Generalized bijective Maps between G-parking functions, spanning trees, and the Tutte polynomial

by

Carrie Frizzell

BA, MidAmerica Nazarene University, 2015

MS, Kansas State University, 2018

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Mathematics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2021

Abstract

We introduce an object called a tree growing sequence (TGS) in an effort to generalize bijective correspondences between G-parking functions, spanning trees, and the multiset of monomials of the Tutte polynomial of a graph G. A tree growing sequence determines an algorithm which can be applied to a single function, or to the set $\mathcal{P}_{G,q}$ of G-parking functions. When the latter is chosen, the algorithm uses splitting operations - inspired by the recursive definition of the Tutte polynomial - to partition $\mathcal{P}_{G,q}$. The result of the TGS algorithm is a pair of bijective maps τ and ρ from $\mathcal{P}_{G,q}$ to the spanning trees of G and Tutte monomials, respectively. The algorithm can also be viewed as a way to classify maps τ that have a coherence property: the splitting operations give rise to a natural bijective map ρ from $\mathcal{P}_{G,q}$ to the multi-set of terms of $T(G;x,y)$. We compare the TGS algorithm to Dhar's algorithm and the family of bijections found by Chebikin and Pylyavskyy in 2005, and obtain commutative diagrams to describe our comparisons. Additionally, we compute the Tutte polynomial of a zonotopal tiling using splitting operations analogous to those in the TGS algorithm.

Generalized bijective maps between G-parking functions, spanning trees, and the Tutte polynomial

by

Carrie R. Frizzell

BA, MidAmerica Nazarene University, 2015

MS, Kansas State University, 2018

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Deparment of Mathematics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2021

Approved by:

Major Professor
Ilia Zharkov

Copyright

Abstract

We introduce an object called a tree growing sequence (TGS) in an effort to generalize bijective correspondences between G-parking functions, spanning trees, and the multiset of monomials of the Tutte polynomial of a graph G. A tree growing sequence determines an algorithm which can be applied to a single function, or to the set $\mathcal{P}_{G,q}$ of G-parking functions. When the latter is chosen, the algorithm uses splitting operations - inspired by the recursive definition of the Tutte polynomial - to partition $\mathcal{P}_{G,q}$. The result of the TGS algorithm is a pair of bijective maps τ and ρ from $\mathcal{P}_{G,q}$ to the spanning trees of G and Tutte monomials, respectively. The algorithm can also be viewed as a way to classify maps τ that have a coherence property: the splitting operations give rise to a natural bijective map ρ from $\mathcal{P}_{G,q}$ to the multi-set of terms of $T(G;x,y)$. We compare the TGS algorithm to Dhar's algorithm and the family of bijections found by Chebikin and Pylyavskyy in 2005, and obtain commutative diagrams to describe our comparisons. Additionally, we compute the Tutte polynomial of a zonotopal tiling using splitting operations analogous to those in the TGS algorithm.

Table of Contents

List of Figures

Chapter 1

Introduction

To fix notation, given a multigraph $G = (V, E)$, label the vertices $V = \{q, v_1, \ldots, v_n\}$, where q is the root. The vertex and edge set will often be specified by $V(G), E(G)$ in context. If there are multiple edges between two vertices, order them. In each rooted subtree T of G, we direct edges toward the root. When necessary, $h(e)$ and $t(e)$ are used for the head and tail of a directed edge $e = (h(e), t(e))$. Recall that a **spanning tree** of G is a spanning, connected subgraph with $|V(G)| - 1$ edges.

Definition 1.1: The **outdegree with respect to** $A \subseteq V$, denoted $outdeg_A(v)$, is the number of neighbors of v not in $A \subseteq V$, with multiplicity.

Definition 1.2: A G**-parking function** is a function $f : V(G) - \{q\} \to \mathbb{Z}_{\geq 0}$ such that any subset $A \subseteq V - \{q\}$ contains a vertex v with $0 \leq f(v) < outdeg_A(v)$.

We write $f = (f(v_1), \ldots, f(v_n))$. Let $\mathcal{P}_{G,q}$ denote the set of parking functions on G with respect to q. Let $G - e$ mean deleting the edge e from G. Contracting G at e means to delete e, then identify the endpoints of e. Denote contraction by G/e.

Definition 1.3: The **Tutte polynomial** $T(G; x, y)$ of G is the universal *Tutte-Grothendieck graph isomorphism invariant* satisfying the following deletion/contraction principal, and defining $T(\bullet;x,y)=1$, $\bullet$ the graph with one vertex.

$$T(G;x,y)=\begin{cases} yT(G-e;x,y) & e \text{ a loop} \\ xT(G/e;x,y) & e \text{ a bridge} \\ T(G-e;x,y)+T(G/e;x,y) & \text{otherwise} \end{cases} \tag{1.1}$$

An equivalent definition is a closed formula over all spanning subgraphs of G. Let $c(A)$ be the number of connected components of a spanning subgraph A. Then

$$T(G;x,y)=\sum_{A\subseteq G}(x-1)^{c(A)-c(G)}(y-1)^{|E(A)|+c(A)-|V|} \tag{1.2}$$

The symbol $\mathcal{M}_G$ will be used to indicate the multi-set with elements the terms of the Tutte polynomial of G. See Figure 1.1 for an example.

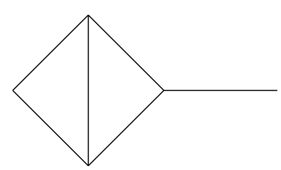


Figure 1.1: The Tutte polynomial for the above graph is $T(G;x,y)=xy^2+2x^2y+xy+x^2+2x^3+x^4$. The multi-set $\mathcal{M}_G=\{xy^2,x^2y,x^2y,xy,x^2,x^3,x^3,x^4\}$.

We focus on the case of finite graphs due to the beautiful bijective correspondences between the terms of the Tutte polynomial, spanning trees, and G-parking functions. Among these is Dhar's burning algorithm [Dha90]; see also [CP05], [CMY10], [Ber08], [CB03]. The burning algorithm is applied to a graph labeled by a function $f: V(G)-\{q\}\to\mathbb{Z}_{\geq 0}$. Start a fire at q, and imagine it burns any edge it reaches. In order to burn through a vertex v, it must first burn through $z=f(v)$ edges which are incident to v. If the fire is able to burn through more than z edges incident to v, it will burn through the vertex. All vertices burn if and only if f is a G-parking function.

$$
\begin{tikzcd}
 & & \{\tau\} \\
\{O_E\} \arrow[urr, "D"] \arrow[r, "R"'] & \{\Sigma\} \arrow[ur, "F"] & \\
 & \{\Pi_G\} \arrow[u, "\Omega"] \arrow[uur, "\Phi"'] &
\end{tikzcd}
$$

Figure 1.2: Commutative diagrams.

We will describe an algorithm which from a G-parking function simultaneously produces a spanning tree T_f of G and a monomial $x^\alpha y^\beta \in \mathcal{M}_T$, through the application of an object called a *tree growing sequence* Σ. This results in two bijections $\tau : \mathcal{P}_{G,q} \to \mathcal{T}_G$, and $\rho : \mathcal{P}_{G,q} \to \mathcal{M}_G$. We prove the main theorem in section 3.1.2.

Theorem: *The maps ρ and τ are bijective.*

The algorithm which achieves this is based on operations which simultaneously split each of the sets $\mathcal{P}_{G,q}$, $\mathcal{T}_G$, and $\mathcal{M}_G$ into two disjoint subsets. We show that these splittings are coherent in that they eventually force $1-1$ correspondences between the sets. As applications of the theorem, sections 3.2.1, 3.2.2, and 3.2.3 describe how Dhar's algorithm with a total edge order O_E [CB03], *proper sets of tree orders* $\{\Pi_G\}$ [CP05], and *process orders* [CMY10], respectively, can be fit into our definition. Let $\{O_E\}$ be the set of edge orders on G. We will define the maps in the diagram below and prove that it commutes. An auxilliary result is the association, via ρ, of a monomial to a G-parking function for the family of bijections in [CP05]. This is evidence that the TGS algorithm can give a non-arbitrary bijection from $\mathcal{P}_{G,q}$ to $\mathcal{M}_G$ for algorithmic bijections between $\mathcal{P}_{G,q}$ and $\mathcal{T}_G$.

The Tutte polynomial is defined more generally for a matroid; see [BO92] for a thorough survey. In section 4.1.1, we compute a polynomial for a cubical zonotopal tiling using similar splitting operations to the TGS algorithm, and show that it is the Tutte polynomial of a specific matroid. In particular, if the vector configuration associated to the tiling is a cographical matroid, then the polynomial is the Tutte polynomial of the underlying finite graph, and we obtain bijections between tiles, $\mathcal{M}_G$, and $\mathcal{T}_G$. We conclude in section 4.1.1 with a discussion that relates zonotopal tilings of cographical matroids to the bijective maps ρ and τ.

Chapter 2

Three Objects Associated to Multigraphs

2.1 Basic Graph Theory Definitions

Although some of the results of this thesis can be applied to disconnected graphs and more generally to matroids (2.5), the focus is on connected multigraphs. A graph is defined to be a tuple $G = (V, E)$, where $V = V(G)$ is a finite set of vertices, and $E = E(G)$ is a finite set of edges. The *order* of a graph is the number of vertices. Each edge is expressed as a tuple $e = (u, v)$, where u, v are the vertices that it joins. Vertices that are joined by an edge are called *neighbors*. A *path* is a sequence of vertices (edges) such that no vertex is repeated, unless the path is a *cycle*, in which case the start and end vertex are the same. A graph is said to be *connected* if for any distinct vertices u, v in $V(G)$, there is a path between them. A *cut-set* is a set of edges which when removed from G result in a disconnected graph. The term *multigraph* is used when more than one edge is allowed between two (not necessarily distinct) vertices.

There are several operations one may perform on graphs. The two operations most relevant to this work are deletion and contraction of edges. The operation of deleting an edge is clear from its name: the graph $G - e$ is obtained by removing the edge e from G and leaving the vertices it joined. The contraction graph G/e is obtained by deleting the edge

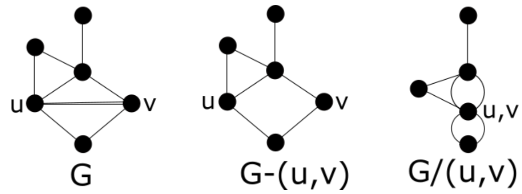


Figure 2.1: The operations of deletion and contraction.

$e = (u, v)$ and identifying the vertices u and v. If any loops or multiple edges are created by this operation, we keep them. Deletion and contraction are central to this work.

2.2 Spanning Trees

A *tree* is a connected graph that contains no cycles (among other characterizations; see theorem below). A *subgraph* is a graph on a subset of the edges $E(G)$ and a subset of the vertices $V(G)$. The subgraph H *spans* G if $V(H) = V(G)$. Spanning trees are a special class of subgraphs of a graph G.

Definition 2.2.1.: Let G be a connected graph of order n. A **spanning tree** of G is a connected, spanning subgraph with $|V(G)| - 1$ edges. If G is disconnected, then a collection of spanning trees on each connected component is called a **spanning forest**.

To obtain a spanning tree of any graph (after removing loops and duplicate edges), successively remove edges from G which are not bridges, until only bridges remain. That is, remove a non-bridge e_1 from $G_1 = G$, one from $G_2 = (V, E - e_1)$, and so on, until the only edges which remain are bridges. The subgraph must be connected, because G was connected, and we never removed a bridge, and it must span G.

There are equivalent definitions for trees, which are stated in the theorem below. (See any introductory textbook on graph theory, such as [Bru12].)

Theorem 2.2.2 : *The following are equivalent for a connected graph $T = (V, E)$ of order n.*

1. *T is a tree.*

2. $|E| = n - 1$.

3. T contains no cycles.

4. Every edge of T is a bridge.

5. There exists a unique (only one) path between every pair of vertices $u, v \in T$.

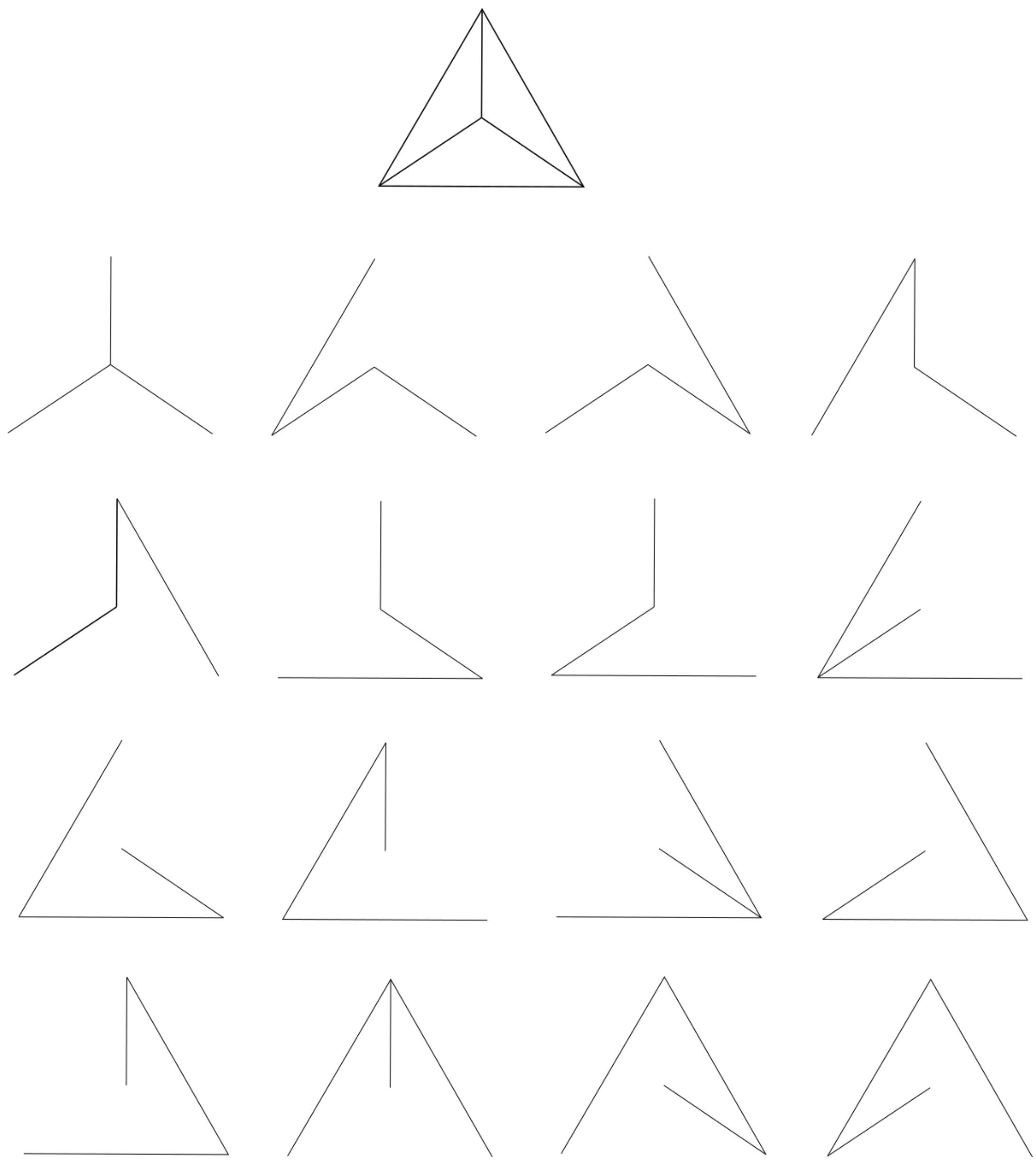


Figure 2.2: Spanning trees of K_4.

2.3 G-Parking Functions

The idea of a G-parking function was introduced by Bak, Tang, and Wiesenfeld [BTW87] and generalized by Dhar [Dha90].

Definition 2.3.1: The **outdegree with respect to** $A \subseteq V$, denoted $outdeg_A(v)$, is the number of neighbors of v not in $A \subseteq V$, with multiplicity.

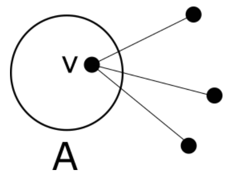


Figure 2.3: A representation of outdegree. We have $outdeg_A(v) = 3$.

Definition 2.3.2: A G**-parking function** is a function $f : V(G) - \{q\} \to \mathbb{Z}_{\geq 0}$ such that any subset $A \subseteq V - \{q\}$ contains a vertex v with $0 \leq f(v) < outdeg_A(v)$.

Example: Let $G = K_4$. Choose a root q, and let the other three vertices be arbitrarily labeled v_1, v_2, v_3. The function on the left of Figure 2.4 is not a G-parking function. If $A = \{v_1, v_2, v_3\}$, then $1 = f(v_i) = outdeg_A(v_i)$ for all i. However, if one of the values of the function is reduced to 0, then we can check that the definition of a G-parking function is satisfied.

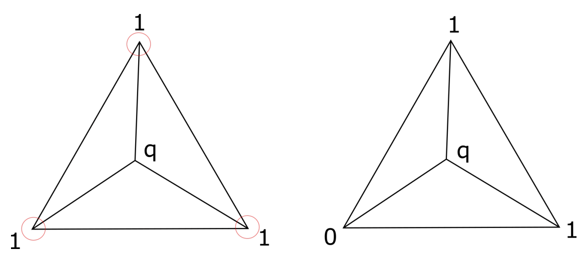


Figure 2.4: Nonexample and example for G-parking functions on K_4.

A classical parking function is a tuple $(a_1, a_2, \ldots, a_n)$ of positive integers such that

$$\#\{k : k \leq i\} \geq i, 1 \leq i \leq n.$$

where $1 \leq a_i \leq n \ \forall i$. If $G = K_n$, then its G-parking functions are the classical parking functions for $n-1$ when 1 is added to every value of f. Since a G-parking function only has non-negative values on $|V(G)|-1$ vertices, the number of G-parking functions for K_n is then $((n-1)+1)^{(n-1)-1} = n^{n-2}$. This number is also known to be the number of spanning trees of K_n. The classical parking functions first appear in the literature in [Pyk59].

Example 2.3.3: Let $G = K_4$. Its G-parking functions with respect to the chosen root q are shown in Figure 2.5. Assigning the labels $\{v_1, v_2, v_3\}$ to the vertices starting with the top vertex and labeling clockwise, if the values of each function are listed as $f(v_1)f(v_2)f(v_3)$ and 1 is added to each value, then we have the set

$$\{(111, 112, 121, 211, 221, 212, 122, 113, 311, 131, 213, 123, 312, 321, 132, 231\}$$

which is precisely the set of classical parking functions for $n = 3$.

2.4 Divisor Theory for Discrete Graphs

Let G be a graph and fix a labeling $\{q, v_1, \ldots, v_n\}$ on the vertices of G, where we have chosen a root q. Let $Div(G) = \mathbb{Z}^{|V(G)|}$ be the group of $\mathbb{Z}$-linear combinations of vertices, written as $f = (a_0, a_1, \ldots, a_n)$, with a_0 the coefficient of q, and a_i the coefficient of v_i for all other i. These are also called configurations or abelian sandpiles. A divisor is commonly denoted

$$D = \sum_{i=1}^{n} a_i v_i.$$

The *degree* of a divisor is $\sum a_i$, and $Div^k(G)$ denotes the set of divisors of degree k.

Variations of the *chip-firing* game can be played on the vertices of a graph. If $v_i \in V$ and $f = (a_0, a_1, \ldots, a_n) \in Div(G)$, then the chip-firing move σ_i is defined by

$$\sigma_i(f)(v_j) = \begin{cases} a_j - deg(v_j) & i = j \\ a_j + n(v, v_j) & i \neq j \end{cases} \tag{2.1}$$

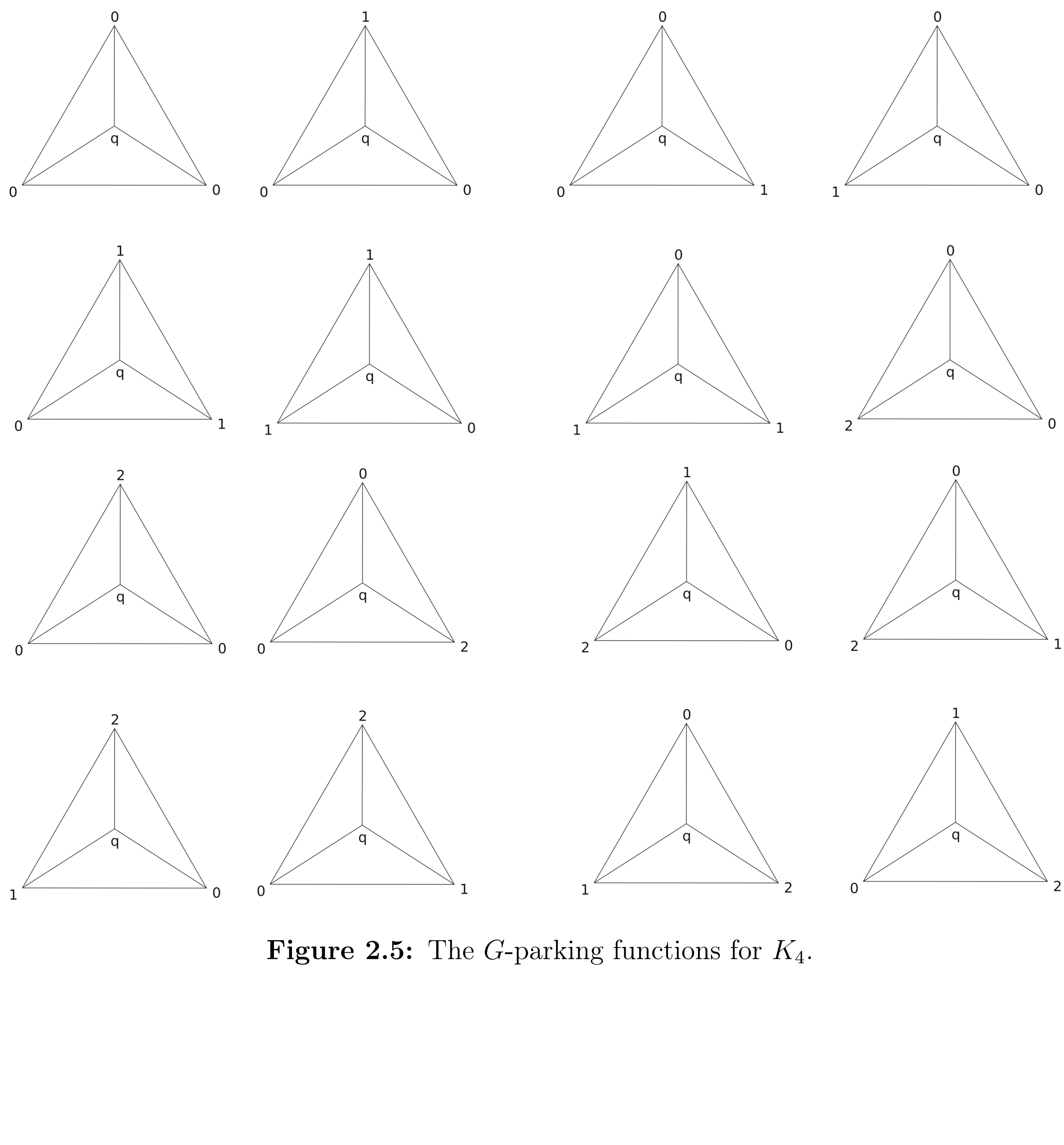
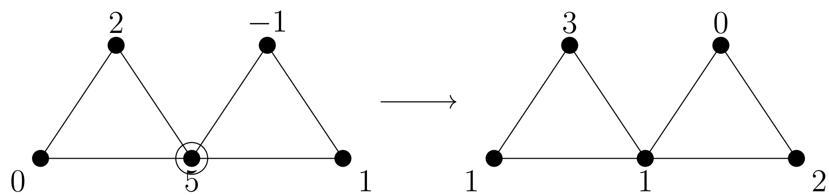


Figure 2.5: The G-parking functions for K_4.

where $n(v, v_i)$ is the number of edges between v_j and v_i. We say that two divisors f and g are **linearly equivalent**, written $f \sim g$, if g can be obtained from f via a sequence of chip-firing moves. A *principal divisor* is one linearly equivalent to 0. Linear equivalence of divisors D and D' also means that $D - D'$ is principal. Every chip-firing move is a sum of σ_i's, so that one can view linear equivalence as being generated by the cuts b_v, where b_v is the set of edges incident to v.

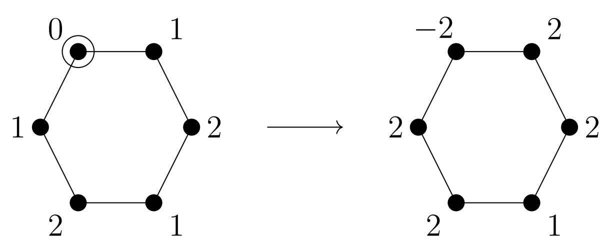
Example 1

The configuration can have a negative value at a vertex.



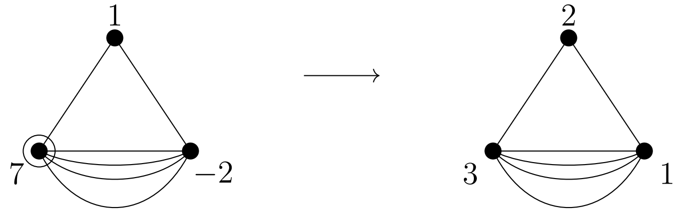
Example 2

Firing a vertex may cause that vertex to have a negative value.



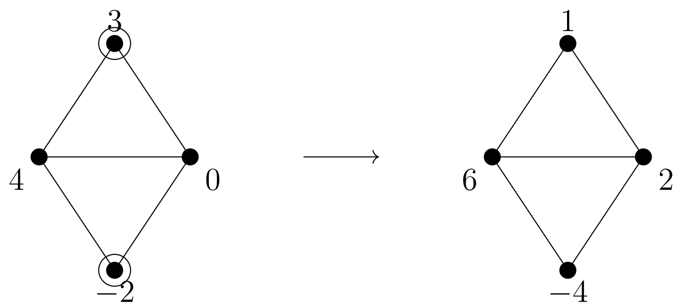
Example 3

If there are multiple edges between two vertices, and one of them is fired, then it gives one chip per edge.



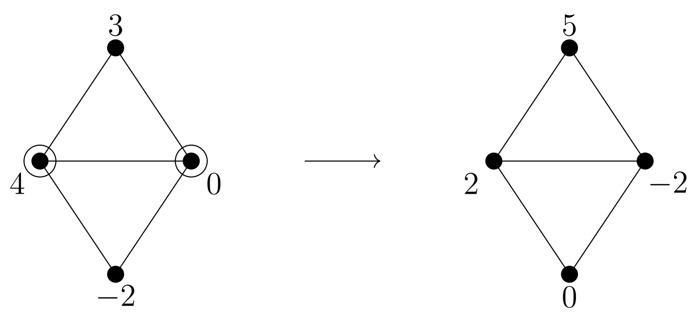
Example 4

We can fire two or more vertices simultaneously, an alternative to firing one by one. Below, the top and bottom vertices are both fired once.



Example 5

If adjacent vertices u, v are fired simultaneously, they will exchange a chip for every edge between them. In other words, the net change of the chips between these vertices is 0, and the total change in chips at each of these vertices only depends on edges not joining u, v. If the vertices with 4 and 0 chips on the left are fired simultaneously, we have the resulting divisor below.



Principal divisors can also be described via the Laplacian matrix L of G. With our fixed labeling $(q, v_1, \ldots, v_n)$ of the vertices of G, consider a vector $x = (x_0, x_1, \ldots, x_n)$ with integer entries. Multiplying Lx is equated to starting from the divisor 0 and firing each vertex v_j, x_j times. This gives a divisor D, where $D(v)$ for each vertex is $(Lx)(v)$. The image of L is the group of principal divisors. The set of all effective divisors ($a_i \geq 0$ for all i) linearly equivalent to a given divisor D is called a *complete linear series*, and is denoted

$|D|$. The *Picard group* of G is defined as $Pic(G) = Div(G)/(D \sim 0)$, and the Jacobian $Jac(G) = Pic^0(G) = Div^0(G)/(D \sim 0)$. The degree map $deg(D) = \sum a_i$ is a surjective homomorphism from $Pic(G)$ to $\mathbb{Z}$, and we get a short exact sequence of abelian groups:

$$0 \longrightarrow Jac(G) \longrightarrow Pic(G) \longrightarrow \mathbb{Z} \longrightarrow 0$$

Since $\mathbb{Z}$ is free (hence, projective), the sequence is split, and we get that $Pic(G) = \mathbb{Z} \bigoplus Jac(G)$. The Jacobian, in a sense, measures the failure of degree 0 divisors to be principal. In the case of a discrete graph, the Jacobian is a finite group, and is isomorphic to the critical group $K(G)$ of critical configurations (see [Big99] for definitions and the proof).

Definition 2.4.1: A **q-reduced divisor** f is a G-parking function when restricted to $V(G) - \{q\}$, and in addition has $f(q) = -\sum_{v \in V(G)-\{q\}} f(v)$. In particular, we can view $\mathcal{P}_{G,q} \subset Div^0(G)$.

Thus, one can state the definition of a G-parking function in the language of chip firing: it is a configuration such that firing any subset of vertices leaves at least one vertex in "debt".

Theorem 2.4.2, [BN06], [MZ08]: *There exists a unique q-reduced representative in every linear equivalence class of $Div^0(G)$.*

Corollary 2.4.3: *Elements of $Pic^0(G)$ ($Jac(G)$) are in bijection with elements of $\mathcal{P}_{G,q}$.*

There is a similar story for metric graphs outlined in Appendix 1 (5).

2.5 Matroids

The theory of matroids is heavily influenced by graph theory and linear algebra. In fact, graphs and vector configurations are common examples of oriented matroids. We will return to these cases in Chapter 3 along with the zonotope associated to a vector configuration. There are several equivalent definitions of a matroid that we include for the interested reader.

Definition 2.5.1: Let E be a set. The **independent sets** of E is a collection $\mathcal{I}$ of subsets

$I \subseteq E$ which satisfies:

1. $\mathcal{I}$ is not empty.

2. If $I \in \mathcal{I}$ and $J \subset I$, then $J \in \mathcal{I}$.

3. If I and J are in $\mathcal{I}$ and $|I| = |J| + 1$, then there exists and element $x \in I - J$ such that $J \cup \{x\} \in \mathcal{I}$. (exchange axiom)

Definition 2.5.2: A **matroid** is a pair $M = (E, \mathcal{I}))$, where E is a ground set and $\mathcal{I}$ is a collection of independent sets.

Example 2.5.3: Let $E = \{a, b, c\}$. Then there is a matroid $M = (E, \mathcal{I})$ with $I = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$. Note that there are several possibilities for defining the collection $\mathcal{I}$ that are consistent with the above properties; more precisely, one will find eight nonisomorphic matroids on the ground set E, where two matroids M_1 and M_2 are isomorphic if there is a bijection $M_1 \to M_2$ betwen the ground sets E_1 and E_2 which preserves independent sets.

Definition 2.5.4: The **rank function** of a matroid $M = (E, \mathcal{I})$ is

$$r : 2^E \to \mathbb{Z}_{>0}$$

$$r(A) = \max_{I \subseteq A, I \in \mathcal{I}} |I|$$

Definition 2.5.5.: The **dual matroid** M^* is the pair $(E, \mathcal{I}^*)$, where a set $J \in \mathcal{I}^*$ is independent in M^* if and only if $E - J$ contains a basis of M. The rank function is $r^*(A) = |A| - r(E) + r(E - A)$ is the dual rank function.

The alternative definitions of a matroid will be useful when we discuss graphic matroids and vector configurations (section 4.1.2). First, let M be a matroid, and let $\mathcal{C}$ be a collection of minimally dependent sets called *circuits*. By a minimally dependent set C we mean that for any $c \in C$, $C - \{d\} \in \mathcal{I}$. The collection of circuits is characterized by the following:

1. $\mathcal{C}$ does not contain $\emptyset$.

2. For any $C, D \in \mathcal{C}$, neither is a proper subset of the other.

3. If $C_1, C_2 \in \mathcal{C}$ are distinct, then for any $x \in C_1 \cap C_2$ we have that $(C_1 \cup C_2) - \{e\}$ contains a member of $\mathcal{C}$.

Theorem 2.5.6: *Let E be a finite set and let $\mathcal{C}$ be its collection of circuits. Let $\mathcal{I}$ be the collection of subsets of E that do not contain a member of $\mathcal{C}$. Then $M = (E, \mathcal{I})$ is a matroid. On the other hand, given a matroid M, its collection of circuits satisfies the above three properties.*

Returning to the example of the three element matroid with ground set $E = \{a, b, c\}$ and independent sets $\mathcal{I} = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$, its collection of circuits is $\mathcal{C} = \{\{c\}\}$.

There is a third characterization of a matroid which is natural to think of - that by its bases. Let $\mathcal{B}$ be the collection of subsets of E which are maximally independent; that is, if $B \in \mathcal{B}$, then $B \in \mathcal{I}$ and for any x not in B, $B \cup x$ is dependent.

Theorem 2.5.7: *The collection $\mathcal{B}$ is the set of bases of a matroid M if and only if it satisfies*

1. *$\mathcal{B}$ is non-empty.*

2. *Let $X, Y \in \mathcal{B}$. For any $x \in X$ there is an element $y \in Y$ such that $(X - \{x\}) \cup \{y\} \in \mathcal{B}$.*

There is only one basis in the collection of bases for our running example: $\mathcal{B} = \{\{a, b\}\}$.

Example 2.5.8: We can construct a matroid $\mathcal{M}(G)$ where the ground set contains the edges of a graph G of order n, possibly disconnected, and the independent sets are the sets of edges which form forests in G. Thus, the edges of spanning forests are maximally independent sets. They also form the collection of bases $\mathcal{B}$ of $\mathcal{M}(G)$. We prove that this collection satisfies the two properties in the previous theorem in the case for G connected. The proof can then be extended to each connected component of a disconnected graph. To prove the first property, we've noted in section 2.2 that every graph has a spanning tree; thus, $\mathcal{B}$ is nonempty.

If T_1, T_2 are spanning trees of G, let $e = (a, b) \in T_1$. Removing e from T_1 results in two components: let one component contain the vertex subset V' and the other component contain the vertex subset V''. Since T_2 is a spanning tree, there must be an edge $f \in T_2$ that connects a vertex in V' to one in V'', else T_2 would be disconnected. Adding f to $T_1 - \{e\}$ will result in a connected graph. Then $(T_1 - \{e\}) \cup \{f\}$ is a spanning tree: it is a connected graph with n vertices and $n - 1$ edges.

We will return to the above example in 4.1.1.

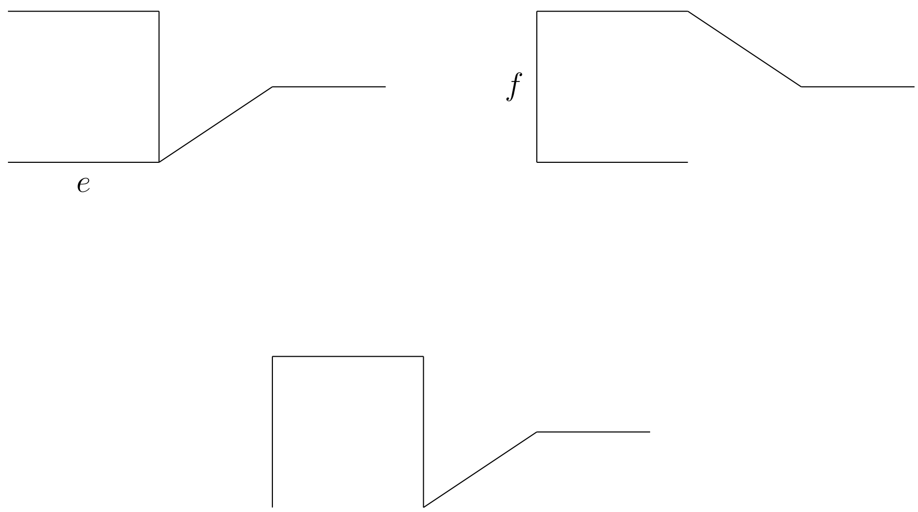


Figure 2.6: Suppose the the two trees in the first row are spanning trees of the same graph G. The tree on the bottom is the "hybrid" of the two trees on top by removing the edge e from the first and adding in f.

2.6 The Tutte Polynomial

In this section we take a closer look at the Tutte polynomial. A graph isomorphism invariant is a function F such that if $G_1 \cong G_2$, then $F(G_1) = F(G_2)$. A *Tutte-Grothendieck isomorphism invariant* F is a function where, for every $e \in E(G)$, the following two properties are satisfied:

$$F(G) = F(G - e) + F(G/e) \text{ if e is neither a loop nor a bridge} \tag{2.2}$$

$$F(G) = F(G/e)F(G - e) \text{ otherwise} \tag{2.3}$$

These properties are defined more generally for a matroid in [BO92].

Definition 2.6.1: The **Tutte polynomial** $T(G;x,y)$ of G is the universal Tutte-Grothendieck graph isomorphism invariant satisfying the following deletion/contraction principal, and defining $T(\bullet;x,y)=1$, $\bullet$ the graph with one vertex.

$$T(G;x,y)=\begin{cases} yT(G-e;x,y) & e \text{ a loop} \\ xT(G/e;x,y) & e \text{ a bridge} \\ T(G-e;x,y)+T(G/e;x,y) & \text{otherwise} \end{cases} \tag{2.4}$$

Figure 2.7 displays the use of the deletion-contraction recursion to compute the Tutte polynomial of the given graph. The edge chosen in each step is in bold. When the edge is ordinary - neither a loop nor a bridge - an arrow left indicates contraction of the edge, and an arrow right indicates deletion of the edge. An arrow down indicates either deletion of a loop (with multiplication by y beside it) or contraction of a bridge edge (with multiplication by x beside it).

The universality of the Tutte polynomial is essentially stated in the following theorem.

Theorem 2.6.2 [BO92]: *Let $\mathcal{G}$ be the isomorphism classes of graphs. There is a unique function T from $\mathcal{G}$ into the polynomial ring $\mathbb{Z}[x,y]$ having the following properties:*

1. *If b is a bridge of G, $T(b;x,y)=x$. If l is a loop of G, $T(l;x,y)=y$.*

2. *If $e\in E(G)$ is neither a loop nor a bridge, then*

$$T(G;x,y)=T(G-e;x,y)+T(G/e;x,y).$$

3. *If e is loop or bridge, then*

$$T(G;x,y)=T(e;x,y)T(M-e;x,y)$$

(Universality) *Moreover, if R is a commutative ring and F is a function from $\mathcal{G} \to R$ (injective) satisfying the recursions* (2.2) *and* (2.3) *when* $|E(G)| \geq 2$, *then*

$$F(G) = T(G; F(b), F(l)).$$

An example is the generating function for the critical configurations of a graph; in [Mer05], it is proven that this function is the evaluation of $T(G; x, y)$ along the line $x = 1$. A corollary to the above theorem which characterizes generalized $T-G$ invariants is stated in Appendix B (6).

An equivalent definition for the Tutte polynomial is as a closed formula over all spanning subgraphs of G. Let $c(A)$ be the number of connected components of a spanning subgraph A. Then

$$T(G; x, y) = \sum_{A \subseteq G} (x-1)^{c(A)-c(G)} (y-1)^{|E(A)|+c(A)-|V|} \tag{2.5}$$

Lemma 2.6.3: *The above formula satisfies the deletion-contraction recursion.*

Proof. Case 1: Let $e \in E(G)$ be a loop. Rewrite the Tutte polynomial as

$$T(G; x, y) = \sum_{A \subseteq G e \notin A} (x-1)^{c(A)-c(G)} (y-1)^{|E(A)|+c(A)-|V|} + \sum_{A \subseteq G e \in A} (x-1)^{c(A)-c(G)} (y-1)^{|E(A)|+c(A)-|V|}$$

$$= \sum_{A \subseteq G e \notin A} (x-1)^{c(A)-c(G)} (y-1)^{|E(A)|+c(A)-|V|} + (y-1) \sum_{A \subseteq G e \in A} (x-1)^{c(A)-c(G)} (y-1)^{|E(A)|-1+c(A)-|V|}$$

$$= \sum_{A \subseteq G e \notin A} (x-1)^{c(A)-c(G)} (y-1)^{|E(A)|+c(A)-|V|} + y \sum_{A \subseteq G e \in A} (x-1)^{c(A)-c(G)} (y-1)^{|E(A)|-1+c(A)-|V|}$$

$$- \sum_{A \subseteq G e \in A} (x-1)^{c(A)-c(G)} (y-1)^{|E(A)|-1+c(A)-|V|}$$

Disregarding coefficients, all three sums are the same. The second and third sums are equivalent to the sum over all spanning subgraphs without e since the additional -1 in the exponent of $(y-1)$ can be interpreted as deleting e from the subgraphs which contain it. Since any subgraph without e can be obtained by removing e from another, then this sum is $T(G-e;x,y)$. Then we have:

$$T(G;x,y) = T(G-e;x,y) + yT(G-e;x,y) - T(G-e;x,y)$$

$$= yT(G-e;x,y)$$

Case 2: Let $e \in E(G)$ be a bridge. Similarly to case 1, we first write:

$$T(G;x,y) = \sum_{A\subseteq G,\, e\notin A} (x-1)^{c(A)-c(G)}(y-1)^{|E(A)|+c(A)-|V|} + \sum_{A\subseteq G,\, e\in A} (x-1)^{c(A)-c(G)}(y-1)^{|E(A)|+c(A)-|V|}$$

$$= (x-1) \sum_{A\subseteq G,\, e\notin A} (x-1)^{c(A)-1-c(G)}(y-1)^{|E(A)|+c(A)-|V|} + \sum_{A\subseteq G,\, e\in A} (x-1)^{c(A)-c(G)}(y-1)^{|E(A)|+c(A)-|V|}$$

$$= x \sum_{A\subseteq G e\notin A} (x-1)^{c(A)-1-c(G)}(y-1)^{|E(A)|+c(A)-|V|} - 1 \sum_{A\subseteq G\ e\notin A} (x-1)^{c(A)-1-c(G)}(y-1)^{|E(A)|+1+c(A)-|V|}$$

$$+ \sum_{A\subseteq G,\, e\in A} (x-1)^{c(A)-c(G)}(y-1)^{|E(A)|+c(A)-|V|}$$

All three sums are $T(G/e;x,y)$. In the second line, subtracting 1 from the exponent of $x-1$ is equivalent to identifying the vertices of e: since e is a bridge, spanning subgraphs which do not contain it must be disconnected (there is no path between the vertices it joins); thus, when these vertices are identified, the number of connected components of A must decrease by one. Then we have:

$$T(G;x,y) = T(G/e;x,y) + xT(G/e;x,y) - T(G/e;x,y)$$

$$= xT(G/e;x,y)$$

Case 3: Let $e \in E(G)$ be an ordinary edge (neither a loop nor a bridge). Then one can write:

$$T(G;x,y) = \sum_{A\subseteq G,\, e\notin A} (x-1)^{c(A)-c(G)}(y-1)^{|E(A)|+c(A)-|V|} + \sum_{A\subseteq G,\, e\in A} (x-1)^{c(A)-c(G)}(y-1)^{|E(A)|+c(A)-|V|}$$

$$= T(G-e;x,y) + T(G/e;x,y)$$

since the spanning subgraphs of G not containing e are in correspondence with spanning subgraphs of $G-e$, and spanning subgraphs of G containing e are in correspondence with spanning subgraphs of G/e. $\square$

The original definition of the Tutte polynomial is as a sum over the spanning trees of G [Tut54], with exponents quantifying the *internal* and *external* activities of a tree T. These activities are defined in Section 3.2.1. It is easy to see with this original definition that $T(G;1,1)$ is the number of spanning trees of G.

A few other interesting evaluations of the Tutte polynomial:

- $T(G;2,0)$ is the number of acyclic orientations of G with no prescribed source.
- $T(G;1,2)$ is the number of connected subgraphs.
- $T(G;2,1)$ is the number of forests of G.

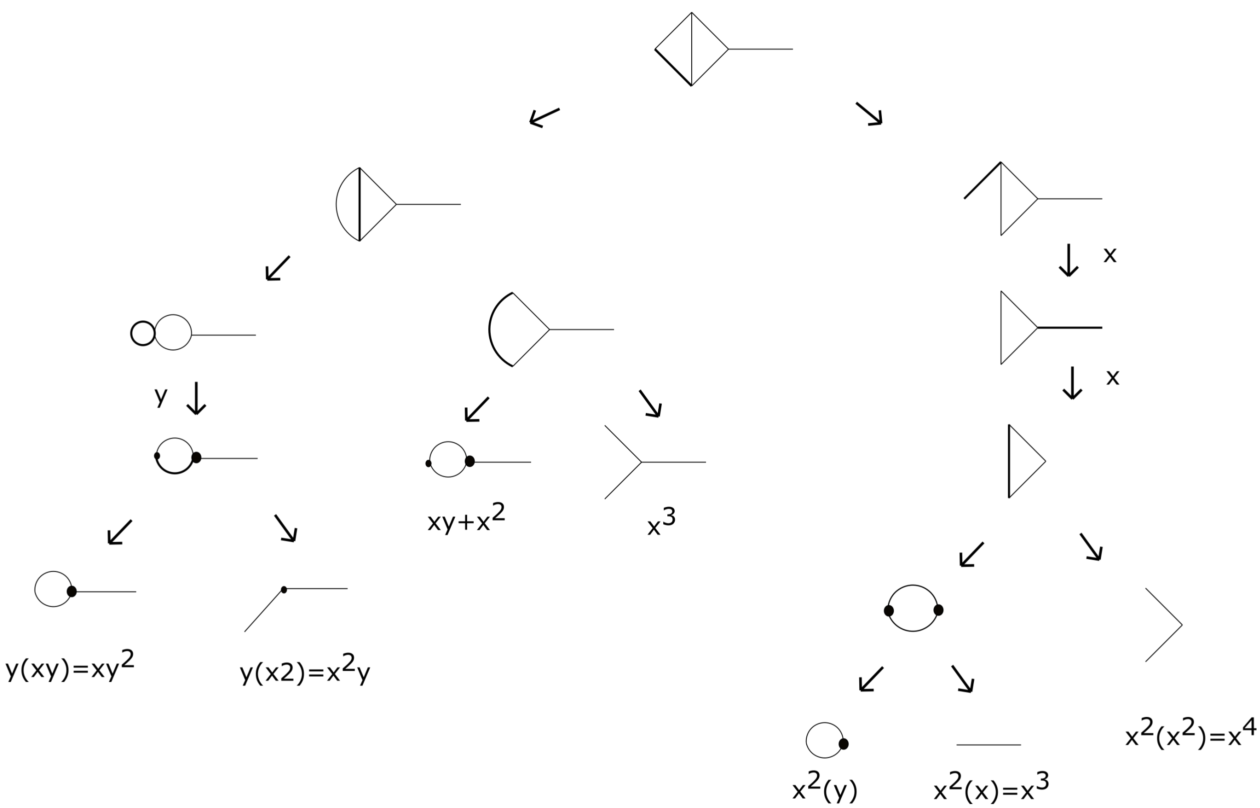


Figure 2.7: Using the deletion-contraction recursion formula to compute the Tutte polynomial for the example in Chapter 1. We have that $T(G; x, y) = xy^2 + 2x^2y + xy + x^2 + 2x^3 + x^4$.

Chapter 3

Combinatorial Bijections of Interest

3.1 The Tree Growing Sequence

3.1.1 Definition and the Main Algorithm

We define the central object of this paper, the *tree growing sequence*. Let $G = (V, E)$ be a connected graph and $\mathcal{S}$ be the set of all connected subgraphs of G containing q as a vertex. For each $S \in \mathcal{S}$, denote by $\mathcal{H}_S$ the set of proper subgraphs of S such that $\mathcal{H}_S \subset \mathcal{S}$.

Definition 3.1.1: A **tree growing sequence (TGS)** is a collection of tuples

$$\Sigma = \{(S, \sigma_S)\}_{S\in\mathcal{S}}$$

where σ_S is a function from $\mathcal{H}_S$ to the edge set $E(S)$ of S such that $\sigma_S(T) \notin E(T)$ and $\sigma_S(T) \cup T$ is connected.

The name "tree growing sequence" is used because the graph T will always be a tree in the application of our algorithm (although it need not be in the above definition). Given a tree growing sequence Σ and a function $f : V(G) - \{q\} \to \mathbb{Z}$, we apply the following algorithm to the tuple $(f, S, U, X, \alpha, \beta)$, where $U \subseteq V(G)$, $X \subseteq E(G)$, and $\alpha, \beta \in \mathbb{Z}_{\geq 0}$. The result will be a tree $T_f = (U, X)$ and a monomial $x^\alpha y^\beta$. Beginning with $S = E(G)$, $U = \{q\}$,

$X = \emptyset$, $\alpha = 0$, and $\beta = 0$, the edge $e = \sigma_G(\{q\}) = (v, q)$ is added to X and v added to U if $f(v) = 0$ and $u \neq q$ (e not a loop). Furthermore, when e is a bridge of G, then α is increased by one. If $v = q$, so that e is a loop, delete it and increase β by one. If $f(v) \geq 1$, the value of $f(v)$ is reduced by one. The edge e is not added to X, and we equate this with edge deletion by replacing $E(G)$ with $E(G) - e$. If it is the case that $f(v) < 0$, we terminate the algorithm.

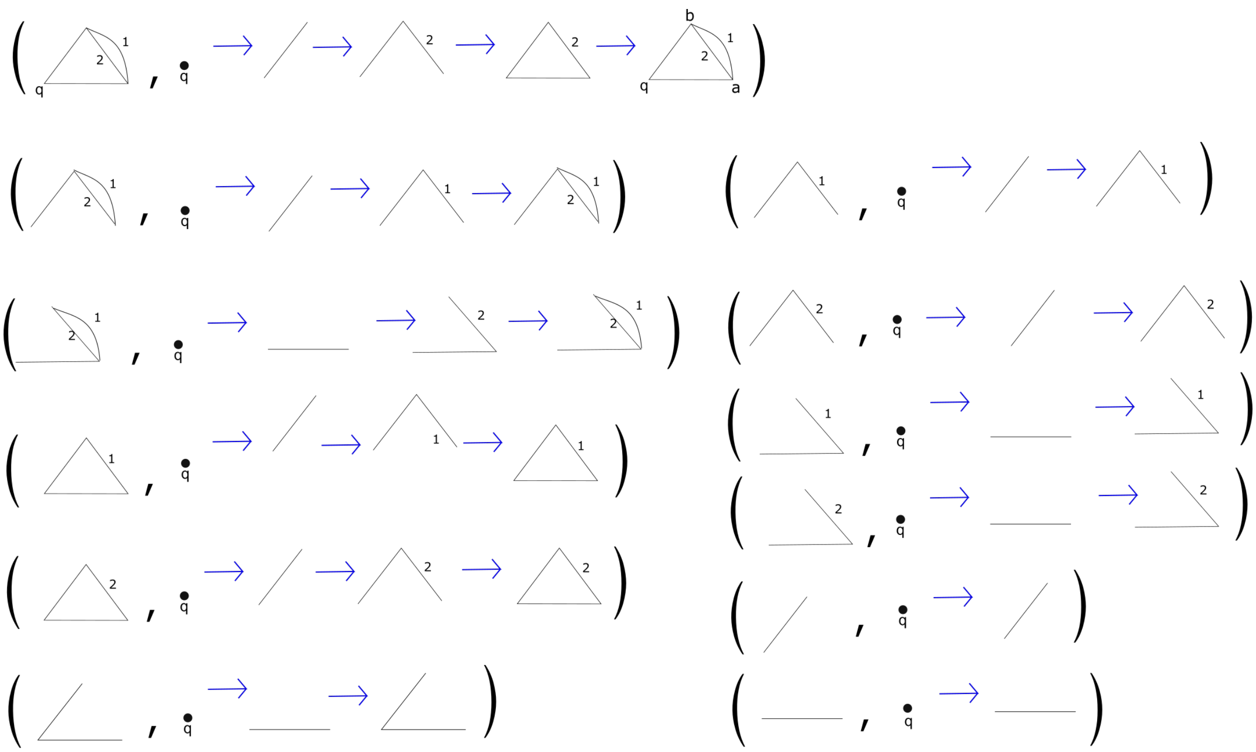


Figure 3.1: An example of the data for a TGS.

In subsequent steps, we consider the tuple $(f, S, U, X, \alpha, \beta)$, where the value of f at some vertices may have been reduced in previous steps. For each image $\sigma_S(T) = e$, we assume that $t(e)$ is a vertex of T. If $h(e)$ is also a vertex of T, then e will be called a loop. The set

$S = E(G) - \{e\}_D$, where the edges $\{e\}_D$ have been deleted. The algorithm is shown below.

Algorithm 3.1.1: TREE GROWING SEQUENCE ALGORITHM $(f, S, U, X, \alpha, \beta)$

Input: A graph $G = (V, E)$ with root vertex q, tree growing sequence Σ, and an integer valued function f on the vertices.

Output: A tree T_f and monomial $x^\alpha y^\beta$.

Initialization:

$S = G, U = \{q\}, X = \emptyset$

$\alpha = 0, \beta = 0, T = (\{q\}, \emptyset)$

while $\sigma_S(T)$ is defined

do

$$
\begin{cases}
\textbf{if } f(\sigma_S(T)) < 0 \\
\quad \textbf{then } \textit{terminate} \\
\quad \textbf{else if } f(\sigma_S(T)) = 0 \\
\quad \textbf{then } \begin{cases}
\textbf{if } e = \sigma_S(T) \text{ a bridge of } S \\
\quad \textbf{then } \begin{cases} \alpha \leftarrow \alpha + 1 \\ X \leftarrow X \cup e \\ U \leftarrow h(e) \\ S \leftarrow S \cup e \\ T = (U, X) \end{cases} \\
\quad \textbf{else } \begin{cases} X \leftarrow X \cup e \\ U \leftarrow h(e) \\ S \leftarrow S \cup e \\ T = (U, X) \end{cases}
\end{cases} \\
\quad \textbf{else if } T \cup e \text{ not a tree} \\
\quad \textbf{then } \begin{cases} \beta \leftarrow \beta + 1 \\ S \leftarrow S - e \end{cases} \\
\quad \textbf{else } \begin{cases} f(h(e)) \leftarrow f(h(e)) - 1 \\ S \leftarrow S - e \end{cases}
\end{cases}
$$

output $(T_f = (U, X),\ x^\alpha y^\beta)$

We illustrate in Figure 3.2 the possibilities for updating the tuple when applying the algorithm.

Proposition 3.1.2: *For any tree growing sequence Σ, applying the TGS algorithm to a function $f : V(G) - \{q\} \to \mathbb{Z}$ will terminate on a spanning tree of G if and only if $f \in \mathcal{P}_{G,q}$.*

Proof. Fix a root vertex q and $f \in \mathcal{P}_{G,q}$. If the algorithm terminates at non-spanning T_f, then T_f spans S but not G. This implies that $V(S) \neq V(G)$ and we have deleted all edges between $V(S)$ and $U = V(G) - V(S)$. Then we can find some $A \subseteq U$ such that $outdeg_A(v) \leq f(v)$ for

$$(f, S, U, X, \alpha, \beta) \xrightarrow{\quad a \quad} (f, S, U \cup e_h, X \cup e, \alpha, \beta)$$

$$(f, S, U, X, \alpha, \beta) \xrightarrow{\quad b \quad} (f, S, U \cup e_h, X \cup e, \alpha + 1, \beta)$$

$$(f, S, U, X, \alpha, \beta) \xrightarrow{\quad c \quad} (f, S - e, U, X, \alpha, \beta + 1)$$

$$(f, S, U, X, \alpha, \beta) \xrightarrow{\quad d \quad} (f, S - e, U, X, \alpha, \beta)$$

Figure 3.2: Possibilities for updating the tuple.

all $v \in A$. However, this is impossible since $f \in \mathcal{P}_{G,q}$. Hence, $V(S) = V(G)$, and T spans G.

Conversely, if $h \notin \mathcal{P}_{G,q}$, then a tree growing sequence will not terminate on a spanning tree of G. Let $A \subseteq V - \{q\}$ be a subset such that all vertices $v \in A$ satisfy $outdeg_A(v) \leq h(v)$. It will suffice to let A consist of a single vertex v, because any such subset A can be thought of as a single vertex with $deg(A) = \sum_{v \in A} outdeg_A(v)$. This translates to $0 < deg(v) \leq h(v)$ (excluding loops). Consider the first time that $\sigma_S(T) = (v, u)$, $u \in V(T)$. This will eventually occur, because $\sigma_S(T)$ is defined as long as $T \neq S$. The edge (v, u) will be deleted because it was assumed that $deg(v) > 0$. Moreover, we reduce $h(v)$ by one. Every time $\sigma_{S'}(T') = (v, u')$, the edge will be deleted, and $h(v)$ reduced by one. Since $deg(v) \leq h(v)$, we will eventually exhaust all edges from A to T. Hence, we will not get a spanning tree by applying Σ to h (Figure 3.3). □

We define the map $\tau : \mathcal{P}_{G,q} \to \mathcal{T}_G$ to be the assignments $f \mapsto T_f$ according to Σ.

Proposition 3.1.3: *If $f \in \mathcal{P}_{G,q}$, then the above algorithm always produces a monomial $x^\alpha y^\beta$ in the multiset $\mathcal{M}_G$ when applied to f.*

Proof. Start with $\alpha = \beta = 0$. If $e = \sigma_S(T)$ is a bridge of S, then increase α by one. If e is a loop, meaning $f(h(e)) = 0$ and $T \cup e$ has a cycle, then delete it, and increase β by one. In light of equation (1), we are simply isolating a monomial of $T(G; x, y)$ when computing it via recursion, and this is the monomial which we associate to f. □

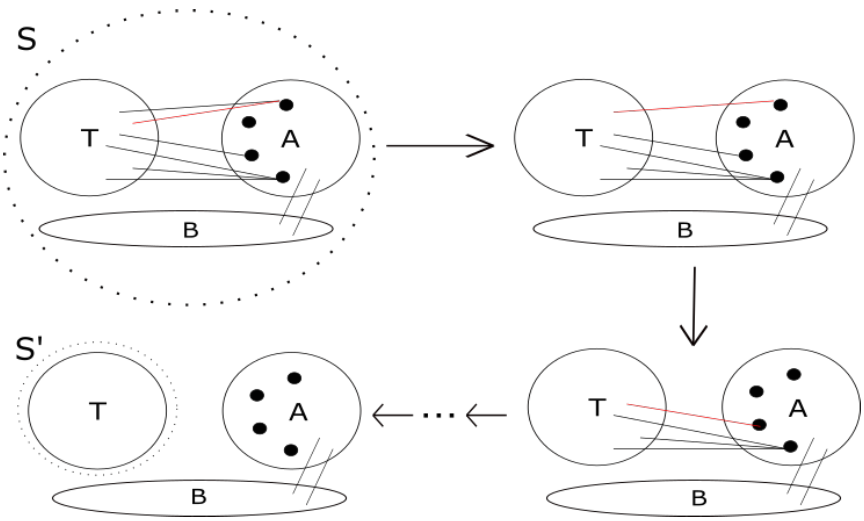


Figure 3.3: The vertex set $B = V(G) - V(T) - A$. The picture shows what happens when $\sigma_S(T) = e$ for an edge joining a vertex in A to one in $V(T)$.

The above proposition is nothing new. However, it is the starting point for a closed formula for the Tutte polynomial as a sum over $\mathcal{P}_{G,q}$ - done by Chang, Ma, and Yeh in [CMY10] - and serves as inspiration to generalize known algorithmic bijections. Also note that the set $X \subset E(S)$ can be viewed as contracted edges, though technically we do not alter the structure of the subgraph when adding an edge to X.

3.1.2 The Splitting of $\mathcal{P}_{G,q}$

We change our philosophy from the previous section: instead of taking a single G-parking function f and producing a spanning tree and monomial, we begin with the set of parking functions $P_{G,q}$ and perform splitting operations with respect to the deletion/contraction principle. That is, split the parking functions according to whether the edge $e = (h(e), t(e))$ is added to X or deleted; see Figure 3.4 for a visual. This splitting will also result in the bijections $\tau : \mathcal{P}_{G,q} \to \mathcal{T}_G$ and $\rho : \mathcal{P}_{G,q} \to \mathcal{M}_G$. To this end, we include the proofs of two lemmas. We use the convention that when an edge e is contracted and $h(e), t(e)$ are identified, the "thickened" vertex is called $t(e)$. We begin with letting $e = (h(e), q) = \sigma_G(\{q\})$ for an arbitrary TGS $\Sigma = \{(S, \sigma_S)\}$.

Lemma 3.1.4: *If l is a loop, then $\mathcal{P}_{G,q} = \mathcal{P}_{G-l,q}$.*

Proof. Let v be a vertex incident to a loop l. Then $outdegree_{A,G}(v) = outdegree_{A,G-l}(v)$ for any vertex subset containing v. $\square$

Lemma 3.1.5: *If e is a bridge, then $P_{G,q}$ is in one-to-one correspondence with $P_{G/e,q}$.*

Lemma 3.1.6 [CMY10]: *There is a bijection ϕ between the set of G-parking functions f with $f(h(e)) = 0$ and the set of (G/e)-parking functions.*

Proof. Define the map $\phi : \mathcal{P}_{G,q} \longrightarrow \mathcal{P}_{G/e,q}$ by $\phi(f)(w) = f(w)$ for any $w \in V(G) - \{h(e), q\}$. Then for a G-parking function $f = (f(v_1), ..., f(h(e)) = 0, ..., f(v_n))$, $\phi(f) = (f(v_1), ..., \widehat{f(h(e))}, ..., f(v_n))$. We claim that

(i) $\phi(f)$ is a (G/e)-parking function, and

(ii) $\phi|_{f(h(e))=0}$ is a bijection.

To prove (i), we need to check that for all subsets A in $V(G/e) - \{q\}$, there is some v with $outdeg_{A,G/e}(v) > \phi(f)(v)$. This is clear, as e_h is absorbed by q, so that for any subset $A \subseteq V(G/e) - \{q\} = V(G) - \{h(e), q\}$, we immediately have that $outdeg_{A,G/e}(v) = outdeg_{A,G}(v)$. For (ii), consider $g \in \mathcal{P}_{G/e}$. Let $f = \phi^{-1}(g) = (g(v_1), ..., g(v_{k-1}), 0, g(v_{k+1}), ..., g(v_n))$. Then f is a G-parking function with $f(h(e)) = 0$. If $h(e) \in A \subseteq V(G) - \{q\}$, then $0 = f(e_h) < outdeg_{A,G}(u)$, as $outdeg_{A,G}(u) \geq 1$ ($h(e)$ is a neighbor of q). If $e_h \notin A$, then $outdeg_{A,G/e}(v) = outdeg_{A,G}(v)$ for all $v \in A$, so $0 \leq f(v) = g(v) < outdeg_{A,G}(v)$ for some v in every $A \subseteq V(G) - \{q\}$, since g is a (G/e)-parking function. $\square$

Lemma 3.1.7 [CMY10]: *There is a bijection ψ between the set of G-parking functions f with $f(e_h) \geq 1$ and the set of $(G-e)$-parking functions.*

Proof. Define the map $\psi : \mathcal{P}_{G,q} \longrightarrow \mathcal{P}_{G-e,q}$ by $\psi(f) = (f(v_1), ..., f(h(e)) - 1, ..., f(v_n))$. Then

(i) $\psi(f)$ is a $(G-e)$-parking function.

(ii) $\psi|_{f(h(e))\geq 1}$ is a bijection.

For (i), we need to check that there is some v such that $\psi(f)(v) < outdeg_{A,G-e}(v)$, for all subsets A in $V(G-e)-\{q\}$. It is obvious that $outdeg_{A,G-e}(v) = outdeg_{A,G}(v)$ if $v \neq e_0$. If $v = e_0$, then $outdeg_{A,G-e}(h(e)) = outdeg_{A,G}(h(e)) - 1$ and $\psi(f)(h(e)) = f(h(e)) - 1$. Then it is immediate that for any $A \subseteq V(G-e)-\{q\}$, we can find some $v \in A$ satisfying the condition. Now, consider $g \in \mathcal{P}_{G-e}$. Let $f = \psi^{-1}(g) = (g(v_1), ..., g(v_{i-1}), g(h(e)) + 1, g(v_{i+1}), ..., g(v_n))$. Then f is clearly a G-parking function with $f(h(e)) \geq 1$ (we only need to consider subsets $A \ni h(e)$, and both $f(h(e))$ and $outdegree_A(h(e))$ increase by 1), giving that ψ is a bijection. $\square$

Corollary 3.1.8: *For any graph S with fixed root q, there is a bijection between $\mathcal{P}_{S,q}$ and $\mathcal{P}_{S/e,q} \sqcup \mathcal{P}_{S-e,q}$.*

Recall that the map $\tau : \mathcal{P}_{G,q} \to \mathcal{T}_G$ is the assignment of each G-parking function f to the spanning tree T_f on which a tree growing sequence Σ terminates, and let $\rho : \mathcal{P}_{G,q} \to \mathcal{M}_G$ be the assignment of a monomial to each f. We will, in general, get different ρ, τ for different Σ.

Theorem 3.1.9: *The maps ρ and τ are bijective.*

Proof. It is a well-known fact that the sizes of the three sets $\mathcal{P}_{G,q}, \mathcal{T}_G, \mathcal{M}_G$ are equal. Hence, it is enough to show that if $f, g \in \mathcal{P}_{G,q}$ are not equal, then $\tau(f) \neq \tau(h)$, and for each $x^\alpha y^\beta \in \mathcal{M}_G$, there is a unique (up to permuting identical elements) f with $\rho(f) = x^\alpha y^\beta$.

Fix $f \neq h$. By Corollary 2.2.3, each splitting produces a bijection between $\mathcal{P}_{S,q}$ and $\mathcal{P}_{S/e,q} \sqcup \mathcal{P}_{S-e,q}$, where $S = G - \{e\}_D$ according to the edges previously deleted. As we never contract edges, we view $\mathcal{P}_{S/e,q}$ as the set of parking functions such that $e = \sigma_S(T)$ is added to X. If $\tau(f) = \tau(h)$, then the same set of edges $\{e^1, \ldots, e^m\}$ is contracted in the paths for both. However, this implies that either f, h have the same path, which implies $f = h$; or f and h split and have the same edges contracted. This is impossible, as there is some e for which f is in the contraction set, and h is in the deletion set. Therefore, $e \in T_f$, but $e \notin T_h$, and $\tau(f) \neq \tau(h)$.

The statement that each monomial $x^\alpha y^\beta$ in the multiset $\mathcal{M}_G$ has a unique preimage $\rho^{-1}(x^\alpha y^\beta) \in \mathcal{P}_{G,q}$ up to permuting repeated elements can be proven by splitting $\mathcal{M}_G$ using formula (1) in section 3.1.1. If e is neither a bridge nor loop, then $\mathcal{M}_G = \mathcal{M}_{G/e} \sqcup \mathcal{M}_{G-e}$; if e is a loop, then $\mathcal{M}_G = y \cdot \mathcal{M}_{G-e}$; and if e is a bridge, $\mathcal{M}_G = x \cdot \mathcal{M}_{G/e}$. Hence, if $e = \sigma_S(T)$ is a loop or bridge, no splitting occurs. If e is neither, then $\mathcal{M}_{G/e}$ corresponds to $\mathcal{P}_{G/e}$, and $\mathcal{M}_{G-e} = \mathcal{P}_{G-e}$. The result of iterating the process until it terminates is that each $f \in \mathcal{P}_{G,q}$ is in correspondence with a unique element of $\mathcal{M}_G$ (again, up to permuting identical monomials). $\square$

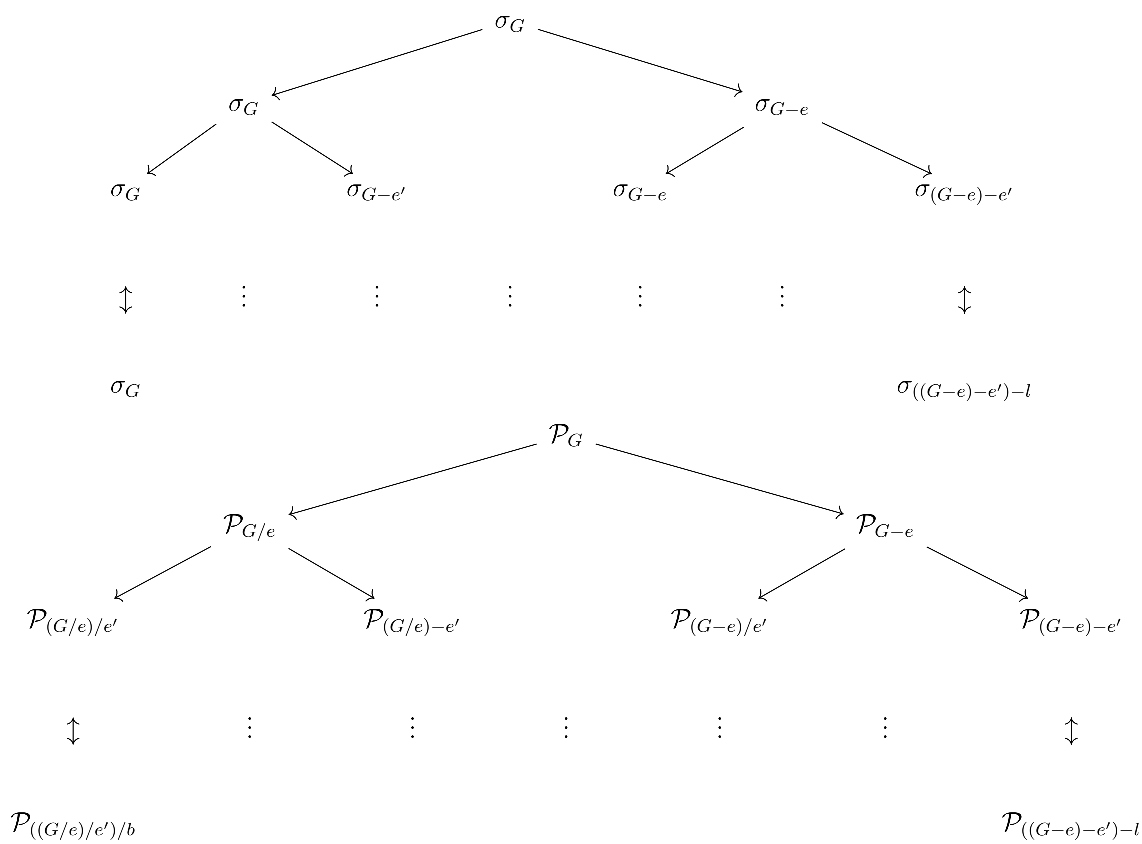


Figure 3.4: Binary trees illustrating how splitting the parking functions corresponds to the application of Σ. Here, b means bridge and l means loop. Note that the edges denoted e' are not necessarily the same on each side of the tree. We can replace $\mathcal{P}$ with $\mathcal{M}$ and the splitting looks the same.

3.2 Comparison of the TGS to Known Algorithms

This section is dedicated to relating tree growing sequences to formerly established bijective algorithms between the three objects of interest. We focus on Dhar's algorithm in Section 3.2.1 and the family of bijections described by Chebikin and Pylyavskyy [CP05] in Section 3.2.2. The bijection between G-parking functions and $\mathcal{M}_G$ given by Chang, Ma, and Yeh [CMY10] is discussed in Section 3.2.3.

3.2.1 Global Edge Orders and Dhar's Algorithm

Given a global edge order $O_E : E(G) \to \{1, \ldots, |E(G)|\}$, we construct a tree-growing sequence Σ_{O_E} by defining for all $S \subseteq E(G)$ and subtrees $T \subseteq S$ the image $\sigma_S(T) = e$ to be the largest available edge which maintains a connected graph at each step. Call this construction the map $R : \{O_E\} \to \{\Sigma\}$ from the set of edge orders to the collection of tree growing sequences. This definition of Σ_{O_E} mimics Dhar's burning algorithm "with memory"; see, i.e. [BS13], [CB03] for explicit algorithms and proofs of the Dhar bijection between G-parking functions (also referred to as q-reduced divisors) and spanning trees using a total edge order. In the notation for the TGS algorithm, Dhar's algorithm chooses the edge $e = max_{O_E}\{(v,u) \,|\, u \in U, v \notin U\}$. The edge e is added to X if $f(v) = |\{(v,u) \in E(G) - E(S)\}|$. Thus, the definition of Σ_{O_E} is almost the same, except it may attempt to grow an edge which creates a cycle. Denote $D_{O_E}(f)$ the image of f under Dhar's algorithm with edge order O_E. For a chosen root q, let Σ_q denote the above TGS where we start at the root.

Proposition 3.2.1: *The map $R : \mathcal{O}_E \to \Sigma_q$ commutes with Dhar's algorithm, for any root q.*

Proof. The map R is defined as above. Fix a root q. If $q \in T \subseteq S$, then $\sigma_S(T) = e$, where $e = \ max_{O_E}(E(S) - E(T))$ and $T \cup e$ is connected. If $h(e) \notin U, t(e) \in U$, then the edge is the same one chosen in Dhar's algorithm. Furthermore, the edge e is deleted if $f(v) \geq 1$ (including after being reduced) which is equivalent to $f(v) > |\{e \in E(S) - X \,|\, v \in e\}|$.

The edge is added to X if precisely $f(v)$ edges incident to v have been deleted. On the other hand, if $h(e), t(e) \in U$, then e will be deleted. Thus, we do not add this edge to X, and since such an edge is never considered in Dhar's algorithm - it is ignored - the diagram commutes. $\square$

$$\begin{tikzcd} \{O_E\} \arrow[r, "R"] \arrow[dr, "D"'] & \{\Sigma\} \arrow[d, "F"] \\ & \{\tau\}, \end{tikzcd}$$

Applying Dhar's algorithm to a G-parking function will also give a bijection with $\mathcal{M}_G$ via the notions of *internal* and *external activity* of the edges of $D_{\mathcal{O}_E}(f)$; this is how Tutte originally defined the polynomial in [Tut54]. An edge e is internally active if it is smallest, according to O_E, in the unique cut-set of $(G - T) \cup e$. Dually, an edge is externally active if it is the smallest in the unique cycle of $T \cup e'$. The Tutte polynomial can be written as a sum over $\mathcal{T}_G$:

$$T(G; x, y) = \sum_{\mathcal{T}_G} x^{ia} y^{ea}$$

with ia and ea denoting the number of internal and external edges, respectively, of the tree T according to $\mathcal{O}_E$. Commutativity of the diagram implies that $\tau(f)$ is the same monomial corresponding to T_f in the above sum. On the other hand, one can ask if the internally and externally active edges match with the bridges and loops in the tree growing sequence algorithm.

Proposition 3.2.2: *If an edge $e \in E(G)$ is internally active for the tree T_f, then it is a bridge when added to X during application of Σ_{O_E}.*

Proof. Say e contributes to the exponent α, where $\tau(f) = x^\alpha y^\beta$. Then at some step of applying Σ_{O_E} to f, $\sigma_S(T) = e$ is a bridge of S. Hence, either e is a bridge of G, or there is a circuit C in S of which e is the smallest among any adjacent edge e' in $G - S$ - i.e. edges which have already been deleted in the tree growing process. Then it is the smallest edge in the unique cocircuit B of $(G - T_f) \cup e$ containing e and any e' as described above. $\square$

Corollary 3.2.3: *The following diagram commutes.*

$$\begin{tikzcd}
\{O_E\} \arrow[r, "R"] \arrow[dr, "K"'] & \{\Sigma\} \arrow[d, "F"] \\
 & \{\rho\}
\end{tikzcd}$$

3.2.2 Proper Sets of Tree Orders

In [CP05], a family of bijections between G-parking functions and spanning trees is produced using an object called a *proper set of tree orders*, Π_G. Let $G = (V, E)$ be a graph and choose a labeling of the vertices $\{v_1, \ldots, v_n\}$. Given an ordering $\pi(T)$ on the vertices of every subtree T rooted at q, the collection $\Pi_G = \{\pi(T)\,|\,T \subset G \text{ a rooted tree}\}$ is a proper set of tree orders if and only if the orders are compatible in the obvious way on overlaps (rooted at q) and a directed edge $(u, v) \in T$ means $v < u$ in $\pi(T)$. Specifically, the former means that if the overlap of T and T' contains a rooted tree, and i, j are vertices in this overlap, then $i <_{\pi(T)} j \iff i <_{\pi(T')} j$. Let $\pi(T)(q) = 0$ for any T. Note that if the trees T and T' differ only by a choice of a set of multi-edges, the orders $\pi(T)$ and $\pi(T')$ must be the same.

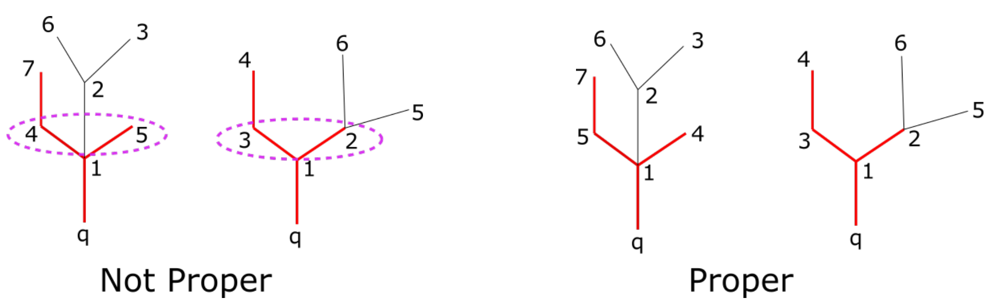


Figure 3.5: Illustrating one criterion for a proper set of tree orders.

Examples of proper sets of tree orders include tree orders induced by vertex orderings constructed by *breadth-first*, *depth-first*, and *vertex adding* algorithms. These orders can all be constructed from the example below.

Example 3.2.4: [CP05] One way to construct Π_G is from a partial order on the set of (open) paths ending at q. The partial order must satisfy the conditions that paths which intersect along another path at q are comparable, and $A \preceq A \cup < v_k, \ldots, v_k' >$.

The partial order $\preceq$ descends to a proper set of tree orders $\Pi_{\preceq}$. Given any rooted subtree $t \subset G$, and distinct vertices $v, w \in V(T)$, the order $\pi_{\preceq}(T)$ is determined by the ordering of the paths from q to v and from q to w. Since these paths intersect along a path starting at q, they are comparable. However, not every Π_G arises in this manner, see [CP05].

We define a map $\Omega : \{\Pi_G\} \to \{\Sigma\}$. Fix Π_G. Consider any subgraph $S \in \mathcal{S}$, and any rooted subtree $T \subseteq S$. Then define $\sigma_S(T) = e$ according to the following:

(i) (a) Take the smallest edge according to $\pi(T)$ from every vertex a neighbor of T. Call this tree T'.

(b) Let $\sigma_S(T) = e$ be the edge in T' such that $h(e)$ is the smallest vertex in $V(T') - V(T)$ according to $\pi(T')$.

(ii) If there is no edge in S which satisfies (i), let $\sigma_S(T) = e'$ for the smallest possible edge e' induced by $\pi(T)$ such that $T \cup e'$ is connected.

If no edge satisfies (i) or (ii), $\sigma_S(T)$ is undefined. Again, this happens when T is equal to the connected component of S containing q.

For example, given $\pi(T) = \{q, \ldots, u_r\}$ where u_j is the $j-th$ vertex in the order, defining σ_T as above ensures that we grow T according to the order $\pi(T)$. That is,

$$\sigma_T(T_k) = e, T_k = (V_k = \{q, u_1, \ldots, u_k\}, E_k), e = (u_{k+1}, u) \in T, u \in V_k\}.$$

Proposition 3.2.5: *The map Ω is an injection from the collection of proper sets of tree orders to the collection of tree growing sequences.*

Proof. For each $T \subseteq S \subseteq G$, there is a unique image $\sigma_S(T) = e$, when defined. If not, there are two edges e, e' satisfying the conditions. This means e, e' are both minimal according to either (i) or (ii), which is impossible. Assembling this data into maps σ_S and letting σ_H be undefined for $q \notin H \subset G$ is precisely the data of a tree growing sequence $\Sigma = \{(S, \sigma_S)\}$.

To show injectivity, we must show that if $\Omega(\Pi_G^a) = \Omega(\Pi_G^b)$, then $\Pi_G^a = \Pi_G^b$. Suppose otherwise. Then there is a rooted subtree $T' \in G$ such that $\pi_G^a(T') \neq \pi_G^b(T')$. Assume that Π_G^a and Π_G^b differ at the k-th vertex, i.e. $u_k^a \neq u_k^b$. Then $\sigma_{T'}^a(T'_{k-1}) \neq \sigma_{T'}^b(T'_{k-1})$, which implies that $\Omega(\Pi_G^a) \neq \Omega(\Pi_G^b)$. Therefore, Ω is injective. $\square$

Example 3.2.6: We will borrow an example from [CP05], pp 33-34, where Π_G is the proper set of tree orders such that $i <_{\pi(t)} j$ if either $d_t(q, i) < d_t(q, j)$, or the distances are equal and $i < j$ in G. Several cases are presented.

In case 1, we have the subtree t of T_1 (left) and T_2 (right) shown with dotted edges. If $S = G$ with vertex order given, then we must have $\sigma_G(t) = (2, 1)$. If we delete $(2, 1)$, we have the subgraph S (below G), and $\sigma_S(t) = (2, 3)$.

In case 2, consider the subtree t'. We need to know how to grow t' - if at all - in a given subgraph. First, let $S = G$. All spanning trees containing t' are shown. We can check that we must define $\sigma_G(t') = (2, 1)$. If we remove $(2, 1)$, the map $\sigma_S, S = G - (2, 1)$ will have image $(3, 1)$ when applied to t'. The graphs to the right of S are maximal subtrees of S.

One may observe that in light of the definition of $\sigma_G(T_2)$, the definition of $\sigma_G(T_1)$ is excess data, because we would be deleting the edge $(2, 1)$ before growing the edge $(2, 3)$. However, we want to define σ_S on all edge subsets S which form a connected subgraph, whether or not the data will be needed when applying $\Omega(\Pi)$.

The above proposition establishes that any proper set of tree orders can tell us how to proceed with the TGS algorithm. However, it is desirable to have commutativity of the diagram in the theorem below. Before the theorem, we describe the bijective map $\Phi_\Pi : \mathcal{P}_{G,q} \to \mathcal{T}_G$, first given in [CP05]. Fix $f \in \mathcal{P}_{G,q}$. Declare $p_0 = q$ and $T_0 = \{q\}$. At each step k, let T_{k-1} be the

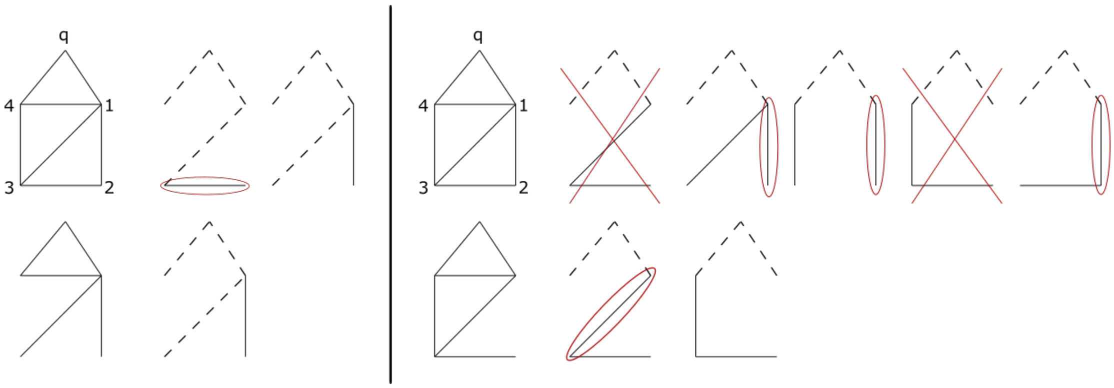


Figure 3.6: Some elements of the tree growing sequence $\Omega(\Pi_G)$.

current subtree grown. The next edge to be grown, denoted $e_k = (p_k, v), v \in V(T_k)$, is the one that satisfies these conditions:

1. There are at least $f(p_k) + 1$ edges from p_k to T_{k-1},
2. The edge e_k is larger than precisely $f(p_k)$ of these edges, and
3. The vertex p_k is minimal among all vertices with edges satisfying $(i), (ii)$, according to the order of the tree obtained from T_{k-1} by adjoining these edges.

The labels $p_0, \ldots, p_n$ comprise the order $\pi(T_f)$, in that $p_0 <_{\pi(T)} \cdots <_{\pi(T)} p_n$ ([CP05], *Lemma 2.3*).

Theorem 3.2.7: *The following diagram commutes.*

$$\begin{tikzcd} \{\Pi_G\} \arrow[r, "\Omega"] \arrow[dr, "\Phi"'] & \{\Sigma\} \arrow[d, "F"] \\ & \{\tau\} \end{tikzcd}$$

Proof. Fix $f \in \mathcal{P}_{G.q}$ and $\Pi \in \{\Pi_G\}$. It will be shown that $\Omega(\Pi)(f) = \Phi_\Pi(f)$. We will argue that if an edge is added to X when applying $\Omega(\Pi)$ to f, then it is in $\Phi_\Pi(f)$. Since we know $T_f = (V, X)$ is spanning by Proposition 2.1.1, this will prove the claim.

Consider the algorithm for constructing $\Phi_\Pi(f) = T_n$. At step k, let V_k be the vertices not in T_{k-1}, $U_k \subseteq V_k$ the vertices adjacent to some vertex in T_{k-1}, and W_k the set of vertices satisfying (1). For $k = 1$, we consider vertices with at least $f(v) + 1$ edges to q. The edge

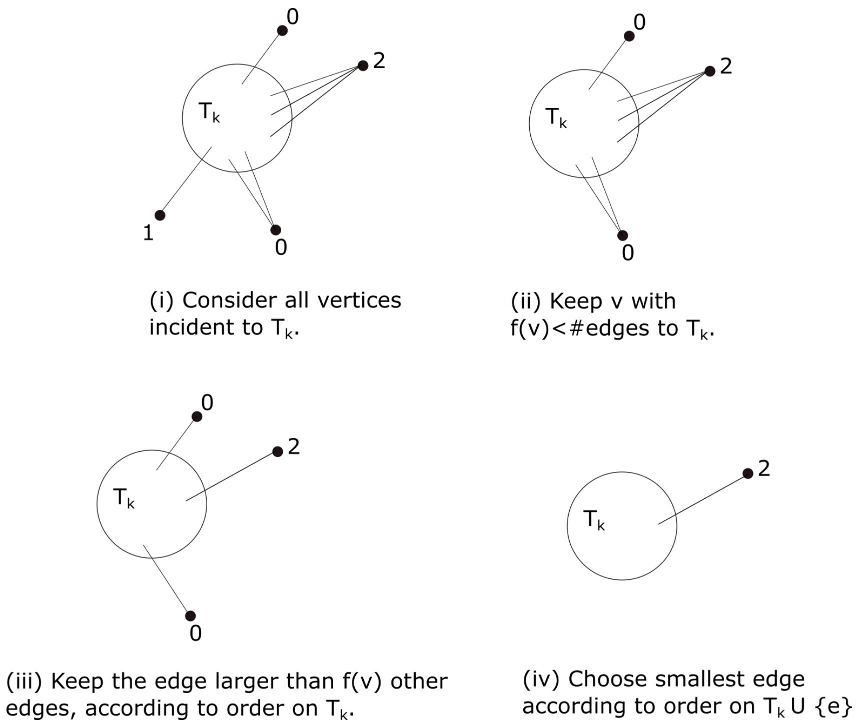


Figure 3.7: Illustration of the algorithm for Φ_Π.

(v, q) satisfying condition (2) will be in $E(T_n)$, for all $v \in W_1$. This is because $\pi(T)(q) = 0$ for all T, so any edge from v to future vertices in T is larger than (v, q). Hence, when applying $\Omega(\Pi)$ to f, if $\sigma_S(T) = (v, q)$, it will be added to X. Thus, the first edge to be added to X when applying $\Omega(\Pi)$ to f will be in $E(T_n)$.

We make a few observations.

- *Observation A:* For any $v \in U_k$, we know that if $e <_{\pi(T_{k-1})} e'$, e and e' both edges from v to T_{k-1}, then $e <_{\pi(T_k)} e'$, and $e, e' <_{\pi(T_k)} e'' = (p_k, v)$, if such an edge exists.

- *Observation B:* When $v \in W_k$, we know the $f(v)$ edges which the map Φ_Π "ignores". That is, the set of smaller edges in condition (2). Call this set E_v.

- *Observation C:* The edge (v, u) which will eventually connect v to T_k for some k is determined as soon as $v \in W_k$.

Elaborating on observation C, suppose $v \in W_k$ for $m \leq k \leq m + i$; i.e. v is in W_k for the first time when $k = m$, and is added to T when $k = m + i$. Then by observation A, the order on the set $E_v \cup e_{m+i}$ is immutable for each of these W_k. In particular, the edges in E_v are always smaller than e_{m+i}. Thus, only condition (3) is not satisfied until $k = m + i$. At step $m + i$, the edge e_{m+i} is smaller than any edge from v to T_{m+i} that is not in E_v.

Assume by induction that thus far $T = (U, X) \subset T_n$. Then the next edge $e = \sigma_S(T)$ that is added to X will be in T_n. Indeed, suppose $\sigma_S(T) = e$ is deleted. Then we know e satisfies $(i), (ii)$, but $f(e_0) \geq 1$. This is true until $\sigma_S(T) = e'$ is added to X. The edge e' is greater than exactly $f(e'_0)$ edges from e_0 to T by observation A, and by observations B and C, we know that e' must be in $E(T_n)$. □

Note that there may be several ways to define a map $\{\Pi_G\} \to \{\Sigma\}$. However, Ω was specifically defined so that it is injective.

We observe that the order in which the vertices are added to T_f according to $\Omega(\Pi)$ may not be the same as the order $\pi(T_f)$. An example is shown in Figure 3.8. Let Π_G be the proper set of tree orders in Example 3.2.5. The top row shows the global order on the vertices. The middle shows the G-parking functions and their images under Φ_Π. The bottom row is the order in which the vertices are added when applying $\Omega(\Pi)$ to the corresponding functions. Nonetheless, the bijection between $\mathcal{P}_{G,q}$ and $\mathcal{T}_G$ is the same.

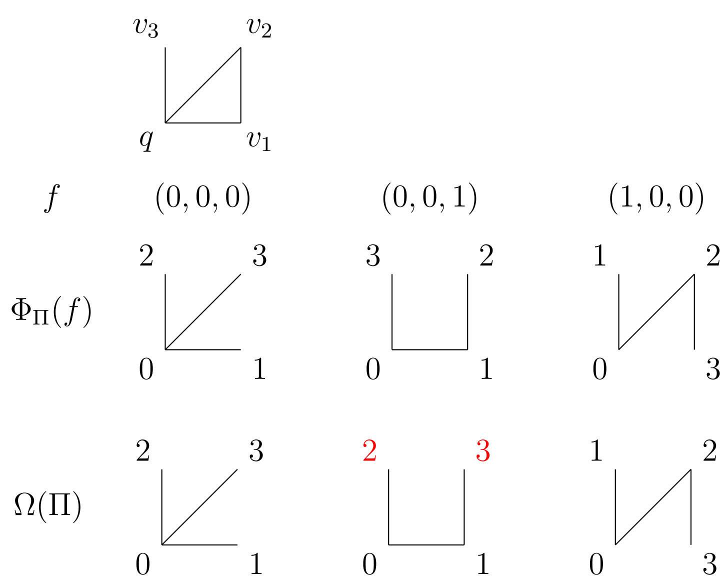


Figure 3.8: The order in which vertices are added does not match between the maps.

3.2.3 Process Orders

A bijection between G-parking functions and monomials of the Tutte polynomial that does not go through spanning trees was constructed by Chang et al [CMY10]. We describe this bijection and compare it to a tree growing sequence. Fix a total order $O_V : V \to \{0, \dots, n\}$ on the vertices of G. For each $f \in \mathcal{P}_{G,q}$, associate a *process order* π_f [KY08]. This is done recursively as follows:

1. Let $\pi_f(0) = v_0 = q$, and $V_0 = V(G) - \{q\}$.

2. Let $\pi_f(i) = min_{O_V}\{w \in V_{i-1} \mid 0 \leq f(w) < outdeg_{V_{i-1}}(w)\}$, where the vertices in $V - V_{i-1}$ have been processed.

3. Increase i by one, and repeat step 2 until all vertices have been processed; i.e. when $i = n - 1$.

One can get the process order for $f \in \mathcal{P}_{G,q}$ from a tree growing sequence. If T does not span S, define Σ_{O_V} by $\sigma_S(T) = (v, u)$, such that $v = min_{O_V}\{w \in V(S) - V(T)\}$, and u is the smallest neighbor of v in T. If T spans S, then define $\sigma_S(T)$ to be the smallest edge according to the lexicographic order induced by O_V. Apply Σ to any f. When the i-th edge is added to $X = E(T)$, identify V_{i+1} with the vertices in $V - V(T)$. If $\sigma_S(T) = (v, u)$ is the edge added to X, then $\pi_f(|X|) = v$. Thus, v will be added to $V(T)$ when at least $f(v)$ edges from v to T have been deleted and it is the minimum among all such candidate vertices, which is exactly statement 2 above.

Denote $K = \{u \in V(G) \mid \pi^{-1}(v) \leq \pi^{-1}(u)\}$. This leads to the definition of a critical bridge vertex of f.

Definition 3.2.8: *A critical bridge vertex v of the parking function f with $\pi_f(i) = v$ is one for which $outdeg_K(v) = f(v) + 1$ in G (criticality), and for every parking function h satisfying: $g(\pi_h(j)) = f(\pi_f(j))$, with $\pi_h(j) = \pi_f(j)$ for $j < i$; $h(v) \geq f(v)$; and $\pi_h(i) \geq_{O_V} v$; we have, in fact, that $\pi_h(i) =_{O_V} v$.*

The last inequality says that there is no vertex strictly greater than v according to O_V which is processed at the same step as v for some other G-parking function h. Let $cb_G(f)$ be the number of critical bridge vertices of f, and $w_G(f) = |E| - |V| + 1 - \sum_{v \in V - \{q\}} f(v)$.

Theorem 3.2.9 [CMY10]: *The Tutte polynomial of G with fixed root q can be expressed as the following closed formula:*

$$T(G; x, y) = \sum_{f \in \mathcal{P}_{G,q}} x^{cb_G(f)} y^{w_G(f)}$$

Any tree growing sequence can be viewed as a way to write such a closed formula from the bijection $\rho : \mathcal{P}_{G,q} \to \mathcal{M}_G$. Simply say $T(G; x, y) = \sum_{f \in \mathcal{P}_{G,q}} \rho(f)$. However, any bijection comes with another bijection $\tau : \mathcal{P}_{G,q} \to \mathcal{T}_G$. We think that the above theorem secretly constructs a spanning tree and can be obtained via some tree growing sequence Σ. Specifically, the tree growing sequence Σ_{O_V} defined above is the most likely candidate, and our conjecture has evidence through several calculations. However, we have not translated the constructions in [CMY10] to our language of tree growing sequences, and at this point we cannot verify the conjecture.

Chapter 4

Tutte Polynomial of a Zonotopal Tiling

4.1 Zonotopes

4.1.1 Zonotopal Tilings

In the same spirit as the tree growing sequence algorithm, we describe a splitting algorithm which can be used to obtain the Tutte polynomial for a cubical tiling of a zonotope. Let M be an n-dimensional vector space, $N = M^\vee$, and $<,>$ the pairing of N with M (viewed as the standard inner product on $\mathbb{R}^n$).

Definition 4.1.1: A **zonotope** is the image of a d-dimensional cube $Q_d = [0,1]^d$ under an affine projection. Equivalently, it is a Minkowski sum

$$Z = \{a_1 v_1 + \cdots + a_d v_d \,|\, 0 \leq a_i \leq 1,\, v_i \in M \cong \mathbb{R}^n\}.$$

We say that the set $X = \{v_1, \ldots, v_d\}$ *generates* Z.

We now discuss zonotopal tilings, where many of the details can be found in [PDC10], [RGZ93], and [Zie95].

Definition 4.1.2: A **parallelotope** is a zonotope generated by vectors which form a basis of $M \cong \mathbb{R}^n$.

Definition 4.1.3: A **cubical zonotopal tiling** $\mathcal{Z}$ of Z is a polyhedral complex comprised of a finite number of zonotopes $\{Z_i\}$ such that the maximal dimensional zonotopes - called **tiles** - are parallelotopes, and $\bigcup_i Z_i = Z$.

Let Z be generated by $\{v_1, \ldots, v_d\}$, and let $\{\mathcal{E}_j\}$ be the equivalence classes of edges of a tiling $\mathcal{Z}$, where an equivalence class is generated by the edges which are opposite a 2-dimensional face of $\mathcal{Z}$. Pick a representative w_j of each $\mathcal{E}_j$. Each w_j is parallel to a vector in the generating set $\{v_1, \ldots, v_d\}$. As a result, we break each v_i in to a finite number of vectors $w_1^{(i)} = k_1^{(i)} v_i$, $w_2^{(i)} = k_2^{(i)} v_i, \ldots, w_l^{(i)} = k_l^i(i) v_i$, such that $k_j^{(i)} > 0$, $\sum k_j^{(i)} = 1$, and $\sum_j w_j^{(i)} = v_i$. Then we can associate to $\mathcal{Z}$ the vector configuration $V_{\mathcal{Z}}$ containing the vectors $\{w_j^{(i)}\}$ for all $1 \leq i \leq d$.

Definition 4.1.4: Let $\mathcal{E}$ be an equivalence class of edges as above with representative w. A **zone** B_w of a zonotope Z is the set of tiles which contain an edge in $\mathcal{E}$. Two zones B_w, $B_{w'}$ are **parallel** if w' is parallel to w.

Each zone B_w has a *positive side* $Z^{w,+}$ and *negative side* $Z^{w,-}$ according to the direction of the vector w. An example is shown below in Figure 7.

Fix a cubical tiling $\mathcal{Z}$ of Z. We compute a polynomial $T^*(\mathcal{Z}; x, y)$ using a splitting algorithm which assigns a monomial to each tile of $\mathcal{Z}$, and $T^*(\mathcal{Z}; x, y)$ is the sum of these monomials. We will be performing two operations - called *shrinking* and *projection* in [RGZ93] - which will split the set of tiles into two disjoint sets at each step.

Definition 4.1.5: Delete the zone B_w and glue the positive and negative sides (see figure 4.1) of B_w along $B_w \cap Z_w^+$ and $B_w \cap Z_w^-$. Denote the result of this operation $Z - B_w$ the **shrinking** of Z with respect to B_w. Explicitly, since w is parallel to kv_i for some $0 < k \leq 1$, we can write this zonotope as

$$Z - B_w = \{a_1 v_1 + \cdots + a_i(1-k)v_i + \ldots a_n v_n \mid 0 \leq a_i \leq 1\}.$$

The tiling of $Z - B_w$ is as before. The associated vector configuration is $V_{\mathcal{Z}} - \{w\}$.

Definition 4.1.6: Define $P_w : M \to M/(\mathbb{R} \cdot w)$. Let $Z|B_w = P_w(B_w)$ be the **projection** of the zone B_w. The tiles of $Z|B_w$ are $\{P_w(Z_i) \mid Z_i \text{ a tile of } B_w\}$. The associated vector configuration is $(V_{\mathcal{Z}} - \{w\})/(\mathbb{R} \cdot w)$.

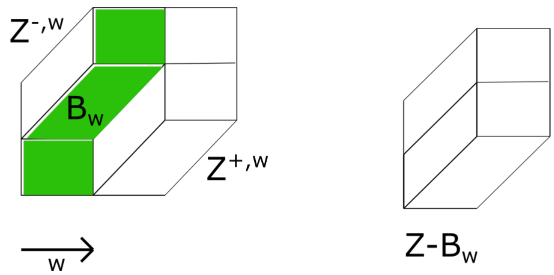


Figure 4.1: Shrinking.

Note that there is a description of Z in terms of $Z - B_w$ and $Z|B_w$ in [PDC10]. Our description is essentially the same, except we keep track of the tilings at each step. The decomposition

$$Z = (Z - B_w) \cup B_w$$

tells us that the set of tiles of $\mathcal{Z}$ splits into the tiles of $Z - B_w$ and tiles of $Z|B_w$.

Start with the tuple (Z, α, β), where initially $\alpha = 0$, and $\beta = 0$. The monomials will be $x^\alpha y^\beta$ where the exponent values will change according to the algorithm. Choose a belt B_w, and apply the shrinking and projection operations. This results in two tuples $(Z - B_w, 0, 0)$ and $(Z|B_w, \gamma, 0)$ associated to the resulting zonotopes, where γ is the number of zones parallel to B_w. If $Z - B_w$ and $Z|B_w$ are zonotopes with the same tiling (i.e. they are equivalent zonotopes), we get a single tuple $(Z - B_w, 0, 1)$. Similar to the TGS algorithm, we repeat the operations for each new zonotope created. When the zonotope has been reduced to a collection of points with assigned tuples $(\bullet, \alpha, \beta)$, define $T^*(\mathcal{Z}; x, y)$ to be the polynomial obtained by summing the monomials $x^\alpha y^\beta$. If we follow the path according to the splitting from each tile of $\mathcal{Z}$ to a point, we can associate a monomial to each tile. Thus, we can write

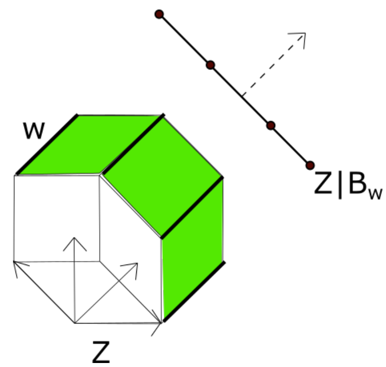


Figure 4.2: Projection with respect to w.

the closed formula

$$T^*(\mathcal{Z}; x, y) = \sum_{\text{tiles of}\mathcal{Z}} x^\alpha y^\beta$$

Written in parallel to the deletion/restriction definition of the Tutte polynomial, and defining $T^*(\bullet; x, y) = 1$, the algorithm gives us the recursive formula:

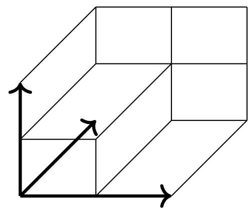
$$T^*(\mathcal{Z}; x, y) = \begin{cases} yT^*(\mathcal{Z} - B_w; x, y) & Z - B_w \cong Z|B_w \\ x^\gamma T^*(\mathcal{Z}|B_w; x, y) + T^*(\mathcal{Z} - B_w; x, y) & \text{otherwise} \end{cases} \tag{4.1}$$

Observation: The exponent γ can be expressed in terms of vector configurations as

$$\gamma = |V_{\mathcal{Z}}| - |(V_{\mathcal{Z}} - \{w\})/\{w\}|,$$

the number of 0-vectors resulting from the projection operation. We will 'ignore' these 0-vectors after projection, and can think of removing them from the configuration.

Example 4.1.7: The zonotope generated by the vectors $v_1 = (2, 0), v_2 = (0, 1.5)$, $v_3 = (1, 1) \in M \cong \mathbb{R}^2$ is a hexagon. Let $V_{\mathcal{Z}} = \{\frac{1}{2}v_1, \frac{1}{2}v_1, \frac{1}{2}v_2, \frac{1}{2}v_2, v_3\}$ be the vector configuration arising from the cubical zonotopal tiling $\mathcal{Z}$ shown below.



The splitting algorithm for $\mathcal{Z}$ is shown in Figure 4.3, where in the first step the belt B_w is chosen, where w is the first $\frac{1}{2}v_1$ in the list. A colored zone means we are shrinking/projecting along that zone. A southwest arrow indicates projection, a southeast arrow indicates shrinking, and a south arrow represents when both are equivalent. Each intermediate zonotope Z_k is tiled and the tiles labeled by the corresponding monomials of $T^*(\mathcal{Z}_k; x, y)$. The arrows are labeled according to where we multiply $T^*(\mathcal{Z}_k; x, y)$ by x^γ or y in the algorithm. The polynomial is $T^*(\mathcal{Z}; x, y) = x^3 + 2x^2 + x + 2xy + y + y^2$, which is the Tutte polynomial for the graph $K_4 - \{edge\}$.

We now recall a few notions related to matroids from chapter 2.5.

Definition 4.1.8: The **rank function** of a matroid $M = (E, \mathcal{I})$ is

$$r : 2^E \to \mathbb{Z}_{>0}$$

$$r(A) = \max_{I \subseteq A, I \in \mathcal{I}} |I|$$

Definition 4.1.9: The **dual matroid** M^* is the pair $(E, \mathcal{I}^*)$, where a set $J \in \mathcal{I}^*$ is independent in M^* if and only if $E - J$ contains a basis of M. The rank function is $r^*(A) = |A| - r(E) + r(E - A)$ is the dual rank function.

Definition 4.1.10: The **Tutte polynomial** of a matroid is defined as

$$T(M; x, y) = \sum_{A \subseteq E} (x-1)^{r(E)-r(A)} (y-1)^{|A|-r(A)} \tag{4.2}$$

If M^* is the matroid dual, then $T(M^*; y, x) = T(M; x, y)$. Evaluating $T(M; 2, 2)$ gives the number of bases of M.

Example 4.1.11: If $G = (V, E)$ is a connected graph, we can define the *cographical matroid* to be the matroid with ground set E and bases $\mathcal{B} = \{\mathbf{b} = E - E(T) \mid \text{T a spanning tree}\}$. Hence, its rank is the genus $g = |E| - |V| + 1$. More thorough expositions on matroids and their duals can be found in the original paper by Whitney [Whi35], and lectures by Tutte [Tut65].

Let $w : E(G) \to \mathbb{R}_{>0}$ be a function assigning length one to every edge of G, so that each edge can be identified to a unit interval. The zonotope $Z(G)$ is the projection of the cube $[0, 1]^{|E(G)|}$ along the lattice of bonds (minimal cut-sets), which are the circuits of the cographical matroid. The dimension of $Z(G)$ is g. Choose a tiling so that the representatives w correspond to the edges of G. Then every tile corresponds to an element $\mathbf{b} \in \mathcal{B}$, the complement of a spanning tree. Hence, each tile corresponds to a unique spanning tree, and we get that any zone B_w is the set of tiles associated to spanning trees which do not contain the edge corresponding to w. Let $\mathcal{W}$ be a matroid with ground set the list of vectors $W = \{w_1, \ldots, w_d\}$ spanning the vector space $U = \mathbb{R}W$, and with independent sets $\mathcal{I} = \{X \mid W - X \text{ is a linearly independent set}\}$. Define $W_1 = W - \{u\}$ and $W_2 = W_1/u$. Then the Tutte polynomial of W satisfies

$$T(W; x, y) = \begin{cases} xT(W_1; x, y) & w \text{ is a coloop } (w = 0) \\ yT(W_2; x, y) & U = \mathbb{R}W_1 \oplus \mathbb{R} \cdot w \text{ (w is a loop)} \\ T(W_1; x, y) + T(W_2; x, y) & U = \mathbb{R}U_1, \ w \neq 0 \end{cases} \tag{4.3}$$

The formula reduces to computing the Tutte polynomial of lists of vectors $V^{(i)} = V_0^{(i)} \sqcup V_1^{(i)}$, where $V_0^{(i)}$ is a list of k linearly independent vectors and $V_0^{(i)}$ is a list of h zero vectors; for such lists, $T(V^{(i)}; x, y) = x^h y^k$. The bases of W are complements of subsets which form a basis for U. See [Moc09], for example, for a treatment of the Tutte polynomial and a multiplicity polynomial for a vector configurations, as well as a discussion of how these polynomials give information about the associated zonotopes.

Observation: Given a zonotopal tiling $\mathcal{Z}$, the bases of the matroid $\mathcal{W}$ with ground set $V_{\mathcal{Z}}$ described above are in bijection with the tiles, since each tile has edges which form a basis for $U = \mathbb{R}V_{\mathcal{Z}}$. We will denote this matroid by $V_{\mathcal{Z}}^*$.

Theorem 4.1.12: *Fix a cubical zonotopal tiling $\mathcal{Z}$ of Z with associated vector configuration $V_{\mathcal{Z}}$. Then $T^*(\mathcal{Z}; x, y)$ is the Tutte polynomial $T(V_{\mathcal{Z}}^*; x, y)$.*

Proof. Suppose we compute the Tutte polynomial $T(V_{\mathcal{Z}}^*; x, y)$ and the polynomial $T^*(\mathcal{Z}; x, y)$ simultaneously, where the choice of w at each step is a *nonzero* vector. If any 0-vectors are created, we choose to remove them immediately from the list. The algorithm for computing $T^*(\mathcal{Z}; x, y)$ gives a bijection

$$\{\text{Tiles of } Z\} \leftrightarrow \{\text{monomials}\}.$$

Hence, both polynomials have the same number of monomials. Moreover, the operations of deletion and restriction applied to $V_{\mathcal{Z}}$ with respect to w yield precisely the vector configurations associated to the tilings of $Z - B_w$ and $Z|B_w$, respectively.

Inductively, let Y be a zonotope with tiling $\mathcal{Y}$ that is created at some step of the algorithm with assoicated vector configuration $V_{\mathcal{Y}}$. Choose $w \neq 0 \in V_{\mathcal{Y}}$. We check that the recursion formulas for the polynomials are the same in all cases.

If $Y - B_w \cong Y|B_w$, then $T^*(\mathcal{Y}; x, y) = yT^*(\mathcal{Y} - B_w; x, y) = yT^*(\mathcal{Y}|B_w; x, y)$. This occurs when Y is a prism of height w, so the vector w is a loop of $V_{\mathcal{Y}}^*$. Then $T(V_{\mathcal{Y}}^*; x, y) = yT(V_{\mathcal{Y}}^* - \{w\}; x, y)$.

If we project Y with respect to w, we multiply $T^*(Z|B_w; x, y)$ by x^γ, where γ is the number of belts parallel to B_w. Recall that this represents throwing out all 0-vectors created by projection. Thus, the integer γ is the number of coloops in $V_{\mathcal{Y}}/\{w\}$, and subsequently contracting all of them gives $T(V_{\mathcal{Y}}^*/\{w\}; x, y) = x^\gamma T(V_{\mathcal{Y}}^*/\{w, 0, \ldots, 0\}; x, y)$.

If we shrink Y with respect to w, and w is not a coloop of W, then the tiles of $Y - B_w$ have the same monomials associated to them as in Y.

Hence, $T^*(\mathcal{Y};x,y) = x^\gamma T^*(\mathcal{Y}|B_w;x,y)+T^*(\mathcal{Y}-B_w;x,y) = x^\gamma T(V^*_{\mathcal{Y}}/\{w,0,\ldots,0\};x,y)+T(V^*_{\mathcal{Z}}-\{w\};x,y)$. This proves that $T^*(\mathcal{Z};x,y) = T(V^*_{\mathcal{Z}};x,y)$. $\square$

Remarks:

1. If a zonotope Z is a prism of height w, then it is a Minkowski sum $Z = Z' + w$, where w orthogonal to Z'; thus, $Z - B_w \cong Z|B_w$. The converse is also true.

2. The bijection between tiles and monomials is dependent on the order in which we choose e. Indeed, if we have the zonotope Z where Z is a segment with two tiles e_1, e_2, then choosing e_1 first will assign x to e_1 and y to e_2. Hence, we can switch the order and get the other possible assignment.

4.1.2 Tiles of a Zonotope and G-parking Functions

We now relate the set of G-parking functions to integer points in the cographical zonotope $Z(G)$. We start with a discussion for which [BLHN97] and [Big99] are used as primary references.

Fix an arbitrary orientation on the edges of G, and write an edge as an ordered tuple $e = (e_h, e_t)$. Let $C_0(G;\mathbb{R}) \cong \mathbb{R}^{|V((G)|}$ and $C_1(G;\mathbb{R}) \cong \mathbb{R}^{|E(G)|}$ be the vector spaces of finite $\mathbb{R}$-linear combinations of the vertices and edges of G, respectively, called the 0-*chains* and 1-*chains*. We have that $Div(G) = C_0(G;\mathbb{Z})$. There is a standard inner product on $C_1(G;\mathbb{R})$ given by $<\sum a_e e, \sum b_e e> = \sum a_e b_e$.

Consider the map

$$C_1 \xrightarrow{d} C_0$$

where $d(\sum_e a_e e) = \sum_e a_e(e_t - e_h)$ is the usual differential. Hence, $d(C_1(G;\mathbb{Z})) = Div^0(G)$, as the image is generated by $d(e) = e_t - e_h$. Denote the 1-cycles by $Z_1 \cong H_1(G;\mathbb{R})$. Note that Z_1 is isomorphic to $\mathbb{R}^g$, where $g = |E| - |V| + 1$. Its orthogonal complement in $C_1(G;\mathbb{R})$ is generated by the cuts b_v. Let $\Lambda = Z_1 \cap C_1(G;\mathbb{Z}) \cong H_1(G;\mathbb{Z})$. We call Λ the *lattice of integral cycles*.

Let P be the orthogonal projection below.

$$C_1(G;\mathbb{R}) \xrightarrow{P} Z_1 \supset \Lambda$$

We now define the Jacobian of the graph again, this time using the language of 1-chains.

Definition 4.1.13: The **Jacobian** of G is the finite group

$$J(G) = \frac{P(C_1(G;\mathbb{Z}))}{\Lambda}.$$

The group operation is addition modulo Λ.

We map $Div^0(G)$ into the real torus $Z_1/\Lambda \cong H_1(G;\mathbb{R})/H_1(G;\mathbb{Z}) \cong \mathbb{R}^g/\mathbb{Z}^g$ as follows. Choose a path p_i - viewed as an element of $C_1(G;\mathbb{Z})$ - from the root q to each vertex v_i, and lift $f = \sum a_i v_i$ to $d^{-1}(f) = \sum_i a_i p_i \in C_1(G;\mathbb{Z})$. Then apply the orthogonal projection P, and take the image modulo Λ.

We get a map

$$A : Div^0(G) \to Z_1/\Lambda$$

$$f \mapsto P(d^{-1}(f))(mod\,\Lambda)$$

This map is a discrete analog of the Abel-Jacobi map originating from complex algebraic geometry. The image is the Jacobian $J(G)$. It is well-defined, as choosing another path p_i' from q to some v_i and lifting f will result in a shift by an element in Λ (that is, $p_i - p_i' \in \Lambda$). Furthermore, if two divisors are linearly equivalent, they are sent to the same point. To see this, observe that linearly equivalent divisors differ by a cut. This leads to the proof of the Abel-Jacobi theorem.

Theorem 4.1.14 (Abel-Jacobi) [BLHN97], [BN06]: *The Abel-Jacobi map induces a group isomorphism between $Pic^0(G)$ and $J(G)$.*

Corollary 4.1.15: *The real torus Z_1/Λ contains $|P_{G,q}|$ integral points.*

Hence, a fundamental domain for Z_1/Λ contains $|P_{G,q}|$ integral points. In fact, the closure of a fundamental domain can be identified with the cographical zonotope $Z(G)$, and we can tile $Z(G)$ as described in Example 4.1.2 of section 4.1.1 (see [ABKS14] for details). We also get the correspondence between spanning trees and tiles. Choose a generic fundamental domain $Z(G)$ in Z_1. The genericity means that integral points will not be vertices of the zonotope, but will lie in the interior of the parallelotopes of the tiling, so that there is precisely one integral point in each tile. Thus, we get a bijection between G-parking functions and tiles of $Z(G)$.

We end with a question which ties together the bijections discussed in this paper. These bijections depend on several choices. There is the choice of a root q for G, which determines the set $\mathcal{P}_{G,q}$. There are possibly several ways to tile the zonotope $Z(G)$ to obtain the bijection between spanning trees and tiles. We have the choice of fundamental domain. There is also the choice of the order in which the algorithm for computing $T^*(\mathcal{Z};x,y)$ is applied. Additionally, we have a choice of a tree growing sequence Σ from which we get the bijective maps ρ and τ. In light of Theorem 3.1.1, the TGS may contain an underlying choice such as a total order on the edges. Thus, we state the following:

Question: *Are there choices which are compatible in that they make the diagram below commute?*

$$\begin{tikzcd}
& \mathcal{P}_{G,q} \arrow[dl] \arrow[dr, leftrightarrow] \arrow[dd, leftrightarrow] & \\
\mathcal{M}_G & & \{\text{Tiles of } Z(G)\} \arrow[ll] \\
& \mathcal{T}_G \arrow[ul] \arrow[ur, leftrightarrow] &
\end{tikzcd}$$

The map $\mathcal{T}_G \to \mathcal{M}_G$ is given by a total edge order O_E from which we read off internal and external activities. The maps from $\mathcal{P}_{G,q}$ to $\mathcal{M}_G$ and $\mathcal{T}_G$ are the bijections ρ and τ arising from a tree growing sequence Σ. We have double-headed arrows for where there are known invertible algorithms.

Theorem 4.1.16: *The lower triangle is commutative.*

Proof. Fix a tiling $\mathcal{Z}$ of $Z(G)$ and a correspondence between edges of G and elements of $V_{\mathcal{Z}}$, which produces the correspondence between tiles and spanning trees. Compute $T^*(\mathcal{Z}; x, y)$, but keep track of additional data. For every edge $e \in E(G)$, let $z(e)$ be the number of times e is a coloop (parallel to the element w chosen) or a loop. Let $e_i < e_j$ if $z(e_i) > z(e_j)$. If $z(e_i) = z(e_j)$, arbitrarily choose which is larger. The resulting total order $e_{i_1} < \cdots < e_{i_m}$ induces a bijection from $\mathcal{T}_G$ to $\mathcal{M}_G$ via external/internal activities, and will match the bijection induced by the zonotope algorithm. In other words, we are determining how active an edge is through this count. As a general rule, the earlier an edge is chosen in a path, the less active it will be, and the higher it is in the total order. $\square$

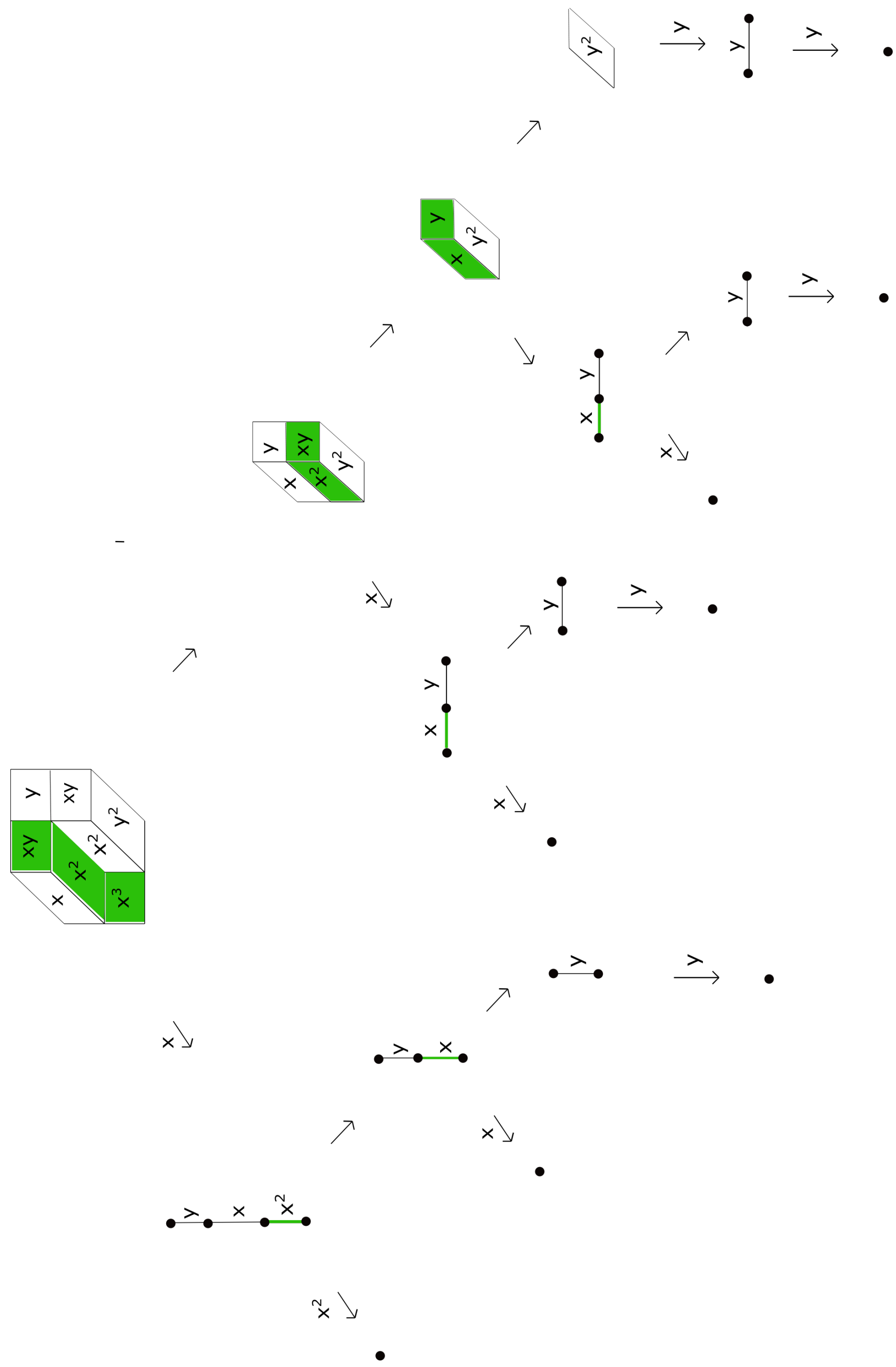


Figure 4.3: $T^*(\mathcal{Z};x,y) = x^3 + 2x^2 + x + 2xy + y + y^2$

Bibliography

[ABKS14] Yang An, Matthew Baker, Greg Kuperberg, and Farbod Shokrieh. Canonical representatives for divisor classes on tropical curves and the matrixâĂŞtree theorem. *Forum of Mathematics, Sigma*, 2: e24, 2014.

[Ber08] Olivier Bernardi. A characterization of the Tutte polynomial via combinatorial embeddings. *Annals of Combinatorics*, 12(2):139–153, 2008. 14 pages.

[Big99] Norman Biggs. Chip-firing and the critical group of a graph. *Journal of Algebraic Combinatorics*, 9(1): 25–45, Jan 1999.

[BLHN97] Roland Bacher, Pierre de La Harpe, and Tatiana Nagnibeda. The lattice of integral flows and the lattice of integral cuts on a finite graph. *Bulletin de la Société Mathématique de France*, 125(2):167–198, 1997.

[BN06] Matthew Baker and Serguei Norine. Riemann-Roch and Abel-Jacobi theory on a finite graph. *arXiv Mathematics e-prints*, page math/0608360, Aug 2006.

[BO92] Thomas Brylawski and James Oxley. *The Tutte Polynomial and Its Applications*, page 123âĂŞ225. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1992.

[Bru12] R.A. Brualdi. *Introductory Combinatorics*. Pearson Education. Pearson Education International, 2012.

[BS13] Matthew Baker and Farbod Shokrieh. Chip-firing games, potential theory on graphs, and spanning trees. *Journal of Combinatorial Theory, Series A*, 120(1): 164 – 182, 2013.

[BTW87] Per Bak, Chao Tang, and Kurt Wiesenfeld. Self-organized criticality: An explanation of the 1/f noise. *Phys. Rev. Lett.*, 59:381–384, Jul 1987.

[CB03] Robert Cori and Yvan Le Borgne. The sand-pile model and tutte polynomials. *Advances in Applied Mathematics*, 30(1):44 – 52, 2003.

[CMY10] Hungyung Chang, Jun Ma, and Yeong-Nan Yeh. Tutte polynomials and g-parking functions. *Advances in Applied Mathematics*, 44(3): 231 – 242, 2010.

[CP05] Denis Chebikin and Pavlo Pylyavskyy. A family of bijections between g-parking functions and spanning trees. *Journal of Combinatorial Theory, Series A*, 110(1):31 – 41, 2005.

[Dha90] Deepak Dhar. Self-organized critical state of sandpile automaton models. *Phys. Rev. Lett.*, 64: 1613–1616, Apr 1990.

[GH78] P. Griffiths and J. Harris. *Principles of Algebraic Geometry.* New York: John Wiley and Sons, 1978.

[KY08] Dimitrije Kostić and Catherine H. Yan. Multiparking functions, graph searching, and the tutte polynomial. *Advances in Applied Mathematics*, 40(1):73 – 97, 2008.

[Mer05] Criel Merino. The chip-firing game. *Discrete Mathematics*, 302(1):188 – 210, 2005. Structural Combinatorics.

[Moc09] Luca Moci. A tutte polynomial for toric arrangements. *Transactions of the American Mathematical Society*, 364, 11 2009.

[MZ08] G. Mikhalkin and I. Zharkov. Tropical curves, their jacobians and theta functions. *Contemporary Mathematics*, 465: 203–230, 2008.

[PDC10] Claudio Procesi and Corrado De Concini. *Topics in Hyperplane Arrangements, Polytopes and Box-Splines.* Springer-Verlag New York, 2010.

[Pyk59] Ronald Pyke. The Supremum and Infimum of the Poisson Process. *The Annals of Mathematical Statistics*, 30(2):568 – 576, 1959.

[RGZ93] Jürgen Richter-Gebert and Günter M. Ziegler. Zonotopal tilings and the bohne-dress theorem. Technical Report SC-93-25, ZIB, Takustr. 7, 14195 Berlin, 1993.

[Tut54] W. T. Tutte. A contribution to the theory of chromatic polynomials. *Canadian Journal of Mathematics*, 6:80–91, 1954.

[Tut65] William T. Tutte. Lectures on matroids. *J. Res. Natl. Bur. Stand., Sec. B: Math. and Math. Phys.*, 69B(1-2):1–47, 1965.

[Whi35] Hassler Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3):509–533, 1935.

[Zie95] Gunter M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag New York, 1 edition, 1995.

Chapter 5

Appendix A: Divisor Theory of Metric Graphs

A (compact) metric graph is a compact and connected metric space such that every point $p \in \Gamma$ has a neighborhood isometric to a star-shaped set. A discrete graph G gives rise to a metric graph Γ by assigning lengths to the edges of G, and viewing 2-valent vertices as interior points. We call G a *model* for Γ.

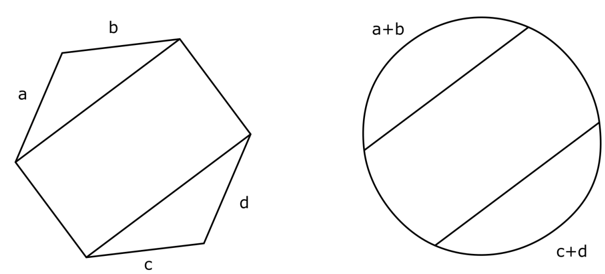


Figure 5.1: A graph G on the left with 2-valent vertices. It is a model for the metric graph on the right (where it is assumed that unlabeled corresponding edges in the graphs have the same length.

There is a one-to-one correspondence between compact tropical curves and metric graphs, and it is convenient to view tropical curves in this way. Connected metric graphs Γ_1, Γ_2 are equivalent as tropical curves if after removing 1-valent vertices and the edges adjacent to them, and treating 2-valent vertices as interior points of edges, they are the same metric graph. When a tropical curve is viewed as a metric graph Γ, the genus is the first Betti number $g = b_1(\Gamma)$. The metric theory is established, and many of the below results found, in [MZ08].

Classically, the Abel-Jacobi map embeds an algebraic curve C in a complex torus, called the Jacobian of C, by fixing a basepoint p_0 and sending a point in C to the vector obtained by integrating each basis element in $H^0(C, \Omega)$ over the path from p_0 to p. The coordinate free definition is

$$J(C) = (H^0(C, \Omega))^*/H_1(C, \mathbb{Z}).$$

This map extends to divisors on C. The definition is analogous for a metric graph, though it is common to use the canonical identification $Jac(\Gamma) = H_1(\Gamma, \mathbb{R})/H_1(\Gamma, \mathbb{Z})$. There is no difference with the definition of a divisor in the previous section. There is a tropical analog to the Abel-Jacobi map. Fix Γ. Denote by $Div^k(\Gamma)$ the group of divisors of degree k. There is an equivalence relation $\sim$ of the group of divisors via principal divisors, as in the discrete case.

Definition 5.1: A *principal divisor* D on Γ is one which comes from a continous, piece-wise linear function f on Γ with integer slopes (i.e. a tropical rational function). Then D is given by $\Delta(f)$, where Δ is the Laplacian operator. With this definition, we see that

$$\Delta(f) = \sum_{p \in C} \sum_{i} \left(\frac{\partial f}{\partial \xi_i}(p) \right) \cdot p$$

where ξ_i is an outward primitive tangent vector of f at p, one for each edge emanating from p. Moreover, a tropical curve is a balanced polyhedral complex, so that $deg(\Delta(f)) = 0$.

As before, the Picard group is defined as $Pic(\Gamma) = Div(\Gamma)/(D \sim 0)$. One manifestation of the Abel-Jacobi Theorem is that the Abel-Jacobi map factors through this equivalence, and ϕ is a bijection. Furthermore, it is an isomorphism for $k = 0$, and we have $Jac(\Gamma) = Pic^0(\Gamma)$. Both the Picard group and the Jacobian of a metric graph are real tori.

If we fix a basepoint $q \in \Gamma$, then ϕ is given by translation by $-k \cdot q$, and it is easy to check

$$
\begin{tikzcd}
Div^k(\Gamma) \arrow[r, "\sim"] \arrow[dr, "\mu"] & Pic^k(\Gamma) \arrow[d, "\phi"] \\
& J(\Gamma)
\end{tikzcd}
$$

that this is well-defined on equivalence classes. It is a fact ([BN06], [MZ08]) that there is a unique q-reduced representative in each linear equivalence class of divisors. A *q-reduced divisor* f is one such that, for every closed, connected subset $B \subseteq \Gamma - \{q\}$, there is some $p \in B$ such that $0 \leq f(p) < outdeg_B(p)$, and f is effective outside of q. If we fix a model $G = (V, E)$ for Γ, with q a vertex of G, and replace the subsets B with $U \subseteq V(G) - \{q\}$, a q-reduced divisor is a G-parking function. It is standard to set $f(q) = -1$. Returning to the language of chip-firing games and critical configurations, a q-reduced divisor is stable - we cannot chip-fire from any vertex without losing effectiveness outside of q.

The Jacobian $J(C)$ of a complex algebraic curve C of genus g is an object which is constructed by integrating abelian differentials $H^0(C, \Omega^1)$ over 1-cycles $H_1(C, \mathbb{Z})$. We have the coordinate free definition $J(C) = H^0(C, \Omega^1)^*/H_1(C, \mathbb{Z})$. One can also choose a canonical basis $\{\gamma_1, ..., \gamma_{2g}\}$ of $H_1(C, \mathbb{Z})$ and with respect to this a normalized basis $\{\omega_1, ...\omega_g\}$ of $H^0(C, \mathbb{Z})$. Then vectors $(\int_{\gamma_i} \omega_1, ..., \int_{\gamma_i} \omega_g)^T$ generate a lattice Λ in $\mathbb{C}^g$, and the Jacobian is then defined as $J(C) = \mathbb{C}^g/\Lambda$, a complex torus.

The curve C is mapped to $J(C)$ by fixing a base point p_0 and sending every $p \in C$ to the point $(\int_{p_0}^{p} \omega_1, ..., \int_{p_0}^{p} \omega_g)$ modulo Λ in $J(C)$. Then, we can naturally extend the map to the group of divisors of degree 0 on C, yielding $D = \sum p_i - \sum q_i \mapsto (\sum \int_{p_i}^{q_i} \omega_1, ..., \int_{p_i}^{q_i} \omega_g)$. Via the divisor-line bundle correspondence, we get the induced map $Pic^0(C) \to J(C)$. The Abel-Jacobi Theorem states that this map is an isomorphism. A great reference for these results is Griffiths-Harris' *Principles of Algebraic Geometry* [GH78].

Chapter 6

Appendix B: The Chromatic Polynomial

A corollary of the Theorem 2.5.1 in section 2.6 introduces more general $T-G$ invariants:

Corollary 6.1: (cite Oxley and Welsh) *Let F be a field and σ, τ be non-zero elements of G. Then there is a unique function T' from $\mathcal{G}$ into $F[x,y]$ such that*

1. If b is a bridge of G, $T'(b;x,y) = x$. If l is a loop of G, $T'(l;x,y) = y$.

2. If $e \in E(G)$ is neither a loop nor a bridge, then

$$T'(G;x,y) = \sigma T'(G-e;x,y) + \tau T'(G/e;x,y).$$

3. If e is loop or bridge, then

$$T'(G;x,y) = T'(e;x,y)T'(M-e;x,y)$$

Moreover,

$$T'(G;x,y) = \sigma^{|E|-r|E|}\tau^{r(E)} \cdot T(G;\frac{x}{\tau},\frac{y}{\sigma}).$$

The chromatic polynomial is a generalized $T-G$ invariant for $\sigma = 1$ and $\tau = -1$. This polynomial counts the number of proper vertex k-colorings of G. It was this polynomial that

Tutte originally had in mind when he defined his namesake polynomial.

Definition 6.2: Let G be a graph and $k \geq 1$. A **proper vertex k-coloring** of G is a function $\kappa : V(G) \to \{1, \ldots, k\}$ such that if $(a, b) \in E(G)$, then $\kappa(a) \neq \kappa(b)$. A graph for which k-coloring exists is called k-colorable.

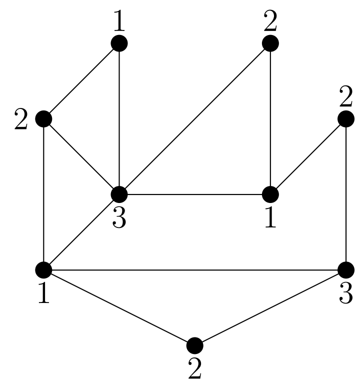


Figure 6.1: A proper vertex 3-coloring.

Definition 6.3: Let $\chi_G(k)$ be the number of k-colorings of a graph $G = (V, E)$. The number is a polynomial in k, and is called the **chromatic polynomial** of G.

If the operation of contraction is slightly changed so that any multi-edges created are deleted (as these do not affect the rules for a k-coloring), then the chromatic polynomial satisfies the simple recursion

$$\chi_G(k) = \chi_{G-e}(k) - \chi_{G/e}(k)$$

If G has m connected components, then

$$\chi_G(k) = k^m(-1)^{|V(G)|-m}T(G; 1-k, 0)$$

Recall that $\chi_G(k)$ also satisfies the recursion formula

$$\chi_G(k) = \chi_{G-e}(k) - \chi_{G/e}(k).$$

See Figure 6.2 for an example.

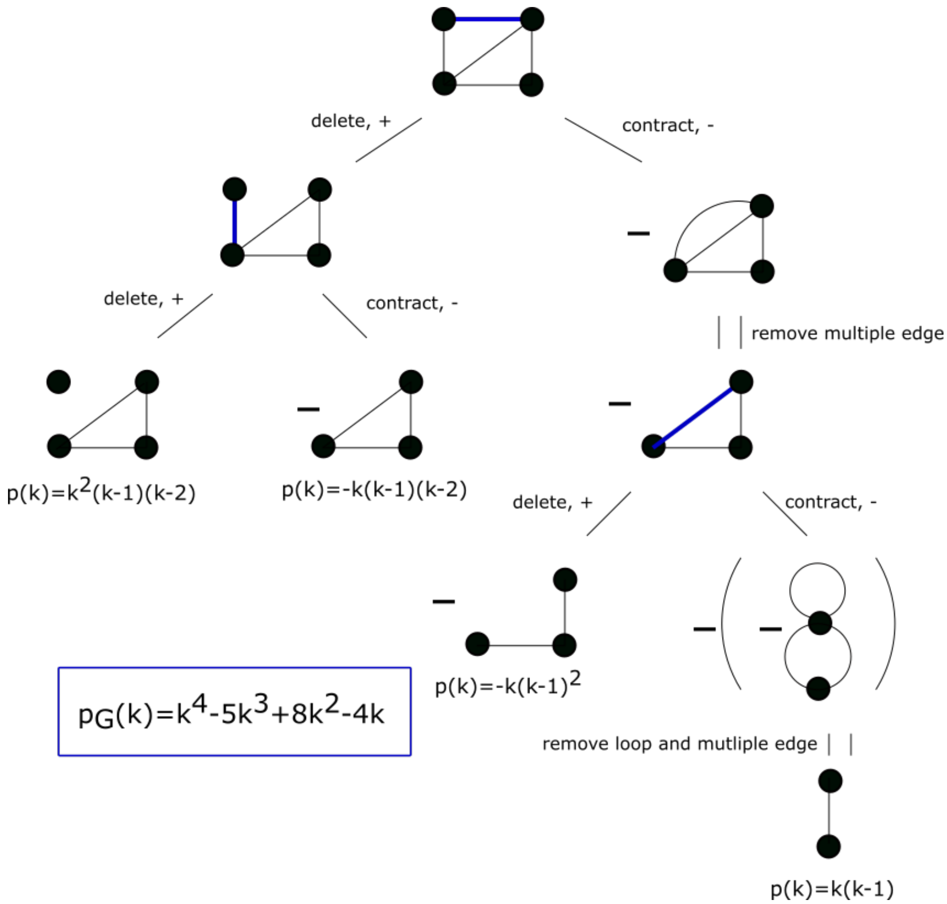


Figure 6.2: Computation of $p_G(k)$ for the graph shown. Notice that we delete loops and any multiple edges at each step. We pair deletion with a $+$ and contraction with a $-$.