Extensions of the Tutte Polynomial and Results on the Interlace Polynomial

by

Josephine Elizabeth Anne Reynes

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2025

© Josephine Elizabeth Anne Reynes 2025

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Thomas Zaslavsky Professor, Dept. of Mathematics and Statistics, Binghamton University
Supervisor(s):	Karen Yeats Professor, Dept. of Combinatorics and Optimization, University of Waterloo Logan Crew Assistant Professor, Dept. of Combinatorics and Optimization, University of Waterloo
Internal Member:	Oliver Pechenik Assistant Professor, Dept. of Combinatorics and Optimization, University of Waterloo
Internal Member:	Jonathan Leake Assistant Professor, Dept. of Combinatorics and Optimization, University of Waterloo
Internal-External Member:	Matthew Satriano Associate Professor, Dept. of Pure Mathematics, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In graph theory, graph polynomials are an important tool to encode information from a graph. The Tutte polynomial, first introduced in 1947, is one of the most important graph polynomials due to its universality. Here, we present three classic definitions of the Tutte polynomial via a deletion-contraction recursion, via rank and nullity, and via activities. We will touch on the significance of this polynomial to the field of mathematics to motivate an extension to signed graphs. Extending the polynomial to retain deletion contraction and inactivity information, we introduce an extended Tutte polynomial to allow for the construction of a Tutte like polynomial on signed graphs. Using the extended information, we examine the monomials of these polynomials as grid walks. Using grid walking and the extended Tutte polynomial, we investigate the relationship between the Tutte polynomial of a graph and that of its bipartite representation. This is done with a view toward the construction of a Tutte like polynomial for oriented hypergraphs.

While many graph polynomials are directly related to the Tutte polynomial, there are also a wide variety of polynomials related in special cases only. One such polynomial is the Martin polynomial and, related to it, the interlace polynomial. Here, we discuss how these two polynomials are related and how results on the Martin polynomial can be extended to the interlace polynomial. The Martin invariant, a specific evaluation of the Martin polynomial, obeys the symmetries of the Feynman period. The Feynman period of a graph is useful in quantum field theory, but difficult to compute and thus there is interest in finding graph invariants that have the same symmetries. It was established that the interlace polynomial on interlace graphs was equal to the Martin polynomial on the associated 4-regular graph. While only graphs that do not contain a set of forbidden vertex minors are interlace graphs, the interlace polynomial is defined over all graphs. We discuss how this provides a way to try and extend the notion of Feynman symmetries via the interlace polynomial and some specific classes of graphs with formulas. Additionally, the interlace polynomial is only equal to the Martin polynomial for interlace graphs of 4-regular graphs, but the Martin polynomial is defined for $2k$-regular graphs. Thus, we work toward creating an interlace-like polynomial for graphs derived from $2k$-regular cases of the Martin polynomial.

Acknowledgements

I would like to begin by thanking Dr. Karen Yeats. Their support has been invaluable and without them this thesis would not exist. Throughout this process you have supported me when I've hit research roadblocks and with personal difficulties. Additionally, through both individual and group meetings you have provided valuable personal and development discussions that have made me a better academic in my teaching, research, and personal life. You have helped make the dream I had for my future a reality. I intend to follow your example and use all the knowledge you have shared with me to show my own students the kindness you showed me. So thank you, from the bottom of my heart.

To my doctoral committee, who read what turned out to be a longer thesis than I was expecting, thank you for the time and effort you have put into reading this work and the ways in which your comments made it better. While I have enjoyed my time here, I cannot stay forever and this thesis was the final step. Thank you all for your time. To Dr. Oliver Pechenik, Dr. Sophie Spirkl, and Dr. Martin Pei, thank you all for writing me letters for job applications. That process was terrifying and I appreciate your support. I'll never know exactly what impact these letters had on my applications, but I have job so they must have helped. I would also like to thank Dr. Eric Panzer for his work with the Martin invariant and the insights he had in a meeting that led to the work I have done with the interlace polynomial. I'm so grateful to have been able to do the work he inspired.

To Dr. Lucas Rusnak, who was my master's thesis advisor and now my friend, thank you. I came to Waterloo because of your support in my undergraduate and master's degrees and despite me no longer being your student you continued to support my career. Working with you has been, and continues to be, a marvelous experience. I have come to deeply love oriented hypergraphs and the incidence structure. I hold contributors and cross-thetas as some of the most fun objects to work with, and I night never know of these topics if it hadn't been for you. I'll never forget coming into your office as an undergrad, when I had been warned you were a scary professor, and asking to do research anyway because I saw your passion in your teaching. You were a wonderful instructor and advisor and I hope to give to my own students the passion, compassion, and knowledge you gave to me. I also think you deserve credit for pushing Maria and I together and so, in a way you helped me find my best friend. I will be forever grateful for knowing you and being able to count you as a friend.

To Maria, who is even better than a best friend. When we moved in together during COVID, on Lucas's advice, I was nervous. I barely knew you and yet we slotted together like we had know each other forever. For the last four years we have lived thousands of

kilometers apart, and yet distance has not dulled our friendship. You have helped me practice presentations and listened to my struggles and doubts. We have played video games and watched movies together and made time for each other on holidays. I still dream of the pickles and jams we made and miss sharing meals with you. Though you are far away physically, you have never been far in my heart and I am beyond grateful to have a friend with whom I feel so at home. This world is a strange place, but it is better for having you. You have given me so much joy and I hope that our dreams of the future come true.

To my family, who always supported my goals. Mom, I am in large part where I am because you supported my love of math. Thank you for never trying to change that part of me. Grandma, I love that you are always just a phone call away and that you will listen to me explain math regardless of whether or not you understood. It is such a gift to have you listen to me. Thank you both for every book and puzzle and for years of happy memories. To the rest of my family, I have always lived secure in the knowledge that I like my family as people and that is truly a gift. Seeing you at Thanksgiving and Christmas makes my year better.

Also, a special thanks to Elise, who doesn't know math but helped me edit some of this and is wonderful to chat with on the phone. I've called you most weeks as I've written this thesis and your support has meant the world to me. While we both love books, I suspect this is the longest thing I will ever write and you helped make it better.

To all my friends, of whom there are too many to reasonably name. Knowing each and every one of you has been a joy and I hope our friendships will continue forever. For those here at Waterloo, you made pursing my doctorate much more fun and I am a better person for all the wonderful people from around the world I met here. For those friends from before Waterloo, you have continued to support me from thousands of kilometers away. I'm so grateful to have preserved our friendship despite time and distance. To all of you, thank you for letting me know you.

Dedication

To my grandmother,
Who filled my shelves with books,
and has been with me for every step of my story.
This chapter ends here,
but "beyond the wild woods come the wild world" [59]
and through it I'll carry you always.

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Abstract iv

Acknowledgements v

Dedication vii

List of Figures x

List of Tables xv

1 Introduction 1

2 Background 10

2.1 Graphs 10

2.1.1 Bidirected and Signed Graphs 18

2.1.2 Bipartite Representations of Graphs and Oriented Hypergraphs . . . 26

2.1.3 Graphic Matroids 33

2.2 Tutte Polynomial 34

2.3 Martin Polynomial 42

2.4 Interlace Polynomial 50

3 Tutte-type Polynomials **57**

3.1 An Extended Tutte Polynomial . . . 57

3.2 An Evaluation of the Kirchhoff Polynomial . . . 67

3.3 A Grid Walking Interpretation . . . 72

3.3.1 Special Classes of Graphs . . . 90

3.4 An Extended Tutte Polynomial for Signed Graphs . . . 93

3.4.1 A Missing Operation . . . 93

3.4.2 An Extended Tutte Polynomial for Signed Graphs . . . 98

3.4.3 Grid walking . . . 107

3.5 Bipartite Representations of Graphs . . . 112

3.5.1 Grid Walking . . . 117

4 Interlace Polynomial **120**

4.1 Interlace Invariant Special Cases . . . 127

4.2 Symmetries and the Interlace Invariant . . . 144

4.3 k-Occurrence Chord Diagrams . . . 154

5 Future Directions **159**

5.1 Open Questions Inspired by Grid Walking . . . 159

5.2 Variations of Tutte-like Polynomials for Signed Graphs . . . 164

5.3 The Tutte Polynomial and Hypergraphs . . . 169

5.4 A Weighted Graph Interlace-like Polynomial . . . 173

5.5 Conclusion . . . 176

References **177**

List of Figures

2.1 Graph with incidences 11
2.2 An simple graph. 12
2.3 Deletion of e_1 13
2.4 Contraction of e_1 14
2.5 A loop (e_2), isthmus (e_1), and another edge (e_3). 14
2.6 The first step in creating a deletion contraction decision tree 15
2.7 The second step in creating a deletion contraction decision tree 15
2.8 The third step in creating a deletion contraction decision tree 16
2.9 A full deletion contraction decision tree 17
2.10 A set of eight spanning trees. 18
2.11 Different orientations of signed edges correspond to positively and negatively signed edges in a signed graph. 20
2.12 A signed Graph can be represented as a bidirected graph in more than one way. 21
2.13 A signed graph and two ways to switch e_1 22
2.14 A balanced signed graph 23
2.15 An unbalanced signed graph 23
2.16 Frustrated edge with respect to a spanning tree 25
2.17 A hypergraph 27
2.18 A graph and its bipartite representation 28
2.19 A hypergraph and its bipartite representation 28

2.20 Deletion and contraction from the bipartite representation of a graph 30
2.21 Deletion and contraction from the bipartite representation of a hypergraph . 32
2.22 Tutte polynomial calculation using rank and nullity 35
2.23 e_5 is internally active 36
2.24 e_4 is externally inactive 37
2.25 Tutte polynomial calculation via activities 38
2.26 A deletion contraction decision tree for the Tutte polynomial 40
2.27 The four ways to resolve a vertex in a 4-regular graphs 44
2.28 Martin polynomial calculation 45
2.29 Illustration of how a cut vertex affect the transition system 45
2.30 A decomplete graph and the associated complete graph obtained by adding a point at infinity. 47
2.31 A Feynman period symmetry for a 3 vertex separator 48
2.32 A Feynman period symmetry for a 4 vertex separator 48
2.33 A Feynman period symmetry for a 3 vertex separator with one side being planar 49
2.34 A Feynman period symmetry for a 4-edge cut 49
2.35 The Martin invariant 4-cut symmetry 50
2.36 The local complementation operations on v and $e = uv$ 53
2.37 The interlace polynomial of K_3 computed recursively 53
2.38 A 4-regular graph, Eulerian cycle, chord diagram and interlace graph 54
2.39 Forbidden interlace graph vertex minors 55
2.40 Interlace graph edge-local complementation obstructions 55

3.1 A calculation of the extended Tutte polynomial 59
3.2 Spanning trees associated to monomials of the extended Tutte polynomial . 63
3.3 Computation of activities for the extended Tutte polynomial 65
3.4 The computation of $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E})$ 68

3.5 A graph with three edge labelings and its spanning trees 70
3.6 A set of J grid walks for $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E})$ 74
3.7 J grid walks for the given edge order and specified trees. 76
3.8 Two spanning trees and their J grid walks under two edge orderings 76
3.9 A graph with two J-inequivalent edge labelings 77
3.10 A J grid walk that is the same for two J-inequivalent edge orderings 77
3.11 A Tutte grid 79
3.12 A Tutte grid with the walk for $dxccy$ shown 81
3.13 A Tutte grid with marked nodes 81
3.14 Two graphs that do not have a x^0y^ϕ corner to $x^{r(G)}y^0$ corner walk on marked nodes 85
3.15 A set of terminal minors whose marked nodes form a x^ry^ℓ to x^0y^ϕ walk . . . 86
3.16 The affect of a twist permutation on some marked nodes of a graph. 88
3.17 Spanning trees corresponding the Tutte grid walks under a twist permutation 89
3.18 A deletion contraction decision tree for a lattice graph and its Tutte grid walks 91
3.19 A deletion contraction decision tree for K_4 and its Tutte grid walks 92
3.20 A deletion contract decision tree for C_4 and its Tutte grid walks 93
3.21 The loading of two spanning trees 94
3.22 The loading of three monomials of a signed graph 96
3.23 The first half of the activity calculations on the loading of half the spanning trees of a graph see figure 3.24 for the rest. 99
3.24 The activity calculations on the loading of the second half of the spanning trees of a graph 100
3.25 The extended Tutte polynomial for signed graphs computed via deletion and contraction 101
3.26 The signed graph extended Tutte polynomial of a balanced graph 103
3.27 The extended Tutte polynomial for signed graphs of a graph with one negative edge 105

3.28 The extended Tutte polynomial for signed graphs of a graph with two negative edges . . . 106
3.29 The extended Tutte polynomial for signed graphs of a graph with all negative edges . . . 107
3.30 A graph G and a double covering graph H . . . 108
3.31 A signed graph Σ and its double cover H . . . 108
3.32 Tutte grid walks for a balanced signed graph in figure 3.26 . . . 110
3.33 Tutte grid walks for an unbalanced signed graph in figure 3.29 . . . 110
3.34 Tutte grid walks for switching equivalent signed graphs in figures 3.27 and 3.28 . . . 111
3.35 A graph and its bipartite representation . . . 112
3.36 Deletion and contraction of i_2 in Γ . . . 113
3.37 The Tutte grid for Γ and the overlay of the Tutte grid of G . . . 117
3.38 Tutte grid walks on G overlayed with Tutte grid walks on Γ . . . 119

4.1 The ways to resolve a vertex with respect to a chosen Eulerian circuit . . . 121
4.2 A word of a chord diagram . . . 122
4.3 How a vertex resolution G'_v impacts the word of a chord diagram . . . 123
4.4 How a vertex resolution G''_v impacts the word of a chord diagram . . . 123
4.5 How a vertex resolution G'''_v impacts the word of a chord diagram . . . 124
4.6 The correlation between vertex transitions, chord diagrams, and interlace operations . . . 125
4.7 A choice of Eulerian circuit producing a disconnected $I(C)$ . . . 126
4.8 The 4-regular graph with Martin invariant one . . . 130
4.9 Computation of $Q(P_1; y)$ . . . 131
4.10 Computation of $Q(P_2; y)$ . . . 132
4.11 Computation of $Q(P_3; y)$ . . . 132
4.12 Computation of $Q(K_3; y)$ . . . 133
4.13 Computation of $Q(K_4; y)$ . . . 133

4.14 Local complementation with a leaf 135
4.15 Local complementation for a locally complete vertex pair 137
4.16 One step of the interlace polynomial of a rooted tree 138
4.17 Local edge complementation in $K_{m+1,n}$ 139
4.18 Local vertex complementation in $K_{m,n}$ 140
4.19 Computation of $I'(K_{3,3})$ 142
4.20 Computation of $I'(C_5)$ 144
4.21 An interlace invariant symmetry based on the Martin invariant product . . . 145
4.22 An interlace invariant symmetry based on the Martin invariant 4-cut 148
4.23 G_1 and G_2 are stars 149
4.24 G_1 has a leaf 150
4.25 G_1 has an apex u besides v_1 151
4.26 G_1 has no apex besides v_1 152
4.27 G_1 has two adjacent non-neighbors of v_1 153
4.28 A 3- and a 4-occurrence chord diagram 155

5.1 Two non-isomorphic terminal minors corresponding to the same marked node 161
5.2 A spanning tree poset for an edge ordering 162
5.3 A spanning tree poset for another edge ordering 163
5.4 An alternative idea for computing a graph Tutte polynomial via activities . 166
5.5 An alternate deletion contraction signed graph Tutte polynomial 168
5.6 Deletion and contraction via the bipartite representation following edge order 170
5.7 Hypergraphs from the bipartite instance in figure 5.6 171
5.8 Deletion and contraction via the bipartite representation follow vertex order 172
5.9 Two vertices with a maximal number of chord crossings in k-occurrence diagrams for $k \in \{2, 3, 4, 5\}$ 174
5.10 A weighted interlace graph from a 3-occurrence chord diagram 174
5.11 Regions inside of the chord diagram that show the shape of the interlace graph 176

List of Tables

2.1 Evaluations of the Tutte Polynomial at points 41
2.2 Coefficients in the Tutte polynomial with specific meanings 42
2.3 The interlace polynomial calculated via matrices of a graph 52

4.1 Interlace forbidden vertex minors and their interlace polynomials and invariants 127
4.2 Interlace graph edge-local complementation obstructions with their interlace polynomials and invariants 128
4.3 A flip, twist, and break on a 3-occurrence chord diagram 158

Chapter 1

Introduction

Mathematics might be one of the oldest fields of scientific study, but graph theory is estimated to originate around 1736 when Euler solved the "Seven Bridges of Königsberg" problem [5]. Graph polynomials trace their origins to more recent history. In 1847, the Kirchhoff polynomial was first introduced [77]. This multivariate polynomial is a sum over spanning trees where each monomial is a product of the edges in the tree. Kirchhoff created this polynomial as part of his work with electrical circuits. Another well-known graph polynomial is the characteristic polynomial of a graph. It has its origins in the work of Sylvester in 1857 [110], but its relation to graph eigenvalues wasn't established for another hundred years when Collatz and Sinogwitz published their work [125]. Graph polynomials truly took off in the 1900's, with the chromatic polynomial's origins appearing in 1912 [14] and becoming formalized in 1946 [15]. Much of what is considered to be the true foundations of the theory of graph polynomials first appears in several papers from 1947, 1949, 1954, and 1967 by William Tutte [118, 119, 120, 121] and three papers by Hassler Whitney from 1932 and 1933 [127, 128, 130]. The 1947 paper of Tutte contains preliminary ideas about counting spanning trees and the Tutte polynomial construction. This was further expanded on in his 1949 doctoral thesis, particularly in relation to the Whitney rank generating polynomial. The 1954 paper is where the relation between the Tutte polynomial and flows is established and the notation we use today is fully developed. Similarly to how much of Tutte's work on his polynomial was achieved while he was a doctoral student, Whitney did much of his work as a doctoral student under Birkhoff. Whitney's work was a result of trying to generalize the chromatic polynomial of Birkhoff and Lewis. While Tutte was not a student of Birkhoff or Lewis, it is clear from his work [122, 123] that he too was inspired as a graduate student to attempt to generalize the chromatic polynomial results. While both Whitney and Tutte were attempting to generalize the same polynomial,

they used differing approaches. Whitney's approach utilized subgraphs chosen based on edge sets to build the rank generating function, while Tutte utilized spanning trees as his foundation and considered deletion-contraction relations on the graph.

The Tutte polynomial is one of the most widely studied graph polynomials, in part because it is the universal polynomial for any invariant that satisfies a specific deletion contraction relationship and two multiplicative properties. There are many applications of the Tutte polynomial such as in network reliability, chip-firing, knot theory, and quantum field theory, including work in [13, 29, 44, 82, 90, 104]. These connections provide a wide range of branches of mathematics in which the Tutte polynomial can be utilized. We will discuss here special evaluations of the Tutte polynomial and a brief history of a variety of polynomials which are specializations of or related to the Tutte polynomial and their applications and relevant results. We more deeply explore two additional polynomials, the Martin polynomial and interlace polynomial, toward the end of this section due to their relevance in chapter 4.

The Tutte polynomial of a graph G, denoted by $T_G(x, y)$, can provide a variety of pieces of information about the graph by evaluating the polynomial at specific points and taking specific terms. This polynomial can be calculated in three primary ways which will be discussed in more detail in chapter 2. It can be computed by taking subsets of the edges and considering the connectivity and nullity of the subgraph induced by these edges. It can also be calculated by a deletion contraction relation where for any non-isthmus, link edge the polynomial is the sum of the polynomials obtained by deleting or contracting that edge. Finally, given all maximal spanning forests of a graph and an edge ordering, the polynomial can also be obtained by counting the number of active internal and external edges for each maximal spanning forest. It is particularly interesting to note that the order of edges for activities or the order in which edges are deleted and contracted does not affect the final polynomial. Summaries of the evaluations of the Tutte polynomial and definitions can be found in [47]. Evaluations of the polynomial include counts of various spanning sub-objects, including trees and subgraphs. Other evaluations can yield the number of acyclic orientations, cyclic orientations, and tetromino tilings. For any graph we can also derive some meaning from specific terms. For example, the term with the highest power of x has coefficient y raised to the number of loops in the graph, and x is raised to the rank. In converse, the highest power of y is raised to the nullity of the graph and has coefficient x to the number of bridges. Another interesting property is that if a graph is 2-connected, then its Tutte polynomial is irreducible in $\mathbb{C}[x, y]$. Finally, when evaluating the Tutte polynomial along certain curves, we obtain the polynomials we will discuss next.

While the idea of the chromatic polynomial was first introduced in 1946 by Birkhoff and Lewis, their paper presents the chromatic polynomial in a limited context focusing

on planar graphs, as map coloring is the primary objective. The chromatic polynomial we discuss here is the generalized form that applies to both planar and non-planar graphs [121]. This polynomial is a specialization of the Tutte polynomial, denoted by $\chi(G;\lambda) = (-1)^{r(G)}\lambda^{k(G)}T_G(1-\lambda,0)$. When evaluated at $k \geq 0$, the polynomial counts the number of proper k-colorings of the graph. Birkhoff established his chromatic polynomial in hopes that proving the $\lambda = 4$ evaluation was non-zero would provide a proof of the four color theorem. While this technique did not yield a proof of the four color theorem, the chromatic polynomial has been used in a variety of other ways including searching for classes of graphs that are distinguished by their chromatic polynomial [11, 50, 66, 69, 79, 103] and finding conditions for when a polynomial is the chromatic polynomial of some graph [81]. Practical applications of the chromatic polynomial relate to the allocation of resources to a system to avoid conflicts. An example of this would be course scheduling where each section of a course is represented by a vertex and an edge represents courses that need to be scheduled at different times. Evaluated at k, the chromatic polynomial of such a graph would provide the number of valid schedules given k time slots.

Given a planar graph, its chromatic polynomial is the flow polynomial of the planar graph's dual. While non-planar graphs do not have planar duals, the concept of flow is independent of planarity. The flow polynomial was introduced by Tutte in 1954 [120] and can be evaluated as $F(G;\lambda) = (-1)^{n(G)}T_G(0,1-\lambda)$. When evaluated at $k \geq 0$, the polynomial counts the number of nowhere-zero $\mathbb{Z}_k$-flows on the graph. This appears to be very similar to the chromatic polynomial evaluation and, in fact, for a planar graph G and its planar dual G^*, $T_G(x,y) = T_{G^*}(y,x)$ [120]. There are many uses of flows in graphs for modeling networks and solving problems for distribution of resources, optimizing routes, or managing capacity of the network [58, 84]. The flow polynomial provides useful ways to store and extract information for such problems. The flow polynomial is used in physics in models like the Potts model to study phase transitions like that in [28] and relates to optimization problems like max-flow min-cut as shown in [109].

The characteristic polynomial of a graph comes in several different forms obtained by taking a determinant or permanent of λI minus the adjacency or Laplacian matrix of a graph. Most often the determinant is used and the roots of this polynomial are eigenvalues of the graph. When the Laplacian is used these eigenvalues are useful for studying connectivity and maximal spanning forests. The adjacency matrix and its eigenvalues are often used in relation to strongly regular graphs. Powers of the adjacency matrix can tell us about walks of a graph, and the characteristic polynomial can be used to derive a generating function to count such walks [80]. While two graphs with the same characteristic polynomial are not necessarily isomorphic, two graphs with different characteristic polynomials are certainly not isomorphic. The spectral decomposition of a graph has a

wide range of applications many of which are studied in algebraic graph theory, see [56] as a standard reference for all these results. The characteristic polynomial can provide a convenient way store the eigenvalues of a graph, and in most cases it is easy for a computer to compute or approximate them. A more computationally expensive polynomial, the total minor polynomial, is based upon the characteristic polynomial can be found in [61]. This polynomial provides a complete characterization of all minors of the graph's adjacency or Laplacian matrix and a characterization of sub-objects called contributors. Contributors and the total minor polynomial were then used to extend Kirchhoff's Laws to signed graphs in [102].

The Kirchhoff polynomial, which satisfies a multivariate deletion contraction relation similar to that of the Tutte polynomial, first appeared in 1847 as a polynomial summing over the edge sets of the maximal spanning forests of a graph [77]. The Kirchhoff polynomial has exactly one term for each maximal spanning forest and so, when evaluated at one for every indeterminate, it counts the number of maximal spanning forests. The Kirchhoff polynomial and its relation to maximal spanning forests will be used in chapter 3. The polynomial can also be computed by the determinant of any first minor of the Laplacian of a graph after replacing all adjacencies with indeterminates. The matrix tree theorem, which counts the number of maximal spanning forests, is also obtained from the determinant of any first minor of the Laplacian without the introduction of any indeterminates [77]. This is clearly equivalent to evaluating the Kirchhoff polynomial at ones. The deletion contraction relation satisfied by the Kirchhoff is the same general idea on the deletion and contraction of an edge as that in the Tutte polynomial, but the Kirchhoff polynomial has one indeterminate per edge and thus cannot be an evaluation of the Tutte, which has two indeterminates. There are multivariate versions of the Tutte polynomial, including the Potts extension and the multivariate chromatic polynomial, which have a variable for each edge, and thus more clearly relate to the Kirchhoff polynomial [47]. An indeterminant for each edge was also separately introduced by [93, 112] and [139] who were all inspired separately by Kaufman to look multivariate Tutte polynomials. The extended Tutte polynomial discussed in chapter 3 actually appears in [139] as a matroid version. As both the Tutte polynomial and Kirchhoff polynomial relate to maximal spanning forests counts, the Kirchhoff polynomial can be considered an evaluation of the multivariate Tutte polynomial, where an indeterminate for each edge is introduced. This close relationship is in part because $T_G(1,1)$ counts the number of maximal spanning forests which is also counted by the Kirchhoff polynomial. While the Kirchhoff polynomial does not specialize to provide any other result provided by the Tutte polynomial, it is incredibly useful in enumerating spanning trees and relates to Kirchhoff's laws for conservation of energy in circuits and various extensions of these results [34, 77, 102]. Additionally, the dual of

the Kirchhoff polynomial shows up in a definition of the Feynman period [126], and its properties are important for the denominator reduction approach to computing Feynman integrals [27].

Beyond the polynomials that are specializations of the Tutte polynomial, there are a variety of polynomials that are related to the Tutte polynomial via skein relations. Skein relations arise from knot theory where the relation considers a crossing of links and all the ways in which the crossing might be configured [35]. We can use skein relations on a polynomial by considering the edges as links and the vertices as crossings where we have yet to determine what the configuration is. Polynomials can be defined recursively via skein relations by taking a vertex and all possible skein resolutions and taking the sum of the polynomial over all these possible resolutions.

The Penrose polynomial, which applies to planar graphs, was first established in 1971 when Roger Penrose wanted to try understand edge coloring in the context of a conjecture by Tait that would have implied the four color theorem [98, 111]. Tutte had already proved this conjecture of Tait to be false [117], but the edge coloring of planar graphs still interested Penrose. The definition of the Penrose polynomial can be extended to graphs embedded topologically which enables non-planar graphs to be considered [46]. This formula almost satisfies a deletion contraction relation but has subtraction instead of addition and thus doesn't satisfy the same relation as the Tutte polynomial. However, for planar, isthmus free, cubic graphs, the Penrose polynomial is $T_G(2, \lambda)$. Later work also has extended this polynomial to matroids and then to delta matroids [3, 26].

For an even regular graph, the transition of a vertex, defined formally in chapter 2, is a notion inspired by knot theory where the vertex is treated like a crossing in a knot and a transition is a choice of pairing the edges up at the vertex to create new edges and removing the vertex. This is similar to the notion of resolving a knot. In 1990, Jaeger generalized these skein polynomials to the transition polynomial of a 4-regular graph [70]. Through evaluations of this polynomial along lines, a variety of other skein polynomials are obtained. This includes the Penrose polynomial, Kauffman brackets (a polynomial for knots), the general circuit partition polynomial, and the Martin polynomial which we will discuss later. While the transition polynomial was originally defined for 4-regular graphs, it can be defined more generally for Eulerian graphs and thus generalizes more versions of the other skein polynomials [48]. Like the Penrose polynomial, the transition polynomial has a generalization to delta matroids [25]. Transition polynomials are useful for a variety of concepts including random walks, phase transitions, Markov chains, and statistical models like the Potts model; some examples of which can be seen in [6, 10, 12, 49, 107].

The Jones polynomial is used primarily in knot theory as it is a link invariant capable

of distinguishing many knots and link. First established in 1984 [72], this polynomial is closely related to the Kauffman bracket polynomial. Indeed, the definition of the Jones polynomial that is most commonly used in the literature, being based on the skein relationship, was established by Kauffman [73]. The Kauffman bracket polynomial, another knot polynomial using skein relations, was defined in 1985. Both polynomials relate to the Tutte polynomial through agreeing on various special cases, for example the Jones polynomial on an alternating link corresponds to the Tutte polynomial of a related planar checkerboard graph [30, 31, 37]. For an alternating knot and its associated graph G, the Jones polynomial of the knot is $(t^{\frac{1}{2}})^{-|E|}T_G(t, \frac{1}{t})$, and for planar graphs the Kauffman bracket polynomial is$(-A^3)^{-|E|}T_G(-A^2, -A^{-2})$. The Jones polynomial is in fact a generalization of the Kauffman bracket polynomial despite coming first, and both are useful in statistical models such as the Potts model and as quantum link invariants as seen in [2, 55, 131].

One field of current study in graph polynomials is the extension of known polynomials to signed graphs and hypergraphs. There are currently several different generalizations of the Tutte polynomial to signed graphs which reduce to the Tutte polynomial on balanced graphs. We first discuss the 1989 version by Kauffman [74] which is based in the notion of edge-2-colored graphs. This polynomial almost obeys the deletion contraction relation, though the expression depends on the sign of the edge. Due to Kauffman's primary interest in knot theory, the polynomial relates nicely with his bracket polynomial and the Jones polynomial when calculated on the medial graph of a link projection. This polynomial defined by Kauffman can be fully calculated recursively over four variables before a substitution is made to a two variable polynomial. While this signed graph Tutte polynomial has nice relations to the Jones polynomial and the Ising model, it lacks any known combinatorial meaning related to the graph as it does not have known evaluations providing nice counts of structural properties. The motivation for this polynomial was derived from knot theory, and since any knot can be represented as a planar graph, this definition lacks clear meaning for non-planar graphs. In 1995, Zaslavsky defined a 2-variable polynomial in the greater generality of biased graphs and gain graphs [134]. This polynomial has a deletion contraction relation and relates to the lift matroid of a graph. His paper presents the first Tutte-like polynomial explicitly addressing balance and sign. Zaslavsky defines here the notion of strong and weak Tutte functions and classifies Tutte functions on matroids into seven types such that every Tutte polynomial which extends to a matroid falls into one of these classifications. What is referred to here at the extended Tutte Polynomial, Zaslavsky calls the parameterized Tutte polynomial and among the presented results are a deletion contraction definition and activity definition. Another signed graph Tutte polynomial was introduced in 2019 [57] by Goodall et al. This is a three variable polynomial which uses the new indeterminate z to track negative edges. This polynomial is invariant

under switching, which the Kauffman polynomial is not, and has a more combinatorial interpretation as it tracks signs separately from x and y which allows it to specialize to known combinatorial properties such as number of proper colorings of a signed graph. Additionally, this polynomial can be evaluated to obtain known extensions of the chromatic and flow polynomials to signed graphs, though these specializations are not clear from a structural or combinatorial perspective as the choice for variable substitution for z is not immediately obvious. This signed graph Tutte polynomial does not have a full deletion-contraction recurrence as the recurrence is defined on positive edges, nor does it have any known computation via activities. Perhaps the most interesting feature of the Goodall et al. signed graph Tutte polynomial is that the variable used to track negative edges has no clear structural interpretation in the final polynomial. Changing the sign of a single edge drastically changes the polynomial and thus not much structural information about the graph is retained including, in particular, basis information. Both Kauffman's and Goodall et al.ś signed graph Tutte polynomials provide meaningful generalizations to signed graphs with different primary concerns. However, neither fully behaves in the same way nor retains the information about the graph that the Tutte polynomial does on general graphs. In this thesis, we make our own attempt in chapter 3 to construct a signed graph Tutte polynomial where we prioritize retaining bases and activity information.

The Martin polynomial, discussed further in chapter 2, was first introduced in 1977 as part of Martin's dissertation where it was defined to count circuit decompositions of 4-regular graphs [91]. In [85, 86, 87], Las Vergnas showed that the Martin polynomial can be defined for Eulerian graphs in general. The Martin polynomial has been shown in [17] and [45] to be an evaluation of the circuit partition polynomial, which is a polynomial counting the number of partitions of a graph into Eulerian circuits. While the Martin polynomial does not satisfy any contraction-deletion relationship, it is multiplicative over disjoint unions and for any connected plane graph G and its directed medial graph $\mathrm{med}(G)$, $T(G;x,x) = m(\mathrm{med}(G);x)$. This is due to the 4-regularity of $\mathrm{med}(G)$ and one-to-one correspondence between the skein relations in $\mathrm{med}(G)$ and deletion-contraction of an edge in G. Another interesting way in which the Martin polynomial relates to another polynomial is that the diagonal coefficient of the Kirchhoff polynomial for a $2k$-regular graph G minus a vertex is $k!m(G)$ [96].

There is a variety of work that has been done with the Martin polynomial. It has been investigated in terms of evaluations and properties in a number of papers including [17, 45, 86] and in [115] where the polynomial is considered as a chromatic polynomial for transition matroids. The Martin polynomial has both a definition for $2k$-regular graphs and 2-in 2-out digraphs. We primarily focus on the $2k$-regular definition with special attention to the 4-regular case. There has also been work done showing relationships

between evaluations of the Tutte polynomial and the Martin polynomial like those shown in [87] and [71]. As shown in [96], one of the ways the Martin is useful is as it relates to the period defined by a Feynman integral. The Martin polynomial has in fact appeared in theoretical physics in another manner as seen in [78], but we are primarily interested in its relation to the Feynman period through the Martin invariant. The Martin invariant, as defined in [96], is an evaluation of the derivative of the Martin polynomial of a $2k$-regular graph. We are mostly concerned with the 4-regular case introduced in [24], but we will discuss Eulerian graphs in general when possible. The symmetries of the Martin invariant are the same symmetries as those obeyed by the Feynman period. The Feynman period of a graph is obtained via an evaluation of a Feynman integral, the denominator of which contains the Kirchhoff polynomial of the graph. The Feynman period is useful in quantum field theory but challenging to compute, so we are interested in finding other graph invariants that have the same symmetries.

The interlace polynomial, first introduced in [7], was inspired by problems related to DNA sequencing of hybridization. There it was observed that the interlace polynomial is related to the Martin polynomial in the 2-in 2-out case, and a majority of the literature on the interlace polynomial focuses on this relationship or on specific evaluations. Additionally, there is a known relationship between the Tutte polynomial of a binary matroid and the interlace polynomial [4, 23]. Beyond the evaluations and the Martin polynomial connection, there is also a generalization of the interlace polynomial to a multivariate version in [36]. The version of the interlace polynomial we are interested in first appeared in [4] where Goodall et al. made a generalized polynomial that relates to the Martin polynomial in the 4-regular case. Unless otherwise specified, in chapter 2 and chapter 4 when referring to the interlace polynomial, this will be the polynomial we are speaking of. This relationship has been further investigated in [19] and [114]. For any 4-regular graph, the Martin polynomial of that graph is equal to the interlace polynomial of the interlace graph (sometimes called a circle graph) of a chord diagram obtained from an Eulerian circuit on the 4-regular graph. For our purposes, we are primarily interested in the Martin invariant and what properties and symmetries we can extend to the interlace polynomial. We will do this by creating an interlace invariant in chapter 4. In [92], a survey of work done on the interlace polynomial shows that the majority of results related to the interlace polynomial are actually results relating to interlace graphs. For the purposes of this thesis, the interlace graph results of relevance involve the local complementation operation used to recursively define the interlace polynomial [113] and characterizations of the forbidden vertex minors and edge-local complement minor obstructions in [21, 88] and [54], respectively. While we focus on the interlace polynomial in this thesis, interlace graphs have been examined in a variety of contexts including isotropic systems and via matroids [23, 18, 19, 22, 42, 43, 51, 53, 75, 105],

graph coloring and χ-boundedness [1, 32, 33, 40, 38, 39, 41, 42, 52, 94], and general graph properties [52, 67].

Chapter 2 will cover the necessary background definitions and theorem. In chapter 3, we will cover the Tutte polynomial where we discuss the extension of the Tutte polynomial to retain deletion-contraction information via the introduction of two additional indeterminates. By examining how edges orderings affect this new polynomial, we consider each monomial of the polynomial as a grid walk in section 3.3. We then combine these extended Tutte polynomials with the notion of edge frustration, inspired by [63], to construct a signed graph Tutte polynomial in section 3.4. This section will conclude with an examination of grid walking for the signed graph Tutte polynomial in section 3.4.3. Chapter 3 concludes with a discussion of the extended Tutte polynomial of bipartite representations of graphs and their grid walks in section 3.5. In chapter 4, we look at various results on the interlace invariant for specific classes of graphs in section 4.1. In section 4.2, we will discuss the Martin invariant properties that have been extended to our construction of the interlace invariant. In section 4.3, we examine how the $2k$-regular Martin polynomial determines a set of operations on k-occurrence chord diagrams. Finally, in chapter 5, this thesis will discuss future directions, including further extension of the Tutte polynomial to oriented hypergraphs and the progress made toward extending the interlace polynomial to weighted graphs, in an attempt to create a connection between an interlace like polynomial on graphs related to the Martin polynomial for $2k$-regular graphs with $k \geq 3$.

Chapter 2

Background

We will begin with a discussion of general graph theory notions used throughout this thesis in section 2.1. In this section, we will cover both standard graphs, bidirected and signed graphs, and bipartite representations of graphs and oriented hypergraphs. In section 2.2, we cover activities in a graph and various definitions of the Tutte polynomial. We present a table of specific evaluations of the Tutte polynomial, many of which were mentioned in Chapter 1. In section 2.3, we cover the necessary definitions and the symmetries of the Martin polynomial and invariant that form the motivation for the work with the interlace polynomial, the background of which is presented in section 2.4. We assume that our graphs are simple graphs in the case of the interlace polynomial, while we allow both loops and multi-edges in the Martin polynomial and Tutte polynomial cases. As these polynomials are multiplicative over connected components, we will assume our graphs to be connected when convenient for the simplicity of presentation of the results here.

2.1 Graphs

The definitions in this section provide the basic structure for a graph that is used in pursuit of a Tutte polynomial for hypergraphs. This definition more concretely defines the notion of an incidence and provides an alternate, but equivalent definition of a graph that extends cleanly to hypergraphs. These definitions are based upon the work of [99] and [101]. There are three reasons for the decision to present graphs in this manner. The first reason is the hope that this Tutte polynomial for signed graphs can be extended to hypergraphs using the results in [61] which rely on the incidence structure, and the second being the convenience of representing signed graphs as bidirected graphs where the incidences are

assigned the sign. This further allows for a clean extension of signed graphs to oriented hypergraphs. Finally, quantum field theory uses half edges and we can work with half edges easily using incidences.

A *graph* $G = (V, E, I, \varsigma, \omega)$ is a set of vertices V, a set of edges E, and a set of incidences I equipped with two functions $\varsigma : I(G) \to V(G)$ and $\omega : I(G) \to E(G)$, where $|\omega^{-1}(e)| = 2$. From the definition of a graph, we consider several properties. The *degree of a vertex* v is $|\varsigma^{-1}(v)|$. A vertex and edge are *incident* via incidence i if $i \in \varsigma^{-1}(v) \cap \omega^{-1}(e)$. We also refer to v as an *end* of e. We say an incidence i is *associated* to a vertex v or edge e if $\varsigma(i) = v$ or $\omega(i) = e$. In figure 2.1 we have a graph with all incidences labeled. Each edge has two incidences mapped to it, and each vertex has one incidence for each incident edge which contributes to the degree. In general we omit the incidence labels in figures for simplicity; however, all edges and vertices are assumed to be connected by incidences.

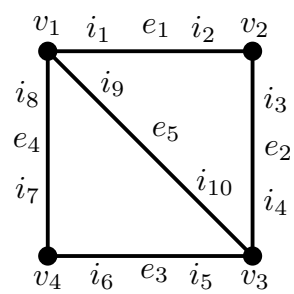


Figure 2.1: Graph with incidences

Path properties of graphs are incredibly important for the analysis of a graph. We define various path properties as follows: A *directed path of length* $n/2$ is a non-repeating sequence

$$\overrightarrow{P}_{n/2} = (a_0, i_1, a_1, i_2, a_2, i_3, a_3, ..., a_{n-1}, i_n, a_n)$$

of vertices, edges and incidences, where $\{a_\ell\}$ is an alternating sequence of vertices and edges, where $a_0, a_n \in V(G)$ and i_h is an incidence between a_{h-1} and a_h. Note this forces n to be even. Let $\overrightarrow{P}_1 = (t, i, e, j, h)$ denote a path of length 1 with a distinguished tail vertex t and head vertex h. The incidences i and j are the *tail-incidence* and *head-incidence*, respectively. A *directed weak walk of* G is the image of an incidence-preserving map of a directed path into G. In general we omit incidences when discussing graphs as any notion of a path or walk can be uniquely determined by the sequence of vertices and edges since in a simple graph each edge and vertex have at most one incidence between them and we generally don't need loops in walks.

Using the above information, we can define many relationships between incidences, edges, and vertices in terms of the path properties. That is to say, we can distinguish different types of paths by placing restrictions on their sequences of incidences, edges, and vertices. To define these relationships we will consider how $\overrightarrow{P}_{n/2}$ can be mapped into a graph and we say this mapping is *monic* in incidences/vertices/edges if no incidence/vertex/edge is repeated. A *path of* G is a vertex-, edge-, and incidence-monic directed path of length $n/2$. A *backstep of* G is an embedding of $\overrightarrow{P}_1$ into G that is neither incidence-monic nor vertex-monic. A *walk* is a weak walk containing no backsteps. A *loop of* G is an embedding of $\overrightarrow{P}_1$ into G that is incidence-monic but not vertex-monic. A *directed adjacency of* G is an embedding of $\overrightarrow{P}_1$ into G that is incidence-monic. A *cycle of* G is an embedding of $\overrightarrow{P}_{n/2}$ into G that is incidence-monic and vertex-monic with the exception of the initial and final vertex $a_0 = a_n$. Note, backsteps are considered to be separate from cycles as they are not incidence-monic. A *digon* is a cycle containing exactly two edges and a *circuit* in a graph G is a closed walk. Two vertices u and v are *adjacent or neighbors* if ui_1ei_2v is a path and the set of all neighbors of u is denoted by $N(u)$. The *closed neighborhood* of a vertex v, denoted by $\overline{N(v)}$, is the set of all neighbors of v and v itself. Using incidences is only strictly necessary for the definition of backsteps, which are not used in the work presented here. This allows use to use the language of traditional graph theory when talking about graphs.

Example 2.1.1. Consider the graph in Figure 2.2. The four vertices are labeled v_1, v_2, v_3 and v_4 and there are five edges e_1, e_2, e_3, e_4 and e_5. This graph is the same as that of figure 2.1 but the incidences have been omitted for simplicity.

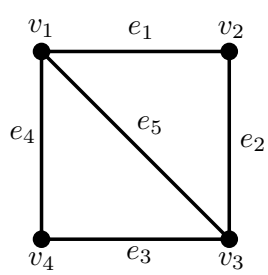


Figure 2.2: An simple graph.

Thus v_1 and v_3 have degree 3 while vertices v_2 and v_4 have degree 2. Each edge is connected to two incidences and so the size of each edge is also 2. The vertices v_1 and v_2 are adjacent since the path $\overrightarrow{P_1} = (v_1, e_1, v_2)$ connects them. The neighborhood of v_1 is $N(v_1) = \{v_2, v_3, v_4\}$. The vertices v_1, v_2, v_3 and v_4 form a cycle of G, which is

$C = v_1e_1v_2e_2v_3e_3v_4e_4v_1$. We may, for simple graphs omit edge or vertex labels in cycles and paths as they are uniquely defined by having either the edge or vertex set.

Given a graph, we can also define various sub-objects of the graph. A *subgraph* H of a graph G is a graph formed by a subset of the original graph's vertex, edge, and incidence sets with a restriction of the original mapping functions, such that for all edges $e \in H$ we still have $|w^{-1}|_H(e)| = 2$. Clearly, a graph must be a subgraph of itself. A *connected component* of G is a subgraph H in which a path exists between all pairs of vertices in H and for any vertex not in H, there is no path in G to any vertex in H. A graph G is *connected* if G is itself a connected component.

We next introduce two useful graph operations. For a graph G the *deletion* of $e \in E(G)$, denoted by $G - e$ or $G \setminus e$, is the graph obtained from G by removing e and all associated incidences. Let G be a graph and e an edge with ends v and w. The *contraction of e in G*, denoted by G/e, is the graph obtained by identifying v, w, e and associated incidences as a single new vertex v_e whose incident edges are $\varsigma^{-1}(w)$ and $\varsigma^{-1}(v)$.

Example 2.1.2. Consider the graph G in figure 2.3 and e_1. We can see that the deletion of e_1 results in a new graph (right) that has the same vertex set as G, but has e_1 and all associated incidences removed.

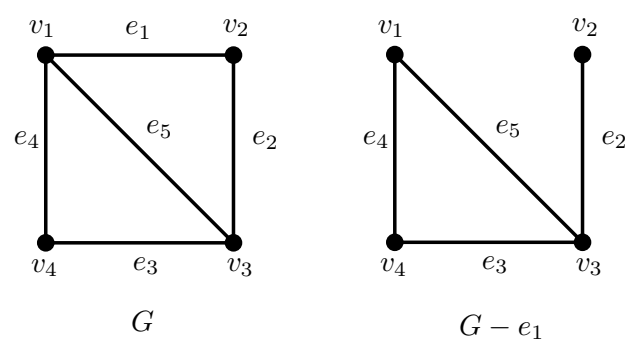


Figure 2.3: Deletion of e_1

Now instead of deleting e_1 in figure 2.4, consider the contraction of e_1. We see that we have a new vertex $v_1e_1v_2$ which is incident to all edges incident to v_1 or v_2, other than e itself.

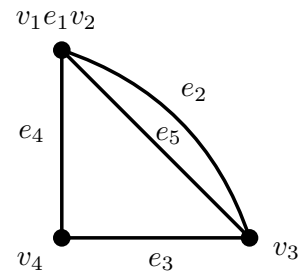


Figure 2.4: Contraction of e_1

These deletion and contraction operations will be essential for one definition of the Tutte polynomial, but first we need a bit of information about a specific type of edge in a graph. An *isthmus*, also called a bridge, is an edge which when removed will disconnect the graph. We also consider loops to be an important type of edge, and we may think of them as edges where both ends are the same vertex. An edge that is not a loop is sometimes called a *link* In figure 2.5 we see that the removal of the isthmus e_1 would disconnect the graph. We can also see that e_2 is a loop as both ends are the same vertex. Finally, e_3 is neither a loop nor an isthmus as it has two distinct ends and its deletion will not disconnect the graph.

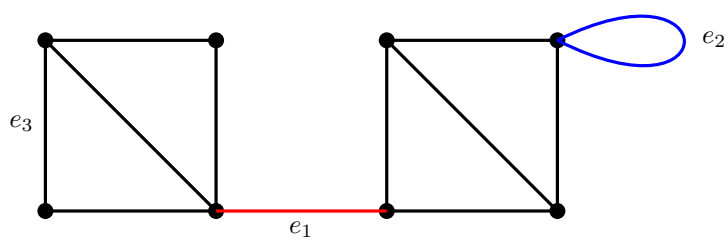


Figure 2.5: A loop (e_2), isthmus (e_1), and another edge (e_3).

A *deletion contraction decision tree* for a graph G is made by taking an edge e that is neither a loop nor an isthmus and creating two new graphs, one obtained by deleting e and one obtained by contracting e. Then for each new graph created a new edge is chosen that is neither loop nor isthmus and the process is repeated till only non-isthmi links remain. The graphs at each stage are organized by creating arrows to two new graphs, one obtained through deletion and the other through contraction, with the edge deleted or contracted indicated by arrows.

Example 2.1.3. For the graph in Figure 2.2 we can create a deletion contraction decision tree as follows. We first take our graph and choose edge e_1 and delete and contract it.

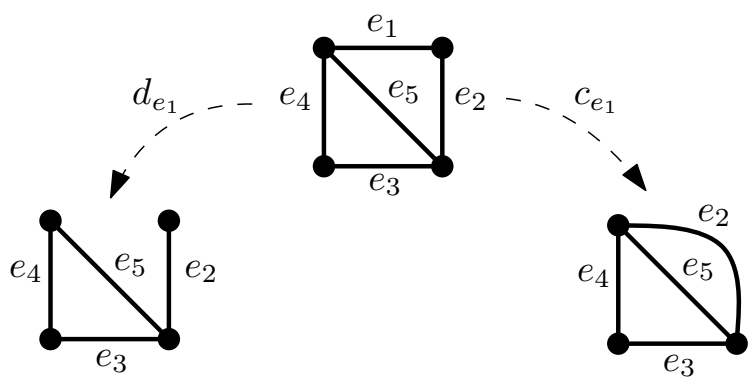


Figure 2.6: The first step in creating a deletion contraction decision tree

Now we have two new graphs that both contain non-isthmus link edges. Focusing on the graph obtained by deleting e_1 we start there to continue building the decision tree. We cannot delete or contract e_2 as it is an isthmus, so we will choose e_3 to delete and contract.

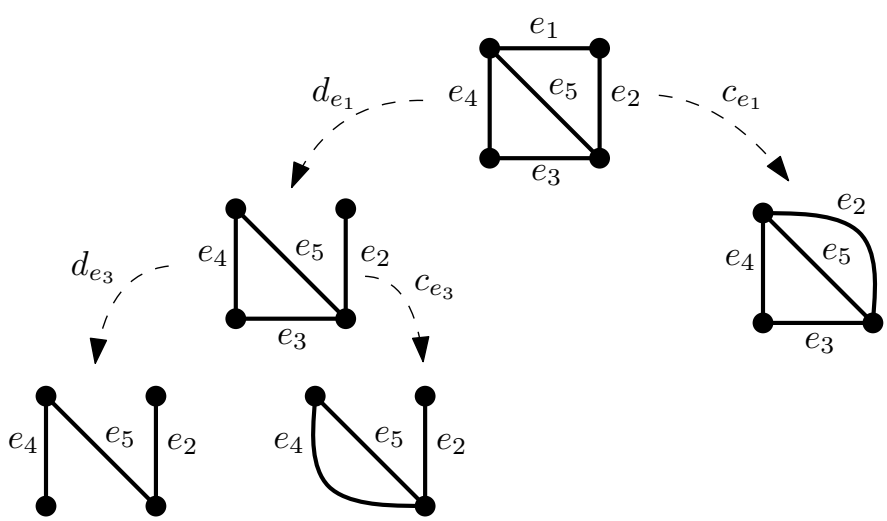


Figure 2.7: The second step in creating a deletion contraction decision tree

Now the graph obtained by deleting e_3 contains only isthmi and thus cannot have further edges deleted or contracted, but the graph obtained by contracting e_3 still has non-isthmus, link edges e_4 and e_5 so we choose e_4 to delete and contract.

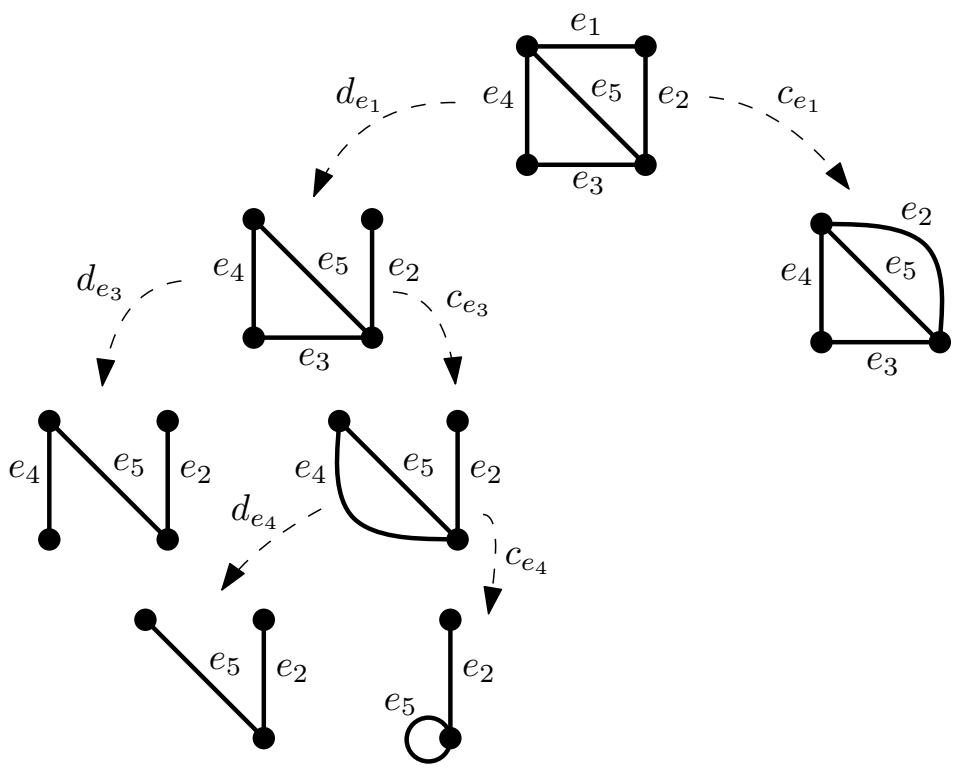


Figure 2.8: The third step in creating a deletion contraction decision tree

The two new graphs created are only loops and isthmi and so we have finished all possible operations on the part of the deletion contraction decision tree that stems from the deletion of e_1. If we continued building the tree from the graph obtained by contraction e_1 we would obtain the full deletion contraction decision tree as shown in figure 2.9.

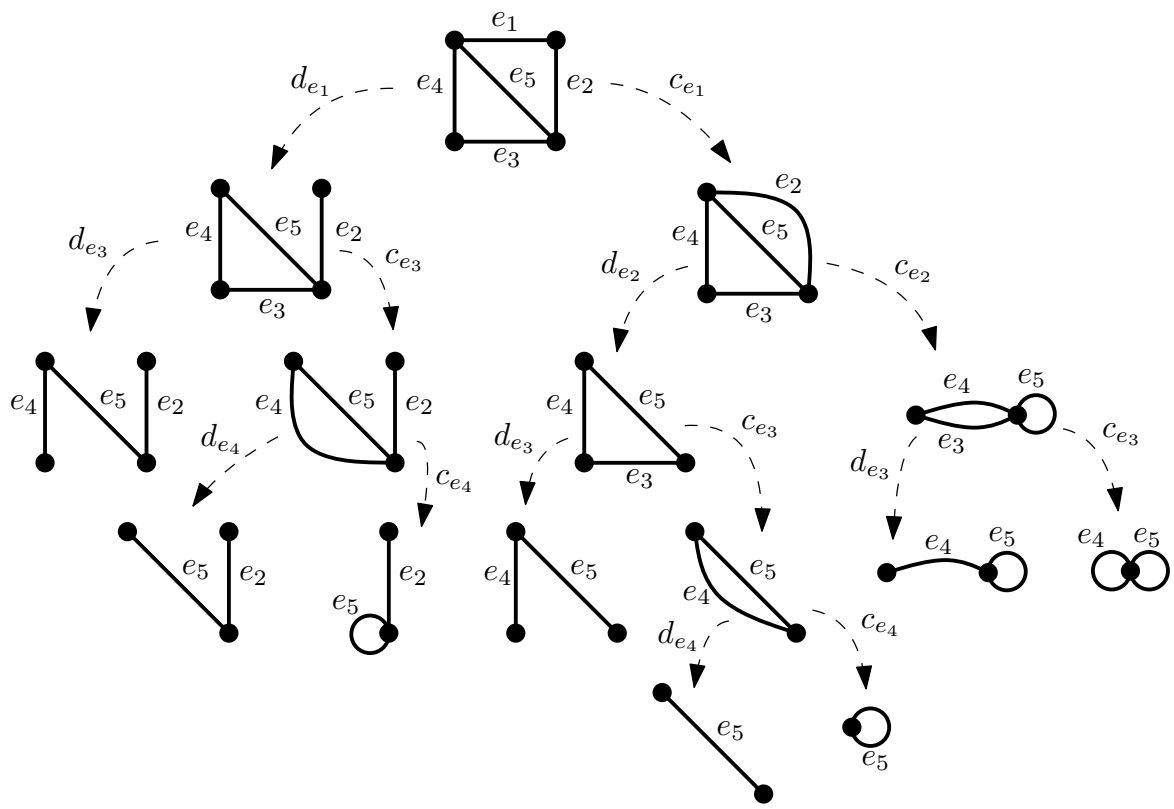


Figure 2.9: A full deletion contraction decision tree

Now that we have defined components and connectedness, we use these notions to define spanning trees and maximal spanning forests. Both notions are essential for our work with the Tutte polynomial. A *tree* is a connected acyclic graph. A subgraph H of G is *spanning* if $V(H) = V(G)$, while a *spanning tree* of G is a subgraph of G that is a tree and spanning. The *tree number of* G is the number of spanning trees of G. A *forest* is a graph where every connected component is a tree. A *spanning forest* of G is a subgraph that is a forest and spanning. Finally, a *maximal spanning forest* F of G is a spanning forest with the property that for any edge $e \in G - F$, $F + e$ contains a cycle. This is equivalent to choosing a spanning tree for each connected component of the graph. It is also worth noting that an isthmus has the property of being in every maximal spanning forest or spanning tree of a graph. For the Tutte polynomial and observations made about spanning trees of a connected graph, these observations can be extended to graphs with multiple components by simply replacing spanning trees with maximal spanning forests.

Example 2.1.4. Recall the graph G in figure 2.2. Each subgraph in Figure 2.10 is acyclic

and connected while containing all vertices of G and so the subgraphs are all spanning trees of G. These eight subgraphs are in fact all the spanning trees of G.

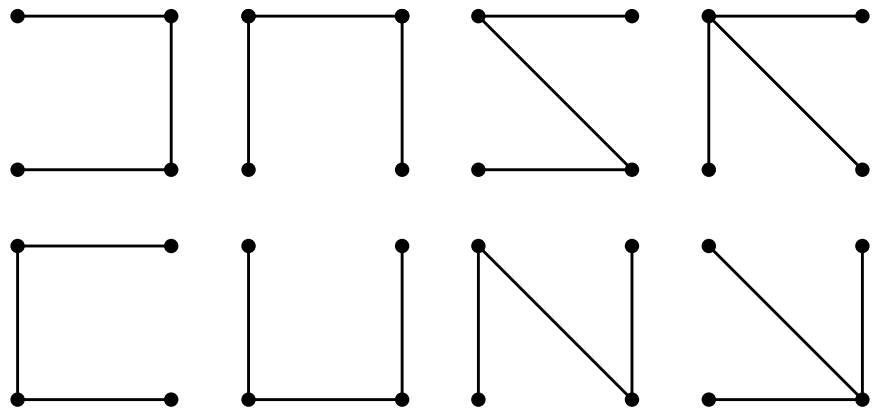


Figure 2.10: A set of eight spanning trees.

The graph G contains four vertices, as does each subgraph. Each subgraph also contains a subset of the original graph's incidences and edges. We can see that for any spanning tree T, adding an edge in $G - T$ creates a unique cycle.

It is worth noting that the deletion contraction decision tree of the graph in figure 2.2 has eight final figures that cannot have further deletions or contractions, as shown in figure 2.9 and has eight spanning trees as shown in figure 2.10. There is a direct relationship between the number of spanning trees of a graph and the number of final objects in the deletion contraction decision tree. This will be further discussed in Chapter 3.

2.1.1 Bidirected and Signed Graphs

A *bidirected graph* G is a graph equipped with an incidence orientation function $\tau : I \to \{+1, -1\}$. An incidence with orientation $+1$ is depicted as an arrow entering a vertex, and an incidence with orientation -1 is depicted as an arrow exiting a vertex. This follows the work in [138]. To give signs to adjacencies, backsteps, loops, and paths we will use the notion of a weak walk. Recall a *directed weak walk* $\overrightarrow{W}$ of length $\frac{n}{2}$ is a sequence

$$\overrightarrow{W}_{n/2} = (a_0, i_1, a_1, i_2, a_2, i_3, a_3, ..., a_{n-1}, i_n, a_n)$$

where $a_0, a_n \in V(G)$ and the a_i alternate between vertices and edges where n is even. This differs from a directed path in that it need not be incidence-, edge-, or vertex-monic. A *weak walk* is a directed weak walk in which we ignore the direction of travel, so given a directed weak walk $\overrightarrow{W}$ from v_1 to v_2 we also have the directed weak walk from v_2 to v_1 obtained by reversing $\overrightarrow{W}$. These two directed weak walks produce the same weak walk. It is clear that paths, cycles, loops, and walks are all weak walks. So the sign of a weak walk is all that is needed. We define the *sign of a weak walk* W as

$$\operatorname{sgn}(W) = (-1)^{n/2} \prod_{h=1}^{n} \tau(i_h).$$

A *signed graph* Σ is a graph G equipped an edge signing function $\sigma : E \to \{+1, -1\}$. The sign of a cycle, path, backstep, loop, or walk, is simply its sign when viewed as a weak walk. In these cases we happen to have more specific requirements about which vertices, incidences, and edges can be included. A simple way to compute the sign of a cycle is simply to count the number of negative edges and observe the cycle is positive if the number of negatives is even and negative if the number of negatives is odd. When working with signed graphs we will simply sign the edges as some choice of incidence signing will provide the edge signs we want. Given that we may determine the signs of paths, cycles, and loops without using the incidences by simply multiplying the signs of the edges we will in general simply write $\Sigma = (G, \sigma)$ for a signed graph with underlying graph G and edge signing function σ.

We briefly consider the relationship between bidirected and signed graphs. Given a bidirected graph there is exactly one signed graph that corresponds to the birdirected graph by taking the sign of the edge e to be negative one times the product of the orientations of an incidence to correspond to e. That is to say we sign the edges such that for each e with $\omega^{-1}(e) = \{i, j\}$ we have $\operatorname{sgn}(e) = -\tau(i)\tau(j)$ where $\tau : I \to \{+1, -1\}$. See figure 2.11 for an example.

Lemma 2.1.5. *Given a bidirected graph $G = (V, E, I, \varsigma, \omega, \tau)$ there exists a unique signed graph $\Sigma = (V, E, I, \varsigma, \omega, \sigma)$ such that for all edges $e \in E$ with $\omega^{-1}(e) = \{i, j\}$ then $\sigma(e) = -\tau(i)\tau(j)$.*

Proof. By construction Σ exists, so it remains to show that Σ is unique. Suppose there exists Σ_1 and Σ_2 both satisfying the lemma. Then the only difference between Σ_1 and Σ_2 must be in σ_1 and σ_2 as all other sets or functions are unchanged from G. So there exists an edge e such that $\sigma_1 \neq \sigma_2$. Let $\omega^{-1}e = \{i, j\}$. Then $-\tau(i)\tau(j) = \sigma_1 \neq \sigma_2 = -\tau(i)\tau(j)$, a contradiction. ■

Starting with a signed graph Σ there are multiple bidirected graphs that have the appropriate incidence orientations that correspond to the edge signs of Σ. To see this observe that for negative edges, $-(+1)(+1) = -(-1)(-1)$ giving two different valid incidence orientation options for each edge. Furthermore given positive edges, the incidence signs associated to distinct vertices are not fixed. So if $vieju$ is a P_1 in Σ with e positive then $\text{sgn}(i) = -1, \text{sgn}(j) = +1$ and $\text{sgn}(i) = +1, \text{sgn}(j) = -1$ result in the same edge sign, but different incidence orientations and thus different bidirected graphs. Fortunately, for the purposes of our work the bidirected graph chosen has no affect on the polynomials we are discussing. The signed graph extended Tutte polynomial we will define can be used on bidirected graphs by first translating them to signed graphs, and this direction produces a unique signed graph.

Example 2.1.6. Observe that using the definition of signing a bidirected edge e with incidences i and j, which is $-\tau(i)\tau(j)$, that the sign of an edge is positive if the arrows appear to flow through the edge and negative if the arrows appear to both enter or both exit the edge.

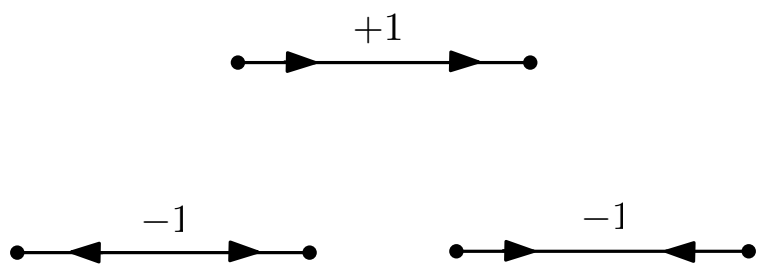


Figure 2.11: Different orientations of signed edges correspond to positively and negatively signed edges in a signed graph.

Now that we have shown the relationship between edge signings and incidence orientations, we will examine how this translates to an entire graph in the next example. For this we follow the definition of orientations of signed graphs presented in [138] and the orientation and edge sign relationships presented in figure 2.11.

Example 2.1.7. Translating the edge orientations demonstrated in Figure 2.11 to the incidences, it can be seen that bidirected graphs are orientations of signed graphs.

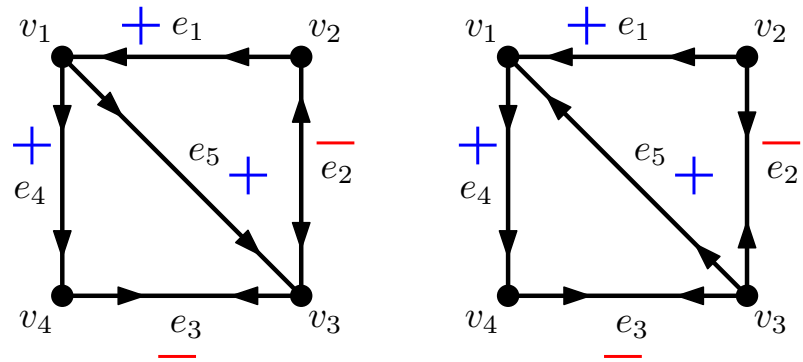


Figure 2.12: A signed Graph can be represented as a bidirected graph in more than one way.

Here we see how two different bidirected graphs correspond to the same signed graph.

Before we can define deletion and contraction for signed graphs, we need the notion of switching. Given a vertex v in a signed graph Σ we define *switching* v by taking all edges incident to v and multiplying their sign by -1. This is equivalent to taking all incidences in a bidirected graph associated to v and multiplying their sign by -1 or reversing the direction of all the arrows in and out of v. A *switching* of an edge e is the act of taking a vertex v incident to e and switching that vertex. Notice that for any edge we can choose either end arbitrarily. The reason we do not care about which end of an edge is switched is that we are primarily interested in the signs of cycles. For any edge in a cycle, both its ends are incident to exactly two edges in the cycle, and thus switching it flips the sign of exactly two edges in any cycle containing the edge and thus the sign of the cycle is unchanged by switching. We will say two graphs are *switching equivalent* if one can be obtained from the other by a sequence of switchings. Further observe that switching a vertex is an involution.

Example 2.1.8. Consider the following signed graph and the edge e_1. The edge is negative in the original graph and has ends v_1 and v_2. We can either switch v_1 or v_2 in order to switch e_1.

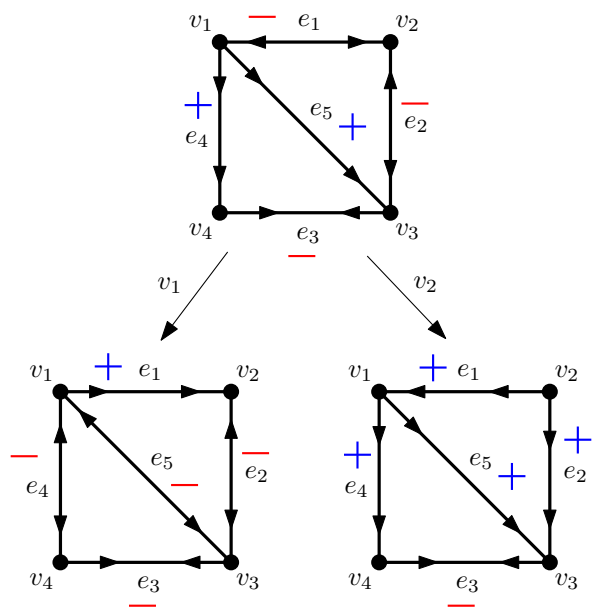


Figure 2.13: A signed graph and two ways to switch e_1

Notice that this graph has three cycles, and the signs of those cycles are preserved under switching. It is fun to observe that these two graphs are switching equivalent regardless of how the switch at v_1 results in only one positive edge while the switch at v_2 results in one negative edge.

There is a direct relationship between the switching equivalence of two signed graphs and their sets of positive cycles. It is an easy observation that if two signed graphs on the same underlying graph are switching equivalent then they have the same set of positive cycles as switching does not change the signs of any cycles. It turns out that the other direction is also true.

Lemma 2.1.9. *[136] Two signed graphs on the same underlying graph are switching equivalent if and only if they have the same set of positive cycles.*

The operation of switching naturally leads to the question of when there exists a sequence of switchings such that we can make every edge positive. Signed graphs with this

property have an easy characterization. A signed graph is *balanced* if every cycle is positive. It is the case that only balanced signed graphs are switching equivalent to an all positive signed graph. This follows directly from the fact that two graphs are switching equivalent if and only if they have the same set of positive cycles.

Lemma 2.1.10. *[136] A graph is switching equivalent to an all positive signed graph if and only if the graph is balanced.*

Example 2.1.11. Using the underlying graph in figure 2.2, which we can view as the all positive signed graph, consider the following different signings on this graph.

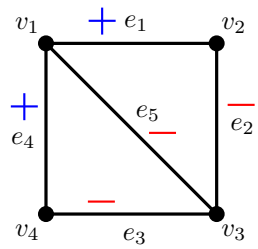


Figure 2.14: A balanced signed graph

Notice that we can compute the signs of all three cycles in the graph in figure 2.14. The cycle $(v_1, e_1, v_2, e_2, v_3, e_3, v_4, e_4, v_1)$ has positive sign as its sign is the product of two negatives and two positives. The cycles $(v_1, e_1, v_2, e_2, v_3, e_5, v_1)$ and $(v_1, e_5, v_3, e_3, v_4, e_4, v_1)$ both have two negative edges and one positive edge. Therefore the graph contains all positive cycles and is thus balanced. If we wanted to switch the graph to the all positive graph we could simply switch v_3 as this would make the three negative edges, e_2, e_3 and e_5 positive.

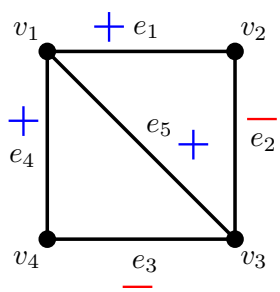


Figure 2.15: An unbalanced signed graph

Unlike our first graph the graph in figure 2.15 is not balanced. The cycle $(v_1, e_1, v_2, e_2, v_3, e_3, v_4, e_4, v_1)$ is unaffected by the change of e_5 from a negative edge to a positive edge, but the cycles $(v_1, e_1, v_2, e_2, v_3, e_5, v_1)$ and $(v_1, e_5, v_3, e_3, v_4, e_4, v_1)$ now both contain exactly one negative edge and are thus negative cycles. No sequence of switching will ever result in all positive edges, but we could switch v_1 to obtain all negative edges.

For the purposes of a signed graph Tutte polynomial we see that a balanced graph is equivalent to an all positive graph under a sequence of switching and for any polynomial that is invariant under switching we won't distinguish between all positive and balanced signed graphs. Like the signed graph Tutte polynomial in [57], our extended signed graph Tutte polynomial in Chapter 3 will also be invariant under switching. Thus unbalanced graphs provide a more interesting class of signed graphs to examine.

The notions of balance and frustration(imbalance) come from Harary in [63] and [64] though the word frustration was not used until later. His primary motivation was to look at a graph as a network and have a notion of "structural balance". He quantified how far a graph is from balanced by looking at how many edges need to be removed or have the signs flipped to create a balanced graph [64]. This is called the *frustration index* of a signed graph. We use these notions of frustration and balance as our aim is to create a signed graph Tutte polynomial which preserves information about maximal spanning forests and retains some information about the cycle space of a graph through tracking which edges form negative cycles with respect to a maximal spanning forest. To discuss how these negative cycles relate to spanning trees we introduce the concept of edge frustration. Let T be a spanning tree of G. An edge $e \notin E(T)$ is *frustrated with respect to* T if $T + e$ contains a negative cycle. This use of frustrated edges will play an important part in our polynomial calculation as new indeterminates will be used explicitly to track frustration.

Example 2.1.12. Taking an unbalanced signed graph we can choose a spanning tree as shown in figure 2.16. Suppose we select e_1, e_3 and e_5 to build our spanning tree. Then we look at the set of edges outside the tree, which is $\{e_2, e_4\}$.

Adding e_2 or e_4 to the spanning tree builds a negative cycle. Thus both edges are frustrated. It is important to note that the sign of the edge (being positive or negative) has no impact on whether or not the edge is frustrated as we care only about the sign of the cycle it creates when added to the spanning tree.

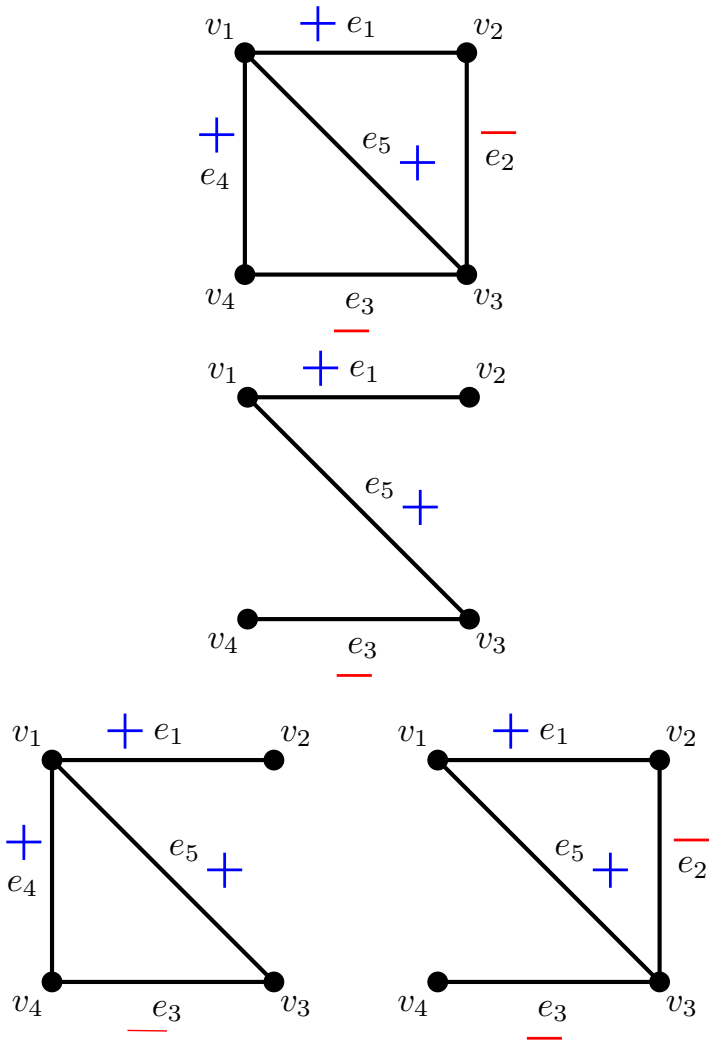


Figure 2.16: Frustrated edge with respect to a spanning tree

Combining these notions of signing with all the graph concepts of the previous section we are now able to examine some further notions on signed graphs. Whether we consider edge signs or consider the graph as a bidirected graph the only definitions from the previous section we need to consider in more detail are deletion and contraction. For a signed graph Σ and edge e we define *deletion*, $\Sigma - e$, to be the graph obtained by deleting e and all incidences mapped to e. This is analogous to the definition for graph. We define *contraction*, G/e, as standard contraction if e is positive and standard contraction after switching one end of e, chosen arbitrarily, if e is negative. Negative edges cannot be

contracted without switching, because contraction of negative edges does not preserve cycle sign and graph balance. Should we wish to contract a negative edge we will use switching to change the edge sign. This notion of not contracting negative edges is common in signed graph theory and aligns with what is done on the signed graph Tutte polynomial in [57]. We will not actually require switching for our extended signed graph Tutte polynomial computation, but we will discuss switching in relation to the polynomial.

2.1.2 Bipartite Representations of Graphs and Oriented Hypergraphs

The definitions for graphs and signed graphs have been presented to try and more seamlessly relate notions on graphs and signed graphs to oriented hypergraphs. In this section we demonstrate this relationship and examine how bipartite representations of hypergraphs can be used to examine hypergraphs as graphs. An *oriented hypergraph* $G = (V, E, I, \varsigma, \omega, \sigma)$ is a set of vertices V, edges E, and incidences I equipped with three functions $\varsigma : I(G) \to V(G)$, $\omega : I(G) \to E(G)$, where $|\omega^{-1}(e)| \geq 1$, and $\sigma : I(G) \to \{+1, -1\}$. We omit σ if all incidences are positive. The *size of an edge* is $|\omega^{-1}(e)|$ and the notion of degree of a vertex and incident vertices and edges is analogous to that for a graph. It is clear that a signed graph with all edges of size two can be viewed as an oriented hypergraph, and a graph with all edges of size 2 without signs can be thought of as assigning all edges +1 and then choosing an orientation on the incidences. In an oriented hypergraph it may not be possible to choose orientations of the incidences such that every adjacency is positive. As more than two vertices may be adjacent to a single edge we consider local travel through edges by looking for weak walks in the hypergraph to move between vertices and take the sign of an adjacency to be the sign of its associated weak walk. Recall that a weak walk is defined as

$$W_{n/2} = (a_0, i_1, a_1, i_2, a_2, i_3, a_3, ..., a_{n-1}, i_n, a_n)$$

where $a_0, a_n \in V(G)$ and $\{a_i\}$ alternates between vertices and edges. One consequence of having multiple incidences between a vertex and edge is that we can have loops within edges. This is to say that an edge e and vertex v may have $|\varsigma^{-1}(v) \cap \omega^{-1}(e)| \geq 2$, but e may also be incidence to a vertex u with $u \neq v$. The incidences used provide clarity on exactly how a weak walk is constructed and where loops are.

Example 2.1.13. This hypergraph contains a 4-edge, a 3-edge, and a 2-edge. We see that v_1 and v_2 are adjacent two ways since we can take $(v_1, i_4, e_2, i_5, v_2)$ or $(v_1, i_4, e_2, i_6, v_2)$ as a path of length one between them. We can also form cycles without using all incidences in an edge.

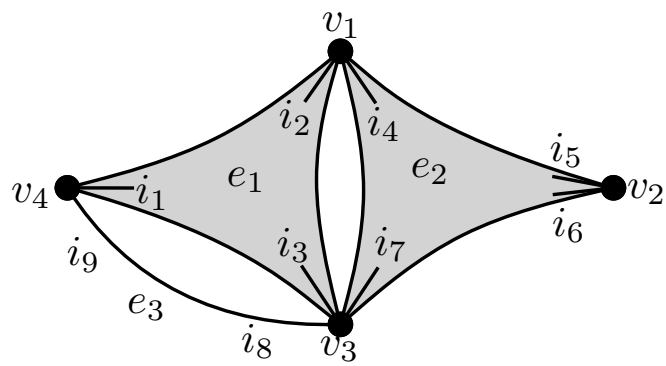


Figure 2.17: A hypergraph

For example, $(v_4, i_1, e_1, i_3, v_3, i_8, e_3, i_9, v_4)$ is a digon or a cycle of length two. We also have a cycle of length three, $(v_4, i_1, e_1, i_2, v_1, i_4, e_2, i_7, v_3, i_8, e_3, i_9, v_4)$. Notice that these cycles are incidence- and edge-monic. We also have a loop $(v_2, i_5, e_2, i_6, v_2)$ within e_2.

The *bipartite representation* of a hypergraph G called Γ_G is the graph with vertex set $V(G) \cup E(G)$, edge set $I(G)$. We take the incidence to be the natural choice to preserve the edge and vertex incident relationships from G which creates an incidence for all $(i, \varsigma(i))$ and $(i, \omega(i))$ such that $\varsigma(i, \varsigma(i)) = \varsigma(i)$, $\varsigma(i, \omega(i)) = \omega(i)$, and $\omega(i, \varsigma(i)) = \omega(i, \omega(i)) = i$. Notice that by construction, all edges have size two as an incidence maps to exactly one vertex and one edge in G. Furthermore for a graph, all the vertices of Γ in $E(G)$ have degree two as they have two associated incidences. For an oriented hypergraph we have degrees corresponding to edge size. As the bipartite representation is a graph regardless of starting with a hypergraph or not we can compute the Tutte polynomial. Thus relationships between the Tutte polynomial of a graph and its bipartite representation would inform similar relations for hypergraphs.

Example 2.1.14. In figure 2.18 we see that each edge in Γ_G has size two and is an incidence of G. Furthermore, since G is a graph each vertex $E(G)$ in Γ has degree two.

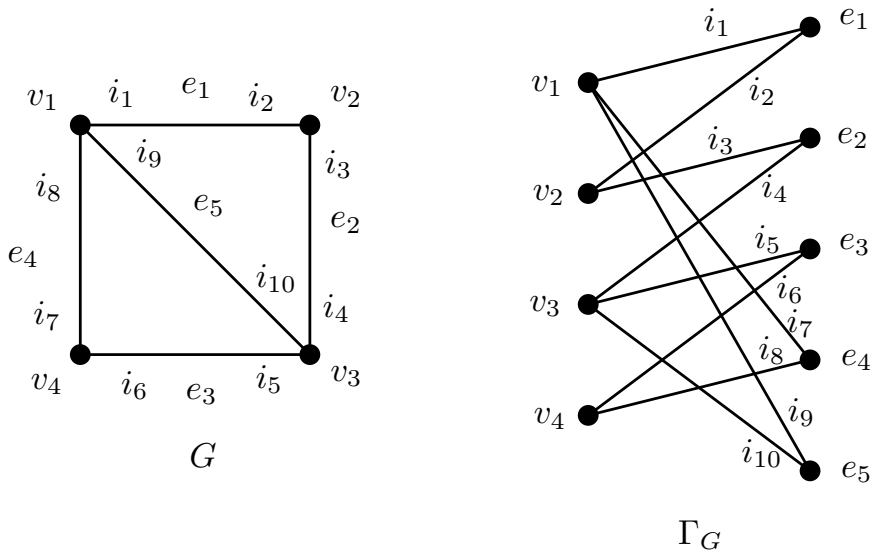


Figure 2.18: A graph and its bipartite representation

In figure 2.19 we see that H has a 4-edge, a 3-edge, and a 2-edge. The corresponding bipartite representation is still a graph with only 2-edges, but we have vertices in $E(H)$ in Γ_H that have degree other than two. Also of note in H is the fact that v_2 is incident to e_2 twice. It is incident once by i_5 and once by i_6 and this shows in Γ_H by making the graph a multigraph, but this has no affect on it being bipartite.

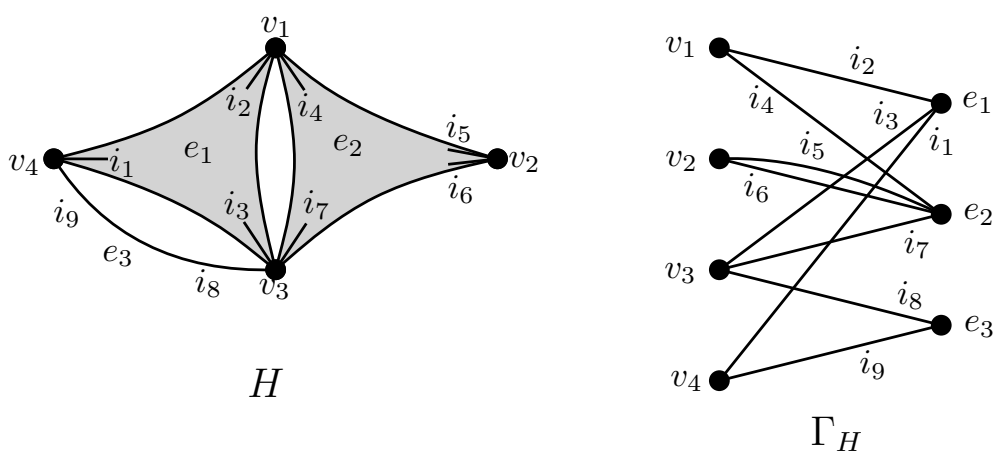


Figure 2.19: A hypergraph and its bipartite representation

When contracting and deleting edges in Γ_G we are manipulating incidences in G. This provides a notion of incidence deletion and contraction, which relates to the concepts of strong and weak deletion [99]. These notions come in to play when considering vertex deletion. Typically when a vertex is deleted from a graph we also delete all incident edges. This is strong deletion. The *strong deletion* of a vertex v from a graph G results in G' with $V(G') = V(G) - v$, and edge set E' with all edges incident to v removed, and incident set I' where all edges incident to v are deleted as are all incidences incident to any edge not in E'. Strong deletion is what we think of as deletion in graphs, but is not the only type of deletion. A *half edge* or 1-edge is an edge incident to exactly one vertex which has exactly one incidence mapped to it by ω. When we delete a vertex but not incident edges, which might leave half edges, we get weak deletion. The *weak deletion* of a vertex v from a graph G results in a graph where we allow 1-edges, G' where $V(G') = V(G') - v$, edge set E' where loops on v are deleted, and incidence set $I - J$ where J is the set of all incidence between v and any edges of G. The idea of weak deletion is more useful in a hypergraph where an edge does not necessarily become a half edge when one of its incident vertices is deleted. Thus having operations of incidence deletion and contraction can provide a more clear approach for defining a Tutte polynomial on oriented hypergraphs.

For edges we have a similar concept, but will use the words partial and complete instead of strong and weak as we are actually choosing incidences to delete. The *partial deletion* of an edge e in a graph G is the deletion of some, but not all incidences mapped to e. The *complete deletion* of an edge e is the deletion of e from the graph and the removal of all incidences associated to e. In order to discuss the relationship, after some type of deletion, between a vertex v and edge e we will say the deletion *disconnects the edge* if v and e were incident in the graph before the deletion, but not after. This is important because when edges and vertices have more than one incidence between them not all partial deletions of the edge will result in disconnecting the edge from a vertex.

When we take a graph and its bipartite representation as seen in figure 2.18, if we delete and contract edges in the bipartite representation this corresponds to manipulating incidences in the original graph.

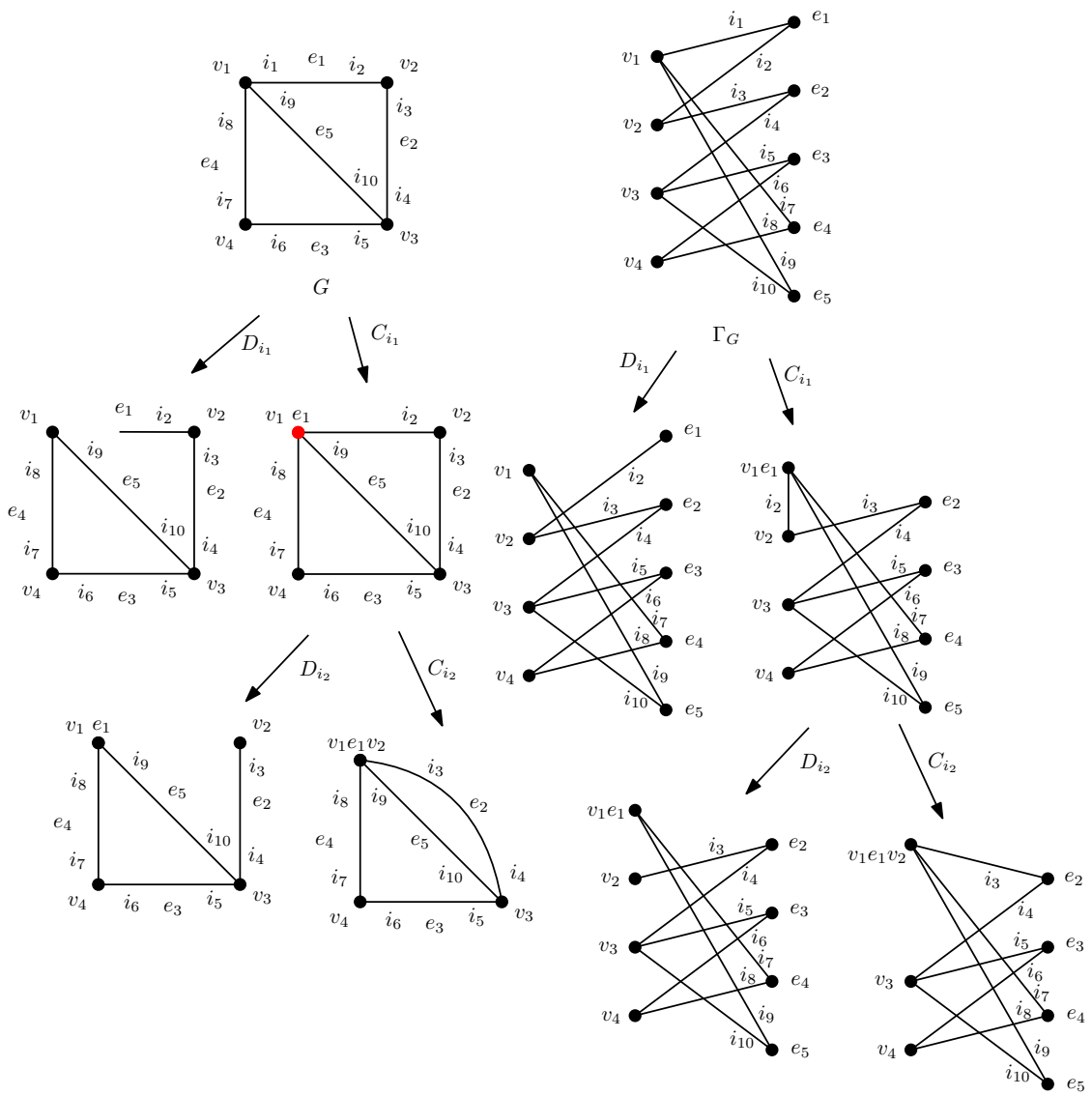


Figure 2.20: Deletion and contraction from the bipartite representation of a graph

When working with a single incidence (which is an edge in the bipartite representation) if we delete it in a graph we leave a half edge as seen by the deletion of i_1 in figure 2.20. Converting between the graph and its bipartite representation we can look for what operations or sequences of operations on the bipartite graph produce valid graphs when performed on the original graph. The first thing we observed is that in order to avoid half

edges in the original graph we cannot start with a deletion. So we consider starting with a contraction. Now looking at figure 2.20 we see that only performing a contraction of an incidence makes an object in the original graph, labeled v_1e_1 and shown in red, that is trying to be both a vertex and edge simultaneously. However, this can be resolved by another deletion or contraction as shown by deleting and contracting i_2. In general it might take more than one more operation to return to a bipartite graph, where the number of operations is dependent on the size of the edge that the incidence is incident to. It is the hope that understanding the affect of deletion and contraction in the bipartite representation will create clear operations of contraction on incidences that we can use to extend the Tutte polynomial to hypergraphs.

When working with a hypergraph and its bipartite representation we can convert between the two to see how deleting and contracting incidences (which are edges in the bipartite representation) affects the hypergraph. In figure 2.21 we see that deleting an incidence, like i_4 does not immediately result in a half edge. Half edges created upon incidence deletion is an immediate consequence of working with an edge of size two and so we suspect there will be more ways to manipulate an edge than just strong deletion and contraction when working with a hypergraph that relate to the edge size.

However we do still get the issue after contraction of an object, in this case v_1e_2 highlighted in red after the contraction of i_4, that wants to simultaneously act like a vertex and edge and the bipartite representation is no longer bipartite. Here we see that working with i_4, which lives in a 3-edge, we are able to obtain three distinct hypergraphs through sequences of deletion and contraction. The two hypergraph that result from contracting i_4 and then deleting or contracting i_2 are what we suspect arise from what we tentatively consider partial and complete contraction. When we delete i_2, v_1 was contracted into one incident edge by i_4 and disconnected from the other incident edge. Whereas when we contract i_2 we see that all edges incident to v_1 were merged.

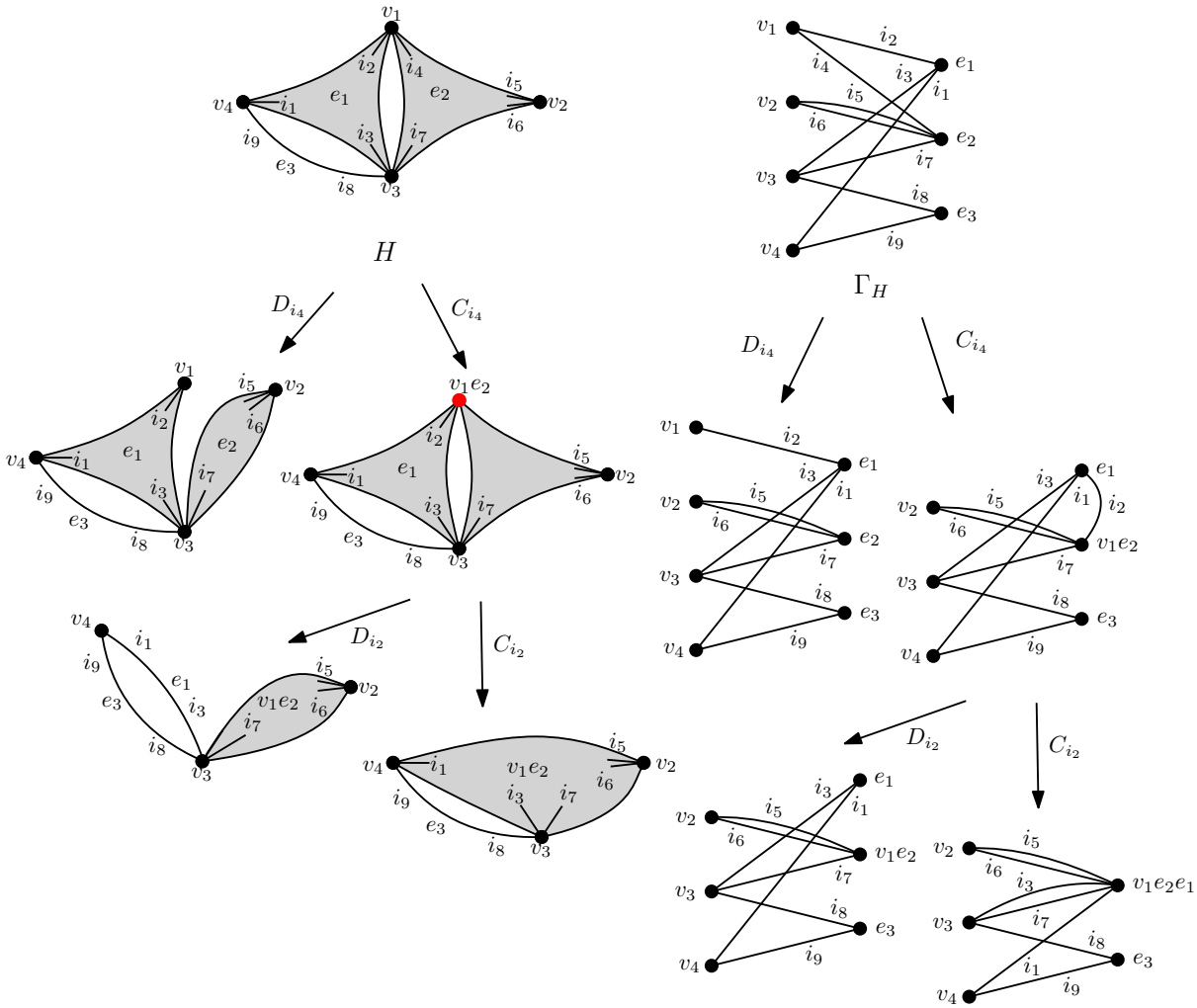


Figure 2.21: Deletion and contraction from the bipartite representation of a hypergraph

In a graph or signed graph we have the notion of an isthmus. While isthmi do still occur in oriented hypergraphs we also need to generalize this notion to edges of any size. A *thorn* is an edge whose strong deletion would disconnect a connected component of an oriented hypergraph. We can see that in figure 2.17, where if we take the strong deletion of edge e_2, that is to say we completely remove it and all associated incidences from the graph, then we disconnect v_2 from the rest of the graph. Thus e_2 is a thorn. Notice further that we could partially delete e_2 by deleting any one incidence and we would not disconnect the graph. We could even delete two incidences, specifically one of i_5 and i_6 and one of i_4 and

i_7 and the hypergraph would remain connected, but deleting i_5 and i_6 would disconnect it. This is due to v_2 being incident to e_2 twice. In chapter 5 we will address how thorns and multiple incidences between and edge and vertex impact a Tutte-like polynomial for hypergraphs.

2.1.3 Graphic Matroids

One of our main motivations behind our signed graph extended Tutte polynomial in Section 3.4 is to preserve the relationship between deletion and contraction, the Tutte polynomial, and maximal spanning forests which are bases for a graphic matroid. In this section we will establish the matroid theory we will use throughout this thesis. We will follow [95] for the notation and definitions presented here.

A *matroid* M is a pair $(\mathcal{E},\mathcal{I})$ where $\mathcal{E}$, called the *ground set*, is a finite set and $\mathcal{I}$ is a set of subsets of $\mathcal{E}$ called *independent sets* which satisfy the following

1. $\mathcal{I}$ is non-empty,
2. If $J \in \mathcal{I}$ then so is every subset of J, and
3. If $J, K \in \mathcal{I}$ and $|J| \geq |K|$ then there exists $j \in J - K$ such that $K \cup \{j\} \in \mathcal{I}$.

For a matroid $M = (\mathcal{E},\mathcal{I})$ we say $J \in \mathcal{I}$ is a *basis* or maximally independent if $J \cup \{e\} \notin \mathcal{I}$ for all $e \notin J$. A subset of $\mathcal{E}$ is dependent if it is not in $\mathcal{I}$ and a dependent set is a *circuit* or minimally dependent if every proper subset of it is independent. Note this differs from the definition of a circuit in a graph, but both terminologies are standard in the literature. Unless explicit discussing matroids, we will use circuit in the graph context here. The *rank* of a matroid is the size of a basis. It can be noted that all bases have the same size.

The *graphic matroid* also known as a *cycle matroid* has ground set as the set of all edges of a graph and a forest is an independent set. Thus the bases are all maximal spanning forests. The circuits are cycles and the rank is the size of a maximal spanning forest.

For signed graphs we use the work of Zaslavesky in [137]. The *lift matroid* has ground set as the edge set of the underlying graph. Its independent sets are forests or sets of edges that contain exactly one negative cycle. Thus if the signed graph is balanced the lift matroid and graphic matroid on the underlying graph have the same set of independent sets and circuits and thus have the same matroid. If the signed graph is unbalanced then the bases are maximal spanning forests plus an edge that forms a negative cycle and the

circuits are positive cycles or two negative cycles joined by a path. For a maximal spanning forest of an unbalanced signed graph, the maximal spanning forest is an independent set but not a basis. So there exists some edge that forms a negative cycle when added to the maximal spanning forest as all maximally independent sets have the same cardinality.

So given a maximal spanning forest of an unbalanced signed graph the set of frustrated edges with respect to that forest can each be used to build a basis for the lift matroid.

2.2 Tutte Polynomial

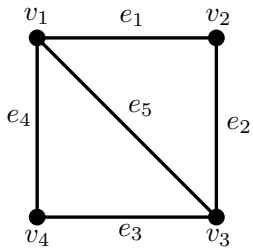
Before defining the Tutte polynomial, we need to define some terminology that allows us to manipulate the edges of a graph as well as describe the relation of the edges to each other and to cycles. The *cyclomatic number* ϕ or *nullity* of a graph G, also denoted by $n(G)$, is the minimum number of edges that must be deleted from G to obtain an acyclic graph, we count loops as cycles and thus they always contribute to the nullity. In quantum field theory this is called the loop order. The *rank* of G, denoted by $r(G)$, is the number of edges in a maximal spanning forest of G which is the same as the rank of the graphic matroid. Clearly, these two definitions are closely related as the sum of the rank and cyclomatic number is the number of edges in a graph. We can define the Tutte polynomial from the rank and nullity of the graph.

Definition 2.2.1. *Let $G = (V, E)$ be a graph. Then the* Tutte polynomial *of G is*

$$T_G(x, y) = \sum_{A \subseteq E} (x-1)^{r(G)-r(G[A])}(y-1)^{n(G[A])} \tag{2.1}$$

where $G[A]$ is the edge-induced spanning subgraph of G, $(V(G), A)$.

This definition of the Tutte polynomial is most closely related to the Whitney rank generating function, which is also a rank generating function for matroids, denoted as $W(G; x, y)$. It is known that $W(G; x, y) = \sum_{A \subseteq E}(x)^{r(G)-r(G[A])}(y)^{n(G[A])} = T_G(x+1, y+1)$ [127, 119]. As seen in figure 2.22 we take all subsets of the edge set of a graph and consider the spanning subgraph created. This definition is not particularly convenient to compute by hand, but is convenient for computing the Tutte polynomial of a graph via a computer.



$r(G) = 3$
$n(G) = 2$

$A \subseteq E$

$\emptyset$

rank= 0 $\quad (x-1)^{3-0}(y-1)^0 = x^3 - 3x^2 + 3x - 1$

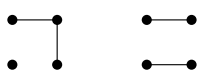
nullity= 0

any one edge (5 choices)

rank= 1 $\quad 5(x-1)^{3-1}(y-1)^0 = 5x^2 - 10x + 5$

nullity= 0

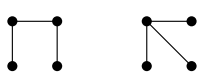
any two edges (8 choices of incident edges and 2 for independent edges)



rank= 2 $\quad 10(x-1)^{3-2}(y-1)^0 = 10x - 10$

nullity= 0

any three edges that don't form a cycle (6 choices for a path and 2 for the star)



rank= 3 $\quad 8(x-1)^{3-3}(y-1)^0 = 8$

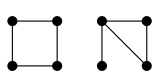
nullity= 0

any three edges that form a cycle(2 choices)

rank= 2 $\quad 2(x-1)^{3-2}(y-1)^1 = 2xy - 2x - 2y + 2$

nullity= 1

any four edges(5 choices)



rank= 3 $\quad 5(x-1)^{3-3}(y-1)$=5y-5

nullity= 1

$E(G)$

rank= 3 $\quad (x-1)^{3-3}(y-1)^2 = y^2 - 2y + 1$

nullity= 2

$$T_G(x,y) = x^3 + 2x^2 + x + 2xy + y + y^2$$

Figure 2.22: Tutte polynomial calculation using rank and nullity

While this definition of the Tutte polynomial provides some insight into the importance of cycles and spanning trees, it does not provide a method for extending the Tutte polynomial to signed graphs and oriented hypergraphs that preserves basis information or activities which is our primary interest. Thus we consider two other methods for defining the Tutte polynomial.

Before we define the Tutte polynomial in other ways, we need a few more definitions. Let G be a connected graph and $\overrightarrow{E} = (e_1, e_2, ..., e_n)$ be a total ordering on the edges where $e_i < e_j$ if and only if $i < j$. Let F be a maximal spanning forest of G and $e_i \in E(F)$. Consider $B_i = \{e_j | F - e_i + e_j$ is a maximal spanning forest of $G\}$ This is called the *fundamental bond* of e_i with respect to F. If $i \geq j$ for all j such that $e_j \in B_i$ then e_i is *internally active*, otherwise e_i is internally inactive. That is to say, e_i is internally active if it is maximal in the cut induced by $F - e_i$ in G. Now let $e_j \notin E(F)$. Consider the unique cycle, which will contain e_j, in $F + e_j$, this is called the *fundamental cycle* of e_j with respect to F. If $j \geq k$ for all e_k in this cycle then e_j is *externally active* otherwise e_j is externally inactive. The standard definition for activities uses minimal as opposed to maximal [47]. We make the choice to use maximal due to a preference for prioritizing maximal spanning forests that contain as many edges higher in the total edge ordering as possible but the choice of maximal or minimal has no impact on any result presented.

Example 2.2.2. Let G be K_4 minus an edge with spanning tree $T = \{e_1, e_3, e_5\}$ as seen in figure 2.23, and consider $T - e_5$. Consider all the ways to add an edge to $T - e_5$ to create a spanning tree. This can be done by adding to $T - e_5$ any of e_2, e_4 or e_5. Observe that e_5 is the maximal edge and thus internally active.

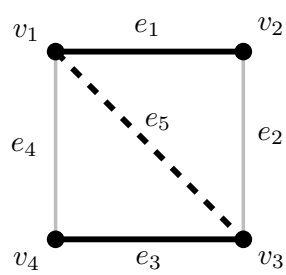


Figure 2.23: e_5 is internally active

Now consider an edge not in T, say e_4. Consider the unique cycle containing e_4 in $T + e_4$ shown in figure 2.24. This cycle contains $\{e_3, e_4, e_5\}$. Observe that $5 > 4$ so e_4 is not maximal and hence e_4 is externally inactive.

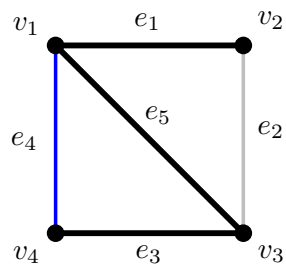


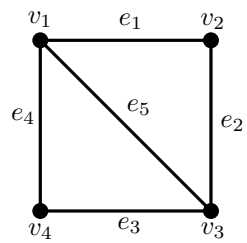
Figure 2.24: e_4 is externally inactive

The Tutte polynomial can be equivalently defined in several other ways that are also independent of the total order assigned to the edges of a graph, which is used in some definitions. The independence from the edge order is a nontrivial argument two different versions of which can be found in [16] and relies on known equivalences to other definitions and uses weight functions. In this same book Bollobás also provides nice proofs of the equivalence of the various definitions of the Tutte polynomial using induction on the size of the edge set of the graph. Given the activities and deletion-contraction we can now discuss the definitions of the Tutte polynomial we will be using.

Theorem 2.2.3. *Let G be a graph and $\overrightarrow{E}$ a total ordering on the edges. Further let $\mathscr{T}_G$ be the set of maximal spanning forests of G and for each $F \in \mathscr{T}_G$ let f_i be the number of internally active edges and f_e the number of externally active edges. Then the* Tutte polynomial of G *is*

$$T_G(x,y) = \sum_{F \in \mathscr{T}_G} x^{f_i} y^{f_e} \tag{2.2}$$

Example 2.2.4. To calculate the Tutte polynomial of a graph we first need the set of all maximal spanning forests. Using the graph G in figure 2.2 and its set of maximal spanning forests, which are spanning trees as G is connected, in figure 2.10 we can compute the activity of each edge with respect to each spanning tree. In figure 2.25 we have each spanning tree with the set of internal and external edges beside it. Edges are then marked I for inactive or A for active. We will discuss in more detail here the spanning trees that provide the y^2 and x^3 monomials. The y^2 monomial comes from the tree containing edges e_1, e_2 and e_3. These happen to be the three smallest edges in the edge order and the external edges e_4 and e_5 are largest in edge order.



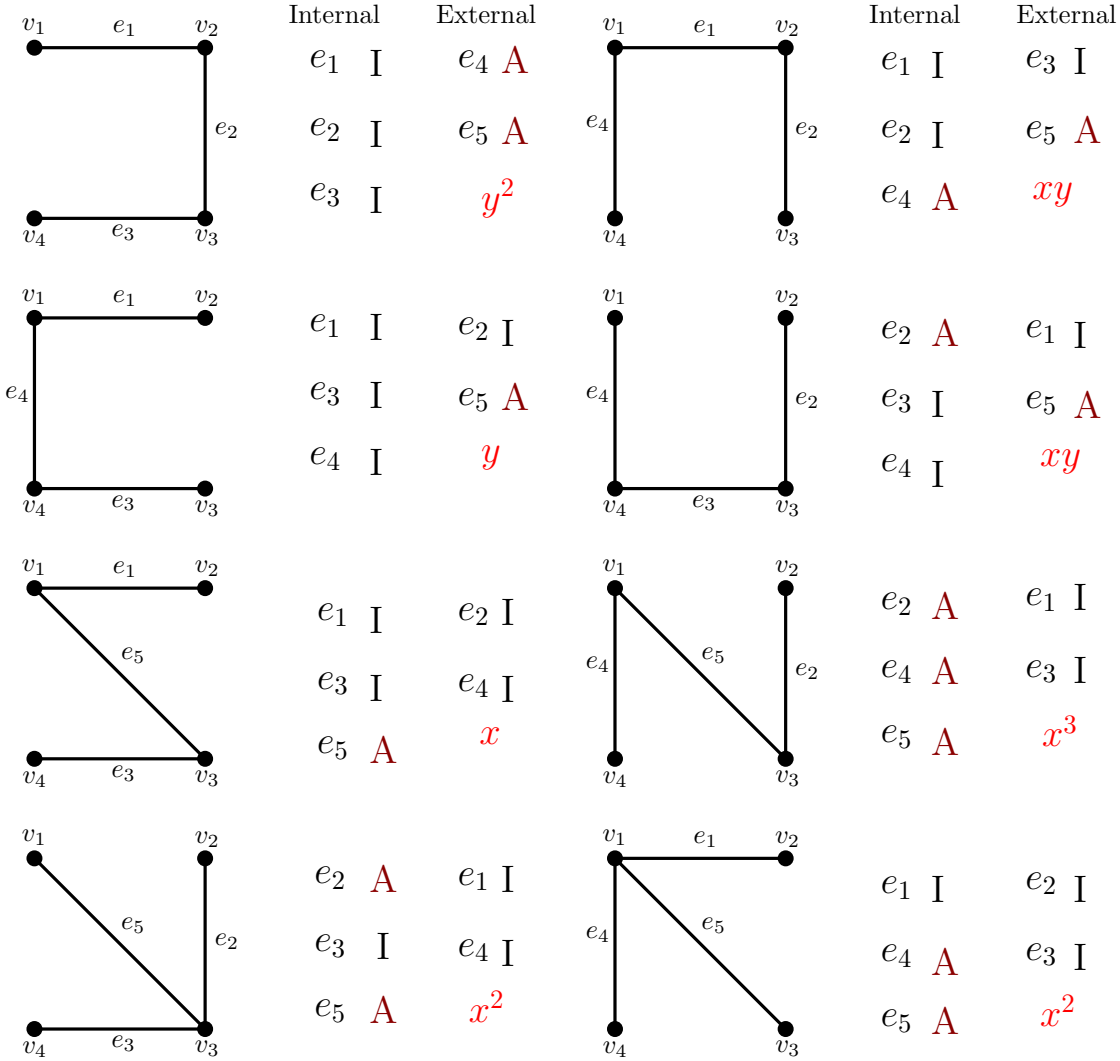


Figure 2.25: Tutte polynomial calculation via activities

The monomial with the highest power of y will always correspond the maximal spanning forest that contains as many edges low in the edge ordering as possible. That is to say, if you construct a maximal spanning forest by starting with no edges and adding the smallest edge in the given edge order that does not build a cycle then this construction will yield the monomial with the largest number of externally active edges. The reverse of this notion can be used to construct the maximal spanning forest with the largest number of internally active edges. For our example in figure 2.25 we can start by choosing edges e_5 and e_4. We cannot then choose e_3 as this builds a cycle so instead we add e_2 to finish building a spanning tree and we can see that this corresponds to the x^3 term, which has the maximal number of internally active edges.

We may make a few further interesting observations, the edge smallest in edge ordering, so e_1, will never be active unless it is a loop (and thus external) or an isthmus (and thus internal). Otherwise we can always choose a larger edge in any cycle containing it or a larger edge in the fundamental bond with respect to any maximal spanning forest. One can see that e_1 is always inactive in figure 2.25. The largest edge in the edge ordering, in our example this is e_5, will always be active regardless of being a loop, isthmus, or non-isthmus, link edge. This occurs as it is the maximal edge automatically and thus there can never be a larger edge in any fundamental bond of fundamental cycle regardless of the choice of maximal spanning forest.

We can also recursively define the Tutte polynomial based on deletion and contraction. This definition relies on sequences of deletions and contractions on the edges where we are not allowed to delete and contract loops or isthmi. A sequence of contractions and deletions terminates when all edges remaining in the graph are loops or isthmi; we call this a *terminal minor*.

Theorem 2.2.5. *Let G be a graph. The* Tutte polynomial *is*

$$T_G(x,y) = \begin{cases} 1, & \text{if } E = \varnothing \\ x^n y^m, & \text{if } G \text{ has only } n \text{ isthmi and } m \text{ loops} \\ T_{G\setminus e}(x,y) + T_{G/e}(x,y) & \text{if } e \in E(G) \text{ is neither an isthmus nor a loop} \end{cases} \tag{2.3}$$

Example 2.2.6. Recall the deletion contraction decision tree from figure 2.9. We may now assign monomials to all terminal objects by counting the number of bridges and isthmi in each terminal minor after deletion and contraction.

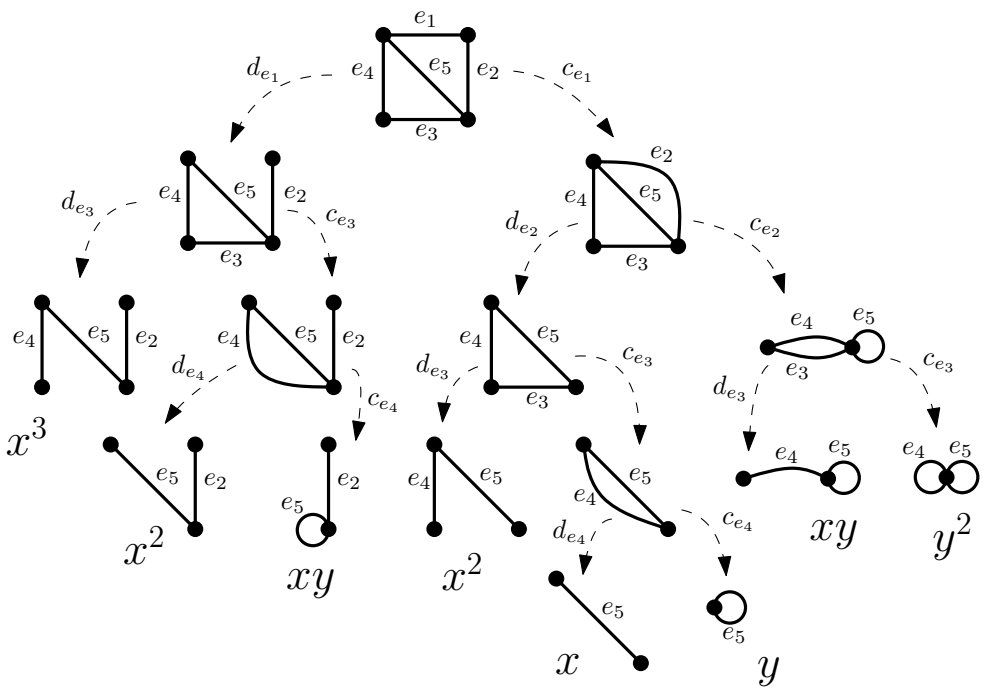


Figure 2.26: A deletion contraction decision tree for the Tutte polynomial

The Tutte polynomial is then the sum over all these monomials, giving us $T_G(x,y) = x^3 + 2x^2 + x + 2xy + y + y^2$.

One important feature of the Tutte polynomial its generality, which Tutte discusses in [118]. Following the modern notation of [16] we present the notion that for any minor closed class of graphs $\mathscr{G}$ and any commutative ring $\mathscr{R}$, if $f : \mathscr{G} \to \mathscr{R}$ is a graph invariant then, if f satisfies for graphs G and H that

1. $f(G) = af(G \setminus e) + bf(G/e)$ for e and ordinary edge,
2. $f(K_1) = 1$ where K_1 is an isolated vertex,
3. $f(G \sqcup H) = f(G) \cdot f(H)$, and
4. $f(G * H) = f(G) \cdot f(H)$ where $G * H$ is the one-point join of G and H

then f is a *Tutte-Grothendieck invariant*. The Tutte polynomial is in fact the most general Tutte-Grothendieck invariant and any other invariant satisfying the conditions must be an evaluation of it. A proof of this in an even more general sense can be found in [16].

Theorem 2.2.7 ([16]). *For any minor-closed class of graphs $\mathscr{G}$, there is a unique map $U : \mathscr{G} \to \mathbb{Z}[x, y, \alpha, \sigma, \tau]$ such that*

$$U(G) = \begin{cases} \alpha^n & \text{if } E(G) = \varnothing \text{ and } |V(G)| = n \\ xU(G \setminus e) & \text{if } e \text{ is an isthmus,} \\ yU(G \setminus e) & \text{if } e \text{ is a loop,} \\ \sigma U(G \setminus e) + \tau U(G/e) & \text{if } e \text{ is an ordinary edge} \end{cases} \tag{2.4}$$

Furthermore,

$$U(G) = \alpha^{k(G)}\sigma^{n(G)}\tau^{r(G)}T_G(\frac{\alpha x}{\tau}, \frac{y}{\sigma}). \tag{2.5}$$

As a consequence of the Tutte polynomial being the Tutte-Gronthendieck invariant we can obtain a wide variety of information about a graph by evaluating its Tutte polynomial on various curves and points. Table 2.1 presents some of the point and curve evaluations and their meaning for the Tutte polynomial of a graph.

$T_G(1,1)$	Number of maximal spanning forests in G
$T_G(1,2)$	Number of connected spanning subgraphs
$T_G(2,1)$	Number of spanning forests
$T_G(2,2)$	Number of spanning subgraphs
$T_G(1,0)$	Number of acyclic orientations with a unique source
$T_G(2,0)$	Number of acyclic orientation
$T_G(0,1)$	G is planar, number of cyclic orientation without clockwise cycle
$T_G(0,2)$	Number of cyclic orientation
$T_G(3,3)$	Number of tetromino tilings
$T_G(0,-2)$	For 4-regular G counts number 2-in, 2-out orientations
$T_G(-1,-1)$	$(-1)^{\lvert E(G)\rvert}(-2)^{dim(B)}$ where B is the bicycle space of G
$T_G(2,1)$	Number of score sequences of G
$(-1)^{r(G)}x^{k(G)}T_G(1-x,0)$	Chromatic polynomial $\chi_G(x)$
$(-1)^{n(G)}T_G(0,1-x)$	Flow polynomial $F_G(x)$
$(1-p)^{n(G)}p^{r(G)}T_G(1,\frac{1}{1-p})$	Reliability polynomial $Rel_G(p)$
$\alpha T_G(1+\frac{q}{e^\beta-1}, e^\beta)$	Potts model $Z_P(G;\beta,q)$ where $\alpha = e^{-\beta\lvert E(G)\rvert}q^{k(G)}(e^\beta-1)^{r(G)}$
$\alpha T_G(1+\frac{q(1-p)}{p}, 1+\frac{p}{1-p})$	Random cluster model $Z_{RC}(G;p,q)$, $\alpha = (1-p)^{n(G)}p^{r(G)}q^{k(G)}$
$T_{G^*}(x,y) = T_G(y,x)$	for planar G and its planar dual G^*
$m_{\overrightarrow{G_m}}(x) = T_G(x,x)$	the directed 2-in 2-out Martin polynomial

Table 2.1: Evaluations of the Tutte Polynomial at points

Proofs of these results can be found, in most cases in [47] or [16]. $T_G(0,1)$ appears in [60] and $T_G(0,-2)$ relates to the Martin polynomial and comes from physics appearing in [89, 97] which also happens to be the number of nowhere-zero $\mathbb{Z}_3$-flows for a 4-regular graph. $T_G(-1,-1)$ was proved in [100] and $T_G(2,1)$ was established in [108]. As a consequence of the flow and chromatic polynomial evaluations, and the planar duality property for a planar graph G and its dual G^*, $\chi_G(x) = xF_{G^*}(x)$ [119]. The relationship between the Tutte polynomial and one of the Martin polynomials holds in the special case where G is a connected plane graph and $\overrightarrow{G_m}$ is its directed medial graph [91].

We can also look at specific coefficients of the Tutte polynomial for a graph. For these purposes we shall assume that $T_G(x,y) = \sum_{i,j} t_{i,j}x^iy^j$ and table 2.2 summarizes some interesting properties of specific $t_{i,j}$ values of which more detailed summaries can be found in [47].

$t_{r(G),j}$	The coefficient of the largest power of x which is $x^{r(G)}$ is y to the number of loops of G
$t_{i,n(G)}$	The coefficient of the largest power of y which is $y^{n(G)}$ is x to the number of isthmi in G
$t_{0,0}$	is zero unless $E(G) = \varnothing$
$t_{1,1}$	is zero only when G is a cycle or its dual
$t_{0,1} = t_{1,0}$	for G with at least two edges
$min\{i+j; t_{i,j} > 0\}$	the number of nontrivial connected components
$\sum_{i+j<k}(-1)^j\binom{k-i-1}{j}t_{i,j} = 0$	if G has at least k edges, $k \geq 1$

Table 2.2: Coefficients in the Tutte polynomial with specific meanings

2.3 Martin Polynomial

As seen in table 2.1 the Martin and Tutte polynomials do share a relationship, specifically in the 2-in 2-out case of the Martin polynomial. However our primary interest is in the $2k$-regular Martin polynomial particularly for $k = 2$. To define the Martin polynomial we first need to discuss Eulerian graphs, circuit partitions, and transition systems. A *circuit* is a closed walk where no edge is repeated. A graph is *Eulerian* if there exists a circuit containing every edge. When we take a *circuit partition* of an Eulerian graph we partition all the edges of the graph into circuits. For an Eulerian graph the Martin polynomial is designed to capture information about the various circuit partitions of the graph. We will focus specifically on the Martin polynomial for $2k$-regular graphs, which are Eulerian,

but the concepts here extend to Eulerian graphs in general. Furthermore, we will assume for this section that the graphs in question are connected, but we will allow loops and multi-edges. To define the Martin polynomial, we will begin with transitions systems of $2k$-regular graphs. A *transition* at a vertex v of an $2k$-regular graph G is a partition of the $2k$ edges incident to v into pairs. Each pair becomes, in essence, a single edge and the vertex v is ignored. Let $t(v)$ denote the set of all $(2k-1)!!$ transitions at v. A *transition system* T is a choice of a transition at each vertex of G yielding a partition of the edges of G into circuits. This occurs because when starting on any edge of the graph we can simply follow a walk by taking the edge in and out of any vertex we pass through that were paired in the transition chosen at that vertex. For a digraph $\overrightarrow{G}$ we require that the transition respect the orientation of the graph. That is to say an edge entering v must be paired with an edge exiting v. Given that the graph is finite and even regular each walk must end where it began creating a circuit. Multiple circuits may be created, but every edge of the graph will be in a single circuit yielding the desired partition. Let $k(T)$ denote the number of circuits induced by a transition system T and $\mathcal{T}(G)$ be the set of all transition systems of G. Finally, denote $c(G)$ the number of connected components of G. If the graph is a digraph, we simply require that all circuits and transition systems respect the orientations of the digraph.

Definition 2.3.1. *[91] For a 2-in 2-out digraph $\overrightarrow{G}$, the* Martin polynomial *is $m(\overrightarrow{G};y) = \sum_{T\epsilon\mathcal{T}(\overrightarrow{G})}(y-1)^{k(T)-c(\overrightarrow{G})}$.*

This is the version of the Martin polynomial that is related to the Tutte polynomial. However, here we are concerned with the 4-regular case as we are motivated by quantum field theory. We will now discuss the $2k$-regular Martin polynomial we will be utilizing and then discuss its connection to quantum field theory through Feynman periods. The $2k$-regular Martin polynomial can be defined very similarly to the 2-in 2-out Martin polynomial.

Definition 2.3.2. *[91, 86] Let G be a $2k$-regular graph. The* Martin polynomial *of G is $m(G;y) = \sum_{T\epsilon\mathcal{T}(G)}(y-2)^{k(T)-c(G)}$.*

The Martin polynomial can also be defined recursively by resolving the transition system at each vertex. See figure 2.27 for a visual representation of a step of the recursion on a 4-regular graph where a vertex can be resolved three ways. It was shown in [91] that definition 2.3.2 and 2.3.3 are equivalent.

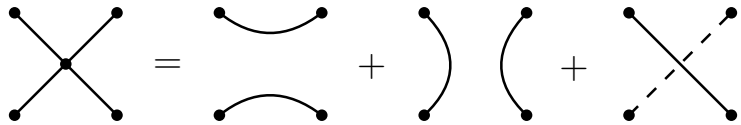


Figure 2.27: The four ways to resolve a vertex in a 4-regular graphs

Definition 2.3.3. *[91, 86] For a $2k$-regular graph G the* Martin polynomial *of G is*

$$m(G;y) = \begin{cases} 1, & \text{if } V = \varnothing \\ y & \text{if } |V| = 1 \\ ym(G';y) & \text{if } v \text{ is a cut vertex and } G' \text{ is obtained by a transition that} \\ & \text{does not increase the number of connected components} \\ \sum_{T \in t(v)} m(G_T;y) & \text{otherwise} \end{cases}$$

where G_T is obtained from $G - v$ by adding edges between $x, y \in G - v$ if $x, y \in N_G(v)$ and their incident edges to v are in the same part of the partition defined by the transition T.

Note that a vertex with a loop is considered a cut vertex as it has a transition system that disconnects the loop from the rest of the system. Not all definitions of the Martin polynomial include a special case for cut vertices. To see how this special case for cut vertices occurs, consider a graph G with a cut vertex and suppose we compute the Martin polynomial calculated using definition 2.3.2. As seen in figure 2.29 we may group the transition systems in groups such that they only differ at the cut vertex and then we obtain $(1 + (y - 2) + 1)m(G' : y)$ or simply $ym(G';y)$. Hence we only need to choose a transition system that does not increase the number of connected components and multiply by y. So the condition of choosing a single transition system if v is a cut vertex it not strictly necessary, but allows us to not compute all transitions at a cut vertex. This argument generalizes to any even vertex, but we demonstrate only in the degree 4 case.

Example 2.3.4. We can calculate the Martin polynomial for K_3 where each edge has been replaced by a pair of edges. First we take all possible transition systems at vertex a and get three new graphs with one fewer vertex. One of these graphs now has a cut vertex and thus we get y times a transition of our choice that does not produce more connected components.

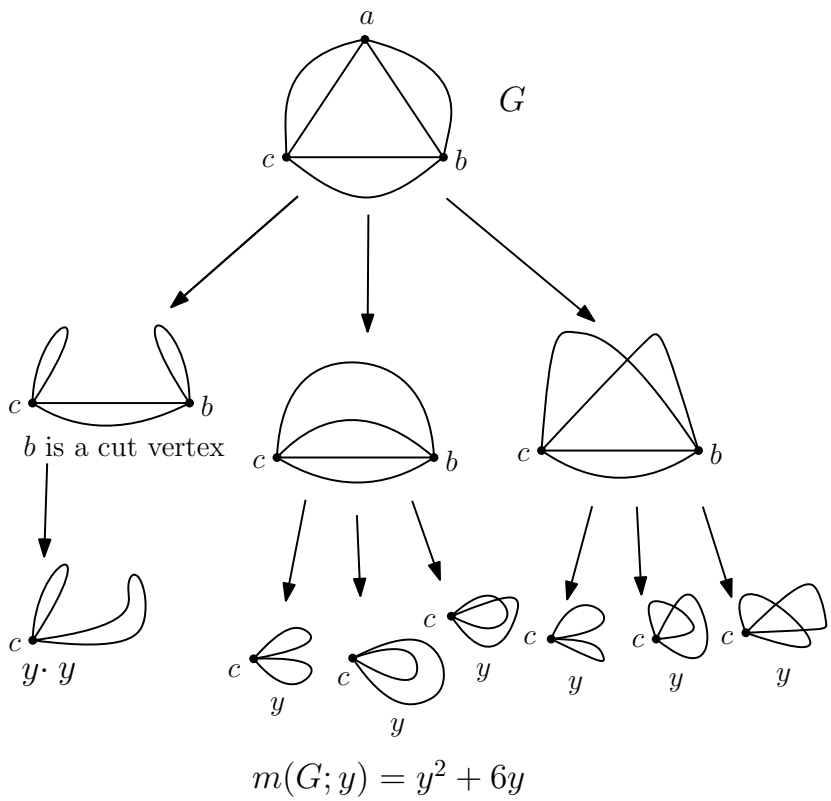


Figure 2.28: Martin polynomial calculation

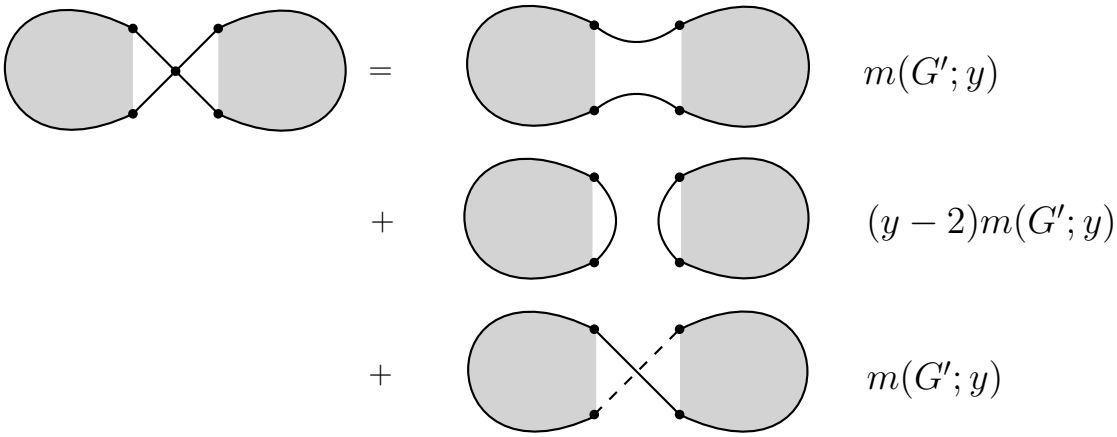


Figure 2.29: Illustration of how a cut vertex affect the transition system

While the Martin polynomial does have a relationship with the Tutte polynomial, this is not the motivation behind the importance of the Martin polynomial to the work here. To establish the connection with quantum field theory through Feynman periods, we begin with the Feynman integral. The origins of the Feynman integral lie in modeling particle interactions. The possibilities of how the particles react can be modeled by graphs and Feynman integrals. The Feynman period is a more specialized case of a Feynman integral that converges if G is isomorphic to $H - v$ where H is a 4-regular graph that is internally 6-edge-connected [133]. This means that the only edge cuts which disconnect a single vertex have 5 or fewer edges. The Feynman period is primarily used in quantum field theory where it is related to the beta function [104]. There has also been a body of work done on computing the Feynman period of various graphs, as well as looking at graph invariants and symmetries under which the period is unchanged. It is these symmetries which will motivate the work here, but we will begin by defining the dual Kirchhoff polynomial and the Feynman period.

Definition 2.3.5. *For a graph G the* dual Kirchhoff polynomial *(first Symanzik polynomial) is $\psi_G = \sum_{F \in \mathcal{F}(G)} \prod_{e \notin F} a_e$ where $\mathcal{F}(G)$ is the set of maximal spanning forests of G.*

Notice that for a connected graph we sum over spanning trees. When working with the Feynman period and Martin polynomial we will assume our graphs are connected.

Definition 2.3.6. *For a graph G the* Feynman period *is the integral* $\mathcal{P}(G) = \int_0^\infty \frac{da_2 \ldots da_{|E(G)|}}{\psi_G^2}\Big|_{a_1=1}$.

A natural field of research is to look for symmetries under which the Feynman period is invariant as computing it can be incredibly difficult unless it belongs to special classes of graphs or is related by a symmetry to a graph with a known period [68, 104]. Before presenting the symmetries we introduce the notion of vertex cleaving. We define *cleaving* to be the act of splitting a vertex v into two new vertices v_1 and v_2 such that all edges incident to v are incident to exactly one of v_1 or v_2. Here we will in general be cleaving cut vertices such that we assign edges to v_1 or v_2 based on which component of the cut they have their other end in, so that we create two components after the cleaving.

Theorem 2.3.7. *[68, 104] For the Feynman period of a graph G that is the decompletion of a graph H which was 4-regular and internally 6-connected the following hold*

- *If G is planar then $\mathcal{P}(G) = \mathcal{P}(G^*)$.*
- *$\mathcal{P}(H - v) = \mathcal{P}(H - u)$ for any $u, v \in V(H)$.*

- *(product) If H has a 3 vertex separator, a, b, c, then $\mathcal{P}(H - v) = \mathcal{P}(H_1 - v) \cdot \mathcal{P}(H_2 - v)$ where H_1 and H_2 are obtained by cleaving the three cut vertices a, b, c and adding edges $a_i b_i, a_i c_i$ and $b_i c_i$ for $i \in \{1, 2\}$. See figure 2.31.*

- *(twist) If H has a 4 vertex separator then $\mathcal{P}(H - v) = \mathcal{P}(H' - v)$ where H' is obtained by taking the 4-vertices as two pairs and swapping in those pairs all incident edges on one side of the cut. See figure 2.32.*

- *If G has a 3 vertex separator, u, w, v, and the subgraph on one side of the cut is planar with u, v, w on the same face, then $\mathcal{P}(G) = \mathcal{P}(G')$ where G' is obtained by cleaving u, w, v according to the cut, taking the planar dual of the planar side where we consider there to be outer faces between the vertices of the cut and we glue u_1 to u_2, w_1 to w_2, and v_1 to v_2 as shown in figure 2.33.*

It is not always the case that the Feynman period converges. There is a particular symmetry where the Feynman period diverges but the graph has a subgraph, known as a sub-divergence, which does not diverge after renormalization. In this particular symmetry, discussed in remark 2.3.8, we have a decomplete graph, which has four half edges. These four half edges represent two particles entering and exiting the graph which represents their interaction. We may complete such a graph by adding a new vertex adjacent to these four vertices, sometimes referred to as the point at infinity. See figure 2.30.

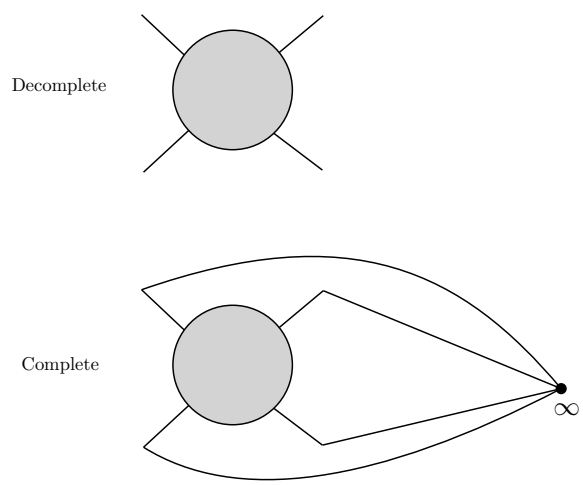


Figure 2.30: A decomplete graph and the associated complete graph obtained by adding a point at infinity.

Remark 2.3.8. [104]

If H has a 4-edge or fewer edge cut then G contains a sub-divergence, meaning the period diverges. However, we may take H_1 and H_2 to be obtained from H by replacing the graph on the side of the cut without the sub-divergence by a single vertex adjacent to the vertices incident to the cut to get G_1 and we may get G_2 by deleting the side of the cut containing the sub-divergence leaving half edges where the cut was. Then after renormalization $\mathcal{P}_G^{ren} = \mathcal{P}_{G_1}^{ren}\mathcal{P}_{G_2}^{ren} + \ell$ where ℓ is some lower order terms. While this integral diverges before renormalization, any invariant on the period should have some property on this sub-divergence. See figure 2.34.

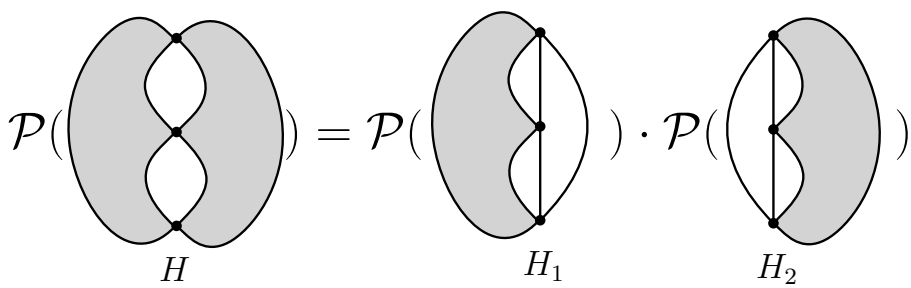


Figure 2.31: A Feynman period symmetry for a 3 vertex separator

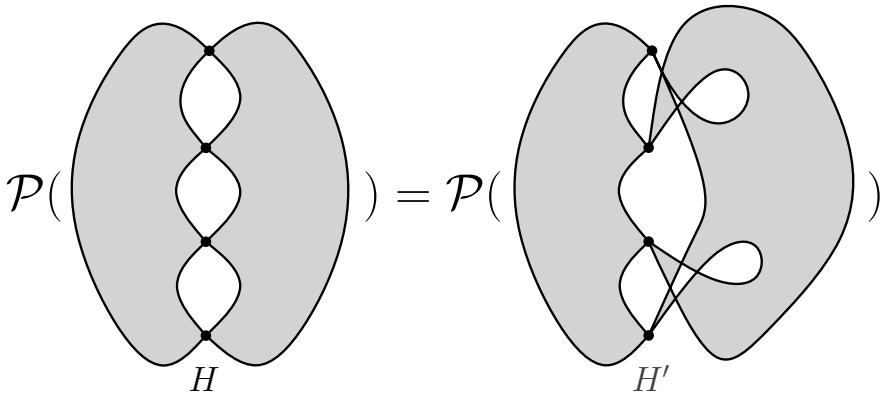


Figure 2.32: A Feynman period symmetry for a 4 vertex separator

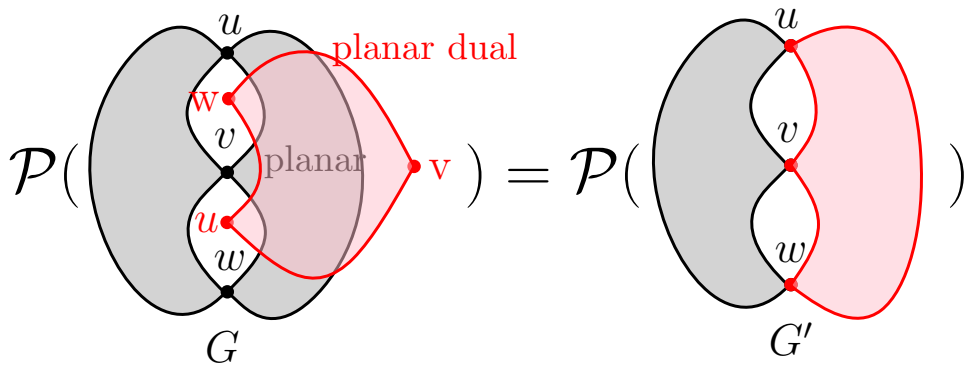


Figure 2.33: A Feynman period symmetry for a 3 vertex separator with one side being planar

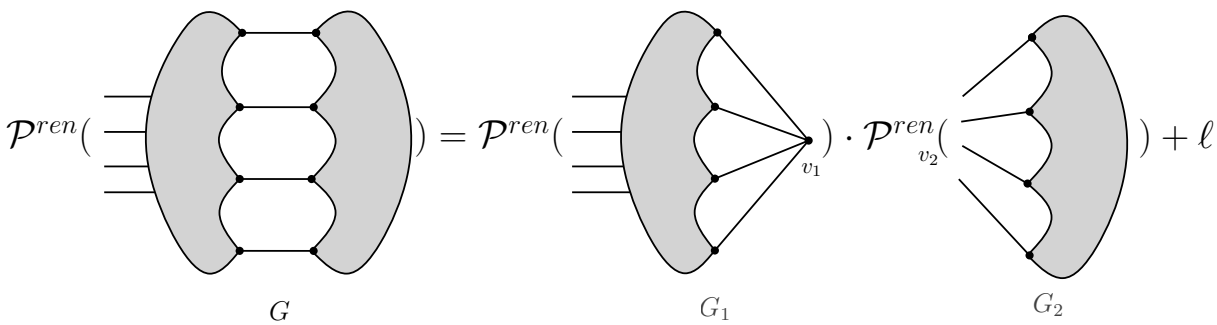


Figure 2.34: A Feynman period symmetry for a 4-edge cut

The Feynman symmetries inspired a search for graph invariants that satisfy these symmetries in an effort to identify graphs which have the same Feynman period. It has been found that some invariants satisfying these symmetries also satisfy the recurrence relation that the Martin polynomial possesses, like the Martin invariant from [96] which is a graph invariant that satisfies these symmetries. For a 4-regular graph G the *Martin invariant* is

$$M(G) = \frac{1}{6} m'(G, 0). \tag{2.6}$$

While we only use the 4-regular Martin invariant here, the Martin invariant is more generally defined as $M(G) = \frac{4(-1)^k}{(k-2)!(2k)!} m'(G, 4-2k)$ for $2k$-regular graphs. This invariant satisfies the Feynman period symmetries and it has been conjectured that two graphs have

the same period if and only if they have the same Martin sequence $M(G^*)$ [96] where the Martin sequence is obtained from $\{M(G^k)\}_{k=1}$ where G^k is found by replacing each edge of G with k copies of that edge. Additionally, for the Feynman symmetry involving the sub-divergence, the Martin invariant satisfies the property that if G has a k-edge cut then $M(G) = k!M(G_1)M(G_2)$ where M_1 and M_2 are obtained by contracting one side of the cut to a single vertex [96]. We call this the k-cut symmetry. See figure 2.35 for an illustration in the 4-edge cut case.

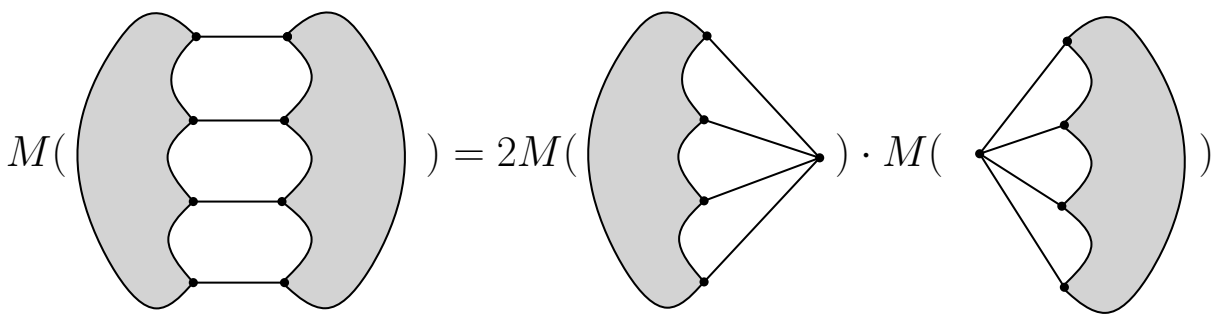


Figure 2.35: The Martin invariant 4-cut symmetry

The Martin polynomial has a close relationship with the interlace polynomial discussed in the next section. It is this relationship and the importance of the Martin invariant to the study of Feynman periods that inspires the results in chapter 4.

2.4 Interlace Polynomial

The interlace polynomial can be computed on any *simple looped* graph which is a graph allowing loops but no multi-edges. When the interlace polynomial is computed on a class of graphs known as interlace graphs, which come from chord diagrams of 4-regular graphs, then the interlace polynomial on the interlace graph is equal to the Martin polynomial on the 4-regular graph. It is this relationship that inspires us to investigate in chapters 4 and 5 how the interlace polynomial might relate to the Feynman period symmetries and the Martin invariant. We begin this section with some background on graph matrices, followed by the interlace polynomial, interlace graphs and chord diagrams, and finally the relationship between the interlace polynomial and the Martin polynomial.

The *adjacency matrix* $\mathbf{A}_G$ of a graph G is the $V \times V$ matrix whose (u, w)-entry is the number of adjacencies between vertex u and vertex w. The *degree matrix* of a graph G is

the $V \times V$ diagonal matrix whose (v, v)-entry is the number of incidences $i \in I$ such that $\varsigma(i) = v$. Shown below are the degree and adjacency matrices for the graph in figure 2.2. More information about graph matrices, including the incidence and Laplacian matrices, can be found in [9] which provides a comprehensive source of information on the relationship between linear algebra and graph theory.

$$\mathbf{D} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Let $n(M)$ be the *nullity* of the matrix M taken over $GF(2)$ and $M[X]$ be the matrix M restricted to the rows and columns index by X. Define D_Y to be a diagonal matrix with $D_{ii} = 1$ if $i \in Y$ and zero otherwise.

Definition 2.4.1. *[7] Let $G = (V, E)$ be a simple looped graph. The* interlace polynomial of G *is* $Q(G; y) = \sum_{X \subseteq V} \sum_{Y \subseteq X} (y-2)^{n(A(G)[X]+D_Y[X])}$.

The interlace polynomial is invariant under the addition or removal of loops [4, 7, 42]. This is relatively straightforward from this definition of the interlace polynomial as for any choice of loops in G we sum over all vertex subsets and all ways to toggle loops within each subset of vertices. Thus regardless of the initial choice of loops in G we will sum over all possible choice of loops. This can be clearly seen in table 2.3. Thus we will generally only consider simple graphs when working solely with the interlace polynomial.

Example 2.4.2. We compute the interlace polynomial of K_3 in detail using definition 2.4.1. This method is convenient to compute via computer, but quite slow to compute by hand. As seen in table 2.3 we simplify the calculation by grouping our choices of X and Y by isomorphic graphs. We then compute $k = n(A(G)[X] + D_Y[X])$ which is the nullity of the adjacency matrix of the the graph induced by X with loops added at vertices in Y.

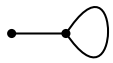
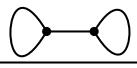
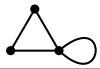
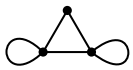
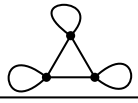
X	Y	$G[X]$ with looped Y	$k = n(A(G)[X] + D_Y[X])$	$(y-2)^k$	# of sets
$\emptyset$	$\emptyset$		0	$(y-2)^0$	1
any vertex	$\emptyset$		1	$(y-2)^1$	3
any vertex	any vertex		0	$(y-2)^0$	3
any pair	$\emptyset$		0	$(y-2)^0$	3
any pair	any vertex		0	$(y-2)^0$	6
any pair	all vertices		1	$(y-2)^1$	3
all vertices	$\emptyset$		1	$(y-2)^1$	1
all vertices	any vertex		0	$(y-2)^0$	3
all vertices	any pair		1	$(y-2)^1$	3
all vertices	all vertices		2	$(y-2)^2$	1

Table 2.3: The interlace polynomial calculated via matrices of a graph

We can now compute the interlace polynomial from this table and we get that $Q(K_3; y) = (y-2)^2 + 10(y-2) + 16(y-2)^0 = y^2 + 6y$.

It will also be useful to us to have a recursive definition. In order to present the recursive definition of the interlace polynomial, we begin by defining some local complementation operations; first for vertices, then for edges. These operations are between sets of vertices. When we complement edges between sets of vertices A and B in a graph G, we mean that for each vertex $v \in A$ (resp. B) in we remove all edges to neighbors of v in B (resp. A) and to add edges to all non-neighbors in B (resp. A). See figure 2.36 for an example of the operations.

For a simple looped graph G and any vertex v of G, the *vertex-local complement* of G at v, denoted by $G * v$ is the graph obtained by taking the complement of the edges in the subgraph induced by the closed neighborhood $\overline{N(v)}$. For a simple graph $G = (V, E)$, let $e = uv$. The *edge-local complementation* (also known as the pivot) of G at e, denoted by $G * e$ is defined as follows. Take $V_1 = \overline{N_G(v)} \setminus \overline{N_G(u)}$, $V_2 = \overline{N_G(u)} \setminus \overline{N_G(v)}$, and

$V_3 = \overline{N_G(u)} \cap \overline{N_G(v)}$. Then $G * e$ is obtained by complementing the edges between each pair of distinct V_i in the subgraph induced on $V_1 \cup V_2 \cup V_3$. So for every $v_i \in V_i$ and $v_j \in V_j$ $e' = v_i v_j \in G * e$ if and only if $e' \notin G$ and all other edges of G remain are unchanged.

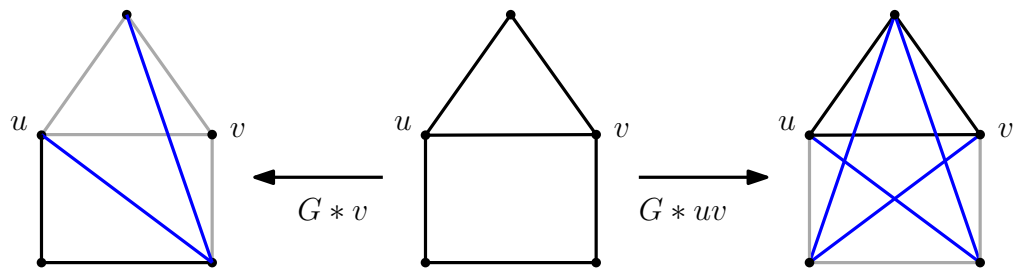


Figure 2.36: The local complementation operations on v and $e = uv$

The recursive definition of the interlace polynomial uses these two complementation operations and vertex deletion.

Definition 2.4.3. *[7] For a simple looped graph $G = (V, E)$ the following recursive relation defines the* interlace polynomial $Q(G; y)$.
1. If $V = \varnothing$ then $Q(G; y) = 1$
2. If v is an isolated vertex of G then $Q(G; y) = yQ(G \setminus v; y)$
3. If $e = vu$ for $u, v \in V(G)$ then $Q(G; y) = Q(G \setminus v; y) + Q((G * e) \setminus v; y) + Q((G * v) \setminus v; y)$

It was proved in [7] that these definitions are equivalent.

Example 2.4.4. We may also compute the interlace polynomial of K_3 recursively. As seen in figure 2.37 this computation is much faster by hand and still yield the same polynomial.

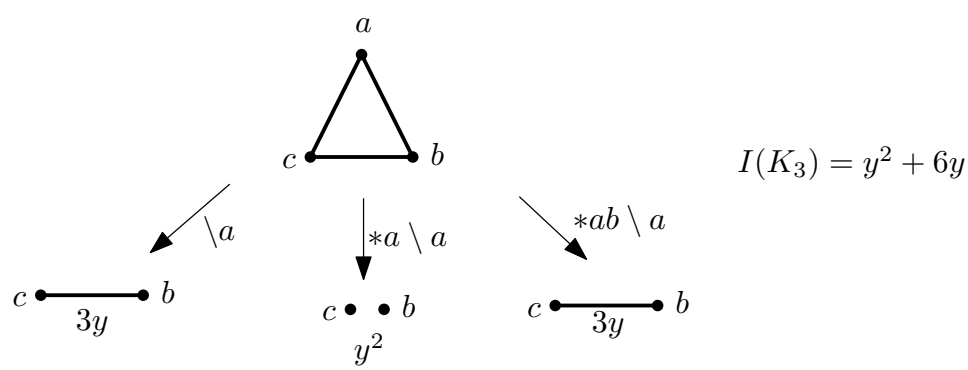


Figure 2.37: The interlace polynomial of K_3 computed recursively

The definitions of local vertex and edge complementation and the interlace polynomial definitions presented here differ slightly from those in most of the literature including [7, 47, 113]. To align with the matrix definition, used in most literature, the recurrence definition adds loops instead of using closed neighborhoods in the complementation operations as a loop forces a vertex to be in its own neighborhood. This is, however, equivalent to closing the neighborhoods in the local complementation operations and not needing to add loops when doing the recursion. Furthermore, it is important to note that the interlace polynomial is invariant under the addition or removal of loops [4, 7, 42] and thus this notational choice has no affect on the polynomial other than to make presentation cleaner by not needing to deal with loops. Thus the definitions presented here have been reformulated to deal with loops more cleanly.

To discuss how exactly the Martin polynomial and interlace polynomial are related, we first need to define a chord diagram. For a connected 4-regular graph G choose any Eulerian circuit C. Then C visits each vertex of G exactly twice. Define the *chord diagram* of this circuit by taking a cycle with $2|V(G)|$ vertices labeled by the vertices of G in the order the circuit visits the vertices. Notice that each label appears exactly twice. We then add a chord through the cycle between vertices of the same label. To create the *interlace graph* $I(C)$ take vertex set $V(G)$ and two vertices are adjacent in $I(C)$ if and only if their chords cross in the chord diagram of C. Interlace graphs can be extended to disconnected 4-regular graphs by considering an Eulerian circuit on each component and constructing a connected component of an interlace graph for each component of the 4-regular graph. This relationship provides a pair of related graphs G and $I(C)$ for which the Martin polynomial of G equals the interlace polynomial of $I(C)$.

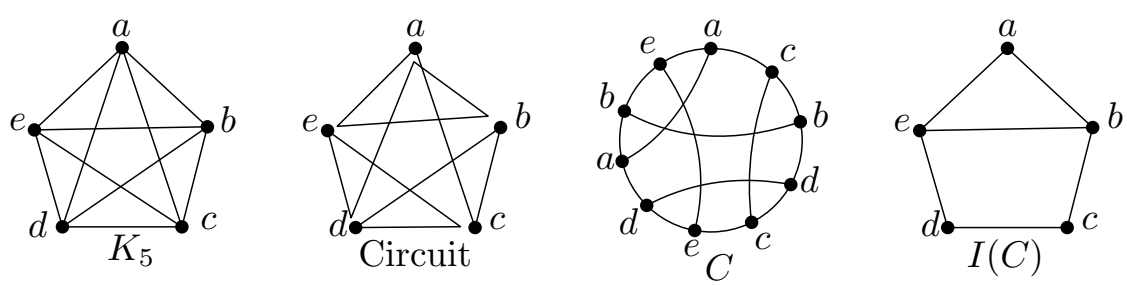


Figure 2.38: A 4-regular graph, Eulerian cycle, chord diagram and interlace graph

Theorem 2.4.5. *[4] Let G be a 4-regular graph and C any Eulerian circuit for G. Then $m(G;y) = Q(I(C);y)$.*

The graph in figure 2.28 has interlace graph K_3 for the circuit that traces the outer

and then inner triangle starting at any vertex. The interlace of this can be seen in figure 2.38 and we get $y^2 + 6y$ for both calculations.

It is the existence of theorem 2.4.5 that inspires the investigation of the Martin invariant properties as applied to the interlace polynomial, but there is the question of when a graph could be the interlace graph for some chord diagram of a 4-regular graph. Fortunately, these have been completely characterized by forbidden vertex minors in [21, 88] and the edge-local complement obstructions in [54]. There are three forbidden vertex minors, shown in figure 2.39 and fifteen forbidden edge-local complementation obstructions, shown in figure 2.40 . A *vertex minor* is any subgraph that can be obtained via a sequence of local complementations and vertex deletions.

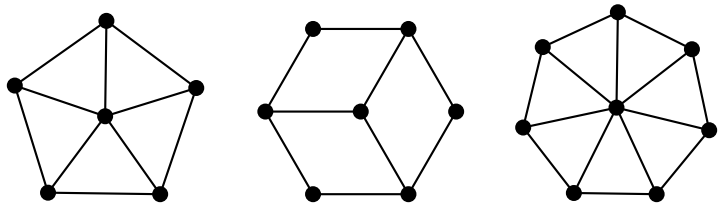


Figure 2.39: Forbidden interlace graph vertex minors

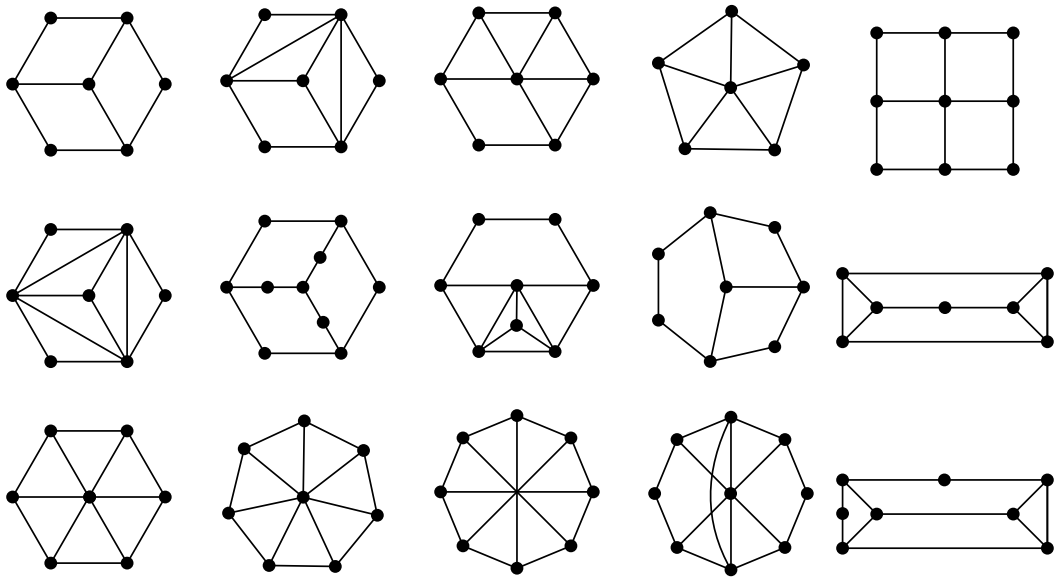


Figure 2.40: Interlace graph edge-local complementation obstructions

In chapter 4 we will construct the interlace invariant based on the Martin invariant and use the close relationship between the Martin and interlace polynomials to look at what results we have been able to extend from the Martin invariant to the interlace invariant. While these results were constructed by examining interlace graphs and the Eulerian graphs they came from, the proofs are not dependent on not including a forbidden vertex minor and thus the results in chapter 4 are more general.

Chapter 3

Tutte-type Polynomials

In this chapter we will discuss a variety of polynomials based on the Tutte polynomial. First we will define an extended Tutte polynomial to retain deletion, contraction, and activity information. We will also define an extended Kirchhoff polynomial by simplifying these notions. The polynomials differ from the multivariate or parametrized generalizations of the Tutte polynomial summarized in [47], as each edge of the graph corresponds to multiple indeterminates but the extended Tutte polynomial here corresponds to a multivariate polynomial defined for matroids in [139]. Using these ideas we will define an extended Tutte polynomial for signed graphs and discuss how all these polynomials relate to grid walking. Grid walking is a relation between operations on each edge in the extended Tutte polynomial and steps in a rectangular grid bounded by the graph's rank and nullity. Furthermore we will investigate how these ideas are useful for bipartite representations of graphs.

3.1 An Extended Tutte Polynomial

When calculating the Tutte polynomial using deletion and contraction we only retain the number of loops and isthmi in each terminal minor. Information about the sequences of deletions and contractions used is not preserved in the traditional Tutte polynomial nor is any information about which edges were deleted, contracted, loops, or isthmi. With a view toward signed graphs, we extend the Tutte polynomial to track the order of edges used using subscripts where a variable x (resp. y) will now appear as x_e (resp. y_e) if the edge e is an isthmus (resp. a loop). Moreover, we modify the traditional Tutte polynomial to include variables c_e and d_e when an edge e is contracted or deleted, respectively. Additionally,

this polynomial is equivalent to a non-commutative Tutte polynomial whose variables are ordered by $\overrightarrow{E}$ subscripts. We will work with the polynomial using subscript labels from $\overrightarrow{E}$, often also listed in edge order, as we can simply convert to a non-commutative version of the polynomial by simply dropping the edge label subscripts. It may seem like this adds a lot of unnecessary information, but this complexity is what is going to allow us to translate between the extended Tutte polynomial of bipartite representations of a graph and the Tutte polynomial of the graph as well as extend our results to signed graphs, and potentially oriented hypergraphs as discussed in chapter 5. It is important to note that definition 3.1.1 is equivalent to what Zaslavsky defined for matroids in [139] with very similar notation. We obtained this polynomial through examining how different terminal minors corresponded to different maximal spanning forests under different edge orderings independently of Zaslavsky's paper. Zaslavsky's paper presents this work in the context of matroids and provides many valuable results on its general properties. Here this polynomial is use primarily for tracking maximal spanning forests under different edge orderings and to construct a new notion of Tutte grid walking.

Definition 3.1.1. *Given a graph G and a total ordering on the edges $\overrightarrow{E}$ define the extended Tutte polynomial recursively by deletion and contraction as follows:*

$$T_G(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}) = \begin{cases} 1 & \text{if } E = \varnothing, \\ x_e \cdot T_{G/e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}-\{e\}) & \text{if } e \text{ is an isthmus in } G, \\ y_e \cdot T_{G\backslash e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}-\{e\}) & \text{if } e \text{ is a loop in } G, \\ d_e \cdot T_{G\backslash e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}-\{e\}) & \text{if } e \text{ is neither isthmus} \\ \quad +c_e \cdot T_{G/e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}-\{e\}) & \quad \text{nor loop in } G, \end{cases}$$

where e is the smallest edge in $\overrightarrow{E}$, and $\mathbf{x}$ consists of all subscripted variables of the form x_e, and similarly for $\mathbf{y}, \mathbf{c}$, and $\mathbf{d}$.

Corollary 3.1.2. *For a graph G, $T_G(x,y) = T_G(x,y,1,1;\overrightarrow{E})$.*

Proof. Observe that setting $c_e = d_e = 1$ and $x_e = x$, $y_e = y$ for all e we immediately obtain the definition of the Tutte polynomial. ■

Example 3.1.3. The process for computing the extended Tutte polynomial via deletion and contraction is almost the same as the computation of the Tutte polynomial. In figure 3.1 we compute the extended Tutte polynomial of the graph G in figure 2.2, whose Tutte polynomial was computed by deletion and contraction in figure 2.26. We can see that the

deletion and contraction decision tree in essence looks no different than when we computed the Tutte polynomial, but we recorded the edges that are deleted and contracted in the monomial for each terminal minor. We also tracked which edges become x and y. For example following the branches of the tree from the original graph. One path is to delete e_1, then e_2 is an isthmus, we can contract e_3, contract e_4, and e_5 is a loop. This gives the monomial $d_{e_1}x_{e_2}c_{e_3}c_{e_4}y_{e_5}$. Using the deletion contraction decision tree we get

$$\begin{aligned} T_G(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}) = {} & d_1x_2d_3x_4x_5 + d_1x_2c_3d_4x_5 + d_1x_2c_3c_4y_5 + c_1d_2d_3x_4x_5 \\ & + c_1d_2c_3d_4x_5 + c_1d_2c_3c_4y_5 + c_1c_2d_3x_4y_5 + c_1c_2c_3y_4y_5 \end{aligned}$$

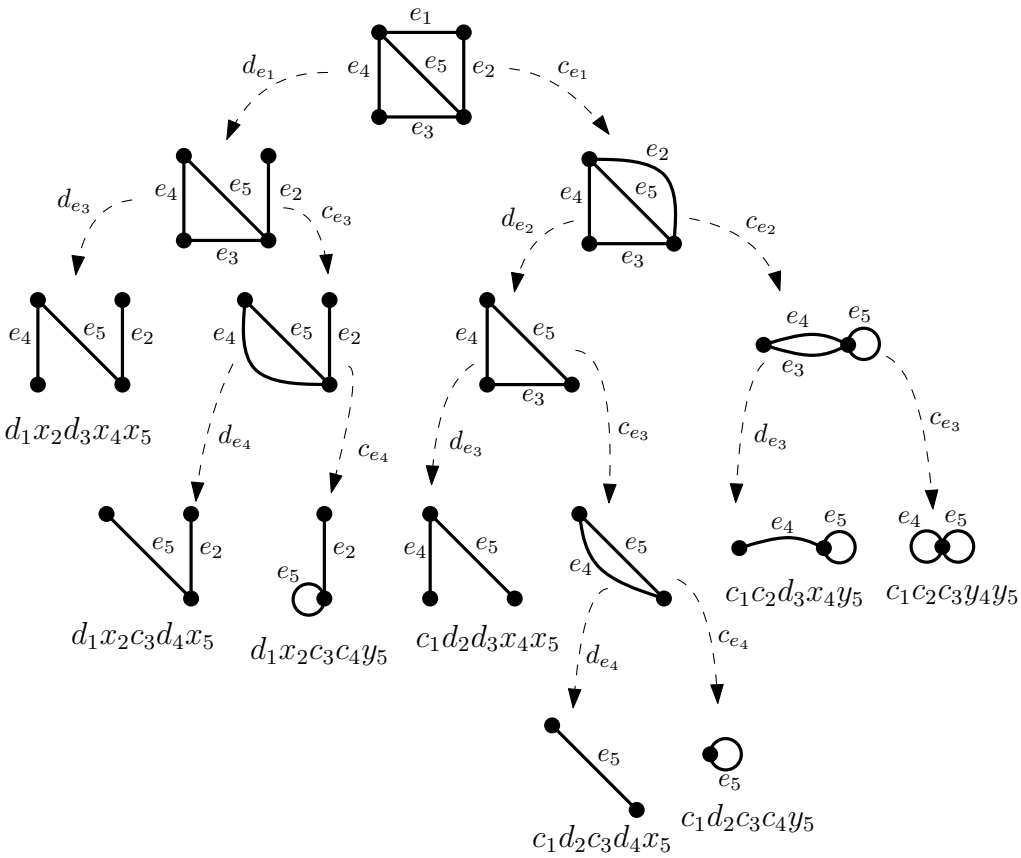


Figure 3.1: A calculation of the extended Tutte polynomial

This polynomial is clearly no longer invariant under edge ordering or the order of

deletion and contraction. While the terminal minors are invariant the associated monomial is not.

Remark 3.1.4. To consider the extended Tutte polynomial as a noncommutative polynomial we fix a total ordering on the edges $\overrightarrow{E}$, and compute the extended Tutte polynomial by addressing each edge in the order of $\overrightarrow{E}$ starting with the smallest edge. We don't have an indeterminate for each edge, we simply have $\{x, y, c, d\}$ and we do not allow them to commute. We will denote the noncommutative polynomial by $T_G(x, y, c, d; \overrightarrow{E})$ as we now have a single indeterminate of each type as opposed to one for each edge. Considering the extended Tutte polynomial computed in example 3.1.3 we can take $\overrightarrow{E}$ to be the total ordering on the edges and we get the non-commutative extended Tutte polynomial by simply dropping the subscripts in the monomials as we wrote the indeterminates in edge ordering. This gives

$$\begin{aligned} T_G(x, y, c, d; \overrightarrow{E}) = {} & dxdxx + dxcdx + dxccy + cddxx \\ & + cdcdx + cdccy + ccdxy + cccyy \end{aligned}$$

As we do not allow the indeterminates to commute $dxcdx$ is distinct from $cddxx$. We also still know which edge corresponds to each indeterminate in each monomial as they appear in the ordering of $\overrightarrow{E}$. For examples it is often convenient to have the edge ordering labels explicitly written and for purposes of the examples and general notation we will consider the edges to be labeled with $\{1, ..., n\}$ and the total edge ordering to also be $\{1, ..., n\}$ unless specifically stated otherwise. This perspective will inform how we think about reordering the edges and the affect this has on the polynomial and Tutte grid walks.

It is a simple observation that the number of terms in the extended Tutte polynomial of a graph G is the same as the number of terminal minors in Tutte polynomial of G from corollary 3.1.2 and it is known from [118] that the number of terms is equal to that of the number of maximal spanning forests as mentioned in chapter 1. We make a further observation on this connection.

Lemma 3.1.5. *Given the extended Tutte polynomial of a graph G the set of edges associated to x or c in a given monomial form a maximal spanning forest. Moreover each monomial produces a different maximal spanning forest and all maximal spanning forests of G are produced by some monomial.*

Proof. We prove this result by constructing a bijection f from the set of monomials of $T_G(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}; \overrightarrow{E})$ to the maximal spanning forests of G for any edge ordering $\overrightarrow{E}$. Let m be

a monomial from $T_G(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}; \overrightarrow{E})$. Then $f(m) = T_m$ is the subgraph induced by the set of all edges assigned to x or c. We will show T_m is a maximal spanning forest first. To see that T_m is a maximal spanning forest we will prove that T_m contains no cycle and $T_m + e$ contains a cycle for every $e \notin E(T_m)$.

For the sake of contradiction, suppose T_m contains a cycle C. Since C is in T_m, m contains all edge of C as subscripts to x or c. Since these are determined in the monomial by isthmi or contraction, every edge in the cycle was an isthmus or contracted at some stage of the polynomial calculation. Since no edge in a cycle can be an isthmus unless the cycle is broken, and contraction never breaks a cycle, we must have that every edge in C is contracted. But then when all but one edge of C has been contracted we have a loop, so some edge of C is a y. Thus no cycles exist in T_m. So T_m is a forest.

To see that T_m must be a maximal spanning forest suppose there exists and edge $e \notin E(T_m)$ such that $T_m + e$ is a forest. Since e is not in T_m and f constructs T_m from all edges in the monomial m sent to x or c, e appears as d_e or y_e in m. So e was a loop or had been deleted in the terminal minor corresponding to this monomial. If e was a loop then every edge in a cycle with e was contracted and this cycle is in T_m or e was a loop in G and is a cycle. So $T_m + e$ contains a cycle if e appeared as y_e in m. So we may assume e appears as d_e in m. However, as e was deleted in the calculation of the extended Tutte polynomial, e was never a loop or isthmus. Let $e = ab$. Since e was never an isthmus, a and b were in the same connected component of G and $G - e$ contains an ab-path. When performing deletions and contraction to find the terminal minor associated to m we may never break all ab-paths as we cannot deleted and contract isthmi. So in the terminal minor associated to m there is either a path between the vertices that a and b were contracted into(this might be the original a and b vertices), or there is a single vertex that both a and b were contracted into. If there is a path between two vertices then every edge in that path was an x and any vertices obtained via contraction have a path between them in G obtained by the contracted edges. So we have an ab-path in $G - e$ that is all x or c in m and thus $T_m + e$ has a cycle. If a and c were contracted to a single vertex then a sequence of contracted edges in $G - e$ form a path between them. Again we have a cycle in $T_m + e$. So T_m is a maximal spanning forest. Therefore f takes each monomial to a maximal spanning forest.

To see f is injective suppose we have monomials m_1 and m_2 such that $T_m = f(m_1) = f(m_2)$. Suppose that $m_1 \neq m_2$. Then they differ on at least one edge. First suppose the edge e they differ on is in T_m. Since it is in T_m for both monomials, without loss of generality we have x_e in m_1 and c_e in m_2. Since e was contracted in m_2, e is in a cycle in G. For e to show up as an x in m_1 all cycles containing e have been broken by deletion, but for e to be contracted in m_2 at least one cycle containing e still exists. So there is an edge e' that was deleted to break a cycle in m_1 that was contracted in m_2. So $f(m_1)$ does

not contain e' in the maximal spanning forest but $f(m_2)$ does. Thus f is injective.

To see f is surjective observe that f is injective and corollary 3.1.2 tells us $T_G(1,1) = T_G(1,1,1,1;\overrightarrow{E})$, so the number of monomials is equal to the number of maximal spanning forests. So f must be surjective. Hence we have a bijection between the monomials, or sequences of deletion and contraction, and the maximal spanning forests of G such that each maximal spanning forest can be uniquely determined by the edges sent to x and c. ■

Lemma 3.1.6. *Given a graph G, edge ordering $\overrightarrow{E}$, and sequence of deletions and contractions leading to a terminal minor with monomial m, an edge e is internally inactive if and only if it was contracted. Additionally, an edge e is externally inactive if and only if it was deleted with respect to the maximal spanning forest formed by the edges of G sent to x and c in m.*

Proof. From lemma 3.1.5 we know that given a terminal minor the set of contracted edges and isthmi form a maximal spanning forest T_m. Given this maximal spanning forest and the edge ordering $\overrightarrow{E}$, if an edge $e = ab$ is contracted then it is in the maximal spanning forest and there is a larger edge in the edge ordering that is in the separator of a and b by the way we use the edge ordering in the construction of the extended Tutte polynomial. Hence e is internally inactive. If e was deleted then it is not in the maximal spanning forest and hence external. When added to this maximal spanning forest there is some edge in the edge ordering in the cycle created by adding e that is larger in the edge ordering, otherwise e would have been an isthmus. Hence e is externally inactive.

For the other direction, suppose $e = ab$ is internally inactive with respect to T_m. Then $e \in E(T_m)$. Since e is internally inactive there exists an edge $d \notin E(T_m)$ that is larger than e in the fundamental bond of $T_m - e$. So when we address e in the edge order there is an ab-path not containing e. So e was deleted or contracted. Since e is in T_m, e was contracted. Now suppose $e = ab$ is externally inactive with respect to T_m. Then e is not maximal in the fundamental cycle in $T_m + e$. Since e is not maximal in a cycle and all other edges of this cycle are in T_m, when computing the monomial we have a cycle containing e when we examine the edge. So e is either deleted or contracted. Since e is external it is not contracted, so e is deleted. ■

This lemma gives us that x is internally active, c is internally inactive, y is externally active, and d is externally inactive. Furthermore we say a monomial or sequence of deletions and contractions is *associated to a maximal spanning forest* defined by the set of edges sent to c and x in the monomial. In figure 3.2 we see the spanning trees associated to the monomials from figure 3.1. We can see this produces the same set of spanning trees as shown in figure 2.10.

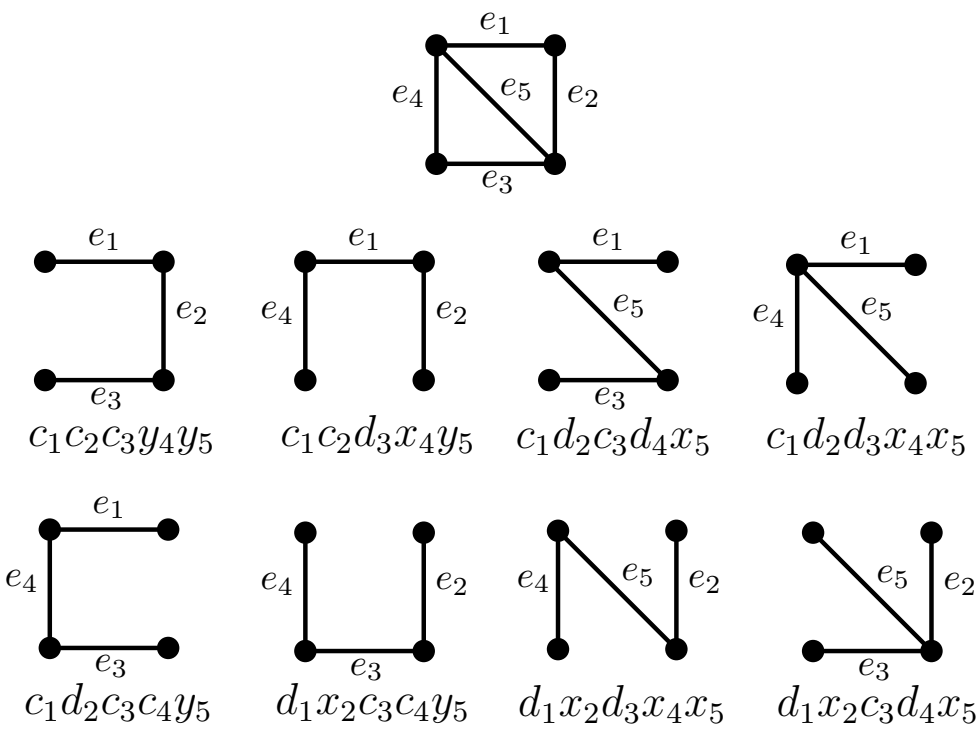


Figure 3.2: Spanning trees associated to monomials of the extended Tutte polynomial

When computing the traditional Tutte polynomial by activities we don't track all possible activity information. We only retain the number of active edges for each maximal spanning forest. We also lose which edges are inactive, active, internal, and external. We also extend the activity definition in a similar manner with c_e being internally inactive edges and d_e being externally inactive edges based upon lemma 3.1.6. We will show that this deletion-contraction extended Tutte polynomial is equivalent to the following activity extended Tutte polynomial.

When a universal edge ordering $\overrightarrow{E}$ is used to determine the order of deletions and contractions, this polynomial discerns between internally and externally active edges via the subscripts on variables x and y, while also tracking internally and externally inactive edges via the subscripts on variables c and d. Zaslavsky also discusses activities in [139] in relation to this polynomial. We still present our notation with c_e and d_e and versions of a few associated results.

Definition 3.1.7. *Let G be a graph with total edge ordering $\overrightarrow{E}$ and set of maximal spanning*

forests $\mathscr{T}$. Define the activity extended Tutte polynomial by:

$$T_G(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}; \overrightarrow{E}) = \sum_{T \in \mathscr{T}} \prod_{e \in \overrightarrow{E}} A_t(e)$$

where

$$A_T(e) = \begin{cases} x_e & \text{if } e \text{ is internally active,} \\ c_e & \text{if } e \text{ is internally inactive,} \\ y_e & \text{if } e \text{ is externally active,} \\ d_e & \text{if } e \text{ is externally inactive,} \end{cases}$$

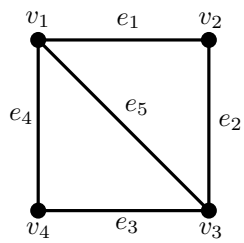
where all activities are calculated with respect to maximal spanning forest T and edge order $\overrightarrow{E}$.

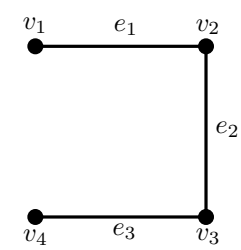
Lemma 3.1.8. *The activity and deletion-contraction extended Tutte polynomial definitions are equivalent.*

Proof. For purposes of this proof, let $T_G = T_G(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}; \overrightarrow{E})$ be the extended Tutte polynomial in definition 3.1.1 and $T'_G = T'_G(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}; \overrightarrow{E})$ the extended Tutte polynomial in definition 3.1.7. Consider a monomial m in T_G and associated maximal spanning forest T_m. By lemma 3.1.5 and lemma 3.1.6 we see that edges sent to x and c are internal, edges sent to y and d are external, edges sent to c and d are inactive and edges sent to x and y are active. Thus the activities with respect to the maximal spanning forest defined by x and c edges give the same results as applying A_T to the edge set. Thus m is a monomial of T'_G. Taking a monomial m' of T'_G we see that c and x define a maximal spanning forest by construction and by lemma 3.1.6 the inactive edges provide a sequence of deletions and contraction. Given the sequence of deletions and contraction and the maximal spanning forest, the isthmi and loops are uniquely determined by x for the isthmi and y for the loops. So m' is a monomial of T_G obtained from the specified sequence of deletions and contractions. ■

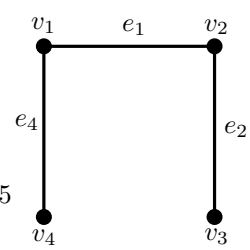
Now that we have this activity definition we present an alternate proof of corollary 3.1.2.

Proof. Observe that $\prod_{e \in \overrightarrow{E}} A_T(e)$ when evaluated at $(x, y, 1, 1)$ will yield x for all internally active edges and y for all externally active edges. The removal of subscripts means the product evaluates to $x^{f_i} y^{f_e}$ where f_i is the number of internally active edges and f_e is the number of externally active edges and the result holds. ■

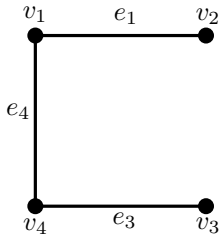




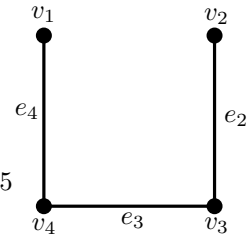
Internal		External	
e_1	c_1	e_4	y_4
e_2	c_2	e_5	y_5
e_3	c_3	$c_1c_2c_2y_4y_5$	



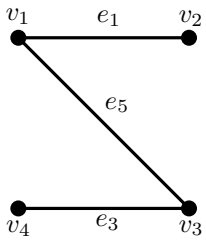
Internal		External	
e_1	c_1	e_3	d_3
e_2	c_2	e_5	y_5
e_4	x_4	$c_1c_2d_3x_4y_5$	



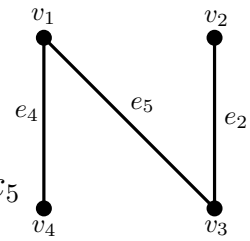
e_1	c_1	e_2	d_2
e_3	c_3	e_5	y_5
e_4	c_4	$c_1d_2c_3c_4y_5$	



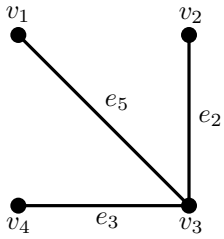
e_2	x_2	e_1	d_1
e_3	c_3	e_5	y_5
e_4	c_4	$d_1x_2c_3c_4y_5$	



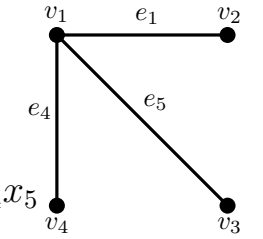
e_1	c_1	e_2	d_2
e_3	c_3	e_4	d_4
e_5	x_5	$c_1d_2c_3d_4x_5$	



e_2	x_2	e_1	d_1
e_4	x_4	e_3	d_3
e_5	x_5	$d_1x_2d_3x_4x_5$	



e_2	x_2	e_1	d_1
e_3	c_3	e_4	d_4
e_5	x_5	$d_1x_2c_3d_4x_5$	



e_1	c_1	e_2	d_2
e_4	x_4	e_3	d_3
e_5	x_5	$c_1d_2d_3x_4x_5$	

Figure 3.3: Computation of activities for the extended Tutte polynomial

Example 3.1.9. The deletion-contraction decision tree from figure 2.9 can now be used to obtain the extended Tutte polynomial (with respect to $\overrightarrow{E}$) shown in figure 3.1, which was

$$\begin{aligned} T_G(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}) = {} & d_{e_1}x_{e_2}d_{e_3}x_{e_4}x_{e_5} + d_{e_1}x_{e_2}c_{e_3}d_{e_4}x_{e_5} + d_{e_1}x_{e_2}c_{e_3}c_{e_4}y_{e_5} + c_{e_1}d_{e_2}d_{e_3}x_{e_4}x_{e_5} \\ & + c_{e_1}d_{e_2}c_{e_3}d_{e_4}x_{e_5} + c_{e_1}d_{e_2}c_{e_3}c_{e_4}y_{e_5} + c_{e_1}c_{e_2}d_{e_3}x_{e_4}y_{e_5} + c_{e_1}c_{e_2}c_{e_3}y_{e_4}y_{e_5}. \end{aligned}$$

Observe that the x and y variables correspond to the internally and externally active edges of their subscripts shown in figure 3.3. The removal of the subscripts produces a non-commutative extended Tutte polynomial as discussed in remark 3.1.4 with variables appearing in the assigned edge ordering $\overrightarrow{E}$. Thus, the extended Tutte polynomial can be written more simply as

$$\begin{aligned} T_G(x,y,c,d;\overrightarrow{E}) = {} & dxdxx + dxcdx + dxccy + cddxx + cdcdx \\ & + cdccy + ccdxy + cccyy. \end{aligned}$$

While this non-commutativity does not appear meaningfully different from the edge labels it informs a how we thinks about grid walking as shown in section 3.3.

Corollary 3.1.10. *For a graph G and each term of $T_G(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E})$, the degree of x and c in the monomial is the size of a maximal spanning forest of G, and the degree of y and d in the monomial is the cyclomatic number of G.*

Proof. From the proof of Theorem 3.1.8 every edge associated to an x or c is internal regardless of activity. Similarly, every edge associated to a y or d is external, and the number of edges outside a maximal spanning forest is the cyclomatic number. This also follows trivially from lemma 3.1.5 ■

Given that we know each monomial of the extended Tutte polynomial or a graph G corresponds to a unique maximal spanning forest of G a natural question is if it is possible to convert between the monomials as one would convert between the maximal spanning forests by adding and removing edges. It seems likely that we can swap between monomials using replacements of consecutive edges provided the edges are in the same cycle, but having an appropriately chosen edge order seems necessary. For example, if we contract a cycle fully to a loop and the last contracted edge and loop are consecutive in the edge ordering we have $c_i y_{i+1}$ in the monomial, we can replace this with $d_i x_{i+1}$ to get another valid monomial. However, identifying where in the monomial this occurs can be unclear and it is an open question how to swap indeterminates to move between monomials in general.

3.2 An Evaluation of the Kirchhoff Polynomial

While having all the deletion, contraction, and activity information is interesting, we want to consider in more detail the maximal spanning forests of a graph. We now consider a simpler extension of the Kirchhoff polynomial based on the maximal spanning forests of a graph. Note that if we choose an edge ordering then the following polynomial can also be written as a non-commutative polynomial by removing edge subscripts and requiring the indeterminates to be placed in edge order as described in remark 3.1.4. In this section we will use an edge labeling, which we will also call $\overrightarrow{E}$ so that we may think of it as an edge ordering when convenient.

The *Kirchhoff polynomial* of a graph G with set of maximal spanning forests $\mathcal{T}$ is $\psi_G(\mathbf{t}) = \sum_{T\in\mathcal{T}} \prod_{e\in T} t_e$. This polynomial provides for each maximal spanning forest a monomial that is the product of the edges in that maximal spanning forest.

Definition 3.2.1. *Let G be a graph and $\overrightarrow{E}$ a labeling on the edges. Let $\mathscr{T}$ be the family of maximal spanning forests of G. For each $T \in \mathscr{T}$ let*

$$\alpha_T(e) = \begin{cases} t_e & \text{if } e \in T \\ z_e & \text{if } e \notin T \end{cases}$$

then,

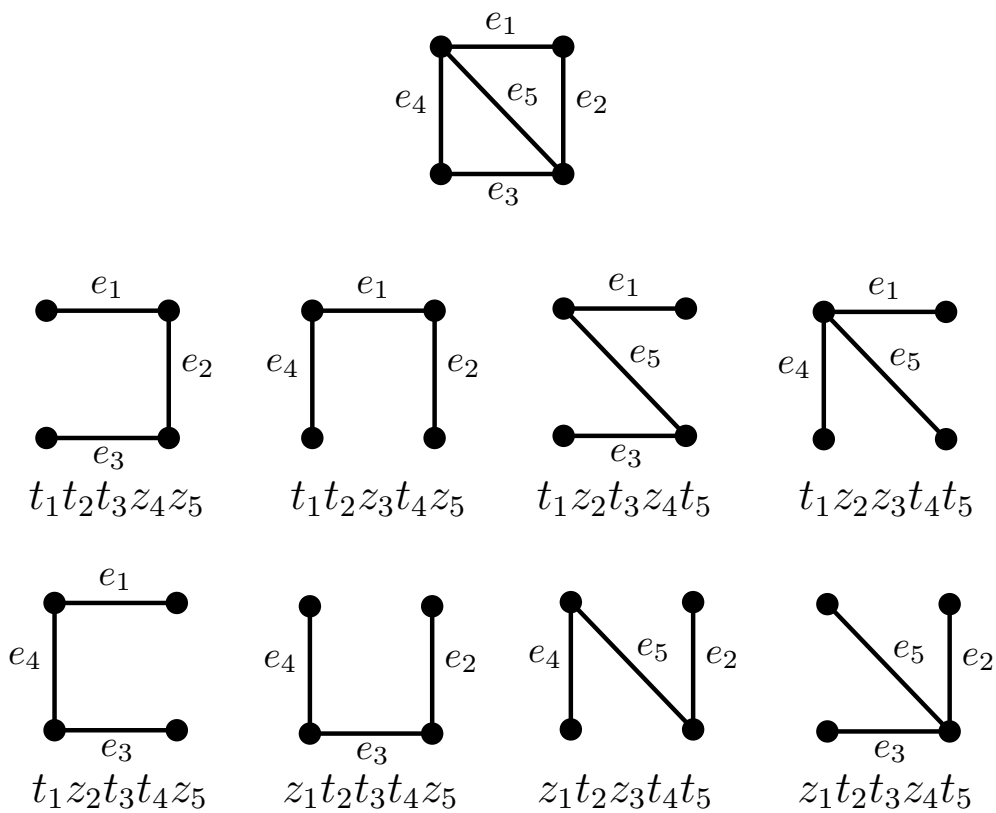
$$J(\mathbf{t}, \mathbf{z}; \overrightarrow{E}) = \sum_{T\in\mathscr{T}} \prod_{e\in\overrightarrow{E}} \alpha_T(e)$$

One thing we wanted to investigate with the extended Tutte polynomial was how different maximal spanning forests were treated by different edge orders. This other polynomial directly shows how changing the edge order shifts the maximal spanning forest labels within the monomials.

Lemma 3.2.2. *For a graph G with edge labeling $\overrightarrow{E}$, $(t_1...t_n)\psi_G(\frac{z_1}{t_1}, ..., \frac{z_n}{t_n}) = J(\mathbf{t}, \mathbf{z}; \overrightarrow{E})$.*

Proof. Observe $(t_1...t_n)\psi_G(\frac{z_1}{t_1}, ..., \frac{z_n}{t_n}) = (t_1...t_n)\sum_{T\in\mathcal{T}} \prod_{e\in T} \frac{z_e}{t_e} = \sum_{T\in\mathcal{T}} \prod_{e\notin T} z_e \prod_{e\in T} t_e = J(\mathbf{z}, \mathbf{t}; \overrightarrow{E})$. ■

This dual Kirchhoff used in defining the Feynman period can also be used to obtain $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E})$ by taking $(z_1...z_n)\tilde{\psi}_G(\frac{t_1}{z_1}, ..., \frac{t_n}{z_n})$ with an analogous proof.



$$\begin{aligned} J_G(\mathbf{t},\mathbf{z};\overrightarrow{E}) = {} & t_1t_2t_3z_4z_5 + t_1t_2z_3t_4z_5 + t_1z_2t_3z_4t_5 \\ & + t_1z_2z_3t_4t_5 + t_1z_2t_3t_4z_5 + z_1t_2t_3t_4z_5 \\ & + z_1t_2z_3t_4t_5 + z_1t_2t_3z_4t_5 \end{aligned}$$

Figure 3.4: The computation of $J(\mathbf{t},\mathbf{z};\overrightarrow{E})$

This relates to the extended Tutte polynomial through a direct variable substitution.

Lemma 3.2.3. $J(\mathbf{t},\mathbf{z};\overrightarrow{E}) = T_G(\mathbf{t},\mathbf{z},\mathbf{t},\mathbf{z};\overrightarrow{E})$.

Proof. This follows directly from lemma 3.1.5 as given an edge labeling $\{1,..,n\}$ we consider it an edge ordering and each monomial has the set of $\mathbf{x}$ and $\mathbf{c}$ form a maximal spanning forest. ■

Observe that each term of the polynomial has $|V(G)| - k$ instance of t_i, where k is the number of connected components, since the t_i's represent the edges in a maximal spanning forest which is a spanning tree for each connected component of the graph.

Lemma 3.2.4. *Let $\mathscr{T}$ be the set of all maximal spanning forests of a graph G. Let $T, T' \in \mathscr{T}$. $T \neq T'$ if and only if $\prod_{e \in \overrightarrow{E}} \alpha_T(e) \neq \prod_{e \in \overrightarrow{E}} \alpha_{T'}(e)$.*

Proof. Let $\overrightarrow{E}$ be a labeling on the edges of a graph G. We prove the forward implication by contrapositive. Suppose $T, T' \in \mathscr{T}$ such that $\prod_{e \in \overrightarrow{E}} \alpha_T(e) = \prod_{e \in \overrightarrow{E}} \alpha_{T'}(e)$. Then $\alpha_T(e) = \alpha_{T'}(e)$ for all $e \in \overrightarrow{E}$. Hence $E(T) = \{e \in \overrightarrow{E} | \alpha_T(e) = t_e\} = \{e \in \overrightarrow{E} | \alpha_{T'}(e) = t_e\} = E(T')$. So $T = T'$.

For the backwards implication observe that if $\prod_{e \in \overrightarrow{E}} \alpha_T(e) \neq \prod_{e \in \overrightarrow{E}} \alpha_{T'}(e)$ then there exists an $e \in \overrightarrow{E}$ such that without loss of generality $\alpha_T(e) = t_e$ and $\alpha_{T'}(e) = c_e$. Thus $e \in T$ and $e \notin T'$ so $T \neq T'$. ■

If $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E}) = J(\mathbf{t}, \mathbf{z}, \overrightarrow{E'})$ for edge labelings $\overrightarrow{E}$ and $\overrightarrow{E'}$ then we say edge labelings $\overrightarrow{E}$ and $\overrightarrow{E'}$ J-equivalent. To determine when this occurs we need the following definitions. A graph G has a *Whitney twist* if G can be expressed as disjoint graphs G_1 and G_2 by identifying the vertices $u_1 \in G_1$ to $u_2 \in G_2$, and $v_1 \in G_1$ to $v_2 \in G_2$. The *Whitney twist* of G is the graph obtained by identifying $u_1 \in G_1$ to $v_2 \in G_2$ and $u_2 \in G_2$ to $v_1 \in G_1$. The Tutte polynomial of a graph is invariant under Whitney twists [129]. When we want to determine if two edge labelings produce the same extended Tutte polynomial we can use a specific type of Whitney twist. A *cycle-preserving twist* is a sequence of Whitney twists on G that produces an isomorphic graph G' where if C is a cycle in G then there exists a cycle C' in G' on the same vertex set with the same set of edge labels, though not necessarily in the same order. So a cycle-preserving twist preserves the set of labels in all cycles. A cycle-preserving twist reorders the edge labels while still preserving the set of all labels in the graph. This produces a new edge labeling on the graph. Whitney twists are also involutions, so a cycle-preserving twist is as well. Since Whitney twists preserve the graph matroid even if a twist does not produce an isomorphic graph [129] any Whitney twist should produce the same polynomial, however as we care about comparing different edge labelings on a single graph we want to exclude Whitney twists that do not result in an isomorphic graph.

Example 3.2.5. Consider the following graph with the three labeled edge orders and the eleven unlabeled spanning trees.

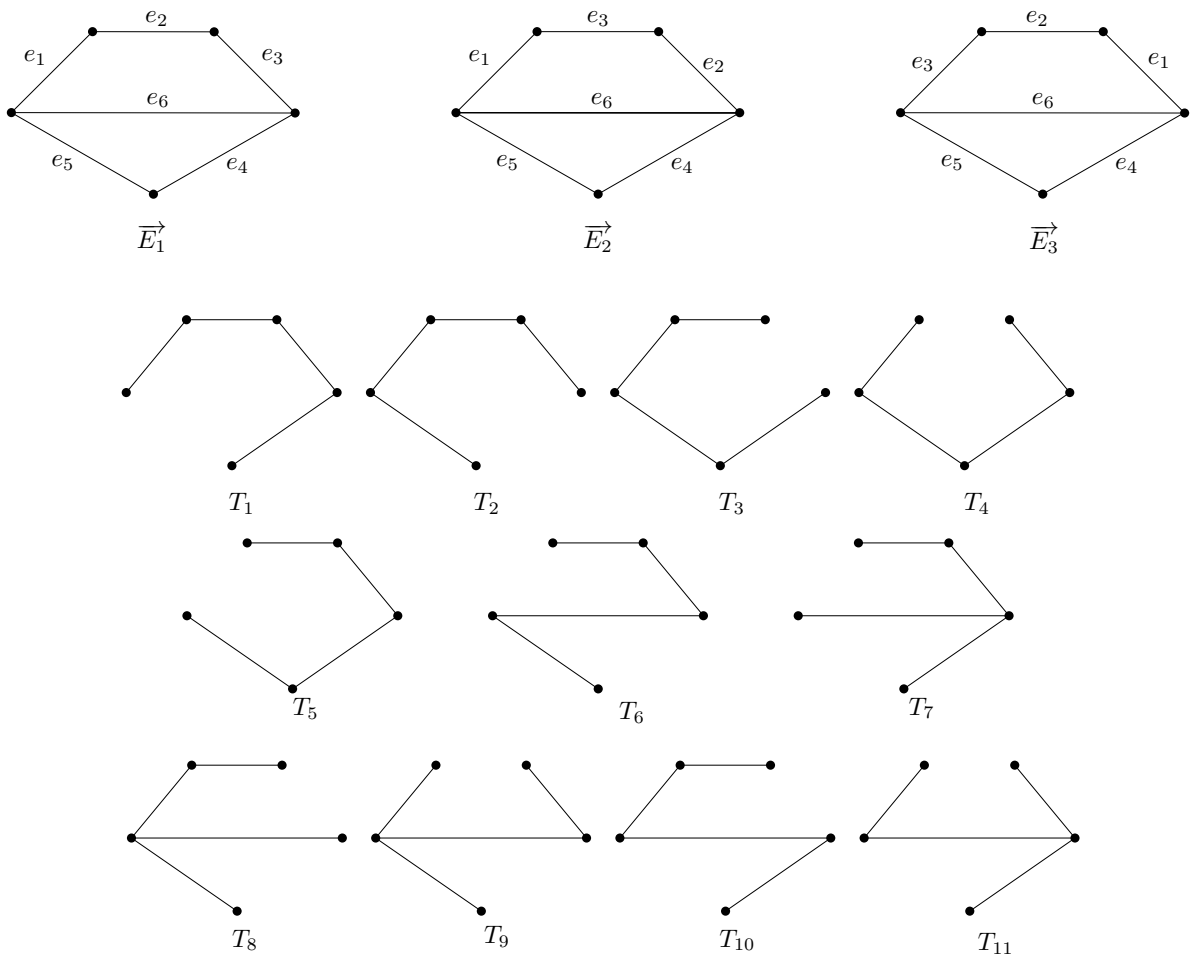


Figure 3.5: A graph with three edge labelings and its spanning trees

We calculate the $J(\mathbf{t},\mathbf{z};\overrightarrow{E})$ for all graphs.

$$\begin{aligned}
J(\mathbf{t},\mathbf{z};\overrightarrow{E_1}) &= t_1t_2t_3t_4z_5z_6 + t_1t_2t_3z_4t_5z_6 + t_1t_2z_3t_4t_5z_6 + t_1z_2t_3t_4t_5z_6 \\
&+ z_1t_2t_3t_4t_5z_6 + z_1t_2t_3z_4t_5t_6 + z_1t_2t_3t_4z_5t_6 + t_1t_2z_3z_4t_5t_6 \\
&+ t_1z_2t_3z_4t_5t_6 + t_1t_2z_3t_4z_5t_6 + t_1z_2t_3t_4z_5t_6 \\
J(\mathbf{t},\mathbf{z};\overrightarrow{E_2}) &= t_1t_2t_3t_4z_5z_6 + t_1t_2t_3z_4t_5z_6 + t_1z_2t_3t_4t_5z_6 + t_1t_2z_3t_4t_5z_6 \\
&+ z_1t_2t_3t_4t_5z_6 + z_1t_2t_3z_4t_5t_6 + z_1t_2t_3t_4z_5t_6 + t_1z_2t_3z_4t_5t_6 \\
&+ t_1t_2z_3z_4t_5t_6 + t_1z_2t_3t_4z_5t_6 + t_1t_2z_3t_4z_5t_6 \\
J(\mathbf{t},\mathbf{z};\overrightarrow{E_3}) &= t_1t_2t_3t_4z_5z_6 + t_1t_2t_3z_4t_5z_6 + z_1t_2t_3t_4t_5z_6 + t_1z_2t_3t_4t_5z_6 \\
&+ t_1t_2z_3t_4t_5z_6 + t_1t_2z_3z_4t_5t_6 + t_1t_2z_3t_4z_5t_6 + z_1t_2t_3z_4t_5t_6 \\
&+ t_1z_2t_3z_4t_5t_6 + z_1t_2t_3t_4z_5t_6 + t_1z_2t_3t_4z_5t_6
\end{aligned}$$

Each polynomial has the terms listed in the order the spanning trees are listed in. While this order is arbitrary, it helps to compare the terms, since what we are really looking at here is when the noncommutative polynomials are equal. Notice that all three polynomials are equal, though the terms appear in different orders. Examining the edge labelings one can see the cycle-preserving twists of edges e_2 and e_3 results in the edge labeling of $\overrightarrow{E_2}$ and that this is the same as in each spanning tree that contains exactly one of e_2 and e_3 having that edge replaced with the edge that is not present between e_2 and e_3. The same argument applies to the first and third edge labeling, with the exchange being between e_1 and e_3, and the cycle-preserving twists being the entire path $e_1e_2e_3$ in the first edge labeling. We can also think of the third edge labeling as performing the cycle-preserving twists e_1e_3 and then e_1e_2 on the labeling obtained after the first cycle-preserving twist.

Recall that two edge labelings $\overrightarrow{E}$ and $\overrightarrow{E'}$ are J-equivalent if $J(\mathbf{t},\mathbf{z};\overrightarrow{E}) = J(\mathbf{t},\mathbf{z};\overrightarrow{E'})$. In order to prove the equality of the two polynomials, we consider the graph as a matroid and note that each maximal spanning forest forms a basis and the set of all maximal spanning forests forms the collection of bases. We also utilize Whitney's 2-isomorphism theorem for graphs.

Theorem 3.2.6. *[129][Whitney's 2-isomorphism theorem] Two graphs G and H have the same graphic matroid if and only if G can be obtained from H by a sequence of Whitney twists, vertex cleaving, or vertex identification.*

Theorem 3.2.7. *Two edge labelings $\overrightarrow{E}$ and $\overrightarrow{E'}$ are J-equivalent if and only if $\overrightarrow{E}$ can be obtained from $\overrightarrow{E'}$ by a cycle-preserving twist.*

Proof. We begin with the forward implication. Suppose that E and $\overrightarrow{E}$ are J-equivalent. Recall that by construction each term of J and J' correspond to a maximal spanning forest of G. Since the two polynomials are equal and the sum is over all maximal spanning forests of G every maximal spanning forest has the same set of edge labels under the two edge labelings. Choose a maximal spanning forest T and $e \notin T$. Then $T + e$ contains a unique cycle and there exists $e' \in T$ such that $T + e - e'$ is also a maximal spanning forest. This maximal spanning forest must had the same set of edge labels under $\overrightarrow{E}$ and $\overrightarrow{E'}$ as the monomial is the same in both polynomials. So the set of edge labels in the cycle in $T + e - e'$ is the same for both edge labelings. Since this holds for all edges and all maximal spanning forest we conclude that the set of edges in every cycle is preserved between the edge labelings. So they have the same graphic matroid and by theorem 3.2.6 they differ by a sequence of Whitney twists as the underlying graphs are isomorphic and thus they cannot differ by vertex cleaving or vertex identification. Since the graphs are isomorphic the sequence of Whitney twists is a cycle-preserving twist.

Suppose that $\overrightarrow{E}$ and $\overrightarrow{E'}$ differ by a sequence of cycle-preserving twists. Consider a maximal spanning forest T of G and an edge $e \notin T$. Since $T + e$ contains a unique cycle and E and E' differ by a sequence of cycle-preserving twists, this cycle must have the same set of edge labels in $\overrightarrow{E}$ and $\overrightarrow{E}'$. So consider a monomial of $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E})$. We can construct a maximal spanning forest with the edges sent to t_e and for any edge e sent to z_e we can construct a cycle that is a cycle with the same set of edge labels in $\overrightarrow{E'}$. So in G with edge labeling $\overrightarrow{E'}$, T' is a maximal spanning forest with the same set of edge labels corresponding to the same monomial in $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E'})$. The reverse argument is true for any monomial in $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E'})$, so the two polynomials are equal. ■

3.3 A Grid Walking Interpretation

Each of the terms of the noncommutative $J(t, z; \overrightarrow{E})$ polynomial can be represented as a walk on a grid where the grid has the height of the cyclomatic number minus the number of loops in the graph and length equal to the rank of the graph minus the number of isthmi in the graph. Call a grid of this size the *Tutte grid* of G. A Tutte grid walk starts in the lower left corner of the grid and walks to the right upper corner by steps corresponding to the edges of G in edge order skipping all edges that are loops or isthmi in G. We move up at step i if edge i is not in the forest and moving right if edge i is in the forest. Thus each maximal spanning forest produces a walk in the grid. For each monomial of the extended Kirchhoff polynomial, the walk moves up on z_i and right on t_i at step i of the Tutte grid

walk for a maximal spanning forest with some edge labeling. We can define the grid and walks formally.

Definition 3.3.1. *For a graph G with ℓ loops, i isthmi, cyclomatic number ϕ, and rank r. Define the* Tutte grid *of G to be grid with height $\phi - \ell$ and width $r - i$ where we label the bottom left node as $t^i z^\ell$, the top left node as $t^i z^\phi$, the bottom right node as $t^r z^\ell$, and the top right node as $t^r z^\phi$. A* J grid walk *is defined to start at the $t^i y^\ell$ node and be determined by a monomial of $J(t, z; \overrightarrow{E})$, the noncommutative polynomial where the walk has steps in order $\overrightarrow{E}$. So the jth step of the walk corresponds to the jth indeterminate of a monomial. Where for the jth edge e in $\overrightarrow{E}$ we skip e if e is a loop or isthmus in G, otherwise we move right if t_e is in the monomial and up if z_e is. The walk terminates at $t^r y^\phi$ as we have $r + \phi$ edges in G and $r + \phi - i - \ell$ of them correspond to steps in the grid with $\phi - \ell$ steps up and $r - i$ steps right.*

Example 3.3.2. Using the calculation of $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E})$ from figure 3.4 we may map the J grid walks as seen in figure 3.6. We can use these J grid walks to see how we might move between the spanning trees. For example the J grid walk for the tree in red $(z_1 t_2 z_3 t_4 t_5)$ differs from the J grid walk for the tree in blue $(z_1 t_2 t_3 z_4 t_5)$ on a single pair of consecutive steps. The third and fourth step specifically are swapped. When two maximal spanning forests differ by swapping a pair of edges between the maximal spanning forests we can see this on the J grid walks as they differ by pairs of steps swapped on those edges.

Another interesting observation we can make is based on which parts of the grid are not used. For example we see in figure 3.6 that no walk ever passes through the $t^0 z^2$ corner and this is because there is no spanning tree containing neither e_1 nor e_2 and the only way to pass through $t^0 z^2$ is to choose neither e_1 nor e_2 for the spanning tree.

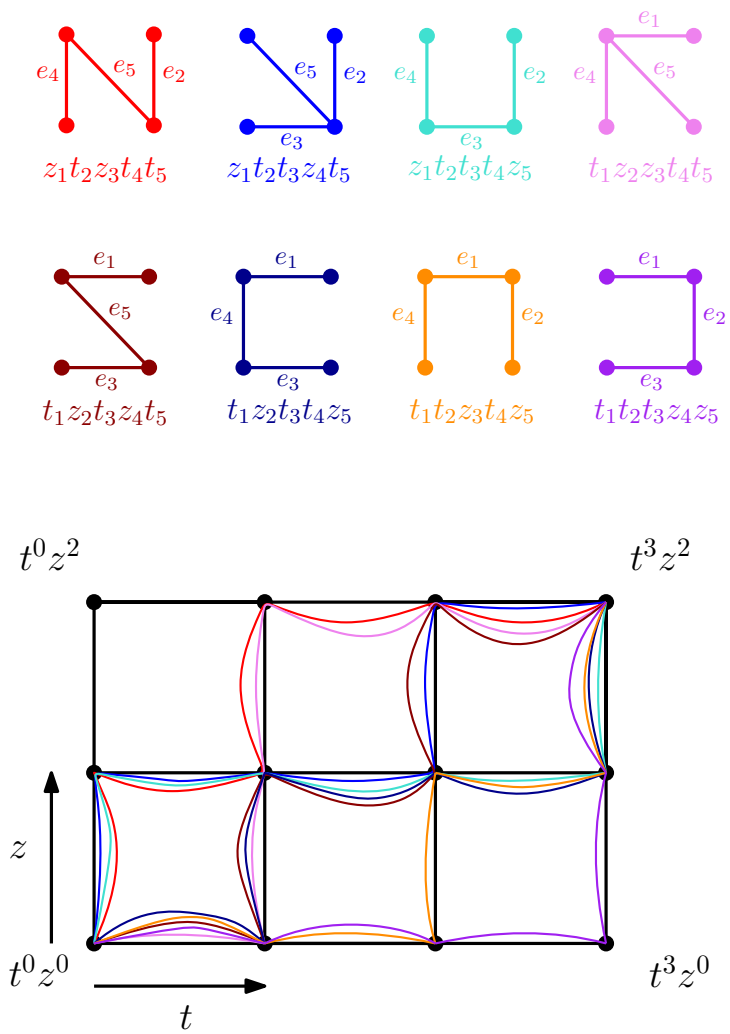


Figure 3.6: A set of J grid walks for $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E})$

Lemma 3.3.3. *Let G be a graph with edge labeling $\overrightarrow{E}$. Each maximal spanning forest of G defines a distinct J grid walk with respect to $\overrightarrow{E}$.*

Proof. To see the result we will show that if $T \neq T'$ are maximal spanning forests of G then they define different J grid walks. Consider the terms of the polynomial $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E})$ associated to T and T'. By lemma 3.2.4, $\prod_{e \in \overrightarrow{E}} \alpha_T(e) \neq \prod_{e \in \overrightarrow{E}} \alpha_{T'}(e)$. Let i be the smallest edge in the labeling for which $\alpha_T(i) \neq \alpha_{T'}(i)$. Then at the ith step of the J grid walk for T and T' one will go up and the other right. Hence the J grid walks are not the same. ■

Given a graph G and an edge ordering $\overrightarrow{E}$, each maximal spanning forest of G defines a distinct J grid walk with respect to $\overrightarrow{E}$. While each maximal spanning forest produces a distinct J grid walk, given an edge ordering not every walk on the grid corresponds to a maximal spanning forest in that edge ordering. We have shown how cycle-preserving twists in the edge ordering affect the $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E})$ polynomials, so a natural question is how twists affect the grid walking. We can think of a twist as a permutation on $\{1, ..., |E(G)|\}$ where we send the label in edge ordering $\overrightarrow{E}$ to the label of that edge in some other edge labeling $\overrightarrow{E'}$. We will call this permutation the *twist permutation from* $\overrightarrow{E}$ *to* $\overrightarrow{E'}$. Extending proposition 3.2.7 we can observe that for a graph G, if π is a twist permutation from $\overrightarrow{E}$ on G to $\overrightarrow{E'}$ on G then for each maximal spanning forest T of G the J grid walk defined by $\overrightarrow{E'}$ can be obtained by taking the J grid walk defined by $\overrightarrow{E}$ and permuting the steps by π. It is important to note that under two different edge orderings that produce equal polynomials, we may have some J grid walks that do not change under the twist permutations. See example 3.3.6. This may also occur even when the two polynomials are not equal, so we cannot use J grid walks to determine the J-equivalence of two edge orderings. See example 3.3.7.

Lemma 3.3.4. *For a graph G and edge orderings $\overrightarrow{E}$ and $\overrightarrow{E'}$ if $\overrightarrow{E'}$ differs from $\overrightarrow{E}$ by a twist permutation π then for every J grid walk associated to $\overrightarrow{E}$ the J grid walk obtained by permuting the steps according to π is a J grid walk associated to $\overrightarrow{E'}$.*

Proof. Let $|V(G)| = n$. Since $\overrightarrow{E}$ and $\overrightarrow{E'}$ differ by a twist permutation π we can consider $\overrightarrow{E}$ to be $(e_1, e_2, ...e_n)$ and and when we apply the permutation π we get the edge ordering of $\overrightarrow{E'}$. Thus given a maximal spanning forest of G the edge labels in $\overrightarrow{E}$ that correspond to edges in the maximal spanning forest get permuted to new edge labels by π, but the maximal spanning forest doesn't change. Hence when constructing the J grid walk the indices that correspond to right steps in the walk are simply permuted by π in the corresponding J grid walk in $\overrightarrow{E'}$. ■

Example 3.3.5. Consider the graph with edge ordering $\overrightarrow{E_1}$ in Figure 3.5. The following grid shows the J grid walks for the specified spanning trees.

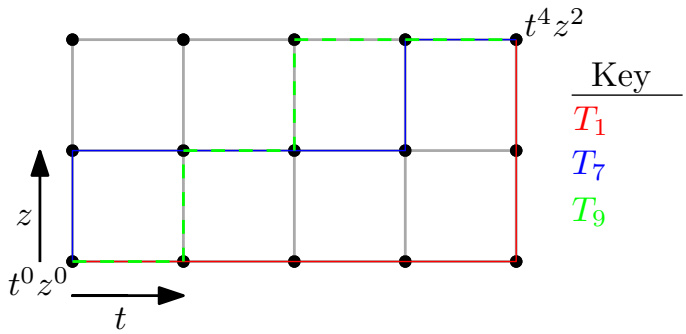


Figure 3.7: J grid walks for the given edge order and specified trees.

Each spanning tree in the edge ordering defines a distinct J grid walk from the bottom left to upper right corners.

Example 3.3.6. Taking the graph in Figure 3.5 consider the following spanning trees and their associated J grid walks under $\overrightarrow{E_1}$ and $\overrightarrow{E_2}$.

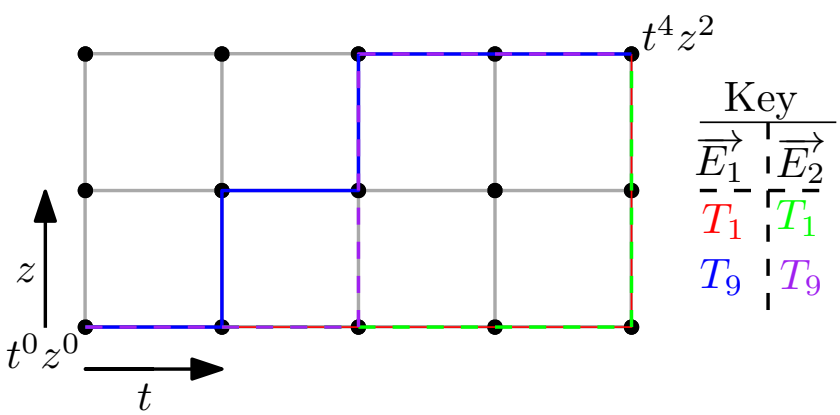


Figure 3.8: Two spanning trees and their J grid walks under two edge orderings

Observe that the twist permutation between these two edge orderings is (23) and this is also the same as permuting the 2nd and 3rd step of the J grid walks each of the two spanning trees. Furthermore it is important to note that this does not always result in a different J grid walk and with T_1 the J grid walk is the same under both edge orderings.

Example 3.3.7. Consider figure 3.9 which has two different edge labelings on the same graph.

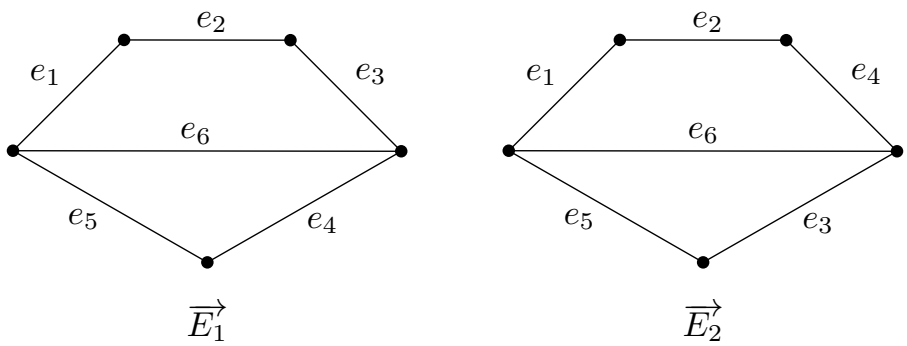


Figure 3.9: A graph with two J-inequivalent edge labelings

Both have a spanning tree that corresponds to deleting e_3 and e_4 and these produce the same J grid walks, but the polynomials are not equal as the edge labels of all cycles are not preserved.

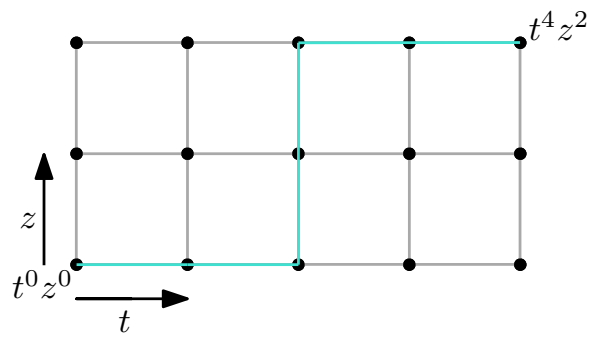


Figure 3.10: A J grid walk that is the same for two J-inequivalent edge orderings

Theorem 3.3.8. *Two edge orderings $\overrightarrow{E}$ and $\overrightarrow{E'}$ are J-equivalent if and only if the maximal spanning forests produce the same set of J grid walks under $\overrightarrow{E}$ and $\overrightarrow{E'}$.*

Proof. For the forward implication, if we know two edge ordering are J-equivalent then then $J(t, z; \overrightarrow{E}) = J(t, z; \overrightarrow{E'})$ and since the J grid walks can be defined by the monomials of the polynomial the set of J grid walks defined by these two edge orderings are the same.

For the backwards implication observe that each J grid walk corresponds to a term of the polynomial by writing the steps of the J grid walk in edge order. Since the sets of J grid walks are the same under the two edge orderings and each walk corresponds to one

maximal spanning forest and one term of the polynomial, then two polynomials are equal and hence the edge orderings are J-equivalent. ■

These results inspired us to attempt the same combinatorial interpretation with the extended Tutte polynomial. Considering each monomial in the noncommutative sense, they can be plotted through a J grid walk. We slightly modify the J grid walk definition used with the $J(t, z; \overrightarrow{E})$ polynomial.

Definition 3.3.9. *For a graph G with ℓ loops, i isthmi, cyclomatic number ϕ, and rank r. Define the* Tutte grid *of G to be grid with height $\phi - \ell$ and width $r - i$ where we label the bottom left node as $x^i y^\ell$, the top left node as $x^i y^\phi$, the bottom right node as $x^r y^\ell$, and the top right node as $x^r y^\phi$. A* Tutte grid walk *is defined by a pair of walks. One walk starts at $x^i y^\ell$ and the other at $x^r y^\phi$. We will call the walk starting at $x^i y^\ell$ the active walk and the walk starting at $x^r y^\phi$ the inactive walk. For each monomial of $T_G(x, y, c, d; \overrightarrow{E})$ we examine the edges in order of the edge ordering. Starting with the first edge e in the edge ordering if e is an isthmus or loop we skip e, otherwise if it is x in the monomial the active walk takes a right step, if it is y, the active walk takes a up step, if it is c the inactive walk takes a left steps, and if it is d the inactive walk takes a down step. We then repeat this step evaluation with the next edge in the edge ordering until all edges have been addressed.*

Lemma 3.3.10. *For a graph G with $T_G(x, y, c, d; \overrightarrow{E})$ and monomial m the active and inactive walk pair that makes up the Tutte grid walk both end on the same node of the grid.*

Proof. We know that the number of x and c steps is $r - i$ by lemma 3.1.5 and thus the number of y and d steps is $\phi - \ell$ since all edges that are not loops or isthmi of G correspond to a step in either the active or inactive walks. Suppose the monomial has k x's and j y's. Then the active walk ends on the node $x^{i+k} y^{\ell+j}$ and the inactive walk has $r - i - k$ c's and $\phi - \ell - j$ d's and so ends on the node $x^{r-(r-i-k)} y^{\phi-(\phi-\ell-j)} = x^{i+k} y^{\ell+j}$ node. Hence the two walks meet and end on the same grid node. ■

We will refer to the nodes where the active and inactive portions of the Tutte grid walks meet as the *marked nodes*. Notice that the marked nodes corresponds to the monomials in the traditional Tutte polynomial.

Consider the graph and its 4-variable Tutte polynomial from Example 3.1. Its grid is shown in figure 3.11.

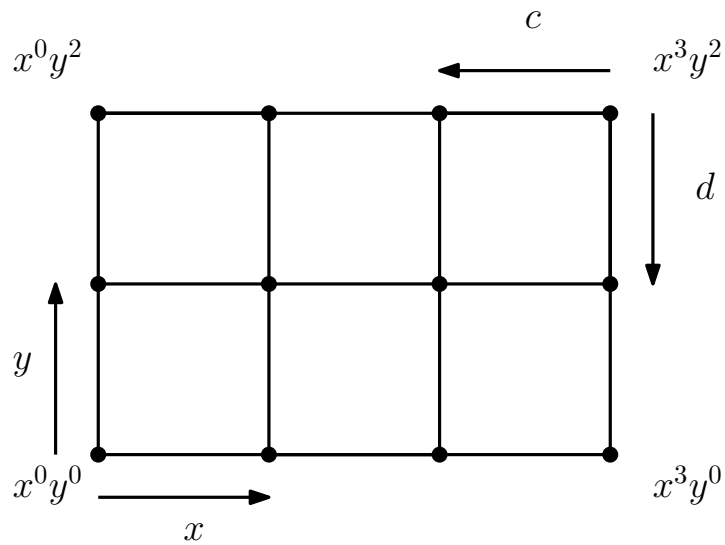


Figure 3.11: A Tutte grid

Each time an x or y appears in the non-commutative extended Tutte polynomial monomial, we walk right or up, respectively. We begin starting in the bottom left corner of the grid, which corresponds to x^iy^l. When a c or d appears, we walk left or down, respectively, starting from the top right corner, which corresponds to x^ry^ϕ. We consider a node in the grid to be T_G *marked* if it corresponds to the value of some terminal minor in the Tutte polynomial of G. Lemma 3.3.11 shows that the T_G marked nodes are the same as the marked nodes obtained by where the active and inactive walks meet. We allow nodes to be marked with multiplicity according to the number of terminal minors represented by that node, as multiple terminal minors may produce the same monomial. Lemma 3.3.11 shows that these notions are interchangable. Walking from the bottom left and top right corner simultaneously allows us to consider each of the four indeterminates as a different kind of step and visualize where on the grid the terminal minors from the original Tutte polynomial appears.

Lemma 3.3.11. *For a graph G with edge ordering $\overrightarrow{E}$ the set of marked nodes is the same as the set of T_G marked nodes.*

Proof. By lemma 3.3.10 the active and inactive portions of the Tutte grid walks meet at the same nodes, the marked nodes. Thus we can determine the marked nodes by the active portion of the walk only. The active portion of the walk is the set of edges sent to

x and y in the monomial. So the marked node can be determined by the monomials of $T_G(x, y, 1, 1; \overrightarrow{E})$ by lemma 3.1.2, which are the T_G marked nodes. ■

Lemma 3.3.12. *For a graph G with edge ordering $\overrightarrow{E}$, $e_1 < e_2 < ... < e_n$, define edge ordering $\overrightarrow{E'}$ to be $e_n < e_{n-1} < .... < e_2 < e_1$. Then for any pair of Tutte grid walks one from $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E'})$ and one from $T_G(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}; \overrightarrow{E})$ that correspond to the same maximal spanning forest we have the same set of edges corresponding to horizontal and vertical steps. Furthermore if all c and d have lower indices than x and y in a monomial in $T_G(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}; \overrightarrow{E})$ then the two walks have the exact same sequence of steps in the inactive portion of the Tutte grid walk.*

The primary reason the set of Tutte grid walks for the two polynomials in not the same is because the Tutte grid walks from the extended Tutte polynomial deshuffle the x and c apart and the y and d apart. It might be the case that a shuffled Tutte grid walk from the extended Tutte polynomial corresponds to a Tutte grid walk for the same or revered edge ordering from $J(\mathbf{t}, \mathbf{z}; \overrightarrow{E})$, but is certainly not always the case. It is an open question if there are relations under which the Tutte grid walks are the same, or if there is a way to given one polynomial and edge ordering determine an edge ordering that will produce the other polynomial with the same set of Tutte grid walks. In small examples we can produce the same sets of Tutte grid walks, but not with the same edges corresponding to all steps and there does not seem to be a clear way to generalize the result. The examples where this has worked do not have a large variety of walks due to being small. Many contain every possible walk and thus provide no insight.

Proof. This follows directly from the fact that $J(t, z; \overrightarrow{E}) = T_G(t, z, t, z; \overrightarrow{E})$ and the observation that both x and c correspond to horizontal steps in the grid as does t, while d, y, and z are all vertical steps. So given a fixed maximal spanning forest this fixes the set of horizontal steps in both Tutte grid walks, and thus fixes the vertical steps though the steps may occur in different places in the grid.

To see that the two walks agree on all in the inactive portion of the walk, observe that a c step is left and a d step is down, while t is right and z is up. So a c steps is the reverse of t and d is the reverse of z. Since the steps c and d walk down from the top right corner and t and z walk up from the bottom left the sequence of the steps from one walk needs to be entirely reversed to correspond to the steps of the other Tutte grid walk. Since the edge order for $T_G(x, y, c, d; \overrightarrow{E})$ is reversed for $J(t, z, ; \overrightarrow{E'})$ the inactive portion of the walk has the order of its steps flipped resulting in the portion of the walk being the same steps, but viewed as up and right corresponding to $J(t, z; \overrightarrow{E})$. ■

Example 3.3.13. In figure 3.12 we see a single Tutte grid walk for the spanning tree and monomial $dxccy$ from example 3.1.9 that converges to the circled point from the x and y side and the d and c side.

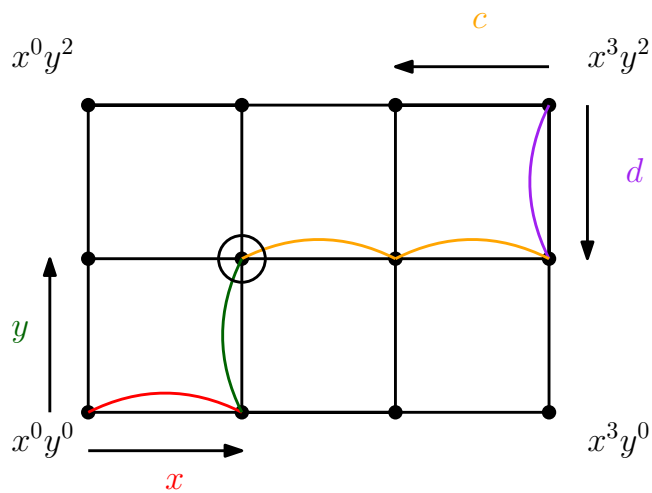


Figure 3.12: A Tutte grid with the walk for $dxccy$ shown

Beyond the walks themselves we consider the locations on the grid corresponding to terminal minors, with multiple instances of a monomial circled according to its multiplicity. In the Tutte polynomial of the graph in figure 2.2 is $x^3 + 2x^2 + x + 2xy + y + y^2$ and we can see that the x^2 and xy nodes are marked with multiplicity.

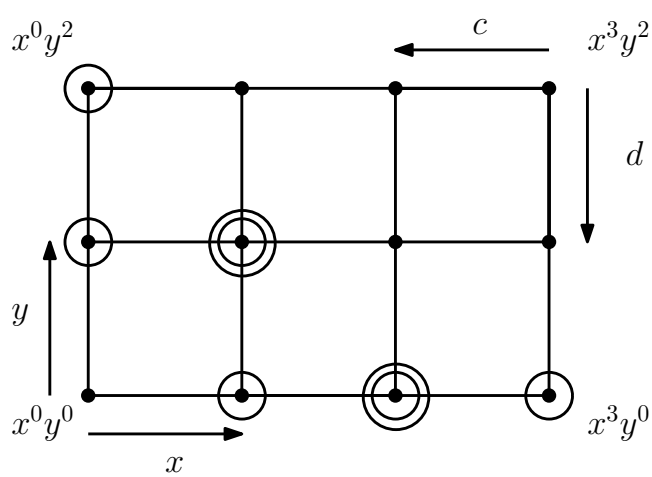


Figure 3.13: A Tutte grid with marked nodes

Theorem 3.3.14. *For a graph G and associated Tutte grid with terminal nodes marked, there is no grid walk between the two corners of the grid $x^i y^\ell$ and $x^r y^\phi$ that does not pass through at least one marked node.*

Proof. First, observe that if G is a graph made up of isthmi and loops then $x^r y^\phi$ is a terminal object in the deletion-contraction relation and thus a circled node in the grid. Similarly if the graph contains no edges then the $x^0 y^0$ node is the entire grid and also a marked node. Thus consider a graph G with a nonempty edge set and at least one cycle.

Suppose for the sake of contradiction that there exists a graph G and we have the Tutte grid with T_G marked nodes such that there is a grid walk W from $x^r y^\phi$ to $x^i y^\ell$ where i is the number of isthmi of G and ℓ is the number of loops of G. Now the T_G marked nodes are independent of the choice of edge ordering by lemma 3.3.11 since they are determined by the monomials of the Tutte polynomial. Since the marked nodes are fixed there is no edge ordering that we can place on G that marks a node in this grid walk W.

Since G has at least one cycle we have some edge e_1 that can be deleted or contracted. We will construct a sequence of deletions and contraction in G that corresponds to W. The first step in W leaving $x^r y^\phi$ is either left or down and we may choose to delete or contract e_1 to match this first step of W as e_1 is neither isthmus nor loop. Now if there are not more edges to delete or contract we would mark this node as the active portion of the walk is fixed by the inactive portion of the walk. Note the steps of the Tutte grid walk that mark this node may not align with W on the active portion, but since the inactive portion of the Tutte grid walk has ended and active and inactive walks meet by lemma 3.3.10 we mark a node on W. However, W does not contain a marked node, so we must have another edge e_2 that is a non-isthmus link.

Now we delete or contract e_2 to match the down or left step taken by W after the e_1 step. By the same logic as with e_1 either this node is marked by some Tutte grid walk that what the same steps as W on the inactive portion of the Tutte grid walk or there is another non-isthmus link in the graph. Continuing in the same manner we cannot encounter a marked node, so there must be another edge we can delete or contract at each step i of our grid walk and we may add this to our edge order as e_i. However, our graph is finite so this process must terminate at some node of the grid where we have no remaining edges we could delete or contract. At this point we may order all remaining edges arbitrarily and append them to the end of our edge ordering. This creates some Tutte grid walk W' that marks a node on the grid walk W. The node in W we are on at this stage must be marked, as the inactive portion of the walk has ended and as seen in the proof of lemma 3.3.10 the marked nodes are determined by where the inactive portion of the walks end. So for

W we can construct an edge order that creates a sequence of deletions and contractions terminating at a marked node on W, a contradiction. ■

For the next result we begin by defining an ear decomposition of a graph. P is an *ear* of G if $G = H \cup P$ where P is internally disjoint from H, meaning it intersect with H only at the endpoints of the path. A loop is considered an ear. We define an *ear decomposition of a graph* G to be a sequence of graphs $G_0, G_1, .., G_n$ where G_0 is a single edge ab, $G_1 = G_0 \cup P_1$ where P_1 is a path that intersects G_0 at a and b, building a cycle, and $G_n = G$. Furthermore $G_i = G_{i-1} \cup P_i$ by adding an ear P_i for $0 < i \leq n$. We now present some interesting results that relate to the marked nodes in the grid.

Theorem 3.3.15. *If G is a 2-connected graph that contains at least two cycles, then there exists a walk from x^0y^ϕ, the upper left corner node, to x^ry^ℓ, the lower right corner node, passing only through marked nodes.*

Proof. Notice that since G has two cycles, the grid has at least height two and width two. Since loops in G contribute only to ϕ and ℓ but not the Tutte grid walks of G we only need to prove the result for G with no loops. Thus we suppose G has no loops. Since G is 2-connected, G has an ear decompositions $G_0, G_1 = G_0 \cup P_1, ..., G_n = G_{n-1} \cup P_n$ where $n \geq 2$ since G has at least two cycles. Choose the ear decomposition such that any ear, P_i, of a single edge other than G_0 has indices larger than all ears with more than one edge. This is possible since these ears add no additional vertices. Since by lemma 3.3.11 the edge order does not determine which nodes are T_G marked we will use sequences of deletions and contractions to prove the nodes along such a walk are marked.

First observe that by a sequence of all deletions we mark x^ry^0 and a sequence of all contractions marks x^0y^ϕ. We will induct on n. If $n = 2$ then if P_2 is not a single edge contract one edge and delete an edge from P_2 and P_1 to mark $x^{r-1}y^0$. If P_2 is a single edge simply delete P_2 and contract an edge in P_1, which has a least two edge, and then delete an edge in P_1 to mark this node. We now repeat the process of constructing terminal minors to mark nodes by contracting one more edge in P_2 than was contracted to mark the previous node and then deleting an edge from P_1 and P_2. Once we have marked nodes by contracting P_2 to a single edge we delete the remaining edge of P_2 and contract edges in P_1. By contracting one edge in P_1 before deleting an edge of P_1 we mark the next unmarked node in the bottom row of the grid. As with P_2 we contract one more edge before deleting an edge of P_1 to mark nodes in the bottom row until we have contracted P_1 to a single edge and then deleting this edge. This will mark xy^0 and all nodes further right in the bottom row from xy^0 to x^ry^0.

Now we mark xy. Contract P_2 to a loop. Then contract P_1 to a single edge and delete this edge. G_0 is now an isthmus and P_2 is a loop so xy is marked. Now we mark just x^0y.

Contract P_2 to a single edge and delete the remaining edge in P_2. Then contract all of P_1, making G_0 a loop. This marks x^0y. Now $n = 2 = \phi$ and we showed xy^ϕ is always marked so we are done. Notice that this does not rely on the graph not having loops as we simply would ignore them in the process and the base case cannot have loops as the graph must have two cycles.

Suppose for any 2-connected graph with at least two cycles and rank at least two and ear decomposition $G_0, G_1 = G_0 \cup P_1 ..., G_k = G_{k-1} \cup P_k,\ k \geq 2$, that there is a walk through marked nodes from x^0y^ϕ to x^ry^ℓ. Consider a graph G with ear decomposition $G_0, G_1 = G_0 \cup P_1 ..., G_{k+1} = G_k \cup P_{k+1}$. If P_{k+1} is a loop then the results holds for $G - P_{k+1}$ and this loop has no impact on the marked nodes or grid so the result holds. Thus suppose that P_{k+1} is not a loop.

Let r be the rank of G, ϕ the cyclomatic number, and j the number of edges in P_{k+1}. Now P_{k+1} contributes one to the cyclomatic number of G and $G - P_{k+1}$ has an ear decomposition and satisfies the inductive hypothesis so it has a walk through marked nodes $x^0y^{\phi-1}$ to x^{r-j+1} since j increases the size of a spanning tree of $G - P_{k+1}$ by $j - 1$ edges. Since x^0y^ϕ is always marked we have a walk passing through marked nodes from x^0y^ϕ to $x^{r-j+1}y^0$. If $j = 1$ then we are done as P_{k+1} does not change the rank. Thus suppose $j > 1$. We know x^ry^0 is marked by deleting one edge from each ear other than G_0. Now for $1 \leq i < j$ contract i edges in P_{k+1} and then delete one edge from each ear other than G_0 to mark nodes $x^{r-j+1+i}$ which makes a walk between marked noes from x^{r-j+1} to x^r. This completes the path. Hence the result holds by induction. ■

We need at least two cycle as we otherwise have no terms with both x and y and so there are no marked nodes outside of the first column and bottom row. In this case there is no way to walk from xy^0 to $x^0y^{\ell+1}$ without passing a node that is not marked. We need the ear decomposition so that we can ensure we have a choice of sending every cycle to x or y and that we can merge cycles together. If we had a cut vertex for example with a cycle on either side of it these two cycles would allow us to get an x or y each, but we there is no way to produce a terminal minor with only one y and no x. We need this single y to allow us to walk into the first column. See figure 3.14 for two graphs which do not posses this walk through marked nodes. This walk over marked nodes does exist for some graphs that do not satisfy the theorem conditions due to what values show up in the monomials, but it is only consistently true when we have have the ear decomposition to work with, allowing control over the number of x and y in each term.

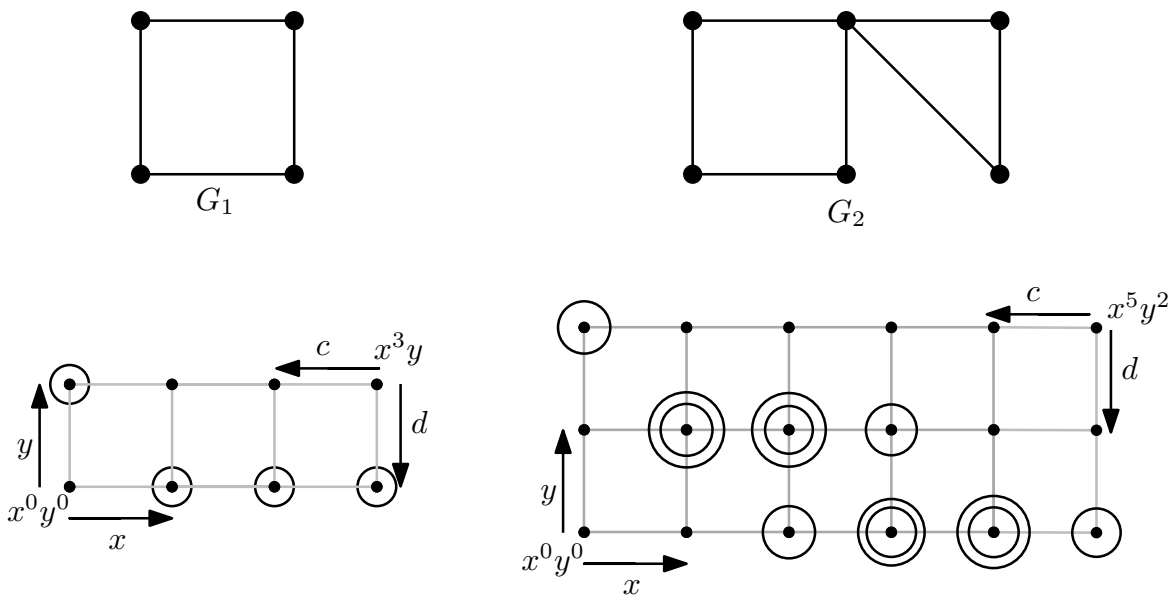


Figure 3.14: Two graphs that do not have a x^0y^ϕ corner to $x^{r(G)}y^0$ corner walk on marked nodes

We see the first marked node in the path is obtained through only deletions. Then we contract the blue ear to one size smaller before using only deletion to obtain a marked node with one fewer instance of x. Next we take the ear (in pink) and use contraction to shrink both by one edge. Then again we use only deletion and now we have two fewer instances of x than we started with, or one fewer than the previous marked node. Now we take an ear, which in this case is the initial cycle, and contract it to a loop. Then we contract all other ears to a pair of edges, the blue ear is already only two edges, and we delete down to x. This gives us the xy term that allows us to move into the middle of the grid.

Example 3.3.16. Considering the graph in figure 2.2 we can see an ear decomposition of this graph in figure 3.15 where the initial edge is shown in black and the ears are shown in blue and pink. In this figure we have the six terminal minors that form the corner to corner grid walk only passing through marked nodes.

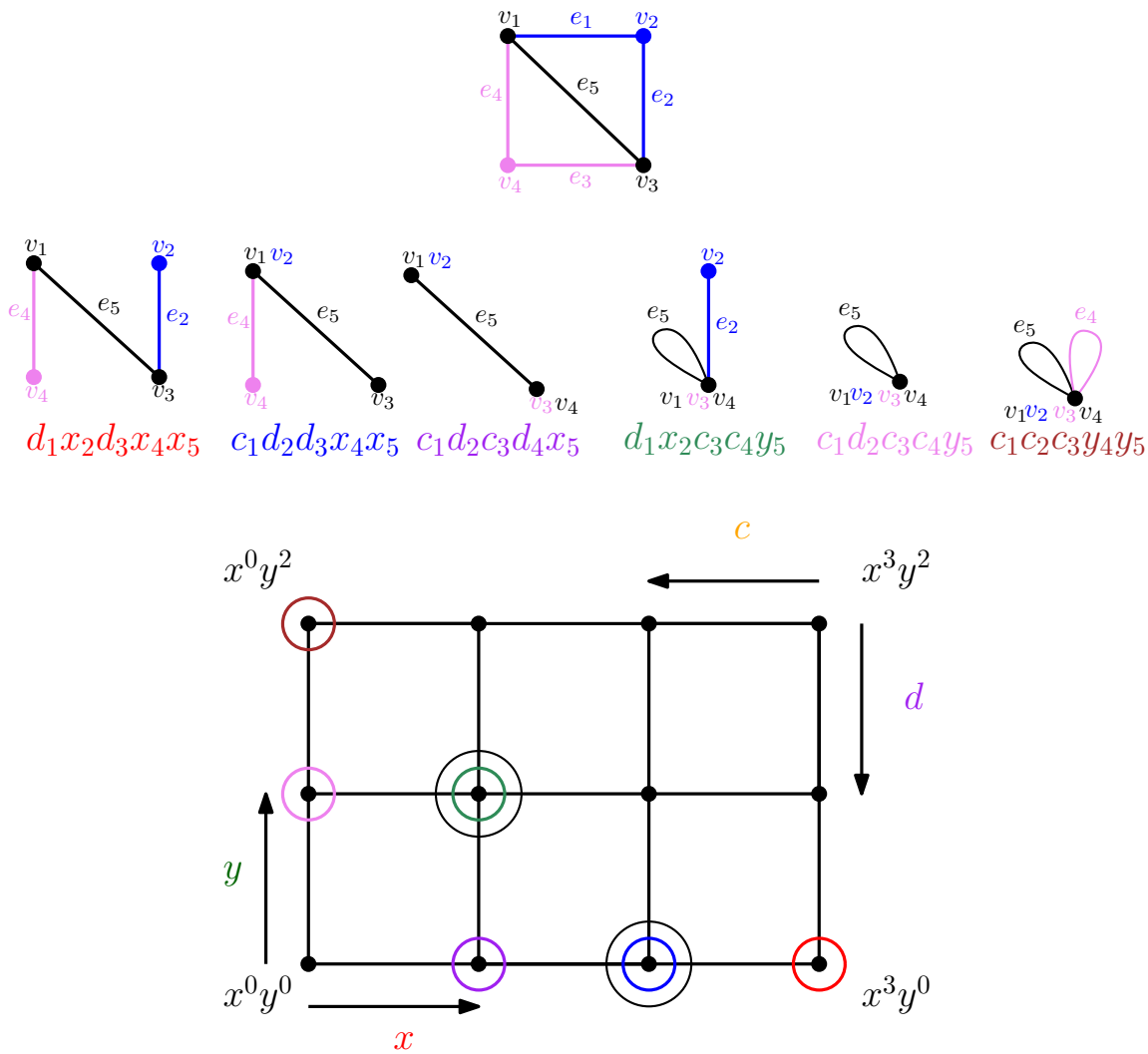


Figure 3.15: A set of terminal minors whose marked nodes form a $x^r y^\ell$ to $x^0 y^\phi$ walk

Now we contract all ears except one to a single edge, where the last ear is contracted to two edges, and we then delete down to just the pair of parallel edges before contracting again. This gives the lowest power of y and lets us walk into the first column of the grid. Finally we simply contract all ears to get the maximum number of y terms giving us the final marked node in the path.

Since we don't care about the edge order when considering marked nodes we don't care about the edge order at any point. So we simply delete and contract in the ears as benefits

us. The initial edge in the ear decomposition, shown in black in figure 3.15 is always going to contribute x or y as we delete down to it or contract the final edge in an ear with it.

Given the relationship between the extended Tutte polynomial and the extended Kirchhoff polynomial we have the following results.

Lemma 3.3.17. *Let G have edge orderings $\overrightarrow{E}$ and $\overrightarrow{E'}$ which differ by a twist permutation π. Then for a maximal spanning forest of G which corresponds to some Tutte grid walk W of $\overrightarrow{E}$ and some Tutte grid walk W' of $\overrightarrow{E'}$ the node marked by W and the node marked by W' are distinct unless the maximal spanning forest contains the same set of edge labels in both edges orderings.*

Proof. Since the edge orderings differ by a twist permutation the sets of edges in each cycle are preserved, but the activities and thus the marked node location depend on the edge ordering relation to the maximal spanning forest. So given any maximal spanning forest of the graph, the activities and thus steps in the Tutte grid walks are equal only when π does not affect the set of edge labels of the maximal spanning forest. If the edge labels in the maximal spanning forest are affected by π then there must be at least two edges whose activities are swapped as we have not changed the maximal spanning forest, but have changed the priority of the edges which have a total ordering. So W and W' mark the same node only when the set of edges in the maximal spanning forest is unchanged by π. ■

An example of this lemma can be seen in figure 3.16 where two marked nodes are affected by the twist permutation and two are not.

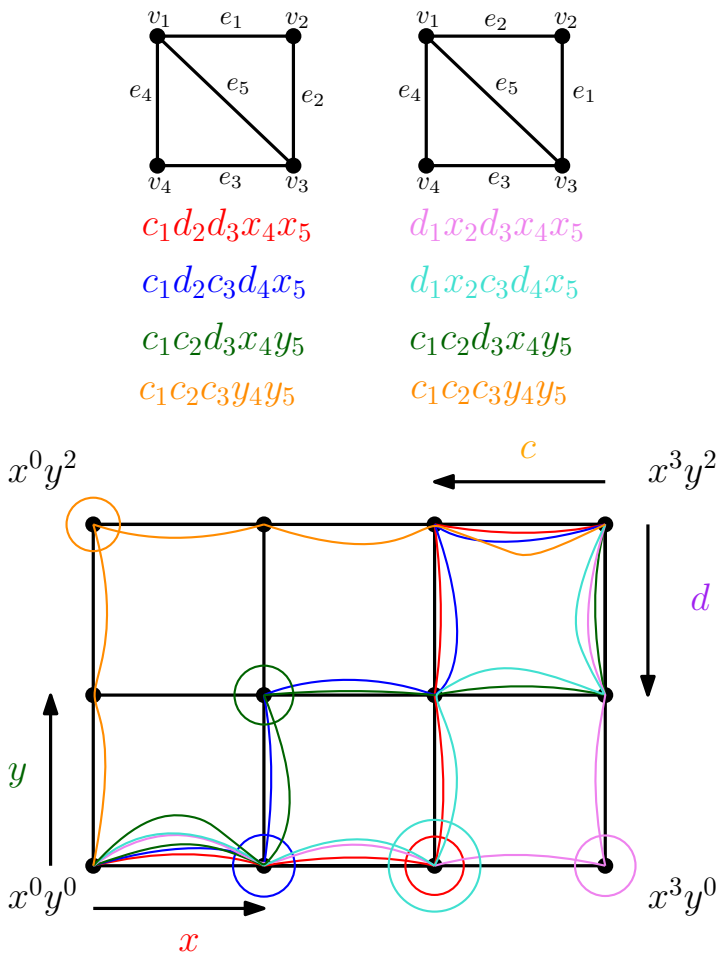


Figure 3.16: The affect of a twist permutation on some marked nodes of a graph.

Theorem 3.3.18. *Let G be a graph. If π is a twist permutation from $\overrightarrow{E}$ on G to $\overrightarrow{E'}$ on G and T is a maximal spanning forest associated to a Tutte grid walk from $T_G(x, y, c, d; \overrightarrow{E})$, then that Tutte grid walk is associated to the maximal spanning forest $\pi(T)$ for the same monomial under edge ordering $\overrightarrow{E'}$.*

Proof. Notice that π preserves the set of labels in any cycle. The set of edges in a maximal spanning forest is determined by x and c from lemma 3.1.5. So if T is a maximal spanning forest from $\overrightarrow{E}$ then $\pi(T)$ is also a maximal spanning forest in $\overrightarrow{E'}$. For every cycle in the graph, T does not contain all edges in that cycle. Then π preserves this fact and so $\pi(T)$ contains no cycles. Thus $\pi(T)$ has the same number of x and c edges as T making it a

maximal spanning forest. ■

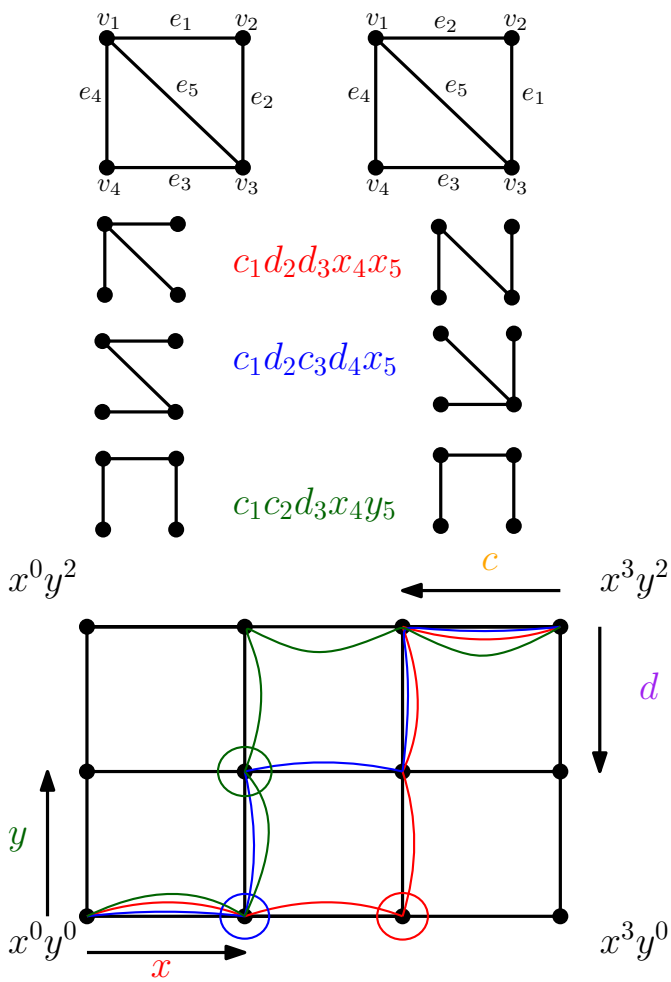


Figure 3.17: Spanning trees corresponding the Tutte grid walks under a twist permutation

Corollary 3.3.19. *If $\overrightarrow{E}$ and $\overrightarrow{E'}$ differ by a twist permutation for a graph G, then the set of all Tutte grid walks determined by $T_G(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}; \overrightarrow{E})$ is the same as the set of Tutte grid walks determined by $T_G(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}; \overrightarrow{E'})$.*

Proof. This follows from theorem 3.3.18 as each maximal spanning forest T corresponds to a Tutte grid walk which is the Tutte grid walk for $\pi(T)$ under the other edge ordering. So the set of all Tutte grid walks is unchanged. ■

3.3.1 Special Classes of Graphs

While it is difficult in general to predict what Tutte grid walks and marked nodes will correspond to a graph and edge ordering, there are some predicable graphs. Forests have a single marked node for the grid as well as forests with loops added to any vertices. These graphs do not make particularly interesting examples given the lack of non-trivial cycles. Cycles however make slightly more interesting example. Figure 3.20 shows an example of these walks. The grid for any cycle is just a single step high, but every node in the bottom row and first column is marked except the first node. Every possible Tutte grid walk is in fact a Tutte grid walk for a cycle and the edge order has no impact on the Tutte grid walks.

Square lattice graphs, as seen in figure 3.18, also make an interesting class of examples, in part because we are plotting walks on a grid on spanning trees of a grid. We can view a square lattice graph as having some number of cells, which are C_4 subgraphs. The grid has height equal to the number of cells and width twice the number of cells plus one. So if the square lattice graph has k cells the upper right node of the grid is $x^{2k+1}y^k$. These are also a fun class of examples because every edge of the grid is used in some walk and if a node is marked in a column then so is every node below and to the left, with the exception of the x^0y^0 node. The multiplicity of the nodes in the row y^0 are also interesting as the multiplicity is symmetrical. This seems to be a direct result of the fact that the square lattice graph is so symmetrical and the ease of combining and breaking cycles.

The final interesting class of graphs we will mention are complete graphs. An example of the Tutte grid walks for K_4 can be seen in figure 3.19. The complete graphs are not particularly interesting in terms of examining how Tutte grid walks change under various edge orderings because every possible Tutte grid walk is a Tutte grid walk for some monomial. The upper right node in the grid is $x^{n-1}y^{\frac{(n-1)(n-2)}{2}}$. Another interesting observation is that it appears to be the case that if x^jy^k is marked m times then x^ky^j is marked m times. This seems to occur because of the symmetry of the grid to the left of these marked nodes and the fact that every possible Tutte grid walk is a Tutte grid walk for some monomial, but we have not tested more than the first few complete graphs. It seems likely this would be provable by using spanning tree counts, the fact that every complete graph K_n has an ear decomposition with G_1 a cycle on n vertices, and the highly symmetrical nature of the graph which appears to allow us to swap deletions and contraction in a sequence that leads to a terminal minor to get another terminal minor.

Conjecture 3.3.20. *For a complete graph K_n if the x^0y^k node is marked m times then the node x^ky^0 is marked m times.*

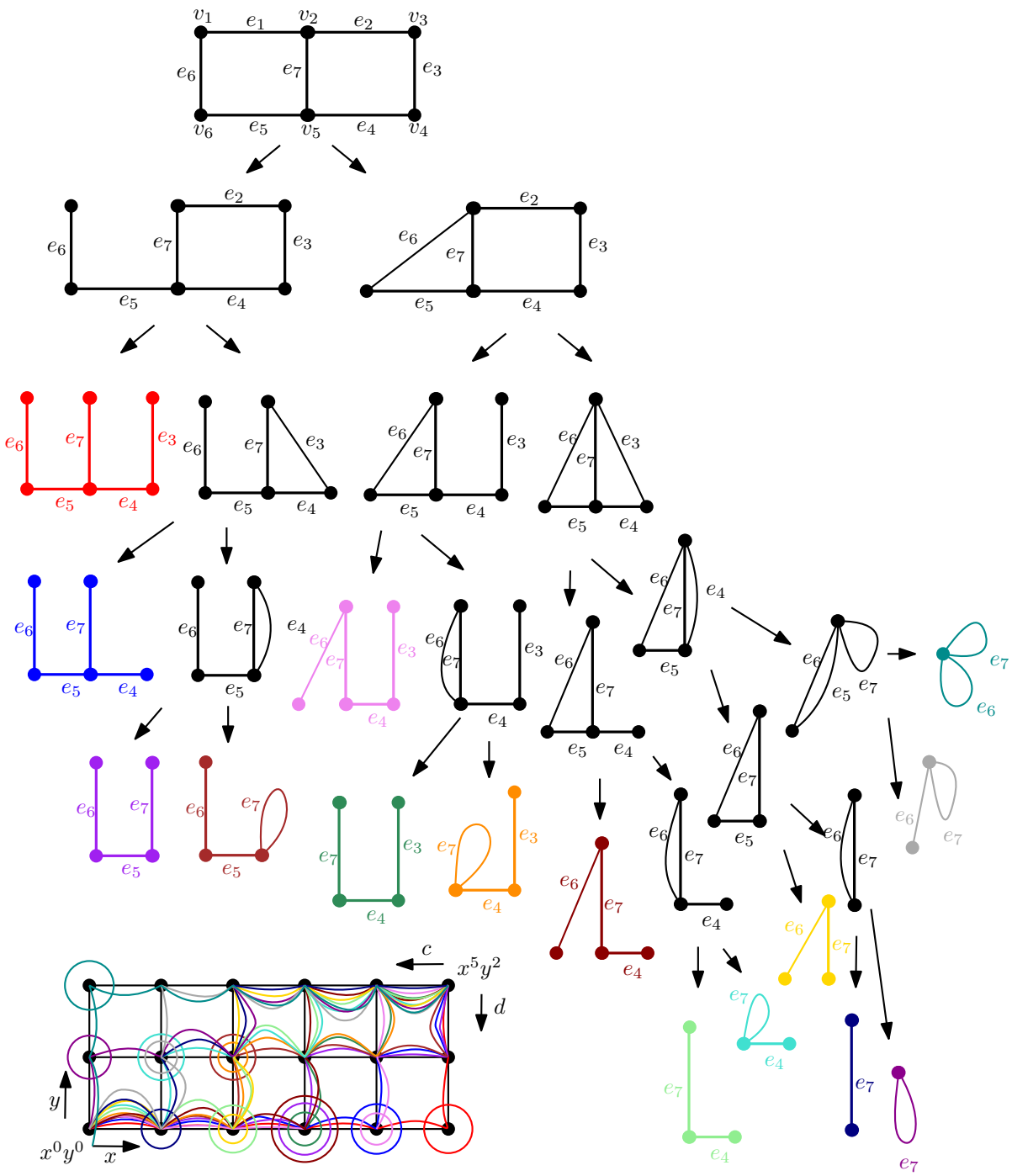


Figure 3.18: A deletion contraction decision tree for a lattice graph and its Tutte grid walks

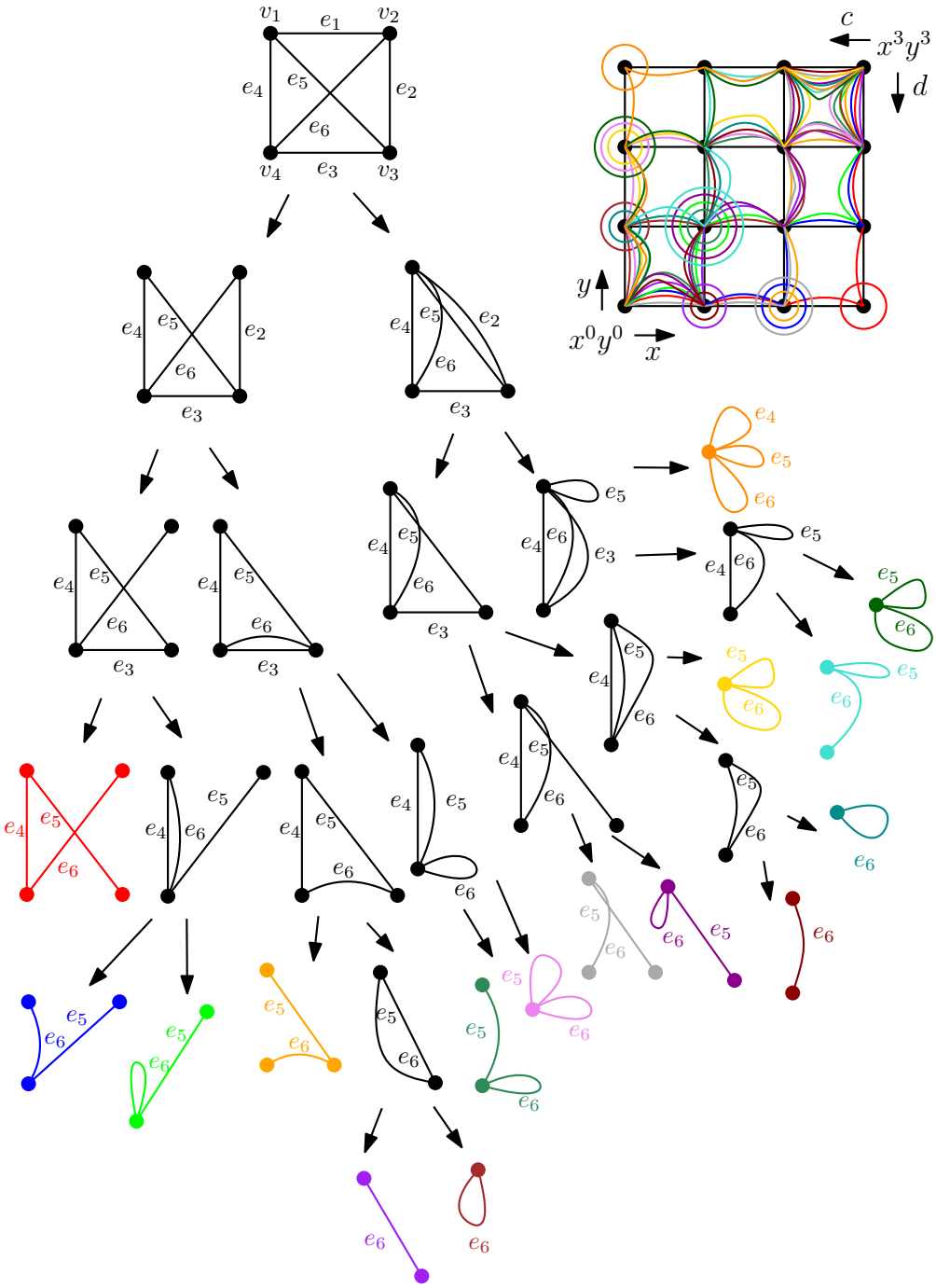


Figure 3.19: A deletion contraction decision tree for K_4 and its Tutte grid walks

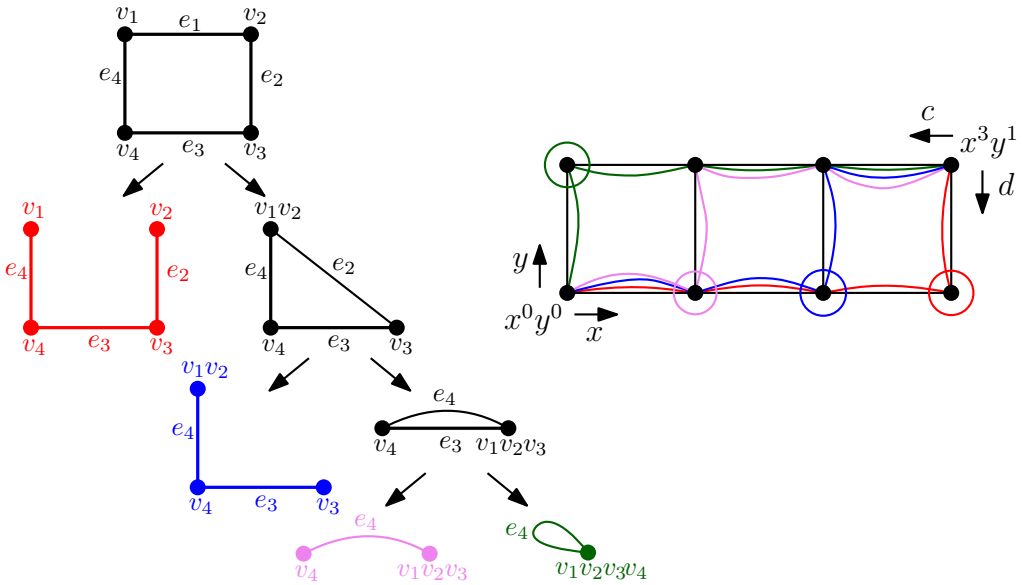


Figure 3.20: A deletion contract decision tree for C_4 and its Tutte grid walks

3.4 An Extended Tutte Polynomial for Signed Graphs

In this section, we explore how we can use the notion of basis as relating to that of lift matroids to construct a signed graph Tutte polynomial. We construct this polynomial using the extended Tutte polynomial and thus are able to obtain both a deletion-contraction and activity version. Before we construct the polynomial we begin with defining an operation that will allow us to construct such a polynomial.

3.4.1 A Missing Operation

To construct this missing operation we use the fact that for a graph or balanced signed graph the set of bases is made up of maximal spanning forests, but for a unbalanced signed graph the set of basis is maximal spanning forests plus a single edge that induces a negative cycle. This new operation, *basis loading* is the act of taking a maximal spanning forest of the underlying graph of a signed graph and adding an edge to form a negative cycle. Notice that for balanced signed graphs this operation cannot form a negative cycle so we instead return the original maximal spanning forest. We will define more specifically the

loading of a maximal spanning forest and a monomial to more nicely enable us to define our version of the singed graph Tutte polynomial.

Let $\Sigma = (G, \sigma)$ be a signed graph and T be a maximal spanning forest of G. The *loading of* T, denoted by $\mathcal{L}(T)$, is the set of all subgraphs $T + e$ where $T + e$ contains a negative cycle if Σ is unbalanced and $\mathcal{L}(T) = \{T\}$ if Σ is balanced.

Example 3.4.1. As seen in figure 3.21, the loading takes every maximal spanning forest (spanning trees in this case) and tests every edge outside of the tree to see if it builds a negative cycle. For the tree T we see that both edges outside of T form a negative cycle, so the loading of this tree produces two bases. The loading of T' on the other hand produces one basis as one of the edges outside T' produces a positive cycle, e_1, e_2, e_3, e_4. If no negative cycles are produced then the loading would return the original maximal spanning forest.

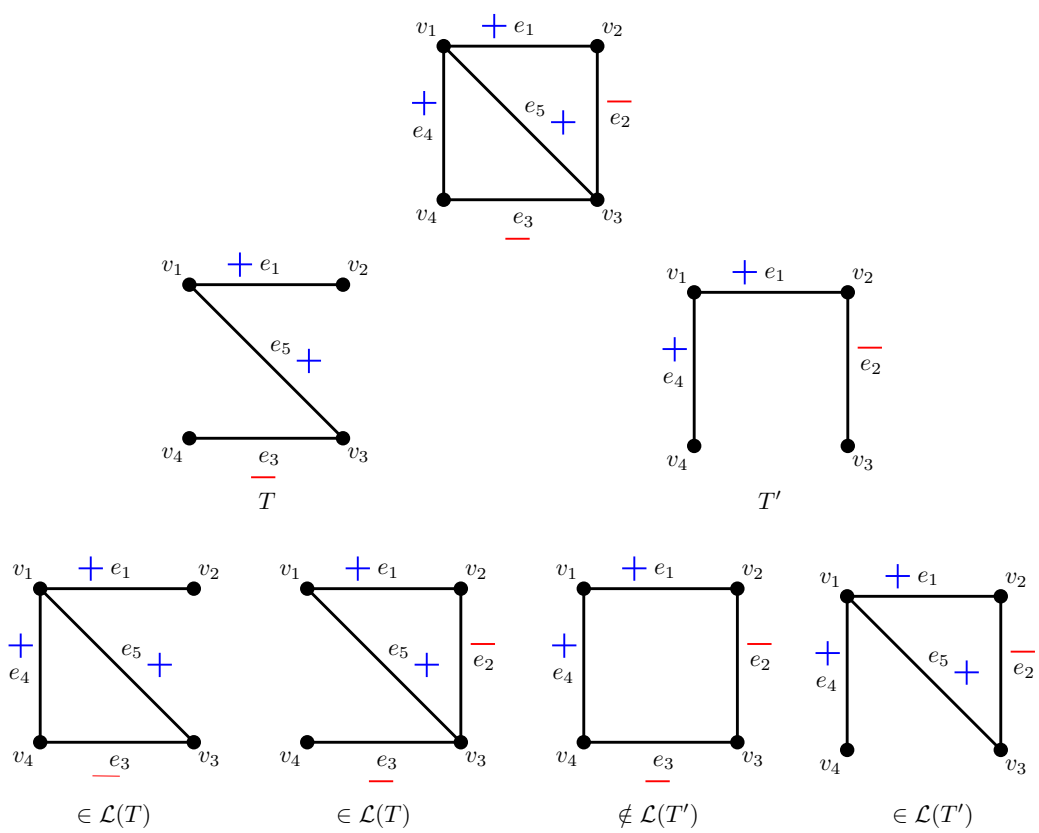


Figure 3.21: The loading of two spanning trees

The loading of a maximal spanning forest is going to enable us to nicely extend our activity definition of the extended Tutte polynomial to signed graphs. However, it is not formulated in a way that clearly indicates how we could use it for our deletion contraction extension. Thus we will define another variation of loading, this time on a monomial.

Recall that a terminal minor of a graph is constructed by a sequence of deletions and contractions, ignoring loops and isthmi at each step. So each terminal minor corresponds to a monomial of the extended Tutte polynomial, which we denote m_S. Notice that m_S is $\prod_{e \in \overrightarrow{E}} M_S(e)$ where

$$M_S(e) = \begin{cases} x_e & \text{if } e \text{ is an isthmus,} \\ c_e & \text{if } e \text{ is contracted,} \\ y_e & \text{if } e \text{ is a loop,} \\ d_e & \text{if } e \text{ is deleted.} \end{cases}$$

Let $\Sigma = (G, \sigma)$ be a signed graph with total edge ordering $\overrightarrow{E}$ and monomial m_S corresponding to some terminal minor of G. The *loading of* m_S, denoted by $\mathcal{L}(m_S)$, is the set of monomials where a single instance of y_e or d_e is replaced by f_e or r_e, respectively, if e is frustrated with respect to the maximal spanning forest determined by the edges associated to the x's and c's if Σ is unbalanced. If Σ is balanced, then we set $\mathcal{L}(m_S) = \{m_S\}$.

It is an important observation that by construction both loadings return the original object if no edges are frustrated with respect to the maximal spanning forest. This is done because in a graph or balanced signed graph we want to preserve the fact that the basis is a maximal spanning forest and no other edges need to be added.

Example 3.4.2. In figure 3.22 we demonstrate the loading of three monomials obtained via deletion and contraction as shown in figure 3.1. The first two monomials have two edges sent to d_e that form negative cycles yielding two new monomials in their loading with an instance of d_e replaced by r_e. The final monomial has a d and a y. The edges assigned to this d and x and c form a positive cycle, so this edge doesn't affect the loading. The edge assigned to y does build a negative cycle with the c and x edges, so the loading produces a single new monomial with the y replaced by f. If the signed graph was balanced the loading would return the original monomials.

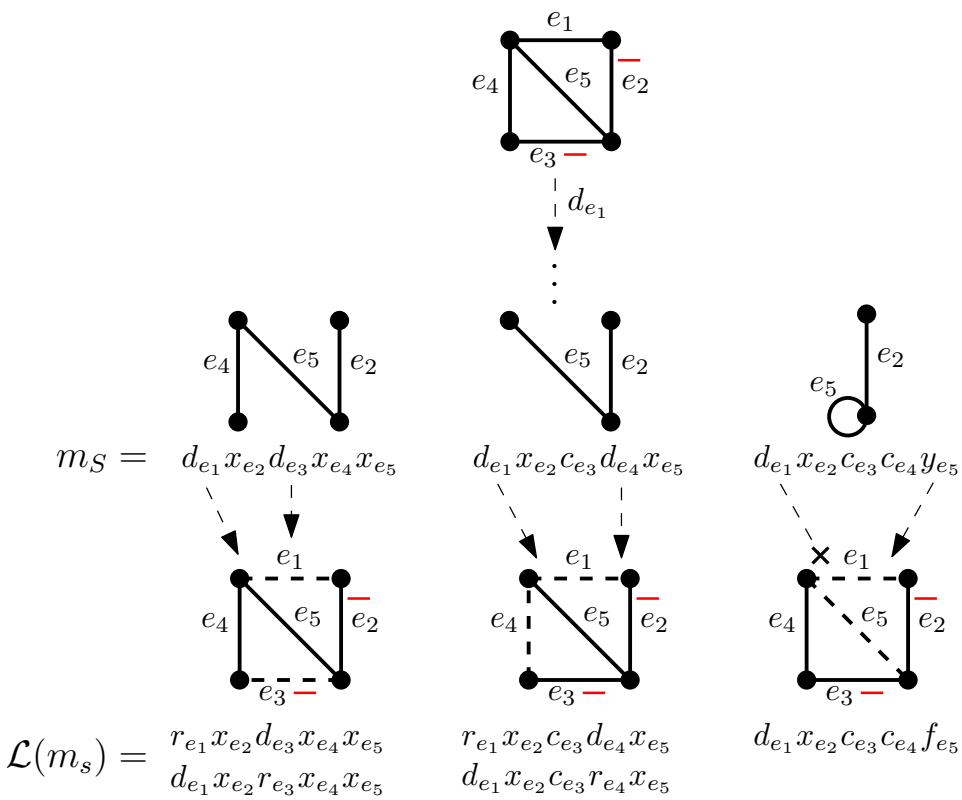


Figure 3.22: The loading of three monomials of a signed graph

Lemma 3.4.3. *$\Sigma = (G, \sigma)$ is balanced if and only if for all maximal spanning forests T, $\mathcal{L}(T) = \{T\}$.*

Proof. If Σ is balanced then for all maximal spanning forests T and edges e, $T + e$ does not contain a negative cycle. Hence $\mathcal{L}(T) = \{T\}$ for all T.

If $\mathcal{L}(T) = \{T\}$ for all T then for every maximal spanning forest there exists no edge e such that $T+e$ contains a negative cycle. For any cycle there exists some maximal spanning forest that contains all but one edge from that cycle and thus if no maximal spanning forest plus one edge ever contains a negative cycle then there are no negative cycles. So Σ is balanced. ■

Lemma 3.4.4. *$\Sigma = (G, \sigma)$ is balanced if and only if for all monomials m_S of G, $\mathcal{L}(m_S) = \{m_S\}$.*

Proof. If Σ is balanced then by definition we have no negative cycles and thus no frustrated edges with respect to any maximal spanning forest. So for any monomial m_S of a terminal minor of G we have $\mathcal{L}(m_S) = \{m_S\}$.

If for all monomials m_S associated to terminal minors of G, $\mathcal{L}(m_S) = \{m_S\}$ then no edge assigned to y or d is frustrated and so no edge forms a negative cycle with respect to the maximal spanning forest determined by the x and c edges. Since every cycle in Σ has all but one edge in some maximal spanning forest of Σ if no negative cycle can ever be built then Σ is balanced by definition. ■

Lemma 3.4.5. *For a signed graph Σ and maximal spanning forest T the loading of T produces that same set of bases as the loading of the monomial associated to T, denoted by m_T from the extended Tutte polynomial of Σ.*

Proof. First observe that either Σ is balanced or it is not. If Σ is balanced, the basis from the loading of T is T and the basis from the loading of m_T is also T as the x and c entries of the monomial are the edges in T. Thus these are trivially the same basis.

Now suppose that Σ is unbalanced. Then $T + e$ contains a negative cycle for some $e \notin T$. Let $N_T = \{e \mid T + e \text{ contains a negative cycle}\}$. Then $\mathcal{L}(T) = \{T + e | e \in N_T\}$. So each element of the basis associated to T is formed by the edges of T plus e. Now m_T has the set of edges assigned to x and c that belong to T. So again, any edge $e \in N_T$ is not in T and thus is a d or y in m_T. Thus each of these edges form a negative cycle and are frustrated. To see no other edges are frustrated observe that by construction N_T contains all edges that form a negative cycle with T. Thus $\mathcal{L}(m_T) = \{m_{T'} | \text{ where } T' \text{ by replacing exactly one edge in } N_T \text{ as } f \text{ or } r\}$. So the set of basis is again the set of edges assigned x and c plus e for each $e \in N_T$. ■

There are a variety of tests for balance that involve counting the number of negative edges in all cycles, partitioning the edges and looking at the cut sets, or testing switchings [62, 63, 64, 65, 132, 135]. When thinking about frustration and not wanting to contract negative edges we realized we could test for balance using this idea.

Lemma 3.4.6. *A signed graph Σ is balanced if and only if in the terminal minor obtain by only contracting edges every loop is positive, where a negative edge must have one of its endpoints switched before being contracted.*

Proof. If Σ is balanced then when we switch negative edges before contracting all non-isthmus, link edges we cannot make the graph unbalanced. Furthermore as all cycles in a balanced graph are positive when a cycle is fully contracted to a loop that loop retains the sign of the cycle and is thus positive.

If Σ is unbalanced then there exists a negative cycle. By switching this cycle cannot be made positive, but any edge can be made positive by switching an endpoint. So as this negative cycle is contracted we may shift the negative signs around the cycle, but an odd number will always remain. When we contract the second to last edge in the cycle either it or the other remaining edge are negative. Either way this negative sign will remain on the final edge of the cycle which becomes a loop. So, the loop is negative. ■

3.4.2 An Extended Tutte Polynomial for Signed Graphs

Now that we have the definitions of loadings we can use these definitions to define our extended Tutte polynomial for signed graph. We begin with the activity based definition.

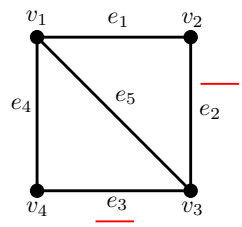
Definition 3.4.7. *Let Σ be a signed graph with total edge ordering $\overrightarrow{E}$ and set of maximal spanning forests $\mathcal{T}$. Define the* activity Tutte polynomial for signed graph *by:*

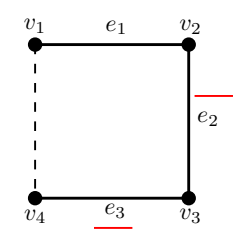
$$T_\Sigma(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}, \mathbf{f}, \mathbf{r}; \overrightarrow{E}) = \sum_{T \in \mathcal{T}} \sum_{B \in \mathcal{L}(T)} \prod_{e \in \overrightarrow{E}} A_B(e)$$

where

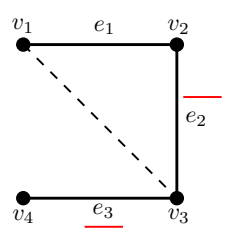
$$A_B(e) = \begin{cases} x_e & \text{if } e \text{ is internally active,} \\ c_e & \text{if } e \text{ is internally inactive,} \\ y_e & \text{if } e \notin B - T \text{ and is externally active,} \\ d_e & \text{if } e \notin B - T \text{ and is externally inactive,} \\ f_e & \text{if } e \in B - T \text{ and is externally active,} \\ r_e & \text{if } e \in B - T \text{ and is externally inactive.} \end{cases}$$

We may refer to the edges sent to f as *actively frustrated* and edges sent to r as *inactively frustrated*. The word frustrated implies external and indicates that we are adding the edge to our basis. In figures 3.23 and 3.24 we see the monomial computed from the basis loadings of all the spanning trees of a graph. Each monomial has exactly one instance of r or f as the graph in question is unbalanced. Further, we can observe that four spanning trees only have one element in their loading as the addition of any other edge outside the tree builds a positive cycle.

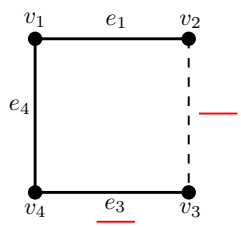




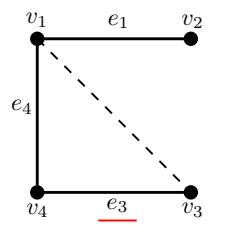
Internal		External
e_1	c_1	$e_4\ y_4$
e_2	c_2	$e_5\ y_5$
e_3	c_3	$\notin \mathcal{L}(T)$



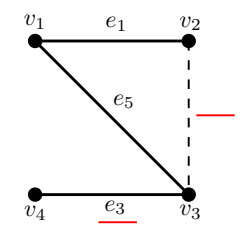
Internal		External
e_1	c_1	$e_4\ y_4$
e_2	c_2	$e_5\ f_5$
e_3	c_3	$c_1c_2c_2y_4f_5$



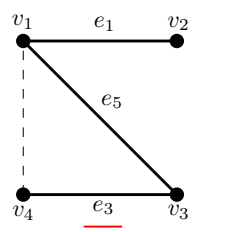
e_1	c_1	$e_2\ d_2$
e_3	c_3	$e_5\ y_5$
e_4	c_4	$\notin \mathcal{L}(T)$



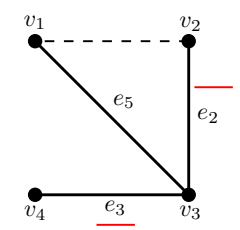
e_1	c_1	$e_2\ d_2$
e_3	c_3	$e_5\ f_5$
e_4	c_4	$c_1d_2c_3c_4f_5$



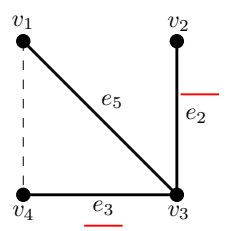
e_1	c_1	$e_2\ r_2$
e_3	c_3	$e_4\ d_4$
e_5	x_5	$c_1r_2c_3d_4x_5$



e_1	c_1	$e_2\ d_2$
e_3	c_3	$e_4\ r_4$
e_5	x_5	$c_1d_2c_3r_4x_5$

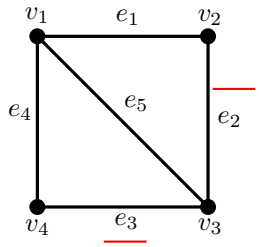


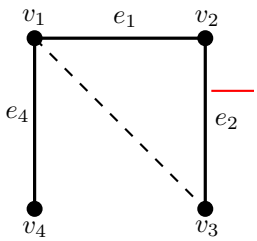
e_2	x_2	$e_1\ r_1$
e_3	c_3	$e_4\ d_4$
e_5	x_5	$r_1x_2c_3d_4x_5$



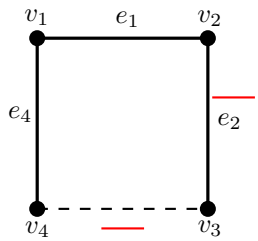
e_2	x_2	$e_1\ d_1$
e_3	c_3	$e_4\ r_4$
e_5	x_5	$d_1x_2c_3r_4x_5$

Figure 3.23: The first half of the activity calculations on the loading of half the spanning trees of a graph see figure 3.24 for the rest.

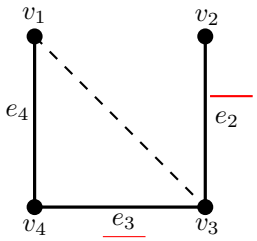




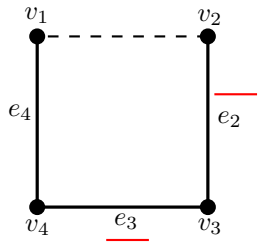
Internal	External
e_1 c_1	e_3 d_3
e_2 c_2	e_5 f_5
e_4 x_4	$c_1c_2d_3x_4f_5$



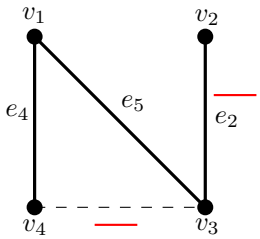
Internal	External
e_1 c_1	e_3 d_3
e_2 c_2	e_5 y_5
e_4 x_4	$\notin \mathcal{L}(T)$



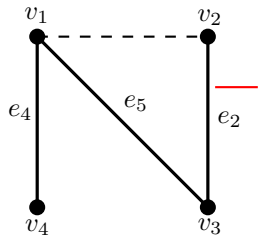
e_2 x_2	e_1 d_1
e_3 c_3	e_5 f_5
e_4 c_4	$d_1x_2c_3c_4f_5$



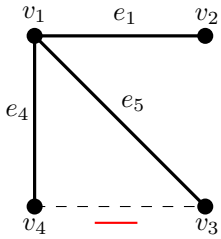
e_2 x_2	e_1 d_1
e_3 c_3	e_5 y_5
e_4 c_4	$\notin \mathcal{L}(T)$



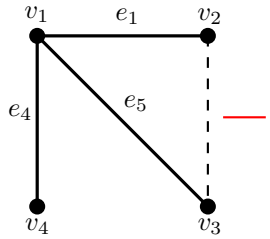
e_2 x_2	e_1 d_1
e_4 x_4	e_3 r_3
e_5 x_5	$d_1x_2r_3x_4x_5$



e_2 x_2	e_1 r_1
e_4 x_4	e_3 d_3
e_5 x_5	$d_1x_2r_3x_4x_5$



e_1 c_1	e_2 d_2
e_4 x_4	e_3 r_3
e_5 x_5	$c_1d_2r_3x_4x_5$



e_1 c_1	e_2 r_2
e_4 x_4	e_3 d_3
e_5 x_5	$c_1r_2d_3x_4x_5$

Figure 3.24: The activity calculations on the loading of the second half of the spanning trees of a graph

We may use the monomial loading and extended Tutte polynomial to define a signed graph extended Tutte polynomial that implicitly uses the deletion and contraction in the calculation of the extended Tutte polynomial $T_G(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}; \overrightarrow{E})$ for the underlying graph G of the signed graph $\Sigma = (G, \sigma)$, but has an additional step of the monomial loading which does not use this deletion contraction relationship. Using the definitions on page 95,

Definition 3.4.8. *Let Σ be a signed graph with total edge ordering $\overrightarrow{E}$. Define the* terminal minor Tutte polynomial for signed graphs *by:*

$$T_\Sigma(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}, \mathbf{f}, \mathbf{r}; \overrightarrow{E}) = \sum_{m_S \in T_G(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d})} \sum_{m \in \mathcal{L}(m_S)} m.$$

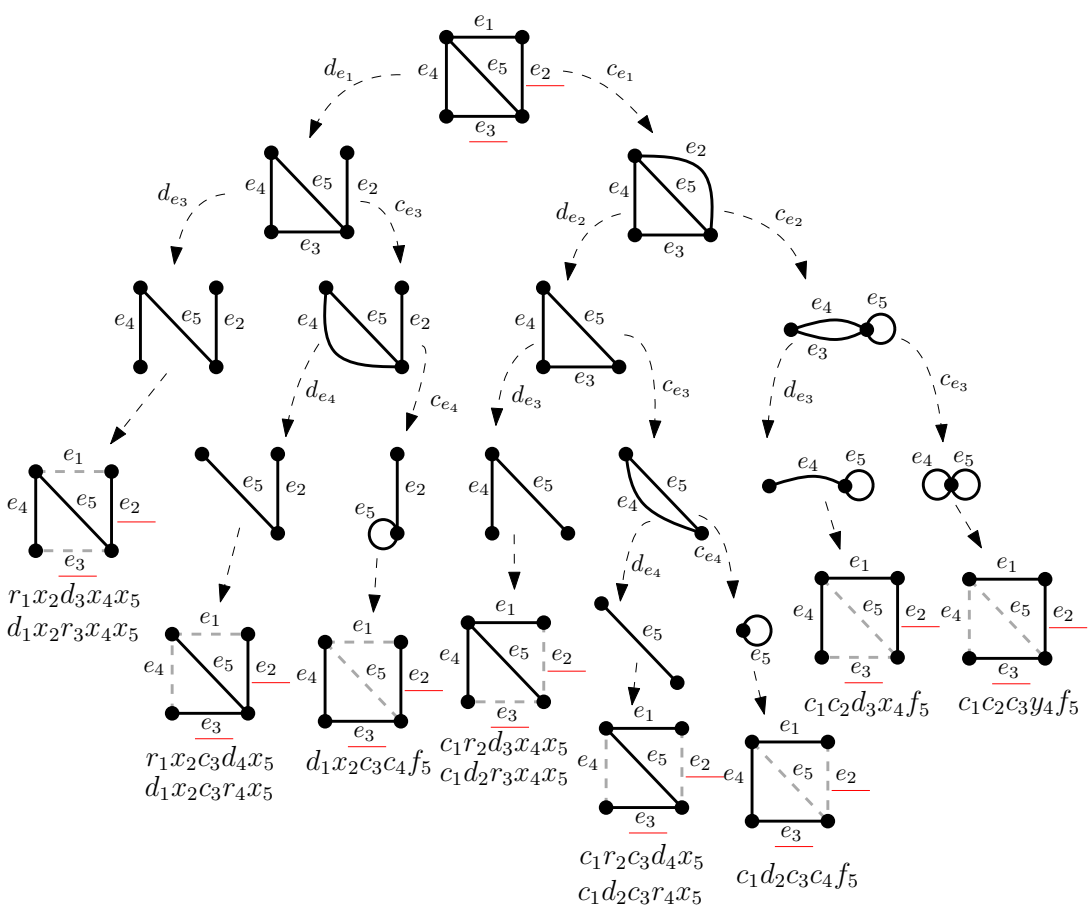


Figure 3.25: The extended Tutte polynomial for signed graphs computed via deletion and contraction

In figure 3.25 we see that the deletion contraction decision tree is the same as that in figure 3.1 but has an extra step where we compute the loading of each monomial. The number of monomials in the polynomial now corresponds to the total number of basis of the signed graph as opposed to the number of maximal spanning forests, see corollary 3.4.12.

Theorem 3.4.9. *For a signed graph $\Sigma = (G, \sigma)$ the signed graph Tutte polynomial defined in definition 3.4.7 is equal to the one defined in definition 3.4.8.*

Proof. Let Σ be a signed graph, T^1_Σ the activity signed graph Tutte polynomial from definition 3.4.7 and T^2_Σ the terminal minor signed graph Tutte polynomial from definition 3.4.8. From lemma 3.4.5 the sum over all maximal spanning forest and then over all elements in their loading is equivalent to the sum over the loadings of all the monomials in the extended Tutte polynomial of G. Thus it remains to show that for an element of the loading of a maximal spanning forest $\prod_{e \in \vec{E}} A_B(e)$ is equal to the loaded monomial with the same underlying basis. By definition there is one frustrated edge sent to f or r or no such edge and this is the same edge the monomial loading changes from d or y to f or r or no such edge is modified. Thus they have the same x, c, and f or r edges and thus by lemma 3.1.6 the same set of edges sent to d and to y. Hence the two definition produces the same polynomial. ■

Corollary 3.4.10. *For a signed graph $\Sigma = (G, \sigma)$, $T_\Sigma(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}, f, r) = T_G(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d})$ if and only if Σ is balanced.*

Proof. This result follows directly from lemma 3.4.3 and lemma 3.4.4 and this implies no r or f are present. ■

Figure 3.26 shows that the extended Tutte polynomial of a balanced signed graph is the same as the extended Tutte polynomial of the underlying graph shown in figure 3.1.

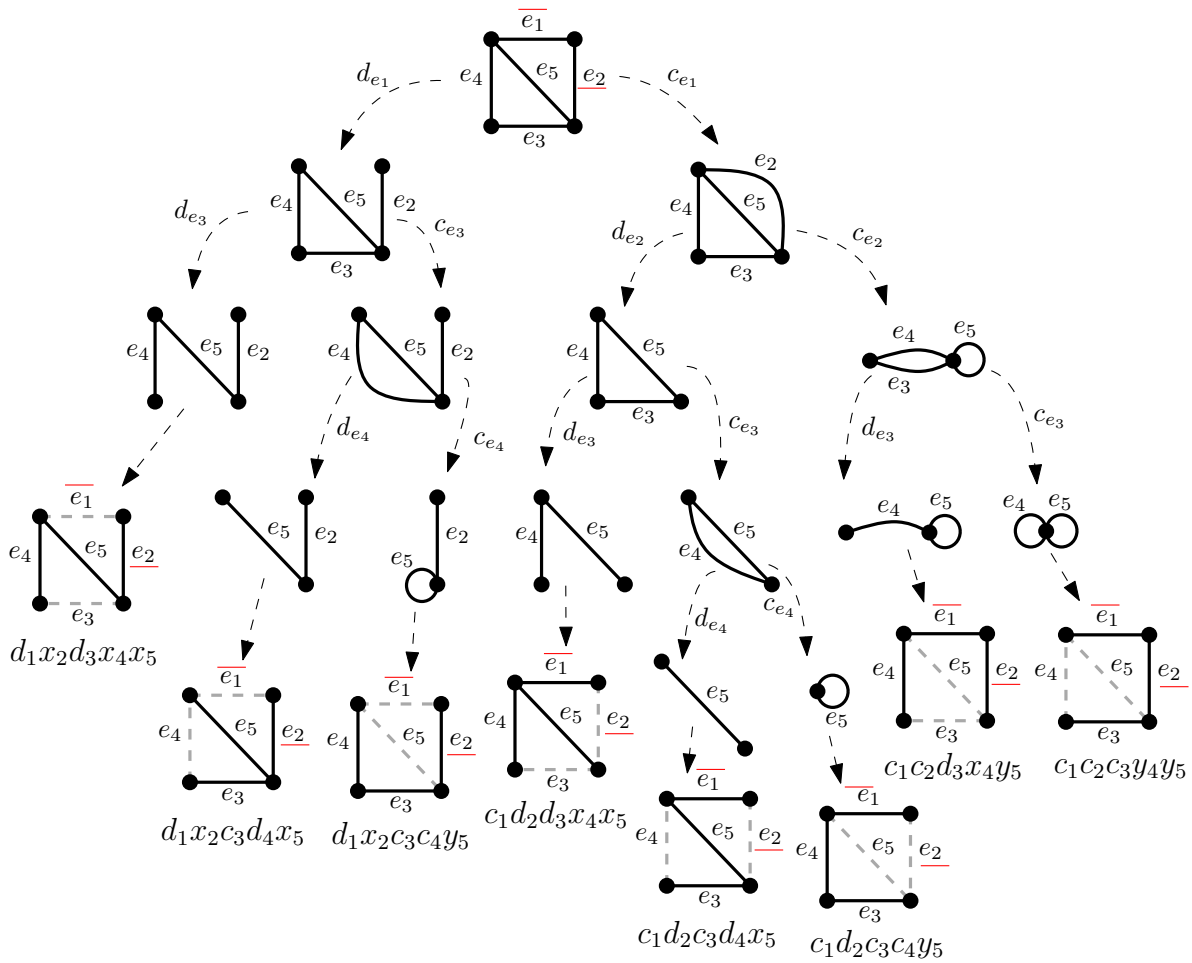


Figure 3.26: The signed graph extended Tutte polynomial of a balanced graph

Definition 3.4.11. *Given a matroid M the* basis generating polynomial *of M with ground set of size n is*

$$g_M(z_1, ..., z_n) = \sum_{B:basis\ of\ M} \prod_{i \in B} z_i$$

Notice that $g_M(1, ...1)$ counts the number of bases of M.

This is a similar notion to the Kirchhoff polynomial where the monomials are maximal spanning forests. Maximal spanning forests are bases for unsigned graphs. For an unbalanced signed graph the bases are maximal spanning forests plus an edge that builds

a negative cycles. The Tutte polynomial for signed graphs presented here counts the edges building a negative cycle as distinct from the maximal spanning forests and thus counts basis with multiplicity based on the size of the negative cycles.

Corollary 3.4.12. *For a signed graph Σ, $T_\Sigma(1,1,1,1,1,1,\overrightarrow{E})$ counts the number of maximal spanning forests if balanced or maximal spanning forests plus an edge that builds a negative cycle if unbalanced.*

Proof. By lemma 3.1.5 each monomial contains the set of edges of a distinct maximal spanning forest. If Σ is balanced then there are no instances of f or r and the result holds. Otherwise each monomial contains exactly one instance of f or r, for which the associated edge forms a negative cycle with respect to the maximal spanning forest formed by the edges associated to the x and c in the monomial by the polynomial construction. ■

Another interesting line of investigation is to study how switching affects the polynomial. For two switching equivalent graphs, $\Sigma = (G,\sigma)$ and $\Sigma' = (G,\sigma')$, the maximal spanning forests are determined by G and since these graphs are switching equivalent they have the same set of negative cycles. Therefore it is not surprising that they produce the same polynomial.

Theorem 3.4.13. *For signed graphs $\Sigma = (G,\sigma)$ and $\Sigma' = (G,\sigma')$, and edge order $\overrightarrow{E}$ of G such that Σ' can be obtain from Σ by switching,*

$$T_\Sigma(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d},\mathbf{f},\mathbf{r};\overrightarrow{E}) = T_{\Sigma'}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d},\mathbf{f},\mathbf{r};\overrightarrow{E}).$$

Proof. Since Σ and Σ' are switching equivalent and $T_G(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E})$ is independent of the signs it remains to argue that the monomial loading of any m in $T_G(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E})$ is the same for Σ and Σ'. This is immediate from the fact that switching preserves the sign of each cycle. So for any d or y edge in m if it forms a negative cycle with respect to maximal spanning forest T_m in Σ then this cycle is also negative with respect to Σ'. So m has the same loading for both graphs and hence the polynomials are equal. ■

Example 3.4.14. In figure 3.27 and figure 3.28 we see two signed graphs that are switching equivalent and that their monomials for each terminal minor are the same.

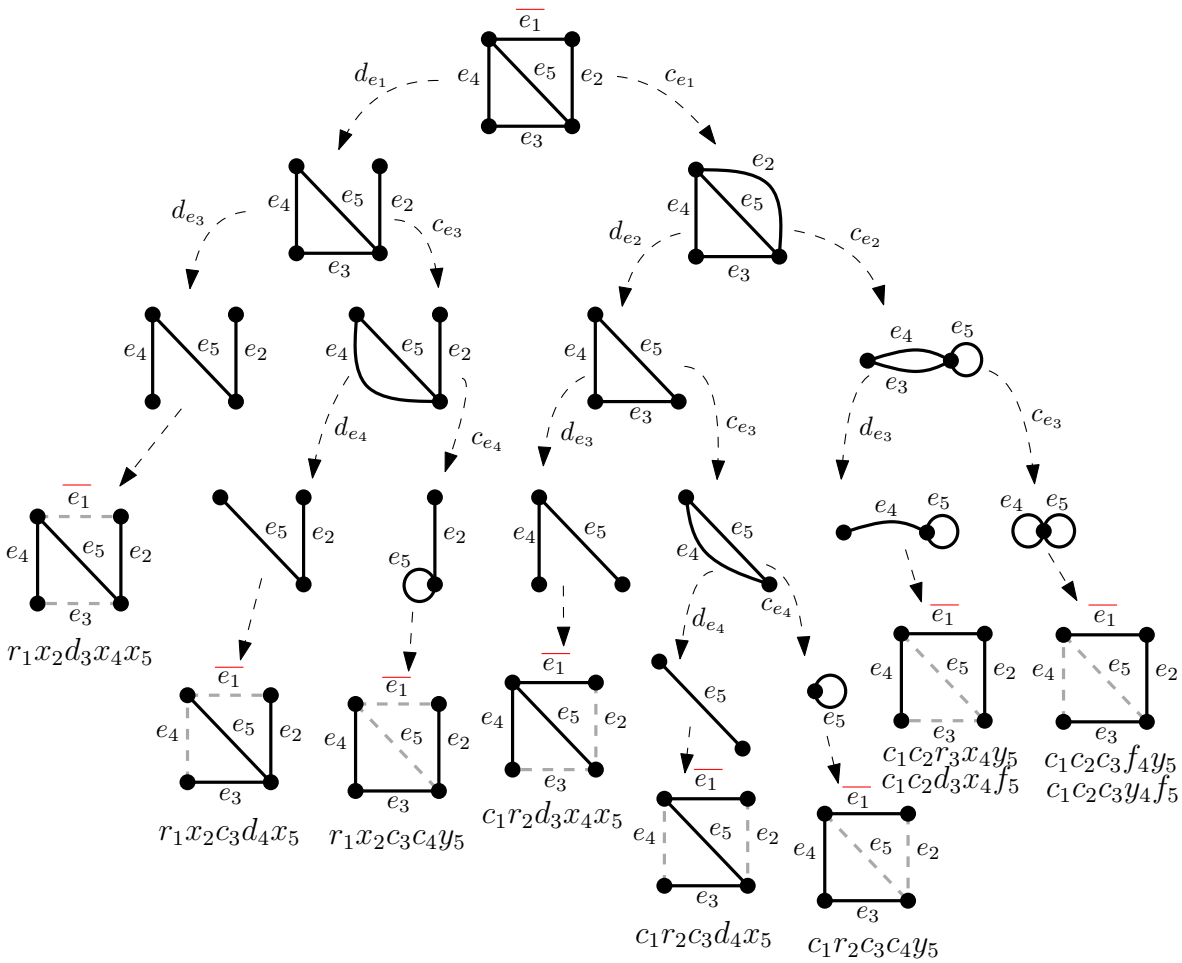


Figure 3.27: The extended Tutte polynomial for signed graphs of a graph with one negative edge

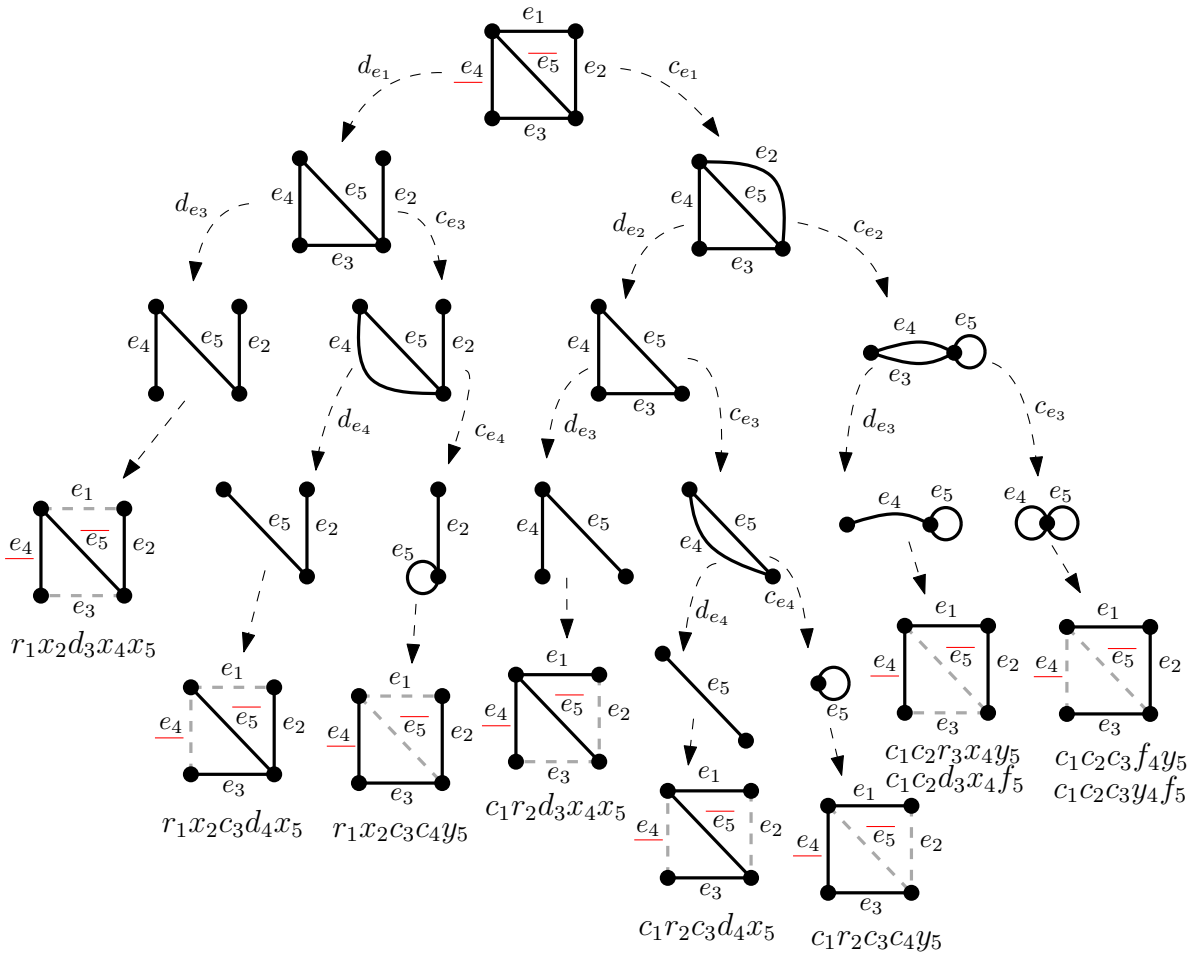


Figure 3.28: The extended Tutte polynomial for signed graphs of a graph with two negative edges

In figure 3.29 we see that this graph is unbalanced as the monomials all have an instance of f or r and further that this graph is not switching equivalent to figure 3.27 or figure 3.28 as the monomials differ. Even the number of bases differ, despite all the graphs having two negative cycles.

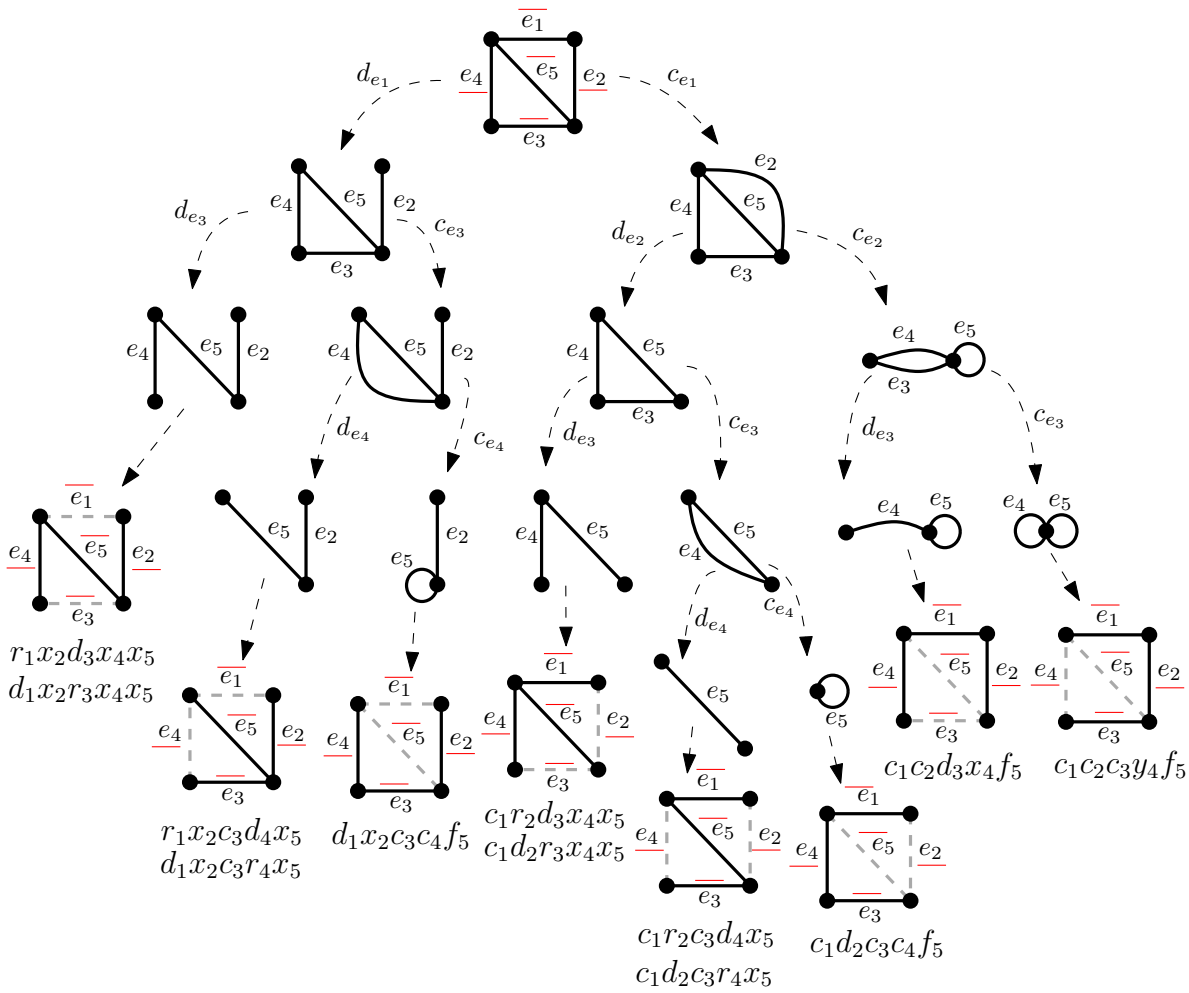


Figure 3.29: The extended Tutte polynomial for signed graphs of a graph with all negative edges

3.4.3 Grid walking

Now that we have added an f and r to our polynomial the Tutte grid walking techniques for the extended Tutte polynomial won't quite work. We can still walk left and right and up and down on the grid, but we need some other kind of step for the frustrated edges. We will fix this issue by having two parallel grids and use the f/r variables to move between the grids. We motivate this choice with the notion of the double covering graph.

We use the notions presented in [136] for double and signed double covering graphs. First we begin with the notion of a double cover. A graph H is a *double covering graph* of G if there exists a graph homomorphism c from H to G that is onto, maps exactly two vertices of H to each vertex of G and for any $v \in V(H)$ the map acts as an isomorphism on the neighbors of v.

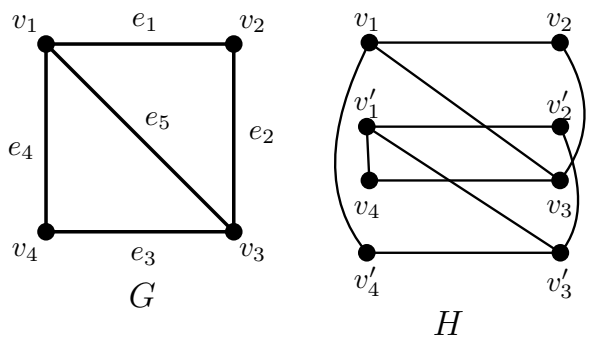


Figure 3.30: A graph G and a double covering graph H

For a signed graph, we say that $\Sigma' = (H, \sigma')$ is a *signed double covering graph* of $\Sigma = (G, \sigma)$ if H is a double covering graph of G with graph homomorphism $c : H \to \Sigma$ and we have $\sigma' : V(H) \to \{+1, -1\}$ such that σ' is a bijection on $c^{-1}(v)$ for all $v \in V(G)$ and $\sigma(c(e)) = \sigma'(u)\sigma'(v)$ where $e = uv \in E(\Sigma')$. This is to say that we consider the pre-images of $v, u \in G$, which both have size two, to have a negative and positive copy of the vertex v and u denoted by u, u^-, v and v^-. If uv is positive in G then we have uv and u^-v^- as edges in H and if uv is negative we have uv^- and vu^- as edges in H. An example can be seen in figure 3.31.

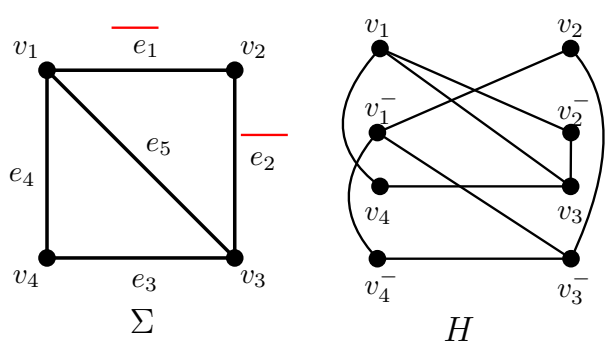


Figure 3.31: A signed graph Σ and its double cover H

We now define Tutte grid walking for a signed graph.

Definition 3.4.15. *For a signed graph $\Sigma = (G, \sigma)$ with ℓ loops, i isthmi, cyclomatic number ϕ, and rank r. Define the* Signed Tutte grid *of Σ to be a pair of Tutte grids U and L, each with height $\phi - \ell$ and width $r - i$ where we label the bottom left node on each grid as $x^i y^\ell$, the top left node on each grid as $x^i y^\phi$, the bottom right node on each grid as $x^r y^\ell$, and the top right node on each grid as $x^r y^\phi$. The two grids are connected such that a node $x^i y^j$ in L is connected to a node $x^i y^{j+1}$ in U. A* signed Tutte grid walk *is defined by a pair of walks. One walk starts at $x^i y^\ell$ on L and the other at $x^r y^\phi$ on U. We will call the walk starting at $x^i y^\ell$ the active walk and the walk starting at $x^r y^\phi$ the inactive walk. For each monomial of $T_G(x, y, c, d, f, r; \overrightarrow{E})$ we address the edges in edge order. Starting with the first edge e if it is an isthmus or loop of G skip it, otherwise if it is x in the monomial the active walk takes a right step, if it is y, the active walk takes a up step, if it is c the inactive walk takes a left steps, and if it is d the inactive walk takes a down step. If it is an r or f, we walk between grids using the $x^i y^{j+1}$ connection. For f, the active walk moves from L to U and for r the inactive walk moves from U to L. We will refer to these f and r edges in the walk as* stall points. *We repeat this process with each edge in order until every edge has been addressed.*

We begin by examining signed Tutte grid walks on balanced signed graphs.

Theorem 3.4.16. *For a signed graph Σ the Tutte grid walks on U and L do not cross to the other grid if and only if Σ is balanced.*

Proof. Σ is balanced by corollary 3.4.10 if and only if there are no f and r in the monomials. So the inactive walk never has an edge sent to r if and only if it remains in U and the active walk never has an edge sent to f if and only if it remains in L. Hence Σ is balanced if and only if there are no Tutte grid walks crossing between U and L. ■

Corollary 3.4.17. *Σ is unbalanced if and only if each walk contains a stall point.*

Proof. By the converse of theorem 3.4.16 Σ is unbalanced if and only if the Tutte grid walks cross between U and L. So every walk contains either f or r. Thus every walk contain a stall point. ■

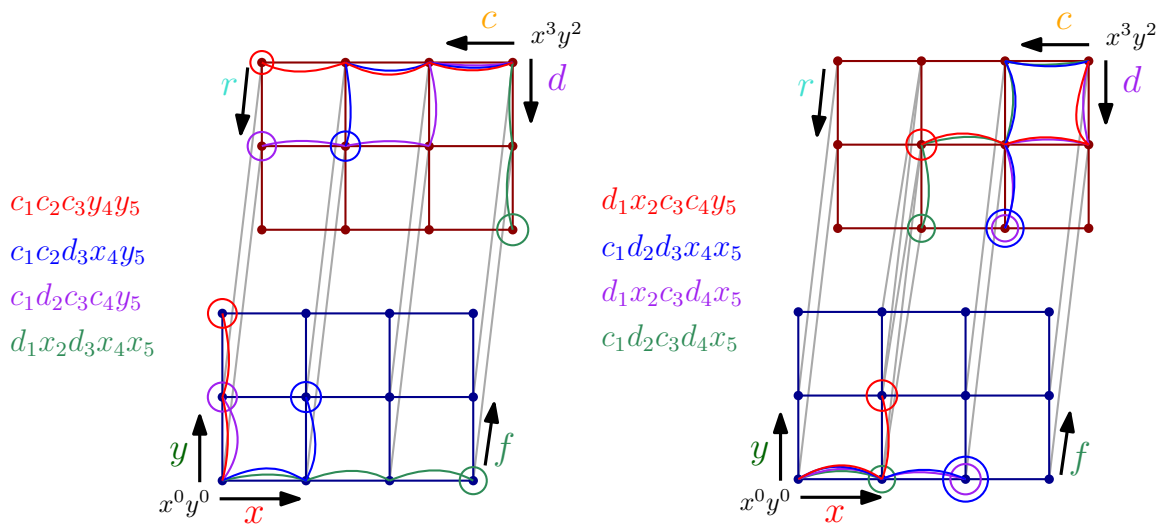


Figure 3.32: Tutte grid walks for a balanced signed graph in figure 3.26

Lemma 3.4.18. *For a balanced signed graph Σ the walk in U and in L end on nodes with the same label.*

Proof. The signed graph Tutte polynomial is determined by the Tutte polynomial of the underlying graph and by lemma 3.3.10 the active and inactive walks end on nodes marked with the same $x^k y^j$ value. ■

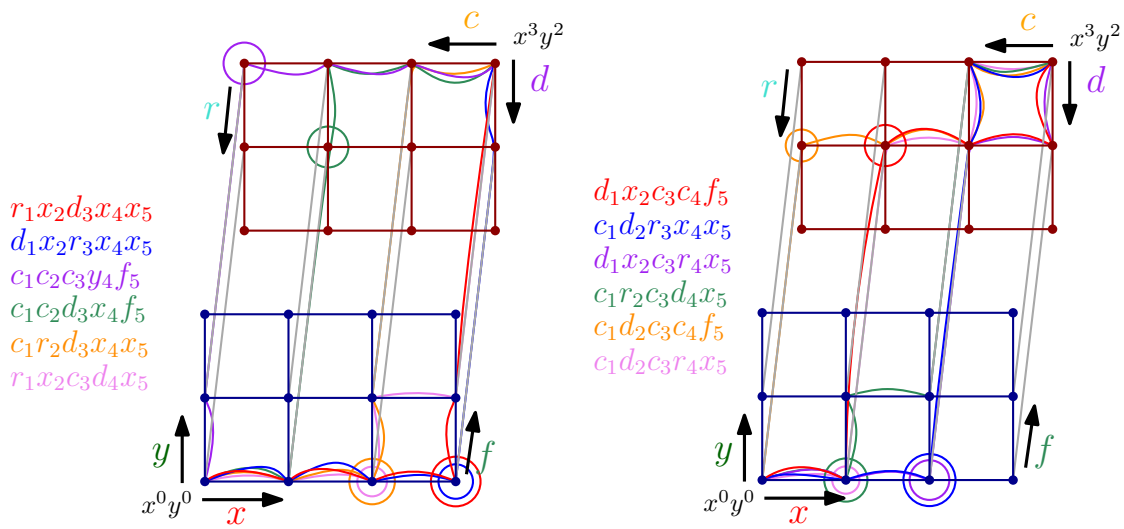


Figure 3.33: Tutte grid walks for an unbalanced signed graph in figure 3.29

Lemma 3.4.19. *For an unbalanced signed graph Σ and monomial of $T_G(x, y, c, d, f, r; \overrightarrow{E})$ the active and inactive walks end on the same node of the same grid, which is in L if the monomial contains r and in U if the monomial contains f.*

Proof. We know from lemma 3.3.10 that the monomials without the f and r end on the same marked node. The presence of f or r causes exactly one of the walks, active or inactive respectively, to move to the other grid. Since the f moves the y value up by one and the r moves the x value down by one in the active and inactive walks respectively, the walks still end on the same $x^k y^j$ node and as exactly one walk switches grids the nodes must also be on the same grid. ■

Corollary 3.4.20. *Two signed graphs $\Sigma = (G, \sigma)$ and $\Sigma' = (G, \sigma')$ have the same set of Tutte grid walks for edge ordering $\overrightarrow{E}$ on G if they are switching equivalent.*

Proof. This follows directly from theorem 3.4.13 as the Tutte grid walks can be determined by the monomials of each polynomial, which are equal. ■

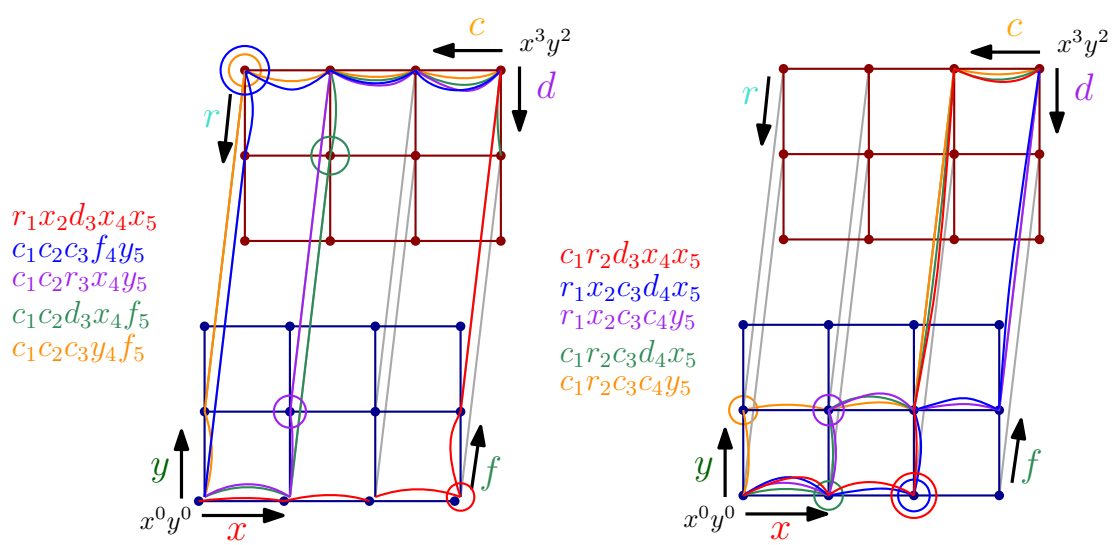


Figure 3.34: Tutte grid walks for switching equivalent signed graphs in figures 3.27 and 3.28

3.5 Bipartite Representations of Graphs

When constructing the Tutte grid walks for signed graphs we were inspired by the notion of double covering graphs. Another related concept is the bipartite representation of a graph since as mentioned in chapter 2 the bipartite representation of a hypergraph is a graph.

Recall that the definition of a bipartite representation doesn't require the edges to have size two and thus naturally extends to hypergraphs. Thus if we can understand the relationship between the Tutte polynomial, or the extended Tutte polynomial and that of its bipartite representation then we can combine that understanding with our signed graph extended Tutte polynomial and have a technique with which to approach oriented hypergraphs. In this section we will explore how the extended Tutte polynomial of the bipartite representation of a graph contains the extended Tutte polynomial of the original graph.

Example 3.5.1. Here we see K_4 minus an edge and its bipartite representation.

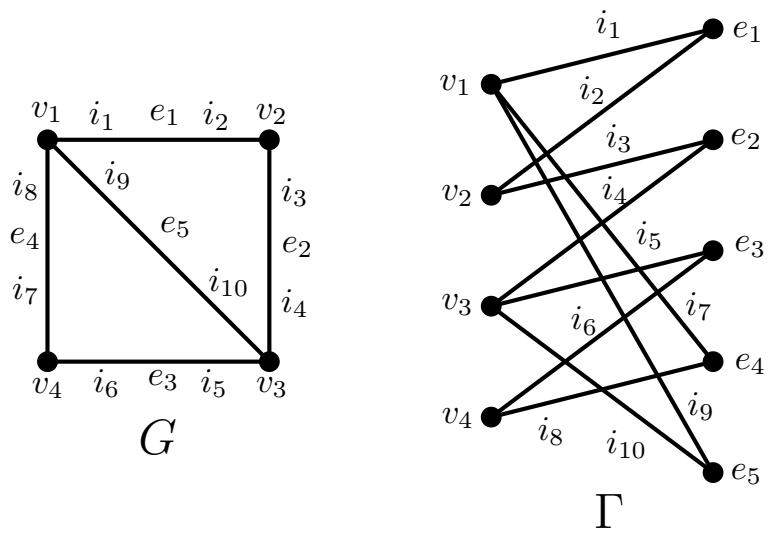


Figure 3.35: A graph and its bipartite representation

Since Γ is a graph itself, we can delete and contract the edges of Γ; i.e. the incidences

of G. The Tutte polynomial for Γ is:

$$\begin{aligned} T_\Gamma(x,y) = {} & x^8 + x^7 + x^6 + x^5 + x^4 + x^3y \\ & + x^7 + x^6 + x^5 + x^4 + x^3 + x^2y \\ & + x^6 + x^5 + x^4 + x^3 + x^2 + xy \\ & + x^5 + x^4 + x^3y + x^2 + x + y \\ & + x^4 + x^3y + x^2 + xy \\ & + x^3y + x^2y + xy + y^2 \end{aligned}$$

Moreover, we can find the Tutte polynomial of G from the extended Tutte polynomial of the bipartite representation Γ by restricting the deletion and contraction of the incidences (edges of Γ) to contraction-first deletion-contraction strings on all incidences associated to an edge before moving on to the next edge.

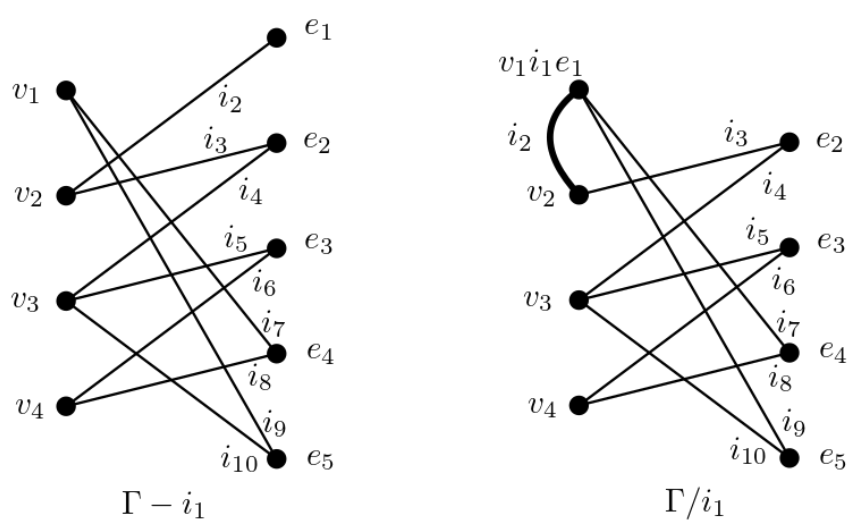


Figure 3.36: Deletion and contraction of i_2 in Γ

We can see that the deletion of i_1 yields a bipartite graph but makes e_1 a 1-edge in a modification of G. Contraction merges a vertex and an edge (so the new Γ is not bipartite!) However if we took the graph obtained by the contraction of i_1 and then deleted i_2 we would once again obtain a bipartite graph.

Observe that in the bipartite representation graph there are $|E|$ more edges and vertices, but no new cycles. Thus, the cyclomatic number does not change, but the number of edges in a maximal spanning forest increases by $|E|$. Furthermore, the number of edges in Γ is twice the number of edges of G since each edge in G has two incidences which are edges of Γ.

Using the extended Tutte polynomial, we delete and contract the incidences according to the edge order from G and are then able to translate between the extended Tutte polynomial of Γ and the Tutte polynomial of G.

Theorem 3.5.2. *Let G be a graph and Γ the bipartite representation of G. Let $\overrightarrow{E} = e_1, e_2, ...e_n$ be an edge ordering on G and $\overrightarrow{E'} = i_1 i_2 .... i_{2n-1} i_{2n}$ be an edge ordering on Γ such that i_{2k-1} and i_{2k} are the incidences associated to e_k in G. Then given $T_\Gamma(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}; \overrightarrow{E'})$ we can find $T_G(\mathbf{X}, \mathbf{Y}, \mathbf{C}, \mathbf{D}; \overrightarrow{E})$ as follows:*

Take each monomial of the noncommutative T_Γ and consider it to be a string of pairs of letters. For each string perform the following substitutions.

1. ***Replace** xx **in** T_Γ **with** X*
2. ***Replace** cy **in** T_Γ **with** Y*
3. ***Replace** cc **in** T_Γ **with** C*
4. ***Replace** cd **in** T_Γ **with** D*
5. ***Replace all other pairs of letter with zero.***

Then $T_G(X, Y, C, D, \overrightarrow{E})$ is the sum of all non-zero monomials after substitution. The Tutte polynomial of G is then obtained by dropping subscripts, setting $C = D = 1$ and allowing X and Y to commute.

Proof. Since Γ has an even number of edges and each monomial has one indeterminate for each edge, and the edge ordering of Γ addresses both incidences of an edge in G in sequence, each string in the Tutte polynomial of Γ can be broken up into consecutive pairs that each correspond to an edge in G. Observe that xx in Γ means that both incidences in the edge in G remain, so the edge remains in G as an isthmus. The string cy in Γ takes the incidences in G and makes both incident to the same vertex in G which creates a loop. The string cc in Γ takes all incidences for an edge in G and contracts them both, which makes the two vertices incident to these edges in G into one vertex — this is contraction

in G. Finally, the string cd in Γ contracts the two incidences of an edge in G to one edge in Γ and then deletes this edge which is the same as deleting both incidence and the edge in G, and this is edge deletion in G.

To see that no other operations create a pair of indeterminates corresponding to an operation in G, observe that the other possible pairs are $xc, xd, xy, yx, yy, yd, yc, cx, dc, dd, dx$, and dy. Since the edges of G determine the order of edges addressed in Γ and each pair of edges share an incident vertex in Γ, both edges must be isthmus or the first of the two edges must be contracted or deleted. Thus xc, xd, xy, yx, yy, yd and yc are not sequences of operations that can appear on two incidences from the same edge of G. The deletion of an edge in Γ leaves a vertex of degree one in the vertices that came from the edges in G so dy cannot occur and both edges cannot be removed so dd cannot occur. Also a degree one vertex cannot be contracted so dc cannot occur. This leaves the string dx. This string is possible in Γ but results in a half edge in G so any term containing this as a substring is not an admissible sequence in G.

We may then apply Corollary 3.1.2 to obtain the Tutte polynomial. ■

The proof of this lemma relies on the fact that the non-commutative polynomial provides a fixed order on the letters in a monomial, leading to a fixed string of pairs of letters as well as the edge ordering ensuring that each pair of letters corresponds to a single edge in the original graph. Thus, we can now convert between the Γ monomials and G monomials. This can be shown for the two following examples.

Example 3.5.3. Consider the graph G and its bipartite representation from Example 3.35. We discuss two monomials from the 4-variable Tutte polynomial (which is not in full presented here due to length as there are thirty two spanning trees).

The Γ monomial $cdxxcdxxxx$ (an x^6 monomial in the Tutte polynomial) corresponds to a monomial of G as every edge of G has two incidences and the monomial can be partitioned as $cdxxcdxxxx = DXDXX$, which is the first term in the original example.

The Γ monomial $dcxxcdxxxx$ (another x^6 monomial in the Tutte polynomial) would NOT correspond to G as the initial pair in the string dc is not allowed.

We can create more generalized statement about connection between the number of maximal spanning forests in a graph G and its bipartite representation Γ by considering equivalent strings of c, d, x, y in Γ. From theorem 3.5.2, the x and c both build maximal spanning forests, while y and d do not. So when we we have a consecutive pair of incidences in Γ that correspond to the same edge we may swap cd with dx and cy with cd. To build the maximal spanning forests, these are precisely the variable exchanges that do not alter

the trees and we can further choose these exchanges as we know we are modifying a pair of incidence in sequence, which come from the same edge in G. In other words, the string cd chooses one of two incidences but it is not active, while dx chooses one of two incidences and it is active. Considering these replacements arose as a consequence of examining how different sequences of choices impacted the polynomial and whether given a monomial we could produce other valid monomials through knowing when we could change letters associated to edges and how pairs of letters where related in the polynomial on the bipartite representation to the polynomial of the original graph. These relationships relate to the following formula on terminal minor counts, hence, maximal spanning forests between Γ and G.

Lemma 3.5.4. *Let G be a connected graph with bipartite representation Γ, with cyclomatic number ϕ, and tree-number τ, then*

$$\tau(\Gamma) = \tau(G)2^{\phi}$$

Proof. Given a spanning tree T of G this tree corresponds to a tree that is a subgraph of Γ which spans all vertices of Γ except those associated to the edges of G not in the tree. There are two edges in Γ incident to each of these vertices and there are ϕ such vertices for each spanning tree. So each spanning tree of G produces 2^{ϕ} spanning trees in Γ. To see no other spanning trees of Γ exist, consider T an arbitrary spanning tree of Γ. Deleting any vertices in E of degree one we obtain a graph which is the bipartite representation of some subgraph F of G. This graph contains no cycles, so it is a spanning forest of G since no vertices in V were deleted. To see F is a spanning tree suppose not. Then F has at least two components. Any edge between these two components in G was deleted from T. Then for any vertices a and b in different components of F the path between them in T used a vertex of degree one. This is not possible, so no such a and b exist and F has one component. So F is a spanning tree and any spanning tree of Γ contains a subgraph which is the bipartite representation of a spanning tree of G plus one half edge for each other edge of G. Hence there are $\tau(G)2^{\phi}$ total spanning trees of G. ■

Lemma 3.5.5. *For a connected graph G and bipartite representation Γ if r is the rank of G and ϕ is the cyclomatic number then Γ has rank $2r + \phi$.*

Proof. As seen in the proof of lemma 3.5.4 each spanning tree of Γ comes from the bipartite representation of a spanning tree of G plus one edge for each edge of G not in the spanning tree of G. So each edge in the spanning tree has two incidences which gives two edges in Γ and we add ϕ additional edges. Hence each spanning tree of Γ has size $2r + \phi$. ■

3.5.1 Grid Walking

Just as we can construct Tutte grid walks for graphs, we can construct Tutte grid walks for the bipartite representation of a graph. Using the results in theorem 3.5.2 we can relate the grids of G and Γ_G.

Definition 3.5.6. *For a graph G with i isthmi, cyclomatic number ϕ, rank r, and bipartite representation Γ. Notice that Γ does not have loops. Define the* Tutte grid *of Γ to be grid with height ϕ and width $2r - 2i + \phi$ where we label the bottom left node as $x^i y^0$, the top left node as $x^{2i} y^\phi$, the bottom right node as $x^{2r+\phi} y^0$, and the top right node as $x^{2r+\phi} y^\phi$. A* Tutte grid walk *is defined by a pair of walks. One walk starts at $x^{2i} y^0$ and the other at $x^{2r+\phi} y^\phi$. We will call the walk starting at $x^{2i} y^0$ the active walk and the walk starting at $x^{2r+\phi} y^\phi$ the inactive walk. For each monomial of $T_G(x, y, c, d; \overrightarrow{E})$ we consider each edge in edge order. Starting with the first edge e, if it is an isthmus or loop of Γ skip it, otherwise if it is x in the monomial the active walk takes a right step, if it is y, the active walk takes a up step, if it is c the inactive walk takes a left steps, and if it is d the inactive walk takes a down step. We repeat this process with the next edge in the edge ordering until all edges have been used.*

Example 3.5.7. Here we see the grid of Γ is $|E| = 5$ wider than the grid for G, but the same height. The grid for G overlays using the cc and xx steps as these steps are not affected by the seperation of active and inactive edges into two seperate walks.

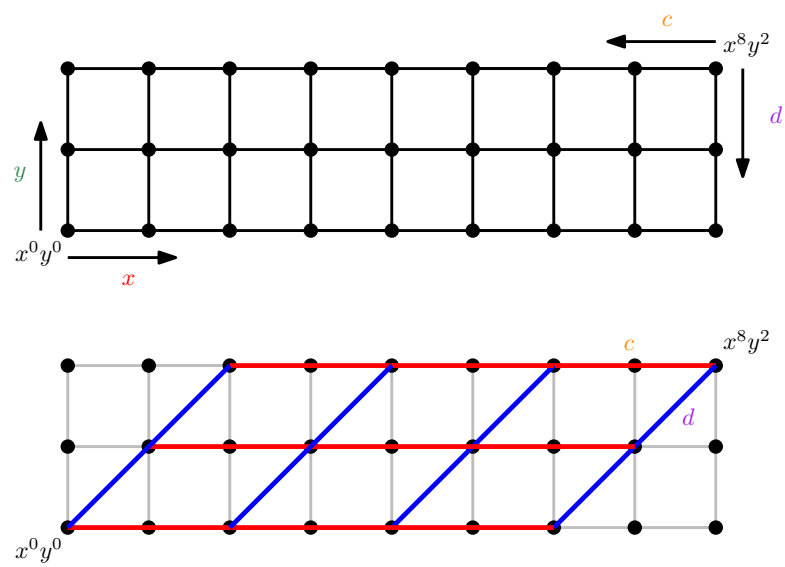


Figure 3.37: The Tutte grid for Γ and the overlay of the Tutte grid of G

The natural deletion and contraction steps are shown in blue and red, while the replacement steps are not depicted. We consider steps on the bipartite representation grid to be inside the Tutte grid of G overlayed if they do not pass below the furthest right d/y steps or above the furthest left d/y steps. They do not have to lie on top of the grid for G exactly within these bounds. This can provide a way to visualize how some terms of the extend Tutte polynomial of the bipartite representation cannot correspond to monomials of the extended Tutte polynomial of the original graph.

Theorem 3.5.8. *Given a graph G and its bipartite representation Γ, the Tutte grid of G fits onto Tutte grid of Γ by slanting the Tutte grid of G and stretching it so that one horizontal step is two steps in the Tutte grid of Γ and one vertical step is the diagonal through a square in the Tutte grid of Γ, or both a horizontal and vertical step.*

Proof. This follows directly from theorem 3.5.2 as cd and cy or a horizontal and vertical step correspond to a vertical step d or y in G. This slants the grid. Also xx and cc corresponds to a horizontal step x or c which stretches the grid. ■

Lemma 3.5.9. *For a graph G and its bipartite representation Γ if a Tutte grid walk comes from a monomial of Γ which does not go to zero when computing the monomials of the extended Tutte polynomial of G from that of Γ, then the Tutte grid walk does not step below the furthest right d/y steps of the grid of G or above the furthest left d/y steps of the Tutte grid of G when overlayed on the Tutte grid of Γ.*

Proof. To pass outside of the Tutte grid of G overlayed on the Tutte grid of Γ, the walk must contain consecutive deletions or consecutive instances of y, or the monomial must start with a pair of letters which begin with a d. No such consecutive letter or starting letter is allowed in a monomial of Γ that is not sent to zero. ■

Figure 3.38 shows the Tutte grid walks in Γ from figure 3.35 for both a monomial that corresponds to a monomial and Tutte grid walk of G and one that does not. The Tutte grid for G and its corresponding walk is overlayed.

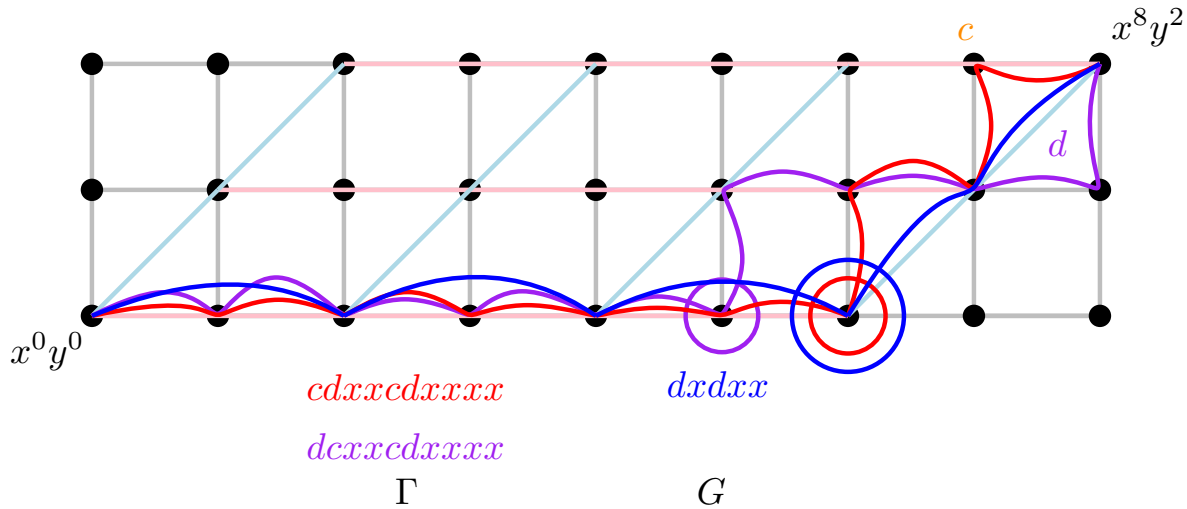


Figure 3.38: Tutte grid walks on G overlayed with Tutte grid walks on Γ

These results leave us with many open avenues of investigation, many of which will be discussed in chapter 5, but these including the following. First there is the open question of whether we can determine the Tutte grid walks of G from those of Γ. We have much more limited edge ordering choices on Γ if we want to have our Tutte grid walks relate to G, but perhaps further investigation on how the edge ordering affects the Tutte grid walks of G and Γ and their relationship would help with understanding how the shuffle of the x and y out of c and d is affected by the edge ordering. We also know that because our Γ edge ordering depends on G we may sometimes be able to replace cd with dx and cy with dx. Perhaps the Tutte grid walks themselves or the relationship to those of G can help narrow down when such replacements are allowed.

Chapter 4

Interlace Polynomial

We define the following notation in order to simplify the statements of results. In the Martin polynomial recursive definition for any $v \in V(G)$, let G'_v be the graph obtained by resolving about the vertex by following E. Let G''_v be the graph obtained by resolving about v in the unique orientation-consistent manner other than G'_v. Finally, let G'''_v be the graph obtained by resolving v in an orientation-inconsistent manner relative to E as seen in figure 4.1. The only vertex resolution that might take E and separate it into multiple disconnected circuits is the operation coming from G''_v as the resolution to G'_v simply removes a vertex v but leaves the circuit essentially unchanged and the resolution G'''_v twists the circuit about v but cannot disconnect E as we follow a portion of the circuit E in the opposite orientation but still pass through v to the rest of the circuit, but if v is a cut vertex this still preserve the new circuit traveling across where the cut vertex was. However G''_v breaks at v and the circuit basically avoids where v was and this can break the circuit if v was a cut vertex by never passing through v to the other portion of the circuit.

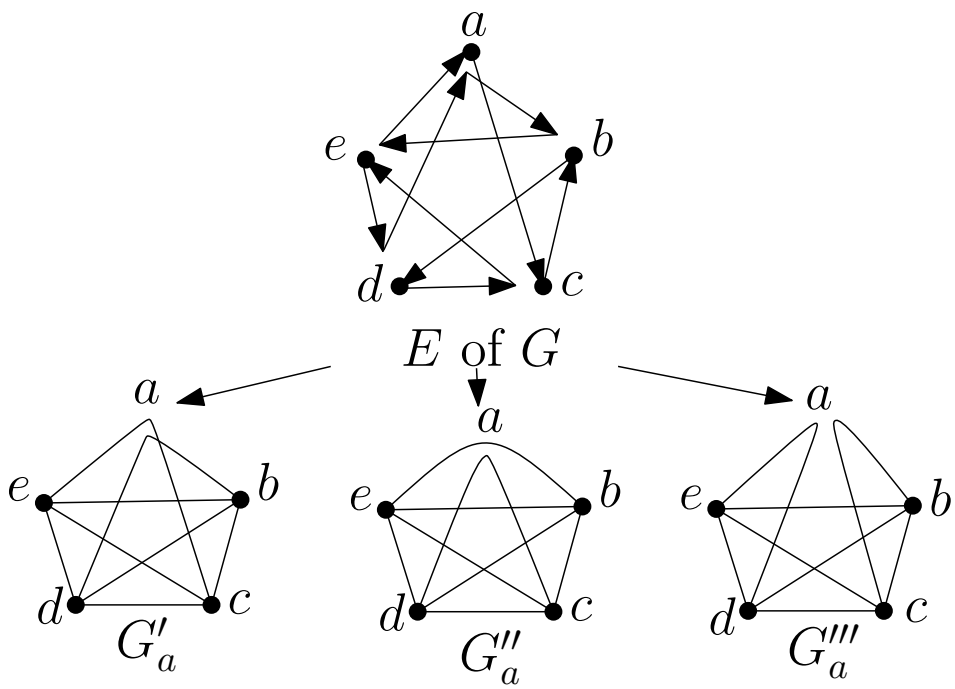
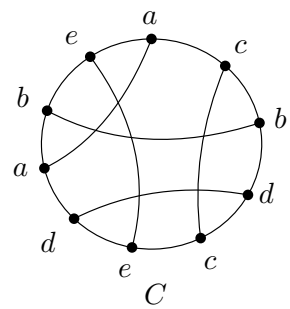


Figure 4.1: The ways to resolve a vertex with respect to a chosen Eulerian circuit

Define w *the word of a chord diagram* to be a double occurrence word of the diagram's labels starting at any label and proceeding clockwise. Note a chord diagram will have multiple associated words that will be cyclic permutations of each other but we will abuse terminology by saying "the word" of a chord diagram, often assuming to start at an instance of a specified letter. In general we are concerned with a pair of crossing chords for vertices $u, v \in C$ and we may write the word of the chord diagram starting at an instance of v and proceeding clockwise as $vw_1uw_2vw_3uw_4$ where w_i is the clockwise list of all vertex labels between the specified vertices. See figure 4.2 for an example. We will consider two chord diagrams to be equivalent if they have the same word up to cyclic permutation or reflection (dihedral symmetries) and we will choose to write the word of the chord diagram starting at an instance of the vertex we wish to resolve when possible then proceeding clockwise.



$$acbdcedabe$$
$$w_1 = cbdc \qquad w_2 = d$$
$$w_3 = b \qquad w_4 = \epsilon$$
$$aw_1ew_2aw_3ew_4$$

Figure 4.2: A word of a chord diagram

Define $\mathfrak{C}_{\overrightarrow{E}}(G) = C$ to be the function that takes a 4-regular graph G and any Eulerian circuit on G, $\overrightarrow{E}$, and outputs the associated chord diagram C. Define ω to be the function that takes a double occurrence word to a chord diagram. Finally, following the notation for the relationship of chord diagrams and interlace graphs, $I(C)$ denotes the interlace graph of the chord diagram C. Given a double occurence word $l_1...l_i...l_j...l_n$ we may take any sub-word $l_i...l_j$ and reverse this portion of the word to obtain the word $l_1...l_j...l_i...l_n$. For a word written as a concatenation of sub words $w_1w_2...w_n$ with $w_i = l_jl_{j+1}...l_k$ we denote the reverse of w_i as $\overleftarrow{w_i}$. The following theorem shows how we can relate transitions at a vertex in a 4-regular graph with operations on a chord diagram and local complementation operations on an interlace graph. In [20, 21], Bouchet discusses the relationships between Eulerian circuits, interlace graphs, and chord diagrams. However, no complete proof of any relationship is given so we present our own detailed proof of the result.

Theorem 4.0.1. *For a 4-regular graph G, Eulerian circuit $\overrightarrow{E}$ and associated chord diagram C and interlace graph $I(C)$ with $v, u \in V(G)$ and $vu \in E(I(C))$ and the word of C being $vw_1uw_2vw_3uw_4$ the following hold.*

- $I(\mathfrak{C}_{\overrightarrow{E}}(G'_v)) = I(\omega(\omega^{-1}(C \setminus \{v\}))) = I(C) \setminus v.$
- $I(\mathfrak{C}_{\overrightarrow{E}}(G''_v)) = I(\omega(u\overleftarrow{w_1}\overleftarrow{w_2}u\overleftarrow{w_3}\overleftarrow{w_4})) = (I(C) * uv) \setminus v.$
- $I(\mathfrak{C}_{\overrightarrow{E}}(G'''_v)) = I(\omega(w_1uw_2\overleftarrow{w_4}u\overleftarrow{w_3})) = ((I(C) * v) \setminus v.$

Proof. Since $uv \in E(I(C))$, the vertices u and v are interlaced by the Eulerian circuit. So we may write the word of the chord diagram as $vw_1uw_2vw_3uw_4$. We begin by showing that $\mathfrak{C}_{\overrightarrow{E}(G'_v)} = \omega(\omega^{-1}(C \setminus \{v\}))$. Observe that G'_v is constructed by resolving v in the unique manner determined by $\overrightarrow{E}$. This is the same as traveling the circuit as if v is not present which corresponds to deleting v from the double occurrence word and associated chord diagram. So $\mathfrak{C}_{\overrightarrow{E}}(G'_v) = \omega(\omega^{-1}(C \setminus \{v\}))$ and hence $I(\mathfrak{C}_{\overrightarrow{E}}(G'_v)) = I(\omega(\omega^{-1}(C \setminus \{v\})))$.

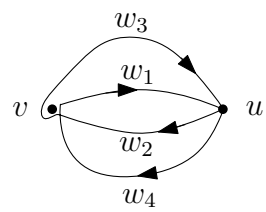


Figure 4.3: How a vertex resolution G'_v impacts the word of a chord diagram

To see that $I(\mathfrak{C}_{\overrightarrow{E}}(G''_v)) = I(\omega(u\overleftarrow{w_1}\overleftarrow{w_2}u\overleftarrow{w_3}\overleftarrow{w_4}))$ observe that G''_v is obtained by resolving $\overrightarrow{E}$ in the orientation-consistent manner other than G'_v. This new orientation follows the circuit through w_1uw_2 and then sends the last edge of w_2 through v back to the start of w_1 and the last edge of w_4 is sent to the first edge of w_3 through v. Now we have two circuits, w_1uw_2 and w_3uw_4 which intersect at u and so we use this intersection to change our original circuit at u to obtain one complete circuit. We can write these two words as $u\overleftarrow{w_1}\overleftarrow{w_2}$ and $u\overleftarrow{w_3}\overleftarrow{w_4}$. We can now send the first letter of w_2, which is the last letter of $\overleftarrow{w_2}$, to the last letter of w_3, which is the first letter of $\overleftarrow{w_2}$, through u which modifies the original circuit at u but gives us a new circuit with word $u\overleftarrow{w_1}\overleftarrow{w_2}u\overleftarrow{w_3}\overleftarrow{w_4}$. So $I(\mathfrak{C}_{\overrightarrow{E}}(G''_v)) = I(\omega(u\overleftarrow{w_1}\overleftarrow{w_2}u\overleftarrow{w_3}\overleftarrow{w_4}))$ as desired.

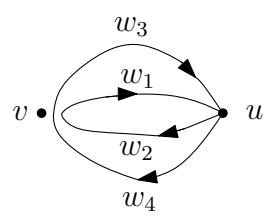


Figure 4.4: How a vertex resolution G''_v impacts the word of a chord diagram

Now consider G'''_v. This is obtained by resolving v in a manner inconsistent with the orientation of $\overrightarrow{E}$. So following the circuit through v we pair the last edge of w_2 which

enters v with the last edge of w_4 which also enters v and we pair the first edge of w_1 and w_3 as these both enter v. So starting at the first edge of w_1 we can write the circuit as $w_1uw_2\overleftarrow{w_4}u\overleftarrow{w_3}$. Thus $I(\mathfrak{C}_{\overrightarrow{E}}(G_v''')) = I(\omega(w_1uw_2\overleftarrow{w_4}u\overleftarrow{w_3}))$.

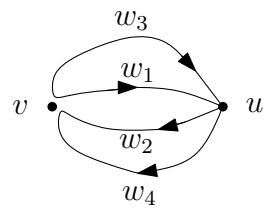


Figure 4.5: How a vertex resolution G_v''' impacts the word of a chord diagram

We will now show that vertex deletion and the local complementation operations on $I(C)$ produce the same graphs as the interlace graphs on the new words and associated chord diagrams. First notice that $I(\omega(\omega^{-1}(C \setminus \{v\})) = I(\omega(w_1uw_2w_3uw_4)) = I(C \setminus \{v\}) = I(C) \setminus \{v\}$ as the removing all instance of v from a chord diagram removes its chord and thus remove the vertex and incident edges in $I(C)$.

To see $I(\omega(u\overleftarrow{w_1}v\overleftarrow{w_2}u\overleftarrow{w_3}v\overleftarrow{w_4}) = (I(C) * uv) \setminus v$ observe that swapping u and v in the word takes all vertices interlaced with just u or just v and interlaces them with the other. Now we consider what reversing each words does. Any two letters which each have an instance in the chord diagram in the same word and their other instance in different words will be interlaced if they were not previously or become noninterlaced if they previously were. If a letter has both instances in the same word it will not have any alterations made to what other letters it interlaces with. If two letters each have an instance in one word and their other instances in the same word they will remain interlaced or not as both words are reversed. So all vertices interlaced with both u and v have one one instance in w_2 and the other in w_4 or one instance in w_1 and the other in w_3 and so the set of vertices in $I(C)$ of $\{u, v, N(u) \cap N(v)\}$ has no internal edges altered. Any vertex interlaced with just u becomes interlaced with v and will swap all vertices it is interlaced with in $N(u) \cap N(v)$ and $N(v) \setminus N(u)$. The same relation with u and v reversed holds for v. Then v is deleted. This is the exact operation defined by $(I(C) * uv) \setminus v$. Since this process is reversible given $(I(C) * uv)$ we may construct the chord diagram with the desired word. Hence $I(\omega(u\overleftarrow{w_1}v\overleftarrow{w_2}u\overleftarrow{w_3}v\overleftarrow{w_4}) = (I(C) * uv) \setminus v$.

Finally we will show that $I(\omega(w_1uw_2\overleftarrow{w_4}u\overleftarrow{w_3})) = ((I(C) + v) * v) \setminus v$. Suppose x is a vertex of $I(C)$. Then either $xv \in E(I(C))$ or $xv \notin E(I(C))$, If xv is not an edge then both instances of x in the word appear in w_1uw_2 or in w_3uw_4. So the reversal of w_3uw_4 will not

affect the vertices interlaced with x. This is due to the fact that at least on end of the interlaced vertex and the intersection with the chord of x will remaining in the half of the word that contains both instances of x and in that half of the word the reversal or lack their of locally makes no change. Now we consider x such that xv is an edge. In this case one end of the chord of x appears in w_1uw_2 and the other appears in w_3uw_4. Now consider y another vertex in $I(C)$ such that yv is an edge. Either $xy \in E(I(C))$ or $xy \notin E(U(C))$. If xy is an edge then reversing w_3uw_4 uncrosses the chords of x and y in C. Similarly if xy is not an edge then reversing w_3uw_4 crosses the chords of x and y in C. This is by construction the operation of local implementation at v in $I(C)$. Given $((I(C))+v)*v)\setminus v$ and C we may observe that this operation is simply crossing all uncrossed chords that cross the chord of v in C and uncrossing all crossed chords that cross the chord of v, which is equivalent to reversing the order of all labels between two instances of v in C in one of the two parts of the circle partitioned by the chord of v. Choosing the w_3uw_4 part we obtain the desired word upon reversal and deletion of v. Hence $I(\omega(w_1uw_2\overleftarrow{w_4}u\overleftarrow{w_3})) = ((I(C) * v) \setminus v$ as desired. ■

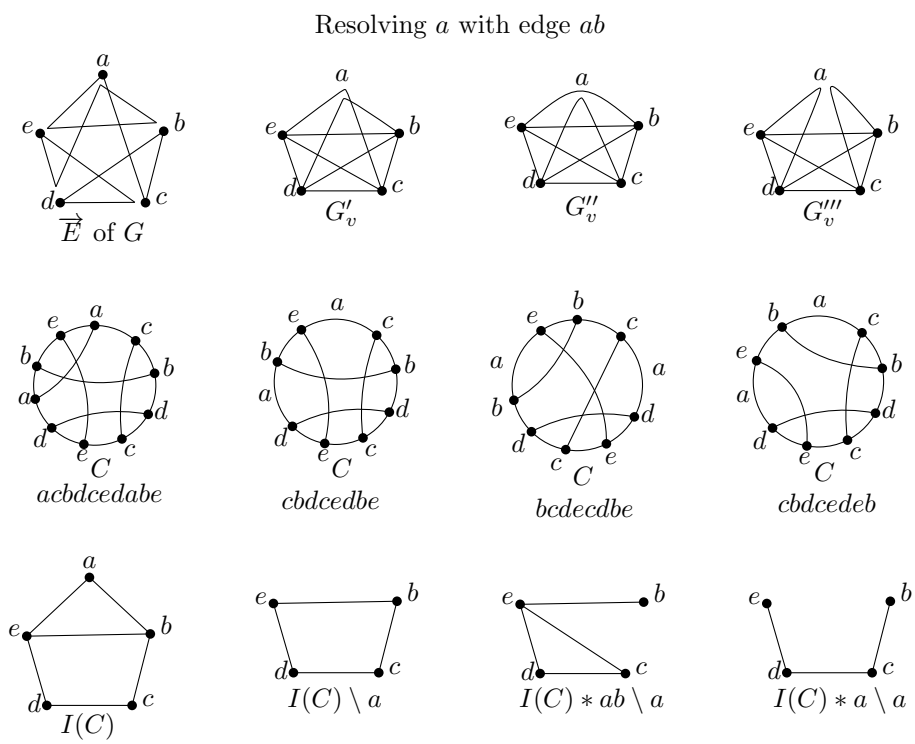


Figure 4.6: The correlation between vertex transitions, chord diagrams, and interlace operations

While the relationship between the words of the chord diagrams and vertex resolutions has been proven, it is not intuitively clear why these operations and word rearrangements correspond, in all but the deletion case. When we resolve a vertex v with respect to an Eulerian circuit $\overrightarrow{E}$ for a 4-regular graph we saw how we could view the resolutions as passing through $\overrightarrow{E}$ as if v was not present for G'_v, as twisting $\overrightarrow{E}$ as it passed through v to obtain G''_v, or as avoiding v which may break $\overrightarrow{E}$. So given a chord diagram C and a pair of interlaced vertices v and u we may take the word of the chord diagram to be $vw_1uw_2vw_3uw_4$. The *skip* of v is the operation of deleting v which results in a chord diagram with word $w_1uw_2w_3uw_4$. The 1-*twist* of v about u is defined by taking the segment of the word from vw_3uw_4 and reversing it and then skipping v to obtain the chord diagram with word $w_1uw_2\overleftarrow{w_4}u\overleftarrow{w_3}$. Finally, the 1-*break* of v with gluing u is obtained by taking the word and splitting it at v and removing v to obtain words w_1uw_2 and w_3uw_4. These words are permuted to start at u giving words $u\overleftarrow{w_1}\overleftarrow{w_2}$ and $u\overleftarrow{w_3}\overleftarrow{w_4}$ and glued at u to obtain the chord diagram with word $u\overleftarrow{w_1}\overleftarrow{w_2}u\overleftarrow{w_3}\overleftarrow{w_4}$. Clearly these operations on the words result in chord diagrams corresponding to the vertex resolutions and deletion and local complementation on the interlace graph. None of these operations explicitly require the words to only have two occurrences of each letter and in section 4.3 we will see how these operations can be extended to k-occurrence chord diagrams.

While we choose to work with connected graphs it is not true that for any choice of Eulerian circuit on a connected 4-regular graph the associated interlace graph is connected. See figure 4.7. This is not an issue as the polynomials are all multiplicative over connected components and the polynomials' relationship is not dependent on connectivity [4]. We simply make the choice to work with connected graphs as having multiple components results in an interlace polynomial where the coefficient of y is zero, which yields an uninteresting interlace invariant of zero, which will be proven in section 4.1. As the interlace polynomial on interlace graphs is independent of the choice of Eulerian circuit, if one choice of Eulerian circuit produces a disconnected interlace graph, they all will.

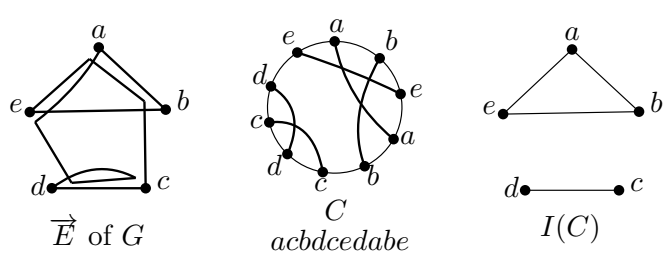
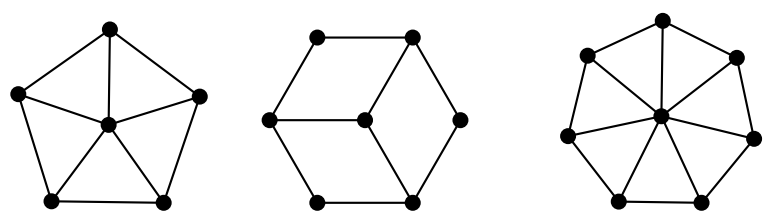


Figure 4.7: A choice of Eulerian circuit producing a disconnected $I(C)$

Given the relationship between the interlace and Martin polynomials we examine how we can extend results on the Martin invariant to the interlace invariant in sections 4.1 and 4.2.

4.1 Interlace Invariant Special Cases

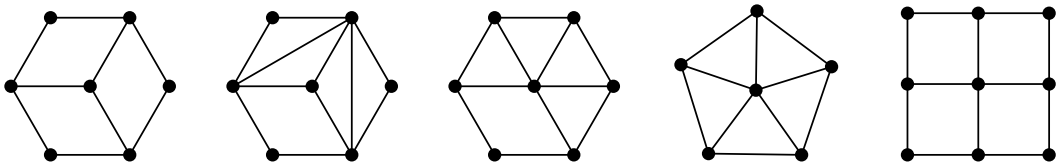
To investigate the Martin invariant on graphs through the interlace polynomial, we began by computing $I'(C)$ for various graphs, including those containing the forbidden vertex minors as the interlace invariant depends only on the interlace polynomial of a graph which is defined for all simple graphs. While not known for the global interlace polynomial, which we are referring to as the interlace polynomial here, other versions of the interlace polynomial have explicit formulas for complete graphs, paths, cycles, and complete bipartite graphs [4, 7, 8]. We obtain an explicit formula for the interlace polynomial for complete graphs as well as explicit formulas for the interlace invariant on these classes of graphs. Of these classes of graph, all but cycles have very similar formulas for the interlace invariant. A formula for the interlace invariant of a cycle has none the less been obtained. We also obtain a new recursive formula for the interlace polynomial of trees, though they must be arbitrarily rooted first.



F_1 F_2 F_3

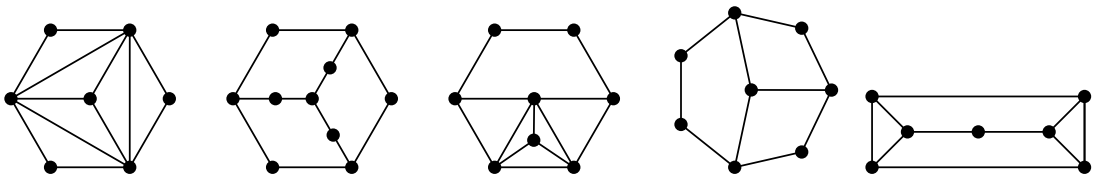
$Q(F_i;y)$	$45y^2+108y$	$y^4+8y^3+138y^2+216y$	$42y^3+399y^2+612y$
$I'(F_i)$	18	36	102

Table 4.1: Interlace forbidden vertex minors and their interlace polynomials and invariants



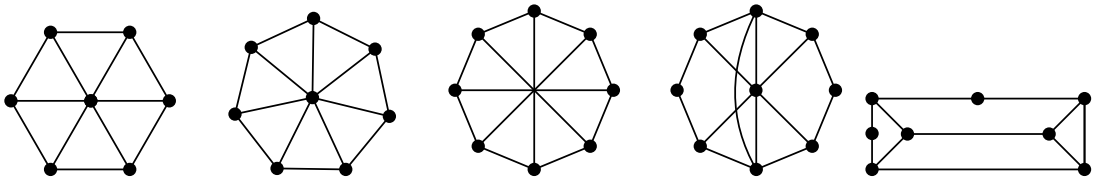
F_1 F_2 F_3 F_4 F_5

$Q(F_i;y)$	$y^4+8y^3+138y^2$ $+216y$	$y^4+8y^3+138y^2$ $+216y$	$9y^3+136y^2+240y$	$45y^2+108y$	$y^5+9y^4+197y^3$ $+1024y^2+1392y$
$I'(F_i)$	36	36	40	18	232



F_6 F_7 F_8 F_9 F_{10}

$Q(F_i;y)$	$y^4+8y^3+138y^2$ $+216y$	$y^6+8y^5+81y^4+607y^3$ $+2634y^2+3240y$	$3y^4+48y^3+374y^2$ $+522y$	$3y^4+48y^3+374y^2$ $+522y$	$9y^3+136y^2+240y$
$I'(F_i)$	36	540	87	87	40



F_{11} F_{12} F_{13} F_{14} F_{15}

$Q(F_i;y)$	$9y^3+136y^2$ $+240y$	$42y^3+399y^2+612y$	$42y^3+399y^2+612y$	$4y^4+185y^3+1084y^2$ $+1536y$	$y^4+49y^3+377y^2$ $+588y$
$I'(F_i)$	40	102	102	256	98

Table 4.2: Interlace graph edge-local complementation obstructions with their interlace polynomials and invariants

Before we prove anything for special classes of graphs we begin with a few lemmas on the properties of the interlace invariant that will be used in many of the proofs. Namely we show that the interlace invariant is zero for disconnected graphs, satisfies the same recurrence as the interlace polynomial, and is a non-negative integer if $|V(G)| \geq 3$.

Lemma 4.1.1. *If G is a disconnected graph then $I'(G) = 0$.*

Proof. The interlace polynomial is multiplicative over connected components. Since G is disconnected, G has at least two components. Since every component of G, using the

interlace recursive definition, has no constant term and thus has one as the smallest possible degree of y, $I(G)$ has smallest degree of a term y^k where $k \geq 2$ is the number of connected components of G. So $I'(G) = \frac{1}{6}(Q'(G))|_{y=0} = 0$. ■‘

Lemma 4.1.2. *If G is a graph with at least two vertices u and v with edge uv then $I'(G) = I'(G - v) + I'(G * v - v) + I'(G * uv - v)$.*

Proof. For the interlace polynomial we know that $Q(G) = Q(G - v) + Q(G * v - v) + Q(G * uv - v)$. Now $I'(G) = (Q'(G))|_{y=0}$. So $I'(G) = \frac{1}{6}(Q'(G)) = \frac{1}{6}(Q'(G - v) + Q(G * v - v) + Q(G * uv - v))|y = 0$. Derivatives are distributive over addition, so $I'(G) = \frac{1}{6}(Q'(G - v))|_{y=0} + \frac{1}{6}(Q'(G * v - v))|_{y=0} + \frac{1}{6}(Q'(G * uv - v))|_{y=0} = I'(G - v) + I'(G * v - v) + I'(G * uv - v)$. ■

Lemma 4.1.3. *If G is a graph on at least three vertices then $I'(G)$ is a non-negative integer.*

Proof. First note that by lemma 4.1.1 if G is disconnected then $I'(G) = 0$ and is an integer. Now we prove the result by induction. For $|V(G)| = 3$, $G = P_3$ or $G = K_3$ as all other graphs are disconnected and thus already have integer values. Now $Q(K_3) = Q(P_3) = y^2 + 6y$ and $I'(K_3) = I'(P_3) = 1$. Now suppose the result holds for all connected graphs on k vertices for $k \geq 3$. Suppose G has $k + 1$ vertices. If G is disconnected we are done, so we may assume G is connected. Let v be a vertex in G. Since G is connected on at least 4 vertices v has a neighbor u. Consider $G - v$, $G * v - v$, and $G * uv - v$. All graphs are either disconnected and have an interlace invariant of zero, or are connected on k vertices and so by induction have an non-negative integer interlace invariant. By lemma 4.1.2, $I'(G) = I'(G - v) + I'(G * v - v) + I'(G * uv - v)$, which is a sum of non-negative integers, some of which may be zero, and is thus a non-negative integer. ■

This result simply provides us with the knowledge that in most cases we are constructing integer values and only relies on the fact that we can stop the recursion at K_3 or P_3 and know we have a sum of integers in these cases. Furthermore, we get these integer values in all but the cases on one and two vertices because of the choice to normalize by $\frac{1}{6}$, which is done in line with the Martin invariant definition. The Martin invariant was designed to produce a value of one for the graph in figure 4.8. As seen in lemma 4.1.4 and theorem 4.1.9 dividing out by $\frac{1}{3}$ gives nice formulas in the interlace case and we divide by the additional $\frac{1}{2}$ to align with the Martin invariant.

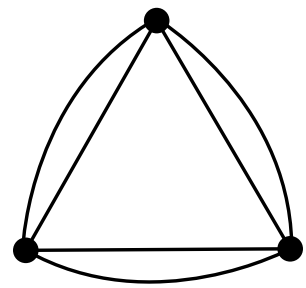


Figure 4.8: The 4-regular graph with Martin invariant one

Lemma 4.1.4. *For all $n \in \mathbb{Z}_{>0}$, $n \geq 2$, $I'(P_n) = 2^{n-3} = I'(K_n)$.*

Proof. We prove the claim by induction on n for P_n and K_n separately. First we induct on n for P_n. Observe that for $n = 2$, $I(P_2) = 3y$, and thus $\frac{1}{6}\frac{\partial}{\partial y}(3y)|_0 = \frac{1}{2} = 2^{2-3}$. For $n = 3$ we see that $I(P_3) = y^2 + 6y$ yielding $\frac{1}{6}\frac{\partial}{\partial y}(y^2 + 6y)|_0 = 1 = 2^{3-3}$ as desired.

Suppose that for some $n \geq 3$ that $\frac{1}{6}\frac{\partial}{\partial y}I(P_k)|_0 = 2^{k-3}$ for all $k \leq n$. Consider P_{n+1} with vertices $v_1, v_2, ..., v_n, v_{n+1}$ such that $v_i v_{i+1}$ is an edge for all $i \in [n]$. By lemma 4.1.2

$$I'(P_{n+1}) = I'(P_{n+1} \setminus v_{n+1}) + I'((P_{n+1} + v_{n+1}) * v_{n+1} \setminus v_{n+1}) + I'(P_{n+1} * v_{n+1}v_n \setminus v_{n+1}).$$

Now $P_{n+1} \setminus v_{n+1} = P_n$, $((P_{n+1} + v_{n+1}) * v_{n+1}) \setminus v_{n+1} = P_n$, and $P_{n+1} * v_{n+1}v_n \setminus v_{n+1} = P_{n-1} \cup v_n$ where v_n is an isolated vertex. Hence $I'(P_{n+1} \setminus v_{n+1}) = I'(P_n)$, $I'(((P_{n+1} + v_{n+1}) * v_{n+1}) \setminus v_{n+1)} = I'(P_n)$, and $I'(P_{n+1} * v_{n+1}v_n \setminus v_{n+1}) = 0$ by lemma 4.1.1. Therefore $I'(P_{n+1}) = 2I'(P_n)$. By the inductive hypothesis

$$I'(P_{n+1}) = 2(2^{n-3}) = 2^{(n+1)-3}.$$

For K_n, we have for $n = 2$, $K_2 = P_2$, and thus $I'(K_2) = 2^{2-3}$. For $n = 3$ we see that $I(K_3) = y^2 + 6y$ yielding $\frac{1}{6}\frac{\partial}{\partial y}(y^2 + 6y)|_0 = 1 = 2^{3-3}$ as desired.

Suppose that for some $n \geq 3$ that $\frac{1}{6}\frac{\partial}{\partial y}I(K_k)|_0 = 2^{k-3}$ for all $k \leq n$. Consider K_{n+1} with vertices $v_1, v_2, ..., v_n, v_{n+1}$. By lemma 4.1.2

$$I'(K_{n+1}) = I'(K_{n+1} \setminus v_{n+1}) + I'((K_{n+1} + v_{n+1}) * v_{n+1} \setminus v_{n+1}) + I'(K_{n+1} * v_{n+1}v_n \setminus v_{n+1}).$$

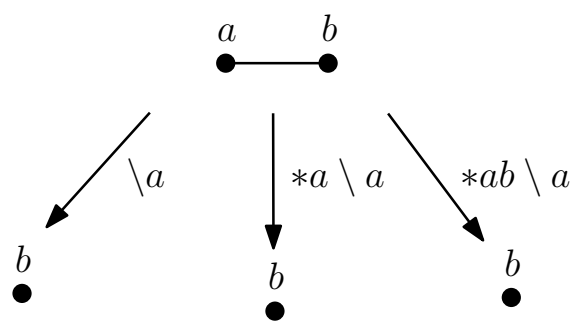
Now $K_{n+1} \setminus v_{n+1} = K_n$ and $K_{n+1} * v_{n+1}v_n \setminus v_{n+1} = K_n$. For $((K_{n+1}) * v_{n+1}) \setminus v_{n+1}$, we see that the local vertex complement removes all edges in the graph so the remaining graph is

n isolated vertices and thus the interlace polynomial is y^n. Hence $I'(K_{n+1} \setminus v_{n+1}) = I'(K_n)$ and $I'(K_{n+1} * v_{n+1}v_n \setminus v_{n+1}) = I'(K_n)$. Therefore $I(K_{n+1}) = 2I(K_n) + y^n$ and $I'(K_{n+1}) = 2I'(K_n) + 0$. By the inductive hypothesis

$$I'(K_{n+1}) = 2(2^{n-3}) = 2^{(n+1)-3}.$$

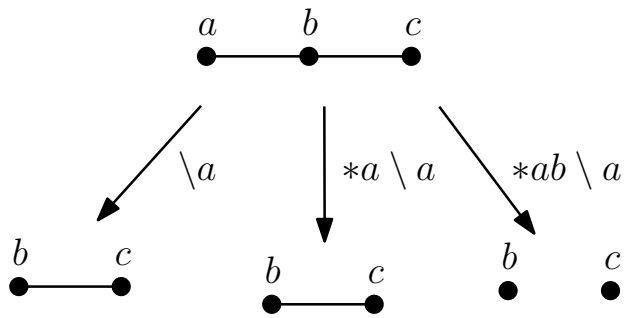
Thus $\frac{1}{6}\frac{\partial}{\partial y}I(P_n)|_0 = 2^{n-3}$ and $\frac{1}{6}\frac{\partial}{\partial y}I(K_n)|_0 = 2^{n-3}$. Hence the result holds. ■

This proof relies on noting that the base cases are the same as $P_2 = K_2$ and then applying the recursion to paths and vertices and observing that two of the terms are identical and the third is not part of the constant term when the derivative is taken. This process can be seen in figures 4.9, 4.10, 4.11, 4.12 and 4.13. As we can see in these figures, each iteration of the computation of the interlace polynomial produces two smaller paths and a graph with two components. Since any disconnected graph has an interlace invariant of zero by lemma 4.1.1 we only care about the two cases where we produce the path of one size smaller. The complete graph has the same scenario and the graph structure that enables this is either a vertex of degree one, or a pair of adjacent vertices who share a closed neighborhood that induces a clique. We will formalize this notion in theorem 4.1.5.



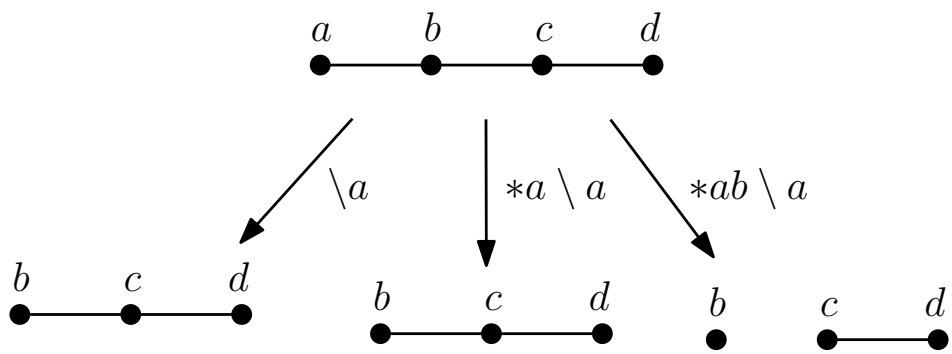
$$Q(P_1; y) = 3y$$

Figure 4.9: Computation of $Q(P_1; y)$



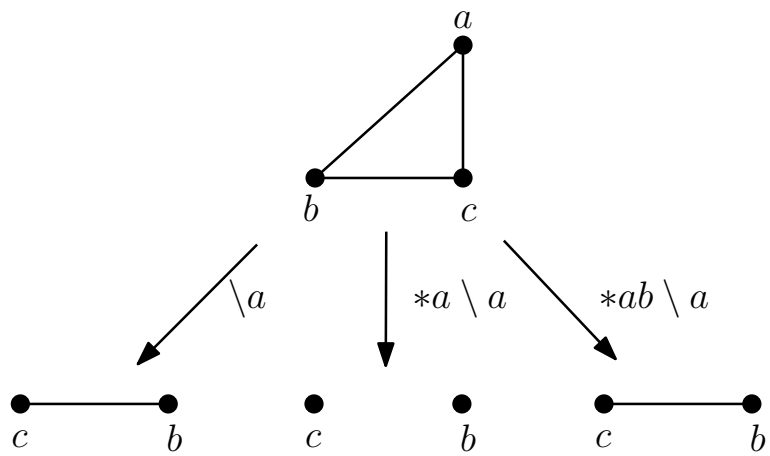
$$Q(P_2; y) = 2Q(P_1; y) + y^2$$

Figure 4.10: Computation of $Q(P_2; y)$



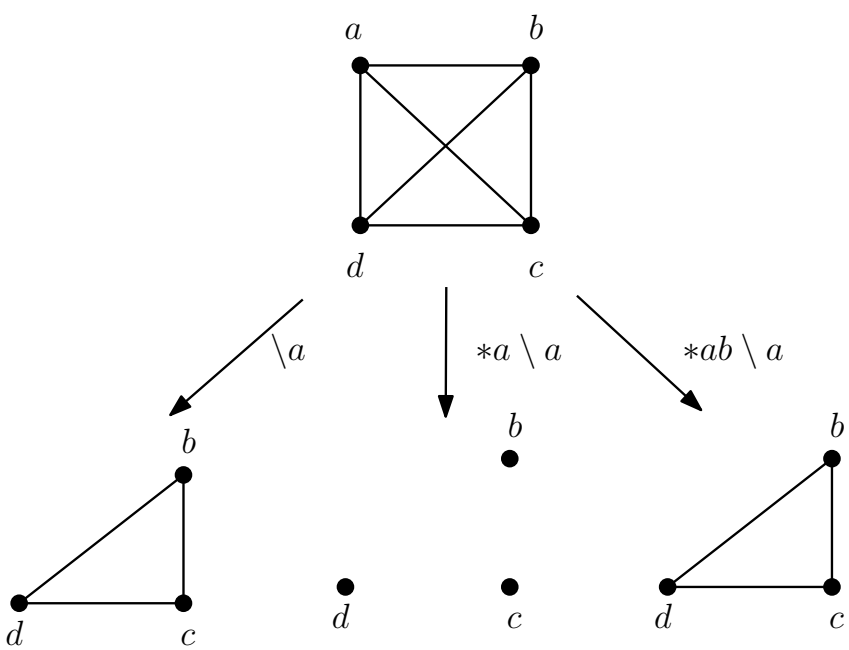
$$Q(P_3; y) = 2Q(P_2; y) + yQ(P_1; y)$$

Figure 4.11: Computation of $Q(P_3; y)$



$$Q(K_3; y) = 2Q(K_2; y) + y^2 = 2Q(P_1; y) + y^2 = 6y + y^2$$

Figure 4.12: Computation of $Q(K_3; y)$



$$Q(K_4; y) = 2Q(K_3; y) + y^3$$

Figure 4.13: Computation of $Q(K_4; y)$

Having two classes of graphs with this doubling pattern raises the question of what the interlace invariant might be counting. Investigating properties of complete graphs and paths on a fixed number of vertices does not yield any obvious answers. We know that in the Martin invariant case, it counts spanning tree partitions as shown in [96], though this fact is not obvious. However it is not clear what, if anything, the interlace invariant might count on graphs. Currently, no graph property examined appears to be the correct thing that is being counted, but the investigation led to the result in theorem 4.1.5 which expands the class of graphs for which the interlace invariant is twice that of the graph with a vertex removed. We will say a pair of vertices u, v in a graph C is *locally complete* if vu is an edge, $\overline{N(u)} = \overline{N(v)}$, so w is a neighbor of u if and only if it is v or a neighbor of v, and the subgraph induced by $\overline{N(v)}$ is a complete graph.

Theorem 4.1.5. *For a graph C if C has a leaf (a degree one vertex) v or a pair of locally complete vertices u and v then $I'(C) = 2I'(C - v)$.*

Proof. We prove this as two separate cases. First suppose C has a leaf v. Then v has exactly one neighbor u, so $C * v \setminus v = C \setminus v$ as the vertex local complement removes the edge uv, which irrelevant as v is deleted. For the edge local complement $N(v) = \{u\}$. There are no common neighbors, so $N(u) - v$, $\varnothing$ and $\{u, v\}$ are the three sets for our edge local complement. We add edges from v to all of $N(u) - v$ and delete edges from u to every vertex in $N(u) - v$. Then we delete v which leaves u an isolated vertex. So $Q(C; y) = 2Q(C - v; y) + y(Q(C * uv - v - u; y))$. Therefore $I'(C) = 2I'(C - v)$ as desired.

Now suppose that C has vertices u and v such that $\overline{N(u)} = \overline{N(v)}$ and $C[\overline{N(v)}]$ is a complete graph. This is to say that all neighbors of v other than u are also neighbors of u and all neighbors of v are neighbors of each other. This means that when we take the edge local complement of uv every vertex is in $\overline{N(v)} \cap \overline{N(u)}$ and so both other sets of vertices are empty. Hence the edge local complement does nothing and so we have $C - v = C * uv - v$. For the vertex local complement, $N(u) = N(v) + \{v\}$ and so all edges incident to u are deleted when taking the vertex local complement of v. This makes u an isolated vertex. Thus $Q(C; y) = 2Q(C - v; y) + yQ(C * v - v - u; y)$. Therefore $I'(C) = 2I'(C - v)$ as desired. ■

A nice observation from this proof is that of a leaf v, $G * v - v$ and $G - v$ are isomorphic and for a pair of locally complete vertices u and v, $G - v$ and $G * uv - v$ are isomorphic. This is because the vertex local complement only affects the edge uv for a leaf and the edge local complement has two empty sets of vertices and thus makes no changes to the edge of G. Examples of this can be seen in figure 4.14 and figure 4.15

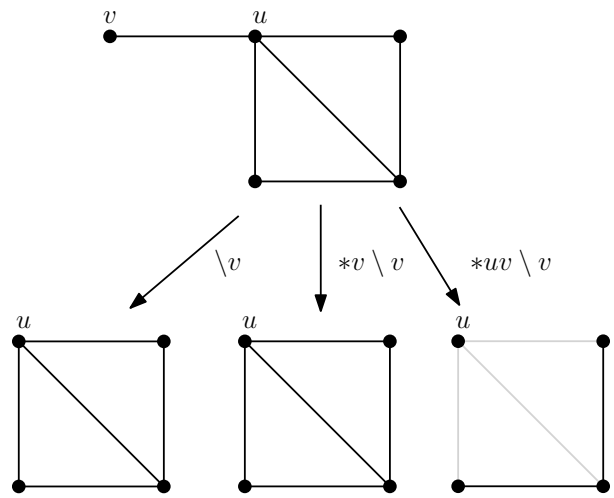


Figure 4.14: Local complementation with a leaf

Furthermore, in the case of the complete graph, K_n, the recursion is very nice as it is always twice the previous case plus y^n. This yields an explicit formula.

Theorem 4.1.6. *For K_n, $n \geq 3$, $Q(K_n; y) = 3{\cdot}2^{n-2}y + \sum_{i=1}^{n-2} 2^{i-1}{\cdot}y^{n-i} = 3{\cdot}2^{n-2} + \frac{y^n}{y-2}\left(1 - \left(\frac{2}{y}\right)^{n-2}\right)$.*

Proof. We begin by observing from the proof of lemma 4.1.4 that $Q(K_{n+1}; y) = 2Q(K_n; y) + y^{n+1}$. Furthermore the coefficient of y is $6 \cdot 2^{(n+1)-3} = 3 \cdot 2^{(n+1)-2}$. We will now proceed by induction. For $n = 3$, we have $Q(K_3; y) = 3 \cdot 2^{3-2}y + \sum_{i=1}^{3-2} 2^{i-1} \cdot y^{3-i} = 6y + y^2$ as desired. Now suppose the result holds for K_k for some $k \geq 3$. Consider K_{k+1}. By the recursion and

inductive hypothesis

$$\begin{aligned}
Q(K_{k+1};y) &= 2Q(K_k;y) + y^{k+1} = 2^0 y^{k+1} + 2(3\cdot 2^{k-2}y + \sum_{i=1}^{k-2} 2^{i-1}\cdot y^{k-i}) \\
&= 3\cdot 2^{(k+1)-2}y + 2^{1-1}y^{k+1} + 2\sum_{i=1}^{k-2} 2^{i-1}y^{k-i} \\
&= 3\cdot 2^{(k+1)-2} + 2^{1-1}y^{k+1} + \sum_{i=1}^{k-2} 2^{i}y^{k-i} \\
&= 3\cdot 2^{(k+1)-2} + 2^{1-1}y^{k+1} + \sum_{i=2}^{k-3} 2^{i-1}y^{k-(i-1)} \\
&= 3\cdot 2^{(k+1)-2} + \sum_{i=1}^{(k+1)-2} 2^{i-1}y^{(k+1)-i}
\end{aligned}$$

as desired. Observe that

$$\begin{aligned}
3\cdot 2^{n-2}y + \sum_{i=1}^{n-2} 2^{i-1}\cdot y^{n-i} &= 3\cdot 2^{n-2} + y^{n-1}\sum_{i=1}^{n-2}(\frac{2}{y})^{i-1} \\
&= 3\cdot 2^{n-2} + y^{n-1}\left(\frac{1-(\frac{2}{y})^{n-2}}{1-\frac{2}{y}}\right) \\
&= 3\cdot 2^{n-2} + \frac{y^n}{y-2}\left(1-\left(\frac{2}{y}\right)^{n-2}\right)
\end{aligned}$$

Hence the result holds. ■

Hopefully further work will lead to closed results for other classes of graphs like paths, cycles, and complete bipartite graphs.

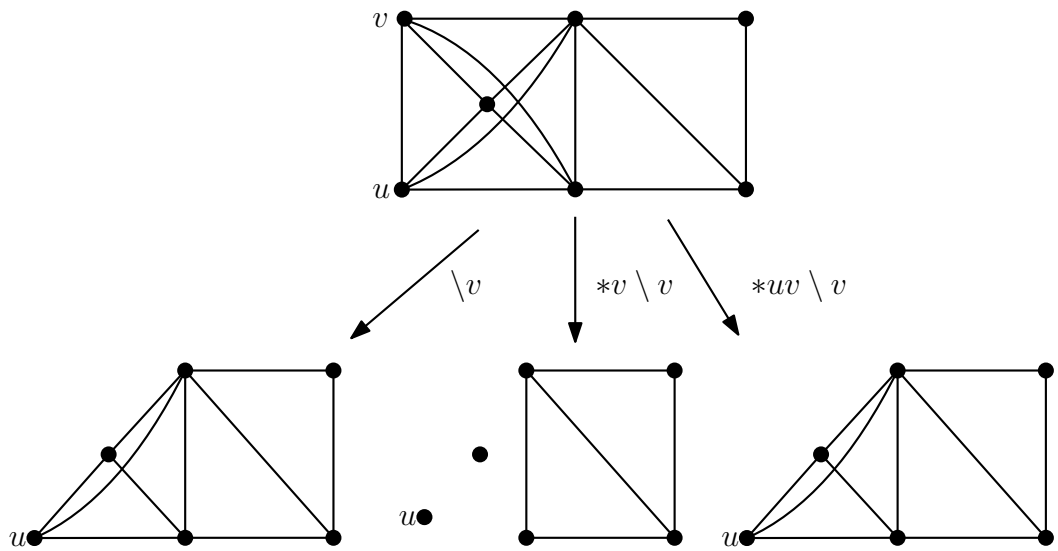


Figure 4.15: Local complementation for a locally complete vertex pair

Another nice class of graphs is trees. So a natural question is what can be proven regarding trees. We know we can compute the interlace invariant of a tree using theorem 4.1.5 since a tree always has a leaf. We can also obtain a slightly more direct formula for the interlace polynomial of trees by arbitrarily rooting them. Rooting a tree in this context simply means making a choice of vertex for the root, which establishes parent, children, and sibling relationships for the vertices. Given a tree T with root r, we may compute the distance of any vertex v to the root. We say v is a *child* of u and u is the *parent* of v if uv is an edge and $dist(u,r) < dist(v,r)$. Two vertices u and v are *siblings* if w is the parent of both u and v. Note that a parent may have many children but each vertex other than the root has exactly one parent as there is a unique path from any vertex to the root.

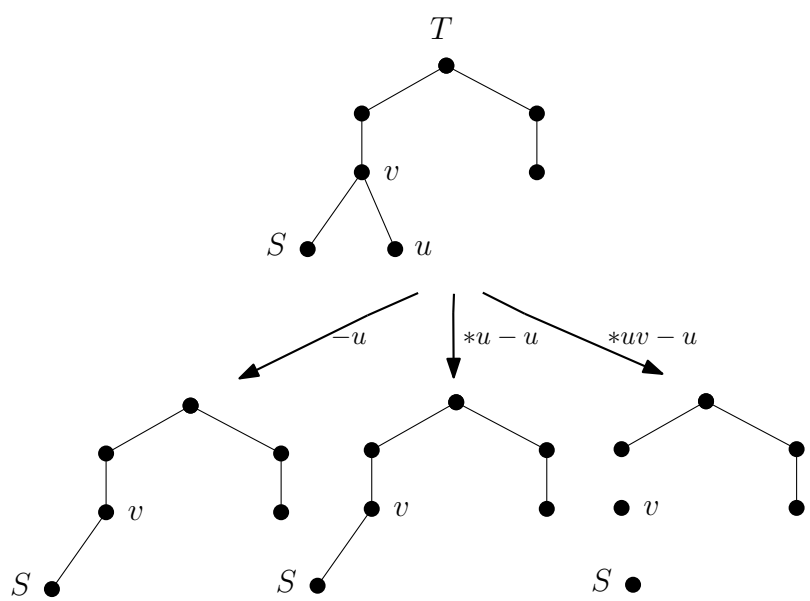
Corollary 4.1.7. *If T is a tree on n vertices then $I'(T) = 2^{n-3}$.*

Proof. By theorem 4.1.5 we may iteratively remove vertices of T when they are leaves since a tree always has a leaf and the removal of a leaf will not disconnect the tree. So $I'(T) = 2I'(T - l_1) = 2(2I'(T - l_1 - l_2)) = ... = 2^{n-3}I'(P_3)$ where l_i is the leaf we remove at each stage and we remove leaves until P_3 remain. Now $I'(P_3) = 1$ by direct computation. So $I'(T) = 2^{n-2}(1) = 2^{n-3}$. ■

Corollary 4.1.8. *If T is a rooted tree and $v \in V(T)$ is a leaf such that all siblings of v, the set of siblings of v is denoted by $S \subset V(T)$, are also leaves, and u is the parent of v then $Q(T;y) = 2Q(T - v;y) + y^{|S|+1}Q(T - S - \{u,v\};y)$.*

Proof. By the recursive formula for the interlace polynomial $Q(T;y) = Q(T-u;y)+Q(T*v-v;y)+Q(T*uv-v;y)$. Now v has u as its only neighbor so $T*v-v = T-v$ and thus $Q(T*v-v;y) = Q(T-v;y)$. Now all of the children of u are leaves, so $Q*uv$ adds edges from v to all its siblings and the parent of u, but these edges are removed when we deleted v and all edges from u to its parent and children are removed. So $T*uv-v$ has isolated vertices $\{u\} \cup S$. Therefore $Q(T*uv-v;y) = y^{|S|+1}Q(T-S-\{u,v\};y)$. Therefore $Q(T;y) = 2Q(T-v;y)+y^{|S|+1}Q;(T-S-\{u,v\};y)$. ■

Note this proof only relies on a root so that we may choose a vertex to remove in a way that ensure we get two isomorphic graphs and a graph with isolated vertices in each stage of the recursion. Choosing a root in any tree, arbitrarily, will allow the use of this result and provides a nice way to describe which vertex and edge to choose for the recursive computation. Furthermore, there is always a leaf in a rooted tree such that all its siblings, if any exist, are also leaves and so we may always apply this theorem to any tree by arbitrarily rooting it first.



$$Q(T;y) = 2Q(T-u;y) + y^2Q(T-u-S-v;y)$$

Figure 4.16: One step of the interlace polynomial of a rooted tree

For the next proof we define what in means for a vertex or set of vertices to be complete to another set of vertices. A set of vertices A in G is *complete* to some subgraph H of G if for all $v \in A$ and $u \in H$, uv is an edge.

Theorem 4.1.9. *For the complete bipartite graph $K_{m,n}$, $I'(K_{m,n}) = 2^{m+n-3}$ for $m, n \geq 1$.*

Proof. Proceeding by induction observe that $Q(K_{1,1}; y) = 3y$ and $\frac{1}{6} \cdot 3 = \frac{1}{2} = 2^{-1} = 2^{1+1-3}$ as desired.

Suppose for some $m, n \geq 1$, that $I'(K_{j,i}) = 2^{j+i-3}$ for all $j \leq m, i \leq n$. Consider $K_{m+1,n}$ and $K_{m,n+1}$. We prove the result for $K_{m+1,n}$ as the proof for $K_{m,n+1}$ is identical by swapping m and n. Let A and B be the sides of the bipartition containing $m+1$ and n vertices respectively.

Pick $v \in A$ and $u \in B$. Notice that $uv \in E(G)$. By the recursive formula for the interlace invariant

$$I'(K_{m+1,n;y}; y) = I'(K_{m+1,n} - v; y) + I'(K_{m+1,n} * v - v; y) + I'(K_{m+1,n} * uv - v; y)$$

First notice that $K_{m+1,n} - v = K_{m,n}$. Now consider $K_{m+1,n} * uv - v$. The vertex sets to be locally complemented are $\{u, v\}$, $A - v$, and $B - u$. The edge local complement removes all edges from u to $A - v$ and v to $B - v$ as seen in figure 4.17. So after v is deleted all vertices in $A - v$ are left as isolated vertices and since $|A| \geq 2$ we have at least two components remaining in the graph and thus by lemma 4.1.1 there is no y term in the interlace polynomial. So this contributes zero to the interlace invariant.

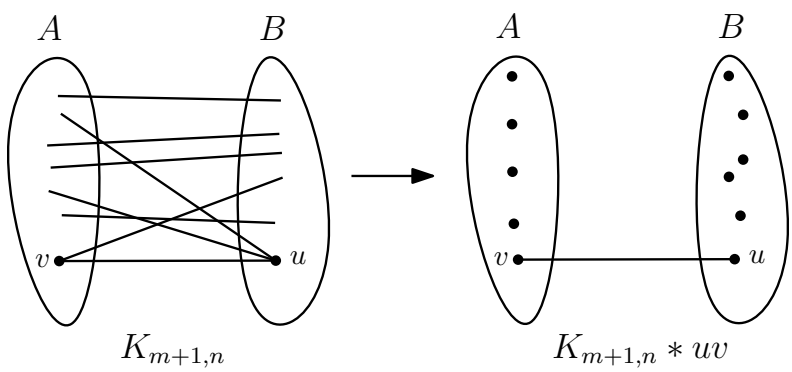


Figure 4.17: Local edge complementation in $K_{m+1,n}$

Now consider $K_{m+1,n} * v - v$. This adds edges between all pairs of vertices in B. So we get $|A| - 1$ vertices complete to a complete graph on B vertices. If B has only one vertex then we get $K_{m,n} = K_{m,1}$. Suppose then that $|B| \geq 2$. We then choose $v_1 \in A$ which exists since $|A| \geq 2$ and note that $v_1 u$ is an edge. Let $K_{m+1,n} * v - v = G_1$, then by lemma 4.1.2

$$I'(K_{m+1,n} * v - v; y) = I'(G_1 - v_1; y) + I'(G_1 * v_1 - v_1; y) + I'(G_1 * v_1 u - v_1; y).$$

Unless v_1 is the only vertex remaining in A, $G_1 - v_1$ creates a graph with a clique on $|B|$ vertices and $|A| - 2$ vertices complete to this clique. Call this graph G_2 and see figure 4.18. If $A = \{v, v_1\}$ then G_1 is K_{n+1} and G_2 is K_n. $G_1 * v_1 - v_1$ complements all edges between neighbors of v_1. Now $N(v_1) = B$, which is a complete subgraph so $G_1 * v_1 - v_1 = K_{m-1,n}$, which is just n isolated vertices if $|A| = 2$. Finally, $G_1 * uv_1 - v_1$ has only two vertex sets to be complemented, $A - \{v, v_1\}$ and $B \cup \{v_1\}$. So $G_1 * uv_1 - v_1$ has isolated vertices $A - \{v, v_1\}$ and does not contribute to the interlace invariant unless $|A| = 2$ when $G_1 * uv_1 - v_1 = K_n$ as this operation simply deletes v_1. This only leaves G_2 as potentially a graph that we don't know how it will contribute to the interlace invariant. However this process can be repeated for every vertex in A until we are left with only one vertex that hasn't been deleted in A and then we will get the two copies of K_n and the n isolated vertices we would have gotten if $|A| = 2$. So this process occurs $m - 1$ times where G_i is produced by vertex deletion at all successive stages as vertex deletion doesn't changes edges within B, but local complementation removes them. So at each stage we produce a graph that is a smaller complete bipartite graph, a graph with a K_n subgraph and a set of vertices complete to that subgraph, and a graph with isolated vertices unless we are at the last stage where G_{m-1} is the K_{n+1}. Thus for $G_0, G_1, ..., G_{m-1}$ where $G_0 = K_{m+1,n}$ we have $I'(G_i) = I'(G_{i+1}) + I'(K_{m-i,n}) + 0$ where the zero comes from the term with isolated vertices. For G_m we get $I'(G_m) = I'(K_{n+1})$ by lemma 4.1.4 and we know this is $2^{n+1-3} = 2^{n-2}$.

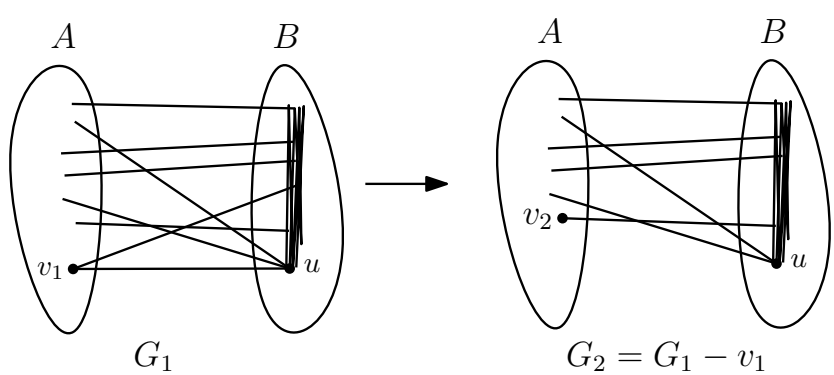


Figure 4.18: Local vertex complementation in $K_{m,n}$

So we can construct this sequence of G_1 and write

$$I'(K_{m+1,n}; y) = I'(K_{n+1}) + \sum_{i=1}^{m-1} I'(K_{i,n})$$

By the inductive hypothesis, result on K_{n+1}, and finite geometric sum, we have

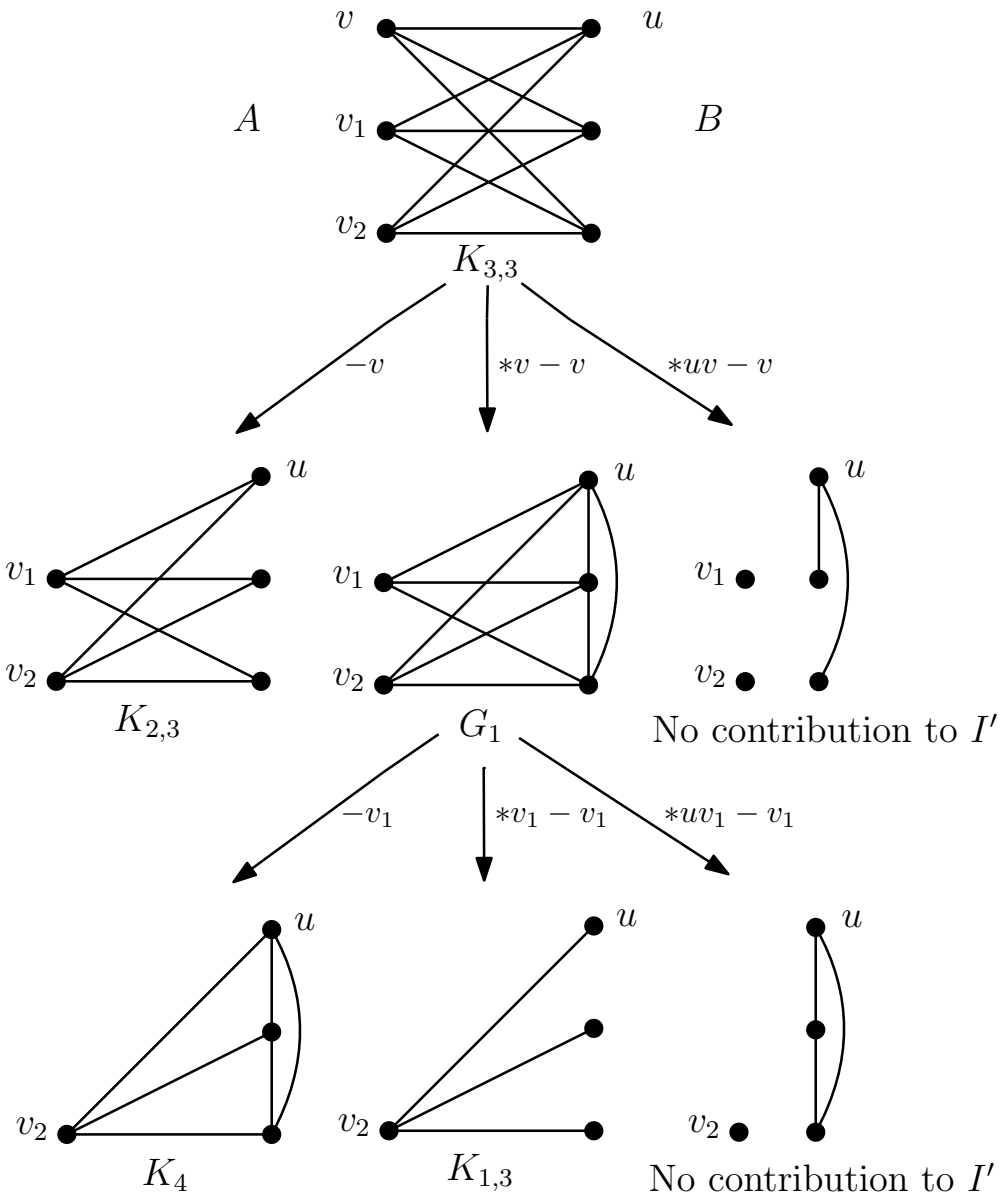
$$\begin{aligned}
I'(K_{m+1,n};y) &= 2^{n-2} + \sum_{i=1}^{m-1} 2^{i+n-3} \\
&= 2^{n-2} + 2^{n-3} \sum_{i=1}^{m-1} 2^{i} \\
&= 2^{n-2} + 2^{n-3} \cdot 2\frac{2^{m+1}-1}{2-1} \\
&= 2^{n-2} + 2^{n-3}(2^{m+1}-2) \\
&= 2^{n-2} + 2^{n-3+m+1} - 2^{n-3+1} = 2^{m+1+n-3}.
\end{aligned}$$

Notice that n is constant throughout this argument, so we are inducting on m in essence and fixing m we can analogously induct on n, which gives us the result for all m and n. Therefore $I'(K_{m,n}) = 2^{m+n-3}$ for all m, n as desired. ■

The local complementation operations play very nicely in a complete bipartite graph due to consistent sets of neighbors and non-neighbors of each vertex of the graph. As seen in figure 4.19 when we take the vertex local complement of a vertex in a complete bipartite graph we add edges between all pairs of vertices in one side of the bipartition. After that, vertex deletion cannot change this structure. Furthermore, vertex local complementation will basically toggle edges between all pairs of vertices in the other side of the bipartition. This returns the graph to a complete bipartite graph at each stage after the first.

Edge local complementation is also nice in a complete bipartite graph in that it will always remove all edges between the two sides of the bipartition and add edges between the ends of the chosen edge and their side of the bipartition. If one side of the bipartition has been completed, then the operation simply isolates all vertices on the other side of the bipartition.

When we performed the recurrence with all but one vertex in one side of our bipartition we were left with a complete graph as one of the two graphs that contributed to the interlace invariant. Since we already have a formula for this contribution we may simply apply it.



$$I'(K_{3,3}) = I'(K_4) + \sum_{i=1}^{2} K_{i,3} = 2^{4-3} + 2^{1+3-3} + 2^{2+3-3}$$
$$= 2 + 2 + 4 = 8 = 2^3 = 2^{3+3-3}$$

Figure 4.19: Computation of $I'(K_{3,3})$

We also have a proof for the formula for the interlace invariant of a cycle. Running code to compute the interlace invariant on cycles we obtained the sequence $1, 2, 6, 14, 34, 78, 178, 398, 882, 1934, 4210, 9102, 19570, 41870$ which appears in exactly one sequence in the Online Encyclopedia of Integer Sequences, A059570 [106]. We conjectured that this sequence, which has a closed form, is the sequence for the interlace invariant of cycles and we were indeed able to prove this result.

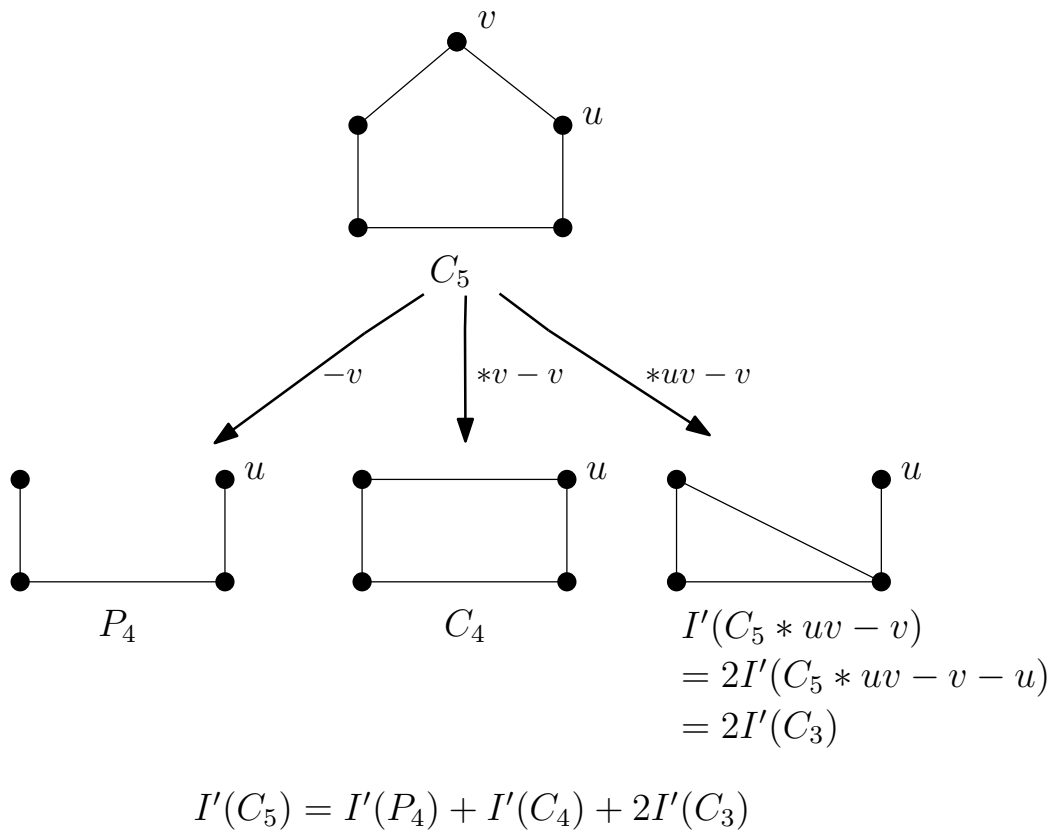
Theorem 4.1.10. *For C_n, $n \geq 3$ $I'(C_n) = \frac{(3(n-2)+4)2^{n-2}}{18} - \frac{(-1)^{n-2}}{9}$.*

Proof. By computation C_3 has interlace polynomial $y^2 + 6y$ and thus $I'(C_3) = 1$. Now $n - 2 = 1$ so $\frac{(3(1)+4)2^1}{18} - \frac{2(-1)^1}{9} = \frac{2(7)}{18} + \frac{2}{9} = \frac{7}{9} + \frac{2}{9} = 1$ as desired. For C_4 we compute the interlace polynomial to be $5y^2 + 12y$ so the interlace invariant is 2. Now $n - 2 = 2$ so $\frac{(3(2)+4)2^2}{18} - \frac{2(-1)^2}{9} = \frac{4(10)}{18} - \frac{2}{9} = \frac{20}{9} - \frac{2}{9} = 2$ as desired.

Suppose the result holds for all cycles on at most $n \geq 4$ vertices and consider C_{n+1}. Take uv an edge in C_{n+1}. Then $I'(C_{n+1}) = I'(C_{n+1} - u) + I'(C_{n+1} * u - u) + I'(C_{n+1} * uv - u)$. Now $C_{n+1} - u = P_n$, $C_{n+1} * u - u = C_n$ and $C_{n+1} * uv - u$ is C_{n-1} with v as a pendent vertex. But then v is a leaf, so by theorem 4.1.5 $I'(C_{n+1} * uv - u) = 2I'(C_{n+1} * uv - u - v) = 2I'(C_{n-1})$. So $I'(C_{n+1}) = I'(P_n) + I'(C_n) + 2I'(C_{n-1})$. So by the inductive hypothesis, $n \geq 3$, and lemma 4.1.4, we have

$$\begin{aligned}
I'(C_{n+1}) &= 2^{n-3} + \frac{(3(n-2)+4)2^{n-2}}{18} - \frac{2(-1)^{n-2}}{9} + 2\left(\frac{(3(n-3)+4)2^{n-3}}{18} - \frac{2(-1)^{n-3}}{9}\right) \\
&= \frac{9 \cdot 2^{n-3}}{9} + \frac{(3n+2)2^{n-3}}{9} + \frac{2(-1)^{n-3}}{9} + \frac{(3n-5)2^{n-3}}{9} - \frac{4(-1)^{n-3}}{9} \\
&= \frac{(9+3n+2+3n-5)2^{n-3}}{9} - \frac{(-2)(-1)^{n-3}}{9} \\
&= \frac{(6n+6)2^{n-3}}{9} - \frac{(-2)(-1)^{n-3}}{9} \\
&= \frac{(3n+3)2^{n-2}}{9} - \frac{2(-1)^{n-2}}{9}
\end{aligned}$$

, as desired. ■



$$I'(C_5) = I'(P_4) + I'(C_4) + 2I'(C_3)$$

Figure 4.20: Computation of $I'(C_5)$

4.2 Symmetries and the Interlace Invariant

In this section we will discuss extending the symmetries of the Martin invariant to the interlace invariant. Recall that the Martin invariant also has the Feynman period symmetries, some of which have a planarity component. It is not clear how planarity translates through the Martin polynomial and interlace polynomial relationship. It is also not clear, in general, how the interlace polynomial treats cut-cycle duality. In this section we will focus on the three vertex cut that is replaced with a triangle and the four edge cut in the Martin invariant 4-regular case and how those translate to the interlace invariant as these are both product type symmetries.

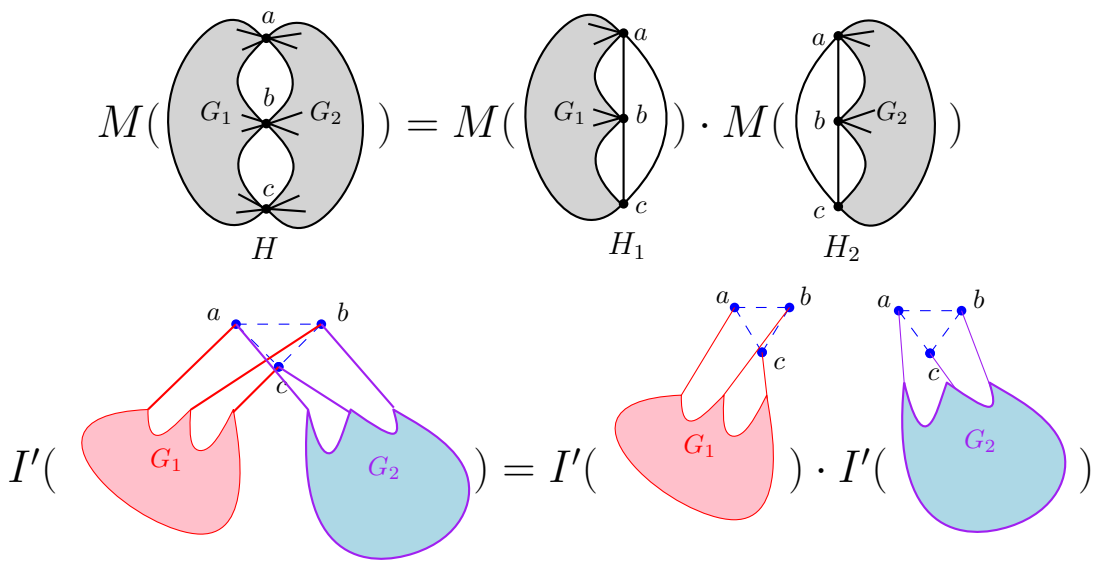


Figure 4.21: An interlace invariant symmetry based on the Martin invariant product

When we convert from a 4-regular graph to an interlace graph we have a choice of Eulerian circuit, that while it won't affect the interlace polynomial, does affect the structure of the interlace graph. So when investigating these symmetries we looked for choices of circuits that would give us structures that were easier to work with. This does mean that not every graph that satisfies these symmetries may have been described. However, we can extend our classes of graph slightly by using the fact that the interlace polynomial is invariant under vertex and edge local complementation. So when investigating the Martin invariant symmetries seen in figures 4.21 and 4.22 we chose Eulerian circuits in a very specific manner. For the product symmetry, in figure 4.21 we chose the Eulerian circuit on G_1 and G_2 first and then glued the circuits together at the cut vertices. We chose the gluing to prevent any chords crossing in the chord diagram between vertices in opposite sides of the cut. We could do this because cleaving the cut vertices results in two even regular graphs that will both have an Eulerian circuit. This gave us interlace graphs that could be split into two graphs intersecting only on three vertices. Once we found this pattern it was only necessary to prove the product property for the interlace invariant holds for such graphs, as we are not claiming that only graphs of this structure can satisfy the symmetry shown in figure 4.21.

Theorem 4.2.1. *Suppose G is a connected graph such that $G = G_1 \cup G_2$ where $|V(G_1) \cap V(G_2)| = 3$ and $G_1[G_1 \cap G_2] = G_2[G_1 \cap G_2]$ and G_1 and G_2 are also connected and for every vertex in $G_1 \setminus G_2$ or $G_2 \setminus G_1$ it is adjacent to at most one vertex in $G_1 \cap G_2$. Further, for*

any pair of adjacent vertices in $G_i - (G_1 \cap G_2)$ if they both have a neighbor in $G_1 \cap G_2$ then it is the same neighbor. Then $I'(G) = I'(G_1)I'(G_2)$.

Proof. Suppose that $G = G_1 \cup G_2 = G_1 = G_2$ where $G_1 = G_2 = P_3$ or $G_1 = G_2 = K_3$. Now $I'(K_3) = I'(P_3) = 2^0 = 1$ by lemma 4.1.4. So $I'(G) = 1 = 1 \cdot 1 = I'(G_1)I'(G_2)$ as desired. If $G_1 = G_2$ is three isolated vertices or P_2 with an isolated vertex then the interlace invariant is zero and the result still holds.

Suppose that for $k \geq 3$, and $|V(G)| = k$ if $G = G_1 \cup G_2$ satisfies the theorem conditions then $I'(G) = I'(G_1)I'(G_2)$. Let G be a connected graph on $k+1$ vertices such that $G = G_1 \cup G_2$ where $|V(G_1) \cap V(G_2)| = 3$ and $G_1[G_1 \cap G_2] = G_2[G_1 \cap G_2]$. Since $|V(G)| > 3$ there exists $v \in V(G)$ such that $v \notin G_1 \cap G_2$. Without loss of generality suppose that $v \in G_1$. Observe that $I'(G) = I'(G - v) + I'(G * v - v) + I'(G * vu - v)$ for some neighbor u of v which exists since G is connected. Choose u such that $u \notin G_1 \cap G_2$ if possible. Observe that since $v \notin G_1 \cap G_2$, $G - v = (G_1 - v) \cup G_2$ and $G_1 - v$ and G_2 still have the appropriate intersection and all vertices have the desired neighborhoods. So by the inductive hypothesis $I'(G - v) = I'(G_1 - v)I'(G_2)$ or $G - v$ is disconnected and thus $G_1 - v$ is disconnected and $I'(G - v) = 0 = I'(G_1 - v)I'(G_2)$ since disconnected graphs have interlace invariant zero.

Now consider $G * v - v$. There are two cases to consider. First suppose that $N(v) \cap (G_1 \cap G_2) = \varnothing$. Then $G * v - v = (G_1 * v - v) \cup G_2$ and the relationship between G_1 and G_2 and the restrictions on the neighborhoods of vertices in relation to the intersection is unchanged. Thus $I'(G * v - v) = I'(G_1 * v - v)I'(G_2)$ by the inductive hypothesis with equality still holding if the deletion of v disconnects the graph as both sides are then zero. Now if it is not the case that $N(v) \cap (G_1 \cap G_2) = \varnothing$ then $G * v - v$ can in theory affect $G_1 \cap G_2$. However, it cannot add or remove edges between vertices in G_1 and G_2 outside of the intersection and since $N(v) \cap (G_1 \cap G_2) \neq \varnothing$, $|N(v) \cap (G_1 \cap G_2)| = 1$. Since v has exactly one neighbor in the intersection and we are complementing the neighborhood of v this neighbor cannot affect edges in $G_1 \cap G_2$. Since v is adjacent to exactly one vertex in $G_1 \cap G_2$ then all of $N(v)$ must be adjacent to this same vertex. So the local complementation removes all edges from $N(v)$ to the intersection and thus the theorem conditions hold for $G * v - v$. Then by the inductive hypothesis, $I'(G_1 * v - v) = I'(G_1 * v - v)I'(G_2)$.

Finally, consider $G * uv - v$. If u is not in $G_1 \cap G_2$ then $u \in G_1$ and $G * uv - v = (G_1 * uv - v) \cup G_2$ as $G_1 \cap G_2$ has at most one vertex in $N(u) \cap N(v)$. Since uv is an edge, if they are adjacent to a vertex in $G_1 \cap G_2$ it is the same vertex w and then all of their neighbors must also be adjacent to this vertex. So the local complementation will remove edges from all neighbors of just u or v to w and also removes all edges from $\overline{N(v)} \cap \overline{N(u)}$ to the neighbors of just u or v. So the only vertices that will still be adjacent to w are in $\overline{N(v)} \cap \overline{N(u)}$ and so the theorem conditions hold for $G * uv - v$. By the

inductive hypothesis, as the relationship between G_1 and G_2 remains unchanged, we have $I'(G*uv-v) = I'(G_1*uv-v)I'(G_2)$. Now suppose that $u \in G_1 \cap G_2$. Since we have no other choice for u or v it must be the case that every vertex not in the intersection of G_1 and G_2 is incident to exactly one vertex, which is in $G_1 \cap G_2$. From this we may further assume that all vertices not in the intersection of G_1 and G_2 are actually in G_1 as every vertex not in the intersection is a leaf. So $G_1 * uv$ removes all edges incident to u as v is a leaf and thus has no other neighbors. this makes u an isolated vertex. Now $G_1 = G$ so $G * uv$ also makes u an isolated vertex. So $G * uv - v$ and $G_1 * uv - v$ have at least two components. Thus both have interlace invariant zero. So $I'(G * uv - v) = 0 = I'(G_1 * uv - v)I'(G_2)$. So in all cases we may write $I'(G * uv - v) = I'(G_1 * uv - v)I'(G_2)$ where this is zero unless there is a way to choose u and v such that they don't modify $G_1 \cap G_2$.

Therefore,

$$\begin{aligned}
I'(G) &= I'(G - v) + I'(G * v - v) + I'(G * uv - v) \\
&= I'(G_1 - v)I'(G_2) + I'(G_1 * v - v)I'(G_2) + I'(G_1 * uv - v)I'(G_2) \\
&= I'(G_2)(I'(G_1 - v) + I'(G_1 * v - v) + I'(G * uv - v)) \\
&= I'(G_2)I'(G_1)
\end{aligned}$$

as desired. ■

While this does provide us a nice class of graphs for which the symmetry holds, there are a number of open questions about further extending this result. For example, what further graphs might we be able to characterize by examining what graphs are edge or vertex local complementations of these graphs. Graph that are edge or vertex local complementations of these graphs have the same interlace invariant, and we should be able to use this in relation to the theorem. Furthermore, this proof requires that any vertex not in the intersection of G_1 and G_2 is adjacent to at most one vertex in the intersection and that two adjacent vertices not in the intersection cannot be adjacent to different vertices in the intersection. These conditions arose from trying to prove the more general pattern obtained from constructing example with the Martin invariant, but it seems plausible that there might be a way to relax this condition. In concrete examples derived from interlace graphs from graph with the 3-cut this property does not appear necessary, but so far proving the result with relaxed conditions has not been possible.

Another symmetry we have extended is the symmetry of the 4-cut, illustrated in figure 4.22. For constructing this class of graphs we chose circuits in our 4-regular graphs that were on the two separate sides of the cut with the added vertex and then glued them at the cut according to the added vertices. This resulted in graphs that appeared to be constructed via a specific 1-join. In the literature there are varying definitions of 1-joins

and the definition we use here aligns with [116, 124] and we will construct our 1-joins in the same manner as [76]. We define the 1-*join* between graphs G_1 and G_2, with chosen vertex sets A and B respectively, to result in a graph G', where G' has vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2) \cup \{(a,b) | a \in A, b \in B\}$. So, as seen in figure 4.22 when we choose Eulerian circuits on G_1 and G_2 we can construct interlace graphs that can combine using the 1-join of the neighborhoods defined by the vertices that are interlaced with the cut. This pattern was somewhat hard to identify because the Martin symmetry doesn't simply split up the vertex set of the original graph. It is the twice the product of two graphs that have a vertex not in the original graph. This property does carry over to the interlace invariant in a class of graphs we determined by constructing examples where we choose Eulerian circuits on G_1 and G_2 and glued them to obtain a circuit on G. This gave us interlace graphs where they were 1-joins with the neighborhood of v_1 and v_2 determining the 1-join. This made sense as the vertices interlaced with v_1 and v_2 were determined by the gluing of the Eulerian circuits we picked in G_1 and G_2. A final definition we need is that of an apex vertex. For a graph G an *apex vertex* v is a vertex such that for all $u \in V(G)$, $u \neq v$ we have $uv \in E(G)$.

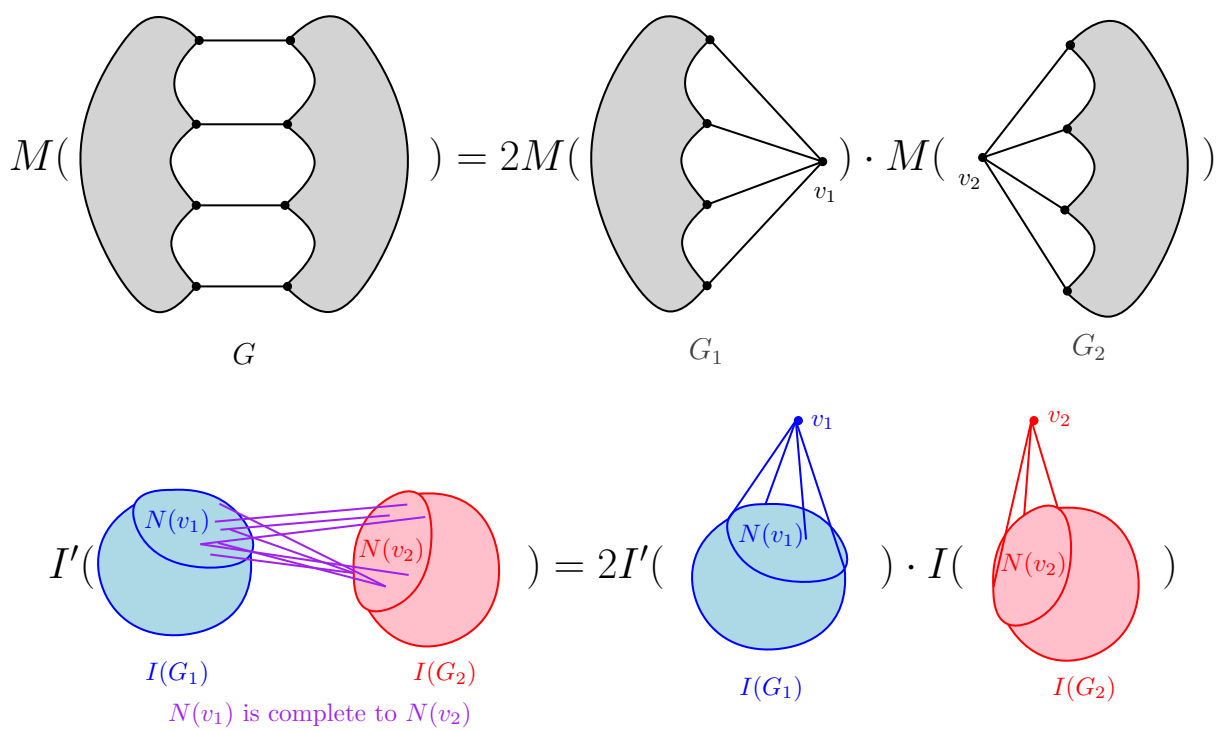


Figure 4.22: An interlace invariant symmetry based on the Martin invariant 4-cut

Theorem 4.2.2. *Let G_1 and G_2 be graphs with at least two vertices and distinguished vertices v_1 and v_2. Further suppose that if both v_1 and v_2 have non-neighbors in G_1 or G_2 respectively, then every non-neighbor of v_1 or v_2 is a leaf or has degree two and at least one of its neighbors is also a non-neighbor. Furthermore, if v_1 or v_2 is the only apex vertex in G_1 or G_2 then all of its neighbors have degree at most two. Then if G is obtained from the 1-join of $G_1 - v_1$ and $G_2 - v_2$ with vertex sets $A = N(v_1)$ and $B = N(v_2)$, $I'(G) = 2I'(G_1)I'(G_2)$.*

Note that the condition on the non-neighbors of v_1 and v_2 is quite technical but is necessary to make the induction in the proof work. Given examples we believe the result holds more generally, but the condition is necessary to preserve the structure of either G_1 or G_2 and the 1-join. So, some alternate proof technique appears necessary to obtain a more general result.

Proof. First we will induct on the number of edges supposing that G_1 and G_2 contain apex vertices v_1 and v_2. Suppose that G_1 and G_2 are stars. Then the result follows directly from theorem 4.1.9, since $I'(G_1) = 2^{|A|+1-3}$, $I'(G_2) = 2^{|B|+1-3}$, and by construction $G = K_{|A|,|B|}$ and so $I'(G) = 2^{|A|+|B|-3} = 2 \cdot 2^{|A|+|B|-4} = 2 \cdot 2^{|A|+1-3} \cdot 2^{|B|+1-3} = 2I'(G_1)I'(G_2)$ as desired. So the result holds for all stars. So we may suppose the theorem holds for all stars as our base cases. An analogous argument can be done to induct on edges of G_2 and thus we only present the induction on G_1. Suppose the result holds for all graphs G where G_1 and G_2 contain apex vertices v_1 and v_2 and where G_1 has $k \geq 0$ edges not incident to v_1.

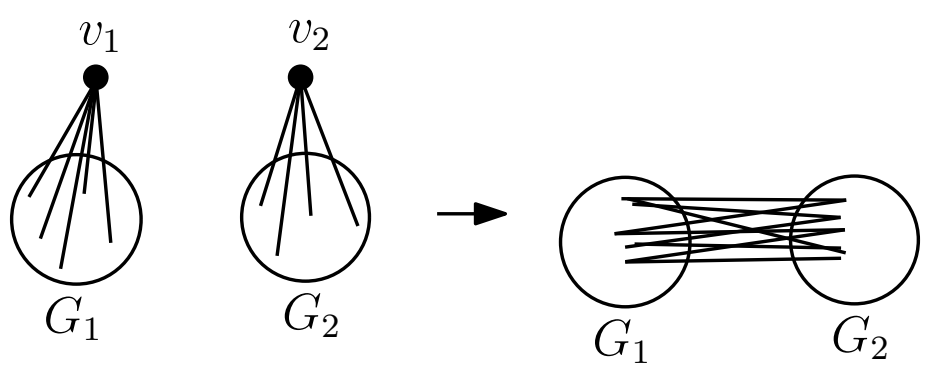


Figure 4.23: G_1 and G_2 are stars

We induct on the number of edges of G_1 not incident to v_1. Notice that the base cases have G_1 and G_2 as stars. Then consider G with $k+1$ edges in G_1.

If G_1 has a leaf, which is not v_1, then $I'(G) = 2I'(G-u)$ by theorem 4.1.5 and $I'(G_1) = 2I'(G_1 - u)$. Also $G - u$ satisfies the inductive hypothesis for $G_1 - u$ and G_2 since we have no vertices not adjacent to v_1 or v_2, so $I'(G) = 2I'(G-u) = 2(2I'(G_1 - u)I'(G_2)) = 2I'(G_1)I'(G_2)$ as desired. So we may assume that G_1 (and by the same logic G_2) has no leaves. By the same logic we may assume there are not two vertices u and w that satisfy the complete condition of theorem 4.1.5 for either G_1 or G_2. Since if a pair of locally complete vertices u and v exist then by theorem 4.1.5 we have $I'(G-u) = 2I'(G-u)$, and $I'(G_1 - u) = 2I'(G_1 - u)$ and G_1 and G_2 still satisfy the inductive hypothesis since v_1 is still an apex vertex. Thus again we see that $I'(G) = 2I'(G-u) = 2(2I'(G_1 - u)I'(G_2)) = 2I'(G_1)I'(G_2)$.

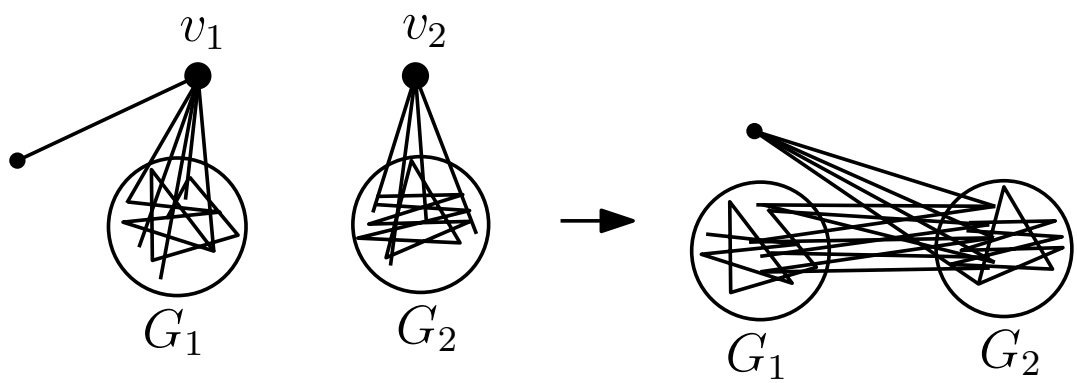


Figure 4.24: G_1 has a leaf

Now suppose that $u \neq v_1$ is also an apex vertex. Since we assume theorem 4.1.5 is not satisfied, there is a neighbor w of u that is not an apex vertex. Then $G-u$ has fewer edges and is the 1-join of $G_1 - u$ and G_2. Since v_1 is an apex vertex the theorem conditions are satisfied. So the inductive hypothesis holds for $G - u$. Now $G * u - u$ removes all edges from $G_1 - v_1$ to $G_2 - v_2$. So $G * u$ has two components and by lemma 4.1.1 the interlace invariant is zero. $G_1 * u$ isolates v_1 and thus has interlace invariant zero too. Further the 1-join between $G_1 * u - u - v_1$ and $G_2 - v_2$ results in a disconnected graph as there is no neighborhood of v_1 to add edges to. So the result holds in this case as we are multiplying by zero. Finally $G * uw - u$ for $w \in V(G_1)$ cannot modify edges in G_2 as anything in G_2 is in the common neighborhood of u and w. So we can still write $G * uw - u$ as a graph with $G_1 * uw - w - v_1$ and $G_2 - v_2$ to build the 1-join between the neighborhoods of v_1 and v_2, and v_1 is still an apex vertex since v_1 is also in the common neighborhood of u and w, so the theorem conditions are satisfied. We have fewer edges since we only add edges between neighbors of u but not w and for each such edge we can add we remove edges from that

vertex to all of $N(v_2)$. So for every edge we add we must remove at least one, and when we delete uw we remove an edge, so we remove at least one more edge than is added. Thus by the inductive hypothesis

$$\begin{aligned} I'(G) &= I'(G-u) + I'(G * uw - u) \\ &= 2I'(G_1 - u)I'(G_2) + 2 \cdot 0 \cdot I'(G_2) + 2I'(G_1 * uw - u)I'(G_2) \\ &= 2I'(G_2)(I'(G_1 - u) + I'(G_1 * u - u) + I'(G_1 * uw - u)) \\ &= 2I'(G_2)I'(G_1) \end{aligned}$$

as desired. Now we may suppose that v_1 is the only apex vertex in G_1.

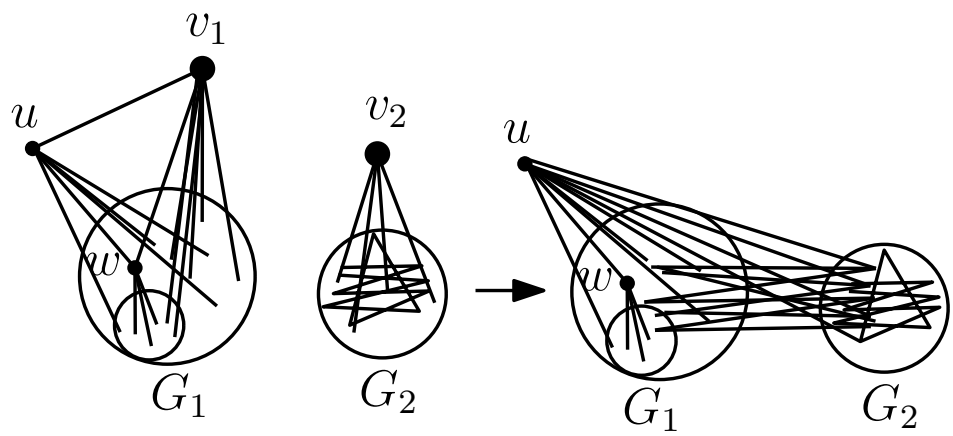


Figure 4.25: G_1 has an apex u besides v_1

Observe that as G_1 is not a star and does not have a leaf, there are at least two vertices other than v_1. Then let u and w be vertices of G_1 other than v_1 such that uw is an edge, which exists as otherwise we have as star.

So $G - u$ has fewer edges and can be expressed as the 1-join of $G_1 - u - v_1$ and $G_2 - v_2$. Next consider $G * u - u$. Since u is not a leaf or an apex vertex we have than $G * u$ modifies edges between G_1 and G_2 vertices and within G_1. Now u has degree two and neighbors w and v_1. So $G * u - u$ removes edges from w to $N(v_2)$ and in $G_1 * u - u$ this removes the edges from w to v_1. Now v_1 is no longer an apex vertex, but w is an isolated vertex so the interlace invariant it zero in both $G * u - u$ and $G_1 * u - u$. So $I'(G * u - u) = 2I'(G_1 * u - u)I'(G_2)$. Finally consider $G * uw - u$. Since v_1 is the only apex vertex, u and w have degree two and are neighbors, they have no neighbors other than each other and v_1. So $G * uw - u$ and $G_1 * uw - u$ are the same graphs as $G - u$ and

$G_1 - u$ and the theorem conditions are satisfied as v_1 is an apex vertex, w is a leaf now and all other vertices remain the same. Furthermore we hace fewer edges, so by the inductive hypothesis, we have $I'(G * uw - u) = 2I'(G_1 * uw - u)I'(G_2)$. Therefore

$$\begin{aligned}
I'(G) &= I'(G - u) + I'(G * u - u) + I'(G * uw - u) \\
&= 2I'(G_1 - u)I'(G_2) + 2I'(G_1 * u - u)I'(G_2) + 2I'(G_1 * uw - u)I'(G_2) \\
&= 2I'(G_2)(I'(G_1 - u) + I'(G_1 * w - u) + I'(G_1 * uw - u)) \\
&= 2I'(G_2)I'(G_1)
\end{aligned}$$

as desired.

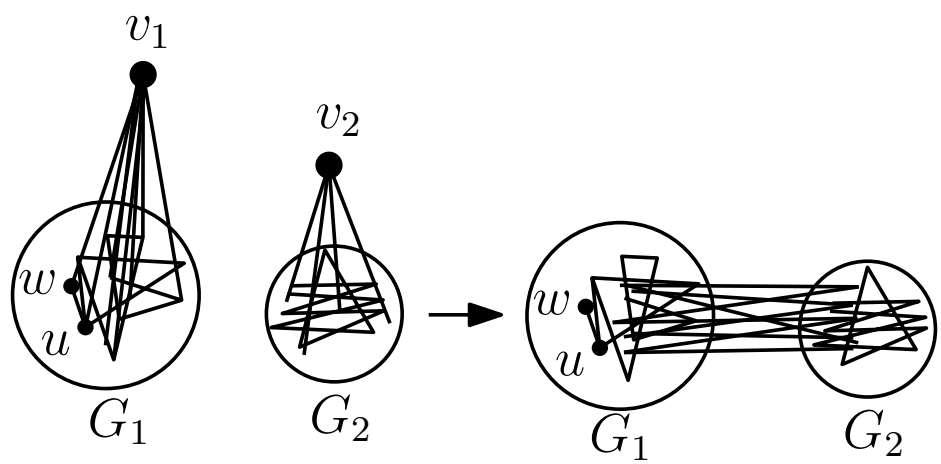


Figure 4.26: G_1 has no apex besides v_1

Hence the result holds for all G constructed from the 1-join where v_1 and v_2 in G_1 and G_2 respectively are apex vertices.

Thus it remains to show that if G_1 does not contain an apex vertex then the result still holds. Now we induct on the number of vertices k not adjacent to v_1. Clearly the $k = 0$ base case is when v_1 is an apex vertex and holds. Suppose the result holds for all G_1 containing at least $k \geq 0$ vertices not adjacent to v_1. If G_1 has a leaf u, which is not v_1, then $I'(G) = 2I'(G - u)$ by theorem 4.1.5 and $I'(G_1) = 2I'(G_1 - u)$. Now $G_1 - u$ has $k - 1$ non-neighbors of v_1. If $k - 1 = 0$ the inductive hypothesis holds as v_1 is an apex vertex in $G_1 - u$. If $k - 1 > 0$ then these vertices are also non-neighbors of v_1 in G_1 so either there are at least two which are neighbors and have degree two or we have at least one other leaf. So $G_1 - u$ still satisfies the inductive hypothesis and so does $G - u$. So

$I'(G) = 2I'(G-u) = 2(2I'(G_1-u)I'(G_2)) = 2I'(G_1)I'(G_2)$ as desired. So we may assume that G_1 has no leaves. Now let u be a vertex not adjacent to v_1. Then either u has a neighbor w that is not adjacent to v_1 or all neighbors of u are adjacent to v_1 in G_1. If w that is not adjacent to v_1 in G_1 exists then by the theorem conditions we have w and u have degree two, so $G-u$ has a leaf w, $G*u-u$ has a leaf w, and $G*uw-u$ has a leaf w, so the theorem conditions are satisfied as any other degree two vertices still satisfy the theorem conditions or are now leaves. So we satisfy the inductive hypothesis for G_1-u, G_1*u-u, and G_1*uw-u with G_2 respectively as the edges from v_1 are not modified by any operation. So

$$\begin{aligned}
I'(G) &= I'(G-u) + I'(G*u-u) + I'(G*uw-u) \\
&= 2I'(G_1-u)I'(G_2) + 2I'(G_1*u-u)I'(G_2) + 2I'(G_1*uw-u)I'(G_2) \\
&= 2I'(G_2)(I'(G_1-u) + I'(G_1*w-u) + I'(G_1*uw-u)) \\
&= 2I'(G_2)I'(G_1)
\end{aligned}$$

as desired.

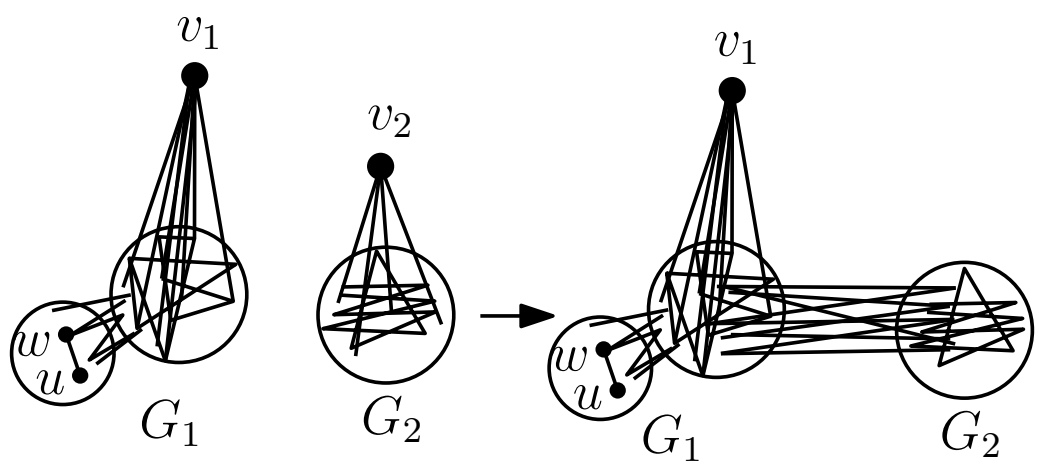


Figure 4.27: G_1 has two adjacent non-neighbors of v_1

Thus suppose that all neighbors of u are adjacent to v_1 and u is not a leaf. Then by the theorem conditions G_2 satisfies v_2 being an apex vertex or v_2 having two non-neighbors which are in turn neighbors of degree two or a leaf, and we may choose an edge and incident

vertex in G_2 using a previously shown case to obtain So

$$\begin{aligned} I'(G) &= I'(G-u) + I'(G*u-u) + I'(G*uw-u) \\ &= 2I'(G_2-u)I'(G_1) + 2I'(G_2*u-u)I'(G_1) + 2I'(G_2*uw-u)I'(G_1) \\ &= 2I'(G_1)(I'(G_2-u) + I'(G_2*w-u) + I'(G_2*uw-u)) \\ &= 2I'(G_1)I'(G_2) \end{aligned}$$

as desired. Thus the result holds for all graphs G with appropriate G_1 and G_2. ■

The condition that G_1 or G_2 have either no vertices not adjacent to the specified vertex or a pair of adjacent vertices which are not adjacent to v_1 or v_2 did not initially appear to be a required condition for the theorem. When constructing examples and computing them in small cases the result held regardless of this property. However, when working with these examples we were unable to find a way to prove that graphs not satisfying this condition would always have the given product property. We tested a number of graphs for which there was a single non-neighbor and there was no choice for vertex and edge that did not affect the graph on either side of the 1-join or create a graph that no longer had the appropriate 1-join when performing operations in G. All still satisfied the product condition but the recursion was not helpful in proving why. The issue seems to lie in how the edge and local complementation operations act in G_1 with an apex vertex with edges between its neighbors or a single non-neighbor. This results in graphs that are no longer have apex vertices and often have no apparent structural pattern. Further local complementation did not seem to fix this issue. Additionally, when doing the complementation operations the graphs would sometimes remain connected and sometimes become disconnected. This led to inconsistency in what terms of the recursion contributed to the interlace invariant. While we have constructed no explicit examples of graphs that can be expressed as a 1-join of G_1 and G_2 with specified vertices that do not satisfy the product property, we also have not been able to find any reason the result should work in a more general sense. Perhaps further work with edge complementation and vertex complementation on these graphs will yield more information.

4.3 *k*-Occurrence Chord Diagrams

While the interlace polynomial is only related to the 4-regular Martin polynomial through chord diagrams and interlace graphs, the Martin polynomial and Martin invariant are both defined on $2k$-regular graphs. Discovering a class of graphs or hypergraphs and an associated interlace-like polynomial to extend this relationship is an open question. One

primary issue with constructing such an extension, is that given a vertex in a $2k$-regular graph there are $(2k-1)!!$ ways to resolve the vertex. So we need $(2k-1)!!$ operations in the recursive formula. This grows quite quickly which makes the question difficult to address. In this section, we discuss the work done in generalizing chord diagrams to k-occurrence chord diagrams and how the vertex resolutions in the $2k$-regular Martin polynomial calculation impact the k-occurrence chord diagrams. In Chapter 5 we will discuss how this has been used to make progress toward an interlace-like polynomial for cases where $k > 2$.

A *k-occurrence word* is a word that contains exactly k occurrences of each letter. A *k-occurrence chord diagram* is constructed from a k-occurrence word by writing the word as vertex labels around a cycle and then adding a chord between all pairs of vertices with the same label. Two letters are interlaced by the total number of times any pair of their chords cross. So for a k-occurrence word there are $\binom{k}{2}$ chords between pairs of letters and if a chord for a letter a intersects a chord for a letter b it will intersect k such chords, so we choose to take the number of crossing divided by two. Figure 4.28 shows an example of this for $k = 3$ and $k = 4$. Another way to view these chords is as hyperedges with k vertices and the intersections of a pair of chords show how the hyperedges interlace their endpoints.

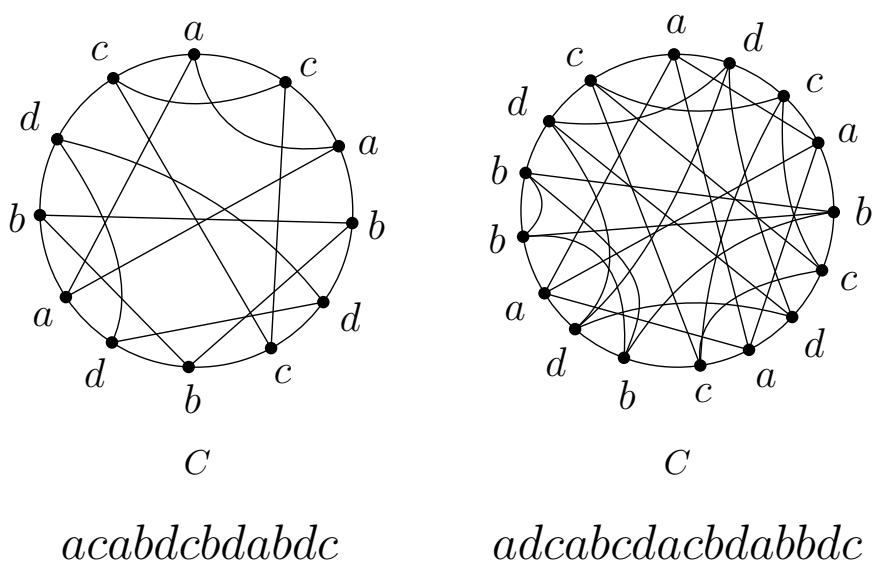


Figure 4.28: A 3- and a 4-occurrence chord diagram

We now generalize the chord diagram operations of skip, flip, and break. The skip

operation, which corresponds to removal of a letter from the word, remains the same regardless of k. We can, for any specified vertex start the word of the k-occurrence diagram at that letter and proceed clockwise. Writing the word in this manner yields $vw_1vw_2...vw_k$. So we can define the flip and break operations with respect to v. For a k-occurrence word we have i-flips and i-breaks for $i \in \{1, ..., k-1\}$, where an i-flip is found by reversing i of the w_j to $\overleftarrow{w_j}$ segments, $2 \leq j \leq k$, of the word and deleting all instances of v. An i-break is found by turning the word into multiple words by breaking it in i places. We break the word between the last letter of w_j $1 \leq j \leq k-1$, for i choices of j, and the next instance of v and then delete v. We mark the break locations with $\|$ and this yields an ordered sequence of words, which we may call a sentence. Given a sentence we may glue the words together by taking a letter a in any pair of consecutive words w_i and w_{i+1} and by a rotation of the words write the letters so that both start at a. We then simply concatenate the words to form a single new word. By repeatedly combining consecutive words we may turn a sentence back into a single word. These operations can also be combined. We also introduce a new operation that is trivial in the 4-regular case called a twist. The *twist* is obtained by taking a pair of w_i and w_j, $2 \leq i \neq j \leq k$ and swapping their position and skipping v. See table 4.3 for examples of these operations. For any operation we then may take the word and construct a new k-occurrence chord diagram. It is important to note that any w_i might be empty and that not all combinations of these operations will necessarily result in a distinct word. However, we can directly relate these operations to revolving a vertex when computing the Martin polynomial on a $2k$-regular graph. Note that when we take the word for our k-occurrence chord diagram from a circuit and resolve a vertex we only modify how the chosen circuit passes through the vertex we are resolving, but this does not guarantee that different vertex resolutions result in distinct graphs. Thus we cannot change the order of vertices in the circuit in segments between instances of the vertex being resolved. This means that different sequences of operations on a k-occurrence chord diagram may result in the same final k-occurrence chord diagram.

Lemma 4.3.1. *Given a $2k$-regular graph with Eulerian circuit $\overrightarrow{E}$, associated k-occurrence chord diagram defined by the word of $\overrightarrow{E}$, and a vertex v, a choice of transition at v in G corresponds to either a skip of v or a sequence of breaks, twists, and flips at v.*

Proof. Let $vw_1....vw_k$ be the word associated to the Eulerian circuit and k-occurrence chord diagram. When resolving v in G if we follow the Eulerian circuit through v we obtain $w_1...w_k$ which is a skip of v. Otherwise each time we enter v we have a choice of what portion of the Eulerian circuit, or which w_i we follow next. Rearranging the order of these segments of w_i corresponds to a sequence of twists, taking a segment in the opposite orientation from $\overrightarrow{E}$ corresponds to a flip of that segment and if we take a segment, or

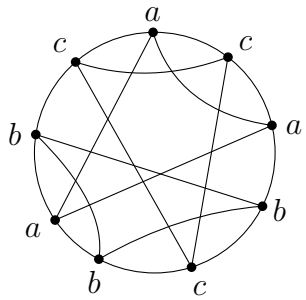
multiple segments of the Eulerian circuit and close them at v by taking the last vertex of some w_j and sending it to the first vertex of w_k for $k < j$ by passing through v. This separates the Eulerian circuit into multiple pieces where each pieces now has its own word and this corresponds to a break. So every time we pass through v we choose a flip, twist, or break. This results in a sequence of breaks, twists, and flips determined by the resolution of v. ■

We further claim that the total number of ways to split, twist, flip, and break a word associated to a k-occurrence chord diagram where we choose the sequence of operations to be distinct is $(2k-1)!!$, which is the number of operations needed for constructing an interlace-like polynomial. To ensure our operations are distinct we will set up the words in a specific fixed manner. We define *valid k-occurrence chord diagram operations* on a word $vw_1...vw_k$ to be any sequences of operations that fix vw_1, perform all twists first, then all breaks, and then all flips, where we never twist w_1, never break at every instance of v unless there are no flips or twists, and we never flip w_1 nor any w_j that has been isolated by a break or is the first w_i in a segment of a word created by a break. By preforming operations in this manner and never altering the first segment of any word we know that each sequence of operations can be defined distinctly from each other sequence as the word structure is fixed. Otherwise we would have to consider how the dihedral symmetry of the words might impact our operations. See example 4.3 for an example of these operations and how they correspond to the vertex resolutions in the computation of the Martin polynomial.

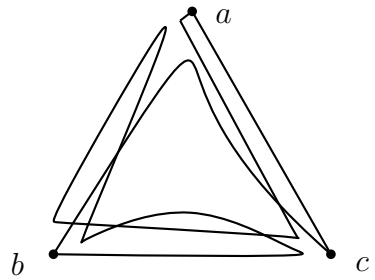
Lemma 4.3.2. *For a k-occurrence chord diagram and associated word $vw_1...vw_k$ there are $(2k-1)!!$ valid operation of the word which create $(2k-1)!!$ k-occurrence chord diagram for $k \geq 2$.*

Proof. We begin by counting the number of ways to arrange $vw_1....vw_k$. There are $(2k)!$ ways to rearrange the letters which counts all twists, but we don't care about the $k!$ ways to rearrange the v's. Furthermore, for each arrangement there are 2^{k-1} ways to place flips and 2^{k-1} ways to place breaks. However, given a string with twists, flips, and breaks, there are 2^{2^k} ways to cyclically write the word. So the total number of valid words is $\frac{(2k)!\cdot(2\cdot(2^{k-1}))}{2^{2k}\cdot k!} = \frac{(2k)!\cdot 2^k}{2^{2k}\cdot k!} = \frac{(2k)!}{2^k\cdot k!} = (2k-1)!!$. Each word can be used to construct a chord diagram and this gives $(2l-1)!!$ k-occurrence chord diagrams. ■

Example 4.3.3. For a 2(3)-regular graph in the Martin polynomial case this corresponds to a 3-occurrence chord diagram and there are fifteen valid sequences of flips, twists, and breaks. See table 4.3 for an example of a flip, a twist, and a break.



C



acabcbabc Eulerian Circuit

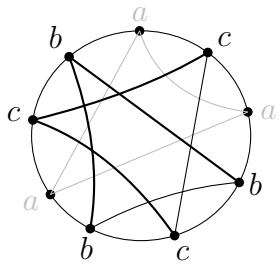
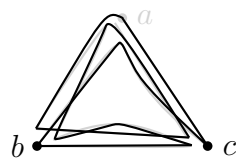
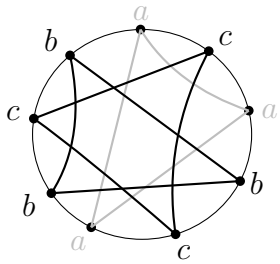
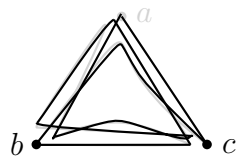
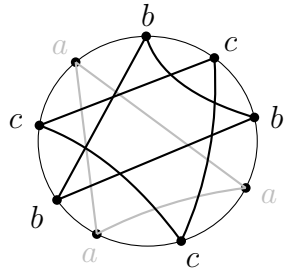
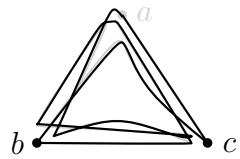
word	k-occurence chord diagram	Eulerian transition
acabcbacb fliped $w_3 = cb$		
acabcabcb twist w_2 and w_3		
acabcb||*abc* *bcbacabca* break after w_2		

Table 4.3: A flip, twist, and break on a 3-occurrence chord diagram

Chapter 5

Future Directions

5.1 Open Questions Inspired by Grid Walking

When looking at the extended Tutte polynomial of graphs, signed graphs, or the bipartite representation of graphs the walks and marked nodes provide a nice way to visualize the monomials in the polynomial. Since Tutte grid walks are not a traditional method by which the Tutte polynomial is visualized there are a variety of questions we can ask about the Tutte grid walks themselves. In this section we will explore some of the ideas and question that we have regarding Tutte grid walks.

Grid walking is not a new concept. More commonly called lattice paths, there has been a variety of work done with enumerating lattice paths with various properties. A relatively recent survey of such results can be found in [83]. Refining what Tutte grid walks are valid for a specific edge ordering and looking for lattice path enumeration results relating to similarly defined walks might yield a new method for counting maximal spanning forests or provide interesting connections between maximal spanning forests of a graph and some other objects. To start this investigation a first step would be to determine which walks are never constructed by any edge ordering and maximal spanning forest. We suspect that any grid walk that remains inside the grid should correspond to a Tutte grid walk that appears under some edge ordering, but have not proven this. We would also want to investigate the multiplicity of walks across all edge orderings. We do know that not every walk appears in every edge ordering, as most grids have more walks than maximal spanning forests, and that in a given edge ordering no walk appears twice. However, we do not know if there are certain walks that appear in more of the edge orderings than others beyond the all deletion and all contraction walks that appear in every edge ordering. We

can easily tell that many Tutte grid walks appear across multiple edge orderings and some clearly appear with greater frequency, but we don't have any general results that count how many edge orderings a specific Tutte grid walk appears for. It seems likely there would be some nice counts of the number of of edge orderings for which a walk will appear, but we have not investigated this question yet. A more thorough understanding of these questions would provide a starting point from which to look for classes of lattice paths that might be enumerated similarly.

When working with the Tutte polynomial we prioritized basis information, or maximal spanning forests. When looking at the activity definition of the polynomial it is clear that different edge orderings change which maximal spanning forest corresponds to which monomial, but the marked nodes on the grid remain unchanged. This leads to the question of whether different edge orderings could be considered to order the maximal spanning forests in some manner. One way we have tried to examine this question is by creating a poset on maximal spanning forests of a graph G. For this poset construction we define a covering relation used to construct the poset. Given an ordering on the edges of G, denoted by $\overrightarrow{E}$ and $T_G(x, y, c, d; \overrightarrow{E})$, and for maximal spanning forests T and T' we say $T < T'$ if T' contains fewer total deletion and contraction than T and differs from T by swapping consecutive, in the edge ordering, pairs of edges of $cd \leftrightarrow dx$ or $dx \leftrightarrow cy$ or $dc \leftrightarrow cc$ or $cd \leftrightarrow dc$ if possible. See figure 5.2 and 5.3 for this poset under two different edge orderings. If a node of the grid is marked with multiplicity then there are multiple monomials that contain the same number of x and y terms. These monomials appear in the same level of the poset and in all tested examples have the same smaller and larger elements. If a node is marked with multiplicity then the terminal minors associated to it are isomorphic. Because nodes marked with multiplicity have the same number of deletions and contractions they are incomparable in the poset. Another way we could construct a poset is by defining the covering relation where $T < T'$ if in $T_G(x, y, 1, 1; \overrightarrow{E})$ the monomial from T divides the monomial from T'. Figure 5.2 and 5.3 shows both these methods as they were the same in this small graph. As mentioned, when constructing these posets it was observed in examples that if two maximal spanning forests corresponded to the same monomial, then they were isomorphic. This is not in general true and we can construct counterexamples easily for disconnected graphs. See figure 5.1. We do not currently have a counterexample, if one exists, in the connected case.

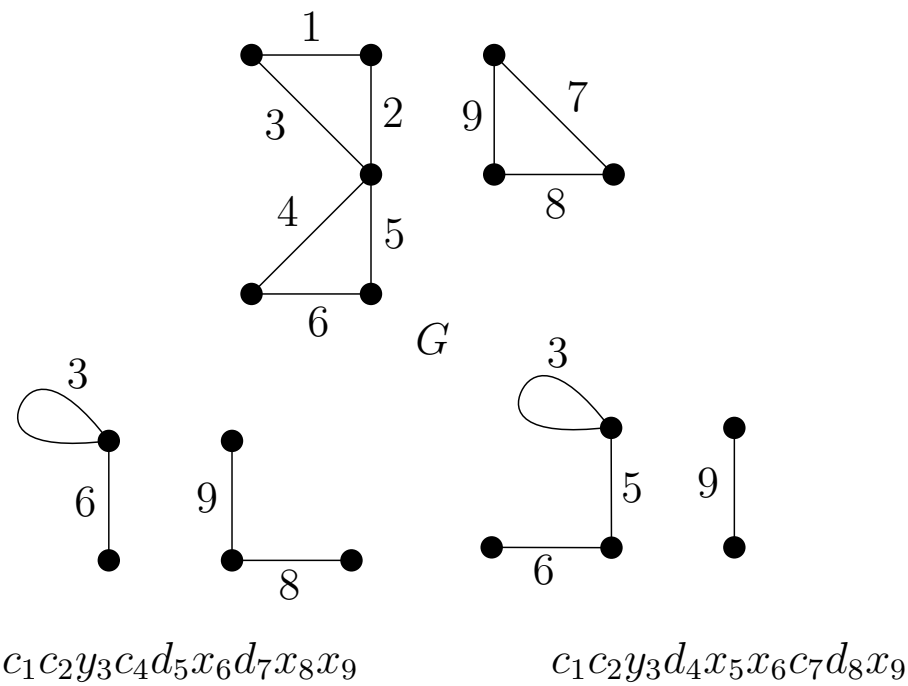


Figure 5.1: Two non-isomorphic terminal minors corresponding to the same marked node

Another way to think about the poset is that it is based on how the terminal minors or marked nodes are discovered. Starting at the bottom left corner of the grid we begin moving up and right to look for marked nodes. So terminal minors and their associated spanning trees are smaller than others in the poset if you can reach the node by moving either right or up from the current position. This creates a special sort of almost lexicographical order on the monomials. Where a monomial is higher in the lexicographical order if the number of instances of x and/or the number of instances of y are greater than another monomial. This can be seen in figure 5.2 and 5.3 where $c_1c_2c_3y_4y_5 > c_1d_2c_3c_4y_5$ but is incomparable to monomials containing x like $c_1c_2d_3x_4y_5$ and $c_1d_2c_3d_4x_5$. However, $c_1d_2c_3c_4y_5 < d_1x_2c_3c_4y_5$ so there are relations between terms with only x or y and terms with both. In the grid we can see this as the walk the the xy node must pass through either xy^0 or x^0y but xy^0 and x^0y are incomparable as we cannot walk through one on a walk to the other. What this is doing is taking the marked nodes on the grid and the associated maximal spanning forests and constructing a poset by considering which maximal spanning forests and their marked nodes are in walks to other marked nodes with their associated maximal spanning forest, with the condition that two maximal spanning forests corresponding to the same marked node are incomparable.

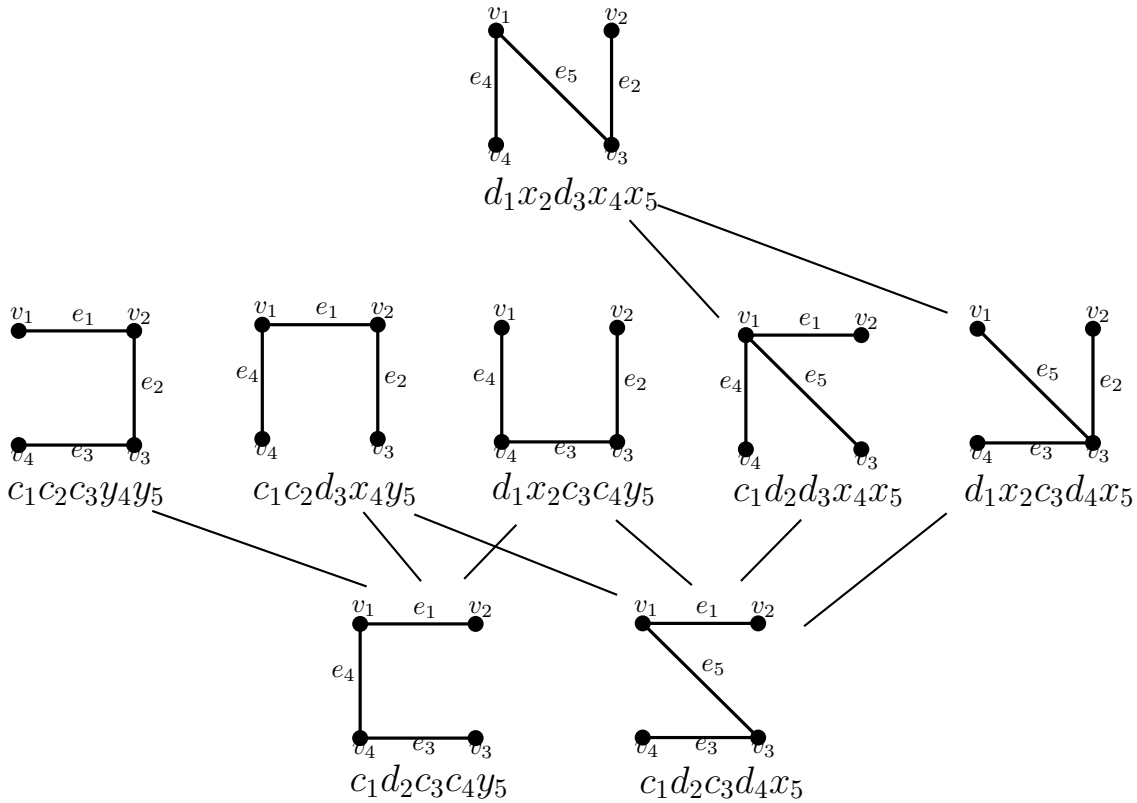


Figure 5.2: A spanning tree poset for an edge ordering

While we can create a poset based on discovering the marked nodes and maximal spanning forests from walks starting in the lower left corner of the grid we can also flip this poset. This corresponds to discovering the marked nodes starting at the upper right corner of the grid. We are now prioritizing monomials with higher number of x and y. So this poset is based on distance from the original graph in terms of fewest number of deletions and contraction needed to find the terminal minor and monomial. This also means that we consider spanning trees with more active edges to be smaller in the flipped poset. Keeping the poset starting from discovery from the bottom left corner therefore says that $T < T'$ if T' contains the activities of T, ignoring the internal or external type. In figure 5.2 and 5.3 this can be see by $c_1c_2d_3x_4y_5 > c_1d_2c_3d_4x_5$ as e_5 is active in both, but externally active in one and internally active in the other. So this poset shows activity containment.

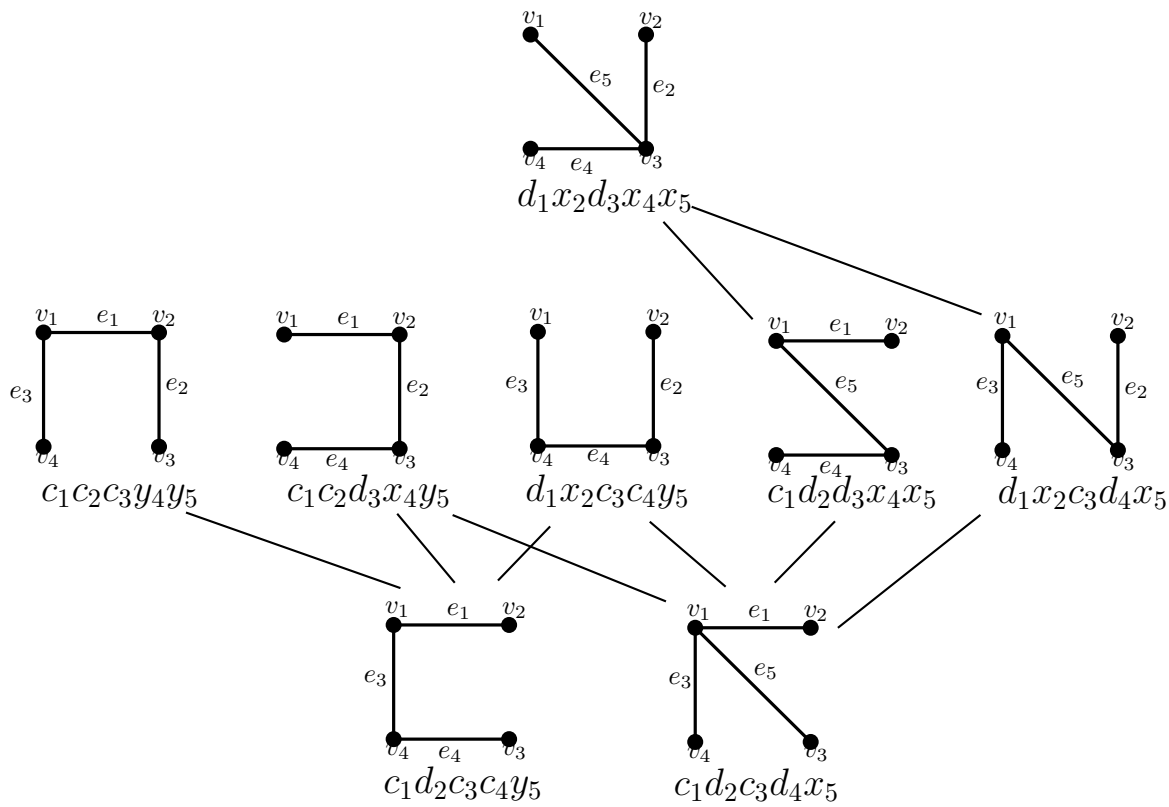


Figure 5.3: A spanning tree poset for another edge ordering

Thinking about activities and the Tutte grid walks leads to a natural question about edge order reconstruction. If given a graph and a set of Tutte grid walks for some unknown edge ordering and marked nodes in the grid with multiplicity and their associated maximal spanning forests, can we recover the edge ordering used. If we have the associated trees to each marked node then it seems likely we could construct an edge ordering that corresponds to the Tutte grid walks, but it seems unlikely that we can reconstruct the monomial with edge labeling otherwise. The walk tells us the number of deletions, contractions, y's, and x's but it removes some of the information about the order in which they occurred due to the deshuffle. We can make some assumptions, for example the last edge must be an x or y and the first edge is c or d if the graph has no loops or isthmi initially, but we have removed enough information about the edge ordering that it seems unlikely it can be reconstructed. For the same reasons it seems even more unlikely that given a single walk the associated maximal spanning forest or activities can be determined.

When working with the grids for bipartite representations of graphs in section 3.5 we

made the observation that it appears we can move between monomials as consecutive operations may be replaced with others such as $cd \leftrightarrow dx$ and $cy \leftrightarrow dx$. We also saw these replacement schemes when defining one containment relation for a poset on maximal spanning forests of a graph. There might be further results we could obtain to move between maximal spanning forests or monomials using these replacement operations. The validity of the replacements likely depends on a well chosen edge ordering, but if it could be proven that these replacements in monomials can be made it would provide a nice way to move between monomials, maximal spanning forests, and Tutte grid walks.

5.2 Variations of Tutte-like Polynomials for Signed Graphs

While we have chosen to generalize our extended signed graph Tutte polynomial so that each term is a basis, this is not the only idea for generalizing the polynomial. We present here two alternate methods.

For the first alternative idea we will discuss, we define the total loading of a basis. The *total loading* of a maximal spanning forest T of a signed graph G, denoted by $\overline{\mathcal{L}}(T)$, is the subgraph $T + \{e \in G - T | T + e \text{ contains a negative cycle}\}$. If $T + e$ contains no negative cycles (the graph is balanced) for any edge e, then we set $\overline{\mathcal{L}}(T) = \{T\}$. We can use this to compute a signed graph Tutte polynomial, presented here by activities, though we can adapt this to loading monomials for a deletion contraction version as well by adjusting the monomial loading to a total monomial loading in the same manner.

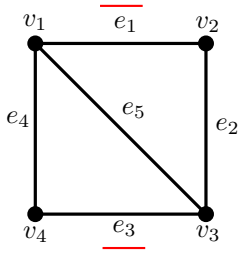
Definition 5.2.1. *Let Σ be a signed graph with total edge ordering $\overrightarrow{E}$ and set of maximal spanning forests $\mathcal{T}$. Define the* total activity extended Tutte polynomial *by:*

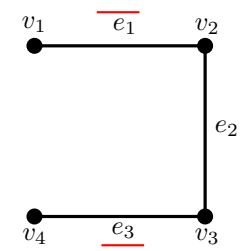
$$\overline{T}_{\Sigma}(\mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{d}, \mathbf{f}, \mathbf{r}; \overrightarrow{E}) = \sum_{\substack{\overline{L}(T), \\ T \in \mathcal{T}}} \prod_{e \in \overrightarrow{E}} A_B(e)$$

where

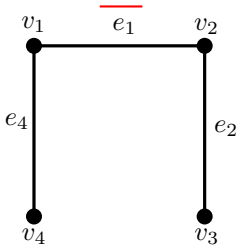
$$A_B(e) = \begin{cases} x_e & \text{if } e \text{ is internally active,} \\ c_e & \text{if } e \text{ is internally inactive,} \\ y_e & \text{if } e \notin B - T \text{ is externally active,} \\ d_e & \text{if } e \notin B - T \text{ is externally inactive,} \\ f_e & \text{if } e \in B - T \text{ and is externally active,} \\ r_e & \text{if } e \in B - T \text{ is externally inactive.} \end{cases}$$

This definition still captures the basis information, just more condensed. As seen in figure 5.4 we compute this polynomial almost in the same manner as that in chapter 3 section 3.4, however we compute the r, f, d or y for all the external edges. As the graph is unbalanced we have at least one r or f in every spanning tree calculation for this graph and we can see that for the four spanning trees including e_5 we get that all external edges are frustrated. This polynomial is nice from the perspective of having the same number of terms as maximal spanning forests of the underlying graph, but we lose the nice ability to visualize the Tutte grid walks as all having the same length. Since the f and r represent movement between the two grids, which we think of as a stall, we might have more than one such move and currently f or r move either up of down and so two copies of r or f in the same monomial would walk outside the grid. Thus we would need a new notion of these steps in the Tutte grid walks. Furthermore, consecutive f or r could try and step off the two grids if we don't modify the definitions of these steps in the grid. Since the f step, for example, moves up to a higher grid and up a square. We could fix this issue by simply defining them to move to the alternate grid but we still have differing numbers of steps between the two grids for different maximal spanning forests. There is the interesting question of whether or not this is better in the sense of the number of steps between grids indicating the number of different bases that can be obtained by adding an edge to the maximal spanning forest for the walk. However, it is then less clear how to mark nodes with multiplicity since the set of marked nodes should just be the same as the set of nodes marked by the underlying graph. Perhaps the multiplicity could be adjusted by the number of stalls in the walk. It is however clear that this polynomial is closely related to that used in chapter 3 as we can simply imagine collapsing or expanding terms based on the x and c edges. So in essence computing one of these polynomials gives the other, though the translation between the two has not been formalized.

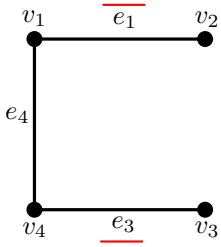




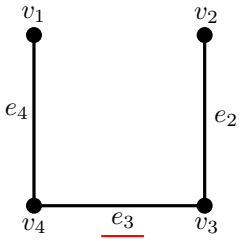
Internal		External	
e_1	c_1	e_4	y_4
e_2	c_2	e_5	f_5
e_3	c_3		



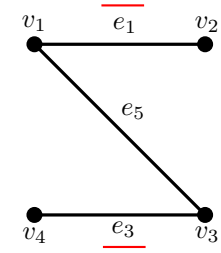
Internal		External	
e_1	c_1	e_3	d_3
e_2	c_2	e_5	f_5
e_4	x_4		



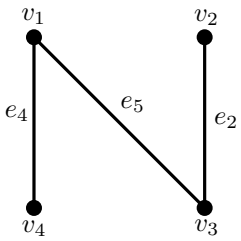
e_1	c_1	e_2	d_2
e_3	c_3	e_5	f_5
e_4	c_4		



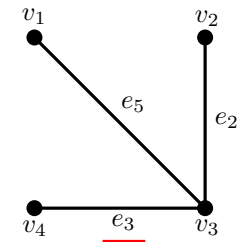
e_2	x_2	e_1	d_1
e_3	c_3	e_5	f_5
e_4	c_4		



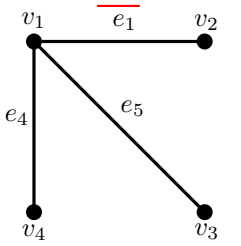
e_1	c_1	e_2	r_2
e_3	c_3	e_4	r_4
e_5	x_5		



e_2	x_2	e_1	r_1
e_4	x_4	e_3	r_3
e_5	x_5		



e_2	x_2	e_1	r_1
e_3	c_3	e_4	r_4
e_5	x_5		



e_1	c_1	e_2	r_2
e_4	x_4	e_3	r_3
e_5	x_5		

Figure 5.4: An alternative idea for computing a graph Tutte polynomial via activities

Another idea was to try and generalize the polynomial recursively with the f and r variables as part of the recursion. This is not equivalent to definition 5.2.1 since it sometimes has a monomial containing an r_e that does not form a negative cycle with the x and c in that monomial or has an edge e that is deleted and does form a negative cycle, and so should be an r_e, but is not.

Definition 5.2.2. *Given a signed graph Σ and total ordering on the edges $\overrightarrow{E}$ define the f, r-extended* polynomial *recursively by deletion and contraction as follows:*

$$T_G^*(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}) = \begin{cases} 1 & \text{if } E = \varnothing, \\ x_e \cdot T_{\Sigma/e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}-\{e\}) & \text{if } e \text{ is an isthmus in } \Sigma, \\ y_e \cdot T_{\Sigma\setminus e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}-\{e\}) & \text{if } e \text{ is a positive loop in } \Sigma, \\ f_e \cdot T_{\Sigma\setminus e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}-\{e\}) & \text{if } e \text{ is a negative loop in } \Sigma, \\ d_e \cdot T_{\Sigma\setminus e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}-\{e\}) & \text{if } e \text{ is a positive link,} \\ \qquad +c_e \cdot T_{\Sigma/e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}-\{e\}) & \qquad \text{non-isthmus edge in } \Sigma, \\ r_e \cdot T_{\Sigma'\setminus e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}-\{e\}) & \text{if } e \text{ is a negative link,} \\ \qquad +c_e \cdot T_{\Sigma'/e}(\mathbf{x},\mathbf{y},\mathbf{c},\mathbf{d};\overrightarrow{E}-\{e\}) & \qquad \text{non-isthmus edge in } \Sigma, \end{cases}$$

where e is the smallest edge in $\overrightarrow{E}$, $\mathbf{x}$ consists of all subscripted variables of the form x_e, and similarly for $\mathbf{y}, \mathbf{c}, \mathbf{d}$, and Σ' is found by switching one end of e.

It would be interesting to see if either of these definitions can be related to the signed graph Tutte polynomial in [57] or modified to then relate to other known definitions. Currently neither has the necessary underlying basis information for lift matroids but could still prove interesting in some other manner. I am particularly interested in if there is a modification that can be made to how edges are assigned to r in definition 5.2.2 that would make this definition equivalent to definition 5.2.1. Perhaps we might be able to construct a rule to remove the r that can show up for terms where that edge doesn't make a negative cycle in relation to the spanning tree formed by the x and c terms. It would also be nice to make this definition switching equivalent, which it is also currently not. Example 5.2.3 demonstrates some of the issues we find with this definition.

Example 5.2.3. Comparing the polynomials obtained from figure 5.4 to figure 5.5 we can see that there are immediate differences. For example we get $r_{e_1}x_{e_2}c_{e_3}c_{e_4}f_{e_5}$ in one and $d_{e_1}x_{e_2}c_{e_3}c_{e_4}f_{e_5}$ in the other. We don't expect to get e_1 as an r in this term because in the spanning tree defined by $\{e_2, e_3, e_4\}$ we don't get a negative cycle, but at the time

e_1 was deleted and contracted using definition 5.2.2 the edge was in the negative cycle e_1, e_2, e_5. However, we do want e_1 to show up as r_{e_1} with the spanning tree from the term $r_{e_1}x_{e_2}r_{e_3}x_{e_4}x_{e_5}$. This seems to indicate that there is no nice way in which to find the r terms while performing the deletions and contraction.

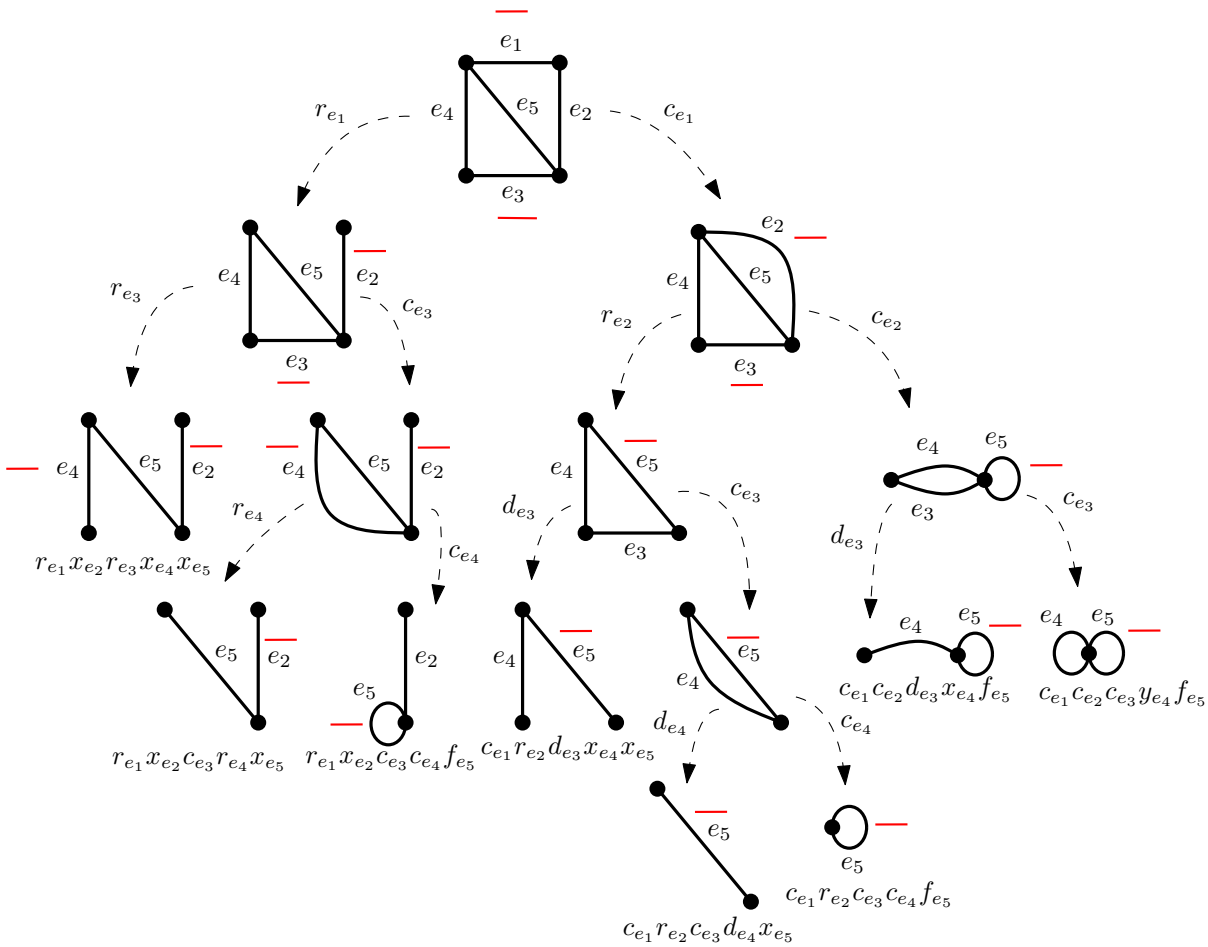


Figure 5.5: An alternate deletion contraction signed graph Tutte polynomial

Translating y to f can occur during the deletion and contraction process as we know a cycle is negative if under switching the loop remaining is negative, but there doesn't seem to be a nice choice for r or d at the time the corresponding edge is deleted. This version is also not switching equivalent. This can be seen by looking at the graph after we contract e_1. We need to switch e_2 to contract further. We chose to switch the end that is incident

to e_3 and e_5 which meant that e_3 was then positive and so yielded d_3 in the next step. However, we could have switched the end incident to e_5 and e_4 and e_3 would have remained negative in the next step yielding r_3.

5.3 The Tutte Polynomial and Hypergraphs

While we would ultimately like to define a Tutte type polynomial for oriented hypergraphs there are several open questions. In this section we are going to examine an example of a hypergraph, its bipartite representation, and two different choices for the sequence of incidences we delete and contract.

When working with graphs and trying to use the extended Tutte polynomial of their bipartite representation to compute Tutte polynomial of the graph we addressed incidences in the bipartite representation in the order of the edges in the graph. Trying this technique with hypergraphs can be seen in figure 5.6. In this figure we can see how this doesn't really give clear operations on the hypergraph. Deletion of an incidence leaves the graph bipartite and is clear on the hypergraph by simply deleting the incidence. However, contraction in the graph becomes much more problematic. In the graph case we could resolve issue by deleting or contracting a single incidence as all edges have size two, but this is not true in the hypergraph case. So the number of problematic edges created increases when working with edges of larger size. While there are sequences of deletions and contractions that yield other bipartite graphs it is not always clear how to relate these back to the hypergraph. We would need to turn vertices into edges or edges into vertices and there are a lot of half edges. Figure 5.7 shows how we can construct some of these hypergraphs with half edges.

The Tutte polynomial is invariant under an operation called cleaving. Recall that we can use cleaving to break a graph at a cut vertex. Doing this repeatedly we may split a graph at a cut vertex by replacing that vertex with k non-adjacent copies of the vertex, one for every component of the graph after the cut vertex is removed and preserving the edges incident to these new vertices in each component. Because the Tutte polynomial is multiplicative over 1-point-joins and connected components, we can use this cleaving operation. There might be some way to cleave edges instead of vertices and thus reattach half edges in a manner that merges them with another edge. As long as we don't construct cycles we suspect we could preserve the polynomial.

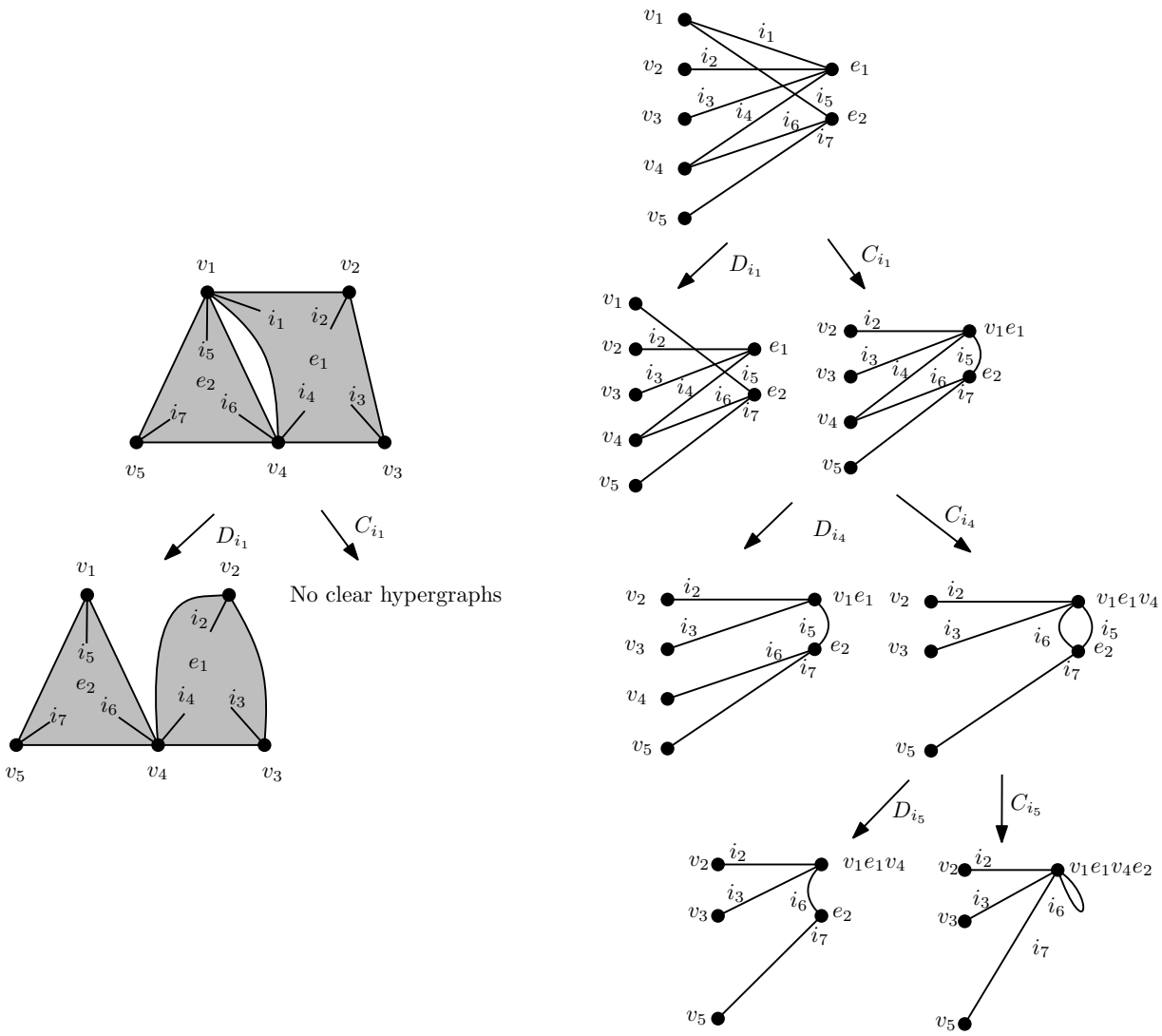


Figure 5.6: Deletion and contraction via the bipartite representation following edge order

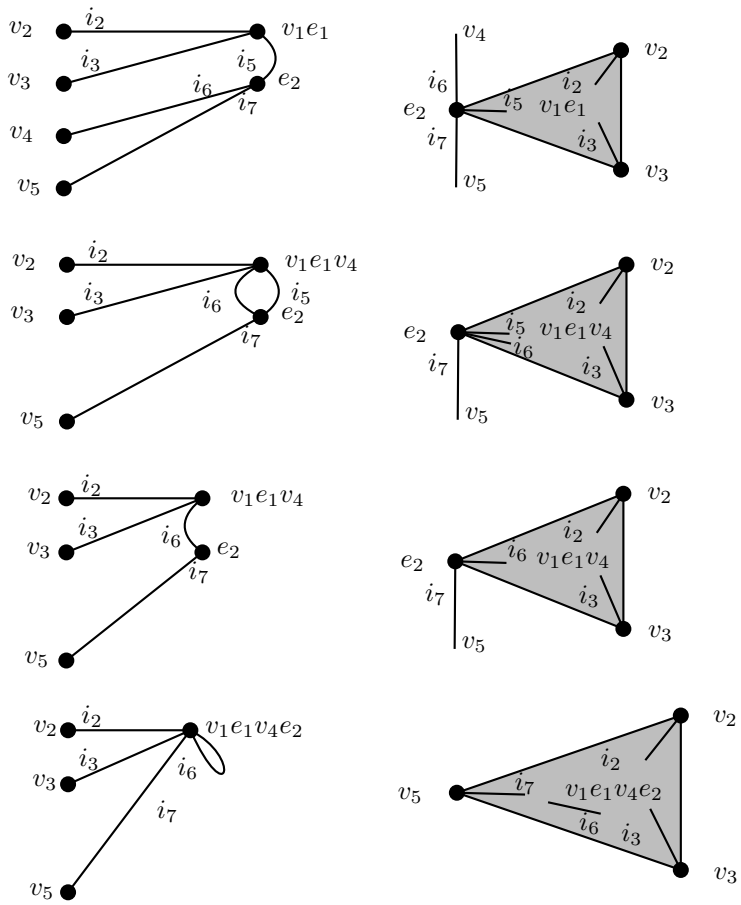


Figure 5.7: Hypergraphs from the bipartite instance in figure 5.6

There does appear to be choice of incidence ordering that consistently yields hypergraphs related clearly to the original hypergraph. As seen in figure 5.8 we can deal with all incidences possible related to a vertex in sequence. This is similar to the method used in the graph case, though in the graph case we dealt with all incidences associated to a single edge in order as opposed to a vertex. This more cleanly seems to result in hypergraphs. We do end up with an interesting question about loops when we have an edge incident to a vertex multiple times. We can see that after contracting i_1 and i_5 in figure 5.8 we have i_4 and i_6 both incident to the only remaining edge, labeled $v_1e_1e_2$. In the bipartite representation, we have a double edge that corresponds to this loop in the hypergraph, so it appears we can delete and contract one of these incidences. However, this removes a loop and a loop should not be deleted or contracted. We also end up with an incidence

that is floating inside an edge. It seems likely that if we have multiple incidences none can be deleted or contracted as this destroys loops, but then it is also unclear what should be done if every incidence is in a loop and we also have cycles. Perhaps the solution is to delete and contract incidences only when in cycles that are not loops or to delete and contract incidences in both cycles and loops but pick up a y.

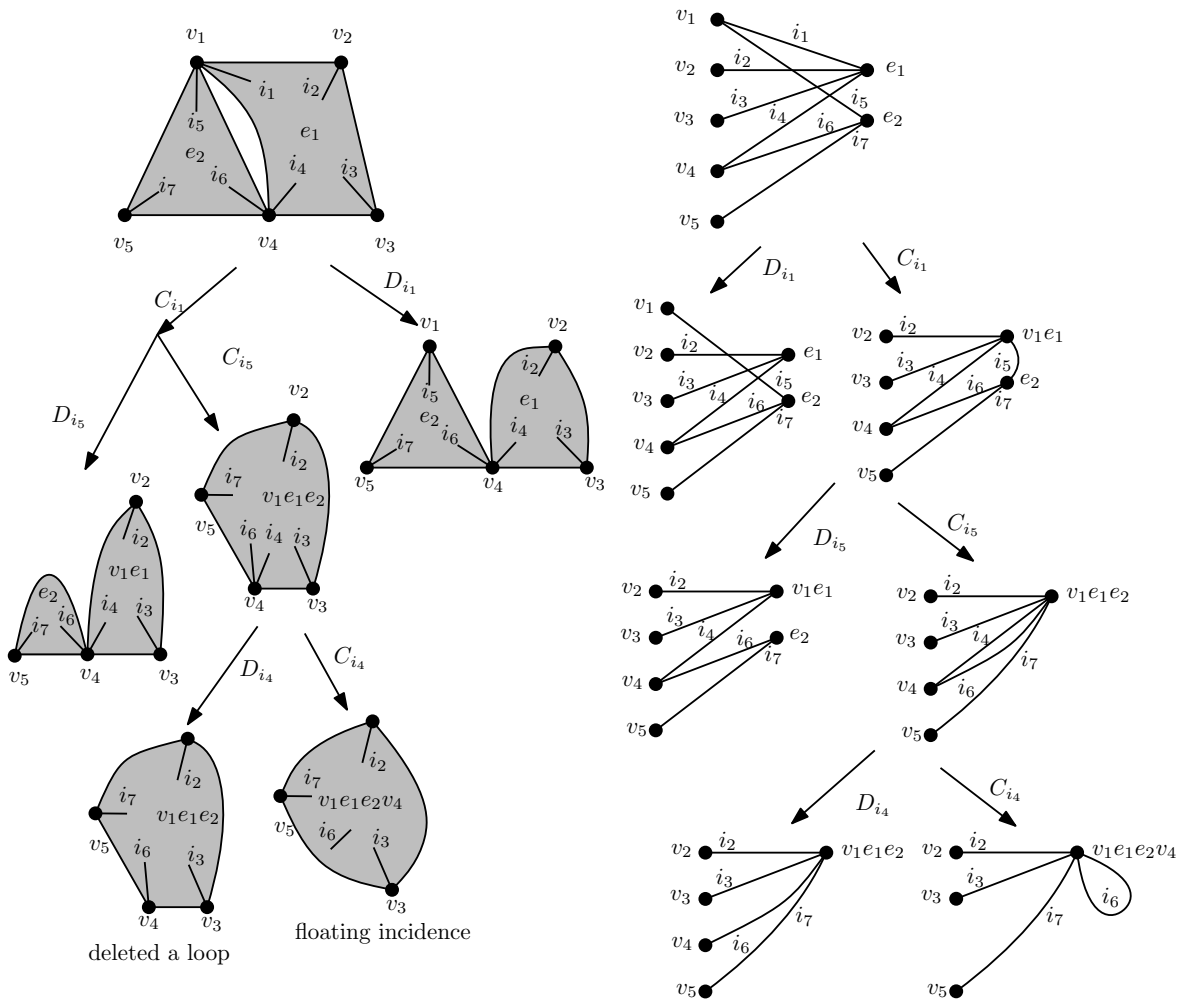


Figure 5.8: Deletion and contraction via the bipartite representation follow vertex order

Another interesting observation is that when we contract in the hypergraph it takes multiple incidence contractions and deletions to obtain another hypergraph, the number seems to correspond to the number of edges incident to the vertex in question. It seems

like there might be some sort of partial contractions. We can see this in figure 5.8 where after contracting i_1 we deleted and contract i_5 which merges v_1 with e_1 in both cases but separates v_1 from the other incident edge after the deletion while merging the other incident edge after the contraction.

To further investigate how we might construct a Tutte polynomial for hypergraphs we need to more fully investigate how deleting and contracting incidences in the bipartite representation corresponding to vertex order might define these partial contractions. We also need to do more examples with loops within larger edges. Another issue is if an edge that is a thorn or a graph with a single k-edge counts as an x or if the size of the edge changes the exponent. Finally it seems likely that a form of cleaving for half edges might provide a way to relate the choice of incidence and hypergraphs obtain in the vertex and edge order of deletion and contraction. After all, the hypergraphs in figure 5.7 and figure 5.8 look related and some sort of cleaving might indicate that the incidence order gives us different hypergraphs, that we could show have the same hypergraphic Tutte polynomial. This would be ideal as we would prefer that edge or incidence order not matter.

5.4 A Weighted Graph Interlace-like Polynomial

In chapter 4 section 4.3, we talked about how to extend the notion of chord diagrams to k-occurrence chord diagrams that relate to Eulerian circuits of $2k$-regular graphs. We observed that we have $(2k-1)!!$ ways to manipulate the chord diagrams in relation to resolving a specific vertex in the $2k$-regular graph. It is the hope that this work lays a foundation to extend the interlace polynomial. We first construct a class of weighted graphs with edge weights modulo k. Given letters a and b we take all chords between two instances of a and all chords between two instances of b and count the number of times a chord from a intersects a chord from b to obtain the number of ab-crossings. Given that the number of crossings in the k-occurrence diagram is an even number between zero and $2k$, it makes sense to take the number of crossings in the diagram, divide by two and then take it modulo k, for $k > 2$, and define this to be the weight of the end between the two vertices in a weighted interlace graph. Figure 5.9 shows the maximal number of crossing for several k-occurrence chord diagrams.

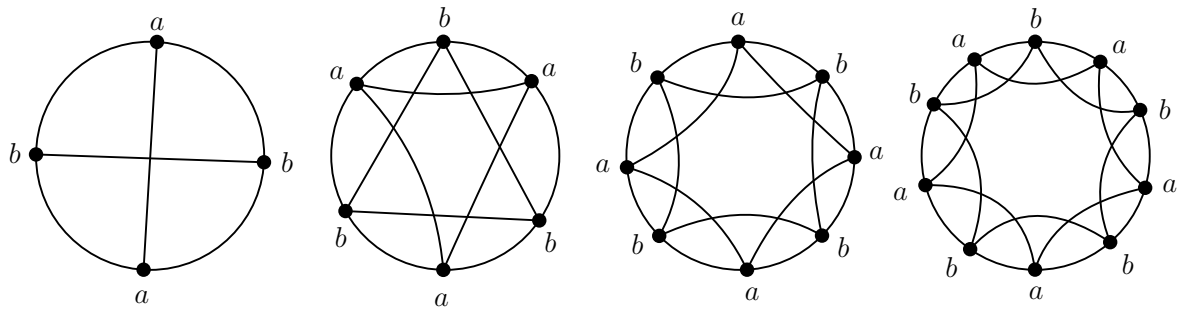


Figure 5.9: Two vertices with a maximal number of chord crossings in k-occurrence diagrams for $k \in \{2, 3, 4, 5\}$

We turn a k-occurrence chord diagram into a weighted interlace graph $I(C)$ by taking the vertex set to be the edge labels ignoring multiplicity and two vertices a and b are adjacent with edge weight l if the total number of ab crossing chords n in the k-occurrence chord diagram divided by two is congruent to $n \mod k$. Figure 5.10 shows how a k-occurrence chord diagram can we converted to a weighted graph. What remains to determine is how the chord diagram operations translate to the weighted graph. When working with small examples many of the operations produce identical results and this is making it difficult to determine how the k-occurrence diagram operations translate to the weighted graph.

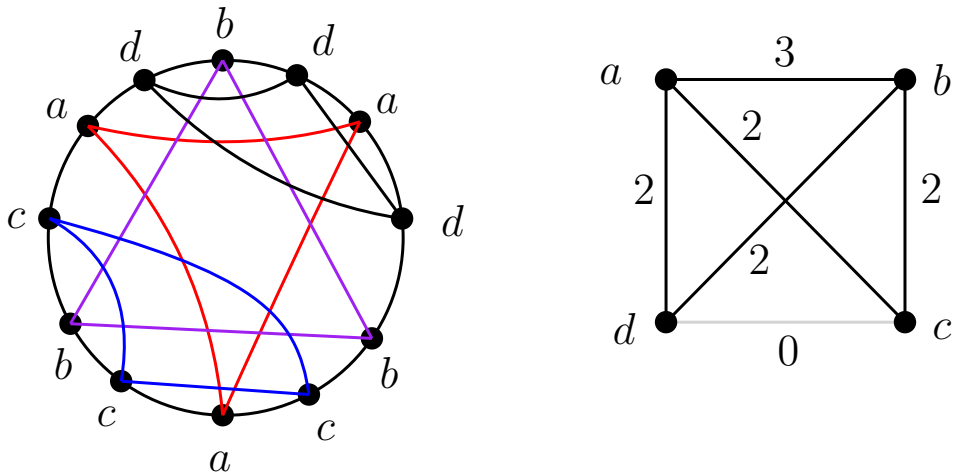


Figure 5.10: A weighted interlace graph from a 3-occurrence chord diagram

Another issue, that will hopefully be resolved by more testing of examples, is that when we perform the break operation on the chord diagrams we then need to glue the

segments together again to construct one single chord diagram and form the weighted interlace graph. This was previously a trivial issue because we had interlaced vertices u and v and could write our word of the chord diagram as $vw_1uw_wvw_3uw_4$ where the only choice of a break gives w_1uw_2 and w_3uw_4. Now both words contained u so we could glue the words by writing $uw_2w_1 \| uw_4w_3 = uw_2w_1uw_4w_3 = u\overleftarrow{w_1}\overleftarrow{w_2}u\overleftarrow{w_3}\overleftarrow{w_4}$. Now that we have more options than interlaced or not it is less clear how gluing should be performed. Since the original Eulerian cycle is connected we know there cannot be a a non-empty word that share no letters with other words, but there may not be a single letter we can use to glue all words back together and there may be many different possible choices of gluing. This makes this operation no longer trivial. In all tested examples, it does not appear to matter what method of gluing is used, but as we don't have a polynomial definition we cannot currently prove the polynomial is equivalent regardless of the choice of gluing. Hopefully upon continued investigation, a better idea of the operations on the k-occurrence chord diagrams and how they affect the weighted interlace graph will give a potential polynomial definition that can be tested against different choices of gluing. Another complications is that the operations appear to vary the weights of the edges, but it can be hard to tell in small examples what the general rule is for whether the weight of an edge is to be increased or decreased. It is hoped that continued larger examples might yield insight into this process.

One final observation we would like to make in this section is about the interesting visual relationship between chord diagrams and interlace graphs. This relationship does not currently appear to extend to our weighted idea of a construction but is still interesting. In the internal structure of the chord diagrams there are various regions bounded by the chords. These regions appear to mimic the structure of the interlace graph by different regions containing cycles in the graph, some of which may overlap. The issue with proving this currently lies in figuring out what we are actually trying to show. It is clear that any bounded region inside the chord diagram corresponds to a cycle in the interlace graph where the chords make up the vertices. It is also clear that crossing chords that do not form a region inside the chord diagram build a path in the interlace graph. These results follow immediately from the definition of when two vertices in the interlace graph are adjacent. Figure 5.11 shows an example of this.

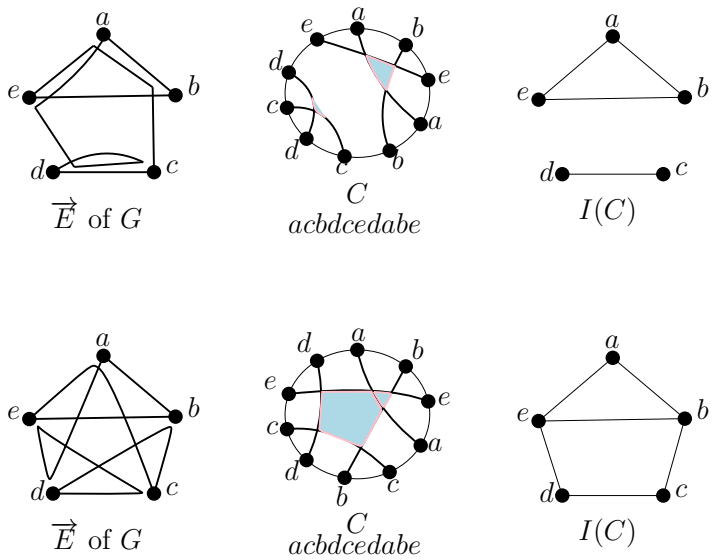


Figure 5.11: Regions inside of the chord diagram that show the shape of the interlace graph

5.5 Conclusion

In conclusion, there has been a great deal of work in the realm of graph polynomials. Much of this work has been dedicated to the extensions or generalizations of known polynomials to new classes of graphs. This can even be seen in the very beginning of graph polynomials where Whitney and Tutte worked on extending and generalizing the chromatic polynomial for planar graphs from that of Birkhoff and Lewis to the Tutte polynomial and Whitney rank generating polynomial, which apply to all graphs. The number of specializations and point evaluations of the Tutte polynomial is truly impressive and many open problems related to the Tutte polynomial remain. Another primary avenue of graph polynomial research is in seeing how various polynomials relate to each other in general or for specific classes of related graphs. We saw this with the Martin polynomial and interlace polynomial where the original interlace polynomial definition was seen to be related to the Martin polynomial of 2-in 2-out graphs, and how these results where quickly extended to an interlace polynomial related to the Martin polynomial of 4-regular graphs. The results and future directions presented here will hopefully allow for the construction of a Tutte polynomial for oriented hypergraphs and an interlace polynomial for k-weighted graphs that relates to the Martin polynomial of $2k$-regular graphs continuing the tradition of generalizing and extending graph polynomials.

References

[1] Alexander A. Ageev. A triangle-free circle graph with chromatic number 5. *Discret. Math.*, 152(1-3):295–298, 1996.

[2] Dorit Aharonov, Itai Arad, Elad Eban, and Zeph Landau. Polynomial quantum algorithms for additive approximations of the Potts model and other points of the Tutte plane, 2007.

[3] Martin Aigner and Hans Mielke. The Penrose polynomial of binary matroids. *Monatshefte für Mathematik*, 131:1–13, 2000.

[4] Martin Aigner and Hein Van Der Holst. Interlace polynomials. *Linear Algebra and its Applications*, 377:11–30, 2004.

[5] Gerald Alexanderson. About the cover: Euler and the Königsberg bridges: A historical view. *Bulletin (New Series) of the American Mathematical Society*, 43:567–573, 10 2006.

[6] Paolo Aluffi and Matilde Marcolli. A motivic approach to phase transitions in Potts models. *Journal of Geometry and Physics*, 63:6–31, January 2013.

[7] Richard Arratia, Bela Bollobas, and Gregory B. Sorkin. The interlace polynomial of a graph. *Journal of Combinatorial Theory, Series B*, 92(2):199–233, August 2004.

[8] Richard Arratia, Bela Bollobas, and Gregory B. Sorkin. A two-variable interlace polynomial. *Combinatorica*, 2004.

[9] Ravindra B. Bapat. *Graphs and Matrices*. Universitext. Springer London, 2 edition, 2014.

[10] Zsuzsanna Baran, Jonathan Hermon, Andela Šarković, and Perla Sousi. Phase transition for random walks on graphs with added weighted random matching, 2023.

[11] Matthias Beck, Erika Meza, Bryan Nevarez, Alana Shine, and Michael Young. The chromatic polynomials of signed Petersen graphs. *Involve, a Journal of Mathematics*, 8(5):825–831, September 2015.

[12] Philippe Biane. Polynomials associated with finite Markov chains, 2014. arXiv 1412.8682.

[13] Norman L. Biggs and Peter Winkler. Chip-firing and the chromatic polynomail. *CDAM Research Report Series*, LSE-CDAM-97-03, 1997.

[14] George D. Birkhoff. A determinant formula for the number of ways of coloring a map. *Annals of Mathematics*, 14(1/4):42–46, 1912.

[15] George D. Birkhoff and Clive Lewis. Chromatic polynomials. *Trans. Amer. Math. Soc.*, 60:355–451, 1946.

[16] Béla Bollobás. *Modern Graph Theory*, volume 184 of *Graduate Texts in Mathematics*. Springer, New York, NY, 1998.

[17] Béla Bollobás. Evaluations of the circuit partition polynomial. *Journal of Combinatorial Theory, Series B*, 85(2):261–268, July 2002.

[18] André Bouchet. Unimodularity and circle graphs. *Discrete Mathematics*, 66(1):203–208, August 1987.

[19] André Bouchet. Graphic presentations of isotropic systems. *Journal of Combinatorial Theory, Series B*, 45(1):58–76, August 1988.

[20] André Bouchet. κ-transformations, local complementations and switching. In Geňa Hahn, Gert Sabidussi, and Robert E. Woodrow, editors, *Cycles and Rays*, pages 41–50. Springer Netherlands, Dordrecht, 1990.

[21] André Bouchet. Circle graph obstructions. *Journal of Combinatorial Theory, Series B*, 60(1):107–144, January 1994.

[22] André Bouchet. Bipartite graphs that are not circle graphs. *Annales de l'institut Fourier*, 49(3):809–814, 1999.

[23] André Bouchet. Graph polynomials derived from Tutte-Martin polynomials. *Discrete Mathematics*, 302(1-3):32–38, 2005.

[24] André Bouchet and Laurence Ghier. Connectivity and β-invariants of isotropic systems and 4-regular graphs. *Discrete Mathematics*, 161(1):25–44, December 1996.

[25] Robert Brijder and Hendrik Jan Hoogeboom. Interlace polynomials for multimatroids and delta-matroids. *European Journal of Combinatorics*, 40:142–167, 2014.

[26] Robert Brijder and Hendrik Jan Hoogeboom. Quaternary bicycle matroids and the penrose polynomial for delta-matroids, 2014. arXiv 1210.7718.

[27] Francis C. S. Brown. On the periods of some Feynman integrals, 2010. arXiv 0910.0114.

[28] Joshua Brown, Terry Bossomaier, and Lionel Barnett. Information flow in first-order potts model phase transition. *Scientific Reports*, 12, 2022.

[29] Manoj Chari and Charles J Colbourn. Reliability polynomials: A survey. *Combinatorica*, 1998.

[30] Sergei Chmutov and Igor Pak. The Kauffman bracket of virtual links and the Bollobás-Riordan polynomial. *Mosc. Math. J.*, 2007.

[31] Sergei Chmutov and Jeremy Voltz. Thistlethwaite's theorem for virtual links. *Journal of Knot Theory and Its Ramifications*, 17(10):1189–1198, 2008.

[32] Hojin Choi, O.-joung Kwon, Sang-il Oum, and Paul Wollan. χ-boundedness of graph classes excluding wheel vertex-minors. *Journal of Combinatorial Theory, Series B*, 135:319–348, March 2019.

[33] Ilkyoo Choi, O.-joung Kwon, and Sang-il Oum. Coloring graphs without fan vertex-minors and graphs without cycle pivot-minors. *Journal of Combinatorial Theory, Series B*, 123:126–147, March 2017.

[34] F. Chung and C. Yang. On polynomials of spanning trees. *Annals of Combinatorics*, 4(1):13–25, 2000.

[35] J.H. Conway. An enumeration of knots and links, and some of their algebraic properties. In John Leech, editor, *Computational Problems in Abstract Algebra*, pages 329–358. Pergamon, 1970.

[36] Bruno Courcelle. A multivariate interlace polynomial and its computation for graphs of bounded clique-width. *The Electronic Journal of Combinatorics*, 15:R69, May 2008.

[37] Oliver T. Dasbach, David Futer, Efstratia Kalfagianni, Xiao-Song Lin, and Neal W. Stoltzfus. The Jones polynomial and graphs on surfaces. *Journal of Combinatorial Theory, Series B*, 98(2):384–399, 2008.

[38] James Davies. Improved bounds for colouring circle graphs. *Proceedings of the American Mathematical Society*, 150(12):5121–5135, 2022.

[39] James Davies. *Local properties of graphs with large chromatic number*. Doctoral Thesis, University of Waterloo, August 2022.

[40] James Davies. Vertex-minor-closed classes are χ-bounded. *Combinatorica*, 42(1):1049–1079, 2022.

[41] James Davies and Rose McCarty. Circle graphs are quadratically χ-bounded. *Bulletin of the London Mathematical Society*, 53(3):673–679, June 2021.

[42] Hubert de Fraysseix. Local complementation and interlacement graphs. *Discrete Mathematics*, 33(1):29–35, January 1981.

[43] Hubert de Fraysseix. A characterization of circle graphs. *European Journal of Combinatorics*, 5(3):223–238, September 1984.

[44] Y. Diao, G. Hetyei, and K. Hinson. Tutte polynomials of tensor products of signed graphs and their applications in knot theory. *Journal of Knot Theory and Its Ramifications*, 18(05):561–589, 2009.

[45] Joanna A. Ellis-Monaghan. New results for the Martin polynomial. *Journal of Combinatorial Theory, Series B*, 74(2):326–352, November 1998.

[46] Joanna A. Ellis-Monaghan and Iain Moffatt. A Penrose polynomial for embedded graphs. *European Journal of Combinatorics*, 34(2):424–445, 2013.

[47] Joanna A. Ellis-Monaghan and Iain Moffatt, editors. *Handbook of the Tutte Polynomial and Related Topics*. Chapman and Hall/CRC, New York, July 2022.

[48] Joanna A. Ellis-Monaghan and Irasema Sarmiento. Generalized transition polynomials. *Congressus Numerantium*, 155, 2002.

[49] Damir Filipović, Martin Larsson, and Sergio Pulido. Markov cubature rules for polynomial processes, 2019.

[50] Loïc Foissy. Chromatic polynomials and bialgebras of graphs. *International Electronic Journal of Algebra*, 30:116–167, 2021.

[51] E. Gasse. A proof of a circle graph characterization. *Discrete Mathematics*, 173(1–3):277–283, 1997.

[52] F. Gavril. Algorithms for a maximum clique and a maximum independent set of a circle graph. *Networks. An International Journal*, 3(3):261–273, January 1973.

[53] Jim Geelen and Edward Lee. Naji's characterization of circle graphs. *Journal of Graph Theory*, 93(1):21–33, January 2020.

[54] Jim Geelen and Sang-il Oum. Circle graph obstructions under pivoting. *Journal of Graph Theory*, 61:1–11, 2009.

[55] Joseph Geraci and Daniel A. Lidar. On the exact evaluation of certain instances of the Potts partition function by quantum computers. *Communications in Mathematical Physics*, 279(3):735–768, March 2008.

[56] Chris Godsil and Gordon Royle. *Algebraic graph theory*, volume 207 of Graduate Texts in Mathematics. Springer-Verlag, 2001.

[57] Andrew Goodall, Bart Litjens, Guus Regts, and Lluís Vena. Tutte's dichromate for signed graphs. *Discrete Applied Mathematics*, 289:153–184, January 2021.

[58] Rohini Gore, Tejal Gore, Namrata Rokade, and Yogesh Mandlik. Applications of graph theory. *International Journal of Latest Technology in Engineering Management and Applied Science*, 14:148–150, 2025.

[59] Kenneth Grahame. *The Wind in the Willows*. Methuen, London, 1908.

[60] Curtis Greene and Thomas Zaslavsky. On the interpretation of Whitney numbers through arrangements of hyperplanes, zonotopes, non-radon partitions, and orientations of graphs. *Transactions of The American Mathematical Society*, 280:97, 1983.

[61] Will Grilliette, Josephine Reynes, and Lucas J. Rusnak. Incidence hypergraphs: Injectivity, uniformity, and matrix-tree theorems. *Linear Algebra and its Applications*, 634:77–105, February 2022.

[62] P. L. Hansen. Labelling algorithms for balance in signed graphs. In *Problèmes Combinatoires et Théorie des Graphes (Colloq. Internat., Orsay, 1976)*, volume 260 of *Colloques Internat. du CNRS*, pages 215–217. Éditions du C.N.R.S., Paris, 1978. MR80m:68057, Zbl 413.05060.

[63] Frank Harary. On the notion of balance of a signed graph. *Michigan Math. J.*, 2(2):143–146, 1953.

[64] Frank Harary. On the measurement of structural balance. *Behavioral Sci.*, 4:316–323, 1959.

[65] Frank Harary and Jerald A. Kabell. A simple algorithm to detect balance in signed graphs. *Mathematical Social Sciences*, 1(1):131–136, 1980.

[66] Roslan Hasni and Ainul Maulid Ahmad. On classification of graphs: By chromatic polynomials. *Applied Mathematical Sciences (Ruse)*, 6, 01 2012.

[67] Robert Hickingbotham, Freddie Illingworth, Bojan Mohar, and David R. Wood. Treewidth, circle graphs and circular drawings. *SIAM Journal on Discrete Mathematics*, 38(1):965–987, 2024.

[68] Simone Hu, Oliver Schnetz, Jim Shaw, and Karen Yeats. Further investigations into the graph theory of ϕ^4-periods and the c_2 invariant. *Ann. Inst. H. Poincare D Comb. Phys. Interact.*, 9(3):473–524, 2022.

[69] Jake Huryn and Sergei Chmutov. A few more trees the chromatic symmetric function can distinguish. *Involve, a Journal of Mathematics*, 13(1):109–116, February 2020.

[70] F. Jaeger. On transition polynomials of 4-regular graphs. In Geňa Hahn, Gert Sabidussi, and Robert E. Woodrow, editors, *Cycles and Rays*, pages 123–150. Springer Netherlands, Dordrecht, 1990.

[71] François Jaeger. On Tutte polynomials and cycles of plane graphs. *Journal of Combinatorial Theory, Series B*, 44(2):127–146, 1988.

[72] V. F. R. Jones. A polynomial invariant for knots via von Neumann algebras. *Bull. Am. Math. Soc.*, 12:103–111, 1985.

[73] Louis H. Kauffman. State models and the Jones polynomial. *Topology*, 26(3):395–407, 1987.

[74] Louis H. Kauffman. A Tutte polynomial for signed graphs. *Discrete Applied Mathematics*, 25(1-2):105–127, October 1989.

[75] Abdullah Khan, Alexei Lisitsa, Viktor Lopatkin, and Alexei Vernitski. Circle graphs (chord interlacement graphs) of Gauss diagrams: Descriptions of realizable Gauss diagrams, algorithms, enumeration, 2021. arXiv 2108.02873.

[76] Ringi Kim, O-joung Kwon, Sang-il Oum, and Vaidy Sivaraman. Classes of graphs with no long cycle as a vertex-minor are polynomially χ-bounded. *Journal of Combinatorial Theory, Series B*, 140:372–386, January 2020.

[77] G. Kirchhoff. Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird. *Annalen der Physik*, 148(12):497–508, January 1847.

[78] H Kleinert and V Schulte-Frohlinde. *Critical Properties of Φ^4-Theories*. World Scientific, 2001.

[79] K.M. Koh and K.L. Teo. The search for chromatically unique graphs — II. *Discrete Mathematics*, 172(1):59–78, 1997.

[80] Vadim Kostrykin and Robert Schrader. Generating functions of random walks on graphs, 2004.

[81] B. D. Kotlyar. Necessary condition for a chromatic polynomial. *Cybernetics and Systems Analysis*, 34(5):779–780, 1998.

[82] T. Krajewski, V. Rivasseau, A. Tanasa, and Zhituo Wang. Topological grap polynomials and quantum field theory, part I: Heat kernel theories, November 2008.

[83] C. Krattenthaler. Lattice path enumeration, 2017. arXiv 1503.05930.

[84] J. Satish Kumar. Applications of graph theory in optimization and network analysis (1st edition). *Journal of Computational Analysis and Applications*, 33(8):2546–2556, 2025.

[85] Michel Las Vergnas. On Eulerian partitions of graphs, in: Graph theory and combinatorics (Proc. Conf., Open Univ., Milton Keynes, 1978). In *Res. Notes in Math*, volume 34, pages 62–75, Pitman, Boston, Mass.-London, 1979.

[86] Michel Las Vergnas. Le polynôme de Martin d'un graphe Eulerien. In C. Berge, D. Bresson, P. Camion, J. F. Maurras, and F. Sterboul, editors, *Combinatorial Mathematics*, volume 75 of North-Holland Mathematics Studies, pages 397–411. North-Holland, 1983-01-01.

[87] Michel Las Vergnas. On the evaluation at (3, 3) of the Tutte polynomial of a graph. *Journal of Combinatorial Theory, Series B*, 45(3):367–372, December 1988.

[88] Edward Lee. *Circle Graph Obstructions*. PhD thesis, University of Waterloo, 2017.

[89] Elliott H. Lieb. Residual entropy of square ice. *Phys. Rev.*, 162:162–172, Oct 1967.

[90] Criel Merino López. Chip firing and the Tutte polynomial. *Annals of Combinatorics*, 1(1):253–259, 1997.

[91] Pierre Martin. *Enumérations eulériennes dans les multigraphes et invariants de Tutte-Grothendieck*. thesis, Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, March 1977.

[92] Ada Morse. The interlace polynomial. *Graph Polynomials*, 2016. arXiv 1601.03003.

[93] Kunio Murasugi. On invariants of graphs with applications to knot theory. *Transactions of the American Mathematical Society*, 314(1):1–49, 1989.

[94] G. V. Nenashev. An upper bound on the chromatic number of circle graphs without k_4. *Journal of Mathematical Sciences*, 184(5):629–633, August 2012.

[95] James G. Oxley. *Matroid theory*. Oxford Science Publications. The Clarendon Press Oxford University Press, New York, 1992.

[96] Erik Panzer and Karen Yeats. Feynman symmetries of the Martin and c_2 invariants of regular graphs. *Combin. Theor.*, 5:2025, 2023.

[97] Linus Pauling. The structure and entropy of ice and of other crystals with some randomness of atomic arrangement. *Journal of the American Chemical Society*, 57(12):2680–2684, 1935.

[98] R. Penrose. Applications of negative dimensional tensors. In *Combinatorial Mathematics and Its Applications (Proc. Conf., Oxford, 1969)*, pages 221–244. Academic Press, London, 1971.

[99] N. Reff and Lucas J Rusnak. An oriented hypergraphic approach to algebraic graph theory. *Linear Algebra and its Applications*, 437(9):2262–2270, 2012.

[100] P. Rosenstiehl and R.C. Read. On the principal edge tripartition of a graph. In B. Bollobás, editor, *Advances in Graph Theory*, volume 3 of Annals of Discrete Mathematics, pages 195–226. Elsevier, 1978.

[101] Lucas J Rusnak. Oriented hypergraphs: Introduction and balance. *Electronic J. Combinatorics*, 20(3), 2013.

[102] Lucas J Rusnak, Josephine Reynes, Skyler J Johnson, and Peter Ye. Generalizing Kirchhoff laws for signed graphs. *The Australasian Journal of Combinatorics*, 2021.

[103] Radmila Sazdanovic and Daniel Scofield. Structure of the chromatic polynomial, 2024. arXiv 2411.15088.

[104] Oliver Schnetz. Quantum periods: A Census of ϕ^4-transcendentals. *Commun. Num. Theor. Phys.*, 4:1–48, 2010.

[105] Christopher-Lloyd Simon. Loops in surfaces, chord diagrams, interlace graphs: operad factorisations and generating grammars, 2024. arXiv 2310.08806.

[106] Neil J. A. Sloane and The OEIS Foundation Inc. The online encyclopedia of integer sequences, 2020.

[107] Alan D. Sokal. The multivariate tutte polynomial (alias Potts model) for graphs and matroids. In Bridget S.Editor Webb, editor, *Surveys in Combinatorics 2005*, London Mathematical Society Lecture Note Series, page 173–226. Cambridge University Press, July 2005.

[108] Richard P. Stanley. Decompositions of rational convex polytopes. In J. Srivastava, editor, *Combinatorial Mathematics, Optimal Designs and Their Applications*, volume 6 of *Annals of Discrete Mathematics*, pages 333–342. Elsevier, 1980.

[109] Hongpeng Sun, Xuecheng Tai, and Jing Yuan. Efficient admm and splitting methods for continuous min-cut and max-flow problems. *SIAM Journal on Scientific Computing*, 43(2):B455–B478, 2021.

[110] J.J. Sylvester. On the partition of numbers. *Q. J. Math*, 1:141–152, 1857.

[111] P. G. Tait. Listing's topologie. *Philosophical Magazine*, 17:30–46, 1884.

[112] Lorenzo Traldi. A dichromatic polynomial for weighted graphs and link polynomials. *Proceedings of the American Mathematical Society*, 106(1):279–286, 1989.

[113] Lorenzo Traldi. On the interlace polynomials. *Journal of Combinatorial Theory, Series B*, 103(1):184–208, January 2013.

[114] Lorenzo Traldi. Interlacement in 4-regular graphs: a new approach using nonsymmetric matrices. *Contributions to Discrete Mathematics*, 9:85–97, 2014.

[115] Lorenzo Traldi. The transition matroid of a 4-regular graph: An introduction. *European Journal of Combinatorics*, 50:180–207, November 2015.

[116] Nicolas Trotignon and Lan Anh Pham. χ-bounds, operations, and chords. *Journal of Graph Theory*, 88(2):312–336, November 2017.

[117] W. T. Tutte. On hamiltonian circuits. *Journal of the London Mathematical Society*, 21(2):98–101, 1946.

[118] W. T. Tutte. A ring in graph theory. *Mathematical Proceedings of the Cambridge Philosophical Society*, 43(1):26–40, January 1947.

[119] W. T. Tutte. *An algebraic theory of graphs*. PhD thesis, University of Cambridge, 1949.

[120] W. T. Tutte. A contribution to the theory of chromatic polynomials. *Canadian Journal of Mathematics*, 6:80–91, January 1954.

[121] W. T. Tutte. On dichromatic polynomials. *Journal of Combinatorial Theory*, 2(3):301–320, May 1967.

[122] W. T. Tutte. On the Birkhoff-Lewis equations. *Discrete Mathematics*, 92(1):417–425, 1991.

[123] W. T. Tutte. The Birkhoff-Lewis Equations for graph-colorings. In John Gimbel, John W. Kennedy, and Louis V. Quintas, editors, *Quo Vadis, Graph Theory?*, volume 55 of Annals of Discrete Mathematics, pages 153–158. Elsevier, 1993.

[124] Carlos E. Valencia and Marcos I. Barrita. 1-join composition for α-critical graphs, 2007. arXiv 0707.4085.

[125] Lothar Von Collatz and Ulrich Sinogowitz. Spektren endlicher grafen. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 21(1):63–77, 1957.

[126] Stefan Weinzierl. Periods and hodge structures in perturbative quantum field theory. In Luis Álvarez Cónsul, José Ignacio Burgos-Gil, and Kurusch Ebrahimi-Fard, editors, *Feynman Amplitudes, Periods and Motives*, volume 648 of Contemporary Mathematics. American Mathematical Society, 2015.

[127] Hassler Whitney. The coloring of graphs. *Annals of Mathematics*, 33(4):688–718, 1932.

[128] Hassler Whitney. A logical expansion in mathematics. *Bulletin of the American Mathematical Society*, 38(8):572–579, August 1932.

[129] Hassler Whitney. 2-Isomorphic graphs. *American Journal of Mathematics*, 55(1):245–254, 1933.

[130] Hassler Whitney. A set of topological invariants for graphs. *American Journal of Mathematics*, 55(1):231–235, 1933.

[131] F.Y Wu and J Wang. Zeroes of the Jones polynomial. *Physica A: Statistical Mechanics and its Applications*, 296(3):483–494, 2001.

[132] Ioannis S. Xezonakis and Danai Xezonaki. An algorithm for testing a signed graph for balance. *International Journal of Computer Applications*, 184(11):41–44, May 2022.

[133] Karen Yeats. *A Combinatorial Perspective on Quantum Field Theory*, volume 15 of SpringerBriefs in Mathematical Physics. Springer, 2017.

[134] T. Zaslavsky. Biased graphs .iii. chromatic and dichromatic invariants. *Journal of Combinatorial Theory, Series B*, 64(1):17–88, 1995.

[135] Thomas Zaslavsky. Characterizations of signed graphs. *Journal of Graph Theory*, 5(4):401–406, 1981.

[136] Thomas Zaslavsky. Signed graphs. *Discrete Applied Mathematics*, 4(1):47–74, 1982.

[137] Thomas Zaslavsky. Biased graphs. ii. the three matroids. *Journal of Combinatorial Theory, Series B*, 51(1):46–72, 1991.

[138] Thomas Zaslavsky. Orientation of signed graphs. *European Journal of Combinatorics*, 12(4):361–375, 1991.

[139] Thomas Zaslavsky. Strong tutte functions of matroids and graphs. *Transactions of the American Mathematical Society*, 334(1):317–347, 1992.