $\mathrm{SD}_{k-1}(\mathrm{SD}(H))$ of all graphs isomorphic to H. From Lemma 4.1.1, since

$$\mathrm{SD}_{k-1}(\mathrm{SD}(G)) = \biguplus_{i=1}^{n-k+2} \mathrm{SD}_{k-2}(G) \uplus \left(\biguplus_{i=1}^{k} \mathrm{SD}_k(G) \right),$$

the sum of the multiplicities in $\mathrm{SD}_{k-1}(\mathrm{SD}(H))$ of all graphs isomorphic to H is $(n - k + 2)I_{k-2}(H \rightarrow H) + kI_k(H \rightarrow H))$. Similarly, the sum of the multiplicities in $\mathrm{SD}_{k-1}(\mathrm{SD}(G))$ of all graphs isomorphic to H is $(n-k+2)I_{k-2}(G\rightarrow H)+kI_k(G\rightarrow H)$. Therefore,

$$\begin{aligned}(n-k+2)I_{k-2}(H\rightarrow H) + kI_k(H\rightarrow H))\\ = (n-k+2)I_{k-2}(G\rightarrow H) + kI_k(G\rightarrow H),\end{aligned}$$

and so

$$k\Big(I_k(H\rightarrow H) - I_k(G\rightarrow H)\Big) = (n-k+2)\Big(I_{k-2}(G\rightarrow H) - I_{k-2}(H\rightarrow H)\Big)$$

that is,

$$I_k(H\rightarrow H) - I_k(G\rightarrow H) = -\frac{n-k+2}{k}\Big(I_{k-2}(H\rightarrow H) - I_{k-2}(G\rightarrow H)\Big)$$

as claimed. ∎

By iteratively applying the result of Lemma 4.1.3, we can get a simple closed-form expression for $I_k(H\rightarrow H) - I_k(G\rightarrow H)$ for even values of k.

Lemma 4.1.4 (Krasikov and Roditty [9, Corollary 2.4])

Let G and H be non-isomorphic graphs VSE on the same set of n vertices. Then for all $k \in \{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\}$,

$$I_{2k}(H\rightarrow H) - I_{2k}(G\rightarrow H) = (-1)^k \binom{\frac{n}{2}}{k}.$$

Proof: By Lemma 4.1.3, we have, for all $k \in \{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\}$,

$$\begin{aligned} I_{2k}(H \to H) - I_{2k}(G \to H) &= -\frac{n - 2k + 2}{2k}\Big(I_{2k-2}(H \to H) - I_{2k-2}(G \to H)\Big) \\ &= -\frac{\frac{n}{2} - k + 1}{k}\Big(I_{2k-2}(H \to H) - I_{2k-2}(G \to H)\Big). \end{aligned}$$

Now we can recursively apply this equation, getting

$$\begin{aligned} &I_{2k}(H \to H) - I_{2k}(G \to H) \\ &\quad = (-1)\frac{\frac{n}{2} - k + 1}{k}(-1)\frac{\frac{n}{2} - k + 2}{k - 1}\Big(I_{2k-4}(H \to H) - I_{2k-4}(G \to H)\Big) \\ &\quad = (-1)\frac{\frac{n}{2} - k + 1}{k}(-1)\frac{\frac{n}{2} - k + 2}{k - 1}\cdots(-1)\frac{\frac{n}{2}}{1}\Big(I_0(H \to H) - I_0(G \to H)\Big) \\ &\quad = (-1)^k\binom{\frac{n}{2}}{k}\Big(I_0(H \to H) - I_0(G \to H)\Big) \\ &\quad = (-1)^k\binom{\frac{n}{2}}{k}(1 - 0) \\ &\quad = (-1)^k\binom{\frac{n}{2}}{k}. \end{aligned}$$

∎

The preceding equations will lead us to a simple alternative proof of Theorem 2.5.3. But first, we prove a simple lemma that results from the fact that switching on a set of vertices has the same effect as switching on the complement of the set.

Lemma 4.1.5 (Krasikov and Roditty [9])

Let G and H be non-isomorphic VSE graphs on the same set of n vertices. Then, for all $k \in \{0, 1, \ldots, n\}$, we have

$$I_k(H \to H) - I_k(G \to H) = I_{n-k}(H \to H) - I_{n-k}(G \to H).$$

Proof: By Lemma 1.4.4, switching on a subset S of $V(G)$ in G produces the same result as switching on $V(G) \setminus S$ in G. Therefore, for all $k \in \{0, 1, \ldots, n\}$, we have

$I_k(H \to H) = I_{n-k}(H \to H)$ and $I_k(G \to H) = I_{n-k}(G \to H)$. Thus,

$$I_k(H \to H) - I_k(G \to H) = I_{n-k}(H \to H) - I_{n-k}(G \to H).$$

∎

And now we have another technical lemma, to be used in the proof of Theorem 4.1.7 below.

Lemma 4.1.6 *Let n be an odd integer. Then $\binom{\frac{n}{2}}{k}$ is not an integer for all $k \in \{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\}$.*

Proof: Observe that

$$\begin{aligned} \binom{\frac{n}{2}}{k} &= \frac{\frac{n}{2}(\frac{n}{2}-1)(\frac{n}{2}-2)\cdots(\frac{n}{2}-k+1)}{k!} \\ &= \frac{1}{2^k}\frac{n(n-2)(n-4)\cdots(n-2k+2)}{k!} \end{aligned}$$

If $\binom{\frac{n}{2}}{k}$ is an integer, then 2^k divides $n(n-2)(n-4)\cdots(n-2k-2)$, and so this product must be even. Hence n must be even. ∎

Theorem 4.1.7 (Krasikov and Roditty [9, Corollary 2.4])

Let G be a graph on n vertices, with $n \not\equiv 0 \pmod 4$. Then G is VSR.

Proof: Suppose G is not VSR. Then there exists some graph H with $V(G) = V(H)$ and $G \not\cong H$ such that H and G are VSE. By Lemma 4.1.4, for all $k \in \{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\}$, we have

$$I_{2k}(H \to H) - I_{2k}(G \to H) = (-1)^k \binom{\frac{n}{2}}{k}.$$

Since $I_{2k}(H \to H) - I_{2k}(G \to H)$ is an integer, by Lemma 4.1.6, n is even. Now by applying the result of Lemma 4.1.5, we get

$$I_{2k}(H \to H) - I_{2k}(G \to H) = I_{n-2k}(H \to H) - I_{n-2k}(G \to H),$$

and so (since $n - 2k$ is even), by Lemma 4.1.4,

$$\begin{aligned} (-1)^k \binom{\frac{n}{2}}{k} &= (-1)^{\frac{n}{2}-k} \binom{\frac{n}{2}}{\frac{n}{2} - k} \\ (-1)^k &= (-1)^{\frac{n}{2}-k} \\ (-1)^{k-(\frac{n}{2}-k)} &= 1 \\ (-1)^{2k-\frac{n}{2}} &= 1. \end{aligned}$$

This means that $2k - \frac{n}{2}$ is even, and therefore $\frac{n}{2}$ is even, or in other words, $n \equiv 0 \pmod 4$. This contradicts the fact that $n \not\equiv 0 \pmod 4$, and so G is VSR. ∎

4.2 Bounds on the Size of a Non-VSR Graph

Krasikov [7, Theorem 7] gives a bound on the number of edges in a non-VSR graph G with $\nu_G \neq 4$, which is described in the following lemma and its corollary.

Lemma 4.2.1 *Let $n \equiv 0 \pmod 4$, $n \neq 4$, and let G be a non-VSR graph of order n. Then there is a set U of $\frac{n}{2}$ vertices of G such that there are exactly $\frac{1}{8}n^2$ edges of G with exactly one endpoint in U.*

Proof: Since G is non-VSR, there is some graph H that is not isomorphic to G but is VSE to G. Recall Lemma 4.1.4, which states that $I_{2k}(G \to G) - I_{2k}(H \to G) = (-1)^k \binom{\frac{n}{2}}{k}$ for all $k \in \{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\}$. Setting $k = \frac{n}{4}$ here gives $I_{\frac{n}{2}}(G \to G) - I_{\frac{n}{2}}(H \to G) = (-1)^{\frac{n}{4}} \binom{\frac{n}{2}}{\frac{n}{4}}$. Since $\binom{\frac{n}{2}}{\frac{n}{4}} \neq 0$, there is some set X of $\frac{n}{2}$ vertices in $V(G)$ such that either $G \divideontimes X \cong G$ or $H \divideontimes X \cong G$. First suppose that $G \divideontimes X \cong G$. Then clearly

$\varepsilon_{G \circledast X} = \varepsilon_G$. Now suppose $H \circledast X \cong G$. Then let ϕ be an isomorphism from G to $H \circledast X$, and let $X' = \phi^{-1}(X)$, so that $H = \phi(G) \circledast X = \phi(G \circledast X')$ (by Lemma 1.4.12), giving $G \circledast X' = \phi^{-1}(H) \cong H$. Since $n \neq 4$, Lemma 1.3.5 tells us that $\varepsilon_G = \varepsilon_H$. We therefore have $\varepsilon_{G \circledast X'} = \varepsilon_H = \varepsilon_G$. Thus in all cases there is a set U (either X or X') such that $\varepsilon_{G \circledast U} = \varepsilon_G$.

Switching on all of the vertices of U in G does not change any of the edges with both endpoints in U or with both endpoints in $V(G) \setminus U$. Therefore, the number of edges with one endpoint in U and the other endpoint in $V(G) \setminus U$ must be the same in G and in $G \circledast U$. Let m be the number of edges of G with one endpoint in U and the other endpoint in $V(G) \setminus U$. Then the number of edges of $G \circledast U$ with one endpoint in U and the other endpoint in $V(G) \setminus U$ is $|U| \cdot (n - |U|) - m = \frac{n}{2}(n - \frac{n}{2}) - m = \frac{1}{4}n^2 - m$. Since $m = \frac{1}{4}n^2 - m$, we have $m = \frac{1}{8}n^2$. Thus, there are exactly $\frac{1}{8}n^2$ edges with exactly one endpoint in U. ∎

Corollary 4.2.2 *Let $n \equiv 0 \pmod 4$, $n \neq 4$, and let G be a non-VSR graph of order n. Then $\frac{1}{8}n^2 \leq \varepsilon_G \leq \binom{n}{2} - \frac{1}{8}n^2$.*

Proof: Lemma 4.2.1 shows that there is a set U of vertices of G such that $E(G)$ contains $\frac{1}{8}n^2$ edges with exactly one endpoint in U. Therefore $\frac{1}{8}n^2 \leq \varepsilon_G$.

Now Corollary 1.4.11 tells us that, since G is non-VSR, its complement $\overline{G}$ is non-VSR as well, and so $\varepsilon_{\overline{G}} \geq \frac{1}{8}n^2$. But $\varepsilon_{\overline{G}} = \binom{n}{2} - \varepsilon_G$, and so $\binom{n}{2} - \varepsilon_G \geq \frac{1}{8}n^2$. Therefore $\varepsilon_G \leq \binom{n}{2} - \frac{1}{8}n^2$. ∎

Krasikov and Roditty [9] describe a stricter bound on ε_G, but their proof of this result contains a flaw, which was discovered by Ellingham and Royle [5]. However, a better bound than that of Corollary 4.2.2 is still possible. The following theorem, which is original to this thesis, improves Krasikov's upper and lower bounds on the number of edges of a non-VSR graph of order n by $\frac{1}{4}n-2$ edges.

Theorem 4.2.3 *Let $n \equiv 0 \pmod 4$, $n \neq 4$, and let G be a non-VSR graph of order n. Then $\frac{1}{8}n^2+\frac{1}{4}n-2 \leq \varepsilon_G \leq \binom{n}{2}-\frac{1}{8}n^2-\frac{1}{4}n+2$.*

Proof: We begin by observing that $n \geq 8$ since $n \equiv 0 \pmod 4$ and $n \neq 4$. We will first show that $\varepsilon_G \geq \frac{1}{8}n^2+\frac{1}{4}n-2$.

Lemma 4.2.1 shows that there is a set U of vertices of G such that $E(G)$ contains $\frac{1}{8}n^2$ edges with exactly one endpoint in U and $|U| = \frac{n}{2}$. Let p be the number of edges of G with both endpoints in U, and let q be the number of edges of G with both endpoints in $V(G) \setminus U$. Then $\varepsilon_G = \frac{1}{8}n^2+p+q$. Let $p \geq q$ without loss of generality. Suppose $p > \frac{1}{2}n-1$. Then $\varepsilon_G > \frac{1}{8}n^2+\frac{1}{2}n-1 > \frac{1}{8}n^2+\frac{1}{4}n-2$, since $n > 2$. Now suppose $p \leq \frac{1}{2}n-1$. Then the maximum degree of the vertices in U is at most $\frac{1}{2}n+p$, since a vertex in U can be an endpoint of at most $\frac{1}{2}n$ edges whose other endpoint is not in U, and of at most p edges whose other endpoint is in U. Similarly, the maximum degree of vertices in $V \setminus U$ is at most $\frac{1}{2}n+q \leq \frac{1}{2}n+p$. Let v be a vertex of G of minimum degree. Since G is not VSR, there must be some graph H which is VSE to G but not isomorphic to G. Then Lemma 3.1.2 tells us that there is some $w \in V(G)$ such that $d_G(v)+d_G(w) = n-2+2e_G(v,w) \geq n-2$. Therefore, $d_G(v) \geq n-2-d_G(w)$, and since $d_G(w) \leq \frac{1}{2}n+p$, we have $d_G(v) \geq n-2-\frac{1}{2}n-p = \frac{1}{2}n-p-2$. Now consider the vertices of $V(G) \setminus U$. Each of these vertices has degree at least $\frac{1}{2}n-p-2$, and there are $\frac{1}{2}n$ of them. Thus the sum of the degrees of the members of $V(G) \setminus U$ is at least $\frac{1}{2}n(\frac{1}{2}n-p-2) = \frac{1}{4}n^2-\frac{1}{2}np-n$. Each edge of G with both endpoints in $V(G) \setminus U$ contributes 2 to this sum. Therefore, the number of edges with exactly one endpoint in $V(G) \setminus U$ is at least $\frac{1}{4}n^2-\frac{1}{2}np-n-2q$. However, the number of these edges is exactly $\frac{1}{8}n^2$, and so

$$\frac{1}{4}n^2-\frac{1}{2}np-n-2q \leq \frac{1}{8}n^2$$
$$\frac{1}{8}n^2-\frac{1}{2}np-n-2q \leq 0$$

$$\begin{aligned} \frac{1}{2}np &\geq \frac{1}{8}n^2 - n - 2q \\ p &\geq \frac{1}{4}n - 2 - \frac{4q}{n}. \end{aligned}$$

Since $n \equiv 0 \pmod 4$ and $n > 4$, let $n = 4k$, where $k > 1$. Then $p \geq \frac{1}{4}n - 2 - \frac{q}{k}$, and so $\varepsilon_G \geq \frac{1}{8}n^2 + \frac{1}{4}n - 2 - \frac{q}{k} + q$. Since $k > 1$, we have $-\frac{q}{k} + q \geq \frac{1}{2}q \geq 0$, and thus in all cases $\varepsilon_G \geq \frac{1}{8}n^2 + \frac{1}{4}n - 2$.

Now Corollary 1.4.11 tells us that since G is non-VSR, $\overline{G}$ is non-VSR as well, and so $\varepsilon_{\overline{G}} \geq \frac{1}{8}n^2 + \frac{1}{4}n - 2$. But $\varepsilon_{\overline{G}} = \binom{n}{2} - \varepsilon_G$, and so $\binom{n}{2} - \varepsilon_G \geq \frac{1}{8}n^2 + \frac{1}{4}n - 2$. Therefore $\varepsilon_G \leq \binom{n}{2} - \frac{1}{8}n^2 - \frac{1}{4}n + 2$. ∎

4.3 Degree Bounds

Krasikov [7] also establishes a bound on the minimum and maximum degree of a non-VSR graph. This result is somewhat more involved, and requires some preliminary work. We begin by noting that, if G and H are non-isomorphic VSE graphs, and i is odd, then the i-switch decks of G and H contain equal numbers of graphs isomorphic to H, as shown by the following lemma.

Lemma 4.3.1 *Let G and H be non-isomorphic VSE graphs on the same set of n vertices. Then, for all odd $i \in \{1, 2, \ldots, n\}$, we have $I_i(H \to H) - I_i(G \to H) = 0$.*

Proof: We will use induction on i. Suppose there exists $v \in V(H)$ such that $H \circledast v \cong H$. Then, since $\varepsilon_H = \varepsilon_{H \circledast v} = \varepsilon_H - d_H(v) + (n - 1 - d_H(v))$, we must have $d_H(v) = n - d_H(v) - 1$. Therefore $d_H(v) = \frac{n-1}{2}$. This is impossible, since n is even (by Theorem 2.5.3). Therefore, there is no such v, and so $I_1(H \to H) = 0$. Similarly, suppose there is some $v \in V(G)$ such that $G \circledast v \cong H$. If $n \neq 4$, then $\varepsilon_G = \varepsilon_H$ (by Lemma 1.3.5), and so again we get $d_G(v) = \frac{n-1}{2}$, an impossibility as n is even.

Therefore $n = 4$. By checking Figure 1.3 (on page 13), we see that there is no row of the table which contains a graph K in the first column and a graph isomorphic to K in the second column. This fact shows there is no vertex v such that $G \circledast v \cong H$, and so $I_1(G \to H) = 0$. Therefore $I_1(H \to H) - I_1(G \to H) = 0 - 0 = 0$.

Now let i be an odd element of $\{1, 2, \ldots, n\}$, $i > 1$. Suppose $I_k(H \to H) - I_k(G \to H) = 0$ for all odd $k < i$. Lemma 4.1.3 says that

$$I_i(H \to H) - I_i(G \to H) = -\frac{n - i + 2}{i}\Big(I_{i-2}(H \to H) - I_{i-2}(G \to H)\Big),$$

and since $I_{i-2}(H \to H) - I_{i-2}(G \to H) = 0$ (by the induction hypothesis, as $i - 2$ is odd),

$$I_i(H \to H) - I_i(G \to H) = 0.$$

Thus, by induction, $I_i(H \to H) - I_i(G \to H) = 0$ for all odd $i \in \{1, 2, \ldots, n\}$. ∎

Next we note that for any graph G, the number of isomorphs of G in the collection of all k-switch decks of some graph H is the same for all graphs H that are VSE to G and not isomorphic to G.

Lemma 4.3.2 *Let G and H be non-isomorphic VSE graphs. Then*

$$\sum_{i=0}^{n} I_i(G \to G) = \sum_{i=0}^{n} I_i(H \to G).$$

Proof: First, since Lemma 4.3.1 shows that $I_i(G \to G) - I_i(H \to G) = 0$ for all odd $i \in \{1, 2, \ldots, n\}$, we have

$$\sum_{i=1}^{n}(I_i(G \to G) - I_i(H \to G)) = \sum_{i=1}^{n/2}(I_{2i}(G \to G) - I_{2i}(H \to G)).$$

Lemma 4.1.4 gives $I_{2k}(G \to G) - I_{2k}(H \to G) = (-1)^k \binom{\frac{n}{2}}{k}$ for all $k \in \{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\}$, and so

$$\sum_{i=0}^{n}\Big(I_i(G \to G) - I_i(H \to G)\Big) = \sum_{i=1}^{\frac{n}{2}}(-1)^i \binom{\frac{n}{2}}{i} + (1 - 0)$$

$$= \sum_{i=0}^{\frac{n}{2}} (-1)^i \binom{\frac{n}{2}}{i}$$
$$= (1-1)^{\frac{n}{2}}$$
$$= 0.$$

Therefore,

$$\sum_{i=0}^{n} I_i(G \to G) = \sum_{i=0}^{n} I_i(H \to G).$$

∎

Our next few lemmas pave the way for Theorem 4.3.8, which gives a bound on the extremal degrees of a non-VSR graph.

Lemma 4.3.3 (Krasikov [7, Theorem 8]) *Let G and H be non-isomorphic VSE graphs. Then* $\sum_{i=0}^{n} |I_i(G \to G) - I_i(H \to G)| = 2^{\frac{n}{2}}$.

Proof: Lemma 4.1.4 says that, for all $k \in \{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\}$,

$$I_{2k}(G \to G) - I_{2k}(H \to G) = (-1)^k \binom{\frac{n}{2}}{k},$$

and so

$$|I_{2k}(G \to G) - I_{2k}(H \to G)| = \binom{\frac{n}{2}}{k} \quad \text{and}$$
$$\sum_{k=1}^{\frac{n}{2}} |I_{2k}(G \to G) - I_{2k}(H \to G)| = \sum_{k=1}^{\frac{n}{2}} \binom{\frac{n}{2}}{k}.$$

Now recall Lemma 4.3.1, which shows that $I_i(G \to G) - I_i(H \to G) = 0$ for all odd $i \in \{1, 2, \ldots, n\}$. This result implies that

$$\sum_{k=1}^{\frac{n}{2}} |I_{2k}(G \to G) - I_{2k}(H \to G)| = \sum_{k=1}^{n} |I_k(G \to G) - I_k(H \to G)|$$

and thus

$$\sum_{k=1}^{n} |I_k(G\to G) - I_k(H\to G)| = \sum_{k=1}^{\frac{n}{2}} \binom{\frac{n}{2}}{k}.$$

Therefore,

$$\begin{aligned}\sum_{k=0}^{n} |I_k(G\to G) - I_k(H\to G)| &= \sum_{k=1}^{\frac{n}{2}} \binom{\frac{n}{2}}{k} + 1\\ &= \sum_{k=0}^{\frac{n}{2}} \binom{\frac{n}{2}}{k}\\ &= (1+1)^{\frac{n}{2}}\\ &= 2^{\frac{n}{2}}.\end{aligned}$$

∎

Definition 4.3.4 Let G be a graph. Then the *switching isomorphism class* of G, labelled $\mathcal{G}_G$, is the set of graphs K such that $V(K) = V(G)$, $K \cong G$, and there exists a subset $U \subseteq V(G)$ such that $K = G \divideontimes U$.

Lemma 4.3.5 (Krasikov [7, Theorem 8]) *Let G be a non-VSR graph of order n. Then $|\mathcal{G}_G| \geq 2^{\frac{n}{2}-2}$.*

Proof: Let K be any graph isomorphic to G and obtained by switching on some subset of $V(G)$. By Lemma 1.4.4, $G \divideontimes U = G \divideontimes (V(G) \setminus U)$ for all $U \subseteq V(G)$. Furthermore, Lemma 1.4.3 shows that if $G \divideontimes U = G \divideontimes U'$ and $U \neq U'$, then $U = V(G) \setminus U$. Therefore, K is counted exactly twice in $\sum_{i=0}^{n} I_i(G\to G)$. Hence, $\sum_{i=0}^{n} I_i(G\to G) = 2|\mathcal{G}_G|$.

Since G is not VSR, there is some graph H such that G and H are VSE, but $G \ncong H$. Now Lemma 4.3.2 shows $\sum_{i=0}^{n} I_i(H \to G) = \sum_{i=0}^{n} I_i(G \to G)$. Thus we have

$$2|\mathcal{G}_G| = \frac{1}{2}\sum_{i=0}^{n}\Big(I_i(G \to G) + I_i(H \to G)\Big).$$

Now for all $i \in \{0, 1, 2, \ldots, n\}$, since $I_i(H \to G)$ is non-negative, we have $I_i(G \to G) + I_i(H \to G) \geq |I_i(G \to G) - I_i(H \to G)|$. Therefore

$$2|\mathcal{G}_G| \geq \frac{1}{2}\sum_{i=0}^{n}|I_i(G \to G) - I_i(H \to G)|.$$

Since Lemma 4.3.3 gives

$$\sum_{i=0}^{n}|I_i(G \to G) - I_i(H \to G)| = 2^{\frac{n}{2}},$$

we thus have $4|\mathcal{G}_G| \geq 2^{\frac{n}{2}}$, that is, $|\mathcal{G}_G| \geq 2^{\frac{n}{2}-2}$ as claimed. ∎

A definition and a brief lemma are now required before moving on to the main result.

Definition 4.3.6 Let G be a graph with vertex set V, and let $v \in V$. Define $\mathrm{Star}(G, v) = (V, E^*)$, where $E^* = \{\, vw : w \in V, vw \in E(G)\,\}$. That is, $\mathrm{Star}(G, v)$ is the spanning subgraph of G containing all edges of G incident with v (and no other edges).

Lemma 4.3.7 *Let G be a graph with vertex set V, let $v \in V$, and let S be a graph with vertex set V, all of whose edges are incident with v. Then there exists at most one graph $K \in \mathcal{G}_G$ such that $S = \mathrm{Star}(K, v)$.*

Proof: First we show that there exists a unique set $U \subseteq V \setminus \{v\}$ such that $S = \mathrm{Star}(G \circledast U, v)$.

We begin by showing existence. Define $U \subseteq V \setminus \{v\}$ as follows. For any $x \in V \setminus \{v\}$, let:

(a) $x \in U$ if $xv \in E(G)$ and $xv \notin E(S)$, or if $xv \notin E(G)$ and $xv \in E(S)$, and

(b) $x \notin U$ if $xv \in E(G)$ and $xv \in E(S)$, or if $xv \notin E(G)$ and $xv \notin E(S)$.

Then $v \notin U$ and $S = \mathrm{Star}(G \circledast U, v)$, as required.

Next, we show uniqueness. Let $U' \subseteq V \setminus \{v\}$ be such that $S = \mathrm{Star}(G \circledast U', v)$. We will show that $U = U'$. Take any $x \in U'$. If $xv \in E(G)$, then $xv \notin E(G \circledast U')$, and hence $xv \notin E(S)$. Thus $x \in U$, by the above definition of U. Similarly, if $xv \notin E(G)$, then $xv \in E(G \circledast U')$, and hence $xv \in E(S)$. Again, it follows that $x \in U$.

Now, take any $x \in (V \setminus \{v\}) \setminus U'$. If $xv \in E(G)$, then $xv \in E(G \circledast U')$, and hence $xv \in E(S)$. Thus $x \notin U$ by the above definition of U. Similarly, if $xv \notin E(G)$, then $xv \notin E(G \circledast U')$, and hence $xv \notin E(S)$. Again, it follows that $x \notin U$.

We conclude that $U = U'$; that is, there exists a unique set $U \subseteq V \setminus \{v\}$ such that $S = \mathrm{Star}(G \circledast U, v)$. (Note, however, that $G \circledast U$ need not be isomorphic to G; that is, it may be that $G \circledast U \notin \mathcal{G}_G$.)

Finally, suppose that there exist $K_1, K_2 \in \mathcal{G}_G$ such that $S = \mathrm{Star}(K_1, v) = \mathrm{Star}(K_2, v)$. We know there exist $U_1, U_2 \subseteq V$ such that $K_1 = G \circledast U_1$ and $K_2 = G \circledast U_2$. Since $G \circledast U_1 = G \circledast (V \setminus U_1)$, we may assume that $v \notin U_i$, for $i \in \{1, 2\}$. It now follows from the above that $U_1 = U_2$; that is, $K_1 = K_2$. Hence there exists at most one $K \in \mathcal{G}_G$ such that $S = \mathrm{Star}(K, v)$. ∎

Now we come to the main result of this section. The best (published) bounds on the minimum and maximum degree of a non-VSR graph are fairly weak, as we shall see.

Theorem 4.3.8 (Krasikov [7, Theorem 8]) *Let G be a graph of order n, with maximum degree Δ and minimum degree δ that satisfy the inequality*

$$\min\left(\binom{n-1}{\Delta}, \binom{n-1}{\delta}\right) < \frac{1}{n} 2^{n/2-2}.$$

Then G is VSR.

Proof: Suppose G is not VSR. Since there is a vertex of G with degree Δ, every element of $\mathcal{G}_G$ (see Definiton 4.3.4) must have a vertex with degree Δ as well. Fix a vertex v of G. Consider the elements of $\mathcal{G}_G$ for which the degree of v is Δ. There are $\binom{n-1}{\Delta}$ distinct sets of Δ vertices that might be adjacent to v in such a graph. Each of these sets corresponds to at most one element of $\mathcal{G}_G$, by Lemma 4.3.7. Therefore, there can be at most $\binom{n-1}{\Delta}$ graphs in $\mathcal{G}_G$ for which the degree of v is Δ. Then, since there are n choices for v, there can be at most $n\binom{n-1}{\Delta}$ elements in $\mathcal{G}_G$. Recall that Lemma 4.3.5 shows $|\mathcal{G}_G| \geq 2^{n/2-2}$. Thus $n\binom{n-1}{\Delta} \geq 2^{n/2-2}$, and so $\binom{n-1}{\Delta} \geq \frac{1}{n}2^{n/2-2}$. Since

$$\min\left(\binom{n-1}{\Delta}, \binom{n-1}{\delta}\right) < \frac{1}{n}2^{n/2-2}, \tag{4.3.1}$$

this means that $\binom{n-1}{\delta} < \frac{1}{n}2^{n/2-2}$.

Similarly, there is a vertex of G with degree δ, and so every element of $\mathcal{G}_G$ must have a vertex of degree δ as well. Again, fix a vertex v of G. Then there are at most $\binom{n-1}{\delta}$ elements of $\mathcal{G}_G$ in which vertex v has degree δ. Therefore, since there are n choices for v, we have $\binom{n-1}{\delta} \geq \frac{1}{n}2^{n/2-2}$. But inequality (4.3.1) tells us that $\binom{n-1}{\delta} < \frac{1}{n}2^{n/2-2}$, which gives us a contradiction. Therefore, G is VSR. ∎

Krasikov [8] also gave bounds on the minimum and maximum degree of a graph that is not reconstructible from its s-switch deck, for all s, but in the case $s = 1$ these bounds are slightly weaker than those of Theorem 4.3.8.

4.4 A Bound on the Number of Automorphisms

The lower bound on the size of the switching isomorphism class of a non-VSR graph (proven by Lemma 4.3.5) led Krasikov to the degree bounds of Theorem 4.3.8. But we can also exploit the switching isomorphism class bound in a different way. The size of the switching isomorphism class of a graph is a lower bound on the size of the isomorphism class of the graph. The multiset of $n!$ graphs produced by permuting the vertices of a graph G of order n in all possible ways can be partitioned into multisets of identical graphs, each of which contains exactly $|\operatorname{Aut}(G)|$ elements. Putting these facts together yields a weak upper bound on the order of the automorphism group of G.

Lemma 4.4.1 *Let G be a non-VSR graph of order n. Then $|\operatorname{Aut}(G)| \leq \frac{n!}{2^{n/2-2}}$.*

Proof: First we show that the number of cosets of $\operatorname{Aut}(G)$ in Sym_n is equal to the number of (labelled) graphs isomorphic to G. For this we define a mapping $\Phi : \operatorname{Sym}_n / \operatorname{Aut}(G) \to \Xi(G)$ by $\Phi(\theta \operatorname{Aut}(G)) = \theta(G)$ for all $\theta \operatorname{Aut}(G) \in \operatorname{Sym}_n / \operatorname{Aut}(G)$. We then show that Φ is a bijection, as follows. If $\theta(G) = \varphi(G)$, then $\varphi^{-1}\theta(G) = G$, and so $\varphi^{-1}\theta \in \operatorname{Aut}(G)$. Thus $\varphi \operatorname{Aut}(G) = \varphi\varphi^{-1}\theta \operatorname{Aut}(G) = \theta \operatorname{Aut}(G)$, and so since $\theta(G) = \varphi(G)$ implies $\theta \operatorname{Aut}(G) = \varphi \operatorname{Aut}(G)$, we see that Φ is injective. Next, for any $H \in \Xi(G)$, let φ be an isomorphism that maps G to H. Then $\Phi(\varphi \operatorname{Aut}(G)) = \varphi(G) = H$, and so Φ is surjective. Thus Φ is bijective, and therefore the number of cosets of $\operatorname{Aut}(G)$ in Sym_n is equal to the number of (labelled) graphs isomorphic to G.

Since the index of $\operatorname{Aut}(G)$ in Sym_n is the number of graphs that are isomorphic to G, i.e., $[\operatorname{Sym}_n : \operatorname{Aut}(G)] = |\Xi(G)|$, Lagrange's Theorem tells us that $|\operatorname{Sym}_n| = |\Xi(G)| \cdot |\operatorname{Aut}(G)|$. Therefore, $|\Xi(G)| = \frac{n!}{|\operatorname{Aut}(G)|}$. Since $|\mathcal{G}_G| \leq |\Xi(G)|$, we have $|\mathcal{G}_G| \leq \frac{n!}{|\operatorname{Aut}(G)|}$. Using the result of Lemma 4.3.5, namely $|\mathcal{G}_G| \geq 2^{n/2-2}$, we thus get $2^{n/2-2} \leq \frac{n!}{|\operatorname{Aut}(G)|}$, and so $|\operatorname{Aut}(G)| \leq \frac{n!}{2^{n/2-2}}$. ∎

Chapter 5

Counting Subgraphs from a Switch Deck

We now examine a result proven by Ellingham and Royle [5], which states that if G is a graph of order n, and if $k < \frac{n}{2}$, then for any graph K on k vertices, the number of induced subgraphs of G which are isomorphic to K is reconstructible from $\mathrm{SD}(G)$. This result can then be used to prove that graphs that contain no triangles are VSR. Using this subgraph counting theorem in an exhaustive search for non-VSR graphs is computationally expensive, since it would require us to calculate, store, and compare the counts of each isomorphism class of the subgraphs of each graph. Therefore we close this chapter with some properties of VSE graphs that are consequences of the subgraph counting theorem, but are computationally faster to check.

5.1 The Subgraph Switch Matrix

In this section, we define the subgraph switch matrix, which encodes the effect of vertex switching on induced subgraphs of a graph.

Definition 5.1.1 Let $\mathcal{S} = (S_1, S_2, \ldots, S_m)$ be an m-tuple of graphs on a fixed vertex set. The *switch matrix* of $\mathcal{S}$ (denoted $M(\mathcal{S})$) is defined by $M(\mathcal{S})_{(i,j)} = I_1(S_j \to S_i)$ for all $i \in \{1, 2, \ldots, m\}$ and all $j \in \{1, 2, \ldots, m\}$.

In other words, the (i, j)th entry of a switch matrix counts the number of graphs in $\mathrm{SD}(S_j)$ which are isomorphic to S_i.

Lemma 5.1.2 *Let $k \in \mathbb{N}$, let m be the number of pairwise non-isomorphic graphs on a fixed set V of k vertices, and let $\mathcal{S} = (S_1, S_2, \ldots, S_m)$ be an ordering of the m pairwise non-isomorphic graphs on V. Then each column sum of the switch matrix of $\mathcal{S}$ is equal to k.*

Proof: Consider any $j \in \{1, 2, \ldots, m\}$. Note that the switch deck of S_j consists of exactly k members of $\mathcal{S}$ (with multiplicities), and so $\displaystyle\sum_{i=1}^{m} I_1(S_j \to S_i) = k$. Since $M(\mathcal{S})_{(i,j)} = I_1(S_j \to S_i)$, we have $\displaystyle\sum_{i=1}^{m} M(\mathcal{S})_{(i,j)} = k$. ∎

If $\mathcal{S}$ is the set of all pairwise non-isomorphic graphs of a given order, and $G \in \mathcal{S}$, then multiplying $M(\mathcal{S})$ on the right by a column vector consisting of a 1 in the position corresponding to the graph G, and 0 in every other position, produces a vector (the column of $M(\mathcal{S})$ corresponding to G) whose coordinate corresponding to a graph H is $I_1(G \to H)$, that is, $\mu_{\mathrm{SD}(G)}(H)$, for all $H \in \mathcal{S}$. Therefore, if $M(\mathcal{S})$ were invertible, we could reconstruct a graph from its switch deck, thereby solving the vertex-switching reconstruction problem. Unfortunately we cannot address the invertibility of this matrix, but we can use it to produce an invertible matrix that describes something interesting about vertex-switching.

Definition 5.1.3 Let k and n be positive integers with $k \leq n$, let m be the number of pairwise non-isomorphic graphs on a fixed set V of k vertices, and let $\mathcal{S} = (S_1, S_2, \ldots, S_m)$ be an ordering of a set of m pairwise non-isomorphic graphs on

V. Then the *subgraph switch matrix* of $\mathcal{S}$ with respect to n (denoted $K(\mathcal{S}, n)$) is $M(\mathcal{S}) + (n-k) \cdot I_m$, where I_m is the identity matrix of order m.

5.2 Reconstructing Subgraph Counts

Consider the following question. Given a graph G on a set of n vertices, pick a subset of k vertices of G and let S_j be the subgraph of G induced on these k vertices. Looking at the same subset of k vertices in each of the n switch cards of G, how many of the n subgraphs of these switch cards which are induced by this set of k vertices belong to each of the m isomorphism classes of the graphs on k vertices? This question is answered by the following lemma.

Lemma 5.2.1 *Let G be a graph on a vertex set V of size n, let $\{C_1, C_2, \ldots, C_n\}$ be the set of switch cards of G, let k be a positive integer with $k \leq n$, let m be the number of non-isomorphic graphs of order k, and let $\mathcal{S} = (S_1, S_2, \ldots, S_m)$ be an ordering of m pairwise non-isomorphic graphs on a set of k vertices. Fix $i, j \in \{1, 2, \ldots, m\}$ and assume there exists $P \subseteq V$ such that $G[P] \cong S_j$. Then $K(\mathcal{S}, n)_{(i,j)}$ equals the number of switch cards C of G such that $C[P] \cong S_i$.*

Proof: Fix $v \in V$ and let $C = G \circledast v$. Then if $v \notin P$, we have $C[P] = G[P]$. Otherwise $C[P]$ is obtained from $G[P]$ (which is isomorphic to S_j) by switching on a single vertex (namely, v). There are $n-k$ switch cards of G of the first type since there are $n-k$ vertices in $V(G) \setminus P$. On the other hand, the number of switch cards C whose induced subgraph $C[P]$ is isomorphic to S_i is $I_1(S_j \rightarrow S_i)$. Therefore, the total number of switch cards whose subgraph induced by P is isomorphic to S_i is equal to $I_1(S_j \rightarrow S_i)$ when $i \neq j$ and $I_1(S_j \rightarrow S_i) + (n-k)$ when $i = j$, that is, is equal to $K(\mathcal{S}, n)_{(i,j)}$ in both cases. ∎

The previous lemma says that $K(S,n)_{(i,j)}$ gives the number of induced subgraphs isomorphic to S_i in all of the switch cards of G produced by each induced subgraph of G isomorphic to S_j. For each such S_j, there will be $n-k$ unchanged copies, since no vertex of S_j is switched on in $n-k$ of the switch cards, and as many will come from each S_j in G as there are occurrences of S_i in the switch deck of S_j.

We now define a few terms that will be of use in subsequent proofs.

Definition 5.2.2 For a proposition $\mathcal{P}$, define

$$T(\mathcal{P}) = \begin{cases} 0, & \text{if } \mathcal{P} \text{ is false;} \\ 1 & \text{if } \mathcal{P} \text{ is true.} \end{cases}$$

Definition 5.2.3 Let S and G be graphs. Then the *subgraph number* of S in G (denoted $\mathrm{Sbg}(G,S)$) is the number of induced subgraphs of G which are isomorphic to S.

The following theorem, as well as the subsequent lemma and corollary, were proven by Ellingham and Royle [5], although the result presented here is less general, and thus the proofs herein use a somewhat different (and simpler) approach.

Theorem 5.2.4 *Let k be a positive integer, and let m be the number of pairwise non-isomorphic graphs of order k. Let G be a graph of order n. Let $\mathcal{S} = (S_1, S_2, \ldots, S_m)$ be an m-tuple of pairwise non-isomorphic graphs on a set of k vertices such that $K(\mathcal{S}, n)$ is invertible. Then for any $S_i \in \mathcal{S}$, we can reconstruct* $\mathrm{Sbg}(G, S_i)$ *from* $\mathrm{SD}(G)$.

Proof: Let X be a column vector of dimension m, where $X_i = \mathrm{Sbg}(G, S_i)$ for all $1 \le i \le m$. Fix $i \in \{1, 2, \ldots, m\}$. We are going to count the total number of induced subgraphs of all of the switch cards of G that are isomorphic to S_i. On the one hand, this number (call it A) is obviously equal to $\sum\limits_{C \in \mathrm{SD}(G)} \mathrm{Sbg}(C, S_i)$. On the other hand, we observe that a switch card of G has an induced subgraph isomorphic to S_i if and only if, for some $P \subseteq V(G)$, we have $C[P] \cong S_i$. Then:

$$\begin{aligned}
A &= \sum_{P \subseteq V(G)} \sum_{C \in \mathrm{SD}(G)} T(C[P] \cong S_i) \\
&= \sum_{j=1}^{m} \sum_{P \subseteq V(G)} T(G[P] \cong S_j) \sum_{C \in \mathrm{SD}(G)} T(C[P] \cong S_i) \\
&= \sum_{j=1}^{m} \sum_{P \subseteq V(G)} T(G[P] \cong S_j) K(\mathcal{S}, n)_{(i,j)} \\
&= \sum_{j=1}^{m} \mathrm{Sbg}(G, S_j) K(\mathcal{S}, n)_{(i,j)} \\
&= \sum_{j=1}^{m} K(\mathcal{S}, n)_{(i,j)} \mathrm{Sbg}(G, S_j).
\end{aligned}$$

Therefore,

$$\sum_{C \in \mathrm{SD}(G)} \mathrm{Sbg}(C, S_i) = \sum_{j=1}^{m} K(\mathcal{S}, n)_{(i,j)} \mathrm{Sbg}(G, S_j)$$

and so

$$K(\mathcal{S}, n) \cdot X = N,$$

where

$$N_i = \sum_{C \in \mathrm{SD}(G)} \mathrm{Sbg}(C, S_i).$$

Since $K(S, n)$ is assumed to be invertible, we have $X = K(S, n)^{-1}N$. Now, N is obtained directly from $\mathrm{SD}(G)$, and $K(\mathcal{S}, n)$ is independent of G. Hence, for each order-k graph H we can find $\mathrm{Sbg}(G, H)$, the number of induced subgraphs of G that are isomorphic to H, from $\mathrm{SD}(G)$ alone. ∎

The proof of the following lemma is due in part to a theorem that has been independently proven many times (see [15]), and which is usually referred to as the Levy-Desplanques Theorem. The Levy-Desplanques Theorem shows that if, for each column of a matrix, the sum of the absolute values of the non-diagonal elements is less than the absolute value of the diagonal element, then the matrix is invertible.

Lemma 5.2.5 *Let k and n be positive integers with $k < \frac{n}{2}$. Let m be the number of pairwise non-isomorphic graphs of order k, and let $\mathcal{S} = (S_1, S_2, \ldots, S_m)$ be an ordering of m pairwise non-isomorphic graphs on a set of k vertices. Then $K(\mathcal{S}, n)$, the subgraph switch matrix of S with respect to n, is invertible.*

Proof: Recall that $K(\mathcal{S}, n) = M(S) + (n-k) \cdot I_m$. Since $M(\mathcal{S})_{(i,j)} = I_1(S_j \to S_i)$ for all i, j, we have that $M(\mathcal{S})_{(i,j)} \geq 0$. In particular, this means that $K(\mathcal{S}, n)_{(i,i)} \geq n-k$. By the assumption, $k < \frac{n}{2}$, whence $K(\mathcal{S}, n)_{(i,i)} \geq n - k > k$. Since each column sum in $M(\mathcal{S})$ is k, the column sums of the absolute values of the non-diagonal elements in $K(\mathcal{S}, n)$ are at most k, and hence are strictly less than the absolute values of the diagonal elements. Symbolically, for all $i \in \{1, 2, \ldots, m\}$, we have

$$\sum_{\substack{j \\ j \neq i}} |K(\mathcal{S}, n)_{(i,j)}| < |K(\mathcal{S}, n)_{(i,i)}|. \tag{5.2.1}$$

Now suppose $K(\mathcal{S}, n)$ is not invertible. Then there exists a non-zero vector x such that $K(\mathcal{S}, n)x = 0$. Let $l \in \{1, 2, \ldots, m\}$ such that $|x_l| \geq |x_j|$ for all $j \in \{1, 2, \ldots, m\}$. Then $\sum_{j=1}^{m} K(\mathcal{S}, n)_{(l,j)} x_j = 0$, and thus

$$|K(\mathcal{S}, n)_{(l,l)}||x_l| = |K(\mathcal{S}, n)_{(l,l)} x_l| = \left| \sum_{\substack{j \\ j \neq l}} K(\mathcal{S}, n)_{(l,j)} x_j \right|.$$

By the triangle inequality, we get

$$\left| \sum_{\substack{j \\ j \neq l}} K(\mathcal{S}, n)_{(l,j)} x_j \right| \leq \sum_{\substack{j \\ j \neq l}} |K(\mathcal{S}, n)_{(l,j)} x_j|,$$

and since $|x_j| \leq |x_l|$ for all $j \in \{1, 2, \ldots, m\}$, we have

$$\sum_{\substack{j \\ j \neq l}} |K(\mathcal{S}, n)_{(l,j)} x_j| = \sum_{\substack{j \\ j \neq l}} |K(\mathcal{S}, n)_{(l,j)}||x_j| \leq \sum_{\substack{j \\ j \neq l}} |K(\mathcal{S}, n)_{(l,j)}||x_l|.$$

Now bringing this all together, we get

$$|K(\mathcal{S},n)_{(l,l)}||x_l| \leq \sum_{\substack{j \\ j\neq l}} |K(\mathcal{S},n)_{(l,j)}||x_l| = |x_l| \sum_{\substack{j \\ j\neq l}} |K(\mathcal{S},n)_{(l,j)}|,$$

and so

$$|K(\mathcal{S},n)_{(l,l)}| \leq \sum_{\substack{j \\ j\neq l}} |K(\mathcal{S},n)_{(l,j)}| \qquad \text{since } |x_l| \neq 0,$$

which contradicts Equation (5.2.1). Consequently, $K(\mathcal{S},n)$ is invertible. ∎

Corollary 5.2.6 *Let K be a graph of order k, and let G be a graph of order n, where $n > 2k$. Then $\mathrm{Sbg}(G,K)$ is reconstructible from $\mathrm{SD}(G)$.*

Proof: Let m be the number of non-isomorphic graphs on k vertices, and let $\mathcal{S} = (S_1, S_2, \ldots, S_m)$ be an m-tuple of pairwise non-isomorphic graphs on a set of k vertices, where $S_i = K$ for some i. From Lemma 5.2.5 we know that $K(\mathcal{S},n)$ is invertible. Therefore, by Theorem 5.2.4, we can reconstruct $\mathrm{Sbg}(G,S_i)$ (and hence $\mathrm{Sbg}(G,K)$) from $\mathrm{SD}(G)$. ∎

Corollary 5.2.7 *Let G and H be VSE graphs of order ≥ 7 on the same vertex set. Let K be a graph of order 3. Then $\mathrm{Sbg}(G,K) = \mathrm{Sbg}(H,K)$.*

Proof: Corollary 5.2.6 shows that $\mathrm{Sbg}(G,K)$ can be reconstructed from $\mathrm{SD}(G)$, and that $\mathrm{Sbg}(H,K)$ can be reconstructed from $\mathrm{SD}(H)$. Since G and H are VSE, $\mathrm{SD}(G) = \mathrm{SD}(H)$. Therefore $\mathrm{Sbg}(G,K) = \mathrm{Sbg}(H,K)$. ∎

5.3 Triangle-Free Graphs

The first use to which we will put our induced subgraph counting result is a proof that a non-VSR graph must contain at least one set of three mutually adjacent vertices. First we will need to define a few very simple terms.

Definition 5.3.1 A *path of length* n in a graph G is a subgraph of G with vertex set $\{v_1, v_2, \ldots, v_{n+1}\}$ and edge set $\{\, v_i v_{i+1} : 1 \leq i \leq n \,\}$.

Definition 5.3.2 A *cycle of length* n of a graph G is a subgraph of G consisting of a path of length $n-1$ together with an edge joining the first and last vertex of this path.

Definition 5.3.3 The *complete graph* on a set V, denoted $K(V)$, is the graph for which $e_{K(V)}(u,v) = 1$ for all $u, v \in V$. Where V is understood, and $n = |V|$, we may write simply K_n.

Definition 5.3.4 A graph G is *triangle-free* if no set of three vertices of G induces a subgraph isomorphic to K_3.

Now we present a short definition which will be used only in the context of the following two results. However, it significantly simplifies the proofs.

Definition 5.3.5 Let G and H be non-isomorphic VSR graphs, and let $v, w \in V(G)$ be an H-switch pair in G. An edge xy of G is a v,w*-balanced edge* if $x \in N_G^1(v,w)$ and $y \in N_G^1(v,w)$. An edge xy of G is a v,w*-unbalanced edge* if one element of $\{x, y\}$ is in $N_G^0(v,w)$ and the other is in $N_G^2(v,w)$.

Note that, in general, an edge of a graph may be neither v,w-balanced nor v,w-unbalanced, for any switch pair v,w. However, we shall soon see that in a triangle-free graph G, for every switch pair v,w, every edge of G is either v,w-balanced or v,w-unbalanced.

The results of this section (below) are due to Ellingham and Royle [5].

Lemma 5.3.6 *Let G be a non-VSR triangle-free graph of order n, where $n \neq 4$. Then G contains no odd cycles.*

Proof: Since G is not VSR, there exists a graph H which is not isomorphic to G but which is VSE to G. Let $v \in V(G)$. By Lemma 3.1.2, there exists some $w \in V(G)$ such that $G \circledast v \circledast w \cong H$. Since Lemma 3.2.4 shows that $|N_G^2(v, w)| > 0$, there is at least one vertex adjacent to both v and w, and so $vw \notin E(G)$ since G is triangle-free.

Let xy be any edge of G with neither endpoint in $\{v, w\}$. Then x and y cannot both be in $N_G(v)$ or $N_G(w)$ since G is triangle-free. Observe that, by Theorem 2.5.3, since G is non-VSR, we have $n \equiv 0 \pmod 4$, and since $n \neq 4$, we have $n \geq 8$. Now we can apply Corollary 5.2.7, which tells us that the number of induced subgraphs of G isomorphic to K_3 must equal the number of induced subgraphs of H isomorphic to K_3, since the order of K_3 is less than half of the order of G, and G and H have isomorphic switch decks. Therefore, H is triangle-free. This means that x and y cannot both be in $\overline{N}_G(v)$ or in $\overline{N}_G(w)$, since otherwise the subgraph of $G \circledast v \circledast w$ induced by $\{v, x, y\}$ or $\{w, x, y\}$ would be isomorphic to K_3. Thus, if $x \in N_G^2(v, w)$, then $y \in N_G^0(v, w)$, and if $x \in N_G^0(v, w)$, then $y \in N_G^2(v, w)$. Furthermore, if $x \in N_G^1(v, w)$, then $y \in N_G^1(v, w)$. Therefore, xy is either a v, w-balanced edge or a v, w-unbalanced edge. Furthermore, the endpoints of a v, w-balanced edge are each adjacent to distinct members of $\{v, w\}$, since G is triangle-free.

Let P be a path in G that includes neither v nor w. Then P must consist entirely of v, w-balanced and v, w-unbalanced edges. Let z be a vertex of P. If $z \in N_G^j(v, w)$, where $j = 0$ or 2, then the edges of P incident with z are v, w-unbalanced edges. In this case, the other endpoints of these two edges must be in $N_G^{2-j}(v, w)$. Thus, the vertices along P alternate between members of the disjoint sets $N_G^0(v, w)$ and $N_G^2(v, w)$. If $z \in N_G^1(v, w)$, then the edges of P incident with z are v, w-balanced edges, and the other endpoints of these edges must be adjacent to the member of

$\{v, w\}$ which is not adjacent to z. Therefore, the vertices along P alternate between members of the disjoint sets $N_G(v) \setminus N_G(w)$ and $N_G(w) \setminus N_G(v)$.

Now consider any cycle C of G of length k which does not contain v or w. If we remove an arbitrary edge of C we are left with a path of length $k-1$. Therefore if k is odd, the first and last vertices of this path will be adjacent to the same subset of $\{v, w\}$. However, these vertices are adjacent in C, which is impossible, and so k must be even. Therefore if G contains a cycle C of odd length, then C must contain either v or w. Without loss of generality, assume C contains v. Then, since v was an arbitrarily chosen vertex, C must contain every vertex of G, and so C has length n. But n is even, which is impossible since C has odd length. Therefore, G contains no cycle of odd length. ∎

Theorem 5.3.7 *Let G be a triangle-free graph of order n, where $n \neq 4$. Then G is VSR.*

Proof: Suppose G is not VSR. Then there is a graph H which is VSE to G but is not isomorphic to G.

Take any vertex $v \in V(G)$ and let w be an H-switch partner of v in G. (Such a vertex must exist by Lemma 3.1.2.) We begin by showing that all edges of G with neither endpoint in $\{v, w\}$ must be either v, w-balanced edges or v, w-unbalanced edges. Suppose, to the contrary, that there is a pair of vertices $x, y \notin \{v, w\}$ that are adjacent in G but that fall into one of the following categories (without loss of generality):

(1) $x \in N_G^1(v, w)$ and $y \in N_G^0(v, w)$,

(2) $x \in N_G^1(v, w)$ and $y \in N_G^2(v, w)$,

(3) $x \in N_G^0(v, w)$ and $y \in N_G^0(v, w)$, or

(4) $x \in N_G^2(v,w)$ and $y \in N_G^2(v,w)$.

In cases (2) and (4), a subgraph of G induced by x, y, and a member of $\{v,w\}$ that is adjacent to x in G is a triangle, which is impossible since G is triangle-free. Now let $K = G \circledast v \circledast w$. Since K and G are VSE, and G is triangle-free, Corollary 5.2.7 shows that K is triangle-free as well. But in cases (1) and (3), K contains a triangle which is induced by x, y, and an element of $\{v,w\}$ that is not adjacent to x in G. Therefore all four cases are impossible, and so every edge of G with neither endpoint in $\{v,w\}$ is either a v,w-balanced edge or a v,w-unbalanced edge.

Next, we show that G has no v,w-balanced edges. Let $z \in N_G^2(v,w)$. (Such a vertex must exist by part (ii) of Lemma 3.2.4.) First, note that $e_G(v,w) = 0$, since otherwise $G[\{v,x,z\}]$ would be isomorphic to K_3, contradicting the assumption of the theorem. Now suppose there exists a v,w-balanced edge xy in G. Without loss of generality, we have $x \in N_G(v) \setminus N_G(w)$ and $y \in N_G(w) \setminus N_G(v)$, and so $z \notin \{x,y\}$. Then (z,v,x,y,w) is a cycle of odd length in G, which is impossible by Lemma 5.3.6. Therefore, for every H-switch pair $\{v,w\}$, no v,w-balanced edges exist in G.

Next, suppose that, for every choice of an H-switch pair $\{v,w\}$, both $N_G(v) \setminus N_G(w)$ and $N_G(w) \setminus N_G(v)$ are empty. Then $V(G) = N_G^0(v,w) \cup N_G^2(v,w) \cup \{v,w\}$, and so $|N_G^0(v,w)| + |N_G^2(v,w)| + 2 = n$. Since $|N_G^0(v,w)| = |N_G^2(v,w)|$, we have $|N_G^2(v,w)| = \frac{n-2}{2}$. Now since $N_G(v) \setminus N_G(w) = \emptyset$, and $e_G(v,w) = 0$, this implies $d_G(v) = |N_G^2(v,w)|$, and so $d_G(v) = \frac{n-2}{2}$ for all $v \in V(G)$. Therefore G is regular, and, by Lemma 3.4.6, is therefore VSR, which contradicts our supposition that G is not VSR.

Thus there exists some $v \in V(G)$ such that, for some H-switch partner w of v in G, we have (without loss of generality) $N_G(v) \setminus N_G(w) \neq \emptyset$. Let x be a vertex in $N_G^1(v,w)$. Suppose there is some vertex $y \notin \{v,w\}$ that is adjacent to x in G. We have previously shown that the edge xy must be a v,w-unbalanced edge. However, this is impossible, since $x \in N_G^1(v,w)$. Therefore, every vertex in $N_G^1(v,w)$ (which contains $N_G(v) \setminus N_G(w)$) has degree 1 in G. Now suppose $b \in N_G(v) \setminus N_G(w)$. Then by Lemma 3.1.2, vertex b has an H-switch partner c in G, where $c \neq b$, and $d_G(b) + d_G(c) = n - 2 + 2e_G(b,c)$. However, $e_G(b,c) = 0$ as shown above. Since $d_G(b) = 1$, we have $d_G(c) = n - 3$. Therefore, c is adjacent to all but one vertex of $V(G) \setminus \{b,c\}$ in G. Let d be this vertex. Suppose $c = w$. Then c is adjacent to all vertices of $V(G) \setminus \{b,v,c\}$ (since $vw \notin E(G)$), and so $N_G^2(b,c) = \emptyset$, which contradicts Lemma 3.2.4. Therefore, $c \neq w$. Suppose $d = v$ or $d = w$. Then we have $c \in N_G^1(v,w)$, and thus $d_G(c) = 1$, a contradiction. Therefore $d \notin \{v,w\}$, and so d is the only vertex in $N_G^0(b,c)$.

Recall that all the edges of G are v,w-unbalanced edges or edges with exactly one endpoint in $\{v,w\}$. Since $\{b,c\}$ is also an H-switch pair in G, there are also no b,c-balanced edges. This means that every edge not incident with b or c must have one endpoint in $N_G^0(b,c)$, and the other endpoint in $N_G^2(b,c)$. However, $N_G^0(b,c) = \{d\}$, and $N_G^2(b,c) = \{v\}$. Therefore, all edges of G are edges incident with b (which can only be bv), edges incident with c, and possibly dv. Now let $J = G \circledast b \circledast c$, and define a permutation φ on $V(G)$ by $\varphi(b) = c$, $\varphi(c) = b$, $\varphi(v) = d$, $\varphi(d) = v$, and $\varphi(x) = x$ for all $x \in V(G) \setminus \{b,c,d,v\}$. We shall show that φ is an isomorphism from G to J. Observe that $e_{\varphi(G)}(b,x) = e_G(c,x) = 1 = e_J(b,x)$ and $e_{\varphi(G)}(c,x) = e_G(b,x) = 0 = e_J(c,x)$, for all $x \in V(G) \setminus \{b,c,d,v\}$. Next, observe that $e_{\varphi(G)}(v,x) = e_G(d,x) = 0 = e_J(v,x)$ and $e_{\varphi(G)}(d,x) = e_G(v,x) = 0 = e_J(d,x)$, for all $x \in V(G) \setminus \{b,c,d,v\}$. Finally, note the following.

$$e_{\varphi(G)}(b,v) = e_G(c,d) = 0 = e_J(b,v)$$
$$e_{\varphi(G)}(b,c) = e_G(c,b) = e_G(b,c) = e_J(b,c)$$
$$e_{\varphi(G)}(b,d) = e_G(c,v) = 1 = e_J(b,d)$$
$$e_{\varphi(G)}(c,v) = e_G(b,d) = 0 = e_J(c,v)$$
$$e_{\varphi(G)}(c,d) = e_G(b,v) = 1 = e_J(c,d)$$

$$e_{\varphi(G)}(v, d) = e_G(d, v) = e_J(v, d).$$

Therefore, $e_{\varphi(G)}(x, y) = e_J(x, y)$ for all $x, y \in V(G)$, and so φ is an isomorphism from G to J. However, J is isomorphic to H (since b and c are an H-switch pair), but H is not isomorphic to G. This contradiction shows that G is VSR, and thus triangle-free graphs of order $\neq 4$ are VSR. ∎

Definition 5.3.8 The *edgeless graph* on a vertex set V, denoted $E(V)$, is the graph for which $e_{E(V)}(u, v) = 0$ for all $u, v \in V$. Where V is understood, and $n = |V|$, we may write simply E_n.

Corollary 5.3.9 *Let G be a graph of order n, where $n \neq 4$, such that no induced subgraph of G is isomorphic to E_3. Then G is VSR.*

Proof: Suppose G is not VSR. Then there is some H which is VSE to G and for which $H \ncong G$. Then $\overline{G}$ and $\overline{H}$ are non-isomorphic. By Lemma 1.4.10, $\overline{G}$ and $\overline{H}$ are VSE as well. Thus $\overline{G}$ is not VSR. Then, by Theorem 5.3.7, there is some set of three vertices of $\overline{G}$ which induces a subgraph of $\overline{G}$ isomorphic to K_3. These same three vertices induce a subgraph of G that is isomorphic to E_3, which contradicts the assumption on G. Therefore, G is VSR. ∎

5.4 Efficient Subgraph Counting

Corollary 5.2.6 provides a useful necessary condition for a graph G to be non-VSR (namely that the number of induced subgraphs of G isomorphic to a given subgraph whose order is less than half of the order of G can be reconstructed from $\mathrm{SD}(G)$).

However, when conducting a search for non-VSR graphs, the result as stated is computationally expensive to check for a given G, since we must create a candidate non-isomorphic VSE graph H and compare the number of induced subgraphs of each isomorphism class between G and H. The following results provide a less expensive method for checking this condition, and are used in the search algorithm presented in Chapter 6.

Definition 5.4.1 Let G be a graph on a vertex set V, and $v, w \in V$. Furthermore, let S be a graph. Define $\mathrm{Sbg}(G, S, +v, -w)$ to be the number of induced subgraphs of G that are isomorphic to S and that contain v but not w in their vertex set, i.e.,

$$\mathrm{Sbg}(G, S, +v, -w) = |\{\, P : P \subseteq V(G) \setminus \{w\},\, G[P] \cong S, \text{ and } v \in P \,\}|.$$

Then, similarly, define the following:

$$\mathrm{Sbg}(G, S, +v, +w) = |\{\, P : P \subseteq V(G),\, G[P] \cong S, \text{ and } \{v, w\} \subseteq P \,\}|$$
$$\mathrm{Sbg}(G, S, -v, -w) = |\{\, P : P \subseteq V(G),\, G[P] \cong S, \text{ and } v, w \notin P \,\}|.$$

Lemma 5.4.2 *Let S be a graph. Let G and H be VSE graphs on the same vertex set V, with $|V(S)| < \frac{1}{2}|V|$. Let $v, w \in V$ be an H-switch pair in G with $v \neq w$, and let $K = G \circledast v \circledast w$. Then*

$$\begin{aligned} &\mathrm{Sbg}(G, S, +v, -w) + \mathrm{Sbg}(G, S, +w, -v) + \mathrm{Sbg}(G, S, +v, +w) \\ &= \mathrm{Sbg}(K, S, +v, -w) + \mathrm{Sbg}(K, S, +w, -v) + \mathrm{Sbg}(K, S, +v, +w). \end{aligned}$$

Proof: From Corollary 5.2.6, we know that $\mathrm{Sbg}(G, S) = \mathrm{Sbg}(H, S)$. Since $K \cong H$, we also know that $\mathrm{Sbg}(K, S) = \mathrm{Sbg}(H, S)$, which means $\mathrm{Sbg}(G, S) = \mathrm{Sbg}(K, S)$. Since

$$\mathrm{Sbg}(G, S) = \mathrm{Sbg}(G, S, -v, -w) + \mathrm{Sbg}(G, S, +v, -w)$$

$$+\operatorname{Sbg}(G,S,+w,-v)+\operatorname{Sbg}(G,S,+v,+w),$$

and $\operatorname{Sbg}(G,S)=\operatorname{Sbg}(H,S)=\operatorname{Sbg}(K,S)$, we have

$$\begin{aligned}&\operatorname{Sbg}(G,S,-v,-w)+\operatorname{Sbg}(G,S,+v,+w)+\operatorname{Sbg}(G,S,+w,-v)+\operatorname{Sbg}(G,S,+v,+w)\\ &=\operatorname{Sbg}(K,S,-v,-w)+\operatorname{Sbg}(K,S,+v,-w)+\operatorname{Sbg}(K,S,+w,-v)+\operatorname{Sbg}(K,S,+v,+w).\end{aligned}$$

Note however that K is obtained by switching only on vertices v and w in G. Hence all induced subgraphs of G that include neither v nor w are unaffected by these switches. Symbolically, $\operatorname{Sbg}(G,S,-v,-w)=\operatorname{Sbg}(K,S,-v,-w)$. Thus we are left with

$$\begin{aligned}&\operatorname{Sbg}(G,S,+v,-w)+\operatorname{Sbg}(G,S,+w,-v)+\operatorname{Sbg}(G,S,+v,+w)\\ &=\operatorname{Sbg}(K,S,+v,+w)+\operatorname{Sbg}(K,S,+w,+v)+\operatorname{Sbg}(K,S,+v,+w).\end{aligned}$$

∎

Lemma 5.4.3 *Let G and H be VSE graphs on the same vertex set V, with $|V|\geq 7$. Let $v,w\in V(G)$ be an H-switch pair in G with $v\neq w$, and let $K=G\circledast v\circledast w$. Then* $\operatorname{Sbg}(K,E_3,+v,+w)=\operatorname{Sbg}(G,E_3,+v,+w)$.

Proof: Suppose $vw\in E(G)$. Then $vw\in E(K)$. Thus $\operatorname{Sbg}(G,E_3,+v,+w)=0=\operatorname{Sbg}(G,E_3,+v,+w)$.

Now suppose $vw\notin E(G)$. Then $vw\notin E(K)$. Therefore, $\operatorname{Sbg}(G,E_3,+v,+w)=|N_G^0(v,w)|$, and $\operatorname{Sbg}(K,E_3,+v,+w)=|N_K^0(v,w)|$. But $N_G^0(v,w)=N_K^2(v,w)$, and $|N_K^2(v,w)|=|N_K^0(v,w)|$ by part (i) of Lemma 3.2.4. Thus $|N_G^0(v,w)|=|N_K^0(v,w)|$, and so in all cases, $\operatorname{Sbg}(K,E_3,+v,+w)=\operatorname{Sbg}(G,E_3,+v,+w)$.

∎

Lemma 5.4.4 *Let G and H be VSE graphs on the same vertex set V, with $|V| \geq 7$. Let $v, w \in V(G)$ be an H-switch pair in G with $v \neq w$, and let $K = G \circledast v \circledast w$. Then* $\mathrm{Sbg}(K, K_3, +v, +w) = \mathrm{Sbg}(G, K_3, +v, +w)$.

Proof: Suppose $vw \notin E(G)$. Then $vw \notin E(K)$. Thus $\mathrm{Sbg}(G, K_3, +v, +w) = 0 = \mathrm{Sbg}(G, K_3, +v, +w)$.

Now suppose $vw \in E(G)$. Then $vw \in E(K)$. Therefore, $\mathrm{Sbg}(G, K_3, +v, +w) = |N_G^2(v, w)|$, and $\mathrm{Sbg}(K, K_3, +v, +w) = |N_K^2(v, w)|$. But $N_G^2(v, w) = N_K^0(v, w)$, and $|N_K^0(v, w)| = |N_K^2(v, w)|$ by part (i) of Lemma 3.2.4. Thus $|N_G^2(v, w)| = |N_K^2(v, w)|$, and so in all cases, $\mathrm{Sbg}(K, K_3, +v, +w) = \mathrm{Sbg}(G, K_3, +v, +w)$. ∎

Lemma 5.4.5 *Let G and H be VSE graphs on the same vertex set V, with $|V| \geq 7$. Let $v, w \in V(G)$ be an H-switch pair in G with $v \neq w$. Then* $\mathrm{Sbg}(G, K_3, +v, -w) + \mathrm{Sbg}(G, K_3, +w, -v)$ *is equal to the number of edges of G with neither endpoint in $N_G(v) \cup \{v, w\}$ plus the number of edges of G with neither endpoint in $N_G(w) \cup \{v, w\}$.*

Proof: Let $K = G \circledast v \circledast w$. From Lemma 5.4.2 we get

$$\begin{aligned} &\mathrm{Sbg}(G, K_3, +v, -w) + \mathrm{Sbg}(G, K_3, +w, -v) + \mathrm{Sbg}(G, K_3, +v, +w) \\ &= \mathrm{Sbg}(K, K_3, +v, -w) + \mathrm{Sbg}(K, K_3, +w, -v) + \mathrm{Sbg}(K, K_3, +v, +w). \end{aligned}$$

Then since Lemma 5.4.4 tells us that $\mathrm{Sbg}(K, K_3, +v, +w) = \mathrm{Sbg}(G, K_3, +v, +w)$, we get

$$\begin{aligned} &\mathrm{Sbg}(G, K_3, +v, -w) + \mathrm{Sbg}(G, K_3, +w, -v) \\ &= \mathrm{Sbg}(K, K_3, +v, -w) + \mathrm{Sbg}(K, K_3, +w, -v). \end{aligned}$$

Let $\mathcal{T}$ be the set of all three-vertex subsets $\{T_1, T_2, \ldots, T_k\}$ of V such that $K[T_i] \cong K_3$ and $v \in T_i$ and $w \notin T_i$ for all $T_i \in \mathcal{T}$. Clearly, $|\mathcal{T}| = \mathrm{Sbg}(K, K_3, +v, -w)$. Then for every $T_i \in \mathcal{T}$, the induced subgraph $G[T_i]$ contains exactly one edge, which joins the two vertices of $T_i \setminus \{v\}$. Conversely, for any subset of three vertices $\{v, x, y\}$ of V where $w \notin \{x, y\}$ and $G[\{v, x, y\}]$ contains only one edge, namely xy, we have $K[\{v, x, y\}] \cong K_3$. Therefore, $|\mathcal{T}|$ is equal to the number of edges of G with neither endpoint in $N_G(v) \cup \{v, w\}$, and this equals $\mathrm{Sbg}(K, K_3, +v, -w)$.

By a similar argument, $\mathrm{Sbg}(K, K_3, +w, -v)$ is equal to the number of edges of G with neither endpoint in $N_G(w) \cup \{v, w\}$. Therefore, $\mathrm{Sbg}(G, K_3, +v, -w) + \mathrm{Sbg}(G, K_3, +w, -v)$ is equal to the number of edges of G with neither endpoint in $N_G(v) \cup \{v, w\}$ plus the number of edges of G with neither endpoint in $N_G(w) \cup \{v, w\}$. ∎

Lemma 5.4.6 *Let G and H be VSE graphs on the same set V of vertices, with $|V| \geq 7$. Let $v, w \in V(G)$ be distinct H-switch partners of each other in G, and let $K = G \circledast v \circledast w$. Then $\mathrm{Sbg}(G, E_3, +v, -w) + \mathrm{Sbg}(G, E_3, +w, -v)$ is equal to the number of pairs of non-adjacent vertices of $N_G(v) \setminus \{w\}$ plus the number of pairs of non-adjacent vertices of $N_G(w) \setminus \{v\}$.*

Proof: From Lemma 5.4.2 we get

$$\begin{aligned}&\mathrm{Sbg}(G, E_3, +v, -w) + \mathrm{Sbg}(G, E_3, +w, -v) + \mathrm{Sbg}(G, E_3, +v, +w)\\ &= \mathrm{Sbg}(K, E_3, +v, -w) + \mathrm{Sbg}(K, E_3, +w, -v) + \mathrm{Sbg}(K, E_3, +v, +w).\end{aligned}$$

Then since Lemma 5.4.3 tells us that $\mathrm{Sbg}(K, E_3, +v, +w) = \mathrm{Sbg}(G, E_3, +v, +w)$, we get

$$\begin{aligned}&\mathrm{Sbg}(G, E_3, +v, -w) + \mathrm{Sbg}(G, E_3, +w, -v)\\ &= \mathrm{Sbg}(G, E_3, +v, -w) + \mathrm{Sbg}(G, E_3, +w, -v).\end{aligned}$$

Let $\mathcal{U}$ be the set of all three-vertex subsets $\{U_1, U_2, \ldots, U_k\}$ of V such that $K[U_i] \cong E_3$, $v \in U_i$, and $w \notin U_i$ for all $U_i \in \mathcal{U}$. Clearly, $|\mathcal{U}| = \mathrm{Sbg}(K, E_3, +v, -w)$. Then for every

$U_i \in \mathcal{U}$, the induced subgraph $G[U_i]$ contains exactly two edges, both of which have an endpoint at v. Conversely, for any subset of three vertices $\{v, x, y\}$ of V where $w \notin \{x, y\}$ and $G[\{v, x, y\}]$ contains exactly two edges, namely vx and vy, we have $K[\{v, x, y\}] \cong E_3$. Therefore, $|\mathcal{U}|$ is equal to the number of pairs of non-adjacent vertices of $N_G(v) \setminus \{w\}$, and this equals $\mathrm{Sbg}(K, E_3, +v, -w)$.

By a similar argument, $\mathrm{Sbg}(K, E_3, +w, -v)$ is equal to the number of pairs of non-adjacent vertices of $N_G(w) \setminus \{v\}$. Therefore,

$$\mathrm{Sbg}(G, E_3, +v, -w) + \mathrm{Sbg}(G, E_3, +w, -v)$$

is equal to the number of pairs of non-adjacent vertices of $N_G(v) \setminus \{w\}$ plus the number of pairs of non-adjacent vertices of $N_G(w) \setminus \{v\}$. ∎

5.5 Related Results

Ellingham [4] has generalized Corollary 5.2.6 by showing that, if $\nu_K < \frac{\nu_G}{2}$, then $\mathrm{Sbg}(G, K)$ is reconstructible from $\mathrm{SD}_s(G)$ for all values of s, if the Krawtchouk polynomial

$$p_s^n(x) = \sum_{i=0}^{s} (-1)^i \binom{x}{i} \binom{n-x}{s-i}$$

has no even roots in the interval $[0, \nu_K]$.

Chapter 6

Searching For Non-VSR Graphs

In this chapter, we are concerned with efficiently searching for a non-VSR graph with a certain number of vertices. We know from Theorem 2.5.3 that we need only search for graphs whose order is divisible by 4, and the 4-vertex graphs have already been placed into switch equivalence classes (see Figure 1.3). Therefore our search need only begin with graphs on 8 vertices (of which there are 12,346 isomorphism classes), and may continue with graphs on 12 vertices (of which there are 165,091,172,592 isomorphism classes).

We perform an efficient search for a non-VSR graph by generating a representative of each isomorphism class of the set of all graphs of the desired order and then checking each candidate for various necessary conditions, in increasing order of algorithmic complexity. Then we compare the switch deck of each graph G that satisfies all of these conditions to the switch decks of all graphs that can be obtained by switching suitable pairs of vertices of G. If switching on a pair of vertices of a graph G produces a graph H with $G \ncong H$ and $\mathrm{SD}(G) \cong \mathrm{SD}(H)$, then that graph is not VSR.

The set of all non-isomorphic graphs of the desired order is generated using Brendan McKay's program `geng`, which is part of the `nauty` package [12]. Lemma 1.4.10

states that if G and H are VSE then $\overline{G}$ and $\overline{H}$ are VSE as well. Therefore, if G is VSR then $\overline{G}$ is VSR as well, and so we need only check one of G or $\overline{G}$ for vertex-switch reconstructibility. This means that we need not check any graph G for which $\varepsilon_G > \frac{1}{4}\nu_G(\nu_G - 1)$. We also have a lower bound on the number of edges of G, as provided by Theorem 4.2.3. Thus we need only search graphs whose number of edges lies in the interval $(\frac{1}{8}\nu_G^2 + \frac{1}{4}\nu_G - 2, \frac{1}{4}\nu_G(\nu_G - 1))$. This can be accomplished by means of a `geng` command-line parameter, namely, "`<lb>:<ub>`", where `<lb>` and `<ub>` are the lower bound and upper bound on ε_G, respectively.

Similarly, Theorem 3.4.4 shows that disconnected graphs are VSR. A command-line switch provided by `geng` (namely "`-c`") allows us to generate only connected graphs.

6.1 The geng Pruning Routine

The `geng` program can be configured to call a user-written "pruning" routine, which can suppress the output of any undesirable graphs. In our case, we will also use this routine to output a list of the potential switch partners for a chosen vertex of each graph which passes all of the easily computable necessary conditions. We now examine these necessary conditions in detail.

Lemma 3.1.2 gives us a fairly simple criterion for excluding a large number of graphs from the search, based solely on their degree sequence. If G is a non-VSR graph, then there must be a graph H which is VSE to G, such that any H-switch partner w of v in G must have $d_G(v) + d_G(w) = \nu_G - 2 + 2e_G(v, w)$. Note that this condition is independent of H. The following algorithm checks for this condition and stores all potential H-switch partners of each vertex in an array.

Algorithm 6.1.1 Finding Potential Switch Partners

Input: G: the graph to check.

Output: $P[\,]$: an array of sets of potential switch partners of each vertex.

```
for v ∈ V(G) do
    P[v] = ∅
    for w ∈ V(G) \ {v} do
        if d_G(v) + d_G(w) = ν_G − 2 + 2e_G(v,w) then
            P[v] = P[v] ∪ {w}
        end if
    next w
next v
```

Once we have this set of potential switch partners, we can use other results to eliminate elements from this set.

Condition 1. We can eliminate pairs $\{v, w\}$ where $|N_G^0(v, w)| = 0$ because of Part (ii) of Lemma 3.2.4.

Condition 2. We can eliminate any potential switch pair which fails the condition specified in Lemma 3.3.2; namely, that the vertices of $N_G^0(v, w)$ are in one-to-one correspondence with the vertices of $N_G^2(v, w)$ such that the degree of each element of $N_G^0(v, w)$ is 2 less than the degree of the corresponding element of $N_G^2(v, w)$. More precisely, we check that $\langle\ d_G(x) + 2 : x \in N_G^0(v, w)\ \rangle = \langle\ d_G(x) : x \in N_G^2(v, w)\ \rangle$.

Condition 3. For each potential H-switch pair $\{v, w\}$ in G, we can add the number of triangles in G that include v but not w to the number of triangles that include w but not v. Then by Lemma 5.4.5, this number must equal the number of edges of G with both endpoints in $\overline{N}_G(v) \setminus \{w\}$ plus the number of edges of G with both endpoints in $\overline{N}_G(w) \setminus \{v\}$. Similarly, Lemma 5.4.6 tells us that the number of non-adjacent pairs of vertices of $\overline{N}_G(v) \setminus \{w\}$ plus the number of non-adjacent pairs of vertices of $\overline{N}_G(w) \setminus \{v\}$ must equal the number of non-adjacent pairs of vertices of

$N_G(v) \setminus \{w\}$ plus the number of non-adjacent pairs of vertices of $N_G(w) \setminus \{v\}$. Any potential switch pair that fails either of these tests is discarded.

Condition 4. For each potential switch pair $\{v, w\}$, we can create the graph $H = G \circledast v \circledast w$. Then if $\nu_G \geq 9$, Lemma 5.4.2 tells us that for each of the 11 graphs of order 4 that are unique up to isomorphism, we can require the number of induced subgraphs of G that are isomorphic to the order-4 graph that contain at least one of v and w to be equal to the number of induced subgraphs of H that are isomorphic to the order-4 graph that contain at least one of v and w. We then discard any switch pair that fails this test.

Since part (i) of Lemma 3.1.2 states that all vertices of our graph must have at least one switch partner, if we find even one vertex with no possible switch partner, we can stop examining G and declare it to be VSR. For the sake of efficiency, we start the search with the vertex that has the fewest potential switch partners, since a smaller set of vertices is more likely to be eliminated quickly than a larger set.

Once we have pared down the list of potential H-switch pairs to those that pass each of the above tests, if the graph has at least one remaining potential switch partner for each vertex, then it is output for further testing, along with a list of all potential switch partners for a chosen vertex. (The chosen vertex is one with the minimum number of potential switch partners.)

Algorithm 6.1.2 CannotBeSwitchPartners(G, v, w, $G_4[11]$)

Input: G : the graph in question.

v, w: a potential switch pair.

$G_4[11]$: an array, indexed by order-4 graph, of the number of induced subgraphs of G isomorphic to the indexing graph. Initially (i.e. during the first call to this routine for a given graph), $G_4[0] = -1$, indicating that the count has not yet been taken.

Output: $R[4]$: an array of boolean values, indexed by condition number, indicating whether the test for each condition has failed. If any element of this array is `true`, then v and w cannot be a switch pair.

```
R[1], R[2], R[3], R[4] ← false

// Check for Condition 1.
N_0 ← 0
DS_0, DS_2 ← ∅
for x ∈ V(G) \ {v, w} do
    if e_G(v, x)e_G(w, x) = 1 then
        DS_2 ← DS_2 ∪ {d_G(x)}
    else if e_G(v, x) + e_G(w, x) = 0 then
        DS_0 ← DS_0 ∪ {d_G(x) + 2}
        N_0 ← N_0 + 1
    end if
next x
if N_0 = 0 then
    R[1] ← true
end if

// Check for Condition 2.
if DS_2 ≠ DS_0 then
    R[2] ← true
end if

// Check for Condition 3.
// Each of the following 4 variables counts the number of induced
      subgraphs of G that contain exactly one element of {v, w},
```

```
      and that have only the specified edge(s).
// T_3 counts those with 3 edges.
// V_3 counts those with 2 edges, both incident with the elt of {v, w}.
// E_3 counts those with no edges.
// I_3 counts those with 1 edge, not incident with the element of {v, w}.
T_3, V_3, E_3, I_3 ← 0
for x ∈ {v, w} do
   for a, b ∈ V(G) \ {v, w} do
      if e_G(x, a) · e_G(x, b) · e_G(a, b) = 1 then
         T_3 ← T_3 + 1
      else if e_G(x, a) · e_G(x, b) · (1 − e_G(a, b)) = 1 then
         V_3 ← V_3 + 1
      else if e_G(x, a) + e_G(x, b) + e_G(a, b) = 0 then
         E_3 ← E_3 + 1
      else if e_G(x, a) + e_G(x, b) − e_G(a, b) = −1 then
         I_3 ← I_3 + 1
      end if
   next a, b
next x
if T_3 ≠ I_3 or E_3 ≠ V_3 then
   R[3] ← true
end if

// Check for condition 4.
Count4Subgraphs(G, G_4)
H ← G ⋇ v ⋇ w
Count4Subgraphs(H, H_4)
```

if $G_4 \neq H_4$ then

  $R[4] \leftarrow$ true

end if

Algorithm 6.1.3 Count4Subgraphs(G, $G_4[11]$)

Input: G : the graph in question.

Output: $G_4[11]$: an array, indexed by order-4 graph, of the number of induced subgraphs of G isomorphic to the indexing graph.

// Store the degree sequence of each isom class of the order-4 graphs.

$A_4[0] \leftarrow \langle 0,0,0,0\rangle$; $A_4[1] \leftarrow \langle 0,0,1,1\rangle$; $A_4[2] \leftarrow \langle 0,1,1,2\rangle$; $A_4[3] \leftarrow \langle 1,1,1,1\rangle$

$A_4[4] \leftarrow \langle 0,2,2,2\rangle$; $A_4[5] \leftarrow \langle 1,1,2,2\rangle$; $A_4[6] \leftarrow \langle 1,1,1,3\rangle$; $A_4[7] \leftarrow \langle 2,2,2,2\rangle$

$A_4[8] \leftarrow \langle 1,2,2,3\rangle$; $A_4[9] \leftarrow \langle 2,2,3,3\rangle$; $A_4[10] \leftarrow \langle 3,3,3,3\rangle$

$G_4[\,] \leftarrow 0$

for $i,j,k,l \in V(G)$ do

  $d_i \leftarrow e_G(i,j) + e_G(i,k) + e_G(i,l)$

  $d_j \leftarrow e_G(j,i) + e_G(j,k) + e_G(j,l)$

  $d_k \leftarrow e_G(k,i) + e_G(k,j) + e_G(k,l)$

  $d_l \leftarrow e_G(l,i) + e_G(l,j) + e_G(l,k)$

  $n \leftarrow h$ such that $A_4[h] = \langle d_i\rangle \uplus \langle d_j\rangle \uplus \langle d_k\rangle \uplus \langle d_l\rangle$

  $G_4[n] \leftarrow G_4[n] + 1$

end do

Note that G and H are two graphs on 4 vertices with the same degree sequence if and only if $G \cong H$, and so here we may safely compare degree sequences rather than entire graphs.

6.2 The IsVSR Program

The output of `geng`, when run with the above pruning algorithm, is a list of potentially non-VSR graphs, along with, for each such graph, a designated vertex and a list of all its potential switch partners. The following algorithm creates, for each input graph G and its designated vertex v, each graph H formed by switching on v and one of its listed potential switch partners. Then for each such H, if $H \ncong G$ then the switch deck of H is compared to the switch deck of G. This is done by successively pairing an unpaired element C of $\mathrm{SD}(G)$ with an unpaired element S of $\mathrm{SD}(H)$ such that $C \cong S$, until no more such pairings are possible. Then the number of such pairings is output, along with a description of G and H. If the number of pairings is the same as the order of G for some H, then G is not VSR.

Note that this algorithm was designed for clarity rather than speed, since, as will be seen in the following section, it only needs to be executed a very small number of times.

Algorithm 6.2.1 IsVSR(G, $P[\,]$)

Inputs: G : the graph in question.

v: the designated vertex.

P: a set of potential switch partners of v.

Output: for each pair $\{v, w\}$ where $w \in P$:

G: the input graph

H: $G \circledast v \circledast w$

I: `true` if and only if $G \cong H$

N: the number of pairings of $\mathrm{SD}(G)$ and $\mathrm{SD}(H)$, whenever $I =$ `false`, or 0 otherwise.

`for` $w \in P$ `do`

  $N \leftarrow 0$

```
    H ← G ⋇ v ⋇ w
    if G ≅ H then
        I ← true
    else
        I ← false
        𝒮 ← SD(H)
        for each C ∈ SD(G)
            if ∃ S ∈ 𝒮 such that S ≅ C then
                𝒮 ← 𝒮 \ {S}
                N ← N + 1
            end if
        next C
    end if
    output N
next w
```

6.3 Search Results

The geng program was run for $\nu_G = 8$ and $\nu_G = 12$.

The following command line was used for the $\nu_G = 8$ case.

```
gengvsr2 -c 8 8:14
```

Here, the "-c" instructs geng to produce only connected graphs; the "8" is the order of the desired graphs, and the "8:14" specifies the lower and upper bound on the number of edges.

In this case, 5827 graphs were examined, and of these, 2065 had at least one potential switch partner for every vertex (such that $d_G(v)+d_G(w) = \nu-2+2e_G(v, w)$).

Conditions passed	# of graphs
1	1493
2	380
1 and 2	53
3	407
1 and 3	67
2 and 3	380
1, 2, and 3	53

Table 6.1: Results of gengvsr2 for order-8 graphs.

The output consisted of 53 graphs, with an aggregate set of 92 potential switch pairs. Table 6.1 gives the number of graphs that passed each combination of conditions.

When these 53 graphs were passed as input to `IsVSR`, the result was that no graphs had 8 pairings. The highest number of pairings was 5, which occurred in 10 different cases.

The following (DOS) command line was used for the $\nu_G = 12$ case.

```
for /L %I in (0,1,999) do
    gengvsr2 -c 12 19:33 %I/1000 >gengvsr2_12_\%I.out
```

This split up the search into 1000 roughly equal pieces. (The "n/1000" command line switch is a feature of `geng`.) The entire search took roughly 130 hours of CPU time on an Intel 6320 processor running at 1.86 GHz, and examined $89,530,434,985$ graphs. The program produced 177 graphs, with an aggregate set of 272 potential switch pairs. Table 6.2 gives the number of order-12 graphs that passed each combination of conditions.

Note that, as shown in Tables 6.1 and 6.2, Condition 3 eliminates no graphs of order 8 or 12 that Condition 2 does not. It seems likely that this is true for graphs

Conditions passed	# of graphs
1	18,595,858,307
2	3,243,322
1 and 2	1,208,103
3	7,364,622
1 and 3	3,452,237
2 and 3	3,243,322
1, 2, and 3	1,208,103
4	29,166
1 and 4	177
2 and 4	29,118
1, 2, and 4	177
3 and 4	29,166
1, 3, and 4	177
2, 3, and 4	29,118
1, 2, 3, and 4	177

Table 6.2: Results of gengvsr2 for order-12 graphs.

Switch cards in common	# of switch pairs
2	9
3	27
4	34
5	15
6	10
7	0
8	7

Table 6.3: Results of IsVSR for order-12 graphs.

of any order.

When these results were passed as input to `IsVSR`, the results were as follows. In 170 cases, the graph produced by switching on the potential switch pair was isomorphic to the original graph. The other 102 cases are described in Table 6.3. In no cases were there 12 switch cards in common between the switch decks of the two graphs.

Assuming the program correctness of `geng`, of the pruning extensions to `geng` implemented in `gengvsr2`, and of `IsVSR`, the results of these searches show that if G is a non-VSR graph with $\nu_G \neq 4$, since $\nu_G \equiv 0 \pmod 4$, then $\nu_G \geq 16$.

Chapter 7

Conclusion

Over the past 25 years, a number of results have been proven about the vertex-switching reconstruction problem, but the question is still far from being solved. In terms of the number of graphs shown to be VSR, Stanley's result (a graph is VSR if its order is not divisible by four) is still by far the most significant. Disconnected graphs, triangle-free graphs, and regular graphs have also been shown to be VSR, but these classes represent very few graphs (for example, of the graphs of order 12, these classes taken together account for less than 0.7% of the total [13]).

Other results have shed light on the structure of a non-VSR graph of order > 4 (should such a graph exist), and might be useful in proving the non-existence of such graphs. The most significant among these is the fact that each vertex of such a graph G is a member of at least one switch pair (i.e., a pair of vertices which, when switched on in G, produce a graph which is isomorphic to some particular graph which has the same switch deck as G). The fact that the number of induced subgraphs of a graph G that are isomorphic to another graph S can be gleaned from G's switch deck (provided $\nu_S < \nu_G/2$) appears to be sufficient to show a large number of graphs to be VSR, based on the results of the computer search presented here.

The contributions of this thesis, apart from the clarification (and in some cases simplification) of the published literature, include an improved bound on the number of edges in a non-VSR graph, the first known bound on the order of the automorphism group of a non-VSR graph, and an enumerative computer verification that all graphs of order 8 or 12 are VSR. A few new structural results were proven as well. Given a switch pair $\{v, w\}$ in a non-VSR graph G, we know that adding 2 to the degree of each vertex that is adjacent to neither vertex of the switch pair produces the multiset of degrees of the vertices adjacent to both vertices of the switch pair. We also know that the multiset of degrees of neighbours of v in G equals the multiset of degrees of neighbours of w in $G \circledast v \circledast w$.

A few areas of research appear promising. The implications of Abatangelo and Dragomir's result (regarding the existence of a non-VSR pair of graphs implying the existence of a graph with certain properties—see Section 2.6) are currently unclear. The uses of Krasikov and Roditty's balance equations have not yet been fully explored. The last section of the proof (Theorem 5.3.7) that triangle-free graphs are VSR hints at a possible extension to Lemma 3.2.4; for instance, there might be more that can be said about the set of vertices adjacent to both members of a switch pair and the set of vertices adjacent to neither members of the switch pair other than the fact that the cardinality of these sets must be equal. It might be possible to show that if a non-VSR graph (other than those of order 4) exists, then its vertices could be partitioned into sets of switch pairs. Such a result would lead to a number of other results that would restrict the structure of a non-VSR graph. Very little work has been done concerning the automorphism group of a non-VSR graph, an important concept in the context of vertex-switch equivalence which should be explored further.

Appendix A

gengvsr2 Program Source Code

```
#include <stdio.h>
#include <string.h>
#include "nauty.h"

#define N4GRAPHS 11
#define NROOTED3GRAPHS 6
#define CONDITION_1 1
#define CONDITION_2 2
#define CONDITION_3 3
#define CONDITION_4 4

typedef struct switchpartner {
  int w;
  int bViable;
} Switchpartner;

int bFirstTime = 1;

int gnCondition[4];
int gnConditions;
int gnGraphsSeen;
```

```
int gnGraphsPassingDSTest;
int gnGraphsPassingTest[4];

void initializeVSR(int n)
{
  int i;
  char s[8];

  gnGraphsSeen = 0;
  gnGraphsPassingDSTest = 0;
  for (i=0; i<16; i++) {
    gnGraphsFailingTestSet[i] = 0;
  }

  // Get, from stdin, a list of conditions to test:
  scanf("%4s", &s);
  if (feof(stdin)) {
    strcpy(s, "1234");
  }
  gnConditions = strlen(s);
  for (i=0; i<gnConditions; i++) {
    gnCondition[i] = s[i] - '0';
    if ((gnCondition[i] < 0) || (gnCondition[i] > (n > 8 ? 4 : 3))) {
      fprintf(stderr, "Invalid␣condition␣number:␣%c\n", s[i]);
      exit(-2);
    }
    gnGraphsPassingTest[i] = 0;
  }
}

void dumpGraph(graph *g, int n)
{
  int i,j;
```

```
  set *gi;
  int bFirst;

  for (i=0; i<n; i++) {
    gi = GRAPHROW(g,i,1);
    printf("%d:", i);
    bFirst = 1;
    for (j=i+1; j<n; j++) {
      if (ISELEMENT(gi,j)) {
        if (!bFirst) {
          printf("␣");
        }
        printf("%d", j);
        bFirst = 0;
      }
    }
    printf(";␣");
  }
  printf("\n");
}


int degreeOf(graph *g, int v)
{
  set *gv = GRAPHROW(g,v,1);
  return(POPCOUNT(*gv));
}

// Given a set of bits respresenting the edges of a (labelled)
//   4-vertex graph, this array gives an index indentifying the
//    unlabelled graph.
int canonical4Graph[64] =  {0,1,1,2, 1,2,2,6, 1,2,2,4, 3,5,5,8,
              1,2,3,5, 2,4,5,8, 2,6,5,8, 5,8,7,9,
```

```
             1,3,2,5, 2,5,4,8, 2,5,6,8, 5,7,8,9,
             2,5,5,7, 6,8,8,9, 4,8,8,9, 8,9,9,10};

// The degree sequences of the 11 unlabelled graphs on 4 vertices,
//  which happen to be unique and thus can identify the graph.
int dsCanonical4Graph[N4GRAPHS][4] = {
  {0,0,0,0},     // E4
  {0,0,1,1},     // single edge
  {0,1,1,2},     // path of length 2 plus a vertex
  {1,1,1,1},     // 2 disjoint paths of length 1
  {0,2,2,2},     // K3 plus 1 vertex
  {1,1,2,2},     // path of length 3
  {1,1,1,3},     // star graph
  {2,2,2,2},     // 4-cycle
  {1,2,2,3},     // K3 plus 1 edge
  {2,2,3,3},     // 4-cycle plus an edge
  {3,3,3,3}};    // K4


void degreeSequenceOf4Subgraph(graph *g, int n, int i, int j, int k,
                 int l, int ds[])
{
  // Returns the sorted (ascending) degree sequence of the subgraph
  //  of g induced by the vertices i, j, k, and l.
  int t;

  ds[0]=ISELEMENT(g+i,j)+ISELEMENT(g+i,k)+ISELEMENT(g+i,l);
  ds[1]=ISELEMENT(g+j,i)+ISELEMENT(g+j,k)+ISELEMENT(g+j,l);
  ds[2]=ISELEMENT(g+k,i)+ISELEMENT(g+k,j)+ISELEMENT(g+j,l);
  ds[3]=ISELEMENT(g+l,i)+ISELEMENT(g+l,j)+ISELEMENT(g+l,k);
  // sort the degree sequence of the induced subgraph
#define BUB_SORT(a,b) if (ds[a]>ds[b]){t=ds[a];ds[a]=ds[b];ds[b]=t;}
  BUB_SORT(0,1);
```

```
  BUB_SORT(1,2);
  BUB_SORT(2,3);
  BUB_SORT(0,1);
  BUB_SORT(1,2);
  BUB_SORT(0,1);
}

void count4SubgraphsIncluding(graph *g, int n, int v, int w,
                int g4Count[])
{
  // Count the number of each of the 11 unlabelled graphs on
  //  4 vertices as they appear as induced subgraphs of g.
  // Each induced subgraph must include vertex v and may not
  //   include vertex w.
  // The result is returned in ds[].
  int j,k,l;
  int s1, s2, s3;
  set *gv, *gj, *gk;
  int ds4[4];
  int h;

  gv = g+v;
  for (j=0; j<n-2; j++) {
    if ((j != v) && (j != w)) {
      gj = g+j;
      s1 = (ISELEMENT(gv,j) ? 1 : 0);
      for (k=j+1; k<n-1; k++) {
        if ((k != v) && (k != w)) {
          gk = g+k;
          s2 = (ISELEMENT(gv,k)?2:0) + (ISELEMENT(gj,k)?8:0);
          for (l=k+1; l<n; l++) {
            if ((l != v) && (l != w)) {
              s3 = (ISELEMENT(gv,l) ? 4 : 0)
```

```
                  + (ISELEMENT(gj,l) ? 16 : 0)
                  + (ISELEMENT(gk,l) ? 32 : 0);

              // Find the deg seq of the induced subgraph
              degreeSequenceOf4Subgraph(g, n, v,j,k,l, ds4);

              // Make sure the canonical 4-graph number
              //  obtained from the degree sequence equals
              //  the one obtained by the array lookup.
              for (h=0; h<N4GRAPHS; h++) {
                if ((dsCanonical4Graph[h][0] == ds4[0]) &&
                  (dsCanonical4Graph[h][1] == ds4[1]) &&
                  (dsCanonical4Graph[h][2] == ds4[2]) &&
                  (dsCanonical4Graph[h][3] == ds4[3])) {
                    if (canonical4Graph[s1+s2+s3] != h) {
                      printf("INTERNAL␣ERROR!\n");
                      exit(-1);
                    }
                }
              }
              g4Count[canonical4Graph[s1+s2+s3]]++;
            }
          }
        }
      }
    }
  }
}

void countRooted3SubgraphsAvoiding(graph *g, int n, int v, int w,
                   int g3Count[])
{
  // Count the number of each of the 6 rooted graphs on 3 vertices
```

```
//  as they appear as induced subgraphs of G
//  which are rooted at v and don't contain w.

int i;
int j;
set *gi;
set *gv;
int bEdgevi, bEdgevj;
int nSubgraph;

for (i=0; i<NROOTED3GRAPHS; i++) {
  g3Count[i] = 0;
}

gv = GRAPHROW(g,v,1);
for (i=0; i<n; i++) {
  if ((i != v) && (i != w)) {
    gi = GRAPHROW(g,i,1);
    bEdgevi = ISELEMENT(gv,i);
    for (j=i+1; j<n; j++) {
      if ((j != v) && (j != w)) {
        bEdgevj = ISELEMENT(gv,j);
        if (ISELEMENT(gi, j)) { // there is an edge ij
          if (bEdgevi && bEdgevj) {
            nSubgraph = 5;  // K3
          } else if (bEdgevi || bEdgevj) {
            nSubgraph = 3;  // 2-path, v at one endpoint
          } else {
            nSubgraph = 1;  // one edge, not incident with v
          }
        } else {  // no induced subgraph edge not incident with v
          if (bEdgevi && bEdgevj) {
            nSubgraph = 4;  // 2-path, v in the centre
```

```
          } else if (bEdgevi || bEdgevj) {
            nSubgraph = 2;  // one edge, incident with v
          } else {
            nSubgraph = 0;  // E3
          }
        }
        g3Count[nSubgraph]++;
      }
    }
  }
 }
}

void degreeSequenceOfGraph(graph *g, int n, int ds[])
{
  int i;

  for (i=0; i<n; i++) {
    ds[i] = degreeOf(g,i);
  }
}

int unequalDegreeSequences(int ds1[], int ds2[], int n)
{
  // Return true iff ds1 and ds2 represent equivalent sets.
  // ds1 and ds2 don't have to be sorted.
  // ds2 will be destroyed.

  int i,j;
  int n2;
  int bFound;

  n2 = n;
```

```
  for (i=0; i<n; i++) {
    bFound = 0;
    for (j=0; j<n2; j++) {
      if (ds1[i] == ds2[j]) {
        // delete the found element from ds2
        n2--;
        ds2[j] = ds2[n2];
        bFound = 1;
        break;
      }
    }
    if (!bFound) {
      return(-1);
    }
  }
  return(0);
}

void switchVertex(graph *g, int n, int v, graph *card)
{
  // Switch on the vertex v in g and store the result in card.
  int i;
  set *gi;
  set *cardi;

  for (i=0; i<n; i++) {
    gi = GRAPHROW(g, i, 1);
    cardi = GRAPHROW(card, i, 1);

    *cardi = *gi;
    if (i == v) {
      *cardi ^= ~((1<<(32-n))-1); // flip the row of the adj mtx
    }
```

```
    FLIPELEMENT(cardi, v);  // flip the col of the adj mtx
  }
}

void dumpVertexAndSPs(int n, int v, int nSPs, Switchpartner nSP[])
{
  int i;
  int bSP[MAXN];

  for (i=0; i<n; i++) {
    bSP[i] = 0;
  }
  for (i=0; i<nSPs; i++) {
    if (nSP[i].bViable) {
      bSP[nSP[i].w] = 1;
    }
  }

  printf("%d ", v);
  for (i=0; i<n; i++) {
    printf("%d ", bSP[i]);
  }
}

int pruneVSR2(graph *g, int n, int maxn)
{
  // Return 0 iff g has at least one potential switch pair.
  int v;
  int w;
  Switchpartner nPotentialSP[MAXN][MAXN];
    // List of verts which might be a SP of each vert.
  int nPotentialSPs[MAXN];
    // Number of elements in each nPotentialSP[].
```

```
int nMinSPs = MAXN+1; // Smallest value in nPotentialSPs[].
int nVertexSPCountOrder[MAXN];  // Verts of g ordered by # of SPs.
graph card[MAXN][MAXN]; // g vswitch v.
set *gv;         // Row v of graph g.
set *gw;         // Row w of graph g.
int gDS[MAXN];       // The degree sequence of g.
int nCount4SubgraphsInG[N4GRAPHS];  // For speed.
  // We only need to count the 4-subgraphs of this g once.
int r;
int bSomeVertHasNoSP;
int nSPsLeftForVert;
int nBestSurvivingVertex;
int nFewestSPsLeft;
int i,j,k;
int t;
int c;
int bBothOrNeither;
int nBoth;
int nNeither;
int nBothDS[MAXN];    // Degs of verts adj to both v and w in g.
int nNeitherDS[MAXN]; // Degs of verts adj to neither v nor w in g.
int n3Subgraphs[MAXN][NROOTED3GRAPHS];
  // Number of each type of 3-vertex graph rooted at v
  //    and avoiding w as an induced subgraph of g.
int nCount4SubgraphsInH[N4GRAPHS];
  // Number of each type of 4-vertex graph
  //    as an induced subgraph of g.
graph h[MAXN];
int bTestedVertex[MAXN];
int jj;

if (n == maxn) { // ignore this call if we're not at the tree bottom
```

```
if (bFirstTime) {
  initializeVSR(n);
  bFirstTime = 0;
}

nCount4SubgraphsInG[0] = -1;  // flag this array as uninitialized
gnGraphsSeen++;
degreeSequenceOfGraph(g,n,gDS);

// Count the number of potential switch partners for each vertex
//    (based only on degrees of vertices):
for (v=0; v<n; v++) {
  nPotentialSPs[v] = 0;
  for (w=0; w<n; w++) {
    if (w != v) {
      if (((gDS[v] + gDS[w] == n)&& ISELEMENT(g+v,w)) ||
        ((gDS[v] + gDS[w] == n-2) && !ISELEMENT(g+v,w))) {
        nPotentialSP[v][nPotentialSPs[v]].w = w;
        nPotentialSP[v][nPotentialSPs[v]].bViable = 1;
        nPotentialSPs[v]++;
      }
    }
  }
  // Find the vertex with the fewest possible switch partners:
  if (nPotentialSPs[v] == 0) {
    return(1);  // prune this graph
    // (Vertex v has no switch partners, so G must be VSR.)
  }
  nVertexSPCountOrder[v] = v;
  card[v][0] = -1;  // mark this vert's card as 'uncomputed'
}
gnGraphsPassingDSTest++;
```

```
// Now sort the vertices according to the # of switch pairs.
// This will help find verts with no actual switch partners
//    more quickly.
for (i=0; i<n-1; i++) {
  for (j=0; j<n-i-1; j++) {
    if (nPotentialSPs[nVertexSPCountOrder[j]] >
      nPotentialSPs[nVertexSPCountOrder[j+1]]) {
        t = nVertexSPCountOrder[j];
        nVertexSPCountOrder[j] = nVertexSPCountOrder[j+1];
        nVertexSPCountOrder[j+1] = t;
    }
  }
}

bSomeVertHasNoSP = 0;
for (c = 0; c < gnConditions; c++) {
  nFewestSPsLeft = n+1;
  for (v=0; v<n; v++) {
    bTestedVertex[v] = 0;
  }
  for (k=0; k<n; k++) {
    v = nVertexSPCountOrder[k];
    gv = GRAPHROW(g,v,1);
    // Compute the card if it hasn't been done already:
    if (card[v][0] == -1) {
      switchVertex(g,n,v,card[v]);
    }

    nSPsLeftForVert = 0;
    for (i=0; i<nPotentialSPs[v]; i++) {
      if (nPotentialSP[v][i].bViable) {
        w = nPotentialSP[v][i].w;
        if (bTestedVertex[w]) {
```

```
      jj = -1;
      for (j=0; j<nPotentialSPs[w]; j++) {
        if (nPotentialSP[w][j].w == v) {
          jj = j;
          break;
        }
      }
      if (jj == -1) {
        fprintf(stderr,
          "ERR␣-␣no␣matching␣switch␣partner.\n");
        exit(-3);
      }
      r = (nPotentialSP[w][jj].bViable
          ? 0 : gnCondition[c]);
    } else {
      r = 0;
      gw = GRAPHROW(g,w,1);
      switch(gnCondition[c]) {
      case CONDITION_1:
        bBothOrNeither = 0;
        for (j=0; j<n; j++) {
          if ((j != v) && (j != w)) {
            // If vtx j is adj to 0 or 2 of v,w
            //   then w can still be a SP of v.
            if ((ISELEMENT(gv,j) != 0) ==
                (ISELEMENT(gw,j) != 0)) {
              bBothOrNeither = 1;
              break;  // vtx v has a SP by cond1
            }
          }
        }
        if (bBothOrNeither == 0) {
          r = CONDITION_1;
```

```
    }
    break;
  case CONDITION_2:
    nBoth = 0;
    nNeither = 0;
    for (j=0; j<n; j++) {
      if (ISELEMENT(gv,j) && ISELEMENT(gw,j)) {
        nBothDS[nBoth] = gDS[j];
        nBoth++;
      } else if ((!ISELEMENT(gv,j)) &&
            (!ISELEMENT(gw,j))
          && (j != v) && (j != w)) {
        nNeitherDS[nNeither] = gDS[j] + 2;
        nNeither++;
      }
    }
    // (redundancy) Check that nBoth = nNeither:
    if (nBoth != nNeither) {
      printf("ERROR!␣nBoth␣!=␣nNeither.\n");
      exit(-1);
    }

    // Ensure adding 2 to each deg of the
    //       non-nbrs of {v,w} gives the degs of
    //     the nbrs of both v and w.
    if (unequalDegreeSequences(nBothDS,
          nNeitherDS, nBoth)) {
      r = CONDITION_2;
    }
    break;
  case CONDITION_3:
    countRooted3SubgraphsAvoiding(g,n,v,w,
        n3Subgraphs[v]);
```

```
    countRooted3SubgraphsAvoiding(g,n,w,v,
        n3Subgraphs[w]);

    // Count E3's, and 2-paths centred on v:
    if (n3Subgraphs[v][0]+n3Subgraphs[w][0] !=
        n3Subgraphs[v][4]+n3Subgraphs[w][4]) {
      r = CONDITION_3;
      break;
    }

    // Count K3's, and 1-paths avoiding v:
    if (n3Subgraphs[v][1]+n3Subgraphs[w][1] !=
        n3Subgraphs[v][5]+n3Subgraphs[w][5]) {
      r = CONDITION_3;
      break;
    }
    break;
  case CONDITION_4:
    // Count 4-subgraphs:
    switchVertex(card[v],n,w,h);

    for (j=0; j<N4GRAPHS; j++) {
      nCount4SubgraphsInG[j] = 0;
    }

    count4SubgraphsIncluding(g, n, v, w,
        nCount4SubgraphsInG);
    count4SubgraphsIncluding(g, n, w, -1,
        nCount4SubgraphsInG);

    for (j=0; j<N4GRAPHS; j++) {
      nCount4SubgraphsInH[j] = 0;
    }
```

```
          count4SubgraphsIncluding(h, n, v, w,
              nCount4SubgraphsInH);
          count4SubgraphsIncluding(h, n, w, -1,
              nCount4SubgraphsInH);

          for (j=0; j<N4GRAPHS; j++) {
            if (nCount4SubgraphsInG[j] !=
                nCount4SubgraphsInH[j]) {
              r = CONDITION_4;
              break;
            }
          }
          break;
        }
      }

      if (r != 0) {
        // Mark this SP as impossible:
        nPotentialSP[v][i].bViable = 0;
      } else {
        nSPsLeftForVert++;
      }

    }  // if w >= 0
  } // for i
  if (nSPsLeftForVert == 0) {
    return(1);     // prune this graph
  }

  gnGraphsPassingTest[c]++;
  if (nSPsLeftForVert < nFewestSPsLeft) {
    nFewestSPsLeft = nSPsLeftForVert;
    nBestSurvivingVertex = v;
```

```
        }
        bTestedVertex[v] = 1;
      } // for k (v)
    } // for c

    // This graph might have a SP, and thus might be non-VSR.
    dumpVertexAndSPs(n, nBestSurvivingVertex,
        nPotentialSPs[nBestSurvivingVertex],
        nPotentialSP[nBestSurvivingVertex]);
    return(0);
  }
  return(0);  // don't prune if we're not at tree bottom (n verts)
} // pruneVSR2

typedef struct
{
    long hi,lo;
} bigint;

void summaryVSR(bigint nout, double cpu)
{
  int c;

  printf("Tests:␣");
  for (c=0; c<gnConditions; c++) {
    printf("%d␣", gnCondition[c]);
  }
  printf("Graphs:␣%d␣%d␣", gnGraphsSeen, gnGraphsPassingDSTest);
  for (c=0; c<gnConditions; c++) {
    printf("%d␣", gnGraphsPassingTest[c]);
  }
  printf("\n");
}
```

Appendix B

IsVSR Program Source Code

```
#include <stdio.h>
#include <string.h>
#include "..\\gtools.h"
#include "..\\gengvsr2\\gengvsr2\\prunevsr2.h"

// Usage: IsVSR <n> [<input file name>]
// Read a list of graphs from stdin and check whether each is VSR.
// A graph is represented by a line of input.
// The line consists of a vertex number (v), followed by
//   n binary values. Each of these is 1 iff
//   the corresponding vertex can be a switch partner
//   of the vertex v.
// This is followed by the certificate of the graph,
//   as produced by geng.

int compareGraphs(graph *g1, graph *g2, int n)
{
  // Return 0 if g1 = g2, -1 if g1 < g2, or +1 if g1 > g2.
  int i;

  for (i=0; i<n; i++) {
```

```
    if (g1[i] < g2[i]) {
      return(-1);
    } else if (g1[i] > g2[i]) {
      return(1);
    }
  }

  return(0);
}

void copyGraph(graph *gSource, graph *gTarget, int n)
{
  // Copy gSource to gTarget.
  int i;

  for (i=0; i<n; i++) {
    gTarget[i] = gSource[i];
  }
}

void sortCards(graph card[][MAXN], int n)
{
  // Put the graphs in card[] in canonical order.
  int i,j;
  int r;
  graph t[MAXN];

  for (i=n-1; i>0; i--) {
    for (j=0; j<i; j++) {
      r = compareGraphs(card[j], card[j+1], n);
      if (r > 0) {
        copyGraph(card[j+1], t, n);
        copyGraph(card[j], card[j+1], n);
```

```
        copyGraph(t, card[j], n);
      }
    }
  }
}

int commonCards(graph g1[][MAXN], graph g2[][MAXN], int n)
{
  // Return the number of cards that g1[] and g2[] have in common.
  // The two decks must have their cards in canonical order.
  int i1=0, i2=0;
  int r;
  int c = 0;

  while ((i1 < n) && (i2 < n)) {
    r = compareGraphs(g1[i1], g2[i2], n);
    if (r < 0) {
      i1++;
    } else if (r > 0) {
      i2++;
    } else {
      c++;
      i1++;
      i2++;
    }
  }
  return(c);
}

void getCanonicalGraph(graph *g, int n, graph *gCanon)
{
  // Order the vertices of g canonically.
  int lab[MAXN], ptn[MAXN], orbits[MAXN];
```

```
  static DEFAULTOPTIONS(options);
  statsblk(stats);
  setword workspace[1000];

  options.getcanon = TRUE;
  nauty(g,lab,ptn,NULL,orbits,&options,&stats,workspace,
    sizeof(workspace)/sizeof(workspace[0]), 1,n, gCanon);
  if (stats.errstatus != 0) {
    printf("ERROR!\n");
    exit(-1);
  }

}

main(int argc, char **argv)
{
  graph g[MAXN];
  int n;
  int v;
  int i,j;
  int c;
  int bPossibleSP;
  int nPossibleSP[MAXN];  // list of possible switch partners
  int nPossibleSPs; // # of possible switch partners of given vtx
  int mDummy, nDummy; // used in calling nauty
  graph card[MAXN]; // a single switch card of G
  graph gCard[MAXN][MAXN];  // G switch v, for each v
  graph gCardCanon[MAXN][MAXN]; // SD(G)
  graph h[MAXN];  // potentially VSE to G
  graph gCanon[MAXN]; // G with vertices in a canonical order
  graph hCanon[MAXN]; // H with vertices in a canonical order
  graph hCardCanon[MAXN][MAXN]; // SD(H)
  FILE *f;
```

```
sscanf(argv[1], "%d", &n);
nauty_check(WORDSIZE, 1, n, NAUTYVERSIONID);

f = (argc > 1 ? fopen(argv[2], "r") : stdin);

while (!feof(f)) {
  if (fscanf(f, "%d", &v) == EOF) {
    break;
  }
  nPossibleSPs = 0;
  for (i=0; i<n; i++) {
    fscanf(f, "%d", &bPossibleSP);
    if (bPossibleSP) {
      nPossibleSP[nPossibleSPs++] = i;
    }
  }
  getc(f);  // swallow the blank
  readg(f, g, 0, &mDummy, &nDummy);

  // Create the switch deck of g:
  for (i=0; i<n; i++) {
    switchVertex(g,n,i,gCard[i]);
    getCanonicalGraph(gCard[i], n, gCardCanon[i]);
  }
  sortCards(gCardCanon, n);

  getCanonicalGraph(g,n,gCanon);
  for (i=0; i<nPossibleSPs; i++) {

    // Create H and output G and H.
    switchVertex(gCard[v], n, nPossibleSP[i], h);
    getCanonicalGraph(h,n,hCanon);
```

```
      printf("G = ");
      dumpGraph(g, n);
      printf("  switch %d,%d gives H = ", v, nPossibleSP[i]);
      dumpGraph(h, n);

      // Check if G and H are isomorphic:
      if (compareGraphs(gCanon,hCanon,n) != 0) {

        // Create SD(H) and compare it to SD(G):
        for (j=0; j<n; j++) {
          switchVertex(h,n,j,card);
          getCanonicalGraph(card,n,hCardCanon[j]);
        }
        sortCards(hCardCanon, n);
        c = commonCards(gCardCanon, hCardCanon, n);
        printf(" %d cards in common.\n", c);

      } else {
        printf(" isomorphic.\n");
      }

    }
  }

  if (f != stdin) {
    fclose(f);
  }

  return(0);
}
```

Bibliography

[1] Abatangelo, Vito and Dragomir, Sorin, "Discrete Fourier Calculus and Graph Reconstruction", *Interdisciplinary Information Sciences*, **13** (2007), No. 2, pp. 163–180

[2] Diaconis, P. and Graham, R.L., "The Radon Transform On $\mathbb{Z}_2^k$", *Pacific Journal of Mathematics*, **118** (1985), No. 2, pp. 323–345

[3] Ellingham, M.N., "Vertex-switching, isomorphism and pseudosimilarity", *Journal of Graph Theory*, **15** (1991), No. 6, pp. 563–572

[4] Ellingham, M.N., "Vertex-Switching Reconstruction and Folded Cubes", *Journal of Combinatorial Theory*, Series B, **66** (1996), pp. 361–364

[5] Ellingham, M.N. and Royle, Gordon F., "Vertex-Switching Reconstruction of Subgraph Numbers and Triangle-Free Graphs", *Journal of Combinatorial Theory*, Series B, **54** (1992), pp. 167–177

[6] Krasikov, I., "A Note On The Vertex-Switching Reconstruction", *International Journal of Mathematics and Mathematical Sciences*, **11** (1988), No. 4, pp. 825–827

[7] Krasikov, I., "Applications of Balance Equations to Vertex Switching Reconstruction", *Journal of Graph Theory*, **18** (1994), No. 3, pp. 217–225

[8] Krasikov, I., "Degree conditions for vertex switching reconstruction", *Discrete Mathematics*, **160** (1996), pp. 273-278

[9] Krasikov, I., and Roditty, Y., "Balance equations for reconstruction problems", *Archiv der Mathematik*, **48** (1987), pp. 458–464

[10] Krasikov, I., and Roditty, Y., "Switching Reconstruction and Diophantine Equations", *Journal of Combinatorial Theory*, Series B, **54** (1992), pp. 189–195

[11] Krasikov, I., and Roditty, Y., "More on Vertex-Switching Reconstruction", *Journal of Combinatorial Theory*, Series B, **60** (1994), pp. 40–55

[12] McKay, Brendan, "`nauty` User's Guide (Version 2.2)", `http://cs.anu.edu.au/~bdm/nauty/nug.pdf`

[13] Sloane, N.J.A., "The On-Line Encyclopedia of Integer Sequences", `http:/www.research.att.com/~njas/sequences/`

[14] Stanley, Richard P., "Reconstruction From Vertex-Switching", *Journal of Combinatorial Theory*, Series B, **38** (1985), pp. 132–138

[15] Taussky, Olga, "A Recurring Theorem On Determinants", *The American Mathematical Monthly*, **56** (1949), No. 10, pp. 672–676