TECHNISCHE UNIVERSITÄT WIEN

DIPLOMARBEIT

Matrix Tree Theorems

zur Erlangung des akademischen Grades
Diplom-Ingenieur

im Rahmen des Studiums
Technische Mathematik

unter der Anleitung von
Ao. Univ. Prof. Dipl.-Ing. Dr. techn. Bernhard Gittenberger

eingereicht von
Fabian Holzer BSc
Matrikelnummer 01226792

ausgeführt am Institut für Diskrete Mathematik und Geometrie
der Fakultät für Mathematik und Geoinformation
der Technischen Universität Wien

Wien, am 28. März 2022

Unterschrift Verfasser

Unterschrift Betreuer

Affidavit

I declare in lieu of oath that I wrote this thesis and performed the associated research myself, using only literature cited in this volume. If text passages from sources are used literally, they are marked as such. I confirm that this work is original and has not been submitted elsewhere for any examination, nor is it currently under consideration for a thesis elsewhere.

Vienna, March 28, 2022

Signature Author

Abstract

The goal of this thesis is to provide a deeper insight into the inner workings of the matrix tree theorem. For this it introduces G. R. Kirchhoff's matrix tree theorem and shows different approaches to prove it: the nowadays canonical proof via the Cauchy–Binet formula, G. R. Kirchhoff's original approach used in 1847, one employing the inclusion–exclusion principle, a similar one applying a sign-reversing involution to remove subgraphs containing cycles, another using deletion–contraction via induction, and a probabilistic proof via random walks. The chapter closes with mentions of similar theorems, including the matrix tree theorem on edges, the matrix tree theorem using the signless Laplacian, and the Markov chain tree theorem. The previously presented methods are then applied to prove the matrix tree theorem on semirings and for hypergraphs. To formulate these generalisations, the symmetric extension of semirings, bideterminants, arboreal hypergraphs, hypertrees, and the Pfaffian are defined.

Das Ziel dieser Masterarbeit ist es, einen tieferen Einblick in verschiedene Aspekte des Matrix-Baum-Theorems zu vermitteln. Es wird eine Einführung in G. R. Kirchhoffs Matrix-Baum-Theorem geboten, wofür verschiedene Beweise präsentiert werden: der heute kanonische Beweis via die Cauchy-Binet Formel, G. R. Kirchhoffs ursprünglicher Zugang aus 1847, ein weiterer via das Inklusion-Exklusion Prinzip, ein sehr ähnlicher Beweis mit einer sogenannten Vorzeichen-umkehrenden Involution, um Subgraphen mit Zyklen zu entfernen, eine weitere Methode via Löschen und Zusammenziehen von Kanten, und ein probabilistischer Beweis mit Irrfahrten. Das Kapitel schließt mit Erwähnungen weiterer ähnlicher Sätze, unter anderem dem Matrix-Baum-Theorem auf Kanten, dem Matrix-Baum-Theorem mit der vorzeichenlosen Laplace'schen Matrix, und dem Markov-Ketten-Baum-Satz. Die zuvor vorgestellten Methoden werden dann benutzt, um das Matrix-Baum-Theorem auf Semiringen und für Hypergraphen zu beweisen. Um diese Verallgemeinerungen zu beschreiben, werden die symmetrische Erweiterung, Bideterminanten, arboreale Hypergraphen, Hyperbäume und die Pfaff'sche Determinante definiert.

Acknowledgements

First and foremost, I want to thank my supervisor, Professor Bernhard Gittenberger, who provided me with the topic of this thesis. He guided my research in the right direction, but also knew when to stop me as "this should only be a master's thesis and not a monograph."

I also want to thank my proofreaders, who thoroughly combed through the text and found many minor errors. They also provided useful comments, which increased the overall quality of this thesis.

Finally, I want to thank friends and family for their support during my studies.

Contents

1 Getting There

This first chapter is concerned with building the foundation required for the matrix tree theorem. It consists of a short historical introduction, basic definitions and some specialised notation. We continue with a short proof employing the Cauchy-Binet formula, which is also proven. Finally, Cayley's theorem and Scoins' theorem are derived from the matrix tree theorem as corollaries.

1.1 Introduction

The matrix tree theorem is a theorem within the field of graph theory. Its first formulation by G. R. Kirchhoff in 1847 is concerned with galvanic currents—nowadays called direct current or DC for short—in electrical circuits [Kir47]. He notes a way to count the combinations without closed shapes (*geschlossene Figuren*) from his two current and voltage laws, which he formulated in 1845 [Kir45]. His electrical circuits will later form the basis for graphs, on which the trees—which are the combinations without closed shapes—will be counted. We will take a closer look at this approach in Section 2.1. Interestingly enough, this first formulation predates both the notion of matrices and of trees.[1]

Work on the theorem did not stop and so W. Ahrens separated the primal form of the theorem from its real-world connection in electrical circuits and gave a "purely mathematical approach" [Ahr97, p. 21], while in the meantime C. W. Borchardt and A. Cayley both independently proved a special case nowadays known as Cayley's theorem. C. W. Borchardt was more interested in a mathematical formulation and A. Cayley himself tried to find a way to count the number of combinations of paraffin wax C_nH_{2n+2} [Cay75, p. 257].[2] Many years later—in 1962—H. I. Scoins proved another special case [Sco62]. In the meantime a generalisation was found, namely Tutte's theorem. It operates on directed graphs, in which every connection has an intrinsic direction. This is in contrast to the electrical circuits used by G. R. Kirchhoff, which carry current in any direction.

An early proof of the matrix tree theorem via the Cauchy–Binet formula can be found in [Tre54]. We will follow this approach in this chapter. In 1970 J. W. Moon dedicated an entire section to the matrix tree theorem and also gave a short synopsis on the then recent developments [Moo70, Section 5]. We mark this as the border between the history of the matrix tree theorem and modern developments. The different proof methods discussed in Chapter 2 were all discovered after 1970—apart from the historical approach, which was only formalised later.

The first chapter is concerned about building the necessary foundation for the matrix tree theorem, which states that the number of spanning trees in a given graph is the determinant of a minor of the Laplacian matrix. Note that we are interested in the number of spanning trees and not the number of different types. For example, the graph in Figure 1.1

[1] The term *matrix* was coined by J. J. Sylvester in 1850 [see Syl50] and *tree* was first used in a mathematical context by A. Cayley in 1857 [see Cay57].

[2] This is often misattributed to [Cay57], but A. Cayley himself wrote:"In the paper of 1857, which contains no application to chemical theory, the number of branches from a knot was unlimited" [Cay75, p. 258]. Since this number models the chemical combinations, it had to be limited to better represent the properties of the elements. Cayley's theorem is therefore only a stepping stone.

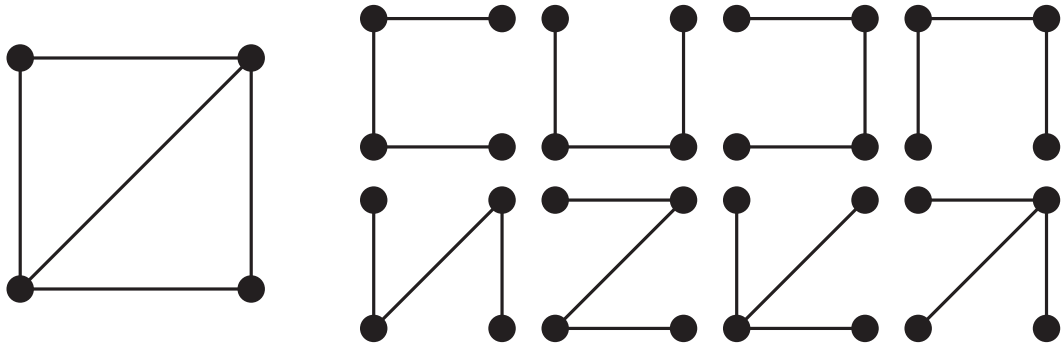


Figure 1.1: The diamond graph and all its eight spanning trees.

contains eight and not two spanning trees. This can be formalised by considering only labelled graphs, in which vertices can be distinguished from each other. The chapter concludes by proving the theorem via the "simple proof […] based upon the theorem of multiplication of determinantal arrays which was discovered independently by Binet and Cauchy" [Tre54, p. 1005] and giving some examples and applications.

The second chapter gives different approaches to only minor variations of the matrix tree theorem. Most proofs in this chapter require minimal additional notation and can be proven with regular methods of graph theory, linear algebra and the like. The main goal is to present these approaches and give deeper insight in the structure and distribution of trees within a graph.

The first of those approaches is the historical approach used by G. R. Kirchhoff, which results in the original formulation, which is rarely used today. We show that it can be used to formulate the matrix tree heorem in its modern form and the cycle theorem, which also gives the number of spanning trees, albeit via a completely different approach.

The second method uses an involution, which removes subgraphs containing cycles. This removal results in keeping all trees and only trees. The natural generalisation of this approach is to directed graphs, for which the theorem was first proven by W. T. Tutte. We use two different variants, where one considers sets of functions and the other working on the trees themselves.

Another way to prove the matrix tree theorem is via an induction on the structure of the graph and merging vertices. Using this approach thus allows one to have several edges between two vertices. Graphs with more than one edge between vertices are called multigraphs. While this approach was first employed by H. Hutschenreuther, he did not consider multigraphs. These were only applied later by M. Lewin.

The final approach originates from the goal of uniformly generating random spanning trees. Thus, it employs much notation from probability theory. Instead of considering the set of all spanning trees, one generates a single one via random walks. Its probability is then proportional to the number of spanning trees.

Many more special cases, variations and generalisations of the matrix tree theorem are known. Some of these are discussed in Section 2.5 and Chapter 3. The generalisations require a more in-depth introduction to their respective fields as they use highly specialised notation. The first one enables us to apply the matrix tree theorem to edge-weighted graphs, where the graphs are in a commutative semiring. The second one generalises graphs to hypergraphs, in which an edge may contain an arbitrary amount of vertices. However, both use previously mentioned proof strategies. Thus, they can be considered applications of Chapter 2.

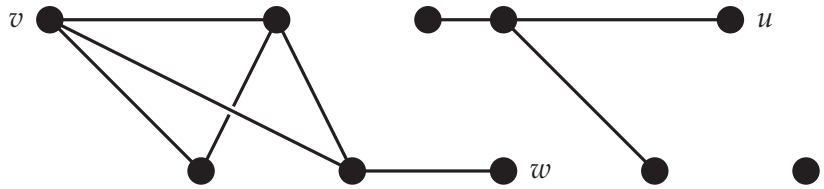


Figure 1.2: A graph with three connected components. Vertices v and w are connected and therefore in the same connected component, u is not. The connected component containing u is also a tree.

1.2 Preliminaries

We will now recall some basic definitions of graph theory, of which the first one is the definition of a graph itself: We distinguish between a directed graph and an undirected graph. A *directed graph* G is a pair (V, E) consisting of the finite set $V := \{v_1, \dots, v_n\}$ of *vertices* and the set $E := \{e_1, \dots, e_m\}$ of *edges*, where $e_i := (v_j, v_k)$. As a shortcut we will often write $\{1, \dots, n\} =: [n]$ for the set of vertices instead. An *undirected graph* is a graph in which the edges are not considered pairs, but sets $e_i := \{v_j, v_k\}$ instead. So they do not contain an intrinsic direction. We will call them digraphs and graphs respectively. A typical identification of undirected graphs is as directed graphs, in which every edge $(v_i, v_j) \in E$ has an inverse edge $(v_j, v_i) \in E$. We will often use this identification implicitly and write (v_i, v_j) for undirected graphs also.

Vertices joined by an edge are called *adjacent* and are each others *neighbours*. The numbers of neighbours of a given vertex v is called the *degree* $\deg(v)$. A *walk* in a graph G is a finite sequence $e_1 e_2 \dots e_k$ of edges, where two subsequent edges share one vertex. A walk that starts on the same vertex it ends on is called a *closed*. So a walk may have repeated vertices and edges. If only vertices are allowed to be repeated, the walk is called a *trail*; if neither are repeated, the walk is called a *path*. Closed trails and closed paths are called *circuits* and *cycles*, respectively. Two vertices are called *connected* iff there exists a path between them. *Connected components* of a graph are the maximal subsets of vertices, which are connected with each other. A *tree* is a connected acyclic graph, a tree is *spanning* iff it is a subgraph of G connecting all vertices.

Example 1.2.1. An example for a graph with three connected components is given in Figure 1.2. It consists of ten vertices, of which three are labelled with u, v and w, and nine edges. The vertices v and w are connected via an unlabelled vertex, and thus must appear in the same connected component. The connected component containing u is a tree, but not a spanning tree as it does not reach all vertices of the graph. In fact, the graph cannot contain a spanning tree as it is not connected. The connected component on the lower right containing only a single vertex and no edges is also considered a tree. ◊

Definition 1.2.2. Let G be a graph, then the *number of all spanning trees* is $\kappa(G)$.

We will heavily employ matrix notation for graphs, so we define some special matrices upfront: The *zero matrix*, denoted by $\mathbb{0}$, contains 0 in every entry, while the *one matrix* $\mathbb{1}$ contains 1 in every entry. We will also often use the *identity Id*, which contains 1 in its diagonal elements and 0 otherwise.

The following definition provides three matrices, which arise naturally working with graphs. While sets of vertices and edges suffice to define graphs, they are often not enough for calculations on graphs.

Definition 1.2.3 [Jan+15, Section 2.1, Section 3.1, and Section 2.20]**.** We will now define three possibilities to condense the definition of a graph into a matrix:

i. The *adjacency matrix* A_G of a graph G is an $n \times n$ matrix, where $(A_G)_{ij} \coloneqq 1$ iff $(i,j) \in E$ and 0 otherwise.

ii. The *oriented incidence matrix* B_G of a graph G is an $n \times m$ matrix, where

$$(B_G)_{ij} \coloneqq \begin{cases} 1 & \text{if } (i,k) = e_j \text{ and } i < k \text{ for some } k \in V, \\ -1 & \text{if } (i,k) = e_j \text{ and } i \geq k \text{ for some } k \in V, \\ 0 & \text{otherwise.} \end{cases}$$

iii. The *Laplacian matrix* L_G of a graph G is an $n \times n$ matrix, where

$$(L_G)_{ij} \coloneqq \begin{cases} \deg v_i & \text{if } i = j, \\ -1 & \text{if } i \neq j, (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

As the order of the vertices is chosen arbitrarily, so is the sign in Definition 1.2.3.ii. In the literature different ways are used, so no real standard exists. We will later generalise the notion of the incidence matrix, which will encompass all standards. For now, the edge-orientation is predetermined by the indexation of the vertices.

The adjacency matrix, the oriented incidence matrix and the Laplacian matrix are interconnected in the following way, which can be proved by a simple calculation.

Lemma 1.2.4 [Jan+15, Section 2.20]**.** *Let G be a graph without loops, then*

$$B_G B_G^{\mathrm{T}} = L_G = \operatorname{diag}(\deg(v_i)) - A_G.$$

Lemma 1.2.5. *Let G be a graph with k connected components, then*

$$\operatorname{rank} B_G = \operatorname{rank} L_G = n - k.$$

Proof. Without loss of generality let $k = 1$. For more connected components the vertices and edges can be sorted to result in a block diagonal matrix. One block now corresponds to one connected component.

Let v be a vector, such that $B_G^{\mathrm{T}} v = 0$. The matrix B_G^{T} is very sparsely populated, so only two entries are ± 1, while the rest are 0. These two entries are exactly the endpoints of an edge. We therefore have $v_i = v_j$ for two neighbouring vertices i and j. This equality transitively permeates through the whole connected component and we conclude that all entries of v must be the same. The kernel of B_G^{T} therefore has a dimension of at most 1.

However, it cannot be 0, since the rows of B_G are linearly dependant. This is, because every edge contributes an entry $+1$ and -1 each in a single row. So if we take the sum over all vertices, i.e. rows, each edge contributes $+1$ and -1 and we get a non-trivial combination of 0. So the rank must be $n - 1$.

The ranks of B_G and L_G are equal, because a vector v is in the kernel of L_G iff $L_G v = B_G B_G^{\mathrm{T}} v = 0$. If we now multiply v^{T} from the left, we get $v^{\mathrm{T}} B_G B_G^{\mathrm{T}} v = \|B_G^{\mathrm{T}} v\| = 0$ and therefore $B_G v^{\mathrm{T}} = 0$. Let us now assume that $B_G w^{\mathrm{T}} = 0$ for some vector w. Then clearly also $B_G^{\mathrm{T}} B_G w^{\mathrm{T}} = L_G w^{\mathrm{T}} = 0$. So L_G and B_G^{T} have a kernel of the same dimensionality. They also have the same number of columns and thus the same rank. □

An immediate consequence of this is that 0 is always an eigenvalue of L_G. Therefore it would also be possible to count the number of connected components by counting the multiplicity of the eigenvalue 0, because every connected component could be interpreted as a graph by itself having its own Laplacian matrix with the eigenvalue 0.

We will later often require so-called *minors* of a matrix. These are matrices, which have several rows or columns removed. As few theorems require these as intensively as the matrix tree theorem, there is no consensus on the notation. One, that has proven to be quite effective, is the following:

Definition 1.2.6. Let A be an $m \times n$ matrix, $R \subseteq [m]$ and $C \subseteq [n]$, then let $A[R \mid C] := (A_{ij})_{i \in R, j \in C}$ be the submatrix of A consisting of rows in R and columns in C. This matrix is called a *minor* of A. Let also be

i. $A(R|C) := A[[m] \setminus R | [n] \setminus C]$,

ii. $A[R|C) := A[R|[n] \setminus C]$,

iii. $A(R|C] := A[[m] \setminus R | C]$,

and let $A[S] := A[S|S]$. If the set itself is of no interest, we often omit the braces and write $A[r_1, \dots, r_k | c_1, \dots, c_\ell] := A[R|C]$, with $R = \{r_1, \dots, r_k\}$ and $C = \{c_1, \dots, c_\ell\}$, respectively for the variants of this notation.

Definition 1.2.7. Let π be a bijective function on a set $[n]$, then π is called a *permutation*.

Theorem 1.2.8 [GYZ13, Definition 2 in Section 6.3.1]**.** *The set S_n of all permutations on the set $[n]$ with composition $\circ$ is a group, called the* symmetric group*.*

Definition 1.2.9. The *sign* of a permutation $\pi \in S_n$ is defined via its number of *inversions* $\nu(\pi) := |\{(i,j) \mid 1 \le i < j \le n, \pi(i) > \pi(j)\}|$ as $\operatorname{sgn}(\pi) := (-1)^{\nu(\pi)}$.

Another common way of defining the sign of a permutation is given in the following lemma. Instead of counting the number of inversions and checking whether it is odd or even, it condenses the sign into a single product.

Lemma 1.2.10. *The sign of a permutation $\pi \in S_n$ can be expressed as*

$$\operatorname{sgn}(\pi) = \prod_{1 \le i < j \le n} \frac{\pi(j) - \pi(i)}{j - i}.$$

Proof. We consider every factor on its own. Thus, if the permutation π has an inversion at (i,j), we get $\pi(i) > \pi(j)$ and the sign of the factor is negative. Otherwise, the factor is positive. So the sign of the product is the same as the one of the permutation.

As π is a bijection and the product contains every pair (i,j) with $i < j$, every pair appears exactly once as a numerator and as a divisor in the product. While the order might be swapped, the absolute values cancel out. □

Theorem 1.2.11. *Let $A_n := \{\sigma \in S_n \mid \operatorname{sgn}(\sigma) = 1\}$ be the set of even permutations, called the* alternating group*. It is a subgroup of S_n.*

The following theorem entered the canon of mathematics as a way to calculate the determinant of the product of two rectangular matrices without explicitly multiplying them together. This is a generalisation of the general rule $\det(AB) = \det(A)\det(B)$ for two square matrices A and B.

Theorem 1.2.12 (Cauchy–Binet)**.** *Let A be an $m \times n$ matrix and B be an $n \times m$ matrix (thus AB is an $m \times m$ matrix), then*

$$\det(AB) = \sum_{S \subseteq [n], |S|=m} \det(A(\emptyset|S]) \det(B[S|\emptyset)). \tag{1.2.1}$$

Proof. At first, let us consider the case $m > n$: Here, it holds that $\operatorname{rank}(AB) \leq \operatorname{rank}(A) \leq n < m$ and therefore $\det(AB) = 0$. On the other hand, there are no subsets of $[n]$ with m elements, so the right-hand side is the empty sum and also 0.

Now let us consider $m \leq n$ and let e_i be the i-th canonical basis vector with 1 in the i-th entry and 0 otherwise. Due to the multilinearity of the determinant we can write

$$\begin{aligned}
\det(AB) &= \det \begin{pmatrix} \sum_{j_1=1}^n A_{1,j_1} B_{j_1,1} & \cdots & \sum_{j_m=1}^n A_{1,j_m} B_{j_m,m} \\ \vdots & \ddots & \vdots \\ \sum_{j_1=1}^n A_{m,j_1} B_{j_1,1} & \cdots & \sum_{j_m=1}^n A_{m,j_m} B_{j_m,m} \end{pmatrix} \\
&= \sum_{i_1,i_2,\dots,i_m=1}^{m} \sum_{j_1,j_2,\dots,j_m=1}^{n} \det(A_{i_1,j_1} B_{j_1,1} e_{i_1}, A_{i_2,j_2} B_{j_2,1} e_{i_2}, \dots, A_{i_m,j_m} B_{j_m,m} e_{i_m}) \\
&= \sum_{j_1,j_2,\dots,j_m=1}^{n} \det(A(\emptyset|j_1,\dots,j_m]) \cdot B_{j_1,1} \cdots B_{j_m,m}.
\end{aligned}$$

Clearly, if $j_\ell = j_k$ for some $\ell \neq k$, the determinant is 0 since the rank would not be full. Let us now assume that all j_k are distinct, then there exists a permutation σ, such that the image $S = \{j'_1, \dots j'_m\}$ is sorted in ascending order. Due to the linearity of the determinant, we can write $\det(A(\emptyset \mid j_1, \dots, j_m]) = \operatorname{sgn}(\sigma) \det(A(\emptyset \mid S])$. Taking a closer look at all the unordered sets contributing to S, after being ordered with their respective $\sigma \in S_m$, we get

$$\begin{aligned}
\sum_{\sigma \in S_m} \det(A(\emptyset|S]) \operatorname{sgn}(\sigma) B_{j_1,1} \cdots B_{j_m,m} &= \det(A(\emptyset|S]) \sum_{\sigma \in S_m} \operatorname{sgn}(\sigma) B_{j_1,1} \cdots B_{j_m,m} \\
&= \det(A(\emptyset|S]) \sum_{\sigma^{-1} \in S_m} \operatorname{sgn}(\sigma^{-1}) B_{j'_{\sigma^{-1}(1),1}} \cdots B_{j'_{\sigma^{-1}(m),m}} \\
&= \det(A(\emptyset|S]) \det(B[S|\emptyset)).
\end{aligned}$$

This holds true for all $S \subseteq [n]$ with m elements. So, if we consider the whole sum, we get desired result. □

We have now established a basic connection to handle graphs and connected components via matrices. A most central part still missing is an identifier for being a tree. We would like a way to characterise the notion of being a tree to some property of a matrix. This is given by the determinant.

Lemma 1.2.13. *Let T be a set of $n-1$ edges of G and v_i a vertex. If T defines a tree, then* $\det B_G(i|T] = \pm 1$ *and* 0 *otherwise.*

Proof. In Lemma 1.2.5 we proved that the rank of $B_G(\emptyset|T]$ is $n-1$ if T is connected. Since there are $n-1$ edges for n vertices, there can only be a circuit iff the vertices in T are not all connected. So the rank is less than $n-1$ and the determinant is 0. Thus, we now only consider a tree T. As $B_G(\emptyset|T]$ is an $n \times (n-1)$ matrix with rank $n-1$, removing a row does not change the rank. So $\det B_G(i|T] \neq 0$ iff T is a tree. Since we have no circuits in T, there exists an ordering of vertices and edges, such that $B_G(\emptyset \mid T]$ is a lower triangular matrix and the vertex corresponding to the removed row is labelled with i. We then have $\det(B_G(i|T]) = \pm 1$, where the sign is directly dependant on the chosen orientation of the edges in i. □

1.3 The Matrix Tree Theorem

Using Theorem 1.2.12, it is possible to formulate the nowadays most canonical proof of the matrix tree theorem. The theorem follows almost immediately from the Cauchy–Binet formula and is thus a great example for a simple application of it.

Theorem 1.3.1 (Matrix Tree Theorem)**.** *Let G be a graph, then* $\det(L_G(n)) = \kappa(G)$.

Proof. According to Lemma 1.2.4, we have $L_G = B_G B_G^{\mathrm{T}}$, so also $L_G(n) = B_G(n|\emptyset) B_G^{\mathrm{T}}(\emptyset|n)$. Using Theorem 1.2.12, the determinant can be calculated via

$$\begin{aligned}\det(L_G(n)) &= \sum_{\substack{S\subseteq[m],\\ |S|=n-1}} \det(B_G(n|S]) \det(B_G^{\mathrm{T}}[S|n)) \\ &= \sum_{\substack{S\subseteq[m],\\ |S|=n-1}} \det(B_G(n|S])^2.\end{aligned}$$

Following Lemma 1.2.13, we know that $\det(B_G(n|S]) = \pm 1$ iff the given selection S forms a tree with n vertices and $n-1$ edges. Since all combinations appear exactly once, every tree contributes 1 and thus $\det(L_G(n)) = \kappa(G)$. □

Example 1.3.2. As an example, we will now calculate the number of spanning trees in the cycle graph C_n of size $n > 2$, which consists of a singular cycle without any branches. The only possible spanning trees are the ones, which remove exactly one edge. So we expect $\kappa(C_n) = n$ and, indeed, $\det(L_{C_n}(n)) = n$ follows.

$$B_{C_n} = \begin{pmatrix} 1 & & & & 1 \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \\ & & & -1 & -1 \end{pmatrix}, \qquad L_{C_n} = \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & -1 & & \ddots & \\ & & \ddots & & -1 \\ -1 & & & -1 & 2 \end{pmatrix} \qquad \lozenge$$

A popular variant of the matrix tree theorem is the following reformulation, which disregards determinants altogether and instead uses the eigenvalues of the matrix. To prove this algebraically—the proofs often rely on spectral theory—we will employ a trick used in [Cam08]. We also require the following theorem.

Theorem 1.3.3 [HJ12, Theorem 1.3.12]**.** *Let A, B be diagonalisable $n \times n$ matrices. Then A and B commute iff they are simultaneously diagonalisable, i.e. there exists a nonsingular matrix S, such that SAS^{-1} and SBS^{-1} are diagonal.*

Theorem 1.3.4 [Big93, Corollary 6.5]**.** *Let G be a graph and $\lambda_1 = 0, \lambda_2, \dots, \lambda_n$ the eigenvalues of L_G, then the following hold:*

1. *Let $i, j \leq n$, then* $\det L_G(1) = (-1)^{i+j} \det L_G(i|j)$,
2. $\lambda_2 \cdots \lambda_n = \sum_{k=1}^{n} \det L_G(k)$,
3. $\kappa(G) = \frac{\lambda_2 \cdots \lambda_n}{n}$.

Proof. We start by proving 1, for which—without loss of generality—we will only prove that $\det L_G(1) = -\det L_G(1|2)$. We already know that $\sum_{k=1}^{n} c_k = 0$, where c_i is the i-th column of L_G. So $c_2 = -(c_1 + c_3 + \cdots + c_n)$ and we can replace this column in the matrix and apply the multilinearity of the determinant:

$$\begin{aligned}\det(L_G(1)) &= \det(c_2, c_3, \dots, c_n)\\ &= \det(-(c_1 + c_3 + \cdots + c_n), c_3, \dots, c_n)\\ &= \det(-c_1, c_3, \dots, c_n) + \sum_{k=3}^{n} \underbrace{\det(-c_k, c_3, \dots, c_n)}_{=0}\\ &= -\det(L_G(1|2))\end{aligned}$$

The proof of 2 will be done in two parts: The first step is to prove $\det(L_G + \mathbb{1}) = n^2 \det(L_G(1))$ and the second $\det(L_G + \mathbb{1}) = n\lambda_2 \cdots \lambda_n$, where $\mathbb{1}$ is the matrix with 1 in every entry.

The first half is done by row and column operations, which do not change the determinant, apart from pulling out a factor n twice. The matrix is transformed into a block triangular matrix, for which the determinant is straightforward to evaluate. Since summing up all rows or columns in the Laplacian gives 0, we end up with n in $L_G + \mathbb{1}$ as the only remaining contribution is from $\mathbb{1}$:

$$\begin{aligned}\det(L_G + \mathbb{1}) &= \det\left(\begin{array}{c|c} n & \\ \vdots & L_G(1|\emptyset) + \mathbb{1} \\ n & \end{array}\right) = n \det\left(\begin{array}{c|c} 1 & \\ \vdots & L_G(1|\emptyset) + \mathbb{1} \\ 1 & \end{array}\right)\\ &= n \det\left(\begin{array}{c|ccc} n & n & \cdots & n \\ \hline 1 & & & \\ \vdots & & L_G(1) + \mathbb{1} & \\ 1 & & & \end{array}\right) = n^2 \det\left(\begin{array}{c|ccc} 1 & 1 & \cdots & 1 \\ \hline 1 & & & \\ \vdots & & L_G(1) + \mathbb{1} & \\ 1 & & & \end{array}\right)\\ &= n^2 \det\left(\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 1 & & & \\ \vdots & & L_G(1) & \\ 1 & & & \end{array}\right) = n^2 \det(L_G(1)).\end{aligned}$$

Clearly, also $L_G\mathbb{1} = \mathbb{0} = \mathbb{1}L_G$ since multiplying with $\mathbb{1}$ is essentially summing up a row or column. Following Theorem 1.3.3 we can find a matrix S, such that both are diagonal. The eigenvalues of $\mathbb{1}$ are $n, 0, \dots, 0$ and of L_G $0, \lambda_2, \dots, \lambda_n$, where the eigenvectors corresponding to n and 0 are $(1, \dots, 1)$ in both cases, and so

$$\det(L_G + \mathbb{1}) = \det(S(\operatorname{diag}(n, \lambda_2, \dots, \lambda_n))S^{-1}) = n\lambda_2 \cdots \lambda_n.$$

Combining both parts, we get $n\lambda_2 \cdots \lambda_n = n^2 \det(L_G(1))$. Item 1 shows that the determinant is independent of the removed row and column and thus we get $\lambda_2 \cdots \lambda_n = \sum_{k=1}^{n} \det L_G(k)$. Finally, 3 follows directly by applying 2 to the matrix tree theorem. □

Example 1.3.5. Let us consider a complete graph K_n with n vertices. In K_n every vertex is connected to every other vertex and thus the Laplacian matrix is

$$L_{K_n} = \begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & & \vdots \\ \vdots & & \ddots & -1 \\ -1 & \cdots & -1 & n-1 \end{pmatrix} = \operatorname{diag}(n, \dots, n) - \mathbb{1}.$$

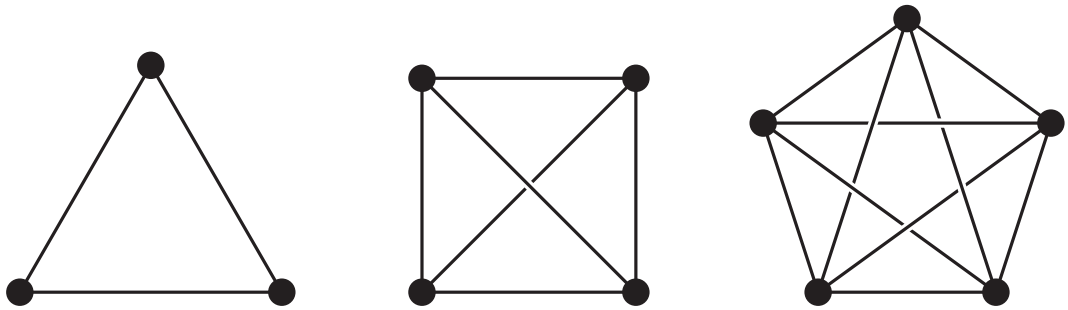


Figure 1.3: The complete graphs K_3, K_4 and K_5.

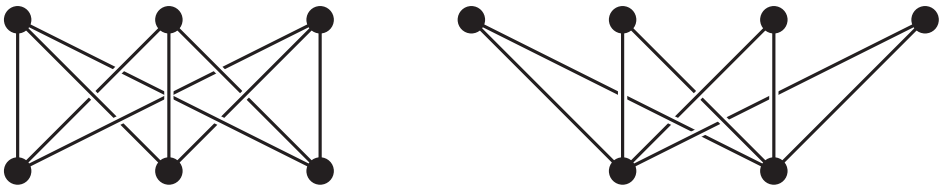


Figure 1.4: The bipartite graphs $K_{3,3}$ and $K_{2,4}$.

A minor of this matrix can easily transformed into an upper triangular matrix via subtracting the last row from all others and then adding an n-th of all other rows to the last. We end up with a matrix with $n, \dots, n, 1$ on the diagonal, $-n+2$ in all other entries of the last column and 0 everywhere else. The determinant then is n^{n-2}.

If we now take a look at the bipartite graph $K_{n,m}$, which consists of two sets with n and m elements and all edges between these two sets, we get the Laplacian matrix $L_{K_{n,m}}$, for which $\det(L_{K_{n,m}}(1)) = n^{m-1}m^{n-1}$.

$$L_{K_{n,m}} = \begin{pmatrix} \operatorname{diag}(m, \dots, m) & -\mathbb{1} \\ -\mathbb{1} & \operatorname{diag}(n, \dots, n) \end{pmatrix} \begin{matrix} \} \, n \text{ rows} \\ \} \, m \text{ rows} \end{matrix} \qquad \lozenge$$

Corollary 1.3.6 (Cayley's Theorem) [Cay89]**.** *Let K_n be a complete graph with n vertices, then K_n contains n^{n-2} spanning trees.*

This result can also be reached by more basic methods. For example, since all vertices are connected, we can count the number of sequences between the vertices. In A. Cayley's original proof [Cay89] he did exactly that on the example of $n = 6$, where he labelled the vertices $\alpha, \beta, \gamma, \delta, \varepsilon, \zeta$ and noted that all vertices were interchangeable. So he could, for example, formulate a canonical representation as a polynomial of the graphs with edges of type $(\alpha, \zeta), (\alpha, \varepsilon), (\alpha, \beta), (\beta, \gamma), (\gamma, \delta) \mapsto \alpha^3\beta^2\gamma^2\delta\varepsilon\zeta$. This type is illustrated in Figure 1.5.

With careful consideration he finds that there are several ways to order these canonical representations and ends up with the polynomial

$$(\alpha + \beta + \gamma + \delta + \varepsilon + \zeta)^4 \alpha\beta\gamma\delta\varepsilon\zeta,$$

which can be solved via the multinomial theorem. As he did not want to list all 126 summands, he allowed some form of relabelling, to ensure, that the powers always decrease from α to ζ. This is acceptable as the ultimate goal is to plug in 1 for every variable. So he

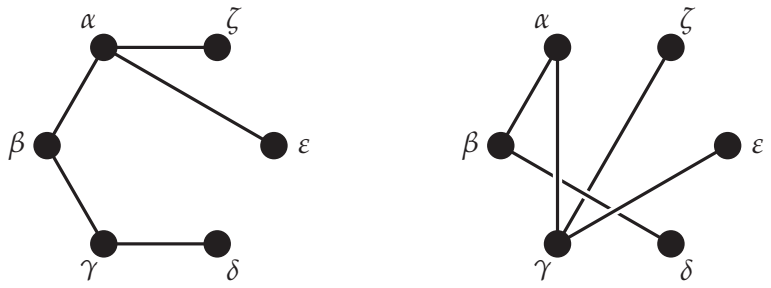


Figure 1.5: A canonical graph used in Cayley's theorem. All relabellings of vertices are identified with this graph, of which the one illustrated on the right can be transformed via $\gamma \mapsto \alpha$, $\alpha \mapsto \beta$ and $\beta \mapsto \gamma$.

counts some canonical representation by the multiplicity of distinct orders. In sum, there are six relabellings of α^4, 30 for $\alpha^3\beta$, 15 for $\alpha^2\beta^2$, 60 for $\alpha^2\beta\gamma$, and 15 for $\alpha\beta\gamma\delta$. If we now plug in 1, we get $6^4 = 1296$.

Corollary 1.3.7 (Scoins' Theorem) [Sco62]**.** *Let $K_{n,m}$ be a bipartite graph with $n + m$ vertices in two sets of sizes n and m, then $K_{m,n}$ contains $n^{m-1}m^{n-1}$ spanning trees.*

H. I. Scoins' original proof is based on generating functions. He constructed it by at first counting all ordered sets with one vertex fixed. Those are not necessarily a tree, but always a tree with $n - p$ and $m - q$ vertices in the two sets combined with a supplementary subgraph with $p + q$ points and $p + q$ edges between the two sets of vertices. Summing up all those combinations, he found all $n^{m-1}m^n$ combinations, where one vertex is fixed. Summing this up leads to

$$\begin{aligned}\sum_{n=1}^{\infty}\sum_{m=0}^{\infty}\frac{m^{n-1}n^{m+1}}{n!m!}x^n y^m &= x + \sum_{p=0}^{\infty}\sum_{q=0}^{\infty}\sum_{r=1}^{\infty}\sum_{s=0}^{\infty}\frac{x^p y^q x^r y^s}{p!q!r!s!}S(p,q)rT(r,s)\\ &= \sum_{p=0}^{\infty}\sum_{q=0}^{\infty}\frac{x^p y^q}{p!q!}S(p,q)\sum_{r=1}^{\infty}\sum_{s=0}^{\infty}\frac{x^r y^s}{r!s!}rT(r,s). \qquad (1.3.1)\end{aligned}$$

This enormous sum contains $S(p,q)$ and $T(r,s)$, which are the number of supplementary subgraphs and trees respectively. He considered $g(x,y) = x + \sum_{r=1}^{\infty}\sum_{s=1}^{\infty}\frac{x^r y^s r^s s^{r-1}}{r!s!}$ and showed that

$$x\frac{\partial g}{\partial x} = \Big(1 + y\frac{\partial g}{\partial y}\Big)g(x,y),$$

for which the left-hand side is equal to the left-hand side of (1.3.1) and the part in braces to the first sum of the right-hand side. So therefore the remaining parts are equal too and $T(r,s) = r^{s-1}s^{r-1}$. In a closing remark he wrote that a similar construction could be done for Cayley's theorem.

2 A Closer Look

In Chapter 1 we established the necessary preliminaries and a first formulation and proof of the matrix tree theorem. As mentioned in the introduction, this is not *the* first proof. G. R. Kirchhoff used a quite different approach and while the proof presented in Chapter 1 is rather short, it does not give much insight into how the matrix tree theorem works.

We change this in this section. The following proofs all highlight different features of the spanning trees within a graph. They are sorted chronically, by the release of their initial formulation, which is not necessarily the one presented here. This is especially prominent for the historical approach formulated in 1847. The proofs using a sign-reversing involution and an induction via deletion–contraction, described in Section 2.2 and Section 2.3, appeared almost simultaneously: S. Chaiken proved the theorem using an involution in 1982 [Cha82]. In the same year, M. Lewin presented the proof using deletion–contraction, citing S. Chaiken's work as "to appear" [Lew82, p. 70]. He also notes that his proof is "leaning on an idea of Hutschenreuther" [Lew82, p. 56], who formulated it in 1967 [Hut67]. Not only M. Lewin's proof appeared before, but in a very similar form also S. Chaiken's—in 1978 by J. B. Orlin [Orl78]. We dedicate Section 2.2.2 to this variant and present S. Chaiken's proof in Section 2.2.3. So the question of "who came first" is rather convoluted.

A comparatively novel approach is presented in Section 2.4. This proof was formulated in 2013 by M. J. Kozdron, L. M. Richards, and D. W. Stroock, based upon Wilson's algorithm to randomly generate a tree [Wil96] and the Markov chain tree theorem, proved in 1989 [AT89]. They combined these facts and deduced the matrix tree theorem from a probabilistic point of view.

2.1 The Historical Approach

In the proof of Section 1.3 we use that the determinant of the oriented incidence matrix of a tree is ± 1, whereas it is 0 if the graph contains a cycle, and thus also a circuit. This lemma was not available to G. R. Kirchhoff, so he had to take a different approach: He tried to identify a minimal set of circuits, which were then cut. To be able to formally describe this, we introduce the following:

Definition 2.1.1 [GYZ13, Definition 17 in Section 6.4.1]**.** Let $G = (V, E)$ be a graph and C_1, C_2 subsets of E, then the *symmetric difference* is

$$C_1 \oplus C_2 := \{e \in E \mid e \in (C_1 \cup C_2) \setminus (C_1 \cap C_2)\}.$$

With this, we can now perform calculations on circuits, but in general the symmetric difference might not be a circuit. For example, we could take two circuits from different connected components. These two circuits are obviously disjunct, so the symmetric difference is exactly those two circuits. We can, however, find a useful algebraic structure:

Definition 2.1.2. A graph is called *Eulerian* iff every vertex is of even degree.

The name stems from the problem of the "Seven Bridges of Königsberg", which was solved by L. Euler, and states that a graph contains a circuit traversing all vertices iff it is connected and every vertex is of even degree [Eul41].

Lemma 2.1.3 [GYZ13, Fact 7 in Section 6.4.1]**.** *Let G be a graph and S be the set of all Eulerian subgraphs of G, then $(S, \oplus)$ is isomorphic to a vector space over the finite field $\mathbb{Z}_2$.*

Proof. We will first prove that $(S, \oplus)$ is an abelian group: The symmetric difference is only concerned about the number of appearances of an edge, so it is associative and commutative. The neutral element is the empty set $\emptyset$. If the symmetric difference of any subgraph is formed with itself, all edges appear twice and are removed. Therefore it is involutory.

The scalar multiplication is trivial since $\mathbb{Z}_2$ only contains 0 and 1 and every Eulerian subgraph is its own inverse. Multiplication with 0 returns $\emptyset$ and multiplication with 1 is the identity. Thus, S is isomorphic to a vector space over $\mathbb{Z}_2$. □

The set S, combined with the symmetric difference, can now be interpreted as a vector space and we will call this algebraic structure the *cycle space* of G. Finding a basis and the dimension of the cycle space is of much interest as it enables counting the required cuts: Every basis cycle has to be cut, i.e. an edge removed, in order to remove all closed shapes. All other Eulerian subgraphs can only be generated via the symmetric difference of basis cycles. So we can define the following:

Definition 2.1.4. Let G be a graph and S its cycle space, then the *circuit rank* of G is $\operatorname{rank} G := \dim S$.

Due to the connection of the circuit rank to the cycle space, this is also sometimes called the *cycle rank*. However, we will not use this name as there exists a non-equivalent definition also called the cycle rank. In a similar sense, we avoid calling the elements of the cycle space *cycles*.

Definition 2.1.5. Let G be a graph and S its cycle space, then a *cycle basis* is a minimal set of cycles generating S. A *fundamental cycle basis* is a cycle basis for which a tree T on every connected component of G exists, such that every cycle in that component has exactly one edge not in T.

Lemma 2.1.6 [GYZ13, Fact 8 in Section 6.4.1]**.** *Every graph has a fundamental cycle basis.*

Proof. Let T be a tree of graph G and $\mathcal{B}_T$ a set of cycles, such that every cycle contains a single different edge not in T. This set $\mathcal{B}_T$ is a fundamental cycle basis iff it is a cycle basis. No cycle of $\mathcal{B}_T$ can be combined by the others as it contains a unique edge. Let now C be an arbitrary Eulerian subgraph of G. We can then write $C = C' \oplus B$, where B is a Eulerian subgraph in the subspace spanned by $\mathcal{B}_T$ and C' is a Eulerian subgraph in T. A tree only contains the trivial Eulerian subgraph $\emptyset = C'$, so $\mathcal{B}_T$ is a cycle basis. □

Corollary 2.1.7. *Let G be a graph and $\mathcal{B}$ a fundamental cycle basis of G, then there exist orderings of edges and cycles, such that $Id = B(\emptyset | [\operatorname{rank} G]]$, where $B_{ij} = 1$ iff the i-th basis cycle contains the j-th edge, and 0 otherwise.*

Proof. Every cycle in a fundamental cycle basis contains an edge only contained in this cycle. If we number the edges and cycles such that the i-th cycle contains the i-th edge, we get the identity in the first $\operatorname{rank} G$ rows and columns of B. □

The circuit rank is closely connected to trees and, thanks to Lemma 2.1.6, we can always choose a fundamental cycle basis. These can be quickly found via finding a tree for every connected component and adding an edge one at a time, to form a single cycle.

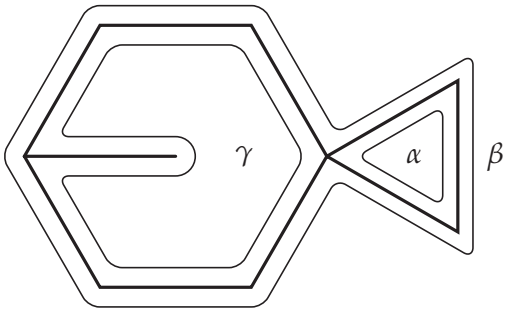


Figure 2.1: A graph with three face cycles, of which α is a cycle, β a circuit and not a cycle, and γ a closed walk and not a circuit. The closed walk γ contains an edge, which the cycle space is oblivious to. As the walk traverses the edge twice, it is equivalent to not traversing it at all, and thus can be ignored.

Theorem 2.1.8 [GYZ13, Fact 13 in Section 6.4.1]**.** *Let G be a graph with n vertices, m edges and c connected components, then* $\operatorname{rank} G = m - n + c$.

Proof. Without loss of generality, we assume G to be connected. A tree of G has circuit rank 0, n vertices and $n - 1$ edges. A fundamental cycle basis has a cycle for every edge removed and therefore $\operatorname{rank} G = m - (n - 1)$ for a connected graph. □

Definition 2.1.9. A graph G is called *planar* iff it can be embedded into the plane, such that no edges intersect.

Definition 2.1.10. Let G be a connected planar graph. It separates the plane into *faces*, where a face is the set of all points of the plane that can be connected with a line not intersecting G. A *face cycle* is the closed walk accompanying a face and circumscribing only it.

The definitions surrounding planarity and faces usually rely on an intuitive understanding of an embedding of a graph into the plane and understanding of connectedness in two dimensions. We too will gloss over the formal definitions as they are quite involved.

Also note that, even though the subgraphs surrounding faces are called face *cycles*, they might not be cycles. This can happen if the graph is connected by a single edge. An example is presented in Figure 2.1. This stems from them being elements of the cycle space, which gets its name from the cycle basis.

Theorem 2.1.11 (Euler's Polyhedron Formula) [GYZ13, Section 1.3.3]**.** *Let G be a connected planar graph with n vertices, m edges and f faces, then*

$$n - m + f = 2.$$

Theorem 2.1.12. *Let G be a connected planar graph, then all but one face cycles form a linearly independent set of size* $\operatorname{rank} G$.

Proof. As G is planar, Theorem 2.1.11 holds and $f = 2 - n + m$. Due to Theorem 2.1.8, we also know that $\operatorname{rank} G = m - n + 1$. Therefore $\operatorname{rank} G = f - 1$ and all that is left to show is the independence of the face cycles.

Let $\mathcal{F} = \{C_1, C_2, \dots, C_{f-1}\}$ be the set of all face cycles. Every edge e in a face cycle C is in exactly one other face cycle C', so to construct a combination of C_1 all other face cycles have to be used in order to avoid remaining edges. Then $\emptyset = C_1 \oplus C_2 \oplus \cdots \oplus C_{f-1}$ and

$C_i = C_1 \oplus \cdots \oplus C_{i-1} \oplus C_{i+1} \cdots \oplus C_{f-1}$. So $\mathcal{F}$ is linearly dependant, but any set $\mathcal{F} \setminus \{C_i\}$ forms a linearly independent set of size rank G. □

This theorem almost gives us a cycle basis. Unfortunately, the face cycles might not be cycles, and thus will not form a cycle basis. If, however, all face cycles are cycles, then the theorem proves that they are maximally linearly independent and that the set is a basis.

Usually, we will remove the face cycle belonging to the outer face containing ∞. However, this is no restriction as the embedding into the plane can be chosen arbitrarily and any face can be drawn as the outer face.

2.1.1 Kirchhoff's Formulation

Kirchhoff uses his two circuit laws to formulate the required equations for the matrix tree theorem [Kir47]. For this, he introduces ε_k, which are ± 1 and describe the direction of the edge k. This direction is chosen arbitrarily, but we will use the convention that the sign is positive iff we move from vertex i to vertex j with $i < j$. We will see that this convention agrees with our definition of the oriented incidence matrix. As we also need a different kind of incidence matrix, we repeat the previous definition and add the new one.

In contrast to our convention of using n for the number of vertices, G. R. Kirchhoff used m for the number of vertices, n for the number of edges, and μ for rank G. So we use the unambiguous notation $|V|$ and $|E|$ in this subsection.

Definition 2.1.13 [Jan+15, Chapter 3]**.** Let G be a graph with $|V|$ vertices and $|E|$ edges.

i. The *oriented (vertex–edge) incidence matrix* B_G^V of G (with an arbitrary edge-orientation) is an $|V| \times |E|$ matrix, where

$$(B_G^V)_{ij} := \begin{cases} 1 & \text{if } (i,k) = e_j \text{ for some } k \in V, \\ -1 & \text{if } (i,k) = -e_j \text{ for some } k \in V, \\ 0 & \text{otherwise.} \end{cases}$$

ii. Let $\mathcal{C}$ be a cycle basis of G. The *oriented cycle–edge incidence matrix* B_G^C of G (with an arbitrary edge-orientation) is a rank $G \times |E|$ matrix, where

$$(B_G^C)_{ij} := \begin{cases} 1 & \text{if } e_j \in C_i, \\ -1 & \text{if } -e_j \in C_i, \\ 0 & \text{otherwise.} \end{cases}$$

We will now state Kirchhoff's two circuit laws, which G. R. Kirchhoff first published in 1845 [Kir45].[3] There, G. R. Kirchhoff considered an electrical circuit consisting of a set of cables and a set of points, which connect cables with each other. In the following we consider a subset C of cables, which form a closed shape, and a subset N of cables, which all share a connecting point. Solving these equations gives the matrix tree theorem in its original form [Kir47].

[3] The distinction "published" is important here as C. F. Gauss formulated the circuit laws in 1833, which can be found in the first letter of chapter *Handschriftlicher Nachlass* in [Gau67]. It is unknown why this was kept unpublished, but "presumably because their [C. F. Gauss' and W. E. Weber's] interest centered on terrestrial magnetism." [May80, p. 305]

Kirchhoff's Voltage Law If the cables $k \in C$ with the resistances w_k, intensities of currents (modern: currents) I_k and electromotive forces (modern: voltage sources) E_k form a closed shape, Kirchhoff's voltage law states:

$$\sum_{k \in C} \varepsilon_k w_k I_k = \sum_{k \in C} E_k$$

Electromotive forces are only introduced through batteries and are otherwise 0. The intensities make up the vector J of unknowns. The electrical network can be interpreted as a graph G, G. R. Kirchhoff then chose a fundamental cycle basis as the target for his current law. This leads to the linear system of equations $W \cdot I = 0$. As we are only concerned with counting, we can ignore the exact values of w_k and only keep the sign governed by ε_k. This sign introduces a direction onto all edges. The value $\varepsilon_k w_k$ is positive iff the cycle moves in the same direction as the edge, and is negative otherwise. If all resistances w_k are equal to 1, we get the oriented cycle–edge incidence matrix B_G^C for W.

Kirchhoff's Current Law Kirchhoff also stated a different law, which is concerned about individual points where cables $k \in N$ collide: Incoming and outgoing intensities at these points have to add up to 0. This means

$$\sum_{k \in N} \varepsilon_k I_k = 0.$$

If we form a matrix out of the equations for all vertices, we get the system of equations $B_G^V \cdot I = 0$, where B_G^V is the oriented vertex–edge incidence matrix. Kirchhoff took all these $\operatorname{rank} G + |V| = |E| - |V| + 1 + |V| = |E| + 1$ equations and formulated the matrix tree theorem:

Theorem 2.1.14 (Original Matrix Tree Theorem) [Kir47]**.** *Let G be a graph and B_G^C and B_G^V its oriented cycle–edge and vertex–edge incidence matrices respectively, and $|E|$ the number of edges in G, then*

$$\det\left(\left(\frac{B_G^C}{B_G^V}\right)(|E|+1\,|\,\emptyset)\right) = \pm\kappa(G). \tag{2.1.1}$$

This first formulation contains no mention of the Laplacian matrix of G but instead uses two different oriented incidence matrices. The original proof shows this relation by using Cramer's rule to solve the determinant [Kir47]. As mentioned before, G. R. Kirchhoff had to use different notation. The required notation used here was not yet invented. His proof uses the notation of the aforementioned sums, where he relabelled the ε_j to α_j^i with $i \in \{1, \dots, \operatorname{rank} G\}$ and $j \in \{1, \dots, |V|\}$ for the voltage laws and $j \in \{|V|+1, \dots, |E|\}$ for the current laws. To make this work, he also had to allow 0 for α_j^i if the corresponding ε_j—and thus the edge labelled with j—would not appear in the equation. Note that he omitted the last equation given by the current law, which would be labelled with $j = |E| + 1$.

We will, however, choose a different route and show the strong connection between this original formulation and the modern matrix tree theorem. This is due to E. C. Kirby et al. in [Kir+04], where they first formulated the cycle theorem in this context.

2.1.2 The Cycle Theorem

We now focus on the equations resulting from the voltage law. As a tree is defined as a connected graph without cycles, we want to resolve all those equations. Assume that we start with a connected graph G. We remove the redundancy giving raise to the equations

via deleting an edge in every equation. The graph stays connected, but the cycle is cut and thus removed. If this is done for all cycles, the remaining graph is a tree. The specific selection of deleted edges leads to a specific graph. So we expect the number of spanning trees to be connected to the different ways of cutting all cycles.

Definition 2.1.15 [Kir+04, pp. 267–268]**.** Let G be a graph and $\mathcal{S}$ a set of Eulerian subgraphs of G, then $M_{\mathcal{S}}$ is the *cycle-overlap matrix*[4], where

$$(M_{\mathcal{S}})_{ij} := \sum_{e \in C_i} \sum_{e' \in C_j} \begin{cases} 1 & \text{if } e \text{ and } e' \text{ overlap and are oriented in the same direction,} \\ -1 & \text{if } e \text{ and } e' \text{ overlap and are oriented in different directions,} \\ 0 & \text{if } e \text{ and } e' \text{ do not overlap.} \end{cases}$$

This definition respects the direction of the Eulerian subgraphs. So if C_i and C_j share one edge and pass it in the same direction it counts as $+1$, whereas if one passes it in the other direction it counts as -1. From $(M_{\mathcal{S}})_{ij} = 0$ it does not follow that C_i and C_j share no edges, it is also possible that they share the same number of edges in the same as in opposite directions.

The diagonal elements of the cycle-overlap matrix indicate the self-overlap of Eulerian subgraphs. For cycles, this is exactly their length $|C_i|$. This may not hold true for general Eulerian subgraphs as can be seen in Example 2.1.21.

Theorem 2.1.16 (Cycle Theorem) [Kir+04; Che97, Corollary 2.24]**.** *Let G be a graph and $\mathcal{C}$ be a set of* rank G *linearly independent Eulerian subgraphs, and T be a spanning tree of G. Let $Z = (z_{ij})$ be the matrix of size* rank $G \times m$ *counting the signed occurrences of edge e_j in C_i, according to an arbitrary but fixed edge-orientation, then*

$$\frac{\det(M_{\mathcal{C}})}{\det(Z(\emptyset \,|\, T])^2} = \kappa(G). \tag{2.1.2}$$

Lemma 2.1.17 [Tut65, Theorem 5.46]**.** *Let G be a connected graph, $\mathcal{C}$ a cycle basis and Z as in Theorem 2.1.16. Let also T and T' be two spanning trees of G, then*

$$\det(Z(\emptyset \,|\, T])^2 = \det(Z(\emptyset \,|\, T'])^2 = 1. \tag{2.1.3}$$

Let S be a subset of edges of G that does not form a spanning tree, then $\det(Z(\emptyset \,|\, S]) = 0$.

Note that the matrix Z is an oriented cycle–edge incidence matrix iff the set of Eulerian subgraphs form a cycle basis. The above lemma thus can be understood as proving a property of the cycle incidence matrix, which will serve as a stepping stone to simplify the proof of the cycle theorem.

Proof. Let us first consider the case that S is not a tree. Then there exists a cycle C in S and a cycle basis $\mathcal{C}'$, such that $C \in \mathcal{C}'$. As both $\mathcal{C}$ and $\mathcal{C}'$ are cycle bases, there exists a full rank transformation matrix A and

$$\det(Z(\emptyset \,|\, S]) = \det(A) \det(Z'(\emptyset \,|\, S]) = 0.$$

Let otherwise $Z(\emptyset \,|\, T]$ have full rank and $\mathcal{C}'$ be the fundamental cycle basis to tree T. Because $\mathcal{C}$ and $\mathcal{C}'$ are bases, there again exists a transformation matrix A, such that $T\mathcal{C} = \mathcal{C}'$ and $\det(T) \det(Z(\emptyset \,|\, T]) = \det(Z'(\emptyset \,|\, T]) = 1$ as $Z'(\emptyset \,|\, T] = Id$. As all these matrices are integer matrices, we know that their determinants must be integers too. So it follows that $\det(Z(\emptyset \,|\, T]) = \pm 1$. □

[4] This name might seem counterintuitive as we not only allow cycles, but even Eulerian subgraphs. However, the name is derived from the cycle space, which is named after the cycle basis.

Theorem 2.1.18 (Cycle Theorem for Cycle Bases) [Tut65, p. 18]**.** *Let G be a graph and $\mathcal{C}$ a cycle basis of G, then*

$$\det(M_{\mathcal{C}}) = \kappa(G). \tag{2.1.4}$$

We want to solve the determinant of the product of two non-square matrices. This is very similar to the proof of the matrix tree theorem. The main difference is using an oriented cycle–edge incidence matrix instead of the oriented vertex–edge incidence matrix.

W. T. Tutte mentions this theorem as a direct result of Lemma 2.1.17:"This generalizes a well known formula for the number of spanning trees" [Tut65, p. 18]. The formula he talks about is the matrix tree theorem, which appears as a parallel theorem to the cycle theorem for us. We will see in Section 2.3.2 how much on point his formulation is.

Proof. Lemma 2.1.17 shows that the determinant of $Z(\emptyset|T]$ is ± 1 iff T is a tree. So if we use Theorem 1.2.12 to calculate $\det(M_{\mathcal{C}}) = \det(ZZ^{\mathrm{T}})$, we will end up evaluating exactly those submatrices:

$$\begin{aligned}\det(M_{\mathcal{C}}) &= \sum_{S\subseteq[m],|S|=n-1} \det(Z(\emptyset|T])\det(Z^T[T|\emptyset)) \\ &= \sum_{S\subseteq[m],|S|=n-1} \det(Z(\emptyset|T])^2 = \kappa(G).\end{aligned}$$

□

Theorem 2.1.18 presents itself very similar to the matrix tree theorem and even the proof consists of only minor modifications. It could therefore be considered a matrix tree theorem by itself. The cycle theorem in combination with the modern matrix tree theorem now form the basis for the original matrix tree theorem. G. R. Kirchhoff derived these results from his observed circuit laws and saw no need to prove them. The above proof shows, that there indeed is a purely formal way not relying on some specific property of electric circuits. All that is left is to combine these two theorems.

Proof of Theorem 2.1.16. From Theorem 2.1.18 we know that $\det(M_{\mathcal{B}}) = \kappa(G)$ for a fundamental cycle basis $\mathcal{B}$, which—using Corollary 2.1.7—can express $Z_{\mathcal{C}}$ as

$$Z_{\mathcal{C}} = \left(\begin{array}{c|ccc} & z_{c+1,1} & \cdots & z_{e,1} \\ A & \vdots & \ddots & \vdots \\ & z_{c+1,c} & \cdots & z_{e,c} \end{array}\right) = \left(\begin{array}{c|ccc} & z'_{c+1,1} & \cdots & z'_{e,1} \\ Id & \vdots & \ddots & \vdots \\ & z'_{c+1,c} & \cdots & z'_{e,c} \end{array}\right) \cdot A = Z_{\mathcal{B}} \cdot A.$$

If we now apply Theorem 2.1.18 to this, we get

$$\det(M_{\mathcal{C}}) = \det(Z_{\mathcal{B}} A A^{\mathrm{T}} Z_{\mathcal{B}}^{\mathrm{T}}) = \det(A)^2 \kappa(G),$$

but A is exactly a square submatrix of $Z_{\mathcal{C}}$ of size rank G with full rank, so $A = Z_{\mathcal{C}}(\emptyset|T]$ for some spanning tree T of G, according to Lemma 2.1.17. □

Proof of Theorem 2.1.14. Let G be a graph with $|V| = n$ vertices and $|E| = m$ edges. We will prove the original formulation of the matrix tree theorem by calculating the determinant of the matrix

$$\left(\frac{B_G^C}{B_G^V}\right) \cdot \left(B_G^{C\mathrm{T}} \;\middle|\; B_G^{V\mathrm{T}}\right) = \left(\begin{array}{c|c} B_G^C B_G^{C\mathrm{T}} & B_G^C B_G^{V\mathrm{T}} \\ \hline B_G^V B_G^{C\mathrm{T}} & B_G^V B_G^{V\mathrm{T}} \end{array}\right).$$

We already know that $B_G^V B_G^{V\mathrm{T}} = L_G$ and $\det(L_G(n)) = \kappa(G)$ by the matrix tree theorem. There also exists a fundamental cycle basis $\mathcal{C}$, such that $B_G^C B_G^{C\mathrm{T}} = M_{\mathcal{C}}$ gives us a cycle-overlap matrix, for which $\det(M_{\mathcal{C}}) = \kappa(G)$ holds by Theorem 2.1.16. All we have to show now is $B_G^C B_G^{V\mathrm{T}} = 0$.

If we take a closer look at one entry of this matrix, we can see that it is defined by

$$\sum_{k=1}^{m} (B_G^C)_{\alpha k} (B_G^V)_{vk},$$

where we sum over all edges, one cycle α, and one vertex v. As α enters v via edge k we have $(B_G^C)_{\alpha k} = \pm 1$, according to the edge-orientation, but as we are entering v we also have $(B_G^V)_{vk} = \mp 1$. So this summand is always -1. When α then leaves v via edge k' we again have $(B_G^C)_{\alpha k'} = \pm 1$, but now the perspective of v coincides with the one of α and we have the same sign and the summand for k' is $+1$. As we only consider cycles, we have to leave every vertex we enter, so this sums up to 0 and the matrix is $\mathbb{0}$.

We can now plug in the known values for the submatrices and we only have to evaluate the determinant of a block diagonal matrix for whose blocks the determinants are known:

$$\begin{aligned}
\det\left(\left(\frac{B_G^C}{B_G^V}\right)(m+1|\emptyset)\right)^2 &= \det\left(\left(\begin{array}{c|c} B_G^C B_G^{C\mathrm{T}} & B_G^C B_G^{V\mathrm{T}} \\ \hline B_G^V B_G^{C\mathrm{T}} & B_G^V B_G^{V\mathrm{T}} \end{array}\right)(m+1)\right) \\
&= \det\left(\begin{array}{c|c} M_{\mathcal{C}} & \mathbb{0} \\ \hline \mathbb{0} & L_G(n) \end{array}\right) \\
&= \kappa(G)^2. \qquad \square
\end{aligned}$$

If we consider a planar graph for which the set of all face cycles contains only cycles, we know that removing one results in a cycle basis. This is very similar to the removal of one vertex in the Laplacian. This is no coincidence as is explored in Section 2.3.2.

2.1.3 An Application

The cycle theorem and the matrix tree theorem form a pair of theorems, which can both be used to find the number of spanning trees in a graph. The cycle theorem requires only a set of rank G independent Eulerian subgraphs. However, it is much more useful to consider a cycle basis, for which we will present two different convenient examples: For planar graphs, this is the face cycle basis, which does not exist for non-planar graphs. In that case we can construct a tree and get the corresponding fundamental cycle basis. Finally, we verify the correctness of the theorem by using a set of unusual and not very practical closed walks.

The underlying graph will always be the sun graph S_3, consisting of six vertices and nine edges, illustrated in Figure 2.2. The arrows indicate the induced edge orientation. We depart from the convention of orienting the edges from the lower to the higher number and instead orient the edges anticlockwise for cycles $\{a,b,c\}$, $\{d,e,f\}$ and $\{g,h,i\}$.

Before we start with using cycles to calculate the number of spanning trees, we calculate it via the matrix tree theorem. This requires us to formulate the 6×6 Laplacian matrix L_{S_3} given in (2.1.5). This gives us $\det(L_{S_3}(6)) = 54$. So there are 54 spanning trees of S_3.

$$L_{S_3} = \begin{pmatrix} 2 & -1 & -1 & & & \\ -1 & 4 & -1 & -1 & -1 & \\ -1 & -1 & 4 & & -1 & -1 \\ & -1 & & 2 & -1 & \\ & -1 & -1 & -1 & 4 & -1 \\ & & -1 & & -1 & 2 \end{pmatrix} \tag{2.1.5}$$

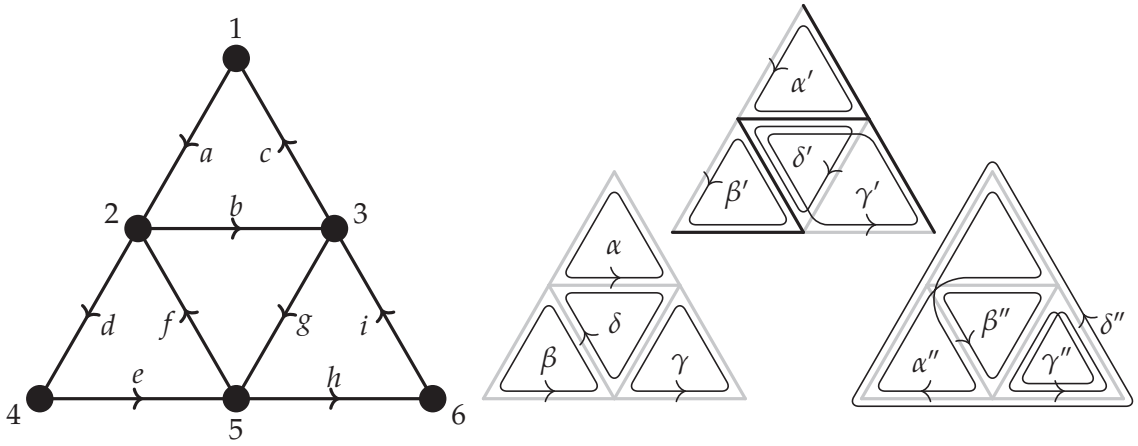


Figure 2.2: The sun graph S_3 with three sets of independent covering closed walks.

Example 2.1.19. At first we will consider the set $\mathcal{S} = \{\alpha, \beta, \gamma, \delta\}$ consisting of four face cycles. All edges of cycle δ are covered by the other three. As every cycle of a fundamental cycle basis has a unique edge not shared by any other cycle, the set $\mathcal{S}$ cannot form a fundamental cycle basis. However, according to Theorem 2.1.12, it is a cycle basis of G. The oriented cycle–edge incidence matrix $B^C_{\mathcal{S}}$ with the given edge-orientations indicated in Figure 2.2 and its cycle-overlap matrix $M_{\mathcal{S}}$ are

$$B^C_{\mathcal{S}} = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ & 1 & & & & 1 & 1 & & \end{pmatrix}, \qquad M_{\mathcal{S}} = \begin{pmatrix} 3 & & & 1 \\ & 3 & & 1 \\ & & 3 & 1 \\ 1 & 1 & 1 & 3 \end{pmatrix}.$$

As $\mathcal{S}$ is a cycle basis, we do not have to use the more general cycle theorem, but instead can use Theorem 2.1.18 for cycle bases. So the determinant of $M_{\mathcal{S}}$ is the number $\kappa(G)$ of spanning trees and, indeed, $\kappa(G) = \det(M_{\mathcal{S}}) = 54$. ◊

Example 2.1.20. A more general approach would be to find a fundamental cycle basis as every graph has one and they can be found quite quickly by constructing a spanning tree and generating the cycles one by one via adding a single unique edge to the tree and therefore generating exactly one cycle. These cycles then form the fundamental cycle basis given by $\mathcal{S}' = \{\alpha', \beta', \gamma', \delta'\}$; the according tree $T = \{b, c, f, e, i\}$ is marked. $\mathcal{S}'$ closely resembles $\mathcal{S}$ and differs only in $\gamma' = \delta \oplus \gamma$, however there are fundamental cycle bases which only contain a single face cycle.

$$B^C_{\mathcal{S}'} = \begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & -1 & & & & -1 & & 1 & 1 \\ & 1 & & & & 1 & 1 & & \end{pmatrix}, \qquad M_{\mathcal{S}'} = \begin{pmatrix} 3 & & -1 & 1 \\ & 3 & -1 & 1 \\ -1 & -1 & 4 & -2 \\ 1 & 1 & -2 & 3 \end{pmatrix}$$

The oriented cycle–edge incidence matrix $B_{\mathcal{S}'}$ contains two negative signs in the cycle γ' as the cycle moves against the chosen direction of b and f. This is also somewhat evident in $M_{\mathcal{S}'}$ as entries at (α', γ') and (β', γ') are -1. However, this only means that the cycles move in opposite directions. Without any further information it is impossible to discern which cycle moves against the direction of b or f. Cycles γ' and δ' share two edges, but move in opposite directions, thus we have $(M_{\mathcal{S}'})_{\gamma', \delta'} = -2$.

In line with set $\mathcal{S}$, we again could use Theorem 2.1.18 to avoid finding a tree, but as we constructed $\mathcal{S}'$, we can also evaluate $\det(B_{\mathcal{S}'}[\emptyset \,|\, T))$. As this is rather trivial, we also calculate it for a different tree $T' = \{a, c, d, e, i\}$. In both cases the determinant is -1 and so $\kappa(G) = \det(M_{\mathcal{S}'}) = 54$.

$$\det(B_{\mathcal{S}'}(\emptyset|T)) = \begin{vmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{vmatrix} = \det(B_{\mathcal{S}'}(\emptyset|T')) = \begin{vmatrix} 1 & & & \\ & & 1 & \\ -1 & -1 & & 1 \\ 1 & 1 & 1 & \end{vmatrix} = -1$$

◊

Example 2.1.21. The final set $\mathcal{S}'' = \{\alpha'', \beta'', \gamma'', \delta''\}$ is different from the previous two as it does not form a cycle basis. On one hand, the closed walk γ'' is not a cycle as it passes through all of its edges twice and $B_{\mathcal{S}''}$ therefore contains 2 in some places. On the other hand, if we disregard the edge-orientation and only consider whether an edge is in a cycle or not, it follows that $\alpha'' \oplus \beta'' \oplus \delta'' = \gamma''$.

$$B^C_{\mathcal{S}''} = \begin{pmatrix} 1 & 1 & 1 & -1 & -1 & -1 & & & \\ & -1 & & & & -1 & -1 & & \\ & & & & & & 2 & 2 & 2 \\ 1 & & 1 & 1 & 1 & & & 1 & 1 \end{pmatrix}, \qquad M_{\mathcal{S}''} = \begin{pmatrix} 6 & & & \\ & 3 & -2 & \\ & -2 & 12 & 4 \\ & & 4 & 6 \end{pmatrix}$$

From the cycle-overlap matrix $M_{\mathcal{S}''}$ we would assume that α'' has no neighbouring cycles as all non-diagonal entries of the first row are 0. This conclusion would be wrong as we consider the signed overlap and cycles α'' and β'' share two edges, but while they move in the same direction in edge f they move in opposite directions in b. A similar argument follows for α'' and δ''.

The walk γ'' also leads to unexpected results as $(M_{\mathcal{S}''})_{\gamma'',\gamma''} = 12$, even if $|\gamma''| = 6$. This is due to γ'' neighbouring itself as it passes through every edge twice. So all six edges in this cycle share an edge with γ'', and so $6+6 = 12$. This is also reflected in $(M_{\mathcal{S}''})_{\beta'',\gamma''} = -2$ as they only share the single edge h, but twice.

As $\mathcal{S}''$ is no cycle basis, we have to use Theorem 2.1.16. The reason is apparent since the determinant is not ± 1:

$$\det(B_{\mathcal{S}''}(\emptyset|T)) = \begin{vmatrix} 1 & -1 & & \\ & & -1 & \\ & & 2 & 2 \\ 1 & 1 & & 1 \end{vmatrix} = (-1)^{2+3}(-1) \cdot \begin{vmatrix} 1 & -1 & \\ & & 2 \\ 1 & 1 & 1 \end{vmatrix} = -4$$

To get the number of spanning trees in this case, we cannot simply calculate the determinant of the cycle overlap matrix, but have to divide by the square of the correction matrix to get

$$\frac{\det(M_{\mathcal{S}''})}{\det(B_{\mathcal{S}''}(\emptyset|T))^2} = \frac{864}{16} = 54.$$

◊

The previous graph illustrated the use of the cycle theorem, but did not show the great benefits, which result from it, especially for mathematical chemists. To show this, we consider the graph of Benzo[f]azulene illustrated in Figure 2.3, given in [Jan+15]. It consists of three cycles of lengths 5, 7 and 6, which are connected at one edge. Double bonds are indicated, but do not contribute to the number of different trees.

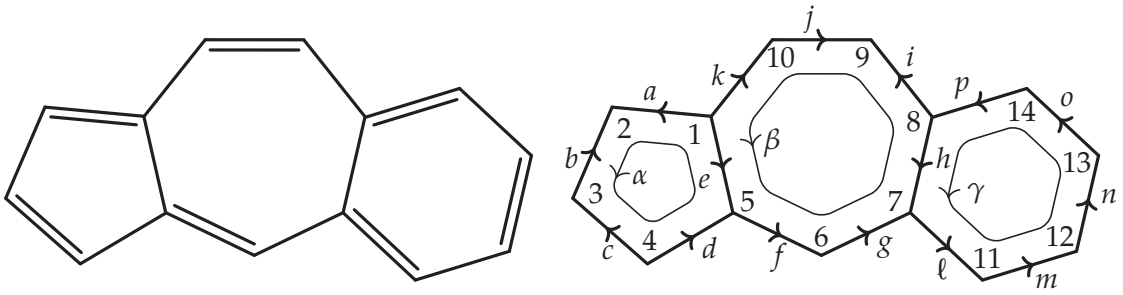


Figure 2.3: The skeletal formula of Benzo[f]azulene and a purely graph theoretical interpretation with arbitrary edge orientations.

After labelling the vertices $1, \dots, 16$, the edges $a, \dots, p$, and the face cycles α, β, and γ we can calculate the number of spanning trees in different ways. Let us at first consider the standard matrix tree theorem. For this, we have to define the oriented vertex–edge incidence matrix of size 14×16. Due to the few branches in the graph this matrix is very sparsely filled and consists of three blocks very similar to B_{C_n} in Example 1.3.2. The sign is dependant on the chosen edge-orientation. Determining the 14×14 Laplacian matrix is much more time-consuming than calculating the cycle-overlap matrix and its determinant. After removing one row and column, we get $\det(L_G(14)) = 199$ spanning trees.

Example 2.1.22. The molecule of Benzo[f]azulene can be depicted as a planar graph as in Figure 2.3, so we can simply choose the face cycles α, β and γ as a cycle basis. The oriented cycle–edge incidence matrix $B_{\mathcal{C}}^{C}$ and the cycle-overlap matrix $M_{\mathcal{C}}$ are given in (2.1.6) and (2.1.7). The number of trees then results from the determinant $\det(M_{\mathcal{C}}) = 199$.

$$B_G^C = \begin{pmatrix} 1 & -1 & -1 & 1 & -1 & & & & & & & & & & & \\ & & & & 1 & 1 & 1 & -1 & 1 & -1 & -1 & & & & & \\ & & & & & & & 1 & & & & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \tag{2.1.6}$$

$$M_{\mathcal{C}} = B_{\mathcal{C}}^{C} B_{\mathcal{C}}^{C\mathrm{T}} = \begin{pmatrix} 5 & -1 & \\ -1 & 7 & -1 \\ & -1 & 6 \end{pmatrix}. \tag{2.1.7}$$

◊

2.2 Counting Combinations

In the last sections, we found different ways to count spanning trees in graphs. Even though they were fundamentally different, all had the luxury of a symmetric Laplacian matrix. This is no coincidence as in a graph every edge can be traversed in both directions. We will now consider so-called *directed graphs*—or *digraphs* for short—in which edges have an orientation. Digraphs were already used in Section 2.1 as the cycles required an edge-orientation. However, we did not consider this orientation while counting the trees.

Counting trees in directed graphs was first done by W. T. Tutte in 1948 [Tut48]. His perspective was completely different and considers the ways to dissect equilateral triangles into smaller equilateral triangles. In his introduction he mentioned that in a previous paper, which is concerned with dissecting rectangles into squares [Bro+40], "the basis of the theory was the association with any rectangle or square dissected into squares of an electrical network obeying Kirchhoff's laws" [Tut48, p. 463] and that he used "an analogue

of the electrical network in which the 'currents' obey laws similar to but not identical with those of Kirchhoff" [Tut48, p. 463]. We will not directly follow his approach, but instead give modern proofs of his generalisation of the matrix tree theorem, which are more in line with other proofs presented in this thesis.

Definition 2.2.1. Let D be a digraph, then the *in-degree* of a vertex $v \in V$ is given by $\deg^-(v) := |\{(w,v) \in E \mid w \in V\}|$, whereas the *out-degree* is defined as $\deg^+(v) := |\{(v,w) \in E \mid w \in V\}|$. So the in-degree counts the incoming edges and the out-degree the outgoing edges.

Definition 2.2.2. A digraph T is an *out-tree* with root $r \in V$ iff T is connected, $\deg^-(r) = 0$, and $\deg^-(v) = 1$ for $r \neq v \in V$, respectively for *in-trees* with root $r \in V$ and the out-degree instead of the in-degree.

Other names for out-trees are *directed rooted tree* or *arborescence*, and another name for an in-tree is *anti-arborescence*. As this naming convention is "out-tree"-focused, we will stick to the more technical, unbiased names. Especially, since we are more interested in in-trees than in out-trees and "anti-arborescence" is much longer than in-tree.

Definition 2.2.3. Let D be a directed graph and $r \in V$ a vertex, then the *number of spanning in-trees rooted at* r is $\kappa_r^-(D)$. Respectively for out-trees and $\kappa_r^+(D)$.

Lemma 2.2.4. *Let T be an in-tree with root r, then there exists exactly one trail from $v \in V$ to r.*

Proof. Any in-tree is connected, so there must exists a trail between v and r, but it cannot start at r as $\deg^+(r) = 0$. So there is a trail from v to r. The out-degree of any non-root vertex is 1 and there is no choice for the next vertex. Thus, there can only be one trail. □

The previous lemma illustrates the motivation of calling such a structure an in-tree. There exists exactly one trail for every vertex that moves into the root. A similar argument can be made for out-trees via reversing all edges, so there exists exactly one trail coming out of the root. Historically, out-trees were studied more closely and were also used in [Cha82], which uses a similar way to proof the matrix tree theorem as in this section.

Definition 2.2.5. Let D be a digraph, then L_D^- is the *in-Laplacian matrix* and L_D^+ the *out-Laplacian matrix*, where

$$(L_D^-)_{ij} := \begin{cases} \deg^+(v_i) & \text{if } i = j, \\ -1 & \text{if } i \neq j, (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

$$(L_D^+)_{ij} := \begin{cases} \deg^-(v_i) & \text{if } i = j, \\ -1 & \text{if } i \neq j, (j,i) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

A drawback of allowing directed edges is that we lose the symmetry of the Laplacian matrix L. Note, however, that $(L_D^-)_{ij} = (L_D^+)_{ji}$ holds for $i \neq j$ as the same edges are counted, but the focus is shifted from the outgoing vertex to the ingoing one. Thus, we cannot find a matrix M, such that $MM^{\mathrm{T}} = L$. Therefore, we require an entirely different approach: The first step is to closer examine the determinant and derive a purely combinatorial understanding of it in this context. This is then used for the inclusion–exclusion principle. Finally, we combine these steps to find an equivalent of the matrix tree theorem for digraphs.

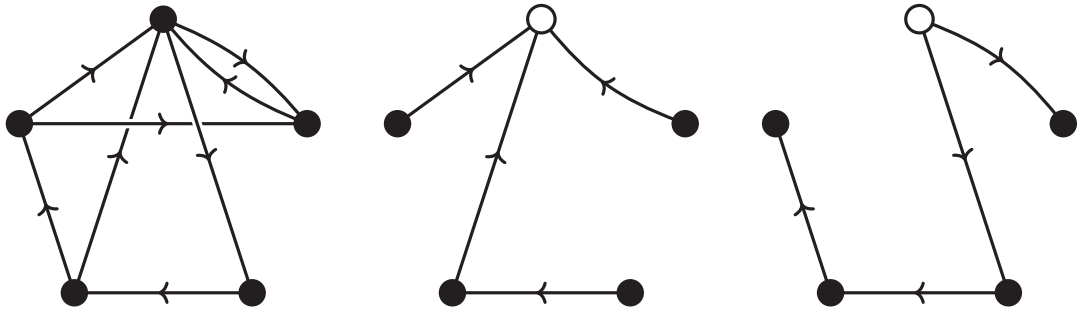


Figure 2.4: A digraph with an in-tree and an out-tree. Roots are marked in white. The lower edge is used in both.

2.2.1 On Counting Functional Graphs

Let us consider the simpler problem of counting the number of possible functions f obeying the edges of a digraph D. In other words, the function f maps the vertices of D onto themselves, but $f(i) = j$ is only allowed if (i, j) is an edge of D.

Definition 2.2.6. A *functional graph* F is a digraph where every vertex v has $\deg^+(v) = 1$.

Definition 2.2.7. Let D be a digraph with $(v, w) \in E$, then a subgraph G of D has a *fixed edge* (v, w) iff $\deg^+(v) = 1$ and $(v, w) \in E_G$. The edge (v, w) is a *forced edge* of G iff it is a fixed edge of $G' = (V, E \cup \{(v, w)\})$.

So a forced edge is possibly a fixed edge of G, but if not, we still allow it. We will use fixed edges—or rather fixed cycles—to identify functional subgraphs, which are not spanning in-trees of the digraph. These unwanted functional subgraphs will be removed from the set of all functional subgraphs and all remaining subgraphs will then be in-trees.

Definition 2.2.8. Let D be a digraph, then $\varphi_S(D)$ is the *number of functional subgraphs* of D with fixed edges S.

Lemma 2.2.9. *Let D be a digraph, then there are $\varphi(D) = \prod_{v \in V(D)} \deg^+(v)$ functional subgraphs of D.*

Proof. As we can choose exactly one outgoing edge for every vertex, we can choose from $\deg^+(v)$ possibilities for every vertex. Thus there are $\prod_{v \in V(D)} \deg^+(v)$ choices. □

Of course, such a functional graph may have cycles. We also want to identify functional subgraphs containing specific cycles, but this is rather simple: If a functional subgraph is required to contain cycle C, the choices are fixed and the vertices in C are removed from the product in Lemma 2.2.9. If C is not in D, then some edge is missing and there are no functional subgraphs containing this cycle.

The ultimate goal is counting spanning in-trees of a digraph D. As the root v of our in-tree does not have an out-degree of 1, but instead 0, we simply ignore the vertex and force an edge between v and a vertex of D. So, similarly to the previous argument about fixed cycles, we have a factor of 1 from the root.

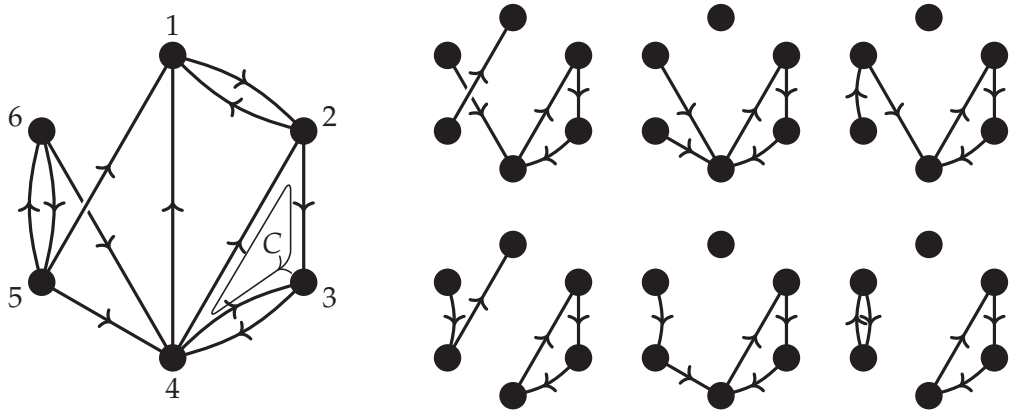


Figure 2.5: A digraph and all functional subgraphs with fixed cycle C and a forced loop $(1,1)$. The lower right subgraph G_6 also contains the cycle $(5,6)$.

Proposition 2.2.10. *Let D be a digraph and S a set of fixed edges, then the number of functional subgraphs of D with fixed edges S is given by*

$$\varphi_S(D) = \begin{cases} \prod_{\substack{v \in V(D), \\ (v,w) \notin S}} \deg^+(v) & \text{if for all } e \in S \text{ also } e \in E(D), \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 2.2.11. *Let D be a digraph. Every in-tree T with root $v \in V$ and forced edge (v,w) is a functional subgraph of D.*

Example 2.2.12. Let D be a digraph with six vertices and a marked root labelled 1. This digraph D is given in Figure 2.5. Let now $C = (2,3,4)$ be a fixed cycle. The number of all functional subgraphs of D with fixed C is then given by $\deg^+(5) \cdot \deg^+(6) = 6$ as the outgoing edges of vertices 2 through 4 are fixed by the cycle C and 1 is fixed arbitrarily since it is the chosen root. Of those subgraphs one also contains the cycle $(5,6)$, and thus would appear again if we fixed this cycle. ◊

2.2.2 Inclusion and Exclusion

If we were to count the spanning in-trees of a digraph D, we could do this by first counting all functional subgraphs with forced edge (v,w). As some of these contain cycles, we have to remove those combinations, which can be counted by Proposition 2.2.10. Example 2.2.12 shows that some functional graphs may be counted twice. So if the sets with one fixed cycle were removed, then the sets with two fixed cycles have to be added again as they were removed twice, and so on for functional graphs with more cycles. We will now introduce the cycle notation of a permutation to formally handle this. This is due to interpreting cycles on vertices as cycles in a permutation, which is the approach taken in [Orl78].

Definition 2.2.13. Let $\pi \in S_n$ be a permutation. The *cycle notation* of π is a set of cycles $(x, \pi(x), \pi^2(x), \dots, \pi^k(x))$, where x is the lexicographically smallest element of a cycle and k is the smallest integer such that $\pi^{k+1}(x) = x$. Trivial cycles of length 1 are omitted. The number $\ell(\pi)$ is the amount of non-trivial cycles of π.

Lemma 2.2.14. *The sign of a permutation π is* $\operatorname{sgn}(\pi) = (-1)^{(k_1-1)+\cdots+(k_{\ell(\pi)}-1)}$.

Definition 2.2.15. Let D be a digraph, then $\varphi_\pi(D)$ is the number of functional subgraphs of D with fixed cycles, which appear as non-trivial cycles of π.

Theorem 2.2.16 (Inclusion–Exclusion Principle)**.** *Let X be a finite set and A_i subsets of X, then*

$$\Big|\bigcup_{i=1}^{n} A_i\Big| = \sum_{\emptyset \neq J \subseteq [n]} (-1)^{|J|+1} \Big|\bigcap_{j\in J} A_j\Big|.$$

Theorem 2.2.17 (Matrix Tree Theorem for Digraphs) [Orl78, Section 5]**.** *Let D be a digraph, then the number $\kappa^-(D)$ of in-trees with root i and forced edge (i,j) is given by*

$$\kappa_i^-(D) = \sum_{\substack{\pi\in S_n,\\ \pi(i)=j}} (-1)^{\ell(\pi)}\varphi_\pi(D) = (-1)^{i+j}\det(L_D^-(i|j)).$$

Proof. Without loss of generality let the forced edge of the root be a loop (n,n) as we could attach the cycle $C = (j,i,i+1,\dots,j-1)$ for $i > j$, and $(i,j,j+1,\dots,i-1)$ otherwise at the end. Relabelling the vertices reduces (j,j) to (n,n). This cycle C introduces a sign -1 if it is of odd length.

The first equation is a direct consequence of the inclusion–exclusion principle as the functional subgraphs containing a single fixed cycle C_i can be interpreted as the sets A_i, and the intersections $A_i \cap A_j$ as the subgraphs the functional subgraphs with the fixed cycles C_i and C_j. In general, the set $\bigcap_{j\in J} A_j$ contains all the functional subgraphs containing fixed cycles C_j with $j \in J$. As n is fixed via a forced loop (n,n), we get

$$\kappa_i^-(D) = \sum_{\pi\in S_{n-1}} (-1)^{\ell(\pi)}\varphi_\pi(D).$$

To connect this formula with the determinant of $L_D^-(n)$, we have to introduce the sign of permutations. However, as the sign of a permutation π is $(-1)^{(k_1-1)+\cdots+(k_{\ell(\pi)}-1)}$ and we implicitly count the number of fixed edges, which is exactly the length of all non-trivial cycles of π, it is possible to write

$$\begin{aligned}\sum_{\pi\in S_{n-1}} (-1)^{\ell(\pi)}\varphi_\pi(D) &= \sum_{\pi\in S_{n-1}} (-1)^{\ell(\pi)+\sum k_i}\varphi_\pi(D)\prod_{e\in\pi}(-1)\\ &= \sum_{\pi\in S_{n-1}} \operatorname{sgn}(\pi)\varphi_\pi(D)\prod_{e\in\pi}(-1)\\ &= \det L_D^-(n).\end{aligned}$$

The last equation is due to Proposition 2.2.10, and the fact that non-diagonal elements of L_D^- contain -1 iff the edge exists. Thus using the Leibniz formula, $\varphi_\pi(D)\prod_{e\in\pi}(-1)$ is equal to $\prod_{i=1}^{n-1}(L_D^-)_{ij}$. □

2.2.3 A Sign-Reversing Involution

In the version of this proof given by S. Chaiken, he did not use the inclusion–exclusion principle. He instead found a function to pair up unwanted functional graphs with each other and annihilate them [Cha82]. In his paper, he proved a more general theorem, which we will encounter in Section 3.1.4. Furthermore, we will also apply his inversion in Section 3.2 to prove a different generalisation on hypergraphs.

Lemma 2.2.18 [Cha82, Section 2]**.** *Let ι be the function, which maps $F \times S_n$ onto itself, where F is the set of functional subgraphs of D with a forced edge from $r \in V$. It is defined by* (2.2.1)*, where C is the lexicographically the smallest cycle in $G \in F$. Then ι is an involution, i.e. $\iota^2 = Id$.*

$$\iota(G,\pi) := \begin{cases} (G,\pi) & \text{if } G \text{ contains only trivial cycles,} \\ (G,\pi \cup \{C\}) & \text{if } C \notin \pi, \\ (G,\pi \setminus \{C\}) & \text{if } C \in \pi. \end{cases} \tag{2.2.1}$$

Proof. Let us first prove that ι is an involution. If G contains no cycles, then $\iota = Id$. On the other hand, if G contains a cycle, then also a lexicographically smallest one C. This cycle is either a cycle of π or it is disjunct from all cycles in π as G is a functional graph. Therefore, if any vertex is in a cycle of π and in C, both cycles are equal since there is only one outgoing edge from every vertex. The function ι is now applied twice, so either C is added and then removed or vice versa. In any case, $\iota^2 = Id$. □

Example 2.2.19. Let us again consider the graph in Figure 2.5. All six functional subgraphs containing C are illustrated on the right. Our previous approach in Section 2.2.2 grouped these together and continued with intersecting this set with others. The approach taken here is different, in such that we now consider a fixed functional graph and try to find all possible ways to fix cycles within it.

Five of these functional subgraphs G_1 through G_5 only contain the single cycle C, and so this cycle can either be fixed or not and simply appear "by chance." As it is the only cycle, it is also the smallest one and the permutation π, which identifies the fixed cycles, may be Id or $\{C\}$. The involution ι therefore maps (G, Id) to $(G, \{C\})$ and vice versa. A way to annihilate both contributions would be to count the number of fixed cycles $\ell(\pi)$.

Counting the number of fixed cycles also works for more than one possible cycle as the sixth functional subgraph G_6 in Figure 2.5 indicates. The cycle $C = (2,3,4)$ is now accompanied by $C' = (5,6)$. It is still the smallest cycle as it contains the smallest vertex of both cycles. The permutations fixing cycles may now choose from $\{C, C'\}$. The involution ι changes now whether C is fixed or not. So if we sum up all possibilities, we get

$$\sum_{\pi \subseteq \{C,C'\}} (-1)^{\ell(\pi)} = \underbrace{(-1)^{\ell(Id)}}_{(G_6,Id)} + \underbrace{(-1)^{\ell(\{C\})}}_{\iota(G_6,Id)} + \underbrace{(-1)^{\ell(\{C'\})}}_{(G_6,\{C'\})} + \underbrace{(-1)^{\ell(\{C,C'\})}}_{\iota(G_6,\{C'\})} = 1-1+1-1 = 0. \ \Diamond$$

Theorem 2.2.20 (Matrix Tree Theorem for Digraphs) [Cha82, Section 3]**.** *Let D be a digraph and the set $\mathcal{F} := \{(G,\pi) \mid G \text{ is a functional subgraph of } D \text{ with cycles } \mathcal{C}, \pi \subseteq \mathcal{C}\}$ of all tuples of functional subgraphs paired with their cycles, then the number $\kappa_i^-(D)$ of in-trees with root i and forced edge (i,j) is given by*

$$\kappa_i^-(D) = \sum_{(G,\pi)\in\mathcal{F}} (-1)^{\ell(\pi)} = (-1)^{i+j} \det(L_D^-(i|j)).$$

Proof. We will now start in a similar manner as in Theorem 2.2.17 and identify the set of all possible ways to generate a functional subgraph of D with forced edge from vertex i. This is given by Proposition 2.2.10. As no cycle is fixed but all possible graphs are chosen, this number is $|F \times \{Id\}|$. This set contains the subset on which ι from Lemma 2.2.18 acts as the identity. We now try to find the other elements and pair them with their image.

Let us consider F and take a closer look at the subset with a fixed subgraph G. If this subgraph contains no cycles, it is an in-tree and we want to preserve it. Otherwise we want to remove it. For this we will use the method of Example 2.2.19 and count the number of non-trivial fixed cycles. So for G containing cycles $\mathcal{C} = \{C_1, \dots, C_k\}$ we get

$$\sum_{\pi \subseteq \mathcal{C}} (-1)^{\ell(\pi)} = 0.$$

Thus only cycle-free functional subgraphs remain and $\kappa_i^-(D) = \sum_{(G,\pi)\in\mathcal{F}}(-1)^{\ell(\pi)}$.

The final step is to connect this to the determinant of $L_D^-(i\,|\,j)$, which is defined by $\sum_{\pi\in S_{n-1}} \operatorname{sgn}(\pi)\prod_{k=1}^{n-1}\ell_{k,\pi(k)}$. The values $\ell_{k,\pi(k)}$ are the number of possible edges for fixed points of π: -1 for every edge in a fixed cycles and 0 if the edge does not exist in D. So the contribution of one pair (G,π) is exactly the sign of π times -1 to the power of vertices in non-trivial cycles. Thus,

$$\operatorname{sgn}(\pi)\prod_{k\in[n],\pi(k)\neq k}(-1) = (-1)^{(k_1-1)+\cdots+(k_{\ell(\pi)}-1)}\cdot(-1)^{k_1+k_2+\cdots+k_{\ell(\pi)}} = (-1)^{\ell(\pi)}.$$

The first equation is due to Lemma 2.2.14. We sum over all pairs $(G,\pi)\in\mathcal{F}$ and so $\det(L_D^-(i|j)) = \sum_{(G,\pi)\in\mathcal{F}}(-1)^{\ell(\pi)}$, which concludes the proof. □

In the above theorem, knowledge about the set $\mathcal{F}$ is usually not required, but—in combination with $\varphi_\pi(D)$ of Theorem 2.2.17—illustrates the difference between both approaches.

All proofs in this section were concerned about the in-trees of a digraph. Similar results exist for out-trees. To transform a proof, the only step is to change the orientation of edges and change L_D^- to L_D^+. Of course, we would not be using functional graphs, but instead graphs with a single in-edge and possibly multiple out-edges.

Theorem 2.2.21 (Matrix Tree Theorem for Out-Trees)**.** *Let D be a digraph, then the number $\kappa^+(D)$ of out-trees with root i and forced edge (j,i) is given by*

$$\kappa_i^+(D) = (-1)^{i+j}\det(L_D^+(j|i)).$$

2.3 Per Induction

A typical way to prove something for countable constructs is to prove it by induction. The matrix tree theorem is no different as M. Lewin showed. His approach can easily be generalised to digraphs, which are used in Section 2.2. However, a much more intuitive generalisation would be to allow multiple edges between two vertices. Intuitive in that sense, that the induction does not care if the smaller already has an edge between two vertices as long as it is a subgraph. So let a *multigraph* be a tuple $G = (V,E)$, where V is the set of vertices and E the multiset of edges. Thus, edges between v and w appear with *multiplicity* $m_{v,w}$.

As a multigraph is a generalisation of graphs, the tools used to describe graphs will be generalised too. The degree $\deg(v)$ of a vertex v counts the adjacent edges including the multiplicity. The generalisation of the Laplacian thus follows directly:

Definition 2.3.1. Let G be a multigraph, then L_G is the *Laplacian matrix* of G, where

$$(L_G)_{ij} := \begin{cases} \deg v_i & \text{if } i = j,\\ -m_{i,j} & \text{if } i\neq j, (i,j)\in E \text{ with multiplicity } m_{i,j}\\ 0 & \text{otherwise.}\end{cases}$$

We start by proving a very narrow version of the matrix tree theorem, which already covers the essence of the proof.

Theorem 2.3.2 [Lew82, Theorem 1]**.** *Let G be a multigraph, then* $\det(L_G(n)) = \kappa(G)$.

Proof. Without loss of generality, ignore loops as they do not contribute to the number of trees. The induction runs on both the number of vertices and edges. Our base case therefore is rather an edge case, in which either G is not connected or contains 2 vertices. There also exists a single (multi-)graph with one vertex, which is also a tree. This coincides with the determinant of the 0×0 matrix. If G is not connected, then it contains no trees and so $\kappa(G) = 0$. If G has 2 vertices, then all edges are between 1 and 2. The Laplacian matrix then is

$$L_G = \begin{pmatrix} m_{1,2} & -m_{1,2} \\ -m_{1,2} & m_{1,2} \end{pmatrix},$$

so the determinants of all minors of L_G are equal. This concludes the induction basis.

The induction step consists of either removing a multiedge or combining two neighbouring vertices into one. Our goal is to describe $\kappa(G)$ via smaller graphs. We split the subtrees of G in those using an edge between i and j and those that do not. Without loss of generality let $i = 1$ and $j = 2$ as this can be ensured by relabelling.

There are $\kappa(G')$ trees not using an edge $(1, 2)$, where G' is G with all edges between 1 and 2 removed, so the induction hypothesis holds. But $L_{G'}$ is very similar to L_G as only $(L_{G'})_{1,1} = (L_G)_{1,1} - m_{1,2}$, $(L_{G'})_{2,2} = (L_G)_{2,2} - m_{1,2}$ and $m'_{1,2} = m'_{2,1} = 0$ are different. If we take a closer look at the determinant of the minor without 1, it is clear that this only affects permutations keeping 2 fixed. This is exactly the determinant of $\det(L_{G'}(1, 2))$ multiplied by $m_{1,2}$, so

$$\kappa(G') = \det(L_{G'}(1)) = \det(L_G(1)) - m_{1,2} \det(L_G(1, 2)).$$

If we consider the trees using an edge between 1 and 2—of which there are $\kappa(G'')$ many—we can contract these vertices into one. All those edges are removed and a new vertex $1'$ is created instead of 1 and 2. As the induction hypothesis holds, we can express $\kappa(G'')$ via the Laplacian. As G'' contracts two vertices into one, but leaves the other part as it is, the Laplacian matrices are only different in $1'$ and 1 and 2 for G'' and G respectively. Therefore

$$\kappa(G'') = \det(L_{G'}(1')) = \det(L_G(1, 2)).$$

Finally, we can take a look at $\kappa(G)$ itself. As any tree of G has to choose at most one edge between 1 and 2 and there are $m_{1,2}$ choices to use an edge and one to not, we can write it as follows:

$$\begin{aligned} \kappa(G) &= \kappa(G') + m_{1,2}\kappa(G'') \\ &= \det(L_G(1)) - m_{1,2} \det(L_G(1, 2)) + m_{1,2} \det(L_G(1, 2)) \\ &= \det(L_G(1)). \end{aligned}$$

□

2.3.1 Tree-Like Structures

In the proof of Theorem 2.3.2 we saw that contracting two vertices works by counting the choices. Any choice corresponds to one edge, which could also be interpreted as a spanning tree of those two vertices. This gives raise to the question if there is a way to count "tree-like" subgraphs containing a fixed structure. We will first consider spanning trees, which contain a fixed tree of a multigraph.

Lemma 2.3.3 [Lew82, Theorem 2]**.** *Let G be a multigraph and T be a tree on vertices $V' \subseteq V$, then the number of spanning trees containing T is $\kappa_T(G) = \det(L_G(V'))$.*

Proof. In a similar manner to the previous proof, we contract the tree in G to a single vertex t of G'. So $\kappa_T(G) = \kappa(T) \cdot \kappa(G')$, but as T is a fixed tree, we have $\kappa(T) = 1$. The number

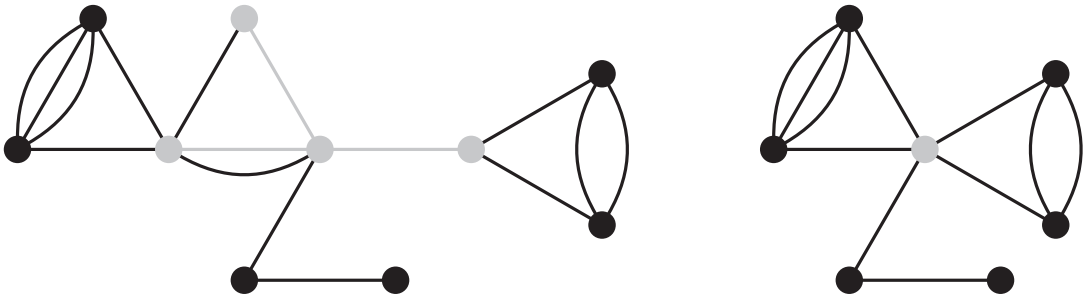


Figure 2.6: A multigraph with a fixed tree in grey. All grey vertices are contracted to a single vertex. On the right: The same graph with the tree contracted.

of spanning trees in G' is the determinant of a minor of $L_{G'}$, so we can choose to remove vertex t. This vertex encompasses exactly all vertices of T and therefore

$$\kappa_T(G) = \det(L_{G'}(t)) = \det(L_G(V')). \qquad \square$$

We can also look at contracting cycles to vertices. If a cycle C of multigraph G is contracted, then $\kappa_C(G)$ counts the pseudotrees with cycle C, where a *pseudotree with cycle C* is a connected subgraph containing only cycle C. In general, a *pseudotree* is a graph, which may contain a single cycle. Thus, every tree is also a pseudotree. Similarly, a *pseudoforest* is a graph, in which every connected component is a pseudotree. The proof follows similarly to the one of the previous lemma.

Lemma 2.3.4. *Let G be a multigraph and C be a cycle on vertices $V' \subseteq V$, then the number of spanning pseudotrees containing C is $\kappa_C(G) = \det(L_G(V'))$.*

Both these results are extremely similar as in that the internal structure of the fixed subgraph is completely lost and ignored. Until now, all edges had to be present in the graph. However, in Section 2.2.1 we introduced forced edges, which were edges added to the graph if not already present and then fixed. We will again use forced edges to count the number of forests consisting of k trees.

Definition 2.3.5. Let G be a graph, then a subgraph F is called *k-forest* iff it has k connected components and every component is a tree.

Theorem 2.3.6 (All Minors Matrix Tree Theorem)**.** *Let G be a connected multigraph, then G contains $\kappa_k(G)$ k-forests with $r_1, \dots, r_k$ in different trees, where*

$$\kappa_k(G) = \det(L_G(r_1, \dots, r_k)).$$

Proof. Let us consider a new graph G' that is similar to G, apart from the forced edges (r_1, r_2),..., (r_{k-1}, r_k) between the roots of the trees. These forced edges form a tree T themselves and G' is connected through T. Therefore Lemma 2.3.3 can be applied and the number of k-forests with $r_1, \dots, r_k$ in different trees is given by

$$\kappa_k(G) = \kappa_T(G') = \det(L_G(r_1, \dots, r_k)). \qquad \square$$

Corollary 2.3.7 [Lew82, Theorem 3]**.** *Let G be a multigraph with k connected components, then G contains $\kappa_k(G)$ spanning forests, where $v_1, \dots, v_k$ are vertices in different connected components and*

$$\kappa_k(G) = \kappa_T(G') = \det(L_G(v_1, \dots, v_k)).$$

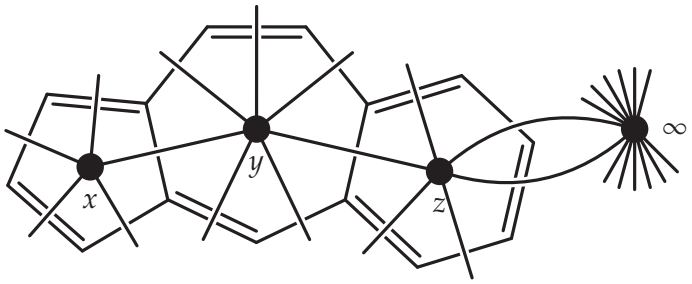


Figure 2.7: Benzo[f]azulene with its dual graph superimposed.

2.3.2 Parallels to the Cycle Theorem

We now return to the cycle theorem of Section 2.1.2, in which we showed a very similar result to the matrix tree theorem. Both results come from Kirchhoff's two circuit laws. Section 2.3.1 shows that the internal structure is lost during contraction of a subgraph S. If the number of spanning trees of S is remembered, we can use it to solve the number of spanning trees by evaluating the partial solutions and combining these. Applying this construction to a planar graph and a face cycle basis would result in following definition.

Definition 2.3.8. Let G be a planar graph and $\mathcal{C}$ be a face cycle basis of G and $C' = \bigoplus_{C \in \mathcal{C}} C$. The dual graph G^* is the graph with vertices $\mathcal{C} \cup \{C'\}$ and edges (C_i, C_j) for every edge in $C_i \cap C_j$.

It is now possible to connect the dual of a given planar graph with the cycle theorem. This is explored implicitly in [Che97, Section 4.4 of Chapter 2].

Theorem 2.3.9. *Let G be a planar graph with n vertices, a cycle rank* $\operatorname{rank} G = k$*, a face cycle basis $\mathcal{C}$, and G^* its dual graph with $k+1$ vertices, then*

$$L_{G^*}(k+1) = M_{\mathcal{C}}.$$

Proof. All elements of the face cycle basis represent a separate, finite face, whereas C' stands for the outer face containing ∞. So every edge of graph G belongs to exactly two face cycles. As the cycles are the vertices of G^*, the overlap of two cycles is a multiedge between those two cycles. This is exactly the definition of the cycle-overlap matrix $M_{\mathcal{C}}$. □

So the Laplacian matrix of the dual graph, of which one vertex is removed, is equal to the cycle-overlap matrix of the face cycle matrix. Thus, we can calculate the number of spanning trees of a planar graph via its dual graph using the cycle theorem. This also shows that Kirchhoff's current law is the dual formulation to Kirchhoff's voltage law and explains why only one law is used in either modern formulation of the matrix tree theorem or the cycle theorem.

Example 2.3.10. We return to Example 2.1.22, in which we calculated the number of spanning trees in a Benzo[f]azulene molecule using the cycle theorem. While it was much quicker than using the matrix tree theorem, it can be sped up even more using the dual graph.

The dual graph in Figure 2.7 contains four vertices, where three are in a finite face and the fourth is at infinity. The outer face cycle, which is labelled with ∞, has a multitude of

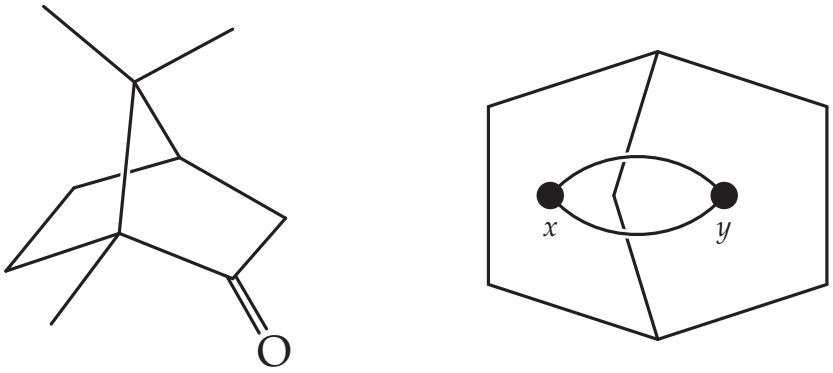


Figure 2.8: The structural formula for a camphor molecule and a planar repesentation without tails. The dual graph is superimposed, with ∞ omitted.

edges: Four edges are connected to x, five to y and five to z—one for every shared edge in the original graph. So the Laplacian matrix is

$$L_{G^*} = \begin{pmatrix} 5 & -1 & & -4 \\ -1 & 7 & -1 & -5 \\ & -1 & 6 & -5 \\ -4 & -5 & -5 & 14 \end{pmatrix}.$$

Thus, the cycle-overlap matrix $M_{\mathcal{C}}$ for the face cycle basis $\mathcal{C}$ is given by removing a row and column from the Laplacian matrix of the dual graph, which corresponds to removing ∞ from the vertices. As before, $\det(L_{G^*}(\infty)) = \det(M_{\mathcal{C}}) = \det(L_G(1)) = 199$. ◊

Example 2.3.11. In the previous example we saw that the dual graph can be used to calculate the number of spanning trees. After removing ∞, this multigraph turned out to be a regular graph. However, this was just coincidence as this example shows.

In contrast to before, the two cycles x and y in Figure 2.8 overlap in two edges. So the dual graph has two edges between x and y. The cycles x and y also each share three edges with ∞, so there are edges with multiplicity 3. The Laplacian matrix of the dual graph is

$$L_{G^*} = \begin{pmatrix} 5 & -2 & -3 \\ -2 & 5 & -3 \\ -3 & -3 & 6 \end{pmatrix}.$$

So we end up with $\det(L_{G^*}(\infty)) = 21$ spanning trees of the camphor molecule. ◊

2.4 Random Walks

All previous proofs came from an algebraic and discrete direction, which is expected as the matrix tree theorem has two concepts from these fields in its name. So it is even more astonishing to find an approach from probability theory using a stochastic process ending up at the same result. The connecting component of graph theory with probability theory is the study of random walks.

2.4.1 Markov Chains

We start by giving a short introduction to Markov chains, which are a simple case of stochastic processes.

Definition 2.4.1. Let Ω be a nonempty set, Σ a σ-algebra on Ω and $\mathbb{P}\colon \Sigma \to [0,1]$ a probability measure. Then the triple $(\Omega, \Sigma, \mathbb{P})$ is called a *probability space*.

Definition 2.4.2. Let $X := (X_i)_{n\geq 0}$ be a sequence of random variables, called *states*, on a finite set V and a probability space $(\Omega, \Sigma, \mathbb{P})$, then X is a *Markov chain* iff the *Markov property*

$$\mathbb{P}(X_{n+1} = x \mid X_0 = x_0, X_2 = x_2, \dots, X_n = x_n) = \mathbb{P}(X_{n+1} = x \mid X_n = x_n)$$

if $\mathbb{P}(X_0 = x_0, X_1 = x_1, \dots, X_n = x_n) > 0$ holds. This property is also sometimes called the *memorylessness* of the Markov chain. In the context of Markov chains, the set Ω is also called *state space*.

Definition 2.4.3. Let X be a Markov chain on set V and let $v, w \in V$, then w is *accessible* from v, written as $v \to w$ iff there exist i and j such that $\mathbb{P}(X_{i+j} = w \mid X_i = v) > 0$.

Definition 2.4.4. A Markov chain X is called *homogenous* iff $\mathbb{P}(X_{i+1} = x \mid X_i = y) = \mathbb{P}(X_{j+1} = x \mid X_j = y)$ for all i, j, x and y. So the conditional probability does not depend on the number of previous steps.

The previous definition enables us to use a *transition matrix* P, where the transition probabilities are $P_{vw} = \mathbb{P}(X_{i+1} = w \mid X_i = v)$, to collect the probabilities of going from vertex v to vertex w. We can now also identify accessibility via powers of the transition matrix, as $v \to w$ iff there is an n, such that $(P^n)_{vw} > 0$. In the following let $\mathbb{P}_x$ be the probability of the chain with $X_0 = x$ and $\mathbb{E}_x$ the expectation according to $\mathbb{P}_x$.

Lemma 2.4.5 (Strong Markov Property)**.** *Let X be a homogenous Markov chain, then the strong Markov property holds:*

$$\mathbb{P}(X_i = x_i : 0 \leq i \leq m+n) = \mathbb{P}(X_i = x_i : 0 \leq i \leq m)\mathbb{P}_{x_m}(X_i = x_i : m+1 \leq i \leq m+n).$$

Proof. As X is homogenous, it holds that

$$\mathbb{P}(X_0 = x_0, \dots, X_{k+1} = x_{k+1}) = \mathbb{P}(X_0 = x_0, \dots, X_k = x_k)P_{x_k x_{k+1}} = \prod_{i=0}^{k} P_{x_i x_{i+1}}.$$

This can be split at any point and the strong Markov property follows. □

Definition 2.4.6. Let X be a Markov chain starting from x and A a subset. The *first hit time* $\tau_A(x) := \inf\{i \mid X_i \in A\}$ is the first time the chain enters set A and the *i-th hit time* $\tau_A^{(i+1)}(x) := \inf\{k > \tau_A^{(i)}(x) \mid X_k \in A, X_{k-1} \notin A\}$ is the i-th time the chain enters the set A. Note that it is acceptable to move within the set without it counting as an extra hit. If $A = \{x\}$ this is also called the *i-th return time*.

The first hit time $\tau_R(x)$ specifies the smallest k, such that $X_k \in R$. So $\tau_R(x) = \infty$ for walks that never enter R. Even a stronger result follows as the next lemma shows.

Lemma 2.4.7 [KRS13, Lemma 3.1]**.** *Let $R \in V$ be accessible from all $v \in V$, then*

$$\max_{v \in V} \mathbb{E}_v(\tau_R(x)) < \infty.$$

Proof. As R is accessible from v, we know that $\mathbb{P}_x(\tau_R(x) > n) \leq \theta_x$. As there are only finitely many different states of the Markov chain, there exists θ and n, such that the previous equation holds for all x. Using the strong Markov property, we can derive

$$\begin{aligned}\mathbb{P}_x(\tau_R(x) > (k+1)n) &= \sum_{y\in V} \mathbb{P}_x(\tau_R(x) > kn, X_{kn} = y)\mathbb{P}_y(\tau_R(x) > n)\\ &\leq \theta\mathbb{P}_x(\tau_R(x) > kn)\end{aligned}$$

and therefore by induction $\mathbb{P}_x(\tau_R(x) > kn) \leq \theta^k$ for all x. □

Definition 2.4.8 [LL10, Section 4.2]**.** For a subset $R \subseteq V$ let g_R be the random walk *Green's function* regarding R, defined by

$$g_R(x,y) := \mathbb{E}_x\Big(\sum_{k=0}^{T_R} \chi_{\{y\}}(X_n)\Big).$$

Definition 2.4.9. Let A be a set, then the *indicator function* χ_A is given by

$$\chi_A(x) := \begin{cases} 1 & \text{if } x \in A,\\ 0 & \text{otherwise.}\end{cases}$$

Theorem 2.4.10 [KRS13, final result of Section 3]**.** *Let P be the transition matrix of a homogenous Markov chain, then the following hold:*

1. *If G is the matrix with $(G(R))_{x,y} = g_R(x,y)$, then $G(R) = (Id - P(R))^{-1}$.*
2. *Let $r_R(x) := \mathbb{P}_x(\tau_x(x) < \tau_R(x))$ be the probability of a random walk starting at x returning to x before hitting R, then*

$$g_R(x,x) = \sum_{k=0}^{\infty} r_R(x)^k = \frac{1}{1 - r_R(x)}.$$

Note that in this theorem we consider the minor of matrix G, which is dependant on the function g_R. Here, one of the drawbacks of using this notation is obvious as one has to be careful whether a function with a parameter or the minor of a matrix is observed.

Proof. The first part is done by evaluating the expectation, which can be rewritten as

$$\begin{aligned}g_R(x,y) &= \mathbb{E}_x\Big(\sum_{k=0}^{\tau_R(x)} \chi_{\{y\}}(X_k)\Big)\\ &= \sum_{k=0}^{\infty} \mathbb{P}_x(X_k = y, \tau_R(x) > k)\\ &= \sum_{k=0}^{\infty} \mathbb{P}_x(X_{\min(k,\tau_R(x))} = y).\end{aligned}$$

The Markov chain $X_R := (X_{\min(k,\tau_R(x))})_{k\geq 0}$ is very similar to $(X_k)_{k\geq 0}$, but as soon as the walk enters R it ends. So the transition matrix for X_R is P_R, where the entries are

$$(P_R)_{x,y} = \begin{cases} P_{x,y} & \text{if } x \notin R,\\ 1 & \text{if } x \in R \text{ and } x = y,\\ 0 & \text{otherwise.}\end{cases}$$

So for $x, y \notin R$ it holds that $\mathbb{P}_x(X_{\min(k,\tau_R(x))} = y) = (P_R^k)_{x,y} = (P(R)^k)_{x,y}$. If we plug this into the sum, we get the geometric series and $G(R) = \sum_{k=0}^{\infty} P(R)^k = (Id - P(R))^{-1}$.

Let $\tau_x^{(i)}(x)$ be the i-th return time for x. For $x \notin R$ it holds that $\tau_x^{(i)}(x) < \tau_R(x)$ iff $\sum_{k=0}^{\tau_R(x)} \chi_{\{x\}}(X_k) > i$. As we can use the strong Markov property and split the chain at the previous visit of x, we can see that

$$\begin{aligned}\mathbb{P}_x(\tau_x^{(i+1)}(x) < \tau_R(x)) &= \mathbb{P}_x(\tau_x^{(i)}(x) < \tau_R(x))\mathbb{P}_x(\tau_x^{(1)}(x) < \tau_R(x)) \\ &= \mathbb{P}_x(\tau_x^{(1)}(x) < \tau_R(x))^{i+1}.\end{aligned}$$

Iteratively applying this result to Green's function, we again get a geometric series and can write it as

$$(Id - P(R))^{-1}_{x,x} = \mathbb{E}_x\Big(\sum_{k=0}^{\tau_R(x)} \chi_{\{y\}}(X_k)\Big) = \frac{1}{1 - \mathbb{P}_x(\tau_x(x) < \tau_R(x))}. \qquad \square$$

We are now able to describe a homogenous Markov chain via its Green's function. This function serves as a bridge between the return times and transition matrix, and will be used later to derive probabilities of paths on a graph G without loops.

2.4.2 Wilson's Algorithm

We now take a closer look at fixed movements on a graph G. This is done by interpreting the state space as the vertices and the transition matrix as the probabilities of moving from one vertex to another. For this we will use so called random walks.

Definition 2.4.11. Let X be a Markov chain, then a *random walk* from x_0 is a sequence $(x_0, \dots, x_n)$, for which $x_0 := X_0$ and $x_i := x_0 + \sum_{k=1}^{i} X_k$.

Definition 2.4.12 [LL10, Section 9.5]**.** Let $\gamma = (x_0, \dots, x_n)$ be a random walk. The *loop erasure* $LE(\gamma) := (x_{i_0}, \dots, x_{i_k})$ of γ is the subsequence of γ with all loops omitted, so $i_0 := 0$, and $i_{j+1} := \max\{i \mid x_i = x_{i_j}\} + 1$.

The loop erasure of a random walk is a subsequence, which has all loops cut away. As every vertex may only appear once, this is a path. Chaining such paths together will give a tree iff one checks whether a previous path is hit. A formalisation of this is exactly Wilson's algorithm [Wil96]:

Algorithm 1 Wilson's algorithm for a uniformly random tree [Wil96, Figure 1]

```
1:  function Wilson(P, V)
2:      T ← ∅
3:      R ← {x_n}                                   ▷ the set of roots
4:      while R ≠ V do
5:          x_0 ← inf(V \ R)          ▷ can also be a random element of V \ R
6:          Y ← LE({X_k | X_0 = x_0, 0 ≤ k ≤ τ_R(x)})
7:          T ← T ∪ {(x_{i_{j-1}}, x_{i_j}) | 0 < j < |Y|}   ▷ add edges of the loop-erased walk
8:          R ← R ∪ Y
9:      end while
10:     return T
11: end function
```

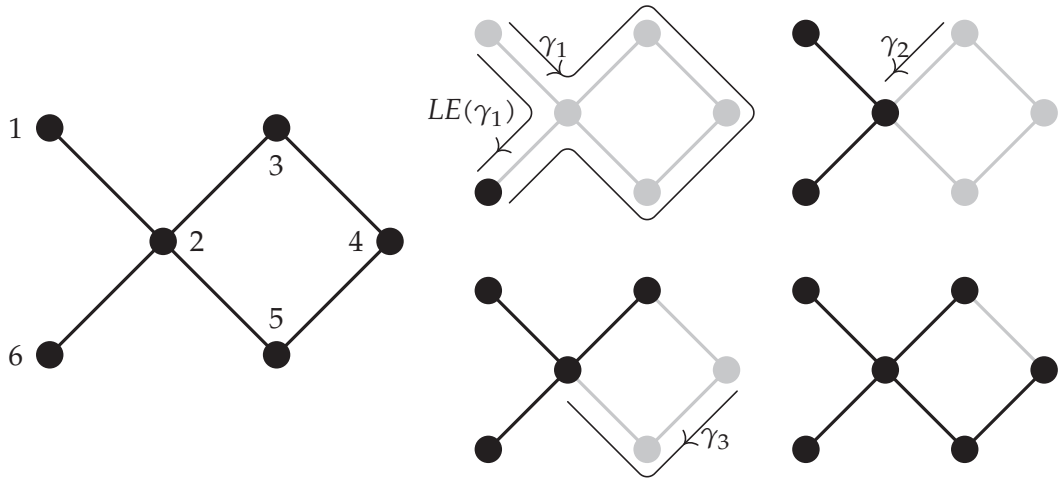


Figure 2.9: A graph and randomly generated spanning tree using Wilson's algorithm.

This algorithm clearly produces a tree if the root x_n is accessible from every vertex. As the main part is "walk, until you hit something", it can also be easily generalised to forests by initialising R with more than one vertex. These vertices will form the roots of the trees.

We have yet to prove that these trees are generated uniformly. In other words, that the probability of being generated is the same for all trees.

Example 2.4.13. In this example we illustrate the way a tree is generated by Wilson's algorithm on the graph of Figure 2.9 on the left. This graph is very simple and the number of trees can be counted by hand: It contains one loop of length four, and therefore all spanning trees omit exactly one edge of this loop. So $\kappa(G) = 4$.

The algorithm begins by initialising $R_1 = \{6\}$ and starting a random walk γ_1 at 1. This walk contains a loop, but this is taken care of by using the loop erasure $LE(\gamma_1)$. The edges of this loop-erased walk are added to T, which will end up as the tree. Finally, $R_2 = R_1 \cup LE(\gamma_1)$ which forms the hitting set for the next iteration of the loop.

As $R_2 \neq V$ there still exist vertices which are not covered. We take the smallest one, which is 3, and start a random walk there. The first step of γ_2 leads into R_2 and the walk ends. The loop-erased walk of γ_2 is equal to it as γ_2 already contains no loops. The single edge of γ_2 is added to T and $R_3 = R_2 \cup \gamma_2$.

Similarly to the previous cycle, $R_3 \neq V$ and there exist uncovered vertices. The walk γ_3 starting at 4 leads to 5 and then into R_3. This walk again contains no loops. So T gets the edges of γ_3 and $R_4 = R_3 \cup \gamma_3$.

The loop ends as $R_4 = V$ and all vertices are covered by an edge. As all loops were erased and the walks ended when hitting a previously hit vertex, the subset T of edges is connected and contains no loop. So T covers all vertices in a connected, loop-less way, thus it is a spanning tree of G. ◊

Lemma 2.4.14 [KRS13, Theorem 2.1]**.** *Let A be a $n \times n$ matrix with non-zero determinant, $\sigma \in S_n$ a permutation and $\Sigma_1 = \emptyset, \Sigma_{i+1} = \Sigma_i \cup \{\sigma(i)\} = \sigma([i])$, then*

$$\det(A)^{-1} = \prod_{k=1}^{n} (A(\Sigma_k))^{-1}_{\sigma(k),\sigma(k)}.$$

Proof. By Cramer's rule, it holds that $(A^{-1})_{i,j} = \frac{\det(A(j|i))}{\det(A)}$ and, for our use, in general $(A(\Sigma_i)^{-1})_{\sigma(i),\sigma(i)} = \frac{\det(A(\Sigma_{i+1}))}{\det(A(\Sigma_i))}$. If this is iteratively combined we end up with a telescoping

product in which only the first and last factor remain, where the determinant of the 0×0 matrix is 1:

$$\prod_{k=1}^{n} (A(\Sigma_k))^{-1}_{\sigma(k),\sigma(k)} = \prod_{k=1}^{n} \frac{\det(A(\Sigma_{k+1}))}{\det(A(\Sigma_k))} = \frac{\det(A([n]))}{\det(A)}. \qquad \square$$

Theorem 2.4.15 [KRS13, Section 5]**.** *Let L_G be the Laplacian matrix of a graph G and T a random spanning tree of G. The probability of generating T via Wilson's algorithm is* $\det(L_G(x_n))^{-1}$.

Proof. The first step is to find the probability of a specific random path, denoted by $\gamma = (x_1, \dots, x_k, x_{k+1})$, starting at x_1 and hitting set R, with x_{k+1} being the first element of γ in R. This probability is obviously 0 if γ contains vertices from different connected components of $V \setminus R$. Without loss of generality, let all $x_i \in \gamma$ be from the same connected component.

We first consider a random walk with hitting set R. The loop erasure of this walk will then be our random path. As loops are cut, many walks generate the same path, and so we have to sum over all these walks. The probability of a given walk is the probability of looping at the first vertex, then moving to the second one, and looping there, without ever returning to the first as this could then be considered as a loop of the first vertex. Thus we define $R^{(1)} := R$ and $R^{(i+1)} := R^{(i)} \cup \{x_i\}$. We then take the return probability r_{R_i} of Theorem 2.4.10 combined with the probability of choosing the edge from x_i to x_{i+1}:

$$\begin{aligned}
\mathbb{P}_R(LE(x_1, \dots, x_{\tau_R(x_1)}) = \gamma) &= \sum_{\ell_1,\dots,\ell_k=0}^{\infty} \prod_{i=1}^{k} r_{R^{(i)}}(x_i)^{\ell_i} P_{x_i,x_{i+1}} \\
&= \prod_{i=1}^{k} \frac{1}{1 - r_{R^{(i)}}(x_i)} \frac{1}{\deg(x_i)} \\
&= \prod_{i=1}^{k} \frac{1}{\deg(x_i)} g_{R^{(i)}}(x_i, x_i).
\end{aligned}$$

A spanning tree T now consists of serveral such random paths, with different starting points and without intersections. Therefore the hitting set increases with every finished path. Let us assume that T is generated via N random walks γ_j with increasing hitting sets $R_1 := \{v_n\}$ and $R_{j+1} = R_j \cup \gamma_j$. Denote the probability of generating T via Wilson's Algorithm by $\mathbb{P}_W(T)$, then

$$\begin{aligned}
\mathbb{P}_W(T) &= \prod_{j=1}^{N} \mathbb{P}_{R_j}(LE(x_{j,1}, \dots, x_{j,\tau_{R_j}(x_1)}) = \gamma_j) \\
&= \prod_{j=1}^{N} \prod_{i=1}^{k_j} \frac{1}{\deg(x_{j,i})} g_{R_j^{(i)}}(x_{j,i}, x_{j,i}), \qquad (2.4.1)
\end{aligned}$$

which was first shown by G. F. Lawler and V. Limic [LL10, Section 9.7]. The $R_j^{(i)}$ are an increasing sequence if ordered first by i and then by j and go from $R_1^{(1)} = \{x_n\}$ to V. We want to apply Theorem 2.4.10.1, so we rewrite $g_{R_j^{(i)}}(x_{j,i}, x_{j,i}) = (G(R_j^{(i)}))$. By Lemma 2.4.14 it follows that

$$\begin{aligned}
\prod_{j=1}^{N} \prod_{i=1}^{k_j} g_{R_j^{(i)}}(x_{j,i}, x_{j,i}) &= \det(G(x_n)) \\
&= (Id - P(x_n))^{-1}.
\end{aligned}$$

As of now, we have just considered the tree generation in the context of a transition matrix P, but the Laplacian matrix L_G and the transition matrix can easily be generated from each other since $L_G(R) = D(R)(Id - P(R))$, where $D = \operatorname{diag}(\deg x_1, \dots, \deg x_n)$. Combining all of this with (2.4.1) leads to

$$\begin{aligned}\mathbb{P}_W(T) &= \frac{\det(G(x_n))}{\det(D(x_n))} \\ &= \frac{1}{\det(D(x_n))\det(Id - P(R))} \\ &= \frac{1}{L_G(x_n)}.\end{aligned}$$

□

Corollary 2.4.16 (Matrix Tree Theorem)**.** *Let G be a graph, then $\det(L_G(n)) = \kappa(G)$.*

Proof. There are exactly $\kappa(G)$ spanning trees of G. Since every tree is generated with the same probability of $\det(L(x_n))^{-1}$, it must hold that $\kappa(G) = \det(L_G(n))$. □

As we were only concerned about the entries of a transition matrix P, the upper proof can be easily modified to allow digraphs, counting the in-trees. As the transition matrix only contains probabilities of moving from one vertex to another, we can also introduce multiple edges by weighing edges differently than others. The usual constraint of counting in whole edges does not apply here. It must be taken into account that the (out-)degrees in D change accordingly. The number of spanning trees then more represents the sum of all weights of spanning trees, where the weight of a spanning tree is the product of the weights of its edges.

Theorem 2.4.17 (Matrix Tree Theorem for Weighted Trees)**.** *Let G be a weighted graph with weights w, then*

$$\det(L_G(n)) = \kappa(G, w),$$

where $\kappa(G, w)$ denotes the sum of weights of all spanning trees of G.

While this theorem could be proven directly with this approach, we will prove a more general version of this weighted matrix tree theorem in Section 3.1 using a method described earlier.

2.5 Close Relatives

Until now, many different approaches to the matrix tree theorem were discussed and most cases allowed us to find small generalisations. As these generalisations often introduce extra steps in the proofs, we reduced them to a minimum, to show the general idea of the proof.

The historical approach provided us with a way to use cycles instead of vertices, which may reduce the size of the required matrices. We later found that this formulation enables us to use the dual graph in finding the number of spanning trees. Explicitly counting led to two ways of proving the theorem. One via the inclusion–exclusion principle and the other via a sign-reversing involution. In either case we were able to generalise the result to digraphs. Induction used the deletion or contraction of subgraphs to reduce the graph to a smaller one. So we were able to force a substructure and derive the theorem for multigraphs and generalise it to the all minors matrix tree theorem. Finally, the probabilistic approach led to using weighted edges and counting not the trees themselves, but the sum of their

weights. If these weights are integers, they can be interpreted as the multiplicity of a multiedge.

Instead of trying to generalise the matrix tree theorem, many tried to modify it and either adapt it to their requirements or simply try to find other related results. This has led to a multitude of similar theorems, with many applications on their own. As they themselves could fill a book, we give only an incomplete list, in which we state the results. The proofs can be found in their respective sources.

2.5.1 A Most Versatile Matrix Tree Theorem

Apart from Section 2.1, every approach highlighted a different generalisation. This most general version of the matrix tree theorem is often considered as the modern formulation. The directed counterpart is often called Tutte's matrix tree theorem as W. T. Tutte first formulated it in 1948 [Tut48].

S. Chaiken and W.-K. Chen both formulated an extremely general version, which will be generalised here to weighted graphs:

Definition 2.5.1. Let G be a (di)graph and $w\colon E \to \mathbb{R}$ a weight function. G with w is then called a *weighted (di)graph*.

The adjacency matrix, Laplacian matrix and the several kinds of incidence matrices can all be generalised to weighted (di)graphs. Instead of marking an edge by writing 1, or in the case of multigraphs, writing the multiplicity m_{ij}, we write the edge weight $w(e)$. In the case of the Laplacian, we have to replace $\deg(v_i)$ with $\sum_{i\in e\in E} w(e)$. If we consider digraphs, we replace $\deg^+(v_i)$ with $\sum_{j\in V} w((i,j))$ as non-existent edges can be considered as edges with weight 0.

Definition 2.5.2. The *weight* of a weighted (di)graph G with weights w is $\prod_{e\in E} w(e)$.

Theorem 2.5.3 (Weighted All Minors Matrix Tree Theorem)**.** *Let G be a weighted (di)graph with n vertices and $I \subseteq \{n\}$, then*

$$\det(L_G(I))$$

is the sum of weights $w(T)$ of spanning (in-)forests T with $|I|$ (in-)trees.

Both got to this theorem in the context of matroid theory, which generalises the concept of linear dependence to arbitrary sets. W.-K. Chen also mentioned the cycle theorem as Corollary 2.24 in [Che97]. S. Chaiken continued to generalise this theorem to gammoids, which are matroids in the context of graphs. However, we will not follow this direction. In general, the path taken in Section 2.1.2 can easily be reformulated to matroids as it already closely resembles them. The cycle theorem is then a result parallel to its dual, using the cut space instead of the cycle space.

In fact, the theorem given above is not a direct generalisation of the theorem S. Chaiken and W.-K. Chen provided as they considered $\det(L_G(I\,|\,J))$ for $I, J \subseteq [n]$. However, this results in a much more complicated expression. We will reach a true generalisation of their theorem later in Section 3.1.4.

2.5.2 The Matrix Tree Theorem on Edges

In Section 1.1 we used the vertex–edge incidence matrix B_G to calculate $L_G = B_G B_G^{\mathrm{T}}$. This results in a matrix, which relates vertices with other vertices. If we reverse the order and

Figure 2.10: The structural formula for n-butane and its corresponding line graph.

calculate $B_G^{\mathrm{T}} B_G$ we end up with a matrix that relates edges with edges. Let this matrix be denoted by K_G. This matrix is dependant on a given edge-orientation.

It cannot be explained as directly as L_G, but shows some interesting relations for special graphs. If G is bipartite, then $K_G = 2Id - A_{G'}$, where G' is the *line graph* of G, which is a similar concept as dual graphs are. The edges of G are the vertices of line graph G' with connecting edges if they share a vertex of G.

Definition 2.5.4 [Wie47, definition of w]**.** Let G be a graph, then the *Wiener index* of G is $W(G)$ defined as the sum of lengths of all shortest paths between two vertices.

Example 2.5.5. H. Wiener calculated in [Wie47] the Wiener index for n-butane and longer straight-chain alkanes explicitly. For these alkanes, the shortest path is quite easy to find as there is only one straight chain of carbon atoms. So for n-butane there is one path of length 3, two paths of length 2 and three paths to a neighbouring vertex of length 1. In sum, the Wiener index of n-butane is $1 \cdot 3 + 2 \cdot 2 + 3 \cdot 1 = 10$.

We generalise this formula for straight-chain alkanes with n carbon atoms. As paths have no direction, we count paths starting at a vertex to all right vertices. So there are no paths for the rightmost vertex, one path of length 1 from the left neighbour, and a sum of $\sum_{i=0}^{k-1} i = \frac{1}{2}(k-1)k$ for the k-th vertex. So for n vertices we get a Wiener index of

$$\sum_{k=0}^{n} \frac{(k-1)k}{2} = \frac{(n-1)n(n+1)}{6}.$$

This is exactly the $(n-1)$-th tetrahedral number. ◊

The Wiener index also appears in the edge version matrix tree theorem if it is applied to a tree. The following theorem is formulated in a more general way in [Mer89].

Theorem 2.5.6 (Edge Version Matrix Tree Theorem) [Mer89]**.** *Let T be a tree and oriented, such that the entries of K_T are non-negative, then*

$$\det K_T(i) = w(e_i),$$

where $w(e_i)$ is the product of the number of vertices on one side of the edge times the ones on the other side.

This theorem can—similarly to the matrix tree theorem—also be formulated via the eigenvalues of the Laplacian. To prove this connection, we first have to define the adjugate of a matrix.

Definition 2.5.7. Let A be an $n \times n$ matrix. The *adjugate* $\operatorname{adj}(A)$ is the transpose of the cofactor matrix. In other words,

$$\operatorname{adj}(A) := \begin{pmatrix} +M_{1,1} & -M_{2,1} & \cdots & (-1)^{n+1} M_{n,1} \\ -M_{1,2} & +M_{2,2} & & \vdots \\ \vdots & & \ddots & -M_{n,n-1} \\ (-1)^{n+1} M_{1,n} & \cdots & -M_{n-1,n} & +M_{n,n} \end{pmatrix},$$

where $M_{i,j} = \det(A(i|j))$.

Lemma 2.5.8. *Let A, B be matrices, then AB and BA have the same eigenvalues with the same multiplicity.*

Proof. Let x be an eigenvector corresponding to eigenvalue λ of AB. Thus, $Ba \neq 0$ and

$$\begin{aligned} \lambda Bx &= B\lambda v \\ &= B(AB)x \\ &= (BA)Bx. \end{aligned}$$

So Bx is an eigenvector corresponding to eigenvalue λ of BA. □

Corollary 2.5.9 [Mer89, Corollary]**.** *Let T be a tree and $\lambda_2, \dots \lambda_n$ the non-zero eigenvalues of L_G. The Wiener index of a tree T is given by*

$$\begin{aligned} W(T) &= \sum_{e \in E} w(e) \\ &= n \sum_{i=2}^{n} \frac{1}{\lambda_i} \end{aligned}$$

Proof. Every edge in T is traversed by every path from the left to the right. So, if there are a vertices to the left and b vertices to the right, then there exist $ab = w(e)$ paths through e. This leads to

$$\begin{aligned} \sum_{e \in E} w(e) &= \sum_{e \in E} \det(K_T(e)) \\ &= \operatorname{tr}(\operatorname{adj}(K_T)) \\ &= n \sum_{i=2}^{n} \frac{\kappa(T)}{\lambda_i}. \end{aligned}$$

The last equation follows as the sum of diagonal elements is the trace of a matrix, and the $K_T(e)$ are the diagonal elements of the adjugate of K_T. In general, if the eigenvalues of a matrix M are $\mu_1, \dots, \mu_n$, then the eigenvalues of the adjugate are $\prod_{i \in [n] \setminus \{k\}} \mu_i$, for $k = 1$ to n. Using Lemma 2.5.8 and the decompositions of K_T and L_T into B_T, in combination with the fact that the product of non-zero eigenvalues of L_T is $\kappa(T)$, we get $\frac{\kappa(T)}{\lambda_i}$ for the eigenvalues of $\operatorname{adj}(L_T)$, but T is a tree, so $\kappa(T) = 1$. □

Example 2.5.10. Let us continue the previous example by calculating the Wiener index via Corollary 2.5.9. The structural formula and the according line graph G' are pictured in Figure 2.10. So the adjacency matrix of G' is

$$A_{G'} = \begin{pmatrix} & 1 & \\ 1 & & 1 \\ & 1 & \end{pmatrix}$$

and $K_G = 2Id + A_{G'}$. Therefore the determinants of the minors $K_G(1), K_G(2)$ and $K_G(3)$ are $3, 4$ and 3 respectively. Calculating the sum gives 10, which is the Wiener index. The other given method of calculating the Wiener index is by using the non-zero eigenvalues of L_G, which are 2 and $2 \pm \sqrt{2}$. So $4 \cdot (\frac{1}{2} + \frac{1}{2+\sqrt{2}} + \frac{1}{2-\sqrt{2}}) = 10$.

$$L_G = \begin{pmatrix} 1 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 1 \end{pmatrix}$$

◊

The Wiener index is of much interest in the chemistry of some alkenes, which are acyclic hydrocarbons. So the restriction of Theorem 2.5.6 to trees is no restriction in its application. H. Wiener himself stated a formula for the boiling point of alkenes in relation to the Wiener index. Later works found many more correlations to the Wiener index [RK02, Table 1].

2.5.3 The Kirchhoff Polynomial

In Section 1.3 we briefly mentioned A. Cayley's original approach of explicitly counting the trees by labelling the vertices. However, in finding the explicit trees it would be more beneficial to directly label the edges as this would give the edges instead of a list of vertices, from which the edges would have to be reconstructed. These lists could then be represented as a polynomial, called Kirchhoff polynomial, in which every monomial represents a tree.

Definition 2.5.11. Let G be a graph and $\mathcal{T}$ the set of all spanning trees on G. The *Kirchhoff polynomial* is given by

$$\mathcal{K}_G := \sum_{T \in \mathcal{T}} \prod_{e_i \in T} x_i.$$

Corollary 2.5.12. *Let G be a graph, then $\kappa(G) = \mathcal{K}_G(1, \dots, 1)$.*

Definition 2.5.13. Let G be a graph, then the combinatorial Laplacian matrix of G is $\mathcal{L}_G$, where

$$(\mathcal{L}_G)_{ij} := \begin{cases} \sum_{i \in e_k \in E} x_k & \text{if } i = j, \\ -x_k & \text{if } i \neq j, e_k = (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

The proof for the matrix tree theorem for the Kirchhoff polynomial is straightforward and can be derived in a similar way to the previous proofs.

Theorem 2.5.14. *Let G be a graph, then $\mathcal{K}_G = \det \mathcal{L}_G(i)$.*

Apart from the use of explicitly listing the spanning trees of a graph via the Kirchhoff polynomial, it found use in quantum theory in solving Feynman integrals on Feynman graphs. This is due to the connection to the first Symanzik polynomial $\mathcal{U}$ [BW10; Bro10], which can be expressed as

$$\mathcal{U}_G(x_1, \dots, x_k) = \mathcal{K}_G\left(\frac{1}{x_1}, \dots, \frac{1}{x_k}\right) \prod_{e_i \in E} x_i.$$

2.5.4 Signless Laplacian Matrix

Another way to modify the known matrix tree theorem is to use a different kind of Laplacian. The classical results all stem from the signed Laplacian matrix $L_G = B_G B_G^{\mathrm{T}}$, where B_G is the oriented incidence matrix. This matrix requires an arbitrary edge-orientation. Getting rid of this choice by simply assigning $+1$ to both ends of an edge gives the *unoriented incidence matrix* $\widetilde{B}_G$, where

$$(\widetilde{B}_G)_{ij} := \begin{cases} 1 & \text{if } (i,k) = e_j, \\ 0 & \text{otherwise.} \end{cases}$$

From this we can derive the signless Laplacian matrix $Q_G := \widetilde{B}_G \widetilde{B}_G^{\mathrm{T}}$. It also holds that

$$(Q_G)_{ij} = \begin{cases} \deg v_i & \text{if } i = j, \\ 1 & \text{if } i \neq j, (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 2.5.15 (Signless Matrix Tree Theorem) [HM19, Theorem 2.9]**.** *Let G be a connected graph and Q_G its signless Laplacian matrix, then $\det(Q_G(i)) = \sum_{H \in \mathcal{P}_o} 4^{c(H)}$, where $\mathcal{P}_o$ is the set of all subgraphs of G with $n-1$ edges, consisting of one tree and $c(H)$ pseudotrees with an odd cycle-length.*

Corollary 2.5.16 [HM19, Corollary 2.11.a]**.** *Let G be a connected graph, then*

$$\det(Q_G(i)) \geq \kappa(G).$$

The regular matrix tree theorem counts the number of trees in a graph, but another interpretation is that it counts the number of "tree-like" subgraphs with $n-1$ edges. This coincides with the number of remaining rows and columns of $L_G(i)$ and, if we consider the all minors matrix tree theorem, which allows the removal of k rows and columns we can also count the number of "tree-like" subgraphs with $n-k$ edges, which are forests. As there are no "tree-like" structures with n edges, we expect $\det(L_G) = 0$, which holds true. However, the same interpretation can be applied to the signless matrix tree theorem. However, this variation counts subgraphs consisting, in which every connected component is either a tree or a pseudotrees with an odd cycle-length. These subgraphs are called *TU-graphs*[5] [HM19]. Thus, we do not expect $\det(Q_G) = 0$, as there are TU-subgraphs with n edges, for example spanning pseudotrees with odd cycle lengths.

Theorem 2.5.17 [HM19, Theorem 3.4]**.** *Let G be a connected graph, then the determinant of Q_G is given by $\det(Q_G) = \sum_{H \in \mathcal{P}_o} 4^{c(H)}$, where $\mathcal{P}_o$ is the set of all subgraphs of G with n edges, consisting of $c(H)$ pseudotrees with an odd cycle-length.*

As a result, we get an upper bound for spanning pseudoforests of G, in which every cycle is of odd length. As every cycle can be extended to a pseudotree, this also gives an upper bound for the number of odd cycles in G.

Corollary 2.5.18 [HM19, Corollary 3.6]**.** *Let G be a graph, then there are at most $\frac{\det(Q_G)}{4}$ spanning pseudoforests of G, in which every cycle is of odd length.*

2.5.5 The Markov Chain Tree Theorem

In Section 2.4 we showed a probabilistic proof of the matrix tree theorem for the minor of the Laplacian matrix, but there is also a version of the theorem via eigenvalues, given in Theorem 1.3.4. The authors of [KRS13] also gave a probabilistic proof for this variant.

In passing, they give a proof of the Markov chain tree theorem, which is concerned about the unique stationary distribution π of a *Q-matrix* $Q := R(Id - P)$, i.e. $\pi Q = 0$ [KRS13, Section 4]:

Definition 2.5.19. Let X be a Markov chain and $x \in V$ an element of the state space. Then x is *recurrent* iff $\mathbb{P}_x(\tau_x < \infty) = 1$.

[5] The name presumably is an abbreviation for "tree or odd unicyclic graph" as pseudotrees, which are not trees, are sometimes called unicyclic.

Lemma 2.5.20. *Let X be a Markov chain and $x, y \in V$ elements of the state space. If x is recurrent and y is accessible from x, then y is also recurrent and x is accessible from y.*

Proof. If x was not accessible from y, but y from x, there exists a path, which enters y before it enters x. Then this path could never return to x and $\mathbb{P}_x(\tau_x < \infty) < 1$. So x is accessible from y, but then any path moving to y is always able to return, thus y is recurrent. □

Theorem 2.5.21 (Markov Chain Tree Theorem) [KRS13, Theorem 4.2]**.** *Let P be a transition matrix, R a diagonal matrix with positive entries, and $Q = R(Id - Q)$ a Q-matrix of a continuous Markov chain. Then the following hold:*

1. $\det(-Q(x)) > 0$ *for all* $x \in V$.
2. *If* $\dim(\ker(Q)) = 1$, *then* $\det(-Q(x]) > 0$ *iff x is recurrent.*
3. *If* $\dim(\ker(Q)) = 1$, *then the unique stationary distribution π with $\sum \pi_x = 1$ is given by*

$$\pi_x = \frac{\det(-Q(x))}{\lambda_2 \cdots \lambda_n},$$

where $\lambda_2, \ldots, \lambda_n$ are the non-zero eigenvalues of $-Q$.

While this is a probabilistic result and the proof given in [KRS13] is also of this nature, there exists also a purely graph theoretical proof [KGW10]. In fact, the Markov chain tree theorem can be proven from the matrix tree theorem and vice versa [PT18].

Expanding the Horizon 3

The matrix tree theorem gives a most central formula in finding the number of spanning trees in a given graph. While the usual approach gives only the number of trees of a graph, this was readily generalisable to give also the number of in-trees of a digraph or multigraph and finally weights of in-trees of a digraph.

We consider two generalisations, which drastically change the matrix tree theorem. In one, we again consider weighted graphs, but now with weights in a commutative semiring. This introduces a very central problem: The notion of determinants requires additive inverses, which are not guaranteed to exist in semirings. The second generalisation is concerned about trees in hypergraphs. The possibility of edges having more than two endpoints restricts us from directly using matrices.

Even though these problems seem very fundamental, much can be salvaged and the general idea of the theorem preserved. We will also give special cases, in which the theorem mostly keeps the known look.

3.1 Semirings

In this section we will generalise the matrix tree theorem to allow graphs with weights in commutative semirings. At first we will give a short introduction to semirings. We continue by defining matrices an a similar notion to determinants on semirings. This mostly follows Chapters 5 and 6 and concludes with Chapter 12 of [Gol03].

Definition 3.1.1. Let S be a non-empty set and $\oplus\colon S \times S \to S$ an associative function, with $a \oplus \varepsilon = \varepsilon \oplus a = a$ for all $a \in S$. Then the triple $(S, \oplus, \varepsilon)$ is called a *monoïd*.

Definition 3.1.2. The tuple $(S, \oplus, \otimes, \varepsilon, e)$ is called *semiring* iff

i. $(S, \oplus, \varepsilon)$ is a commutative monoïd,

ii. $(S, \otimes, e)$ is a monoïd, and

iii. the *multiplication* $\otimes$ is distributive with respect to *addition* $\oplus$, so

$$a \otimes (b \oplus c) := (a \otimes b) \oplus (a \otimes c),$$
$$(a \oplus b) \otimes c := (a \otimes c) \oplus (b \otimes c).$$

In a similar way to omitting the operator $\cdot$ in the usual multiplication, we will also omit $\otimes$ whenever there is little chance of confusing it with a different operation. Additionally, we will often write S instead of $(S, \oplus, \otimes, \varepsilon, e)$ for the semiring.

Definition 3.1.3 [Gol03, p. 14]**.** Let $(S, \oplus, \otimes, \varepsilon, e)$ be a semiring. If for an element $A \in S$ it holds that $a \oplus b = a \oplus c$ iff $b = c$ for all $b, c \in S$, then a is called *cancellable*. If all elements of S are cancellable, then the semiring is *cancellative*.

Example 3.1.4. If we consider the set of non-negative integers $\mathbb{N}$ with the usual addition and multiplication, the tuple $(\mathbb{N}, +, \cdot, 0, 1)$ is a semiring:

$(\mathbb{N}, +, 0)$ is a commutative monoïd as $(a+b)+c = a+(b+c)$, so addition is associative, and 0 is the neutral element. The addition is commutative as $a + b = b + a$.

A similar argument follows for $(\mathbb{N}, \cdot, 1)$. As this too is a commutative monoïd and multiplication is distributive with respect to addition, we even have a commutative semiring.

We can also check whether the cancellative property holds. As 0 is neutral element, it holds that $0 + a = 0 + b$ iff $a = b$. The successor function on $\mathbb{N}$ is injective, so $(c + 1) + a = (c + 1) + b$ iff $(c + a) + 1 = (a + b) + 1$ iff $c + a = c + b$, which holds by induction hypothesis. ◊

Definition 3.1.5 [Gol03, p. 14]**.** Let S be a semiring. An element ∞ is called an *infinite element* iff $a \oplus \infty = \infty$ for all $a \in S$.

The infinite element is necessarily unique since if there are two infinite elements ∞ and ∞', we have $\infty = \infty \oplus \infty' = \infty' \oplus \infty = \infty'$. It also follows directly that infinite elements are cancellative iff the semiring S is trivial. Otherwise there are $a \neq b$ in S, but $a \oplus \infty = \infty = b \oplus \infty$.

Example 3.1.6. In the previous example, we saw that the naturals form a cancellative, commutative semiring in the natural sense. If we add an infinite element ∞ to the naturals, we get $\mathbb{N}_\infty$. The semiring $(\mathbb{N}_\infty, +, \cdot, 0, 1)$ is still commutative, but not cancellative. For example $\infty + 1 = \infty + 0$, but $1 \neq 0$. ◊

Example 3.1.7. Let again $\mathbb{N}_\infty := \mathbb{N} \cup \{\infty\}$ be the naturals with infinity ∞. If we take the minimum $a \oplus b = \min(a, b)$ as the addition and the sum $a \otimes b = a + b$ as multiplication with $\varepsilon = \infty$ and $e = 0$, we get the *tropical semiring* [Gol03, p. 16]. So in this case, the zero element is ∞ and the one element is 0.

This is another common example for a non-cancellative, commutative semiring as $\min(0, 0) = \min(0, 1)$, but $0 \neq 1$. ◊

Example 3.1.8. If a commutative semiring $(S, \oplus, \otimes, \varepsilon, e)$ is given, we can also define the polynomials over S. A polynomial is defined by $p(X) := \bigoplus_{k=0}^{N} s_k X^k$, with $s_k \in S$ and X being an indeterminate. The set of these polynomials is denoted by $S[X]$ and is a semiring on its own, where the zero element is εX^0 and the one element is eX^0. ◊

Definition 3.1.9 [Gol03, p. 5]**.** Let $(S, \oplus, \otimes, \varepsilon, e)$ be a semiring. A non-empty subset $I \subseteq S$ is called a (two-sided)[6] *ideal* iff

i. I is closed under addition,

ii. if $a \in I$ and $b \in S$, then $ab \in I$ and $ba \in I$.

Usually, ideals are defined over rings. The definition does not change visibly as both parts are some closing condition: On one hand, ideals are closed under addition with respect to the ideal itself, and closed under multiplication with respect to the semiring.

Lemma 3.1.10. *Let $(S, \oplus, \otimes, \varepsilon, e)$ be a semiring and I an ideal of S, then $(I, \oplus, \varepsilon)$ is a submonoïd of $(S, \oplus, \varepsilon)$.*

[6] One could also consider one-sided, i.e. left- or right-ideals. However, they will not be required later.

Proof. As any ideal I is non-empty, it contains at least one element a. We can then consider $a\varepsilon = \varepsilon \in I$, due to Definition 3.1.9.ii. So I contains ε and is closed under additions, and it is a submonoïd. □

Definition 3.1.11. Let S be a semiring and $I \subseteq S$ an ideal of S, then $[a]_I$ denotes the set of elements which only differ in an element of I, so

$$[a]_I := \{a \oplus i \mid i \in I\}.$$

Theorem 3.1.12. *Let S be a semiring and $I \subseteq S$ an ideal of S, then $S/_I := \{[a]_I \mid a \in S\}$ with the operations $[a]_I \oplus_I [b]_I := [a \oplus b]_I$ and $[a]_I \otimes_I [b]_I := [a \otimes b]_I$.*

Proof. At first, we show that the operations are well-defined. Let $a, a' \in [a]_I$ and $b, b' \in [b]_I$, then there are i_a and $i_b \in I$, such that $a \oplus i_a = a'$ and $b \oplus i_b = b'$. So $a' \oplus b' = a \oplus i_a \oplus b \oplus i_b = a \oplus b \oplus i$ for some $i \in I$, by Lemma 3.1.10. Thus $\oplus_I$ is well-defined.

Consider now $a' \otimes b' = (a \oplus i_a) \otimes (b \oplus i_b) = ab \oplus (ai_b \oplus i_a b \oplus i_a i_b)$. Due to Definition 3.1.9.ii all summands are in the ideal, and due to Definition 3.1.9.i the sum is too. So $a' \otimes b' \in [a \otimes b]_I$ and $\otimes_I$ is well-defined as well.

The zero element of $S/_I$ is $[\varepsilon]_I = I$ and the one element is $[e]_I$. As $(S, \oplus, \varepsilon)$ is a commutative monoïd, it follows that $(S/_I, \oplus_I, I)$ is too. From $(S, \otimes, e)$ being a monoïd it also follows for $(S/_I, \otimes_I, [e]_I)$. The distributive laws in $S/_I$ too are a direct consequence from the laws in S. □

For any semiring S, the subsets $\{\varepsilon\}$ and S are ideals. But we can derive even more. Let the set of all ideals of S be denoted by $\mathcal{I}(S)$. It is then possible to formulate the following theorem:

Theorem 3.1.13 [Gol03, p. 7]**.** *Let $(S, \oplus, \otimes, \varepsilon, e)$ be a semiring and $\mathcal{I}(S)$ be the set of all ideals of S, then $(\mathcal{I}(S), \oplus_\mathcal{J}, \otimes_\mathcal{J}, \{\varepsilon\}, S)$ is a semiring, where $A \oplus_\mathcal{J} B := \{a \oplus b \mid a \in A, b \in B\}$ and $A \otimes_\mathcal{J} B := \{\bigoplus_{k=1}^{N} a_k \otimes b_k \mid a_k \in A, b_k \in B, N \in \mathbb{N}\}$.*

Proof. We start by showing that $(\mathcal{I}(S), \oplus_\mathcal{J}, \{\varepsilon\})$ is a commutative monoïd. Any ideal A already contains ε, so $A \oplus_\mathcal{J} \{\varepsilon\} = A$ and as $\oplus$ is commutative, so is $\oplus_\mathcal{J}$. What remains is that $C = A \oplus_\mathcal{J} B$ is an ideal. Let $a \oplus b$ and $a' \oplus b'$ be elements of C and $s \in S$, then

$$\begin{aligned}(a \oplus b) \oplus (a' \oplus b') &= (a \oplus a') \oplus (b \oplus b') \in A \oplus_\mathcal{J} B,\\ (a \oplus b) \otimes s &= (a \otimes s) \oplus (b \otimes s) \in A \oplus_\mathcal{J} B,\\ s \otimes (a \oplus b) &= (s \otimes a) \oplus (s \otimes b) \in A \oplus_\mathcal{J} B.\end{aligned}$$

We now show that $(\mathcal{I}(S), \otimes_\mathcal{J}, S)$ is a monoïd. The semiring S is the multiplicative neutral element as every ideal A is already closed under multiplication with respect to S, so $a \otimes s \in A$ for $a \in A$ and $s \in S$. As A is also closed under addition, it holds that $\bigoplus_{k=1}^{N} a_k \otimes s_k \in A$ for $a_k \in A$ and $s_k \in S$. In general, $A \otimes_\mathcal{J} B$ is an ideal as sums can be concatenated and

$$\begin{aligned}\Big(\bigoplus_{k=1}^{N} a_k \otimes b_k\Big) \otimes s &= \bigoplus_{k=1}^{N} a_k \otimes (b_k \otimes s) \in A \oplus_\mathcal{J} B,\\ s \otimes \Big(\bigoplus_{k=1}^{N} a_k \otimes b_k\Big) &= \bigoplus_{k=1}^{N} (s \otimes a_k) \otimes b_k \in A \oplus_\mathcal{J} B.\end{aligned}$$

Only the proof for left-distributiveness is given as right-distributiveness follows from a similar argument. Let A, B and C be ideals of S, then

$$\begin{aligned} A \otimes_{\mathcal{J}} (B \oplus_{\mathcal{J}} C) &= \Big\{ \bigoplus_{k=1}^{N} a_k \otimes (b_k \oplus c_k) \;\Big|\; a_k \in A, b_k \in B, c_k \in C \Big\} \\ &= \Big\{ \bigoplus_{k=1}^{N} (a_k \otimes b_k) \oplus (a_k \otimes c_k) \;\Big|\; a_k \in A, b_k \in B, c_k \in C \Big\} \\ &= (A \otimes_{\mathcal{J}} B) \oplus_{\mathcal{J}} (A \otimes_{\mathcal{J}} C). \end{aligned}$$ □

The ideals of a semiring again form a semiring. However, if the ideals of a ring are considered, nothing is gained and they still only form a semiring. This is due to the absence of additive inverses of ideals.

3.1.1 Matrices

J. S. Golan noted that "matrix semirings over semirings are very closely associated with graph-theoretical problems" [Gol03, p. 64]. As we are in the process of generalising the graph-theorecical problem of counting spanning trees to semirings, this will also be the case here. Therefore we give a small introduction to matrices in this context, although much follows from the known cases of matrices over fields.

The matrix operations will be defined in the usual sense, which is element-wise addition and summation of the product of the column of the first matrix and the row of the second matrix for matrix multiplication. The operations are lifted from the underlying semiring. As we are only interested in square matrices, we will only consider this case. However, the general case can be easily derived. For matrices A and B in the set $\mathcal{M}_n(S)$ of all $n \times n$ matrices with entries in S, it holds that

$$A + B := \begin{pmatrix} a_{11} \oplus b_{11} & \cdots & a_{1n} \oplus b_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} \oplus b_{n1} & \cdots & a_{nn} \oplus b_{nn} \end{pmatrix}$$

$$A \cdot B := \begin{pmatrix} \bigoplus_{k=1}^{n} a_{1k} \otimes b_{k1} & \cdots & \bigoplus_{k=1}^{n} a_{1k} \otimes b_{kn} \\ \vdots & \ddots & \vdots \\ \bigoplus_{k=1}^{n} a_{nk} \otimes b_{k1} & \cdots & \bigoplus_{k=1}^{n} a_{nk} \otimes b_{kn} \end{pmatrix}.$$

Lemma 3.1.14 [Gol03, p. 59]**.** *Let $(S, \oplus, \otimes, \varepsilon, e)$ be a semiring, then the set $\mathcal{M}_n(S)$ of all $n \times n$ matrices is a semiring $(\mathcal{M}_n(S), +, \cdot, \mathbb{0}, Id)$, where $\mathbb{0}$ is the zero element with ε in every place and $Id := \operatorname{diag}(e, \dots, e)$ the one element.*

Proof. $(\mathcal{M}_n(S), +, \mathbb{0})$ is a commutative monoïd as for $A, B \in \mathcal{M}_n(S)$ we have element-wise addition. It also follows that $\mathbb{0}$ must be chosen as the zero element.

Showing that $(\mathcal{M}_n(S), \cdot, Id)$ is a monoïd is a bit more involved, but follows a similar approach. Let c_{ij} be the element of $C = AB$ in the i, j coordinate, then $c_{ij} := \bigoplus_{k=1}^{n} a_{ik} \otimes b_{kj}$. The addition $\oplus$ is commutative, so the order within the sum is irrelevant, however AB is not necessarily the same as BA, which is similar to matrices over fields. Multiplying with Id gives the same matrix, so it can be chosen as the one element.

Finally, $A(B + C) = AB + AC$ can be shown by a straightforward calculation. So multiplication is distributive with respect to addition. □

Lemma 3.1.15 [Gol03, p. 60]**.** *Let $A, B \in \mathcal{M}_n(S)$ a matrix semiring, then $(AB)^{\mathrm{T}} = B^{\mathrm{T}} A^{\mathrm{T}}$.*

3.1.2 Symmetric Extension

While in a semiring S additive inverses may not exist, we want to classify elements, which have some notion of inverses. In general, not all elements will have an inverse. Some elements may already have one, for some others there might be an extension of S, such that there is an inverse in the extension. If this is possible for all elements of S, the semiring S is embeddable into a ring.

Lemma 3.1.16 [Gol03, Proposition 6.1]**.** *Let S be a semiring and I an ideal of S. The set $\mathcal{D}(S,I)$ is a subsemiring of $\mathcal{M}_n(S)$, where*

$$\mathcal{D}(S,I) := \left\{ \begin{pmatrix} a & b \\ b & a \end{pmatrix} \in \mathcal{M}_n(S) \;\middle|\; a \in S, b \in I \right\}.$$

$\mathcal{D}(S,I)$ is a commutative subsemiring iff S is commutative.

Proof. Clearly, $\mathcal{D}(S,I) \subseteq \mathcal{M}_n(S)$ and all that is left to show is that $\mathcal{D}(S,I)$ is a semiring. Every ideal I contains ε, so $\mathbb{0}$ and Id are elements of $\mathcal{D}(S,I)$. Addition is straightforward to show as

$$\begin{pmatrix} a & b \\ b & a \end{pmatrix} = \begin{pmatrix} a & \varepsilon \\ \varepsilon & a \end{pmatrix} + \begin{pmatrix} \varepsilon & b \\ b & \varepsilon \end{pmatrix}$$

and $a \in S$, $b \in I$. As addition is applied element-wise and S and I are closed under it, so is $\mathcal{D}(S,I)$. We now consider the product of two matrices

$$\begin{pmatrix} a & b \\ b & a \end{pmatrix}\begin{pmatrix} c & d \\ d & c \end{pmatrix} = \begin{pmatrix} ac \oplus bd & bc \oplus ad \\ bc \oplus ad & ac \oplus bd \end{pmatrix},$$

for which $ac \oplus bd$ is in S and both bc and ad are the product of an element of S with an element of I, so they must be in I. Since ideals are closed under addition, the sum $bc \oplus ad$ is in I. Therefore $\mathcal{D}(S,I)$ is closed under multiplication.

For S commutative, we can rearrange the elements in $ac \oplus bd$ and $bc \oplus ad$, which, in turn, reverses the order of the matrices in the matrix product. On the other hand, if $ac \oplus bd$ or $bc \oplus ad$ cannot be rearranged in general, the order of matrices is relevant. Thus, $\mathcal{D}(S,I)$ is a commutative subsemiring iff the underlying semiring is commutative. □

In the above proof, it is possible to show that for ideals $I, I' \subseteq S$ the subset $\mathcal{D}(I,I')$ is an ideal of $\mathcal{D}(S,I)$. The argument follows similarly. The only difference is restricting the elements a and c to I instead of S. As ideals are closed under multiplication with respect to the semiring, the products are in I too.

We will now take a closer look at the largest subsemiring of this kind, which is $\mathcal{D}(S,S)$. In general, the subsemirings have the same ordering as the ideals defining them. The smallest possible ideal included in all others is $\{\varepsilon\}$, while the largest ideal is S itself.

Definition 3.1.17 [Gol03, pp. 82–83]**.** Let S be a semiring. Then $\hat{S} := \mathcal{D}(S,S)$ is called the *symmetric extension* of S. Furthermore, we define three subsets on $\hat{S}$:

i. Elements of the form $\begin{pmatrix} a & \varepsilon \\ \varepsilon & a \end{pmatrix}$ are denoted by $\hat{p}_a$ and are called *positive*,

ii. Elements of the form $\begin{pmatrix} \varepsilon & b \\ b & \varepsilon \end{pmatrix}$ are denoted by $\hat{n}_b$ and are called *negative*,

iii. Elements of the form $\begin{pmatrix} a & a \\ a & a \end{pmatrix}$ are denoted by $\hat{h}_a$ and are called *balanced*.

Lemma 3.1.18. *The semiring S is isomorphic to $\mathcal{D}(S, \{\varepsilon\})$.*

Proof. The mapping $\gamma(a) = \hat{p}_a$ for all $a \in S$ is the required isomorphism. It is bijective as all matrices $\hat{p}_a$ are unique and all matrices in $\mathcal{D}(S, \{\varepsilon\})$ are diagonal. The compatibility of the mapping with the operations can be shown via a straightforward calculation. □

The names of the elements in Definition 3.1.17 suggest that a positive element plus its negative results in ε. While this is not true, they do act similarly to positives and negatives as $\hat{p}_a\hat{p}_b = \hat{n}_a\hat{n}_b = \hat{p}_{ab}$ and $\hat{n}_a\hat{p}_b = \hat{p}_a\hat{n}_b = \hat{n}_{ab}$. Also, there is some sense of zero in the symmetric extension. If we take a balanced number and multiply it with another number, it will result in a balanced number, and if we add a positive to its negative, we get $\hat{p}_a + \hat{n}_a = \hat{h}_a$. It would therefore be useful to factor out these balanced numbers.

Lemma 3.1.19 [Gol03, p. 83]**.** *Let S be a semiring, then the set $\mathcal{B}(S)$ of balanced elements of the symmetric extension $\hat{S}$ is an ideal of $\hat{S}$.*

Proof. The zero matrix $\mathbb{0}$ is $\hat{h}_\varepsilon$ and all matrices only have one distinct entry, so the balanced elements are closed under element-wise addition. As every element of $\hat{S}$ can be written as $\hat{p}_a + \hat{n}_b$, it holds that

$$(\hat{p}_a + \hat{n}_b)\hat{h}_c = \hat{p}_a\hat{h}_c + \hat{n}_b\hat{h}_c = \hat{h}_{ac\oplus bc}. \qquad \square$$

Definition 3.1.20 [Gol03, p. 5]**.** Let S be a semiring and $\hat{S}$ its symmetric extension. Then the *Bourne relation* $\equiv$ for the set $\mathcal{B}(S)$ of balanced elements of $\hat{S}$ is given by $\hat{a} \equiv \hat{b}$ iff there exist elements $\hat{h}, \hat{h}' \in \mathcal{B}(S)$ such that $\hat{a} + \hat{h} = \hat{b} + \hat{h}'$.

The Bourne relation was first mentioned by S. Bourne as a generalisation of "the concept of the Jacobson radical […] to arbitrary semirings" [Bou51, p. 164]. This proves quite useful as we can factor a given semiring by the Bourne relation and end up with a ring:

Lemma 3.1.21 [Gol03, Proposition 6.12]**.** *Let S be a cancellative semiring, then it holds that $\hat{p}_a \equiv \hat{p}_b$ iff $a = b$.*

Proof. If $a = b$, then $\hat{p}_a + \mathbb{0} = \hat{p}_b + \mathbb{0}$ and $\hat{p}_a \equiv \hat{p}_b$.

If now $\hat{p}_a \equiv \hat{p}_b$ holds, then $\hat{p}_a + \hat{h}_c = \hat{p}_b + \hat{h}_d$ for some $c, d \in S$. Since positive elements represent diagonal matrices, it must also hold that $\varepsilon \oplus c = \varepsilon \oplus d$ and $a \oplus c = b \oplus d = b \oplus c$. The semiring is cancellative, so $a = b$. □

Lemma 3.1.22 [Gol03, Proposition 6.15]**.** *Let S be a cancellative semiring such that $\equiv$ is not the universal relation, then $\hat{S}/_\equiv$ is a non-trivial ring.*

Proof. Let $\hat{p}_a + \hat{n}_b$ be an element of $\hat{S}$, then $(\hat{p}_a + \hat{n}_b) + (\hat{p}_a + \hat{n}_b)\hat{n}_e = \hat{h}_{a\oplus b}$. If $\equiv$ is taken into account, we get $\hat{h}_{a\oplus b} \equiv \hat{h}_\varepsilon$ and therefore every element has an inverse. □

Corollary 3.1.23 [Gol03, Corollary 6.16]**.** *A non-trivial semiring is cancellative iff it can be embedded into a non-trivial ring.*

Example 3.1.24. Let $\mathbb{N}$ be the semiring of the naturals as in Example 3.1.4. The semiring is cancellative, so by Lemma 3.1.21 we expect that $\hat{p}_a \equiv \hat{p}_b$ iff $a = b$. This can easily be verified by hand. So $\widehat{\mathbb{N}}/_\equiv$ can be embedded into a ring. We can even show that $\widehat{\mathbb{N}}/_\equiv$ is isomorphic to $(\mathbb{Z}, +, \cdot, 0, 1)$. The ideal $\mathcal{B}(\mathbb{N})$ is mapped to 0, while $[\hat{p}_a]_\equiv$ and $[\hat{n}_a]_\equiv$ are mapped to a and $-a$ respectively. ◊

Example 3.1.25. We continue Example 3.1.6, in which we formulated the semiring of positive integers with an infinity element ∞. This is an example for a commutative, but not cancellative semiring.

Let us consider the symmetric extension $\widehat{\mathbb{N}_\infty}$. In this semiring, the balanced elements contains $\hat{h}_\infty$ and it holds that $\infty = a + \infty$ for all $a \in \mathbb{N}_\infty$. This makes the Bourne relation rather uninteresting, since then

$$\begin{pmatrix} a & b \\ b & a \end{pmatrix} + \hat{h}_\infty = \begin{pmatrix} \infty & \infty \\ \infty & \infty \end{pmatrix} = \begin{pmatrix} c & d \\ d & c \end{pmatrix} + \hat{h}_\infty.$$

Thus all elements of $\widehat{\mathbb{N}_\infty}$ are in relation with each other and the ring $\widehat{\mathbb{N}_\infty}/_\equiv$ is the trivial ring with one element $\varepsilon = e$. While unsatisfying, this was to be expected as in any other case we would have to find an inverse for ∞, such that $(a + \infty) - \infty = a$ for all elements in $\widehat{\mathbb{N}_\infty}/_\equiv$. This is not possible if there is more than one element. ◊

The symmetric extension of a semiring proved itself very useful in gaining some notion of inverses. While elements in the shape of $\hat{n}_a$ are not *the* inverses of $\hat{p}_a$, we can still use them like negative elements. The biggest drawback is that $\hat{p}_a + \hat{n}_a \neq \mathbb{0}$ in general. So, in order to allow negatives, we have to give up the uniqueness of the zero element and thus the uniqueness of all elements. In the lucky case of being in a cancellative semiring, we can consider the quotient semiring $S/_\equiv$, which is a ring.

3.1.3 Determinants and Bideterminants

The lack of additive inverses prohibits us in defining the determinant. This is a fundamental problem as the determinant is commonly defined via

$$\det(A) := \sum_{\sigma \in S_n} \left(\operatorname{sgn}(\sigma) \prod_{k=1}^{n} a_{k,\sigma(k)} \right).$$

There exists an alternative to the determinant that omits the sign of the permutation, called the *permanent* of A. This is of no use as it would count something completely different. If we consider the approach in Section 2.2, we can see that the permanent would not remove subgraphs with cycles and, instead, still add them to the count. So, even though the permanent can be defined on semirings, it would not aid in counting spanning trees.

The symmetric extension, however, does help as it allows some sense of negative elements. Using this, we can split the determinant into a positive and a negative part. The positive part consists of the even permutations given in Theorem 1.2.11, and the negative part of all odd permutations.

Definition 3.1.26 [Gol03, pp. 181–182]**.** Let $A \in \mathcal{M}_n(S)$ be an $n \times n$ matrix with entries in a semiring S. The *positive bideterminant* $\det^+(A)$ and *negative bideterminant* $\det^-(A)$ are combined to form the *bideterminant*. They are defined by

$$\begin{aligned} \det^+(A) &:= \bigoplus_{\sigma \in A_n} \bigotimes_{k=1}^{n} a_{k,\sigma(k)}, \\ \det^-(A) &:= \bigoplus_{\sigma \in S_n \setminus A_n} \bigotimes_{k=1}^{n} a_{k,\sigma(k)}, \\ \operatorname{bid}(A) &:= \hat{p}_{\det^+(A)} + \hat{n}_{\det^-(A)}. \end{aligned}$$

The positive and negative bideterminants are elements of S, whereas the bideterminant is an element of the symmetric extension $\hat{S}$. If it is possible to embed the semiring into a ring via the mapping $\gamma(a) = \hat{p}_a$ of Lemma 3.1.18, the bideterminant is mapped to the determinant. The aforementioned permanent $\operatorname{perm}(A)$ can be written as $\hat{p}_{\det^+(A)} + \hat{p}_{\det^-(A)}$ and therefore $\det(A) = \operatorname{perm}(A) \ominus \det^-(A) \ominus \det^-(A)$ in rings.

While we defined bideterminants for all semirings, most properties of determinants are lost if the multiplication is non-commutative. Many attempts have been made to fix this [e.g. Asl96; GR91], but they come with their own drawbacks. Even if we were to preserve some properties, the matrix tree theorem would be generalised for generalisation's sake as the edges of a graph have no distinct order.

Example 3.1.27. As an example of a very fundamental property that is lost, we will calculate the determinants of two matrices over the skew field of the quaternions $\mathbb{H}$. The quaternions are a similar extension of the reals like the complex numbers, with the difference of having three independent imaginary parts i, j and k, for which $i^2 = j^2 = k^2 = -1$, but $ij = k, jk = i$ and $ki = j$. If we reverse the order, the sign is reversed too, so $ji = -k$, $kj = -i$ and $ik = -j$.

$$A = \begin{pmatrix} 1 & i \\ j & k \end{pmatrix}, \quad B = \begin{pmatrix} i & 1 \\ k & j \end{pmatrix}, \quad C = \begin{pmatrix} 1 & ii \\ j & ki \end{pmatrix}, \quad D = \begin{pmatrix} 1i & i \\ ji & k \end{pmatrix}.$$

We now consider the determinant of matrix $A \in \mathcal{M}_2(\mathbb{H})$ and B, which is A with the columns reversed. So $\det(A) = 1k - ji = k + k = 2k$, but $\det(B) = ij - k1 = k - k = 0$. Thus, we cannot swap the order of rows or columns of A.

The determinant also loses its multilinearity as we would expect $\det(C) = \det(D) = \det(A)i$, but instead only $\det(C) = \det(A)i = 2j$ holds by coincidence as the last column is multiplied from the right, which leads to multiplying every product of the determinant with i from the right. Since multiplication is non-commutative and the order of when to multiply with i is different in D, we get $\det(D) = 0$.

At least one aspect of multilinearity is preserved as adding rows or columns can be interchanged with calculating the determinant. This holds due to the distributive laws of the semiring. ◊

We will now prove a series of properties for bideterminants, which hold for determinants and can be generalised to commutative semirings. A few properties also work on semirings, in which case we present that proof.

Proposition 3.1.28 [Gol03, Proposition 12.24]**.** *Let S be a semiring and $A \in \mathcal{M}_n(S)$. If a column of A is ε in every entry, then* $\operatorname{bid}(A) = \hat{h}_\varepsilon$.

Proof. The positive bideterminant is given by $\det^+(A) = \bigoplus_{\sigma \in A_n} \bigotimes_{k=1}^{n} a_{k,\sigma(k)}$. Every product is ε as one element from each column is chosen, but one column consists only of ε. Thus, the sum is ε. A similar argument follows for the negative bideterminant. □

Proposition 3.1.29 [Gol03, Proposition 12.27]**.** *Let S be a semiring and $A \in \mathcal{M}_n(S)$ with the i-th column of A being $a'_i + a''_i$, then*

$$\operatorname{bid}(A) = \operatorname{bid}(a_1, \dots, a'_i, \dots, a_n) + \operatorname{bid}(a_1, \dots, a''_i, \dots, a_n).$$

Proof. We consider a fixed permutation σ. The according product in the bideterminant is

$$\begin{aligned} a_{1,\sigma(1)} \otimes \cdots \otimes (a'_{i,\sigma(i)} \oplus a''_{i,\sigma(i)}) \otimes \cdots \otimes a_{n,\sigma(n)} &= a_{1,\sigma(1)} \otimes \cdots \otimes a'_{i,\sigma(i)} \otimes \cdots \otimes a_{n,\sigma(n)} \\ &\oplus a_{1,\sigma(1)} \otimes \cdots \otimes a''_{i,\sigma(i)} \otimes \cdots \otimes a_{n,\sigma(n)} \end{aligned}$$

due to the distributive laws of S. As this holds for all permutations, we get the desired result. □

Proposition 3.1.30 [Gol03, Proposition 12.26]**.** *Let S be a commutative semiring, $A \in \mathcal{M}_n(S)$ and let A' be similar to A with two columns swapped, then* $\operatorname{bid}(A') = \hat{n}_e \operatorname{bid}(A)$.

Proof. Let the two swapped columns have indices i and j, with $i < j$, $\sigma \in S_n$ be a permutation, and $\sigma' = \sigma \circ (i,j)$ with an additional swap of i and j. The sign of σ' is always different to the sign of σ as (i,j) is a permutation with a single cycle of length 2, so using Lemma 2.2.14, its sign is -1. As S is commutative, we get

$$\begin{aligned} a_{1,\sigma\circ(i,j)(1)} \otimes \cdots \otimes a_{n,\sigma\circ(i,j)(n)} &= a_{1,\sigma'(1)} \otimes \cdots \otimes a_{i,\sigma'(i)} \otimes \cdots \otimes a_{j,\sigma'(j)} \otimes \cdots \otimes a_{n,\sigma'(n)} \\ &= a_{1,\sigma'(1)} \otimes \cdots \otimes a_{j,\sigma'(j)} \otimes \cdots \otimes a_{i,\sigma'(i)} \otimes \cdots \otimes a_{n,\sigma'(n)}. \end{aligned}$$

So for every permutation $\sigma \in A_n$ there exists a permutation $\sigma' \in S_n \setminus A_n$, such that the according summand of $\det^+(A)$ is a summand of $\det^-(A')$. Thus, it holds that $\det^+(A) = \det^-(A')$ and $\det^-(A) = \det^+(A')$ and therefore $\operatorname{bid}(A) = \hat{n}_e \operatorname{bid}(A')$. □

An immediate result from this is that if a matrix A in a commutative semiring S has two columns, which are the same, $\operatorname{bid}(A) \in \mathcal{B}(S)$ holds.

Proposition 3.1.31. *Let S be a commutative semiring, then the bideterminant of S is a multilinear function.*

Proof. To show multilinearity, we have to check whether vector addition and scalar multiplication can be swapped with applying the bideterminant. This means showing

$$\begin{aligned} \operatorname{bid}(a_1, \dots, a_i' + a_i'', \dots, a_n) &= \operatorname{bid}(a_1, \dots, a_i', \dots, a_n) + \operatorname{bid}(a_1, \dots, a_i'', \dots, a_n), \\ \operatorname{bid}(a_1, \dots, ra_i, \dots, a_n) &= \hat{p}_r \operatorname{bid}(a_1, \dots, a_i, \dots, a_n). \end{aligned}$$

The first equation holds in all semirings via Proposition 3.1.29. The second equation holds for commutative semirings as we need to swap places in the multiplication. We proof the equation for the positive bideterminant as the negative is proven in a similar way and it follows for the bideterminant. Let $A' = (a_1, \dots, ra_i, \dots, a_n)$, then

$$\begin{aligned} \det^+(A') &= \bigoplus_{\sigma \in A_n} (a_{k,\sigma(k)} \otimes \cdots \otimes ra_{i,\sigma(i)} \otimes \cdots \otimes a_{n,\sigma(n)}) \\ &= r \bigoplus_{\sigma \in A_n} (a_{k,\sigma(k)} \otimes \cdots \otimes a_{i,\sigma(i)} \otimes \cdots \otimes a_{n,\sigma(n)}) \\ &= r \det^+(A) = \det^+(A) r. \end{aligned}$$

We now know that $\det^+(A') = r \det^+(A)$ and $\det^-(A') = r \det^-(A)$, and therefore $\operatorname{bid}(A') = \hat{p}_r \operatorname{bid}(A)$. So both equations hold and the bideterminant is a multilinear function. □

3.1.4 The Matrix Tree Theorem on Commutative Semirings

The previous subsections did the groundwork for generalising the matrix tree theorem to commutative semirings. We are now able to allow weights in commutative semirings and show that the theorem still holds. It takes on a different look as subtraction is not possible. So the determinant cannot be used. This is why in Section 3.1.3 we formulated an alternative, which does not require subtraction. However, we also use subtraction in a not yet addressed place. The Laplacian matrix is most central in the matrix tree theorem, but is based on the relation $L_G = \operatorname{diag}(\deg(v_i)) - A_G$ in the original sense.

Definition 3.1.32. Let S be a commutative semiring and D a digraph. The pair (D, w) is called a *weighted digraph in S* iff $w\colon E \to S$ is a function, which assigns weights to all edges of D.

The *weight* of a weighted digraph in S is $w(D) \coloneqq \bigotimes_{e\in E} w(e)$.

It is allowed to assign weights of ε and so one could always choose the complete graph K_n as the underlying graph and remove edges from being counted via assigning them a weight of ε. This is mirrored in the following definition.

Definition 3.1.33. Let (D, w) be a weighted digraph in a semiring S, then the *Minoux matrix* M_D is a $2n \times 2n$ matrix, where

$$M_D \coloneqq \left(\begin{array}{c|c} A_D & W \\ \hline Id & Id \end{array}\right), \tag{3.1.1}$$

with $W \coloneqq \operatorname{diag}(\bigoplus_{k=1}^{n} w(1,k), \dots, \bigoplus_{k=1}^{n} w(n,k))$ and A_G being the adjacency matrix, where $(A_D)_{i,j} \coloneqq w(i,j)$, or ε if i and j do not share an edge.

The Minoux matrix separates the parts of diagonal elements and the adjacency matrix as subtraction is not possible. If we were to calculate the Minoux matrix over a ring, we could transform the matrix into

$$M'_D = \left(\begin{array}{c|c} -L_D & A_D \\ \hline \mathbb{0} & Id \end{array}\right), \tag{3.1.2}$$

for which if we were to remove the i-th row and column, we would get $\det(M'_D(i)) = \det(-L_D(i)) \otimes \det(Id) = \det(-L_D(i))$. This looks very much like the matrix tree theorem and is a central goal of this section. M. Minoux himself called this reduced matrix B and defined it slightly different in [Min97, definition of B in Section 3.1], but used the aforementioned arrangement in [Min99, definition of B in Section 3.1].

To reach this goal, we will employ a similar approach used by S. Chaiken, described in Section 2.2.3. We will again find an involution between functional subgraphs with an even or odd number of cycles and what remains are the functional subgraphs with no cycles. The main difference here is that we are not directly counting spanning trees, but instead adding their weights.

Similarly to S. Chaiken in [Cha82], M. Minoux followed the same convention and proved the theorem for out-trees in [Min97]. This is changed here to in-trees, to enable using functional graphs in the proof. This does not substantially change the proof.

We recapitulate some definitions of Section 2.2.1 and reformulate them in the context of weighted digraphs in semirings.

Definition 3.1.34. Let (D, w) be a weighted digraph with $(v, w) \in E$, then a subgraph G of D has a *fixed edge* (v, w) iff $A_{v,v} = w(v, w)$ and $(v, w) \in E_G$. The edge (v, w) is a *forced edge* of G iff it is a fixed edge of $(V, E \cup \{(v, w)\})$ and $w(v, w) = e$.

Definition 3.1.35. Let (D, w) be a digraph, then $\varphi_F(D, w)$ is the *weight* of functional subgraphs of D with fixed edges F.

Lemma 3.1.36. *Let (D, w) be a weighted digraph, then the weight of functional subgraphs is $\varphi(D, w) = \bigotimes_{v\in V} A_{v,v}$.*

Proof. The approach used here is the one of Lemma 2.2.9: The values $A_{v,v}$ consist of sums over the outgoing edges of v. A functional graph chooses exactly one outgoing edge for every vertex $v \in V$. If we iteratively apply the distributive laws to $\bigotimes_{v\in V} A_{v,v}$, we get a

sum with $\prod_{v\in V(D)} \deg^+(v)$ summands. Every one of those represents the weight of an in-tree. □

Definition 3.1.37. Let (D, w) be a weighted digraph, then the *weight* of in-trees rooted at vertex i of D is $\kappa_i^-(D, w)$, and the *weight* of out-trees of D is $\kappa_i^+(D, w)$.

The next step is to restrict the functional graphs as we did in Proposition 2.2.10. This again follows directly from our definition of fixed edges as we consider a subgraph and can apply Lemma 3.1.36 to it. The only thing to do is to reformulate it in the context of the whole graph D.

Proposition 3.1.38. *Let (D, w) be a weighted digraph and F a set of fixed edges, then the weight of functional subgraphs of D with fixed edges F is given by*

$$\varphi_F(D, w) = \begin{cases} \bigotimes_{e\in F} w(e) \otimes \bigotimes_{\substack{v\in V(D),\\ (v,w)\notin F}} A_{v,v} & \text{if for all } e \in F \text{ also } e \in E(D), \\ \varepsilon & \text{otherwise.} \end{cases}$$

Lemma 3.1.39. *Let (D, w) be a weighted digraph, $\sigma \in S_{2n}$ and let $\bigotimes_{i\in[2n]} (M_D)_{i,\sigma(i)} \neq \varepsilon$, then σ is uniquely defined by (the partial permutation) $\sigma(i) = j$ for $i, j \leq n$.*

Proof. We have to show that if we know the fixed values $\sigma(i) = j$ for $i, j \leq n$, we can derive all other elements of this permutation. If $\sigma(i) > n$ for $i \leq n$, then it has to be $\sigma(i) = n + i$ as otherwise the element of M_D would be ε.

For all $i + n$, with $i \leq n$, there are only two possible elements of $(M_D)_{i+n,\sigma(i+n)}$ not equal to ε. One choice is $\sigma(i + n) = i$ and the other is $\sigma(i + n) = i + n$. If $\sigma(i) > n$ and thus $\sigma(i) = i + n$, we have to choose $\sigma(i + n) = i$. Else if $\sigma(i) \leq n$, we choose $\sigma(i + n) = i + n$. The case, in which both i and $i + n$ are already occupied by images of $i \leq n$ can never be relevant as permutations are bijections and thus $(M_D)_{i+n,\sigma(i+n)} = \varepsilon$ and the product evaluates to ε. □

This rather technical lemma states that if the product of the permutation is nonzero, we only require knowledge of the upper left block and the rest follows. So, in reverse, we can consider the partial permutations σ of S_{2n} on $[n]$ and, if we restrict us to "relevant" permutations in the sense of the previous lemma, we get a bijection θ between those permutations and all permutations π on S_n. The bijection is

$$\theta\colon\ \sigma \mapsto \left(i \mapsto \begin{cases} \sigma(i) & \text{for } i \text{ with } \sigma(i) \leq n, \\ \sigma(i) - n & \text{otherwise.} \end{cases} \right). \tag{3.1.3}$$

So it overlays the upper two blocks and ignores the lower blocks. This is similar to the connection between the Minoux matrix and the Laplacian matrix.

Theorem 3.1.40 (Matrix Tree Theorem on Commutative Semirings) [see Min97, Section 3.1]**.** *Let (D, w) be a weighted digraph with weights in a commutative semiring, then*

$$\hat{n}_e^{n-1} \operatorname{bid}(M_D(i)) = \hat{p}_{\kappa_i^-(D,w)} + \hat{h}_{\det^-(M_D(i))}.$$

As we are mostly interested in the weight of the in-trees of D, we could write only the positive part of the main result, which is $\det^+(M_D(i)) = \kappa_i^-(D, w) \oplus \det^-(M_D(i))$. If the semiring can be embedded into a ring, we can rewrite this via the determinant as $\det(M_D(i)) = \det^+(M_D(i)) \ominus \det^-(M_D(i)) = \kappa_i^-(D, w)$. Using the transformations on the Minoux matrix from (3.1.2), we get the standard matrix tree theorem.

Proof. We start by considering the positive bideterminant $\det^+(M_D(i))$. It consists of a sum over permutations, of which we showed in Lemma 3.1.39 that only a small part suffices to identify a unique permutation. So the positive bideterminant can be written as

$$\det^+(M_D(i)) = \bigoplus_{\substack{\sigma\in A_{2n},\\ \sigma(i)=i}} \bigotimes_{k\in[2n]\setminus\{i,i+n\}} (M_D)_{k,\sigma(k)},$$

in which we forced the choice $\sigma(i) = i$. This choice results in a forced $\sigma(i+n) = i+n$ as the $i+n-1$-th row of $M_D(i)$ has the single non-zero entry e in the $i+n-1$-th column.

In fact, the positive determinant of $M_D(i)$ is not quite given by the above formula as $M_D(i)$ is a $2n-1\times 2n-1$ matrix, and thus the permutations are in S_{2n-1}. It is possible to add one fixed point at the end and relabelling the elements closes the gap.

It would be ideal if the sign of σ were the same as $\theta^{-1}(\sigma)$, with θ being the bijection in (3.1.3). Unfortunately, this is not the case. However, the non-trivial cycles of σ are preserved and loops at vertices v are changed to cycles $(v, v+n)$. Thus, to link the positive bideterminant with a sum of weights of functional graphs, we can identify the non-trivial cycles with fixed cycles F in the weight $\varphi_{F,w}(D)$ of functional subgraphs of (D,w). The fixed loop (i,i) is also reflected by forcing the choice of $\theta^{-1}(\sigma)(i) = i$ for all permutations in the bideterminant. So the sign of $\theta^{-1}(\sigma)$ can be expressed via the non-trivial cycles $\mathcal{C}_\pi$ of σ and loops $\mathcal{L}_\pi$, which are split into cycles of length 2. The sign is

$$\begin{aligned}\operatorname{sgn}(\theta^{-1}(\sigma)) &= \prod_{C\in\mathcal{C}_\pi} (-1)^{|C|-1} \prod_{C\in\mathcal{L}_\pi} (-1)\\ &= (-1)^{n-1}(-1)^{\ell(\pi)},\end{aligned}$$

where $\ell(\pi)$ is the number of non-trivial cycles, defined in Definition 2.2.13. The formula holds as σ contains $n-1$ elements and every one either appears in a cycle or loop. We require the bideterminant to rely on the number of non-trivial cycles and have to keep track of where the in-trees are located. This is guaranteed by possibly swapping the positive and negative bideterminant. Applying this to the sum for $n-1$ even leads to

$$\begin{aligned}\det^+(M_D(i)) &= \bigoplus_{\substack{\pi\in\theta(A_{2n}),\\ \pi(i)=i}} \bigotimes_{k\in[2n]\setminus\{i,i+n\}} (M_D)_{k,\theta^{-1}(\pi)(k)}\\ &= \bigoplus_{\substack{\pi\in\theta(A_{2n}),\\ \pi(i)=i}} \varphi_{\pi,w}(D),\end{aligned}$$

where w is the weight function of the digraph, but the forced loop (i,i) has to be considered, so $w(i,i) = e$. A similar result follows for the negative bideterminant, which can be rewritten as

$$\det^-(M_D(i)) = \bigoplus_{\substack{\pi\in\theta(S_{2n}\setminus A_{2n})\\ \pi(i)=i}} \varphi_{\pi,w}(D).$$

If $n-1$ is odd, then a swap between the positive and negative bideterminant has to be considered.

What is left to show is to identify the in-trees in these subsets and show equivalence of the rest. As in-trees do not have any cycles except for the forced loop and the only permutation without non-trivial cycles is the identity $Id \in A_{2n}$, the weights of in-trees are only counted in $\varphi_{Id,w}(D)$.

For all other functional subgraphs a corresponding graph in the sets of the negative bideterminant is required. To prove this, we again utilise the sign-reversing involution ι

of Section 2.2.3. All cycles in a functional graph are disjunct as every vertex has only one outgoing edge. So, as soon as two cycles overlap, they can never separate and must be the same. We take the lexicographically smallest cycle C. The graph and its weight do not change by fixing C, but it changes whether they are counted towards the positive or negative bideterminant as can be seen in Example 2.2.19. A $+1$ in this example can be considered as a contribution to the positive or negative bideterminant and a -1 as a contribution to the other. Thus

$$\det^+(M_D(i)) = \kappa_i^-(D,w) \oplus \det^-(M_D(i)).$$

This solves the positive part of the formula. However, the negative part is rather trivial as ε is the additive neutral element and thus $\det^-(M_D(i)) = \varepsilon \oplus \det^-(M_D(i))$. Combining this with the above results and implementing the swap via $\hat{n}_e^{n-1}$ gives the formula. □

If we compare the matrix tree theorem on commutative semirings to Theorem 2.2.20, we see that it is not a direct generalisation as in semirings we require the Minoux matrix, which is a $2n \times 2n$ matrix, and not the Laplacian matrix. So the minor cannot be directly understood as the graph with one vertex removed. This can be easily fixed as the following corollary shows. This is also the theorem given by M. Minoux.

Corollary 3.1.41 [Min97, Section 3.1]**.** *Let (D,w) be a weighted digraph with weights in a commutative semiring, then*

$$\hat{n}_e^{n-1} \operatorname{bid}(M_D(i, i+n)) = \hat{p}_{\kappa_i^-(D,w)} + \hat{h}_{\det^-(M_D(i,i+n))}.$$

Proof. Forcing the choice $\sigma(i) = i$ in the determinant also results in a forced $\sigma(i+n) = i+n$ as the $i+n-1$-th row of $M_D(i)$ has the single non-zero entry e in the $i+n-1$-th column. So every permutation contains both fixed points, of which the contribution is a multiplicative e, which changes nothing. □

Definition 3.1.42 [Min99, Section 2.3; Cha82, Section 2]**.** Let A and B be two subsets of $[n]$ with equal cardinality. A bijection from A to B is called a *matching*.

The notion of a matching is almost forced upon us, however they are often called *partial permutations*. As we later describe perfect matchings, we choose this name. It should be noted that the name *matching* also defines two different notions, where the other is a set of edges not sharing any vertices.

Clearly, every permutation is a matching as we can choose $A = B = [n]$. So it makes sense to also generalise a few definitions on permutations to matchings. We write $S_{A,B}$ for matchings between A and B.

Usually, matchings are defined in a narrower sense, in which we additionally require that $A \cap B = \emptyset$. This would disallow loops, which are desirable here.

Definition 3.1.43 [Cha82, Section 2]**.** The *sign* of a matching $\pi\colon A \to B$ is defined via its number of *inversions* $\nu(\pi) := \left|\{(i,j) \in A^2 \mid i < j, \pi(i) > \pi(j)\}\right|$ as $\operatorname{sgn}(\pi) := (-1)^{\nu(\pi)}$.

Note that the product formula for the sign given in Lemma 1.2.10 does not hold for matchings. A simple counterexample is the matching mapping 1 to 2 and 4 to 3, for which the formula would give $\frac{1}{3}$. However, the sign of the product still coincides with the sign of the matching.

Definition 3.1.44. A matching $\pi'\colon A' \to B'$ *extends* another matching $\pi\colon A \to B$, written $\pi' \supseteq \pi$ iff $A' \supseteq A$, $B' \supseteq B$ and $\pi'(a) = \pi(a)$ for all $a \in A$.

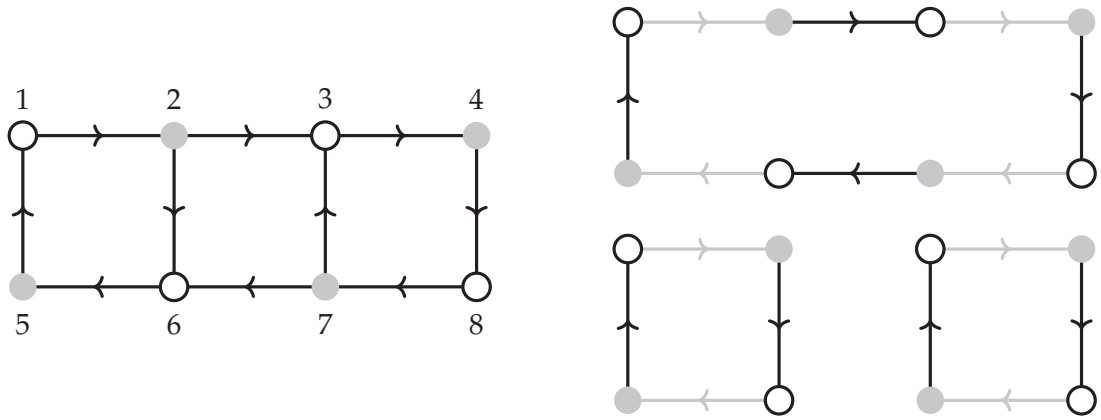


Figure 3.1: A digraph with eight vertices. Its chosen roots are marked in white and vertices, of which every tree must contain exactly one, are marked in grey. The leftmost and rightmost edges are necessarily in-trees in all subforests, but two choices are available for the center, illustrated on the right. The resulting forced edges are marked in grey.

If we have a matching π, which is just missing a single value, there is a unique extension to a full permutation π'. This is not the case with more missing values, since if $i, i' \in A$ and $j, j' \in B$ are not assigned, we can choose either $\pi'(i) = j, \pi'(i') = j'$ or $\pi'(i) = j', \pi'(i') = j$ as extensions. Even worse, these extensions have different signs.

These matchings prove to be very useful in describing permutations with forced edges, which might not be explicitly given, but only given through edges (i, j) of subsets $i \in I$ and $j \in J$. These forced edges are then exactly the missing part of matchings. Up until now, we either considered trees with forcing an edge (i, j) as in Section 2.2, or forests, with interpreting the forced edges as a fixed subgraph as in Section 2.3, which is already solved. So there would have been little to no gain in using matchings to handle forced edges. This changes here as we have to closely follow the sign of a given permutation to decide whether it is in the positive or negative bideterminant.

Furthermore, we will usually force edges, which do not already form a cycle. So there might be forests, whose underlying matchings and permutations have different signs as the following example shows. They will therefore be counted differently.

If we interpret matchings in a set theorist's view as sets of tuples, we can write $\pi \cup \pi'$ for two matchings $\pi\colon A \to B$ and $\pi'\colon A' \to B'$, as long as $A \cap A' = B \cap B' = \emptyset$. This new matching $\pi \cup \pi'$ can then be described via

$$\pi \cup \pi'(i) = \begin{cases} \pi(i) & \text{if } i \in A, \\ \pi'(i) & \text{if } i \in A'. \end{cases}$$

The matching $\pi \cup \pi'$ thus extends both π and π'. We will use this notation to extend matchings to permutations.

Example 3.1.45. We consider the digraph illustrated in Figure 3.1 and try to find subforests, with roots marked in white and vertices, of which every tree must contain exactly one, marked in grey. This configuration has only two different forests, which are easily found by hand. Both are given on the right in Figure 3.1. The corresponding cycle for the upper solution F is one large cycle along the outer edges, whereas the lower solution F' contains two smaller cycles using the inner edges. Thus, the upper forest has a sign $(-1)^7 = -1$ and the lower a sign $(-1)^3(-1)^3 = 1$. We expect them to be in the negative and positive bideterminant, respectively.

However, the bideterminant of the minor of M_D would only be concerned about the matchings from $\{2,4,5,7\}$ to $\{1,3,6,8\}$ as the other vertices (and their indices) do not appear in the reduced matrix. The matchings are

$$\pi_F = \begin{pmatrix} 2 & 4 & 5 & 7 \\ 3 & 8 & 1 & 6 \end{pmatrix}, \qquad \pi_{F'} = \begin{pmatrix} 2 & 4 & 5 & 7 \\ 6 & 8 & 1 & 3 \end{pmatrix}$$

and their signs are $\operatorname{sgn}(\pi_F) = -1$ and $\operatorname{sgn}(\pi_{F'}) = 1$. So, once again, their signs are different. In general, the sign of a matching and one of its extending permutations might be different. ◊

We have seen that it is not possible to find a direct connection between the sign of a matching and its extensions. If we fix the extension, we can find a indirect connection, in which we need the sign of subsets of $[n]$.

Definition 3.1.46. Let $A \subseteq [n]$, then the *number of inversions* of A is

$$\nu(A) := |\{(i,j) \mid i < j, i \in [n] \setminus A, j \in A\}|$$

and the *sign* is $\operatorname{sgn}(A) := (-1)^{\nu(A)}$.

This definition is well-defined even without specifying n as long as $n \geq \max(A)$ since all $k \in [n]$, which are greater than all elements in A, are removed via $i < j$. At first, it might seem counterintuitive to use a notation for permutations, but we can find a bijection between the subsets of $[n]$ and the permutations with n elements. This bijection is identified by the inversions given in Definition 3.1.46.

Example 3.1.47. Let $A = \{1,3,7\}$. To find the sign of A, we have to list all inversions. The largest element of A is 7, so we choose $[7]$ as our basis set. The number of inversions thus is

$$\nu(A) = |\{(2,3),(2,7),(4,7),(5,7),(6,7)\}| = 5.$$

We will now construct the corresponding permutation, which consists of exactly these inversions. We notice that there are no inversions including 1, so $\sigma(1) = 1$. The next integer 2 has two inversions, so $\sigma(2) > \sigma(3)$, $\sigma(2) > \sigma(7)$ and $\sigma(2) > \sigma(1)$ as this would otherwise be counted as an inversion with 1. Thus, $\sigma(2) = 4$. No further inversions contain 3, so it is assigned the next free value $\sigma(3) = 2$. This pattern continues, until we reach the permutation

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 4 & 2 & 5 & 6 & 7 & 3 \end{pmatrix}.$$

The sign of A is -1 and, by construction, so is the sign of σ as we chose the same inversions. ◊

Lemma 3.1.48 [Cha82, Section 2]**.** *Let A be a subset of $[n]$, then the number of inversions of A is given by*

$$\nu(A) = \sum_{a \in A} a - \frac{|A|(|A|+1)}{2}.$$

Proof. Due to $i < j$ in the definition of the inversions, all first elements of the tuples are smaller than the second ones, and all inversions are between A and $\overline{A}$. The first restriction

introduces $j-1$ inversions for $j \in A$. From this, we have to remove all combinations between different elements of A, which is a choice of two different elements of A. So

$$\begin{aligned} \nu(A) &= \sum_{a\in A} (a-1) - \binom{|A|}{2} \\ &= \sum_{a\in A} a - \frac{|A|(|A|+1)}{2}. \end{aligned}$$

□

Proposition 3.1.49 [Min99, Lemma 1; Cha82, Corollary in Section 2]**.** *Let A, B be subsets of $[n]$, and $\pi\colon A \to B$, $\pi' : \overline{A} \to \overline{B}$ be matchings, such that $\pi \cup \pi'$ is a permutation on $[n]$. Then the sign of $\pi \cup \pi'$ is given by*

$$\operatorname{sgn}(\pi \cup \pi') = \operatorname{sgn}(\pi)\operatorname{sgn}(\pi')\operatorname{sgn}(A)\operatorname{sgn}(B). \tag{3.1.4}$$

Proof. The proof consists of disassembling the permutation $\pi \cup \pi'$ into different permutations, for which the signs can be deduced in a straightforward way. At first, we transform π into a permutation on a possibly smaller set $[k]$. Let a_i denote the i-th smallest element of A and b_i the i-th smallest element of B, then π can be written as

$$\pi = \begin{pmatrix} a_1 & a_2 & \cdots & a_k \\ b_{\sigma(1)} & b_{\sigma(2)} & \cdots & b_{\sigma(k)} \end{pmatrix}.$$

The permutation σ has the same inversions of π, but all gaps, which appear in A and B, are removed. This can be imagined as shifting all values of A and B together.

We perform a similar construction for π' and σ'. Instead of shifting the ℓ values of σ' to $[\ell]$, we shift them to $[n] \setminus [k] = \{k+1, \dots, k+\ell\}$. Doing this introduces no further inversions.

So σ and σ' can be combined into one permutation $\sigma \cup \sigma'$, its sign being $\operatorname{sgn}(\sigma)\operatorname{sgn}(\sigma')$. This is, because σ and σ' do not exchange values between each other, and thus no inversions are introduced by combining both. The final step is to reorder the permutations, such that the correct values are mapped via $\pi \cup \pi'$. We can do this by utilising the bijection presented in Example 3.1.47. It follows that the permutation $\pi \cup \pi' = \sigma_B \circ (\sigma \cup \sigma') \circ \sigma_A$, where σ_A and σ_B are the permutations corresponding to A and B respectively.

As the permutation $\sigma \cup \sigma'$ was constructed via respecting the inversions, its sign is $\operatorname{sgn}(\pi)\operatorname{sgn}(\pi')$. Reordering this permutation via σ_A and σ_B introduces their signs and the formula follows. □

Lemma 3.1.50 [Min99, Lemma 2; Cha82, Theorem in Section 2]**.** *Let A, B be subsets of $[n]$, and $\pi\colon A \to B$ be a matching. Let $\pi'\colon \overline{A} \to \overline{B}$ be a matching given by*

$$\pi'(i) := \begin{cases} i & \text{for } i \in \overline{A} \cap \overline{B}, \\ j & \text{if } \pi \text{ contains a path from } j \notin B \text{ to } i \notin A, \end{cases}$$

then the sign of π can be written as

$$\operatorname{sgn}(\pi) = \operatorname{sgn}(\pi')\operatorname{sgn}(A)\operatorname{sgn}(B) \prod_{P \text{ is a path in } \pi} (-1)^{|P|} \prod_{C \text{ is a cycle in } \pi} (-1)^{|C|-1}.$$

Note that π' is uniquely defined, since π is a matching and thus a bijection. If it is now interpreted as a graph, then consists of disconnected cycles and paths. The cycles play no role in π' and the paths each have exactly one element not in B, and one not in A.

Proof. Every path of length $|P|$ of π is completed to a cycle of $\pi \cup \pi'$ of length $|P|+1$ as the only added edge is from the last vertex i to the first vertex j of the path. The sign of a permutation can be expressed via Lemma 2.2.14 and its cycles, so

$$\begin{aligned}\operatorname{sgn}(\pi \cup \pi') &= \prod_{C \text{ is a cycle in } \pi\cup\pi'} (-1)^{|C|-1} \\ &= \prod_{P \text{ is a path in } \pi} (-1)^{|P|} \prod_{C \text{ is a cycle in } \pi} (-1)^{|C|-1}.\end{aligned}$$

Another expression results from applying Proposition 3.1.49. Solving for $\operatorname{sgn}(\pi)$ and plugging in the above formula completes the proof. □

Example 3.1.51. Let π and π' be two matchings, which can be combined to form an extending permutation $\pi \cup \pi'$, defined via

$$\pi = \begin{pmatrix} 1 & 3 & 7 \\ 2 & 1 & 4 \end{pmatrix}, \qquad \pi' = \begin{pmatrix} 2 & 4 & 5 & 6 \\ 5 & 3 & 6 & 7 \end{pmatrix}.$$

The first step is to transform those matchings into permutations on the correct sets as to prepare them for combination. In none of the transformation steps a new inversion is introduced, so the sign does not change.

$$\pi = \begin{pmatrix} 1 & 3 & 7 \\ 2 & 1 & 4 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} = \sigma,$$

$$\pi' = \begin{pmatrix} 2 & 4 & 5 & 6 \\ 5 & 3 & 6 & 7 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 3 & 4 \end{pmatrix} \mapsto \begin{pmatrix} 4 & 5 & 6 & 7 \\ 5 & 4 & 6 & 7 \end{pmatrix} = \sigma'.$$

It remains to calculate the permutations $\sigma_{\{1,3,7\}}$ and $\sigma_{\{1,2,4\}}$. The first permutation is the one appearing in Example 3.1.47. The second one contains only a single inversion $(3,4)$, thus it consists only of the swap of 3 and 4. Combining all those steps leads to

$$\pi \cup \pi' = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 4 & 3 & 5 & 6 & 7 \end{pmatrix} \circ \left(\begin{array}{ccc|cccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 1 & 3 & 5 & 4 & 6 & 7 \end{array}\right) \circ \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 4 & 2 & 5 & 6 & 7 & 3 \end{pmatrix},$$

from which the sign of $\pi \cup \pi'$ can be calculated. The signs of σ_A, σ_B, π and π' are all negative, so the sign of $\sigma \cup \sigma'$ is positive and so is the sign of $\pi \cup \pi'$. ◊

With matchings and a deeper understanding of the relation of their signs we can formulate the all minors matrix tree theorem on commutative semirings. However, as the goal is the most general formulation, allowing in-forests with roots in R and a set C of chosen vertices, from which every in-tree must contain exactly one.

This causes a problem as the in-forests were constructed by forcing edges between roots and chosen vertices. These forced edges can be considered as a matching between R and C. These matchings can have a sign of either $+1$ or -1, which dictates whether they appear in the positive or negative bideterminant, and thus also if the corresponding in-forest is counted positively or negatively. This is not a problem if the in-forest consists of one tree as then $|R| = |C| = 1$ and the only matching has a positive sign.

If the in-forests contain more than one tree, the problem of distributing the in-forests over two subsets, according to the sign of the corresponding matching, arises. However, not all matchings appears as in-forests. An additional requirement is that every tree contains exactly one element of R and C. If there are elements in $R \cap C$, the matching must fix

them as a forced loop. Otherwise, the tree, which contains $r \in R$ and the chosen vertex $c \in R \cap C$, contains two elements in R. Similarly for a tree with root $r \in R \cap C$ and chosen vertex $c \in C$.

These forced loops prove to be very beneficial as, if $R = C$, the only allowed matching is the identity, which has positive sign. Thus, all trees are described by the positive bideterminant. We generalise the definition of $\kappa_i^-(D, w)$ to consider the sign of the matching and allow us to formulate the matrix tree theorem in a more general way.

Definition 3.1.52. Let (D, w) be a weighted digraph, $R \subseteq V$ the set of roots and $C \subseteq V$ a set of chosen vertices with $|R| = |C|$. Then the *weight* of in-forests of D rooted at R, where every in-tree contains exactly one vertex of C, is $\kappa_{R,C}^-(D, w)$, and the *weight* of out-forests is $\kappa_{R,C}^+(D, w)$.

Let also $\kappa_{R,C}^{-\oplus}(D, w)$ and $\kappa_{R,C}^{-\ominus}(D, w)$ be the weight of in-forests of D rooted at R, where every tree contains exactly one vertex of C and the corresponding matching is even, respectively odd. For semirings S, the *signed weight* of all those in-forests is defined as

$$\hat{\kappa}_{R,C}^-(D, w) := \begin{pmatrix} \kappa_{R,C}^{-\oplus}(D, w) & \kappa_{R,C}^{-\ominus}(D, w) \\ \kappa_{R,C}^{-\ominus}(D, w) & \kappa_{R,C}^{-\oplus}(D, w) \end{pmatrix} \in \hat{S}.$$

With this generalised and extended notation the above findings can now be summarised in the following lemma. It will be used to bridge the gap between the general all minors matrix tree theorem and the special cases of Chapter 2.

Lemma 3.1.53. *Let (D, w) be a weighted digraph. If $R = C$ or $|R| = |C| = 1$, then*

$$\kappa_{R,C}^{-\ominus}(D, w) = \varepsilon.$$

Theorem 3.1.54 (All Minors Matrix Tree Theorem on Commutative Semirings)**.** *Let (D, w) be a weighted digraph with weights in a commutative semiring S, $R, C \subseteq [n]$, and $\Delta \in S$, then*

$$\hat{n}_e^{n-|R|+\sum_{i\in R} i+\sum_{j\in C} j} \operatorname{bid}(M_D(R|C)) = \hat{\kappa}_{R,C}^-(D, w) + \hat{h}_\Delta.$$

This is again not exactly the version given in [Min99, Section 3.1] as M. Minoux removed additional rows and columns. The theorem stated in [Min99, Section 3.1] is given as Corollary 3.1.55.

Proof. The idea to this proof is similar to the one of Theorem 3.1.40. The main difference is that not all in-forests have to be counted in the positive bideterminant as can be seen in Example 3.1.45. None the less, we start by taking a closer look at the positive bideterminant. In contrast to the previous proof, we now employ matchings. The previous proof does not benefit from introducing them as the matchings would be permutations with a fixed point, which is not counted. In this case, we remove rows R and columns C, so we can interpret the permutations on $M_D(R|C)$ as matchings on M_D from $R' := [2n] \setminus R$ to $C' := [2n] \setminus C$.

$$\begin{aligned} \det^+(M_D(R|C)) &= \bigoplus_{\pi \in A_{2n-|R|}} \bigotimes_{k=1}^{2n-|R|} (M_D(R|C))_{k,\pi(k)} \\ &= \bigoplus_{\pi' \in A_{R',C'}} \bigotimes_{k \in [2n]\setminus R} (M_D)_{k,\pi'(k)}. \end{aligned}$$

It turns out that many of those matchings do not contribute to the bideterminant as they contain ε in the product. To identify the relevant matchings, we approach the problem

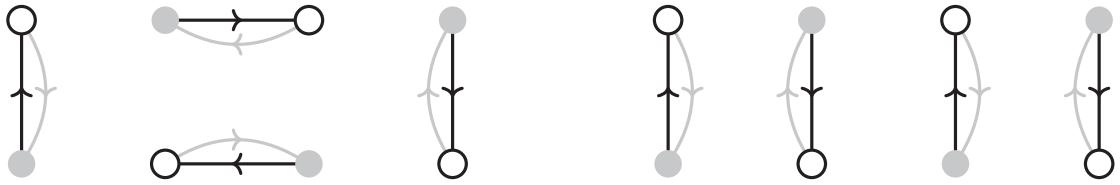


Figure 3.2: The in-forests of Figure 3.1 with forced edges in line with Lemma 3.1.50. The choice of σ does not change the outcome. Vertices in grey are in C and white roots are in R. Grey edges are forced by σ.

from another side. The matching π on $S_{\overline{R},\overline{C}}$ can be interpreted as a functional graph with fixed cycles $\mathcal{C}_\pi$ in $\overline{R} \cap \overline{C}$ and fixed paths $\mathcal{P}$ from $\overline{R}$ to $\overline{C}$. This is a functional graph for a function from $\overline{R}$ to $\overline{C}$. These graphs are extended to functions on $[n]$ via σ. This extension σ is not unique, but as long as it is fixed for a set of fixed paths from $\overline{R}$ to $\overline{C}$, it can be chosen arbitrarily. So it is chosen in line with Lemma 3.1.50. A graphical interpretation of this is loops for roots, which require only themselves as marked vertex in their in-trees, and a forced edge from the root to the chosen vertex otherwise. This always yields legal in-forests and a sign

$$\begin{aligned}\operatorname{sgn}(\pi) &= \operatorname{sgn}(\sigma)\operatorname{sgn}(\overline{R})\operatorname{sgn}(\overline{C}) \prod_{P\in\mathcal{P}_\pi} (-1)^{|P|} \prod_{\substack{Z\in\mathcal{C}_\pi \\ Z \text{ is non-trivial}}} (-1)^{|Z|-1} \prod_{\substack{Z\in\mathcal{C}_\pi \\ Z \text{ is a loop}}} \underbrace{(-1)^{|Z|-1}}_{=(-1)^0} \\ &= \operatorname{sgn}(\sigma)\operatorname{sgn}(\overline{R})\operatorname{sgn}(\overline{C})(-1)^{|\overline{R}|-|\mathcal{C}_\pi|}.\end{aligned}$$

What remains is to show the contribution of θ^{-1}. The bijection keeps most elements the same, but changes fixed points v to cycles $(v, v+n)$. This shows in the sign in the different sign of loops, which leads to the introduction of the number of non-trivial cycles $\ell(\pi)$.

$$\begin{aligned}\operatorname{sgn}(\theta^{-1}(\pi)) &= \operatorname{sgn}(\sigma)\operatorname{sgn}(\overline{R})\operatorname{sgn}(\overline{C}) \prod_{P\in\mathcal{P}_\pi} (-1)^{|P|} \prod_{\substack{Z\in\mathcal{C}_\pi \\ Z \text{ is non-trivial}}} (-1)^{|Z|-1} \prod_{\substack{Z\in\mathcal{C}_\pi \\ Z \text{ is a loop}}} \underbrace{(-1)^{|\theta^{-1}(Z)|-1}}_{=(-1)^1} \\ &= \operatorname{sgn}(\sigma)\operatorname{sgn}(\overline{R})\operatorname{sgn}(\overline{C})(-1)^{|\overline{R}|-\ell(\pi)}.\end{aligned}$$

Matchings $\theta^{-1}(\pi)$ appear in the bideterminant and the sign decides whether a matching is counted towards the positive or negative bideterminant. The contribution of $\overline{R}$ and $\overline{C}$ is the same for all matchings and can be considered fixed. Let us assume that no swap is required as similar steps hold for the other case. So the sign is only dependant on the sign of σ, which is not yet determined, and the number of non-trivial cycles of π. By the definition of $\kappa^{-\oplus}_{R,C}(D,w)$ and $\kappa^{-\ominus}_{R,C}(D,w)$ in Definition 3.1.52, we define σ such that for in-forests $\operatorname{sgn}(\theta^{-1}(\pi)) = \operatorname{sgn}(\pi)$ holds. In-forests do not contain any cycles and so do the corresponding matchings, thus $+1 = \operatorname{sgn}(\pi) = \operatorname{sgn}(\sigma)$ for positive in-forests, and $-1 = \operatorname{sgn}(\pi) = \operatorname{sgn}(\sigma)$ for negative in-forests respectively.

The bijection θ associates the non-trivial matchings in the bideterminant with the non-trivial matchings identifying the functional graphs. So the above steps work for all required matchings and, with the continued assumption of no required swap, we can

express the positive bideterminant as

$$\begin{aligned}\det^+(M_D(R|C)) &= \bigoplus_{\pi' \in A_{R',C'}} \bigotimes_{k \in [2n]\setminus R} (M_D)_{k,\pi'(k)} \\ &= \bigoplus_{\pi \in A_{\bar{R},\bar{C}}} \bigotimes_{k \in [2n]\setminus R} (M_D)_{k,\theta^{-1}(\pi)(k)} \\ &= \bigoplus_{\pi \in A_{\bar{R},\bar{C}}} \varphi_\pi(D,w),\end{aligned}$$

and if $\operatorname{sgn}(\bar{R})\operatorname{sgn}(\bar{C})(-1)^{|\bar{R}|} = -1$, as a similar expression with summing over $S_{\bar{R},\bar{C}} \setminus A_{\bar{R},\bar{C}}$ instead of $A_{\bar{R},\bar{C}}$. In the last step, Proposition 3.1.38 is used to express the product via the weight of functional subgraphs with fixed transitions π. A similar expression exists for the negative bideterminant.

The final step is formulating the sign-reversing involution ι of Section 2.2.3, also appearing in the proof of Theorem 3.1.40. The functional graphs corresponding to in-forests may now have cycles, but all of those cycles contain an edge between R and C. So this time the involution considers the smallest cycle, which does not contain such an edge. This corresponds to π containing a cycle. Let the set of these pairs (G_π, π) be $\mathcal{F}_c^+$ for even matchings and $\mathcal{F}_c^-$ for odd ones. Everything else stays the same in comparison to the previous usages: If the smallest cycle C gets fixed, the functional graph and its weight stays the same, whereas its contribution switches from the positive determinant to the negative and vice versa. So the involution equates the contributions of the positive and negative bideterminant-parts not relating to in-forests. So

$$\bigoplus_{(G,\pi)\in\mathcal{F}_c^+} w(G) = \bigoplus_{(G,\pi)\in\mathcal{F}_c^-} w(G) = \Delta.$$

All that is left is some clean-up. At first the symmetric extension is used to display the equations. Secondly, the constant factor $\operatorname{sgn}(\bar{R})\operatorname{sgn}(\bar{C})(-1)^{|\bar{R}|}$ consists of signs of complements and $(-1)^{|\bar{R}|} = (-1)^{n-|R|}$. So, via Lemma 3.1.48 and the knowledge that $|R| = |C|$, an expression with all complements taken care of follows from

$$\begin{aligned}\nu(\bar{R}) + \nu(\bar{C}) &= \sum_{i\in\bar{R}} i - \frac{|\bar{R}|(|\bar{R}|+1)}{2} + \sum_{j\in\bar{C}} j - \frac{|\bar{C}|(|\bar{C}|+1)}{2} \\ &= \Big(\sum_{i\in\bar{R}} i + \sum_{j\in\bar{C}} j + 2\sum_{k\in[n]} k\Big) - 2\sum_{k\in[n]} k \\ &= \Big(\sum_{i\in\bar{R}} 2i + \sum_{i\in R} i + \sum_{j\in\bar{C}} 2j + \sum_{j\in C} j\Big) - 2\sum_{k\in[n]} k \\ &= \sum_{i\in R} i + \sum_{j\in C} j + 2\Big(\sum_{i\in\bar{R}} i + \sum_{j\in\bar{C}} j - \sum_{k\in[n]} k\Big).\end{aligned}$$

Thus, it holds that $\operatorname{sgn}(\bar{R})\operatorname{sgn}(\bar{C})(-1)^{|\bar{R}|} = (-1)^{n-|R|+\sum_{i\in R} i+\sum_{j\in C} j}$. This can be used to collect all above cases into one formula as the sign swaps the positive and negative bideterminants, which is exactly the multiplication with $\hat{n}_e$. Deciding the case thus reduces to a multiplication with $\hat{n}_e^{n-|R|+\sum_{i\in R} i+\sum_{j\in C} j}$. □

In the following we write for set A and integer n the element-wise sum $A + n := \{a + n \mid a \in A\}$ and grant it higher precedence than the union. This is used here as the Minoux matrix is composed of four blocks. We proved the all minors matrix tree theorem

on commutative semirings by removing only rows and columns in the upper left block. This can be extended to also remove rows and columns in the lower right block, which further reduces the size of the matrix.

Corollary 3.1.55 [Min99, Section 3.1]**.** *Let (D, w) be a weighted digraph with weights in a commutative semiring, then*

$$\hat{n}_e^{n-|R|+\sum_{i\in R} i+\sum_{j\in C} j} \operatorname{bid}(M_D(R \cup (R \cup C) + n | C \cup (R \cup C) + n)) = \hat{\kappa}^-_{R,C}(D, w) + \hat{h}_\Delta.$$

Proof. The non-zero entries of M_D with indices beyond n all have indices $(i, i+n)$, $(i+n, i)$ or $(i+n, i+n)$.

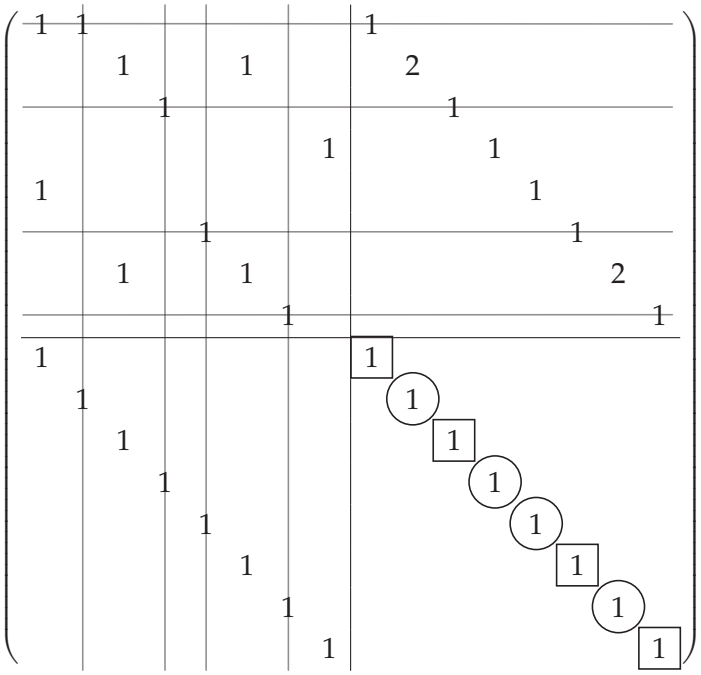
If the row $r \in R$ is removed, then so is the entry $(r, r+n)$, thus all permutations, which have a non-zero contribution to the determinant, have to choose $(r+n, r+n)$. As the entry $(M_D)_{r+n,r+n} = e$, it can be omitted and thus the row and column $r+n$ removed.

If the column $c \in C$ is removed, then so is the entry $(c+n, c)$ and a similar conclusion follows. So the $c+n$-th row and column can be removed. In conclusion, we can remove the rows and columns with indices in $(R \cup C) + n$. □

Example 3.1.56. We continue Example 3.1.45 and calculate the signed weight of all in-forests of Figure 3.1 with roots in $R = \{1, 3, 6, 8\}$ and chosen vertices in $C = \{2, 4, 5, 7\}$. So $\nu(R) + \nu(C) = 36$ and thus $\hat{n}_1^{36} = \hat{p}_1$. We already explicitly counted all those in-forests of (D, w), and know, that $\kappa^{-\oplus}_{R,C}(D, w) = \kappa^{-\ominus}_{R,C}(D, w) = 1$. In fact, there are no other functional subgraphs with forced edges from R to C, so we expect

$$\hat{n}_1^{8-4+\sum_{i=1}^{8} i} \operatorname{bid}(M_D(R \cup (R \cup C) + n | C \cup (R \cup C) + n)) = \hat{\kappa}^-_{R,C}(D, w) = \hat{h}_1.$$

The Minoux matrix of (D, w) is a 16×16 matrix, of with we can remove twelve rows and columns. The crossed out rows and columns directly correspond to the roots and chosen vertices. The boxed values are all the only non-zero entry in their column and so must be chosen. Their rows and columns can therefore be crossed out too. A similar argument follows for the circled values as they are the only non-zero entry in their respective rows. Due to the rare coincidence that $R \cup C = [n]$, we can remove all rows and columns with indices beyond n.

$M_D =$ 

The reduced Minoux matrix is thus a 4×4 matrix. This matrix has two rows, which are equal, so due to Proposition 3.1.30, the positive and negative bideterminant must be equal. Their exact values are e as σ is the only odd permutation and σ' the only even. Their corresponding matchings from R to C are given in Example 3.1.45 and are $\pi_{F'}$ and π_F respectively.

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix}, \qquad \sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 1 \end{pmatrix}. \qquad \lozenge$$

While the all minors matrix tree theorem on commutative semirings is the most general version of the matrix tree theorem we can expect to find, we reformulate it in the context of rings as calculating the bideterminant is computationally very expensive. A closer overview can be found in Section 3.2.4.

Corollary 3.1.57 (All Minors Matrix Tree Theorem on Commutative Rings)**.** *Let (D, w) be a weighted digraph with weights in a commutative ring, then*

$$(\ominus e)^{\sum_{i \in R} i + \sum_{j \in C} j} \det(L_D(R|C)) = \kappa^-_{R,C}(D, w).$$

Note, that we can use the additive inverse $\ominus e$ as the weights are now elements of a commutative ring. This is not possible in semirings, which led to the introduction of the symmetric extension in Section 3.1.2.

Proof. The proof uses the transformation given in (3.1.2). The sign of $n - |R|$ columns has to be inverted, so the determinant has to be multiplied via $(\ominus e)^{n-|R|}$, which thus appears twice and can be ignored. □

Using the all minors matrix tree theorem on commutative semirings it is now possible to derive all previous versions of the matrix tree theorem. The undirected case can be simulated by using a directed graph and replacing the undirected edge with a pair of directed edges. We can also derive the Kirchhoff polynomial from Section 2.5.3 if the weights of edges are in the commutative polynomial ring $\mathbb{R}[x_1, \dots, x_m]$, and $w(e_i) = x_i$.

3.1.5 Grassmann–Berezin Calculus

A similar result in commutative rings can be reached via an approach using the notation of quantum field theory. This path was chosen and generalised in a different direction by A. Abdesselam. The introduction to the Grassmann–Berezin calculus is echoed and shortened here.

Definition 3.1.58. Let R be a commutative ring with units containing $\mathbb{Q}$. The *Grassmann algebra* $R[\chi] := R[\chi_1, \dots, \chi_n]$ is defined by

$$R\langle \chi_1, \dots, \chi_n \rangle / (\chi_i \chi_j + \chi_j \chi_i \mid i, j \in [n]),$$

where $R\langle \chi_1, \dots, \chi_n \rangle$ denotes the free R-algebra (which can be understood as a polynomial ring with non-commuting variables) and $(\chi_i \chi_j + \chi_j \chi_i \mid i, j \in [n])$ the ideal generated by all expressions $\chi_i \chi_j + \chi_j \chi_i$.

The Grassmann algebra is intensively used to study fermions, which obey the anticommutative laws dictated by the ideal [NO98].

Proposition 3.1.59 [Abd04, Proposition 1]**.** *$R[\chi]$ is a free R-module with basis*

$$B := \{\chi_{i_1} \cdots \chi_{i_k} \mid 0 \leq k \leq n, 1 \leq i_1 < \cdots < i_k \leq n\}.$$

This basis has 2^{k-1} elements with k factors and therefore the basis B contains 2^n elements in total. So every element $f \in R[\chi]$ can be uniquely written as $\sum_{b\in B} f_b b$, with $f_b \in R$. It is also useful to consider the distinction between even and odd elements f of $R[\chi]$, where $f_b = 0$ if b consists of an odd number of factors, respectively an even number. The sub-R-module of even elements is written as $R[\chi]_{\text{even}}$ and of the odd elements $R[\chi]_{\text{odd}}$.

Theorem 3.1.60 (Pauli Exclusion Principle) [Abd04, Proposition 2]**.** *Let $f \in R[\chi]_{\text{odd}}$, then $f^2 = 0$.*

We also define derivations $\frac{\partial}{\partial\chi_i}$ in the usual sense, in such that

$$\frac{\partial}{\partial\chi_j}\chi_{i_1} \cdots \chi_{i_k} := \begin{cases} 0 & \text{for } j \notin \{i_1, \dots, i_k\}, \\ (-1)^{\ell-1}\chi_{i_1} \cdots \chi_{i_{\ell-1}}\chi_{i_{\ell+1}} \cdots \chi_{i_k} & \text{for } j = i_\ell, \end{cases}$$

where the sign stems from swapping $j = i_\ell$ to the left via the application of the anticommutative laws. So we can now consider a sub-Grassmann algebra $R[\chi_I] := R[\chi_{i_1}, \dots, \chi_{i_k}]$ with $I = \{i_1, \dots, i_k\}$, which can be naturally embedded into $R[\chi]$. The image of the derivation $\frac{\partial}{\partial\chi_j}$ lies completely within $R[\chi_{\overline{\{j\}}}]$, which is the Grassmann algebra $R[\chi]$ with χ_j removed. Thus, we introduce the *Berezin integral*, which relies on a matching σ on subsets of $[n]$ as

$$\int \mathrm{d}\chi_{\sigma(1)} \cdots \mathrm{d}\chi_{\sigma(k)} : \begin{cases} R[\chi] \to R[\chi_{\overline{\sigma([k])}}] \\ f \quad \mapsto \left(\frac{\partial}{\partial\chi_{\sigma(1)}} \circ \cdots \circ \frac{\partial}{\partial\chi_{\sigma(k)}}\right) f. \end{cases}$$

If the integral integrates over all variables, the result is in R and the sign is determined by the sign of the permutation σ as in the j-th step, the correct variable χ_j has to be swapped to the left. So all inversions are applied and are counted via the anticommutative laws. This results in

$$\begin{aligned} \int \mathrm{d}\chi_{\sigma(1)} \cdots \mathrm{d}\chi_{\sigma(n)} f &= \operatorname{sgn}(\sigma) \int \mathrm{d}\chi_1 \cdots \mathrm{d}\chi_n f \\ &= \operatorname{sgn}(\sigma) f_{\chi_1\chi_2\cdots\chi_n}. \end{aligned} \tag{3.1.5}$$

One can also define the exponential function on $R[\chi]$. For this, we use that all $f \in R[\chi]$ with $f_1 = 0$ are nilpotent as can be shown via Theorem 3.1.60. So the infinite sum in (3.1.6) ends after finitely many steps.

$$\exp(f) := \sum_{i=0}^{\infty} \frac{f^i}{i!} \tag{3.1.6}$$

We also identify the values $\chi_1, \dots, \chi_n$ with their vector χ, and write $\psi^{\mathrm{T}} A\chi := \sum_{i,j=1}^n \psi_i A_{ij}\chi_j$, which is an element of $R[\chi]$.

The goal is to formulate the determinant in the context of the Grassmann–Berezin calculus. This requires so-called *entangled variables* ψ and $\overline{\psi}$, for which a shorthand notation $\psi_{i_1}\overline{\psi}_{j_1} \cdots \psi_{i_k}\overline{\psi}_{j_k} =: (\psi_I\overline{\psi}_J)_{\text{ent}}$ is defined. The convention for I and J is to be increasingly ordered, so $i_\ell < i_{\ell'}$ and $j_\ell < j_{\ell'}$ for $\ell < \ell'$. We also define the *entangled Berezin integral* over the Grassmann algebra with entangled variables $R[\psi, \overline{\psi}]$ as

$$\int \mathrm{d}(\overline{\psi}, \psi)_{\text{ent}} := \int \mathrm{d}\overline{\psi}_1 \mathrm{d}\psi_1 \cdots \mathrm{d}\overline{\psi}_n \mathrm{d}\psi_n.$$

From this, the formula for the determinant follows after some calculation steps.

Theorem 3.1.61 [Abd04, Proposition 5]**.** *Let $A \in \mathcal{M}_n(R)$ and I, J be subsets of $[n]$ with the same cardinality, then the determinant can be expressed via*

$$(-1)^{\sum_{i\in I} i+\sum_{j\in J} j} \det(A(I|J)) = \int \mathrm{d}(\bar{\psi},\psi)_{\mathrm{ent}} (\psi_I \bar{\psi}_J)_{\mathrm{ent}} \exp(-\bar{\psi}^{\mathrm{T}} A \psi).$$

The above theorem enables us to interpret the determinant of a reduced matrix as a Berezin integral. We could now prove the matrix tree theorem for commutative rings, but we already proved it for a more general case of commutative semirings. A. Abdesselam did not just prove the theorem, but instead generalised it in a different direction. He considered a general matrix, in which the rows may not add up to zero. This leads to the enumeration of a different set of forests.

Instead of explicitly giving a set I of roots and a set J of chosen vertices, of which every tree must contain exactly one vertex, we also allow an arbitrary amount of trees in addition to the fixed trees. The roots of those variable trees are collected in $\mathcal{R}$. A pair $(F, \mathcal{R})$ is called *admissible* with relation to I and J iff F is a forest and every tree either contains exactly one vertex of I and J or one vertex of $\mathcal{R}$. Let the set of all admissible pairs of a graph be denoted by $\mathcal{A}$. Let σ_F be also the unique matching from I to J, such that $\sigma(i)$ and j are in the same tree for all $i \in I$. The generalised matrix tree theorem is as follows:

Theorem 3.1.62 [Abd04, Theorem 1]**.** *Let $A \in \mathcal{M}_n(R)$ and I, J be subsets of $[n]$, then*

$$\det(A(I|J)) = (-1)^{\sum_{i\in I} i+\sum_{j\in J} j} \sum_{(F,\mathcal{R})\in\mathcal{A}} \operatorname{sgn}(\sigma_F) \prod_{j\in\mathcal{R}} \Big(\sum_{i=1}^{n} A_{ij}\Big) \prod_{(i,j)\in F} (-A_{ij}).$$

At first it is necessary to make sense of the product assigned to an admissible pair $(F, \mathcal{R})$. The inner product $\prod_{(i,j)\in F}(-A_{ij})$ is nothing new as this resembles the edge-weights $w(i,j) = -A_{ij}$ of the forest. The sum before is new. If we interpret A_{ii} as the usual sum of edge-weights to vertex i, this would evaluate to 0. This is in line with our usual understanding of the theorem, which disregards forests, in which not all vertices are connected to a root in I. These forests were imagined as non-spanning sub-forests of the graph. However, we now allow extra trees, which are not forced by I and J, but are rooted at $r \in \mathcal{R}$. These trees require a starting weight $w(r)$, which is assigned to the vertex r. The diagonal elements then become

$$A_{rr} = w(r) - \sum_{i\neq r} w(i,r) = w(r) + \sum_{i\neq r} A_{ir},$$

and the sum in the formula evaluates to the weight of the emerging root r. The weight of the admissible pair is thus the weight $w(F)$ of the forest multiplied with the weight of the emergent roots $w(r)$.

The following proof presented here omits a most central part, which corresponds to identifying whether a sub-graph is counted or not. The complete proof can be found in [Abd04, Section 3]. There, A. Abdesselam also mentioned that a proof using the determinant expansion of [Moo94] would be possible, but did not state it, and neither did J. W. Moon. This direct proof would not require any notation from quantum field theory.

Proof. Due to Theorem 3.1.61, we can express the determinant as an entangled Berezin integral multiplied by the factor $(-1)^{\sum_{i\in I} i+\sum_{j\in J} j}$. The next step is to separate the vertex-

weight from the rest of the matrix, which we express as

$$\begin{aligned}\bar{\psi}^{\mathrm{T}} A\psi &= \sum_{i,j=1}^{n} \bar{\psi}_i A_{ij} \psi_j \\ &= \sum_{j=1}^{n} \bar{\psi}_j \Big(\underbrace{\sum_{i=1}^{n} A_{ij}}_{=w(j)} \Big) \psi_j + \sum_{i,j=0}^{n} (\bar{\psi}_i - \bar{\psi}_j) A_{ij} \psi_j .\end{aligned}$$

As the Grassmann algebra is anti-commutative, swapping two factors inverts the sign. So the integral can be expressed via

$$\begin{aligned}&\int \mathrm{d}(\bar{\psi},\psi)_{\mathrm{ent}} (\psi_I \bar{\psi}_J)_{\mathrm{ent}} \exp\Big(- \sum_{j=1}^{n} w(j) \bar{\psi}_j \psi_j - \sum_{i,j=1}^{n} A_{ij} (\bar{\psi}_i - \bar{\psi}_j) \psi_j \Big) \\ &= \int \mathrm{d}(\bar{\psi},\psi)_{\mathrm{ent}} (\psi_I \bar{\psi}_J)_{\mathrm{ent}} \Big(\prod_{j=1}^{n} e^{-w(j)\bar{\psi}_j \psi_j} \Big) \Big(\prod_{j=1}^{n} e^{-A_{ij}(\bar{\psi}_i - \bar{\psi}_j)\psi_j} \Big) \\ &= \int \mathrm{d}(\bar{\psi},\psi)_{\mathrm{ent}} (\psi_I \bar{\psi}_J)_{\mathrm{ent}} \Big(\prod_{j=1}^{n} (1 - w(j) \bar{\psi}_j \psi_j) \Big) \Big(\prod_{j=1}^{n} (1 - A_{ij}(\bar{\psi}_i - \bar{\psi}_j)\psi_j) \Big),\end{aligned}$$

where the last line follows by the Pauli exclusion principle and that all elements of $R[\chi]$ with $f_1 = 0$ are nilpotent. We can now separate the vertex-weights and the matrix A from the integral to get

$$(-1)^{\sum_{i\in I} i + \sum_{j\in J} j} \det(A(I|J)) = \sum_{(F,\mathcal{R}) \in [n]^2 \times [n]} \Big(\prod_{r\in\mathcal{R}} w(r) \Big) \Big(\prod_{(i,j)\in F} (-A_{ij}) \Big) \Omega_{F,\mathcal{R}},$$

where $\Omega_{F,\mathcal{R}}$ is $\mathrm{sgn}(\sigma_F)$ for admissible pairs and 0 otherwise. The expression hidden behind $\Omega_{F,\mathcal{R}}$ contains the Berezin integral with all variables:

$$\Omega_{F,\mathcal{R}} := \int \mathrm{d}(\bar{\psi},\psi)_{\mathrm{ent}} (\psi_I \bar{\psi}_J)_{\mathrm{ent}} \Big(\prod_{j=1}^{n} (\bar{\psi}_j \psi_j) \Big) \Big(\prod_{j=1}^{n} (\bar{\psi}_i - \bar{\psi}_j) \psi_j \Big)$$

The evaluation of $\Omega_{F,\mathcal{R}}$ takes up a good part of the proof of the theorem and is omitted here. For an in-depth proof see [Abd04, Lemma 1]. □

Example 3.1.63. Consider Example 1.3.5, in which we calculated the number of spanning trees of the complete graphs K_n. We can now use Theorem 3.1.62 to provide a different proof. As the graphs K_n are undirected we can select an arbitrary vertex i as the designated root for a spanning tree. Let $i = 1$. We now force no roots or chosen vertices in the usual sense, but disregard trees, which are not rooted in i. This is done by setting the vertex-weights to $w(1) = 1$ and all others 0. This leads to the matrix

$$A = \begin{pmatrix} n & -1 & \cdots & -1 \\ -1 & n-1 & & \vdots \\ \vdots & & \ddots & -1 \\ -1 & \cdots & -1 & n-1 \end{pmatrix}.$$

As we do not remove any rows or columns, the number of spanning trees is given by the determinant $\det(A) = n^{n-2}$, which agrees with the result of Example 1.3.5.

A different question would be to count the number of spanning forests on the complete graph K_n. For this we allow all vertices to spawn trees, and thus $w(v) = 1$ for all vertices. This leads to

$$A' = \begin{pmatrix} n & -1 & \cdots & -1 \\ -1 & n & & \vdots \\ \vdots & & \ddots & -1 \\ -1 & \cdots & -1 & n \end{pmatrix},$$

which is $L_{K_{n+1}}(i)$. So we can conclude that the complete graph with n vertices contains n^{n-2} spanning trees and $(n+1)^{n-1}$ spanning forests. ◊

3.2 Hypergraphs

While graphs have been intensively studied, hypergraphs are a much more recent generalisation. The step from graphs to hypergraphs is rather intuitive as the main restriction of graphs is the number of endpoints for an edge. So the most reasonable generalisation would be to allow any number of vertices within an edge.

Definition 3.2.1 [Ber89, p. 1]**.** Let V be a non-empty set of *vertices* and E a set of non-empty subsets of V. These subsets $e \in E$ are called *hyperedges*. If no confusion can arise, they will be also called edges for short. For $|e|$ odd the edge is called *odd*. The tuple (V, E) is called a *hypergraph*.

We also can immediately generalise paths and connectedness, but then the easy generalisations stop already as even the generalisation of a cycle or tree is not as easy to formulate and not consistent in the literature. We consider two possible perspectives for trees, but first we have to establish other notation, which would be trivial for graphs.

Definition 3.2.2 [Ber89, p. 1]**.** A hypergraph H is called *simple* iff for all edges $e, e' \in E$ it follows that $e \subseteq e'$ implies $e = e'$.

Definition 3.2.3 [Ber89, p. 2]**.** A hypergraph H is an *n-graph*, also called an *n-uniform hypergraph*, for $n \in \mathbb{N}$ iff for all edges $e \in E$ it holds that $|e| = n$.

Clearly, every n-graph is a simple hypergraph as all edges have the same number of vertices, and a graph in the usual sense is a 2-graph. We will also require a way to construct a graph out of a hypergraph.

Definition 3.2.4 [MV02, p. 1404]**.** Let H be a hypergraph and $\mathcal{E} := \{(v, e) \in V \times E \mid v \in e\}$, then the graph $|H| := (V \cup E, \mathcal{E})$ is the *topological realisation* of H.

The topological realisation $|H|$ of a hypergraph H is always a bipartite graph as the edges of $|H|$ always connect an edge of H with a vertex of H. So a cycle in the topological realisation can be written as a sequence $v_1 e_1 \cdots e_n v_1$. It is possible that two hyperedges connect two vertices without being the same edge, so it is relevant which edge was chosen for a given cycle. Thus, a *cycle* within a hypergraph is defined via an alternating sequence of vertices and hyperedges, such that the cycle of H is a cycle in $|H|$. The length of a cycle is still defined by the number of distinct vertices.

This definition of a cycle is sometimes called a *Berge-cycle* as it was first formulated in [Ber89]. This is to separate it from the notion of *tight cycles*, in which the i-th hyperedge not only contains the i-th vertex of the cycle, but all $\ell - 1$ previous vertices also, where $\ell = |e_i|$, and *loose cycles*, in which the edges are as separated as possible. This means that $e_{i-1} \cap e_i = \{v_i\}$ and $e_i \cap e_j = \emptyset$ for $j \notin \{i, i+1\}$. For a more in-depth explanation see [Ver16].

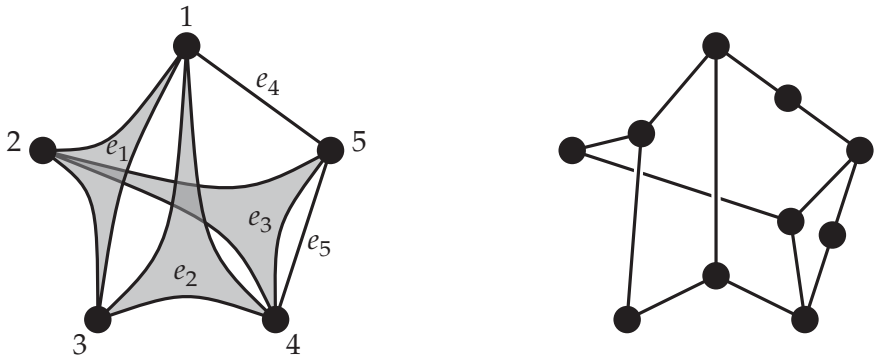


Figure 3.3: The hypergraph H with five vertices and five edges and its topological realisation $|H|$, which consists of ten vertices and thirteen edges.

Example 3.2.5. Consider the hypergraph H with five vertices and hyperedges $e_1 = \{1,2,3\}$, $e_2 = \{1,3,4\}, e_3 = \{2,4,5\}, e_4 = \{1,5\}$ and $e_5 = \{4,5\}$, illustrated in Figure 3.3. The edges e_4 and e_5 have only two endpoints, while the other edges have three endpoints. Due to e_5 being a subset of e_3, this is not a simple graph. The topological realisation $|H|$ thus contains $|V| + |E| = 10$ vertices and $\sum_{e \in E} |e| = 13$ edges.

The hypergraph H contains many cycles. If we consider the cycle $1, e_4, 5, e_5, 4, e_2, 1$, we see that all edges have an intersection of exactly one vertex. So this cycle is also a loose cycle. The cycle $3, e_1, 2, e_3, 4, e_2, 3$ is an example for a non-loose cycle as e_1 and e_2 share two vertices. ◊

We are now ready to identify two different notions of hypertrees. One is widely used and is characterised in [Ber89] as *arboreal hypergraphs*, which we will adopt. We will use the other to define *hypertrees*.

Definition 3.2.6 [Ber89, Section 5.4]**.** Let H be a simple hypergraph. It is called *arboreal* iff the following two properties hold:

i. If J is a subset of edges such that $e \cap e' \neq \emptyset$ for $e, e' \in J$, then $\bigcap_{e \in J} e \neq \emptyset$.

ii. All cycles of length greater than 2 contain at least three edges with non-empty intersection.

The first property is also called the *Helly property* [Ber89, Section 1.5].

Definition 3.2.7 [see MV02, Definition 3.2]**.** A hypergraph H is called a *hypertree* iff it is connected and contains no cycles.

Hypergraphs and hypertrees are direct generalisations of graphs and trees. Thus, we will generalise the notions of weights of graphs and trees in a direct sense. The definition of $\kappa(H, w)$, which is the sum of weights of all hypertrees in H, follows directly.

C. Berge gave in [Ber89] another characterisation for arboreal hypergraphs, which he contributed to [Duc78]. It is concerned with finding a tree on the vertices, which is always a tree if the vertices are restricted to the ones of any hyperedge.

Theorem 3.2.8 [Ber89, Theorem 13 in Chapter 5]**.** *A hypergraph $H = (V, E)$ is arboreal iff there exists a tree T on V, such that the subgraphs of T restricted to the vertices of hyperedges $e \in E$ are still trees.*

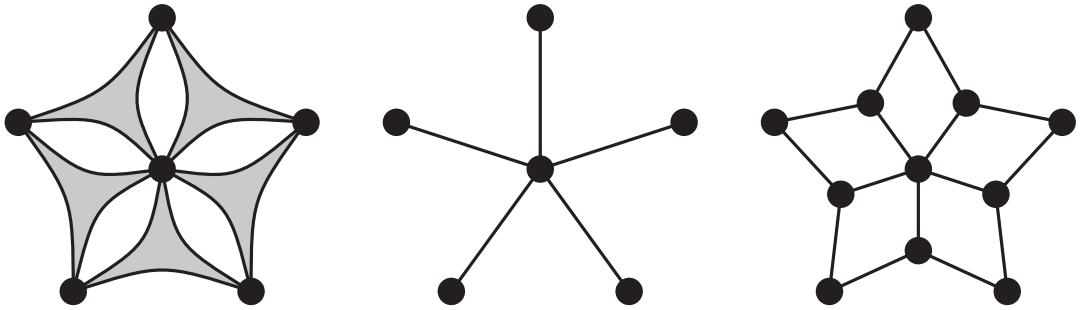


Figure 3.4: A hypergraph, a tree on the same vertices, and the topological realisation. This hypergraph is arboreal, but not a hypertree.

Lemma 3.2.9 [MV02, Proposition 3.3.ii]**.** *Let H be a k-uniform hypertree, then the number of vertices of H is given by $|V| = (k-1)|E| + 1$.*

Proof. The lemma is proven by a simple induction on the number of edges. A hypertree consisting of a single hyperedge contains exactly those vertices. As H is k-uniform, the edge contains k vertices. Every further edge must be connected to exactly one vertex of the hypertree. If otherwise the edge is connected to two vertices, the topological realisation would contain a cycle. If the edge is not connected at all, then the hypergraph would not be connected and thus not be a hypertree. □

Definition 3.2.10. Let H be a hypergraph with weights w, then the *weight* of all k-uniform hypertrees of H is defined as $\kappa^{[k]}(H, w)$.

Eventhough G. Masbaum and A. Vaintrob mentioned that Proposition 3.3 in [MV02] proves the equivalence of arboreal hypergraphs and hypertrees, this is not the case. Problems arise when the choice of a subtree within a hyperedge is restricted by neighbouring hyperedges as the following example shows.

Example 3.2.11. Figure 3.4 shows a hypergraph, which is an arboreal hypergraph, but not a hypertree. To check whether it is arboreal, we could, on one hand, check all subsets and cycles of length ≥ 3, or find a tree, such that the restriction to the vertices of a single hyperedge is still a tree. The only such tree is given in the middle. All other trees of these six vertices have a disconnected subtree.

The topological realisation of H, illustrated on the right, contains cycles. Thus, the hypergraph H is not a hypertree. This is to be expected as there are many hyperedges e and e', which share two vertices v and v', so the topological realisation contains the cycle e, v, e', v', e.

A different way to check whether the graph is a hypertree, is to check the requirement given in Lemma 3.2.9. As there are six vertices, but five edges, the equation $6 = 2 \cdot 5 + 1$ is false.

If we consider Figure 3.3, we can find arboreal subhypergraphs and spanning hypertrees. For example, the hyperedges e_1, e_2 and e_5 form an arboreal subhypergraph, which is not a hypertree, and the edges e_1 and e_2 form a spanning hypertree. Every spanning hypertree is arboreal as they contain no cycles at all. ◊

Thus, we can map hypertrees and arboreal hypergraphs to graphs. As hypertrees are a special class of arboreal hypergraphs, we can interpret them in two different ways as

a graph. In the first one, we add vertices for edges and form a bipartite graph, while in the second we choose a spanning tree on the same vertices, such that hyperedges induce subtrees. Both interpretations avoid cycles. We will now introduce a third interpretation, which introduces a cycle for every hyperedge.

Definition 3.2.12. A connected graph G is a *cactus graph* iff any two cycles overlap in at most one vertex. So no cycles share any edges.

So any hypertree can be mapped to a cactus graph via interpreting all hyperedges as a cycles. This cycle is not unique as many can be formed on a set of vertices. To restrict this to a unique graph, we introduce an orientation, which we imagine as walking through all vertices in a anticlockwise manner. In this case, the orientation is dependant on the visualisation of the hypergraph, so we aim for a more formal approach.

3.2.1 Orientations

We define an orientation on hyperedges in an arbitrary manner and build the aforementioned intuition from there. The advantage of this approach is that we are independent of the way the hypergraph is pictured.

Definition 3.2.13 [see MV02, Definition 3.6]**.** Let e be a hyperedge. An *orientation* $\mathrm{o}(e)$ is defined as a cyclic permutation on the vertices of e.

The first question is if the orientation of a hypergraph is well-defined. Initially, there is no predetermined sequence of hyperedges, but as it turns out, this does not matter as we are only interested in the length of the cycle $\prod_{e\in H} \mathrm{o}(e)$ for a hypergraph H.

Proposition 3.2.14 [see MV02, Proposition 3.4]**.** *Let T be an hypergraph with only odd hyperedges, then $\prod_{e\in H} \mathrm{o}(e)$ is a cyclic permutation for all orientations of e and permutations of $\mathrm{o}(e)$ iff T is a hypertree.*

Proof. We begin by assuming that T is a hypertree with k edges and prove by induction on the number of edges that $\sigma := \prod_{e\in H} \mathrm{o}(e)$. If $k = 1$ consists of a single edge, this is true as $\mathrm{o}(e)$ is already the whole tree and a cyclic permutation.

For the induction step, let the edges $e_1, \dots, e_{i-1}, e_{i+1}, \dots, e_k$ form a subtree T' of T. As a cyclic rearrangement of the product $\sigma' := \mathrm{o}(e_1) \circ \cdots \circ \mathrm{o}(e_{i-1}) \circ \mathrm{o}(e_{i+1}) \circ \cdots \circ \mathrm{o}(e_k)$ can be interpreted as $\pi^{-1} \circ \sigma' \circ \pi$ with $\pi = \mathrm{o}(e_1) \circ \cdots \circ \mathrm{o}(e_{i+1})$, we can relabel the edges and assume that the subgraph without e_k is a hypertree. So the permutation for T is given by $\sigma'\,\mathrm{o}(e_k)$ and is cyclic as e_k and T' share exactly one vertex and so do their permutations. The cycle is indicated in Figure 3.5 on the right.

If, on the other hand, the hypergraph is not a tree, it is either disconnected, in which every connected component is a hypertree and thus defines a cycle in the permutation, or it contains a cycle C. We now take a closer look at the cycle C and show that its permutation consists of two cycles. Let $e_1, \dots, e_j$ be the edges of C and $v_i := e_i \cap e_{i+1}$ be the vertices between edges e_i and e_{i+1}.

The orientation of edge e_i can therefore be expressed as $(v_i \sigma_i' v_{i+1} \sigma_i'')$. All elements of σ_i' and σ_i'' only appear in edge e_i, whereas v_i appears in exactly two edges. So, once a cycle enters σ_i', it has to walk through all elements. A permutation of C is thus given by

$$(v_1\sigma_1'v_2\sigma_1'') \circ (v_2\sigma_2'v_3\sigma_2'') \circ \cdots \circ (v_j\sigma_j'v_1\sigma_j'') = (v_j\sigma_j'\sigma_1'v_2\sigma_2'v_3 \cdots \sigma_{j-1}')(v_1\sigma_j''\sigma_{j-1}'' \cdots \sigma_1'').$$

This permutation contains two cycles. We can imagine this as cycle C splitting the walked path in two, one on the "left" side of the cycle and one on the "right" as illustrated in

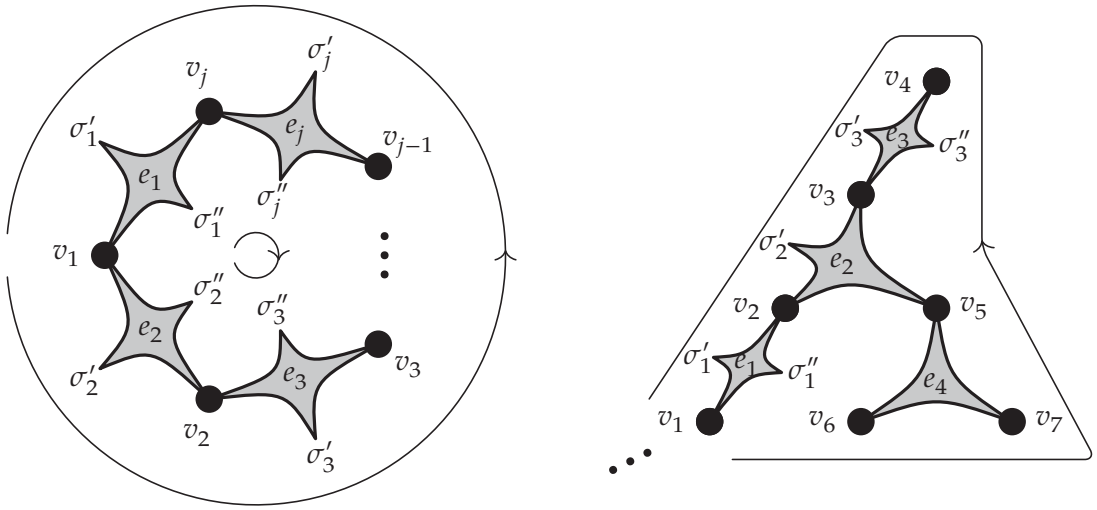


Figure 3.5: On the left: A cycle C in a hypergraph. The corresponding permutation consists of two cycles, with one containing all σ''_i and the other containing all σ'_i. On the right: A hypertree with a branch. This hypertree can be expressed as a cyclic permutation. Every hyperedge is assumed to be oriented anticlockwise.

Figure 3.5 on the left. This split happens as soon as a cycle is a subgraph. Hypertrees growing from vertices of C can be reduced to a cyclic permutation with one point shared with one of the cycles of C as of the first part of the proof. □

Using Proposition 3.2.14, we can identify odd hypertrees via their permutation. We will now build upon this permutation and define another permutation, which we will call the *essence* of a hypertree. The sign of the essence does not depend on the number of vertices, and so different spanning hypertrees on the same graph may lead to different signs.

Definition 3.2.15. Let T be an odd hypertree with fixed edge orientations and sequence, then the permutation $\mathrm{ess}(T)$ is called the *essence* of T and is given by

$$(\mathrm{ess}(1)\ \ \mathrm{ess}(2)\ \cdots\ \mathrm{ess}(n)) := \prod_{e\in T} \mathrm{o}(e), \tag{3.2.1}$$

where the left-hand side is written in cycle notation. The *sign* of an odd hypergraph is given by $\mathrm{sgn}(\mathrm{ess}(T))$.

The essence of a hypertree is not unique and depends on the edge orientation and the order of the edges in the product. While the order does not change whether the permutation is cyclic, the exact permutation may change. Luckily, the sign of essences is the same for a fixed hypertree.

Lemma 3.2.16 [MV02, Proposition 3.8]**.** *Let T be an odd hypertree, then the sign of T does not depend on the edge order. If the orientation of an edge e is changed from $\mathrm{o}'(e)$ to $\mathrm{o}(e)$, then the sign of T changes multiplicatively by $\mathrm{sgn}(\sigma)$, where $\mathrm{o}(e) = \sigma \circ \mathrm{o}'(e) \circ \sigma$.*

Proof. We start by proving the second part. Every cyclic permutation π can be transformed into any other σ via chaining the swaps (ij) before and after the permutation, such that $\pi = (ij) \circ \cdots \circ (i'j') \circ \sigma \circ (i'j') \circ \cdots \circ (ij)$. Without loss of generality, we can restrict us to

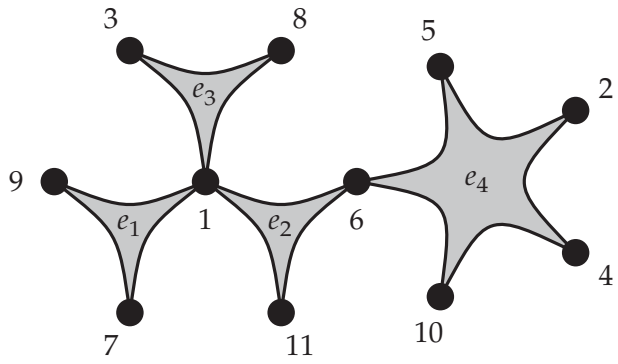


Figure 3.6: An odd hypertree consisting of the edges $\{1,7,9\}$, $\{1,3,8\}$, $\{1,6,11\}$ and $\{2,4,5,6,10\}$.

changing the orientation from $\mathrm{o}'(e)$ to $\mathrm{o}(e)$ of a single edge via a single swap of (ij). These orientations have different signs. The corresponding essences σ' and σ can be related by $\sigma' = (ij) \circ \sigma$. Thus, the sign changes too.

It is left to show that changing the order does not change the sign. Without loss of generality, let the two swapped edges e and e' be next to each other. If they do not share a vertex, nothing changes. So we assume that they share a single vertex 1, which can be reached via relabelling. Let this vertex be the root of subtrees S and S' containing e and e' respectively. These subtrees define cyclic permutations σ and σ', which appear in the cyclic permutation $\prod_{e \in T} \mathrm{o}(e)$. These two subsequences swap places. Let v and v' be the starting vertices of these subsequences, then the later sequence is moved to the front, whereas the first sequence, and everything in between, is moved to the back. Let k and k' be the respective lengths. The essence is based upon these permutations, and the goal is to prepend a permutation, to move from one to the other. We get

$$\pi = \begin{pmatrix} 1 & 2 & \cdots & v & & \cdots & & v+k-1 & \cdots & v' & \cdots & v'+k'-1 & \cdots & n \\ 1 & 2 & \cdots & v' & \cdots & v'+k'-1 & \cdots & & v & \cdots & & v+k-1 & \cdots & n \end{pmatrix}.$$

This permutation essentially swaps the indices, such that the essences can be mapped to each other. The most important part is that, since T is an odd hypertree, the subtrees contain an odd number of vertices and thus can be expressed through odd cycles. We remove the vertex 1 from both subtrees and swap these now even cycles. Since the step-sizes k and k' are even, they may never swap neighbouring vertices. So for every cycle in π, there exists a twin, which does the same, except moved to the right by one. Thus, the sign of π is always even, and subsequently, swapping does not change the signs of essences. □

The properties mentioned above seem somewhat technical, but they allow us to easily calculate essences and signs of hypertrees. For one, we do not have to consider the order in which we choose the edges. We can also choose an arbitrary orientation, which may be changed later as long as we include the changing sign.

Example 3.2.17. We consider a very similar hypertree to the one presented in [HR04], illustrated in Figure 3.6. It consists of four hyperedges $\{1,7,9\}$, $\{1,3,8\}$, $\{1,6,11\}$ and $\{2,4,5,6,10\}$ and eleven vertices.

We fix an orientation for all edges, which is chosen anticlockwise in Figure 3.6. For example, the orientation of the edge with five vertices is $\mathrm{o}(\{2,4,5,6,10\}) = (2,5,6,10,4)$.

The next step is to find a cyclic permutation for the tree. So we choose an arbitrary order of edges and get

$$o(\{1,3,8\}) \circ o(\{1,11,6\}) \circ o(\{2,4,5,6,10\}) \circ o(\{1,7,9\}) = (1\ 9\ 7\ 11\ 6\ 10\ 4\ 2\ 5\ 8\ 3).$$

This cyclic permutation can be easily constructed by reading off all the vertices in the graph in Figure 3.6 in an anticlockwise manner, starting at 1: We first encounter 9 and 7, as 1 is already known we pass it, and add $11, 6, 10, 4, 2$ and 5. Then we skip 6 and add 8 and finally 3. This closes the cycle. It also contains several interesting subsequences. If we remove 1 from it, we get the sequences $(9\ 7)$, $(11\ 6\ 10\ 4\ 2\ 5)$ and $(8\ 3)$, which are all the remaining vertices in anticlockwise order of subtrees rooted at 1. After the first observation, this is expected as we walk through all subtrees in an anticlockwise orientation.

However, we are more interested in the sign of the essence. The corresponding essence to the above permutation can easily be read off if we interpret the cycle notation as the one-line notation of permutations.[7] So both look exactly the same, but look similar. The essence thus can be written as

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 9 & 7 & 11 & 6 & 10 & 4 & 2 & 5 & 8 & 3 \end{pmatrix} = (2\ 9\ 5\ 6\ 10\ 8)(3\ 7\ 4\ 11),$$

and its sign is $(-1)^{5+3} = 1$.

Instead of going anticlockwise, we can also choose to walk clockwise. In this case, we get the cyclic permutation $(1\ 7\ 9\ 3\ 8\ 6\ 5\ 2\ 4\ 10\ 11)$ and thus the essence $(2\ 7\ 5\ 8)(3\ 9\ 4)$. This cyclic permutation corresponds to

$$(1\ 6\ 11) \circ (2\ 5\ 6\ 10\ 4) \circ (1\ 3\ 8) \circ (1\ 7\ 9).$$

The sign here is $(-1)^{3+2} = -1$. This follows from Lemma 3.2.16 as the orientation of every edge was changed. Changing the orientation of a hyperedge with three vertices from anticlockwise to clockwise requires a single swap. So, for example, we get $(1\ 6\ 11) = (6\ 11) \circ (1\ 11\ 6) \circ (6\ 11)$. Since $\operatorname{sgn}(6\ 11) = -1$, the sign of the tree changes accordingly. To change the orientation of the edge with five vertices, we have to apply two swaps and the sign of $(4\ 5)(6\ 10)$ is even. Thus, three edges change the sign an we get the expected result. ◊

We can now describe hypertrees via permutations and can assign them signs via their essences. This sign will later take on the role of the sign in the hypertree variant of the matrix tree theorem.

3.2.2 Pfaffians

Another required change is the move from the determinant to the so-called Pfaffian of a matrix. It can be imagined as the root of the determinant as it behaves very much like it.

Definition 3.2.18 [MV02, Section 4]**.** Let A be a $2n \times 2n$ skew-symmetric matrix, then the *Pfaffian* $\operatorname{pf}(A)$ is given by

$$\operatorname{pf}(A) \coloneqq \sum_{\sigma \in P} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} A_{\sigma(2i-1),\sigma(2i)}, \tag{3.2.2}$$

where $P \coloneqq \{\sigma \in S_{2n} \mid \sigma(2i-1) < \sigma(2i+1) \text{ and } \sigma(2i-1) < \sigma(2i) \text{ for } i \in [n]\}$.

[7] The one-line notation is not mentioned before and will not be mentioned later as it is rather confusing if used in conjunction with the cycle notation of cyclic permutations. It is essentially the two-line notation omitting the first line.

This definition is restricted to matrices with an even number of rows and columns. Sometimes, the Pfaffian is defined to be 0 for odd skew-symmetric matrices. As we will consider only Pfaffians of odd hypergraphs with the root removed, we will always be in the first case.

The set P is very restrictive on its permutations. The first inequality forces the odd indices to be increasingly ordered, while the second inequality implies that the image of even numbers is greater than the image of their predecessors. So we can deduce that $\sigma(2i-1) < \sigma(k)$ for all $i \in [n]$ and $k > 2i-1$. This does not hold true for even values. Thus, the image of 1 is smaller than all others, and $\sigma(1) = 1$.

A common identity of the Pfaffian is given via *perfect matchings*, which can be considered a special case of Definition 3.1.42, in which $A \cup B = [n]$ and $A \cap B = \emptyset$. So A and B form a partition of $[n]$. As the matching is a bijection, A and B have the same number of elements and a perfect matching may only exist for even n. As the direction is not relevant for us, we further restrict a perfect matching π by requiring $i < \pi(i)$ for all i. If this is not the case, we can transform the matching by using π^{-1} in that case. However, it is more useful to consider perfect matchings as sets of edges, given by $\{(i, \pi(i)) \mid i \in I\}$. We will identify theses sets with the notion derived from matchings.

Definition 3.2.19 [HR04, p. 60]**.** The *crossing number* of a perfect matching π is

$$\operatorname{cross}(\pi) := |\{(i, j, \pi(i), \pi(j)) \mid i < j < \pi(i) < \pi(j)\}|.$$

Lemma 3.2.20 [MV02, Equation (4.4); see Ste90, Section 2]**.** *Let A be a $2n \times 2n$ skew-symmetric matrix. The Pfaffian of A can also be expressed as*

$$\operatorname{pf}(A) = \frac{1}{2^n n!} \sum_{\sigma \in S_{2n}} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} A_{\sigma(2i-1),\sigma(2i)} \tag{3.2.3}$$

$$= \sum_{\substack{\pi \in S_{I,\bar{I}}, i < \pi(i), \\ I \subset [2n], |I| = n}} (-1)^{\operatorname{cross}(\pi)} \prod_{i \in I} A_{i,\pi(i)}. \tag{3.2.4}$$

Note that J. R. Stembridge, who first formulated (3.2.4), showed equivalence via the identity Theorem 3.2.26 [Ste90, Proposition 2.2]. However, this is only a strong indicator and does not prove the equivalence of our definition and (3.2.4) as the signs may differ.

Proof. The first expression can be proven in a straightforward way if the permutations are grouped by their products. The product does not change for σ and σ', where $\sigma' = \sigma$, except for $\sigma'(2i+1) = \sigma(2i-1)$, $\sigma'(2i+2) = \sigma(2i)$, $\sigma'(2i-1) = \sigma(2i+1)$ and $\sigma'(2i) = \sigma(2i+2)$. So the images of $2i$ and $2i+2$, as well as the images of $2i-1$ and $2i+1$ are swapped. This leads to a swap in the product as $A_{\sigma(2i-1),\sigma(2i)} = A_{\sigma'(2i+1),\sigma'(2i+2)}$ and $A_{\sigma(2i+1),\sigma(2i+2)} = A_{\sigma'(2i-1),\sigma'(2i)}$, but multiplication is commutative. This accounts for $n!$ permutations. We can also swap the images of $\sigma(2i-1)$ and $\sigma(2i)$. As the matrix is skew-symmetric, we introduce a negative sign this way, but we also introduce an inversion to the permutation. Thus, the products in (3.2.3) stay the same. This can be done for every pair, of which there are n. In total, there are $2^n n!$ permutations in (3.2.3) for every permutation in (3.2.2).

The second equation is a reformulation of the definition as $I = \{\sigma(2j-1) \mid j \in [n]\}$ and π maps every $\sigma(2j-1)$ to its corresponding partner $\sigma(2j) \in \bar{I}$. The sign of σ stems from a specific ordering, which changes from the number of inversions of σ to the crossing number $\operatorname{cross}(\pi)$. □

This further shows that the Pfaffian is 0 for matrices of odd sizes as there are no perfect matchings on an odd number of vertices. We can also describe the crossing number between two perfect matchings.

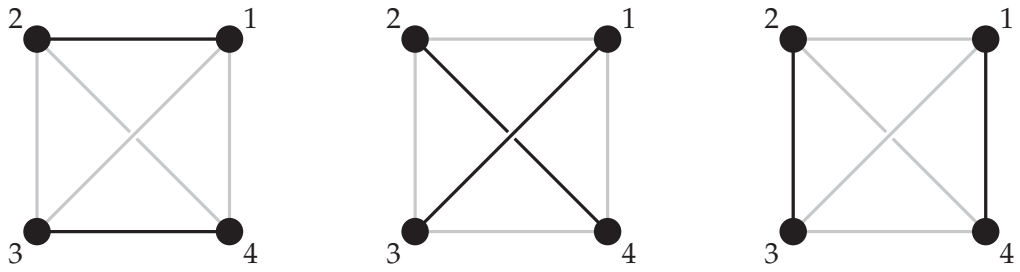


Figure 3.7: The three perfect matchings of the complete graph K_4. The crossing number is visible if the vertices are positioned anticlockwise.

Lemma 3.2.21 [Ste90, Lemma 2.1]**.** *Let π be a perfect matching and let π' be the perfect matching, which swaps $\pi(i)$ with $\pi(j)$ for an arbitrary pair (i,j), then* $\operatorname{cross}(\pi)$ *and* $\operatorname{cross}(\pi')$ *differ by an odd number.*

Proof. As π and π' are equal apart from $S \coloneqq \{i, \pi(i), j, \pi(j)\}$, the crossings are the same. On one hand, one crossing is added or removed by swapping $\pi(i)$ with $\pi(j)$. So let k be matched with $\pi(k)$ and let c be the number of crossings with S. Changing from π to π' introduces or removes two crossings, so for odd c the new number of crossing c' is still odd, respectively for even c and c'. Thus, the crossings change, but as we consider powers of -1 in the crossing numbers $\operatorname{cross}(\pi)$ and $\operatorname{cross}(\pi')$ they are only different due to the crossing within S. So the sign changes. □

Example 3.2.22. We consider the 4×4 matrix A. It is given by $A_{ii} = 0$ and $A_{ij} = -A_{ji}$. There are 4! permutations in S_4, but we are only interested in perfect matchings, of which there are only three. This also follows by $\frac{4!}{2^2 2!} = 3$. All permutations can be grouped in these three subsets.

The perfect matchings on K_4 are illustrated in Figure 3.7 and are written as

$$\pi_1 = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}, \qquad \pi_2 = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \qquad \pi_3 = \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}.$$

The illustrations show a quick way to visualise the crossing number: If the vertices are ordered by their number and are connected via the matching, the crossovers of those edges are the crossings counted by the crossing number. So they are given by $\operatorname{cross}(\pi_1) = \operatorname{cross}(\pi_3) = 0$ and $\operatorname{cross}(\pi_2) = 1$. Thus, the Pfaffian of A can be expressed as

$$\begin{aligned} \operatorname{pf}(A) &= A_{1,\pi_1(1)} A_{3,\pi_1(3)} + A_{1,\pi_2(1)} A_{2,\pi_2(2)} - A_{1,\pi_3(1)} A_{2,\pi_3(2)} \\ &= A_{1,2} A_{3,4} - A_{1,3} A_{2,4} + A_{1,4} A_{2,3}. \end{aligned}$$ ◊

For the inductive approach to the hypertree variant of the matrix tree theorem we will also need a recursive formula for the Pfaffian. This proof will then follow a similar line of thought to the deletion and contraction argument in Section 2.3.

Lemma 3.2.23 [MV02, Equation (4.5)]**.** *Let A be a $2n \times 2n$ skew-symmetric matrix. The Pfaffian of A can be calculated via the minors of A:*

$$\operatorname{pf}(A) = \sum_{i=1}^{2n} (-1)^i A_{1,i} \operatorname{pf}(A(1,i)). \tag{3.2.5}$$

Proof. We construct the sum by evaluating all possible first elements $A_{\sigma(1),\sigma(2)}$. Every permutation in P of (3.2.2) contains $\sigma(1) = 1$ as the image of 1 has to be smaller than all other images. So we can remove 1 and $\sigma(2)$ from σ and get a smaller matching. By removing this part, we also remove $\sigma(2)$ inversions. So these have to be considered. It is then possible to use the permutation σ' used in the proof of Proposition 3.1.49, which fills the gaps in σ. These σ' will be the permutations operating on $A(1, \sigma(2))$. □

As a final result, we will prove the statement, with which we motivated the Pfaffian. To show that the Pfaffian acts like the square root of the determinant for some matrices, we first have to prove a normal form for skew-symmetric matrix. This normal form can be used to prove a stepping stone in the final result.

Lemma 3.2.24 [Hab15, Theorem 2]**.** *Let A be a $2n \times 2n$ skew-symmetric matrix, then there exists a $2n \times 2n$ matrix B, such that*

$$BAB^{\mathrm{T}} = \operatorname{diag}\left(\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \dots, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, 0, \dots, 0\right).$$

Proof. This can be shown by induction. As the matrix is skew-symmetric, its diagonal is 0. We also observe that the first row of A is equal to the first column, with its sign reversed. So we apply row-operations to transform the first column into $(0, -1, 0, \dots, 0)^{\mathrm{T}}$ respectively. Afterwards the same operations are applied as column-operations, which leads to $(0, 1, 0 \dots, 0)$ in the top row. As the first column is now 0, apart from in the second row, we can add multiples of it to the other columns, such that they are also zeroed. The same column-operations are applied afterwards, and such we get the first block $\left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$. So we can observe the remaining matrix and apply the same induction on it if it is neither empty nor $\mathbb{0}$. Otherwise the induction is complete. □

Lemma 3.2.25 [Hab15, Theorem 3]**.** *Let B and A be skew-symmetric matrices. Then*

$$\operatorname{pf}(BAB^{\mathrm{T}}) = \operatorname{pf}(A) \det(B). \tag{3.2.6}$$

Proof. If A has an odd number of rows and columns, then both Pfaffians are 0 and the equation holds.

Otherwise let the matrices have $2n$ rows and columns. We then use the expression of (3.2.3), in which all permutations appear. So the Pfaffian of BAB^{T} can be rewritten as

$$\begin{aligned}
\operatorname{pf}(BAB^{\mathrm{T}}) &= \frac{1}{2^n n!} \sum_{\sigma \in S_{2n}} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} (BAB^{\mathrm{T}})_{\sigma(2i-1),\sigma(2i)} \\
&= \frac{1}{2^n n!} \sum_{\sigma \in S_{2n}} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} \sum_{k,\ell=1}^{2n} B_{\sigma(2i-1),k} A_{k,\ell} B_{\sigma(2i),\ell}.
\end{aligned}$$

As $A_{ii} = 0$ for all i, we can ignore the case for $k = \ell$, and such we can define a permutation φ, with $\varphi(\sigma(2i-1)) = k$ and $\varphi(\sigma(2i)) = \ell$ in a given summand. So for every σ there exists a corresponding φ and let $\psi = \varphi \circ \sigma$. We now swap sum and product and get

$$\begin{aligned}
\operatorname{pf}(BAB^{\mathrm{T}}) &= \frac{1}{2^n n!} \sum_{\sigma \in S_{2n}} \operatorname{sgn}(\sigma) \operatorname{sgn}(\psi) \prod_{i=1}^{n} B_{\sigma(2i-1),\psi(2i-1)} B_{\sigma(2i),\psi(2i)} A_{\psi(2i-1),\psi(2i)} \\
&= \frac{1}{2^n n!} \sum_{\varphi \in S_{2n}} \operatorname{sgn}(\varphi) \prod_{i=1}^{n} B_{2i-1,\varphi(2i-1)} B_{2i,\varphi(2i)} A_{\varphi(2i-1),\varphi(2i)} \\
&= \operatorname{pf}(A) \det(B).
\end{aligned}$$

□

This also enforces the knowledge that Pfaffians are—similarly to determinants—0 for matrices without full rank. We use this lemma to prove the general relation between the Pfaffian and the determinant hinted at in the beginning of this subsection.

Theorem 3.2.26 [Hab15, Theorem 4]**.** *Let A be a skew-symmetric matrix, then*

$$\operatorname{pf}(A)^2 = \det(A). \tag{3.2.7}$$

Proof. If A is singular, then we can transform the matrix via Lemma 3.2.24 into a canonical form. As there are rows consisting entirely out of zeroes, every product in the Pfaffian contains 0 and the Pfaffian itself equals 0. A similar argument follows for the determinant. If A has an odd number of rows, then the Pfaffian is by definition 0. The determinant of an odd skew-symmetric matrix is 0 too as $\det(A) = \det(A^{\mathrm{T}}) = \det(-A) = -\det(A)$.

Thus, we can assume A to be of full rank with $2n$ rows and columns. We then apply Lemmata 3.2.24 and 3.2.25, to transform the Pfaffian of A to the Pfaffian of the diagonal matrix $D = \operatorname{diag}\left(\left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right), \dots, \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)\right)$, which can now be calculated by hand:

$$\begin{aligned} \operatorname{pf}(A) &= \operatorname{pf}(BDB^{\mathrm{T}}) \\ &= \operatorname{pf}(D)\det(B) \\ &= 1 \cdot \det(B). \end{aligned} \tag{3.2.8}$$

On the other hand, we can simplify the determinant of A in a similar manner, such that we get $\det(A) = \det(D)\det(B)^2 = 1 \cdot \det(B)^2$. If we square (3.2.8) and divide it by this matrix identity, we get

$$\frac{\operatorname{pf}(A)^2}{\det(A)} = \frac{\det(B)^2}{\det(B)^2}.$$

The right-hand side is 1, and thus $\operatorname{pf}(A)^2 = \det(A)$. □

This does not mean that $\sqrt{\det(A)} = \operatorname{pf}(A)$ as the sign might be negative. Such an assertion can only be made if the sign is already known. An example would be counting hypertrees as there can never be a negative amount of trees. However, this may not be extended to weighted hypergraphs.

We conclude this subsection by proving a similar theorem to Theorem 1.3.4.1, which equates the determinant of minors of the Lagrangian. As the Pfaffian is only defined on skew-symmetric matrices, we choose $i = j$. The choice of L_G in the theorem is translated to the restriction $\sum_{i=1}^{n} A_{ij} = 0$ for all j.

Theorem 3.2.27 [MV02, Lemma 4.1]**.** *Let A be a $k \times k$ skew-symmetric matrix satisfying the equation $\sum_{i=1}^{k} A_{ij} = 0$ for all j, then*

$$\operatorname{pf}(A(1)) = (-1)^{i-1} \operatorname{pf}(A(i)). \tag{3.2.9}$$

Proof. If k is odd, then the Pfaffian is 0 and the equation holds. Otherwise let $k = 2n$. Without loss of generality let $i = 2$. From Lemma 3.2.24 and the fact that $A(1)$ and $A(2)$ have the same rank, we can find B, such that $BA(1)B^{\mathrm{T}} = A(2)$. Due to the requirement $\sum_{i=1}^{k} A_{ij} = 0$, this B is explicitly given by

$$B = \begin{pmatrix} -1 & -1 & -1 & \cdots & -1 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Using Lemma 3.2.25, we get $\operatorname{pf}(A(2)) = \operatorname{pf}(A(1))\det(B) = -\operatorname{pf}(A(1))$. □

3.2.3 The Pfaffian Matrix Tree Theorem

We have seen many parallels between the determinant and the Pfaffian. A strong indicator of a similar theorem to the matrix tree theorem is given by Theorem 3.2.27 as something similar also holds for the Laplacian matrix as is shown in Theorem 1.3.4.1. Lemma 3.2.9 shows that k-graphs contain hypertrees iff the number of vertices is one more than a multiple of $(k-1)$. As Pfaffians are zero for matrices of odd sizes, we hope for a connection between 3-uniform hypertrees and the Pfaffian of a 3-graph. We will formulate two proofs of this connection in this subsection: One using the approach of deletion and contraction described in Section 2.3, and the other using the sign-reversing involution of Section 2.2.

For this, we have to generalise the Laplacian matrix. As seen before, the notation of hypergraphs is not yet standardised, thus there are many non equivalent notions of Laplacians for hypergraphs.

Definition 3.2.28 [MV02, Equation (1.8)]**.** Let H be a 3-graph, then the *hyper-Laplacian matrix* Λ_H of H is an $n \times n$ matrix with 0 in its diagonal and

$$\sum_{e \in E} \operatorname{sgn}_e(i,j) w(e) \tag{3.2.10}$$

in entries (i,j), where $w(e)$ is the weight of e and $\operatorname{sgn}_e(i,j) := 1$ iff $\mathrm{o}(e)(i) = j$, -1 iff the roles of i and j are reversed and 0 otherwise.

This definition leads to a skew-symmetric matrix as entries (i,j) and (j,i) contain the same sum, but the cyclic permutations are swapped, so the sign changes too. This also one way to explain the zeroes on the diagonal. In the i-th row we have the contribution from (i,j,k) in the entry (i,j), but also (i,k,j) in entry (i,k). So, if we assign the diagonal elements the sum of all edges with i, we get 0.

The sign in the definition of the hyper-Laplacian is very similar to the sign described in [RR12], but different.[8] N. Reff and L. J. Rusnak used it to define the oriented adjacency matrix of a hypergraph. This highlights further parallels and differences between the regular and hyper-Laplacian: The Laplacian used for graphs is based upon the adjacency matrix, but then contains the degree of vertices as the diagonal elements, whereas the diagonal here is 0. This is due to the sign, which is dependant on the direction.

In [RR12] they used the oriented adjacency matrix to define their version of the Laplacian for hypergraphs. This Laplacian is more in line with the one for graphs as its diagonal contains the degrees of the vertices. It is again slightly different from the very early definition of the Laplacian of hypergraphs in [Chu93], which includes a normalisation factor.

With our choice for the hyper-Laplacian, we can now state the central theorem of this section. The general look does not change much. The only changes are a correction factor of the sign and a different function on a minor of the Laplacian. The function used here is the Pfaffian instead of the determinant or bideterminant of Section 3.1.

Theorem 3.2.29 (Pfaffian Matrix Tree Theorem) [MV02, Theorem 5.1]**.** *Let H be a 3-graph and $1 \leq i \leq n$, then*

$$(-1)^{i-1} \operatorname{pf}(\Lambda_H(i)) = \kappa^{[3]}(H,w).$$

Thus, this theorem can be used to calculate the weight $\kappa^{[3]}(H,w)$ of all 3-hypertrees of H defined in Definition 3.2.10. As mentioned before, two proofs will be given here in chronological order. The first one, proven by G. Masbaum and A. Vaintrob in [MV02], uses

[8] The sign in [RR12] is defined via $-\sigma(i,e)\sigma(j,e)$ and, thus, cannot change if i and j are swapped.

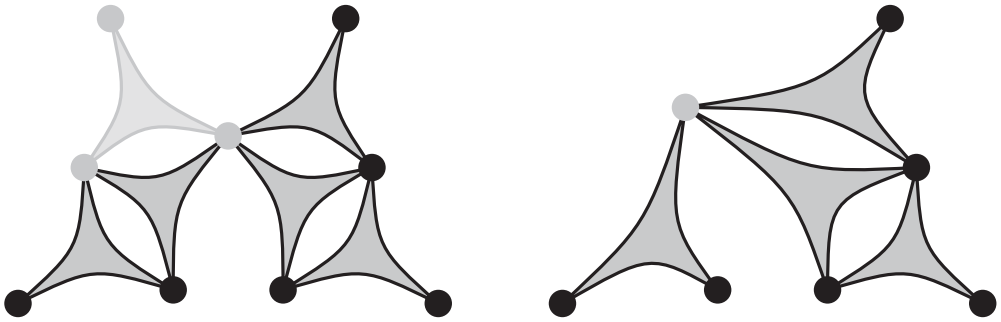


Figure 3.8: On the left: A 3-graph, on which the deletion–contraction step is applied to the edge marked in grey. On the right: The same graph with the grey edge contracted. Note that another edge was removed as it shared two vertices with the grey edge. Deletion of would lead to a disconnected graph as the vertices remain.

deletion and contraction, which was first used by M. Lewin and described in Section 2.3. Deletion–contraction employs an induction on the number of vertices and edges, which gives some insight on the construction of graphs, but none on the distribution of graphs within the sum. The second proof, by S. Hirschman and V. Reiner in [HR04], employs a sign-reversing involution, which was first applied to the matrix tree theorem by S. Chaiken. This involution concentrates on the internal distribution of graphs and exploits the implied direction of a cycle.

Proof via deletion–contraction. The proof closely follows the induction of Theorem 2.3.2. On one hand we have the sum of weights, which we count directly. On the other hand, the formula via the Pfaffian is used. Lemma 3.2.9 shows that 3-uniform hypertrees may only exist for an odd number of vertices. If n is even, there are no spanning hypertrees. Simultaneously, the Pfaffian is 0 for odd matrices. The minors of Λ_H are odd, so the theorem holds in these trivial cases.

The other case with n odd is solved via induction. The base case is the hypergraph with one vertex and no edges. The only spanning hypertree is consists of this single vertex and no edges. The minor of the hyper-Lagrangian is the 0×0 matrix, of which the Pfaffian is 1. A further required base case as we not only contract edges, but also delete them without reducing the number of vertices, is the hypergraph with $n > 1$ vertices and no edges. This is equivalent to every weight being 0, so we calculate the Pfaffian of the zero-matrix, which itself is 0. As there are no spanning trees, this base case holds too.

The induction step now consists of reducing a hypergraph with n vertices and k edges to hypergraphs with either $n-2$ or n vertices and $k-1$ edges. Without loss of generality let $i = 1$. So $\kappa^{[3]}(H, w)$ can be separated into the trees using edge e and those that do not. Without loss of generality, let $e = \{1, 2, 3\}$ as this can be ensured by relabelling.

The trees using this edge can be described via $\kappa^{[3]}(H', w')$, where H' is the hypergraph, where the vertices of e are contracted to a single vertex $1'$. The weight function w' is similar to w, but with vertices 1, 2 and 3 replaced by $1'$. All edges that contain at least two vertices of e are removed as they are never used in a tree and would degenerate to edges with two vertices. The resulting graph H' is smaller, thus we can apply the induction hypothesis and get $\kappa^{[3]}(H', w') = \operatorname{pf}(\Lambda_{H'}(1')) = \operatorname{pf}(\Lambda_H(1, 2, 3))$. To reconstruct the weights of the hypertrees in H, we have to multiply the weights of trees in H' with $w(e)$ as this edge is missing in all trees.

What remains are the trees, which do not use edge e. We call the hypergraph with e removed H''. This graph has fewer edges, and thus the induction hypothesis can be applied. So we get $\kappa^{[3]}(H'', w'') = \operatorname{pf}(\Lambda_{H''}(1))$. The hyper-Laplacian of H'' is very similar to the one of H as the weights w and w'' only differ in e. We can now apply Lemma 3.2.23 and express the Pfaffian of H'' via its minors:

$$\begin{aligned}\operatorname{pf}(\Lambda_{H''}(1)) = &- \sum_{k=1}^{n} \operatorname{sgn}_{\{2,2,k\}}(2,k) w''(\{2,2,k\}) \operatorname{pf}(\Lambda_{H''}(1,2,2)) \\ &+ \sum_{k=1}^{n} \operatorname{sgn}_{\{2,3,k\}}(3,k) w''(\{2,3,k\}) \operatorname{pf}(\Lambda_{H''}(1,2,3)) \\ &\mp \cdots + \sum_{k=1}^{n} \operatorname{sgn}_{\{2,n,k\}}(n,k) w''(\{2,n,k\}) \operatorname{pf}(\Lambda_{H''}(1,2,n)).\end{aligned}$$

We now exploit the similarity of w and H to w'' and H''. The only differences are if $w(\{1,2,3\})$ appears. This is either everywhere in the first row and column of H, which does not concern us, or in $(2,3)$ and $(3,2)$. The Pfaffian is expanded along the second row and column, so both remaining entries do not appear in the minors of H'', and thus $H''(1,2,k) = H(1,2,k)$. The edge e appears exactly once: in the second summand with $k = 1$ as $\operatorname{sgn}_{\{2,3,1\}}(3,1) w''(\{2,3,1\}) \operatorname{pf}(\Lambda_{H''}(1,2,3)) = 0$. This missing part is exactly the contribution from H'. So we conclude that

$$\begin{aligned}\kappa^{[3]}(H,w) &= w(e)\kappa^{[3]}(H',w') + \kappa^{[3]}(H'',w'') \\ &= w(e) \operatorname{pf}(\Lambda_H(1,2,3)) + \operatorname{pf}(\Lambda_{H''}(1)) \\ &= - \sum_{k=1}^{n} \operatorname{sgn}_{\{2,2,k\}}(2,k) w(\{2,2,k\}) \operatorname{pf}(\Lambda_H(1,2,2)) \\ &\quad + \sum_{k=1}^{n} \operatorname{sgn}_{\{2,3,k\}}(3,k) w(\{2,3,k\}) \operatorname{pf}(\Lambda_H(1,2,3)) \\ &\quad \mp \cdots + \sum_{k=1}^{n} \operatorname{sgn}_{\{2,n,k\}}(n,k) w(\{2,n,k\}) \operatorname{pf}(\Lambda_H(1,2,n)) \\ &= \operatorname{pf}(\Lambda_H(1)),\end{aligned}$$

which again follows from applying Lemma 3.2.23. This proves the theorem for $i = 1$. Other i can be derived via Lemma 3.2.25 and using the correct transformation matrix B to swap rows and columns. This introduces $\det(B) = (-1)^{i-1}$ to the equation. □

The structure of this proof is similar to Theorem 2.3.2. The main difference is that the entries of Λ_H do not contain the weight of a single edge, but the weights and orientations of all edges that share two vertices. Similarly, we will have to extract the weight of a single hypergraph from the sums of Λ_H to apply the sign-reversing involution.

Proof via a sign-reversing involution. At the core of the proof lies the sign-reversing involution also used in Section 2.2. This involution maps a hypergraph with a cycle to the same hypergraph with the cycle interpreted differently, such that the sign is inverted. Hypertrees do not have any cycles, so the involution will be the identity on these hypergraphs.

We start with removing a vertex 1. This vertex is akin to the root of all hypertrees. The next step is to utilise the alternative formulation for the Pfaffian using perfect matches. Let π be the perfect matching from I to $\bar{I}$. We define a function $f\colon I \to I \cup \{n\}$. This function connects matched pairs with another and, in combination with π, generates hyperedges $\{i, \pi(i), f(i)\}$.

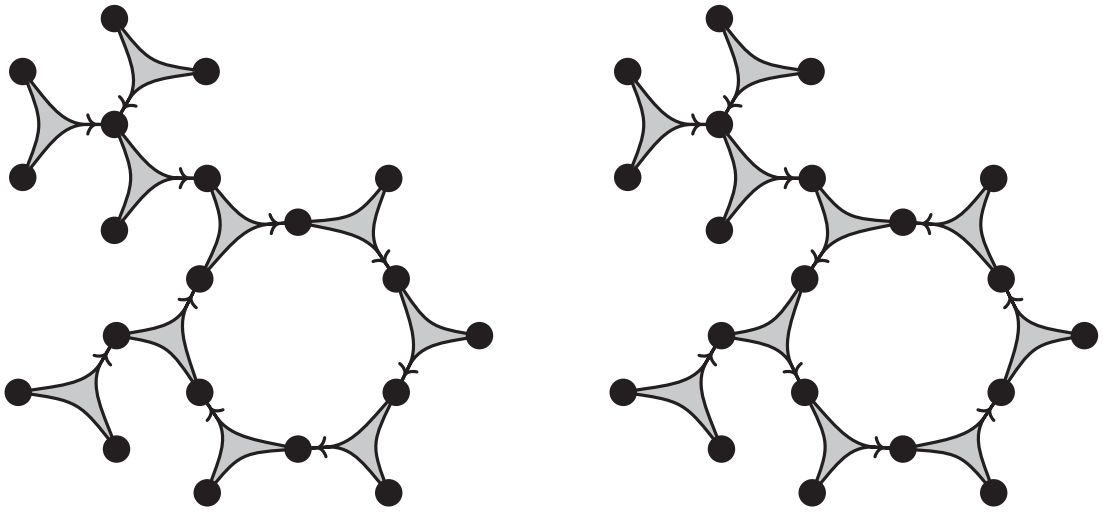


Figure 3.9: A 3-graph with a single cycle. The two vertices on the same side of the edges are considered to be matched. The involution ι changes the matching, such that the cycle changes its direction from clockwise to anticlockwise.

These hyperedges contribute to the sums in Λ_H with the sign $\operatorname{sgn}(o(\{i,\pi(i),f(i)\}))$ and their weight $w(\{i,\pi(i),f(i)\})$. Using f, we can swap the sum and product in the Pfaffian of Λ_H. For this we define a set of perfect matchings with functions f to complete them to 3-graphs. Let $M := \{(\pi,f) \mid I \subset [2n], |I| = n, \pi \in S_{I,\bar{I}}, i < \pi(i), f\colon\ I \to I \cup \{n\}\}$ be this set. So the Pfaffian can be expressed as

$$\begin{aligned}\operatorname{pf}(\Lambda_{H''}(1)) &= \sum_{\substack{\pi\in S_{I,\bar{I}}, i<\pi(i),\\ I\subset[2n],|I|=n}} (-1)^{\operatorname{cross}(\pi)} \prod_{i\in I}\sum_{e\in E} \operatorname{sgn}_e(i,\pi(i))w(e)\\ &= \sum_{(\pi,f)\in M} (-1)^{\operatorname{cross}(\pi)} \prod_{i\in I} \operatorname{sgn}_{\{i,\pi(i),f(i)\}}(i,\pi(i))w(\{i,\pi(i),f(i)\}).\end{aligned}$$

If this pair (π,f) generates a hypertree, then we want to preserve it, otherwise (π,f) should be paired up with $\iota(\pi,f) = (\pi',f')$, such that the sign is different. As the sign is dependant only on π, we define the involution by carefully choosing a beneficial π'. The only additional restriction is that there must exist an f', such that the hypergraph generated by (π',f') is the same as the one of (π,f).

Relabelling the vertices may change the sign as follows by Lemma 3.2.25, but it does so simultaneously for (π,f) and (π',f'). So let the vertices in the cycle be $[2k]$ and the restriction of matching π be from $I := \{1,\dots,k\}$ to $\bar{I} = \{k+1,\dots,2k\}$. The vertices in $\bar{I}$ form the core of the cycle, while the hanging tips are in I as is illustrated in Figure 3.9. The image of ι is thus defined as

$$\pi'(i) := \begin{cases}\pi(i) & \text{if } i > 2k,\\ f(i) & \text{otherwise,}\end{cases} \qquad f'(i) := \begin{cases}f(i) & \text{if } i > 2k,\\ \pi(i) & \text{otherwise.}\end{cases}$$

We now have to check that the sign changes. On one hand, the π' can be reconstructed by cyclically swapping the images. This requires $k-1$ swaps, and changes the sign by $(-1)^{k-1}$. On the other hand, we are comparing edges $\{i,\pi(i),f(i)\}$ and $\{i,\pi'(i),f'(i)\} = \{i,f(i),\pi(i)\}$. The edges are the same, but their orientation is reversed. With Lemma 3.2.16 we deduce that the sign changes by $(-1)^k$ if there are k edges in the cycle. Combined, this

gives a change of $(-1)^{2k-1} = -1$ and thus the contributions of hypergraphs given by (π, f) and $\iota(\pi, f)$ annihilate each other. What remains are all contributions of hypertrees. □

We have now proven the Pfaffian matrix tree theorem using two different methods. A. Abdesselam, who used the Grassman–Berezin calculus to prove a generalisation to the all minors matrix tree theorem on commutative semirings, also gives a generalisation of the Pfaffian matrix tree theorem using the same formalism. It is the main result of [Abd04] and titled "hyperpfaffian cactus theorem" as he interprets the hypertrees as cacti.

Theorem 3.2.30 (Hyperpfaffian Cactus Theorem) [Abd04, Theorem 2]**.** *Let H be a weighted odd hypergraph with weights w, then the weight of all odd hypertrees in H is given by*

$$\kappa^{[odd]}(H, w) = \int \mathrm{d}\chi_n \cdots \mathrm{d}\chi_1 \, \chi_i \exp\Big(\sum_{\substack{3 \le k \le n \\ k \text{ odd}}} \frac{1}{(k-1)!} \sum_{e=(\alpha_1,\ldots,\alpha_k)\in[n]^k} w(e) \chi_{\alpha_2} \cdots \chi_{\alpha_k} \Big).$$

This theorem is given without proof here, which can be found in [Abd04]. Comparing it to the Pfaffian matrix tree theorem gives many parallels: The exponent is due to the Berezin integral and can be considered the Pfaffian in the 3-uniform case, the inner sum adds all contributions of hyperedges, using the weight and—hidden in the order of the χ_{α_i}—the sign of the orientation of the hyperedge. This is due to the anti-commutative property expressed in (3.1.5), which introduces the sign of the required reordering.

A. Abdesselam also mentioned the "special case where all the y tensors [i.e. the edge-weights $w(e)$] are zero except for a specific odd integer k, $3 \le k \le n$" [Abd04, p. 65], which leads to a formula for the weight $\kappa^{[k]}(H, w)$ of all k-uniform hypertrees. He also noted that choosing $k = 3$ leads to the Pfaffian matrix tree theorem [Abd04, p. 66].

3.2.4 Complexity

Until now, the complexity of finding the number or weight of all trees in a graph was rather uninteresting. It is essentially as complex as computing a determinant, which can be done in polynomial time. Checking whether a graph contains a spanning tree is similarly easy as a matching algorithm can check for connectedness, which suffices for the existence. This problem is essentially the same for forests or directed graphs. Answering these questions for hypergraphs is not as easy as Example 3.2.11 shows. To discuss the complexity of these problems, we define[9] some complexity classes and formalise the problems.

A *decision problem* is a YES/NO-question. A decision problem is in P, respectively in NP iff for an input of size k there exists a deterministic program, respectively non-deterministic, such that the required number of calculation steps to answer this question is at most polynomial in k.

A *counting problem* is, in comparison to the decision problem, a question, which does not ask "does there exist", but instead asks "how many do exist". Thus, the counting problem corresponding to a given decision problem is in ♯P iff it counts the number of possible calculation paths to reach YES requiring at most polynomially many computation steps [Val79a]. Clearly, the counting problem is at least as hard as the corresponding decision problem as knowing the number of solution implies knowing of the existence of solutions.

[9] We abstain from defining them via one of the usual formalisms like computably enumerable sets, Turing machines or RAM machines, and instead use an intuitive approach via programming languages.

We are now ready to define the following decision problems:

i. k-Uniform Spanning Hypertree (k-SHT):
Given a k-uniform hypergraph, does there exist a spanning hypertree?

ii. Exact Cover by k-Sets (XkC):
Given sets X and C, such that $c \subseteq X$ and $|c| = k$ for all $c \in C$ holds, does there exist a subset C' of C, such that $\bigcup_{c \in C'} c = X$ and $c \cap c' = \emptyset$ for all $c, c' \in C'$?

As mentioned before, the matrix tree theorem is based on evaluating the determinant, for which polynomial algorithms exist [KV05]. This theorem even returns the number of solutions, so it is the corresponding counting problem to 2-SHT and in P. A slight modification of the determinant—the permanent—is known to be ♯P-complete and thus one of the hardest problems in ♯P [Val79a]. Therefore calculating the bideterminant is at least as hard as a simple reduction from it is given by $\operatorname{perm}(A) = \det^+(A) \oplus \det^-(A)$. So, by our methods, we can deduce that the matrix tree theorem on commutative semirings is ♯P-complete as knowing the bideterminant implies knowing $\det^+(A)$ and $\det^-(A)$.

Now the questions arises: How complex is it to determine the existence and number of k-uniform hypertrees in a given hypergraph? To answer this question, we will find a reduction from XkC to $(k+1)$-SHT. Exact Cover, without the restriction on the size of the sets, is one of R. M. Karp's 21 NP-complete problems [Kar72]. X3C is also NP-complete [GJ79] and thus XkC for $k \geq 3$ as we could add superfluous vertices to every hyperedge to transform X3C to XkC.

Proposition 3.2.31 [Car+08; DF95, Theorem 4]**.** *k-SHT is NP-complete for $k \geq 4$ and in P for $k \leq 2$.*

Proof. The cases $k = 0$ and 1 are trivial. If $k = 2$, then we can compute the existence via the determinant, which is in P [KV05].

We now formulate the reduction from XkC to $(k+1)$-SHT. The goal is to find an exact cover of the set via finding a hypertree. Every subset contains k vertices, whereas every edge contains $k+1$ vertices. This is due to the addition of a single vertex ∞, which is contained in every edge. The other vertices of the edges consist of the subsets. Thus, if a hypertree is found, the sets corresponding to these edges are the desired cover. So $(k+1)$-SHT is at least as hard as XkC.

On the other hand, $(k+1)$-SHT cannot be more complex as checking whether a given solution is correct can be done in polynomial time via a marking algorithm. This algorithm recursively traverses all connected edges and marks its vertices as visited. If any edge contains a marked vertex (apart from its root) the hypergraph contains a cycle. If the algorithm terminates and not all vertices were visited, the graph is not connected. This algorithm is in NPsince it non-deterministically guesses the solution in polynomial time and then checks its correctness. □

As X2C is essentially finding a perfect matching, where the sets of X2C are interpreted as edges, this problem can be solved polynomially by J. Edmonds's blossom algorithm [Edm65]. So the reduction of X2C to 3-SHT would only yield that 3-SHT is at least polynomial in its complexity. However, the corresponding counting problem is ♯P-complete [Val79b, Problem 2]. Thus, using the same reduction as above, we get the following proposition:

Proposition 3.2.32. *The corresponding counting problem to k-SHT is ♯P-complete for $k \geq 3$ and in P for $k \leq 2$.*

So all that is missing is the complexity of 3-SHT. We omit the proof here, but it can be shown that it can be solved polynomially [LP86]. Summarising, we find that checking the existence is "easy" for graphs and 3-graphs, but otherwise "hard". Usually, we are more interested in finding the weight of all hypertrees, and thus do not only care about the existence, but also the count. This changes nothing for graphs, but does so for hypergraphs as the problem is now ♯P-complete.

4 Conclusion and Outlook

G. R. Kirchhoff's matrix tree theorem was born out of physical properties of electric circuits. Abstraction and new notation, like the determinant, raised it to a fully fleshed theorem, which entered the mathematical canon at a time as the concept of trees was just being formulated.

While the historical approach gives much insight in the structure of the underlying graph, modern approaches often use the Cauchy–Binet formula as proving the theorem does not require any additional knowledge about the graph apart from the Laplacian matrix. Unfortunately, the insight is lost with it. While Chapter 1 presents all that is usually required from the matrix tree theorem, Chapter 2 takes a closer look at different approaches. These might seem cumbersome in comparison to the Cauchy–Binet formula, but all highlight different aspects of the internal structure.

The historical approach, described in Section 2.1, features a seemingly roundabout way of proving the theorem via closed walks and using incidence matrices instead of the Laplacian. It is thus much closer to G. R. Kirchhoff's circuit laws. Combining this approach with the modern theorem enables the formulation of two separate matrix tree theorems: On one hand, we get the classical matrix tree theorem, but on the other hand, we can also derive the cycle theorem. The matrix tree theorem uses the current law, while the cycle theorem uses the voltage law. This is especially useful if the graph consists of few, but long cycles, and thus interesting for organic chemists, who often have to deal with these kinds of graphs in the form of molecules.

The approach explored next was via a sign-reversing involution, formulated by S. Chaiken. The idea is as ingenious as it is simple: We construct all subgraphs and assign them their weight. A result of calculating the weight of all trees is the introduction of a seemingly arbitrary sign. This sign is used by the involution as it is also assigned to subgraphs containing cycles. The proof takes the permutations π of the determinant and interprets them as fixing edges for $\pi(i) \neq i$, the fixed points are considered free choices. Now a graph with a cycle appears multiple times: A choice exists for every cycle whether it has been fixed or not. Since the sign in the determinant is dependant on the permutation π, it changes accordingly. It turns out that fixing a cycle inverts the sign. Thus, a graph containing a cycle annihilates itself. What remains are the trees. A very similar approach can be taken via the inclusion–exclusion principle, which operates not on the graphs themselves, but on the sets of graphs with the same fixed cycles.

Another approach is given by induction on the structure of the graph. A graph consists of vertices and edges in between, so it is natural to try to reduce the graph to smaller ones. The two obvious choices are either removing an edge if it is not needed, or contracting it, so its two endpoints merge if it is always used. This exactly describes the two operations in the deletion–contraction induction: Trees avoiding a specific edge must be connected in another way and thus are trees, even if the edge is removed. On the other hand, if a tree uses an edge, this edge gives no additional information and a similar tree exists in its place if the edge is contracted. Only the weight of the edge has to be considered. So the trees of a graph can be grouped into two classes: those that use a given edge, and those that do not. This encapsulates the induction step. The induction basis consists of graphs without

edges, which are only trees if they contain one vertex.

Using Markov chains to prove the matrix tree theorem takes on only a minor part in this thesis as it uses completely different notation and concepts. However, especially because of this, it must not be omitted as it shows a strong connection to a completely different field of mathematics. The idea is to determine a root and then starting a random walk on the graph. This walk will be used to form the branch of a tree, but as it may contain loops, these have to be removed. So the probability of choosing a specific walk without loops consists of the probabilities of the walks with their loops removed. If the first walk is chosen, the visited vertices are added to the tree. Another walk is generated in a similar sense. It stops if it reaches an already visited vertex. If all vertices were visited, the algorithm ends and the output is the collection of loop-erased walks, which combined form a tree. The incredible property is now that every tree is generated with equal probability, which is exactly the inverse of the usual result of the matrix tree theorem.

All these approaches allow us to prove the same matrix tree theorem, but each also force a generalisation upon us: The historical approach gives the cycle theorem almost in passing, the sign-reversing involution internally uses directed graphs even for the undirected theorem, deletion–contraction automatically contracts all edges between two vertices and thus uses multigraphs, it can also contract subgraphs and, if they are forced, allows us to count forests instead of trees, and Markov chains require a transition probability, which can be changed arbitrarily, resulting in weighted graphs. Every one of those generalisations can be combined in one single theorem, given in Section 2.5.1. This theorem is the culmination of the classical approaches.

Of course, modifying the premise of the matrix tree theorem also leads to many useful theorems, of which some are presented in Section 2.5. Each one of them with a plethora of generalisations, special cases and applications.

The last chapter consists of two generalisations, which do not follow by themselves. After generalising the matrix tree theorem to weighted graphs, it directly follows for weights in rings. However, generalising the theorem to semirings implies much work, all stemming from the lack of an additive inverse. Thus, we redefine the determinant in two halves, called the positive and negative bideterminant. They are then paired up in the symmetric extension of the semiring, which simulates the additive inverse, to form the bideterminant. Using this bideterminant, we can prove the all minors matrix tree theorem on commutative semirings via a sign-reversing involution.

The other generalisation shown in this thesis is for 3-graphs, which are hypergraphs, in which every edge has exactly three endpoints. Hypertrees are still identified as connected subhypergraphs with no cycles. The determinant again has to be swapped out with a different function, called the Pfaffian. The Pfaffian is only defined for skew-symmetric matrices and acts similarly to the square root of the determinant. We also cannot use the Laplacian as now several different hyperedges may connect two vertices i and j. Thus, we define the hyper-Laplacian, which sums up all these edges. This leads to edges $\{i,j,k\}$ contributing to entries (i,j), (j,k), (i,k) and (j,i), (k,j), (k,i). While these all correspond to the same edge, they are interpreted differently and lead to a different sign. This sign can be described by introducing an orientation on all edges. With all those preparation steps done, two proofs of the Pfaffian matrix tree theorem are presented: The first uses deletion–contraction and the second a sign-reversing involution. Both proofs thus are closely related to the respective proofs of the matrix tree theorem.

Further generalisations of those presented in Chapter 3 are mentioned, both of which are proven using the Grassman–Berezin calculus. Only a short overview is given, to enable the formulation of the theorems. While A. Abdesselam's paper and this thesis discuss

$\kappa^{[3]}$ and $\kappa^{[\text{odd}]}$, there do not seem to exist similar theorems for $\kappa^{[\text{even}]}$ or $\kappa^{[2k]}$ with $k > 1$. However, A. Abdesselam's paper [Abd04] marks the border of this thesis' scope.

Of course, A. Abdesselam's paper does not mark the endpoint in generalisations of the matrix tree theorem. A. Duval, C. Klivans, and J. Martin formulated the matrix tree theorem for simplicial complexes, which can be interpreted as a class of special hypergraphs. Much work has also been done to formulate higher order matrix tree theorems [BS06; BPT15; DKM16; Aal+18].

A generalisation related to cacti can also be reached by proving the Good–Lagrange formula, presented in [Bou+03; GK97]. These include coloured vertices, which are then used to form k-uniform graphs, for which a matrix tree theorem is proven. Another way of generalising graphs is via matroid theory, which readily gives the cycle theorem. While the results might be well known and discussed in [Che97], this formalism was not used in this thesis. However, it gives further insight in the connections of the graph and its dual interpretations. This connection is hinted at in Section 2.3.2. A similar parallel to the cycle theorem and the cycle space uses the so-called cut space. Instead of cycles, the cut space considers edges as vertices and thus operates on the line graph.

It appears that not much literature exists on the connection between the cycle or cut space and their use in combination with hypergraphs. As the dual graph is only defined for planar graphs, the connection shown in Section 2.3.2 cannot be generalised directly. If one tries to find the dual of a non-planar graph using the cycle space, the resulting graph is a hypergraph. While this connection might be implicitly given, there does not seem to exist a visual connection.

In Section 2.5.2 we show an edge version of the matrix tree theorem. There has also been development, which led to a generalisation to the parallel theorem to the all minors matrix tree theorem [BGK00] not explored here. Another aspect not discussed here is the application of the proof-concepts to other theorems. An example for this is the Lindström–Gessel–Viennot lemma, which can be proven using a sign-reversing involution [GVM89], or Knuth's theorem, which provides a similar statement to the matrix tree theorem for line graphs [Knu67].

Bibliography

[Aal+18] Ghodratollah Aalipour, Art M. Duval, Woong Kook, Kang-Ju Lee, and Jeremy L. Martin. "A weighted cellular matrix-tree theorem, with applications to complete colorful and cubical complexes". In: *Journal of Combinatorial Theory, Series A* 158 (2018), pp. 362–386. DOI: 10.1016/j.jcta.2018.03.009.

[Abd04] Abdelmalek Abdesselam. "The Grassmann–Berezin calculus and theorems of the matrix-tree type". In: *Advances in Applied Mathematics* 33.1 (July 2004), pp. 51–70. DOI: 10.1016/j.aam.2003.07.002.

[Ahr97] Wilhelm Ahrens. "Ueber das Gleichungssystem einer Kirchhoff'schen galvanischen Stromverzweigung". In: *Mathematische Annalen* 49.2 (June 1897), pp. 311–324. DOI: 10.1007/BF01444208.

[Asl96] Helmer Aslaksen. "Quaternionic determinants". In: *The Mathematical Intelligencer* 18.3 (June 1996), pp. 57–65. DOI: 10.1007/BF03024312.

[AT89] Venkat Anantharam and Pantelis Tsoucas. "A proof of the Markov chain tree theorem". In: *Statistics & Probability Letters* 8.2 (1989), pp. 189–192. DOI: 10.1016/0167-7152(89)90016-3.

[Ber89] Claude Berge. *Hypergraphs: Combinatorics of Finite Sets*. Ed. by Claude Berge. Vol. 45. North-Holland Mathematical Library. Amsterdam, NL: Elesvier Science Publishers, 1989. ISBN: 0-444-87489-5.

[BGK00] Ravindra B. Bapat, Jerrold W. Grossman, and Devadatta M. Kulkarni. "Edge version of the matrix tree theorem for trees". In: *Linear and Multilinear Algebra* 47.3 (2000), pp. 217–229. DOI: 10.1080/03081080008818646.

[Big93] Norman Biggs. *Algebraic Graph Theory*. 2nd. New York, NY: Cambridge University Press, 1993. ISBN: 0-521-45897-8.

[Bor60] Carl Wilhelm Borchardt. "Ueber eine der Interpolation entsprechende Darstellung der Eliminations-Resultante". In: *Journal für die reine und angewandte Mathematik* 1860.57 (1860). Ed. by Carl Wilhelm Borchardt, pp. 111–121. DOI: 10.1515/crll.1860.57.111.

[Bou+03] Michel Bousquet, Cedric Chauve, Gilbert Labelle, and Pierre Leroux. "Two bijective proofs for the arborescent form of the Good–Lagrange formula and some applications to colored rooted trees and cacti". In: *Theoretical Computer Science* 307.2 (2003), pp. 277–302. DOI: 10.1016/S0304-3975(03)00220-2.

[Bou51] Samuel Bourne. "The Jacobson radical of a semiring". In: *Proceedings of the National Academy of Sciences*. Vol. 37. 3. National Academy of Sciences, Mar. 1951, pp. 163–170. DOI: 10.1073/pnas.37.3.163.

[BPT15] Yurii Burman, Andrey Ploskonosov, and Anastasia Trofimova. "Matrix-tree theorems and discrete path integration". In: *Linear Algebra and its Applications* 466 (Feb. 2015), pp. 64–82. DOI: 10.1016/j.laa.2014.09.049.

[Bro+40] Rowland Leonard Brooks, Cedric Austen Bardell Smith, Arthur Harold Stone, and William Thomas Tutte. "The dissection of rectangles into squares". In: *Duke Mathematical Journal* 7.1 (1940), pp. 312–340. DOI: 10.1215/S0012-7094-40-00718-9.

[Bro10] Francis C. S. Brown. "On the periods of some Feynman integrals". 2010. arXiv: 0910.0114v2 [math.AG] (visited on Mar. 21, 2022).

[BS06] Yurii Burman and Boris Shapiro. "Around matrix-tree theorem". In: *Mathematical Research Letters* 13.5 (Jan. 2006), pp. 761–774. DOI: 10.4310/MRL.2006.v13.n5.a7.

[BW10] Christian Bogner and Stefan Weinzierl. "Feynman graph polynomials". In: *International Journal of Modern Physics A* 25.13 (May 2010), pp. 2585–2618. DOI: 10.1142/s0217751x10049438.

[Cam08] Peter Jephson Cameron. *Matrices with Zero Row and Column Sum*. 2008 URL: www.maths.qmul.ac.uk/~pjc/odds/zero.pdf (visited on Mar. 21, 2022).

[Car+08] Sergio Caracciolo, Gregor Masbaum, Alan Sokal, and Andrea Sportiello. "A randomized polynomial-time algorithm for the spanning hypertree problem on 3-uniform hypergraphs". 2008. arXiv: 0812.3593v1 [cs.CC] (visited on Mar. 21, 2022).

[Cay57] Arthur Cayley. "On the theory of the analytical forms called trees". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*. 4th ser. 13.85 (Mar. 1857). Ed. by David Brewster, Richard Taylor, Richard Phillips, Robert Kane, and John Tyndall, pp. 172–176. DOI: 10.1080/14786445708642275.

[Cay75] Arthur Cayley. "On the analytical forms called trees, with application to the theory of chemical combinations". In: *Report of the British Association for the Advancement of Science*. London, UK: Taylor & Francis, Aug. 1875, pp. 257–305 URL: https://www.biodiversitylibrary.org/page/29863653 (visited on Mar. 21, 2022).

[Cay89] Arthur Cayley. "A theorem on trees". In: *The Quarterly Journal of Pure and Applied Mathematics* 23 (1889), pp. 376–378 URL: https://gdz.sub.uni-goettingen.de/id/PPN600494829_0023?tify=%7B%22pages%22:[396,397,398]%7D (visited on Mar. 21, 2022).

[Cha82] Seth Chaiken. "A combinatorial proof of the all minors matrix tree theorem". In: *SIAM Journal on Algebraic and Discrete Methods* 3.3 (1982), pp. 319–329. DOI: 10.1137/0603033.

[Che97] Wai-Kai Chen. *Graph Theory and Its Engineering Applications*. Advanced Series in Electrical and Computer Engineering. Singapore, SG: World Scientific, 1997. ISBN: 978-981-02-1859-1.

[Chu93] Fan R. K. Chung. "The Laplacian of a hypergraph". In: *Expanding Graphs: Proceedings of a DIMACS Workshop, May 11–14, 1992*. Ed. by Joel Friedman. Vol. 10. DIMACS. Providence, RI, 1993. ISBN: 978-0-8218-6602-3. DOI: 10.1090/dimacs/010.

[DF95] Lars Døvling Andersen and Herbert Fleischner. "The NP-completeness of finding A-trails in Eulerian graphs and of finding spanning trees in hypergraphs". In: *Discrete Applied Mathematics* 59.3 (May 1995), pp. 203–214. DOI: 10.1016/0166-218X(95)80001-K.

[DKM09] Art Duval, Caroline Klivans, and Jeremy Martin. "Simplicial matrix-tree theorems". In: *Transactions of the American Mathematical Society* 361.11 (Nov. 2009), pp. 6073–6114. DOI: 10.1090/S0002-9947-09-04898-3.

[DKM16] Art M. Duval, Caroline J. Klivans, and Jeremy L. Martin. "Simplicial and cellular trees". In: *Recent Trends in Combinatorics. The IMA Volumes in Mathematics and its Applications*. Ed. by Andrew Beveridge, Jerrold R. Griggs, Leslie Hogben, Gregg Musiker, and Prasad Tetali. Vol. 159. Cham, CH: Springer, 2016, pp. 713–752. ISBN: 978-3-319-24298-9. DOI: 10.1007/978-3-319-24298-9_28.

[Duc78] Pierre Duchet. "Propriété de Helly et problèmes de représentation". In: *Problèmes Combinatoires et Théorie des Graphes*. Vol. 260. Paris, FR: CNRS Éditions, 1978, pp. 117–118. ISBN: 978-2-222-02070-7.

[Edm65] Jack Edmonds. "Paths, trees, and flowers". In: *Canadian Journal of Mathematics* 17 (1965), pp. 449–467. DOI: 10.4153/CJM-1965-045-4.

[Eul41] Leonhard Euler. "Solutio problematis ad geometriam situs pertinentis". In: *Commentarii Academiae scientiarum imperialis Petropolitanae* 8 (1741), pp. 128–140 URL: https://www.biodiversitylibrary.org/page/10125793 (visited on Mar. 21, 2022).

[Gau67] Carl Friedrich Gauss. *Mathematische Physik*. Vol. 5. Werke. Göttingen, DE: Königliche Gesellschaft der Wissenschaften, 1867 URL: https://gdz.sub.uni-goettingen.de/id/PPN236006339?tify=%7B%22pages%22:[606,607,608]%7D (visited on Mar. 21, 2022).

[Gil19] Maria Gillespie. *A Linear Algebra-Free Proof of the Matrix-Tree Theorem*. Dec. 2019 URL: https://www.mathematicalgemstones.com/gemstones/a-linear-algebra-free-proof-of-the-matrix-tree-theorem/ (visited on Mar. 21, 2022).

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979. ISBN: 978-0-7167-1045-5.

[GK97] Ian P. Goulden and Dhananjay M. Kulkarni. "Multivariable Lagrange inversion, Gessel–Viennot cancellation, and the matrix tree theorem". In: *Journal of Combinatorial Theory, Series A* 80.2 (1997), pp. 295–308. DOI: 10.1006/jcta.1997.2827.

[Gol03] Jonathan S. Golan. *Semirings and Affine Equations over Them: Theory and Applications*. Vol. 556. Mathematics and its Applications. Dordrecht, NL: Springer-Science+Business Media, 2003. ISBN: 978-94-017-0383-3. DOI: 10.1007/978-94-017-0383-3.

[GR91] Izrail Moiseevich Gel'fand and Vladimir Retakh. "Determinants of matrices over noncommutative rings". In: *Functional Analysis and Its Applications* 25 (Apr. 1991), pp. 91–102. DOI: 10.1007/BF01079588.

[GVM89] Ira Gessel, Xavier Viennot, and Departement Mathematiques. "Determinants, paths, and plane partitions". Sept. 1989 URL: https://people.brandeis.edu/~gessel/homepage/papers/pp.pdf (visited on Mar. 21, 2022).

[GYZ13] Jonathan L. Gross, Jay Yellen, and Ping Zhang. *Handbook of Graph Theory*. 2nd. Boca Raton, FL: Chapman & HallCRC, 2013. ISBN: 978-1-4398-8018-0.

[Hab15] Howard E. Haber. *Notes on Antisymmetric Matrices and the Pfaffian*. Jan. 2015 URL: http://scipp.ucsc.edu/~haber/webpage/pfaffian2.pdf (visited on Mar. 21, 2022).

[HJ12] Roger Alan Horn and Charles Royal Johnson. *Matrix Analysis*. 2nd. New York, NY: Cambridge University Press, Dec. 2012. ISBN: 978-0-521-54823-6.

[HM19] Keivan Hassani Monfared and Sudipta Mallik. "An analog of matrix tree theorem for signless Laplacians". In: *Linear Algebra and its Applications* 560 (Jan. 2019), pp. 43–55. DOI: 10.1016/j.laa.2018.09.016.

[HR04] Scott Hirschman and Victor Reiner. "Note on the Pfaffian matrix-tree theorem". In: *Graphs and Combinatorics* 20 (Mar. 2004), pp. 59–63. DOI: 10.1007/s00373-003-0537-9.

[Hut67] Hermann Hutschenreuther. "Einfacher Beweis des ‚Matrix-Gerüst-Satzes' der Netzwerktheorie". In: *Wissenschaftliche Zeitschrift der Technischen Hochschule Ilmenau* 13.4 (1967), pp. 403–404.

[Jan+15] Dušanka Janežič, Ante Miličević, Sonja Nikolić, and Nenad Trinajstić. *Graph Theoretical Matrices in Chemistry*. 2nd. Boca Raton, FL: CRC Press, Apr. 2015. ISBN: 978-1-4987-0122-8. DOI: 10.1201/b18389.

[Kar72] Richard M. Karp. "Reducibility among combinatorial problems". In: *Complexity of Computer Computations*. Ed. by Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger. Boston, MA: Springer, 1972, pp. 85–103. ISBN: 978-1-4684-2001-2. DOI: 10.1007/978-1-4684-2001-2_9.

[KGW10] Alex Kruckman, Amy Greenwald, and John R. Wicks. *An Elementary Proof of the Markov Chain Tree Theorem*. 2010 URL: https://math.berkeley.edu/~kruckman/MCTT.pdf (visited on Mar. 21, 2022).

[Kir+04] Edward Cameron Kirby, Douglas Jay Klein, Roger B. Mallion, Paul Pollak, and Horst Sachs. "A theorem for counting spanning trees in general chemical graphs and its particular application to toroidal fullerenes". In: *Croatica Chemica Acta* 77.1–2 (2004), pp. 263–278 URL: https://hrcak.srce.hr/en/102674 (visited on Mar. 21, 2022).

[Kir45] Gustav Robert Kirchhoff. "Ueber den Durchgang eines elektrischen Stromes durch eine Ebene, insbesondere durch eine kreisförmige". In: *Annalen der Physik und Chemie* 140.4 (1845). Ed. by Johann Christian Poggendorf, pp. 497–514 URL: https://www.digitale-sammlungen.de/de/view/bsb10130377?page=515 (visited on Mar. 21, 2022).

[Kir47] Gustav Robert Kirchhoff. "Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird". In: *Annalen der Physik und Chemie* 148.12 (1847). Ed. by Johann Christian Poggendorf, pp. 497–508 URL: https://www.digitale-sammlungen.de/de/view/bsb10130385?page=511 (visited on Mar. 21, 2022).

[Knu67] Donald Ervin Knuth. "Oriented subtrees of an arc digraph". In: *Journal of Combinatorial Theory* 3.4 (1967), pp. 309–314. DOI: 10.1016/S0021-9800(67)80101-7.

[KRS13] Michael J. Kozdron, Larissa M. Richards, and Daniel W. Stroock. "Determinants, their applications to Markov processes, and a random walk proof of Kirchhoff's matrix tree theorem". 2013. arXiv: 1306.2059v1 [math.PR] (visited on Mar. 21, 2022).

[KV05] Erich Kaltofen and Gilles Villard. "On the complexity of computing determinants". In: *Computational Complexity* 13 (Feb. 2005), pp. 91–130. DOI: 10.1007/s00037-004-0185-3.

[Lew82] Mordechai Lewin. "A generalization of the matrix-tree theorem". In: *Mathematische Zeitschrift* 181.1 (Mar. 1982), pp. 55–70. DOI: 10.1007/BF01214981.

[LL10] Gregory F. Lawler and Vlada Limic. *Random Walk: A Modern Introduction*. Cambridge Studies in Advanced Mathematics. Cambridge, UK: Cambridge University Press, 2010. DOI: 10.1017/CBO9780511750854.

[LP86] Laszlo Lovász and Michael D. Plummer. *Matching Theory*. Vol. 121. North-Holland Mathematics Studies. Providence, RI: Elsevier Science Publishers, 1986. ISBN: 978-0-444-87916-5.

[May80] Kenneth Ownsworth May. "Gauss, Carl Friedrich". In: *Emil Fischer–Gottlieb Haberlandt*. Ed. by Charles Coulston Gillispie. Vol. 5. Dictionary of Scientific Biography. New York, NY: Charles Scribner's Sons, 1970–1980, pp. 298–315. ISBN: 0-684-16962-2.

[Mer89] Russell Merris. "An edge version of the matrix-tree theorem and the Wiener index". In: *Linear and Multilinear Algebra* 25.4 (Oct. 1989), pp. 291–296. DOI: 10.1080/03081088908817955.

[Min97] Michel Minoux. "Bideterminants, arborescences and extension of the matrix-tree theorem to semirings". In: *Discrete Mathematics* 171.1–3 (June 1997), pp. 191–200. DOI: 10.1016/S0012-365X(96)00015-5.

[Min99] Michel Minoux. "A generalization of the all minors tree theorem to semirings". In: *Discrete Mathematics* 199.1–3 (Mar. 1999), pp. 139–150. DOI: 10.1016/S0012-365X(98)00204-0.

[Moo70] John W. Moon. *Counting Labelled Trees*. Vol. 1. Canadian Mathematical Monographs. Montreal, CA: Canadian Mathematical Congress, 1970.

[Moo94] John W. Moon. "Some determinant expansions and the matrix-tree theorem". In: *Discrete Mathematics* 124.1–3 (Jan. 1994), pp. 163–171. DOI: 10.1016/0012-365X(92)00059-Z.

[MV02] Gregor Masbaum and Arkady Vaintrob. "A new matrix-tree theorem". In: *International Mathematics Research Notices* 2002.27 (Jan. 2002), pp. 1397–1426. DOI: 10.1155/S1073792802111044.

[NO98] John W. Negele and Henri Orland. *Quantum Many-Particle Systems*. Advanced Book Classics. Boca Raton, FL: Westview Press, Nov. 1998. ISBN: 978-0-7382-0052-1. DOI: 10.1201/9780429497926.

[Orl78] James Berger Orlin. "Line-digraphs, arborescences, and theorems of Tutte and Knuth". In: *Journal of Combinatorial Theory, Series B* 25.2 (1978), pp. 187–198. DOI: 10.1016/0095-8956(78)90038-2.

[PT18] Jim Pitman and Wenpin Tang. "Tree formulas, mean first passage times and Kemeny's constant of a Markov chain". In: *Bernoulli* 24.3 (Aug. 2018), pp. 1942–1972. DOI: 10.3150/16-BEJ916.

[RK02] Dennis H. Rouvray and R. Bruce King. *Topology in Chemistry: Discrete Mathematics of Molecules*. Chichester, UK: Woodhead Publishing, 2002. ISBN: 978-1-898563-76-1.

[RR12] Nathan Reff and Lucas J. Rusnak. "An oriented hypergraphic approach to algebraic graph theory". In: *Linear Algebra and its Applications* 437.9 (Nov. 2012), pp. 2262–2270. DOI: 10.1016/j.laa.2012.06.011.

[Sco62] Hubert Ian Scoins. "The number of trees with nodes of alternate parity". In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 58. 1. Cambridge University Press, Jan. 1962, pp. 12–16. DOI: 10.1017/S0305004100036173.

[Ste90] John R. Stembridge. "Nonintersecting paths, Pfaffians, and plane partitions". In: *Advances in Mathematics* 83.1 (Sept. 1990), pp. 96–131. DOI: 10.1016/0001-8708(90)90070-4.

[Syl50] James Joseph Sylvester. "Additions to the articles in the September number of this journal, 'On a new class of theorems,' and on *Pascal's* theorem". In: *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science* 37.251 (Nov. 1850). Ed. by David Brewster, Richard Taylor, Richard Phillips, and Robert Kane, pp. 363–370. DOI: 10.1080/14786445008646629.

[Tre54] Horace M. Trent. "A note on the enumeration and listing of all possible trees in a connected linear graph". In: 40.10 (Oct. 1954), pp. 1004–1007. DOI: 10.1073/pnas.40.10.1004.

[Tut48] William T. Tutte. "The dissection of equilateral triangles into equilateral triangles". In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 44. 4. Cambridge University Press, 1948, pp. 463–482. DOI: 10.1017/S030500410002449X.

[Tut65] William T. Tutte. "Lectures on matroids". In: *Journal of Research* 69B.1–2 (Jan. 1965), pp. 1–47. DOI: 10.6028/jres.069B.001.

[Val79a] Leslie Gabriel Valiant. "The complexity of computing the permanent". In: *Theoretical Computer Science* 8.2 (1979), pp. 189–201. DOI: 10.1016/0304-3975(79)90044-6.

[Val79b] Leslie Gabriel Valiant. "The complexity of enumeration and reliability problems". In: *SIAM Journal on Computing* 8.3 (1979), pp. 410–421. DOI: 10.1137/0208032.

[Ver16] Jacques Verstraëte. "Extremal problems for cycles in graphs". In: *Recent Trends in Combinatorics*. Ed. by Andrew Beveridge, Jerrold R. Griggs, Leslie Hogben, Gregg Musiker, and Prasad Tetali. Cham, CH: Springer International Publishing, Apr. 2016, pp. 83–116. ISBN: 978-3-319-24298-9. DOI: 10.1007/978-3-319-24298-9_4.

[Wie47] Harry Wiener. "Structural determination of paraffin boiling points". In: *Journal of the American Chemical Society* 69.1 (Jan. 1947), pp. 17–20. DOI: 10.1021/ja01193a005.

[Wil96] David Bruce Wilson. "Generating random spanning trees more quickly than the cover time". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, PA: Association for Computing Machinery, May 1996, pp. 296–303. ISBN: 978-0-89791-785-8. DOI: 10.1145/237814.237880.

Index