THE COMPLEXITY OF GREEDOID TUTTE POLYNOMIALS

A thesis submitted for the degree of Doctor of Philosophy

by

Christopher N. Knapp

Department of Mathematics, Brunel University London

ABSTRACT

We consider the computational complexity of evaluating the Tutte polynomial of three particular classes of greedoid, namely rooted graphs, rooted digraphs and binary greedoids. Furthermore we construct polynomial-time algorithms to evaluate the Tutte polynomial of these classes of greedoid when they're of bounded tree-width. We also construct a Möbius function formulation for the characteristic polynomial of a rooted graph and determine the computational complexity of computing the coefficients of the Tutte polynomial of a rooted graph.

ACKNOWLEDGEMENTS

I would like to express my most sincere gratitude to my supervisor, Steve Noble. This thesis is a reflection of his expertise, innovation and progressive ideas. Not only this, but it also reflects the many hours he spent tirelessly guiding me with the upmost patience. His charisma and sense of humour both contributed to making our meetings so engaging, and I will be forever indebted to him for having confidence in me even when I didn't have it in myself.

I would also like to thank my second supervisor, Rhiannon Hall, for her much-needed support and constructive criticism, and the administrative staff in the department, in particular Frances Foster and Linda Campbell, for going above and beyond to answer all of my daily queries.

I dedicate this thesis to my two biggest supporters - my mum and my nana, both of whom I sadly lost in 2016.

Contents

Chapter 1

Introduction

1.1 Introduction

This thesis comprises four chapters. Chapter 1 contains the foundations for the thesis. Two combinatorial structures, namely matroids and their generalization greedoids, are discussed and relationships between particular classes of them are explored. We discuss the fundamental ideas of computational complexity theory and delve into the wide range of results surrounding the renowned Tutte polynomial.

Evaluating the Tutte polynomial of a graph at most fixed rational points is shown to be #P-hard in [31]. The k-thickening operation plays a significant role in the proof of this result. We begin Chapter 2 by generalizing the k-thickening operation to greedoids. We introduce two more greedoid constructions and give expressions for the Tutte polynomials resulting from these constructions. We then use these to find analogous results to those in [31] by completely determining the computational complexity of evaluating the Tutte polynomial of a rooted graph, a rooted digraph and of a binary greedoid at a fixed rational point. For the rooted graph case we also strengthen our results by restricting the problem to cubic, planar, bipartite rooted graphs.

Many computational problems that are intractable for arbitrary graphs become easy for the class of graphs of bounded tree-width. In [42] Noble illustrates this result by constructing a polynomial-time algorithm to evaluate the Tutte polynomial of a graph of bounded tree-width. In Chapter 3 we construct polynomial-time algorithms to evaluate the Tutte polynomials of the classes considered in the previous chapter when each is of bounded tree-width.

The characteristic polynomial of a matroid is a specialization of the Tutte polynomial. In [64] Zaslavsky gives an expression for the characteristic polynomial of a matroid in terms of the Möbius function. Gordon and McMahon generalize the characteristic polynomial to greedoids in [22] and show that several of the matroidal results have direct greedoid analogues. In Chapter 4 we begin by constructing a Möbius function formulation for the characteristic polynomial of a rooted graph. In [2] Annan determines the computational complexity of computing the coefficients of the Tutte polynomial of a graph. He shows that for fixed non-negative integers i and j, computing the coefficient of $x^i y^j$ in the Tutte polynomial of a graph is #P-hard. We continue Chapter 4 by finding analogous results for the coefficients of the Tutte polynomial of a rooted graph. We then study the smallest integer i such that the coefficient of x^i in the Tutte polynomial expansion of a rooted graph is non-zero. In [34] Kook et al present a convolution formula for the Tutte polynomial of a matroid. A natural question would be to ask if we could extend this convolution formula to greedoids. We conclude Chapter 4 by conjecturing that if the conditions hold in the proof of this convolution formula for an interval greedoid then that interval greedoid is a matroid.

1.2 Graphs and Matroids

We assume familiarity with the basic ideas of graph theory. For a more in-depth introduction to graphs, the reader is referred to [14] or [24], for example, from where the following definitions and basic results are taken.

A graph G is a pair of sets $(V(G), E(G))$ where the elements of $V(G)$ and $E(G)$ are vertices and edges of G respectively. We can omit the argument when there is no fear of ambiguity. For the following graph definitions assume $G = (V, E)$ and let $A \subseteq E$ and $S \subseteq V$. Two vertices of a graph are *adjacent* if there exists an edge between them. Similarly two edges are *adjacent* if they share a common endpoint. An edge is *incident* to a vertex if the vertex is an endpoint of the edge. A set of edges are in parallel if they all share the same endpoints. The *parallel class* of an edge $e \in E$ is the maximal set of edges that are in parallel with e in G, including e itself. The *multiplicity* of e is the cardinality of its parallel class, denoted by $m(e)$. A set of pairwise non-adjacent vertices is said to be *independent.* A *subgraph* of G is a graph $G' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq E$. A subgraph G' of G is *spanning* if $V(G') = V(G)$.

The graph obtained by restricting the edges to those in A is given by $G|A = (V, A)$. The graphs obtained by deleting the sets A and S are given by $G \setminus A = (V, E - A)$ and $G \setminus S = (V - S, E - E')$ respectively such that E' is the set of edges incident to any vertex in S. If e is an edge of a graph G, then the graph G/e, the *contraction* of e from G, is obtained by removing e and identifying its endpoints into a single new vertex. If e and f are edges of G, then $G/e/f = G/f/e$, so the definition of contraction may be generalized to sets of edges without any ambiguity. Moreover, $G/e \setminus f = G \setminus f/e$, so we define a *minor* of G to be any graph obtained from G by deleting and contracting edges. The order of the deletions and contractions does not matter.

We say G is *connected* if there is a path between any two vertices in G. A *component* of a graph is a maximal connected subgraph. We denote the number of components of $G|A$ by $\kappa_G(A)$ and let $\kappa(G) = \kappa_G(E(G))$. The *rank* of A, denoted by $r_G(A)$, is given by $r_G(A) = |V(G)| - \kappa_G(A)$. We let $r(G) = r_G(E(G))$. We can omit the subscripts in $\kappa_G(A)$ and $r_G(A)$ when the context is clear.

A graph is *planar* if it can be embedded in the plane without any of its edges crossing, and it is a *plane graph* if it is embedded in the plane in such a way. Every plane connected graph G has a *dual graph* G^* formed by assigning a vertex of G^* to each face of G, and j edges between vertices in G^* whenever the corresponding faces of G share j edges in their boundaries. For a plane non-connected graph G we define the dual G^* to be the union of the duals of each connected component. Clearly duality is an involution, that is $(G^*)^* = G$ for all G. If G is plane and $A \subseteq E(G)$ we have $r_{G^*}(A) = |A| - r(G) + r_G(E - A)$. A more generalized variation of this result is proven in [62].

A *loop* of a graph is an edge whose endpoints are the same vertex, and a *coloop*, otherwise known as a bridge or an isthmus, is an edge whose deletion would increase the number of components. Thus a coloop is an edge that is not contained in any cycle. Note that contracting or deleting a loop gives the same graph. We say G is *simple* if it contains no loops and $m(e) = 1$ for all $e \in E(G)$. We do not restrict ourselves to simple graphs unless otherwise stated.

If G contains no cycles then it is a *forest*, and if in addition it is connected then it is a *tree*. The *leaves* of a graph are the vertices with degree 1. A *leaf edge* is an edge incident to a leaf. If a subgraph of G is a tree then it is called a *subtree* of G. A loop is contained in no spanning tree, whereas a coloop is contained in every spanning tree.

An *orientation* of a graph G is an assignment of a direction to every edge in G. We say that an orientation of G is *acyclic* if it contains no directed cycles, and *totally cyclic* if every edge is contained in a directed cycle. A vertex v is a *source* (respectively a *sink*) if all incident edges are directed away from (respectively towards) v.

An *isomorphism* of graphs G and G' is a bijection $f : V(G) \to V(G')$ between the vertex sets of G and G', such that any pair of vertices u and v are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in G'. If an isomorphism exists between two graphs G and G' then they are said to be *isomorphic* to one another and we write $G \cong G'$.

A matroid is a structure that generalizes linear independence in vector spaces. Matroids were first introduced by Whitney in 1935 in his founding paper [62]. Many connections between matroids and other fields have been found in the ensuing decades, with notable contributors Tutte, Rota and Welsh, to name a few. Edmonds later used them to characterize the class of optimization problems that can be solved using greedy algorithms [16]. There are many equivalent definitions of a matroid, for a full collection along with proofs of their equivalence see [43]. We now present the definition in terms of independent sets.

Definition 1.2.1 (Matroid - Independent Sets)**.** A *matroid* is an ordered pair $(E, \mathcal{I})$ consisting of a finite set E and a non-empty collection $\mathcal{I}$ of subsets of E satisfying the following three axioms:

(M1) $\emptyset \in \mathcal{I}$;

(M2) If $I \in \mathcal{I}$ and $I' \subseteq I$, then $I' \in \mathcal{I}$;

(M3) If $I, I' \in \mathcal{I}$ and $|I'| < |I|$, then there exists some element $x \in I - I'$ such that $I' \cup x \in \mathcal{I}$.

Axiom $(M3)$ is often referred to as the *augmentation axiom.* If M is the matroid $(E, \mathcal{I})$, then E is the *ground set* of M and the members of $\mathcal{I}$ are the *independent sets* of M. Two matroids $M_1 = (E_1, \mathcal{I}_1)$ and $M_2 = (E_2, \mathcal{I}_2)$ are *isomorphic*, denoted by $M_1 \cong M_2$, if there exists a bijection $f : E_1 \to E_2$ that preserves the independent sets.

Every graph G has an associated matroid $(E, \mathcal{I})$ called the *cycle matroid* of G, where $E = E(G)$ and a subset of edges A is in $\mathcal{I}$ if and only if $G|A$ is a forest. Any matroid which is isomorphic to a cycle matroid is called *graphic*. It is easy to see that all matroids on three elements are graphic. We let $M(G)$ denote the cycle matroid of a graph G.

Let N be an $m \times n$ matrix over some field $\mathbb{F}$ and let v_i be the i-th column of N, viewed as a vector in $\mathbb{F}^m$. Let $E = \{1, 2, \ldots, n\}$ and let $\mathcal{I}$ be the set containing all subsets A of E such that $\{v_i : i \in A\}$ is linearly independent. Then $(E, \mathcal{I})$ is a matroid called the *vector matroid* of N. We let $M(N)$ denote the vector matroid of a matrix N.

A *basis* of a matroid is a maximal independent set. Axiom (M3) implies that every basis of a matroid has the same cardinality. An element $x \in E$ is a *loop* in M if it is not contained in any independent set of M, and is a *coloop* in M if it is contained in every basis of M.

We can extend the notion of the rank function from graphs to matroids as follows. Given a matroid $M = (E, \mathcal{I})$ and a subset A of E, let $r_M(A) = \max\{|A'| : A' \subseteq A, A' \in \mathcal{I}\}$. We can omit the subscript in $r_M(A)$ when the context is clear. Let $r(M) = r_M(E)$ and consequently $r(M)$ is the size of any basis of M. The set of bases and the rank function both determine the matroid, for the set A is independent if and only if it is contained in some basis, or if and only if it satisfies $r(A) = |A|$.

We can generalize the deletion and contraction operations to matroids. These are best defined using the rank function, so we need the following characterization.

Proposition 1.2.2. *The rank function r of a matroid M with ground set E takes integer values and satisfies each of the following:*

(MR1) For any subset A of E, $0 \leq r(A) \leq |A|$;

(MR2) For any subsets A and B of E with $A \subseteq B$, $r(A) \leq r(B)$;

(MR3) For any subsets A and B of E, $r(A) + r(B) \geq r(A \cap B) + r(A \cup B)$.

Moreover if E is a finite set and r is a function from the subsets of E to the integers, then r is the rank function of a matroid with ground set E if and only if r satisfies conditions (MR1)–(MR3) above.

Let $M = (E, r)$ be a matroid specified by its rank function and let A be a subset of E. Then the *deletion* of A from M, denoted by $M \setminus A$, has ground set $E - A$ and rank function $r_{M \setminus A}$, where $r_{M \setminus A}(X) = r_M(X)$ for all subsets X of $E - A$. The *contraction* of A from M, denoted by M/A, has ground set $E - A$ and rank function $r_{M/A}$, where $r_{M/A}(X) = r_M(X \cup A) - r_M(A)$ for all subsets X of $E - A$. It is straightforward to check that both the deletion and contraction are indeed matroids. Moreover if A and B are disjoint subsets of E, then $M/A/B = M/B/A = M/(A \cup B)$, $M \setminus A \setminus B = M \setminus B \setminus A = M \setminus (A \cup B)$ and $M/A \setminus B = M \setminus B/A$. Furthermore if e is either a loop or a coloop, then $M/e = M \setminus e$.

It follows from the definition that if I is a subset of $E - A$, then it is independent in M if and only if it is independent in $M \setminus A$. To determine the independent sets of M/A requires a little more work. Let I' be a maximal independent subset of A. Then $r_{M/I'}(A - I') = r_M(A) - r_M(I') = 0$. Thus every element of $A - I'$ is a loop in M/I' and $M/A = M/I' \setminus (A - I')$. Therefore if I is a subset of $E - A$, then it is independent in M/A if and only if $|I| = r_{M/I'}(I) = r_M(I \cup I') - r_M(I') = r_M(I \cup I') - |I'|$, which is equivalent to $I \cup I'$ being independent in M.

For a matroid $M = (E, \mathcal{I})$, we define the *dual matroid* M^* of M to be the matroid with ground set E and whose bases are precisely the complements of the bases of M. Obviously we have the property $(M^*)^* = M$. For $A \subseteq E$ we have $r_{M^*}(A) = |A| - r(M) + r_M(E - A)$, a result proved in [62].

1.3 Rooted Graphs, Rooted Digraphs and Greedoids

A *rooted graph* is a graph with a fixed "special" vertex called the *root*. Many of the definitions for graphs can be applied to rooted graphs in the natural way. We let $G = (r, V(G), E(G))$ denote a rooted graph with vertex set $V(G)$, edge set $E(G)$ and root vertex $r \in V(G)$. We can omit the argument when there is no fear of ambiguity.

We say (r', V', E') is a *rooted subgraph* of (r, V, E) if (V', E') is a subgraph of (V, E) and $r' = r$. We now define analogues of trees and subtrees. These definitions will differ from standard practice in that we will allow trees to have one or more isolated vertices. This turns out to significantly simplify the explanation later on. A graph (r, V, E) is a *rooted tree* if $r \in V$ and (V, E) is a forest such that the components not containing r are isolated vertices. A graph (r', V', E') is a *rooted spanning subtree* of the graph (r, V, E) if it is a rooted tree and $V' = V$ and $r' = r$.

For $A \subseteq E$ the definitions of $G|A$ and $G \setminus A$ are analogous to those for a graph. To obtain G/A, we contract the edges of a largest subtree of (r, V, A) and delete the remaining edges from A. This is equivalent to deleting all the edges of A and replacing the vertices of G which may be reached from r along paths comprising edges from A with a single vertex. Therefore it does not depend on the choice of largest subtree. The root vertex is unchanged by these operations except when an edge incident to the root is contracted, in which case the new vertex created becomes the root. An edge set $A \subseteq E$ of a rooted graph $G = (r, V, E)$ is *feasible* if and only if $G|A$ is a rooted subtree. The rank function of a subset of edges of a rooted graph is dependent on the choice of root vertex. We define the *rank* of A to be

$$\rho_G(A) = \max\{|A'| : A' \subseteq A, G|A' \text{ is a rooted subtree}\}.$$

We can omit the subscript when the context is clear. We let $\rho(G) = \rho_G(E)$. A set of edges A is feasible if and only if $\rho(A) = |A|$ and, if in addition $\rho(A) = \rho(G)$, A is said to be a *basis* of G.

A *rooted digraph* is a rooted graph in which every edge has a fixed direction from one endpoint to the other. Again many of the definitions for graphs can be applied to rooted digraphs in the natural way. We let $D = (r, V(D), \overrightarrow{E}(D))$ denote a rooted digraph with vertex set $V(D)$, directed edge set $\overrightarrow{E}(D)$ and root vertex $r \in V(D)$. Once again we can omit the argument when there is no chance of ambiguity.

We say that the underlying rooted graph of a rooted digraph is what we get when we remove all orientations on the edges. A rooted digraph D is called a *rooted arborescence* if the underlying rooted graph of D is acyclic and for every vertex $v \in V(D)$ there is either a directed path from the root to v or v is an isolated vertex. In other words a rooted arborescence is a directed rooted tree in which all of the edges are directed away from the root. For $A \subseteq \overrightarrow{E}$ the definitions of $D \setminus A$ and $D|A$ are analogous to those for a graph. To obtain D/A, we contract the edges of a largest subdigraph of (r, V, A) that is a rooted arborescence and delete the remaining edges from A. This is equivalent to deleting all the edges of A and replacing the vertices of D which may be reached by paths directed away from r comprising edges from A with a single vertex. Therefore it does not depend on the choice of the largest arborescence. An edge set $A \subseteq \overrightarrow{E}$ of a rooted digraph $D = (r, V, \overrightarrow{E})$ is *feasible* if and only if $D|A$ is a rooted arborescence.

The *rank* of A is defined by

$$\rho_D(A) = \max\{|A'| : A' \subseteq A, D|A' \text{ is a rooted arborescence}\}.$$

We can omit the subscript when the context is clear. We let $\rho(D) = \rho_D(\overrightarrow{E})$. A set of edges A is feasible if and only if $\rho(A) = |A|$ and, if in addition $\rho(A) = \rho(D)$, A is said to be a *basis* of D.

A greedoid is a generalization of a matroid, first introduced by Korte and Lovász in 1981 in [35]. The initial purpose was to generalize Edmond's class of optimization problems that are solvable by greedy algorithms.

Definition 1.3.1 (Greedoid)**.** A *greedoid* Γ is an ordered pair $(E, \mathcal{F})$ consisting of a finite set E and a non-empty collection $\mathcal{F}$ of subsets of E satisfying the following axioms:

(G1) $\emptyset \in \mathcal{F}$.

(G2) For all $F, F' \in \mathcal{F}$ with $|F'| < |F|$ there exists some $x \in F - F'$ such that $F' \cup x \in \mathcal{F}$.

The members of $\mathcal{F}$ are the *feasible sets* of Γ, and E is again referred to as the *ground set* of Γ. We now prove that we still define a greedoid if we replace axiom (G1) with

(G1$'$) For all $F \in \mathcal{F}$ with $F \neq \emptyset$, there exists some $x \in F$ such that $F - x \in \mathcal{F}$.

Proposition 1.3.2. *Let $\Gamma = (E, \mathcal{F})$. Then Γ is a greedoid if and only if it satisfies (G1$'$) and (G2).*

Proof. First suppose $\Gamma = (E, \mathcal{F})$ is a greedoid and let $F \in \mathcal{F}$ with $|F| = k > 0$. We claim there exist feasible sets $F_0, F_1, \ldots, F_k$ with $|F_i| = i$ for all i such that

$$F_0 = \emptyset \subseteq F_1 \subseteq F_2 \subseteq \ldots \subseteq F_{k-1} \subseteq F_k = F.$$

Clearly F_0 exists by (G1). Suppose $F_0, \ldots, F_i$ exist for $0 \leq i \leq k-1$. By (G2) there exists $x_{i+1} \in F - F_i$ such that $F_{i+1} = F_i \cup x_{i+1} \in \mathcal{F}$. Hence by induction the claim is true.

Let x be the unique element in $F - F_{k-1}$. Then $F - x = F_{k-1} \in \mathcal{F}$. Therefore (G1$'$) holds.

Now suppose $\Gamma = (E, \mathcal{F})$ satisfies (G1$'$) and (G2). Showing that (G1) must also be true is straightforward because (G1$'$) implies we can delete an element of a feasible set to obtain a feasible set whose cardinality is one smaller. We can continue to do this until we get a feasible set with no elements, i.e. $\emptyset \in \mathcal{F}$. This completes the proof. $\square$

Two greedoids $\Gamma_1 = (E_1, \mathcal{F}_1)$ and $\Gamma_2 = (E_2, \mathcal{F}_2)$ are isomorphic, denoted by $\Gamma_1 \cong \Gamma_2$, if there exists a bijection $f : E_1 \rightarrow E_2$ that preserves the feasible sets.

Let G be a rooted graph. Suppose we have $\Gamma = (E, \mathcal{F})$ such that $E = E(G)$ and a subset A of E is in $\mathcal{F}$ if and only if $G|A$ is a rooted subtree. We claim that Γ is a greedoid. When $A = \emptyset$, $G|A$ is the rooted subtree comprising entirely isolated vertices, hence $\emptyset \in \mathcal{F}$. Let A and B be subsets of E such that both $G|A$ and $G|B$ are rooted subtrees of G and $|B| > |A|$. Then there is some vertex x to which there is a path P from r in $G|B$ but not in $G|A$. The path P must include an edge e joining a vertex connected to r in $G|A$ to a vertex not connected to r in $G|A$. Then $e \in B - A$ and $G|(A \cup e)$ is a rooted subtree of G. Therefore our claim is true. Any greedoid which is isomorphic to a greedoid arising from a rooted graph in this way is called a *branching greedoid*. The branching greedoid of a rooted graph G is denoted by $\Gamma(G)$.

Similarly suppose we have a rooted digraph D and let $\Gamma = (E, \mathcal{F})$ such that $E = \overrightarrow{E}(D)$ and a subset A of $\overrightarrow{E}$ is in $\mathcal{F}$ if and only if $D|A$ is a rooted arborescence. Then Γ is a greedoid. The proof of this is analogous to that mentioned above. Any greedoid which is isomorphic to a greedoid arising from a rooted digraph in this way is called a *directed branching greedoid*. They were originally called "search greedoids" in [35], where they were introduced. The directed branching greedoid of a rooted digraph D is denoted by $\Gamma(D)$.

Let $\Gamma = (E, \mathcal{F})$ be a greedoid. We now generalize the rank function from rooted graphs and rooted digraphs to greedoids. The *rank* of $A \subseteq E$ is given by

$$\rho_\Gamma(A) = \max\{|A'| : A' \subseteq A, A' \in \mathcal{F}\}.$$

We can omit the subscript when the context is clear. Let $\rho(\Gamma) = \rho_\Gamma(E)$. A set A is *feasible* if and only if $\rho(A) = |A|$, and a basis if in addition $\rho(A) = \rho(\Gamma)$. Axiom $(G2)$ implies that every basis has the same cardinality. Note that the rank function determines Γ but the collection of bases does not. For example suppose a greedoid $\Gamma = (E, \mathcal{F})$ has $E = \{1, 2\}$ and unique basis $\{1, 2\}$. Then $\mathcal{F}$ could either be $\{\emptyset, \{1\}, \{1, 2\}\}$, $\{\emptyset, \{2\}, \{1, 2\}\}$ or $\{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$. Note that the greedoids defined by the former two sets are isomorphic to one another.

The rank function of a greedoid can be characterized in a similar way to the rank function of a matroid [36].

Proposition 1.3.3. *The rank function ρ of a greedoid G with ground set E takes integer values and satisfies each of the following.*

(GR1) For any subset A of E, $0 \leq \rho(A) \leq |A|$;

(GR2) For any subsets A and B of E with $A \subseteq B$, $\rho(A) \leq \rho(B)$;

(GR3) For any subset A of E, and elements e and f, if $\rho(A) = \rho(A \cup e) = \rho(A \cup f)$, then $\rho(A) = \rho(A \cup e \cup f)$.

Moreover if E is a finite set and ρ is a function from the subsets of E to the integers, then ρ is the rank function of a greedoid with ground set E if and only if ρ satisfies conditions (GR1)–(GR3) above.

The following lemma will be useful both in the proof of this result and later.

Lemma 1.3.4. *Let E be a finite set and ρ be a function from the subsets of E to the integers satisfying (GR1)–(GR3). Let A and B be subsets of E such that for all $b \in B$, $\rho(A \cup b) = \rho(A)$. Then $\rho(A \cup B) = \rho(A)$.*

Proof. We prove the result by induction on $|B|$. If $|B| \leq 1$ then the result is immediate. Suppose then that $|B| \geq 2$. Let b_1, b_2 be distinct elements of B. Then by induction $\rho(A \cup (B - b_1 - b_2)) = \rho(A \cup (B - b_1)) = \rho(A \cup (B - b_2))$. So (GR3) implies that $\rho(A \cup B) = \rho(A \cup (B - b_1 - b_2)) = \rho(A)$, as required. □

Proof of Proposition. Suppose that $\Gamma = (E, \mathcal{F})$ is a greedoid and $\rho(A) = \max\{|A'| : A' \subseteq A, A' \in \mathcal{F}\}$. Properties (GR1) and (GR2) follow immediately from the definition of ρ. Now suppose that $\rho(A) = \rho(A \cup e) = \rho(A \cup f)$. Assume for contradiction that $\rho(A) \neq \rho(A \cup e \cup f)$. Then from (GR2) it follows that $\rho(A) < \rho(A \cup e \cup f)$. Let F_1 and F_2 be the largest feasible subsets of A and $A \cup e \cup f$ respectively. Then $|F_2| > |F_1|$ so there exists $x \in F_2 - F_1$ such that $F_1 \cup x$ is feasible. The element x cannot come from A without contradicting the definition of F_1 so it must be either e or f. But this contradicts $\rho(A) = \rho(A \cup e) = \rho(A \cup f)$. Thus $\rho(A) = \rho(A \cup e) = \rho(A \cup f) = \rho(A \cup e \cup f)$.

Now suppose that ρ is an integer valued function defined on the subsets of E. Let $\mathcal{F} = \{A : \rho(A) = |A|\}$. We shall show that $(E, \mathcal{F})$ is a greedoid. From (GR1) we have $\rho(\emptyset) = 0$, so $\emptyset \in \mathcal{F}$ and (G1) is satisfied. Let X and Y satisfy $\rho(X) = |X|$, $\rho(Y) = |Y|$ and $|Y| > |X|$. If $e \in Y - X$ and $\rho(X \cup e) > \rho(X)$ then it follows from (GR1) that $\rho(X \cup e) = |X \cup e|$ and (G2) is satisfied. Suppose then, for contradiction, that for all $e \in Y - X$, $\rho(X \cup e) \leq \rho(X)$ and therefore by (GR2) $\rho(X \cup e) = \rho(X)$. It follows from Lemma 1.3.4 that $\rho(X \cup Y) = \rho(X \cup (Y - X)) = \rho(X) < \rho(Y)$ which contradicts (GR2). □

An element of a greedoid is a *loop* if it does not belong to any feasible set. Thus if $G = (r, V, E)$ is a rooted graph then an edge e is a loop of $\Gamma(G)$ if it does not lie on any path from r. Similarly if $D = (r, V, E)$ is a directed rooted graph then an edge e is a loop of $\Gamma(D)$ if it does not lie on any directed path from r.

Let $\Gamma = (E, \mathcal{F})$ be a greedoid and $e, f \in E$. Then e and f are said to be *parallel* in Γ if for all $A \subseteq E$ we have

$$\rho(A \cup e) = \rho(A \cup f) = \rho(A \cup e \cup f).$$

Lemma 1.3.5. *Any two loops in a greedoid are parallel.*

Proof. Let $\Gamma = (E, \mathcal{F})$ be a greedoid and let $l_1, l_2 \in E$ be loops. Then, because neither l_1 nor l_2 belong to any feasible set, for all $A \subseteq E$ we have

$$\rho(A \cup l_1) = \rho(A \cup l_2) = \rho(A \cup l_1 \cup l_2) = \rho(A).$$

Therefore l_1 and l_2 are parallel in Γ. $\square$

Lemma 1.3.6. *No loop is parallel with a non-loop in a greedoid.*

Proof. Let $\Gamma = (E, \mathcal{F})$ be a greedoid and let l be a loop and e be a non-loop in Γ. Repeatedly applying (G1′), shows that there are feasible sets F' and F'' with $F'' - F' = \{e\}$. Now $\rho(F' \cup e) = \rho(F'')$, whereas $\rho(F' \cup l) = \rho(F'') - 1$ since l does not belong to any feasible set. Therefore e and l cannot be parallel in Γ. $\square$

We now show that being parallel in a greedoid is an equivalence relation. If e and f are parallel in a greedoid we write $e \bowtie f$.

Lemma 1.3.7. $\bowtie$ *is an equivalence relation.*

Proof. Let $\Gamma = (E, \mathcal{F})$ be a greedoid. Proving $\bowtie$ is reflexive and symmetric is easy. Now suppose we have $e, f, g \in E$ such that $e \bowtie f$ and $f \bowtie g$. Then for all $A \subseteq E$ we have

$$\rho(A \cup e) = \rho(A \cup f) = \rho(A \cup e \cup f)$$

and

$$\rho(A \cup f) = \rho(A \cup g) = \rho(A \cup f \cup g).$$

Therefore $\rho(A \cup e) = \rho(A \cup f) = \rho(A \cup g)$. Applying (GR3) to the set $A \cup f$ and elements e and g, we see that $\rho(A \cup e \cup f \cup g) = \rho(A \cup f)$. Now by (GR2),

$$\rho(A \cup e) = \rho(A \cup f) = \rho(A \cup e \cup f \cup g) \geq \rho(A \cup e \cup g) \geq \rho(A \cup e).$$

So equality must hold throughout and $\rho(A \cup e \cup g) = \rho(A \cup e) = \rho(A \cup g)$. Hence $e \bowtie g$ and $\bowtie$ is an equivalence relation. $\square$

Let $\Gamma = (E, \mathcal{F})$ be a greedoid and $e \in E$. The *parallel class* of e is the equivalence class of the equivalence relation $\bowtie$ on Γ that contains e. The *multiplicity* of e is the cardinality of its parallel class, denoted by $m(e)$.

The deletion and contraction constructions generalize to greedoids in the following way. Let $\Gamma = (E, \mathcal{F})$ be a greedoid and $A \subseteq E$. Define the *deletion* of A from Γ by $\Gamma \setminus A = (E - A, \mathcal{F}_1)$ where $\mathcal{F}_1 = \{X \subseteq E - A : X \in \mathcal{F}\}$, and when A is feasible we define the *contraction* of A from Γ

by $\Gamma/A = (E - A, \mathcal{F}_2)$ where $\mathcal{F}_2 = \{X \subseteq E - A : X \cup A \in \mathcal{F}\}$. We now show that $\Gamma \setminus A$ and Γ/A are in fact both greedoids.

Lemma 1.3.8. *Let $\Gamma = (E, \mathcal{F})$ be a greedoid. Then*

1. *$\Gamma \setminus A$ is a greedoid for all $A \subseteq E$.*

2. *Γ/A is a greedoid for all feasible $A \subseteq E$.*

Proof. 1. Let $\Gamma \setminus A = (E - A, \mathcal{F}_1)$ where $\mathcal{F}_1 = \{X \subseteq E - A : X \in \mathcal{F}\}$ and $A \subseteq E$. It is straightforward to see that $\emptyset \in \mathcal{F}_1$ since it is in $\mathcal{F}$. The sets in $\mathcal{F}_1$ are in one-to-one correspondence with those in $\mathcal{F}$ that do not contain any elements of A. That is $F \in \mathcal{F}_1$ if and only if $F \in \mathcal{F}$ and $F \cap A = \emptyset$. Suppose we have $F, F' \in \mathcal{F}_1$ such that $|F'| > |F|$. Then $F, F' \in \mathcal{F}$ and $F \cap A = F' \cap A = \emptyset$. By (G2) there exists $x \in F' - F$ such that $F \cup x \in \mathcal{F}$. Now $(F \cup x) \cap A = \emptyset$ so $F \cup x \in \mathcal{F}_1$. Hence $\Gamma \setminus A$ is a greedoid.

2. Let $\Gamma/A = (E - A, \mathcal{F}_2)$ where $\mathcal{F}_2 = \{X \subseteq E - A : X \cup A \in \mathcal{F}\}$ and $A \subseteq E$ is feasible. It is straightforward to see that $\emptyset \in \mathcal{F}_2$ since $\emptyset \cup A = A \in \mathcal{F}$. The sets in $\mathcal{F}_2$ are in one-to-one correspondence with those in $\mathcal{F}$ that contain A. That is $F \in \mathcal{F}_2$ if and only if $F \cup A \in \mathcal{F}$ and $F \cap A = \emptyset$. Suppose we have $F, F' \in \mathcal{F}_2$ such that $|F'| > |F|$. Then $F \cup A, F' \cup A \in \mathcal{F}$ and $F \cap A = F' \cap A = \emptyset$. By (G2) there exists $x \in (F' \cup A) - (F \cup A) = F' - F$ such that $F \cup A \cup x \in \mathcal{F}$. Therefore $F \cup x \in \mathcal{F}_2$. Hence Γ/A is a greedoid.

□

Note that Γ/A is a greedoid if and only if A is feasible, otherwise $\emptyset \notin \mathcal{F}_2$. Deletion and contraction on a greedoid are commutative. That is, for A feasible in Γ and $\Gamma \setminus B$ we have

$$\mathcal{F}((\Gamma/A) \setminus B) = \mathcal{F}((\Gamma \setminus B)/A).$$

Similarly if A and B are disjoint subsets of E, then $(\Gamma \setminus A) \setminus B = (\Gamma \setminus B) \setminus A$. If, additionally, A, B and $A \cup B$ are all feasible then $(\Gamma/A)/B = (\Gamma/B)/A$.

For all $X \subseteq E - A$, we have

$$\rho_{\Gamma \setminus A}(X) = \rho_{\Gamma}(X)$$

and providing A is feasible in Γ, for all $X \subseteq E - A$, we have

$$\rho_{\Gamma/A}(X) = \rho_\Gamma(X \cup A) - \rho_\Gamma(A). \tag{1.1}$$

We now define a large class of greedoids, characterized by the 'interval property'. An *interval greedoid* is a greedoid $\Gamma = (E, \mathcal{F})$ satisfying the *interval property*:

(IG) For $A \subseteq B \subseteq C$ with $A, B, C \in \mathcal{F}$, if there exists $x \in E-C$ such that $A \cup x \in \mathcal{F}$ and $C \cup x \in \mathcal{F}$, then $B \cup x \in \mathcal{F}$.

Every greedoid of rank less than three satisfies the interval property. An example of an interval greedoid is given by $\Gamma = (\{1, 2, 3, 4, 5\}, \mathcal{F})$ where

$$\begin{aligned}\mathcal{F} =& \{\emptyset, \{1\}, \{2\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 5\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 3, 4\}, \\ & \{1, 3, 5\}, \{1, 4, 5\}, \{2, 3, 5\}, \{2, 3, 4\}, \{2, 4, 5\}\}.\end{aligned}$$

Interval greedoids tend to behave better than general greedoids in many respects. For instance we can contract arbitrary subsets of an interval greedoid as opposed to restricting ourselves to feasible subsets. Let $\Gamma = (E, \mathcal{F})$ be an interval greedoid and $A \subseteq E$. Let X be a maximal feasible subset of A. The contraction Γ/A is obtained by contracting X and then deleting $A - X$.

We show that contraction of arbitrary sets is well-defined in interval greedoids. We require the following two lemmas, the first of which is a restatement of the interval property in terms of the rank function. All of these results are adaptations of results from [37].

Lemma 1.3.9. *A greedoid (E, ρ) is an interval greedoid if and only if whenever A, B and C are subsets of E with $A \subseteq B \subseteq C$ and x is an element of $E - C$ satisfying $\rho(A \cup x) > \rho(A)$ and $\rho(C \cup x) > \rho(C)$, we have $\rho(B \cup x) > \rho(B)$.*

Proof. The condition is clearly sufficient because the special case where A, B and C are all feasible is the interval property. Let F_A be a maximal feasible subset of A. As F_A is a feasible subset of B, it may be extended to a maximal feasible subset F_B of B. Similarly F_B may be extended to a maximal feasible set F_C of C. Because $\rho(A \cup x) > \rho(A)$, there is a feasible subset F of $A \cup x$ with $|F| > |F_A|$. Hence there is an element y of $F - F_A$ such that $F_A \cup y$ is feasible. But y cannot belong to A without contradicting the maximality of F_A. Consequently $y = x$ and $F_A \cup x$ is feasible. Using exactly the same argument $F_C \cup x$ is feasible. Hence, by the interval property, $F_B \cup x$ is feasible and $\rho(B \cup x) \geq |F_B \cup x| > |F_B| = \rho(B)$. □

Lemma 1.3.10. *Let $\Gamma = (E, \rho)$ be an interval greedoid, X and Y be subsets of E and F be a feasible set of Γ with $\rho(X \cup F) = \rho(X)$. Then $\rho(X \cup Y \cup F) = \rho(X \cup Y)$.*

Proof. Let $U = \{e \in F : \rho(X \cup Y \cup e) = \rho(X \cup Y)\}$, and let F_U be a maximal feasible subset of U. If $F_U = F$ then the result follows by applying Lemma 1.3.4 with $A = X \cup Y$ and $B = F$. So we may assume, for contradiction, that $F_U \neq F$. Thus there exists x in $F - F_U$ such that $F_U \cup x$ is feasible. We have $F_U \subseteq X \cup U \subseteq X \cup U \cup Y$ and $\rho(F_U \cup x) > \rho(F_U)$, by the definition of x. Furthermore, by the maximality of F_U, $x \notin U$, so $\rho(X \cup U \cup Y \cup x) \geq \rho(X \cup Y \cup x) > \rho(X \cup Y) = \rho(X \cup U \cup Y)$, where the last equality follows by applying Lemma 1.3.4 with $A = X \cup Y$ and $B = U$. Applying Lemma 1.3.9, we deduce that $\rho(X \cup U \cup x) > \rho(X \cup U)$. On the other hand $X \subseteq X \cup U \subseteq X \cup U \cup x \subseteq X \cup F$. Thus $\rho(X) \leq \rho(X \cup U) \leq \rho(X \cup U \cup x) \leq \rho(X \cup F)$. But we have $\rho(X) = \rho(X \cup F)$ and consequently equality holds throughout. So $\rho(X \cup U) = \rho(X \cup U \cup x)$, giving a contradiction. □

We are now ready to prove that contraction of arbitrary sets in an interval greedoid is well-defined.

Proposition 1.3.11. *Let $(E, \mathcal{F})$ be an interval greedoid, let A be a subset of E and X be a subset of $E - A$. Let F_1 and F_2 be maximal feasible subsets of A. Then $F_1 \cup X$ is feasible if and only if $F_2 \cup X$ is feasible.*

Proof. Suppose that $F_1 \cup X$ is feasible. We apply Lemma 1.3.10. We have $\rho(F_1 \cup F_2) = \rho(F_2)$. Consequently

$$\rho(F_2 \cup X) = \rho(F_1 \cup F_2 \cup X) \geq \rho(F_1 \cup X) = |F_1 \cup X| = |F_2 \cup X|,$$

with the last equality following because $|F_1| = |F_2|$ and $X \cap A = \emptyset$. Thus $F_2 \cup X$ is feasible. The converse is identical. □

We now show that $\Gamma \setminus A$ and Γ / A are in fact interval greedoids.

Lemma 1.3.12. *Let $\Gamma = (E, \mathcal{F})$ be an interval greedoid. Then*

1. *$\Gamma \setminus A$ is an interval greedoid for all $A \subseteq E$.*

2. *Γ / A is an interval greedoid for all $A \subseteq E$.*

Proof. 1. Let $\Gamma \setminus A = (E - A, \mathcal{F}_1)$ where $\mathcal{F}_1 = \{X \subseteq E - A : X \in \mathcal{F}\}$ and $A \subseteq E$. By Lemma 1.3.8 $\Gamma \setminus A$ is a greedoid. We have $F \in \mathcal{F}_1$ if and only if $F \in \mathcal{F}$ and $F \cap A = \emptyset$. Suppose we have $B, C, D \in \mathcal{F}_1$ such that $B \subseteq C \subseteq D$. Then $B, C, D \in \mathcal{F}$ and $B \cap A = C \cap A = D \cap A = \emptyset$. Suppose there exists $x \in (E - A) - D$ such that $B \cup x, D \cup x \in \mathcal{F}_1$, then $B \cup x, D \cup x \in \mathcal{F}$. By (IG) we have $C \cup x \in \mathcal{F}$. Now $(C \cup x) \cap A = \emptyset$ since $x \notin A$, so we have $C \cup x \in \mathcal{F}_1$. Hence $\Gamma \setminus A$ is an interval greedoid.

2. It suffices to prove the result when A is feasible. The general case follows by combining this with part 1. Let $\Gamma/A = (E - A, \mathcal{F}_2)$ where $\mathcal{F}_2 = \{X \subseteq E - A : X \cup A \in \mathcal{F}\}$ and $A \subseteq E$ is feasible. By Lemma 1.3.8 Γ/A is a greedoid. We have $F \in \mathcal{F}_2$ if and only if $F \cup A \in \mathcal{F}$ and $F \cap A = \emptyset$. Suppose we have $B, C, D \in \mathcal{F}_2$ such that $B \subseteq C \subseteq D$. Then $B \cup A, C \cup A, D \cup A \in \mathcal{F}$ and $B \cap A = C \cap A = D \cap A = \emptyset$. Suppose there exists $x \in (E - A) - D$ such that $B \cup x, D \cup x \in \mathcal{F}_2$, then $B \cup A \cup x, D \cup A \cup x \in \mathcal{F}$. By (IG) we have $C \cup A \cup x \in \mathcal{F}$. Now $(C \cup x) \cap A = \emptyset$ since $x \notin A$, so we have $C \cup x \in \mathcal{F}_2$. Hence Γ/A is an interval greedoid.

$\square$

We now present results on the relationship between interval greedoids and two particular classes that they specialize to.

Theorem 1.3.13. *Let $\Gamma = (E, \mathcal{F})$ be an interval greedoid with $x \in \mathcal{F}$ for all non-loop elements $x \in E$. Then Γ is a matroid.*

Proof. We want to show that the feasible sets of an interval greedoid satisfying this property are closed under taking subsets. Let $F \in \mathcal{F}$. We use induction on $|F|$. When $|F| = 1$ there exists one element $x \in F$. The subsets of F are therefore $\emptyset$ and $\{x\}$, both of which are feasible.

Assume for some integer $k \geq 1$ every subset of a feasible set F of size k is feasible. Take $F' \in \mathcal{F}$ with $|F'| = k + 1$. Then by (G1$'$) there exists an $x \in F'$ such that $F'' = F' - x$ is feasible. By induction all subsets of F'' are feasible. We now need only show that the subsets of F' containing x are feasible. Let $F'' = F' - x$ and $S \subseteq F''$. The subsets of F' containing x are the subsets $S \cup x$ for all possible S. Now by (IG) if we let $A = \emptyset$, $B = S$ and $C = F''$, then since $x \in E - F''$ and $\emptyset, x, S, F''$ and F' are feasible, so is $S \cup x$. Therefore every subset of a feasible set of size $k + 1$ is feasible. This completes the proof. $\square$

Theorem 1.3.14. *Every branching greedoid is an interval greedoid.*

Proof. Let $G = (r, V, E)$ be a rooted graph and $\Gamma = (E, \mathcal{F})$ such that $\Gamma = \Gamma(G)$. Recall that a set $E' \subseteq E$ is feasible in G if and only if $G|E'$ is a rooted subtree. Let $A, B, C \in \mathcal{F}$ such that $A \subseteq B \subseteq C$. Suppose there exists an element $x \in E - C$ such that $G|(A \cup x)$ and $G|(C \cup x)$ are rooted subtrees. Since A, B and $A \cup x$ are feasible, we know that the components of $G|(B \cup x)$ not containing the root are isolated vertices. Moreover $C \cup x$ is feasible so $G|(B \cup x)$ cannot contain any cycles. Thus $B \cup x$ is feasible. Therefore $\Gamma(G)$ is an interval greedoid. $\square$

A similar proof is required to show that all directed branching greedoids are interval greedoids.

We now define a class of greedoids called the Gaussian elimination greedoids whose structure underlies the Gaussian elimination algorithm, hence its name.

Let M be an $m \times n$ matrix over an arbitrary field. For a positive integer n, we let $[n] = \{1, 2, \ldots, n\}$. It is useful to think of the rows and columns of M as being labelled by the elements of $[m]$ and $[n]$ respectively. If X is a subset of $[m]$ and Y is a subset of $[n]$ then $M_{X,Y}$ denotes the matrix obtained from M by deleting all the rows except those with labels in X and all the columns except those with labels in Y. The *Gaussian elimination greedoid* $(E, \mathcal{F})$ [19] of M is a greedoid such that $E = [n]$, the columns of M, and

$$\mathcal{F} = \{A \subseteq E : \text{ the submatrix } M_{[|A|],A} \text{ is non-singular}\}.$$

By convention the empty matrix is considered to be non-singular. We now prove that this does in fact define a greedoid. The axiom (G1) is a consequence of the convention that the empty set is non-singular. To establish (G2), it is helpful to prove that (G1′) holds. Suppose that the submatrix $M_{[|A|],A}$ is non-singular and $|A| \geq 2$. Then the determinant of $M_{[|A|],A}$ is non-zero, so its cofactor expansion about its bottom row, must include a term with a non-zero cofactor. Such a cofactor corresponds to an element e of A with $M_{[|A|-1],A-e}$ non-singular. This establishes (G1′). Now suppose that the submatrices $M_{[|A|],A}$ and $M_{[|B|],B}$ are non-singular and $|B| > |A|$. Because (G1′) holds, it suffices to prove that (G2) holds in the case when $|B| = |A| + 1$. The columns of $M_{[|A|],A}$ form a linearly independent set of size $|A|$. Consequently the columns of $M_{[|A|+1],A}$ also form a linearly independent set. The columns of $M_{[|B|],A}$ form a larger linearly independent set in the same vector space, so there is an element e of $B - A$ such that the columns of $M_{[|A|+1],A\cup e}$ form a linearly independent set. Thus $M_{[|A|+1],A\cup e}$ is non-singular. Therefore (G2) holds.

A Gaussian elimination greedoid over the field F is called *F-representable*. We let $\Gamma(M)$ denote the F-representable greedoid corresponding to the matrix M with entries from F. A greedoid that is $\mathbb{Z}_2$-representable is called *binary*.

Example 1.3.15. *Let*

$$M = \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \end{array}.$$

The binary greedoid $\Gamma(M)$ *has ground set* $\{1,2,3,4\}$ *and feasible sets*

$$\{\emptyset, \{1\}, \{4\}, \{1,3\}, \{1,4\}, \{3,4\}, \{1,2,3\}, \{1,2,4\}, \{2,3,4\}\}.$$

Lemma 1.3.16. *Let* M *be an* $m \times n$ *binary matrix with columns* $E = \{1,2,\dots,n\}$ *and let* M' *be obtained from* M *by adding row* i *to row* j *where* $i < j$*. Then* $\Gamma(M') \cong \Gamma(M)$*.*

Proof. Let $A \subseteq E$ and consider $N = M_{[|A|],A}$ and $N' = M'_{[|A|],A}$. If $|A| < j$ then $N = N'$ so A is feasible in $\Gamma(M)$ if and only if it feasible in $\Gamma(M')$. If $|A| \geq j$ then N' is obtained from N by adding row i of N to row j. Row operations do not affect the rank of a matrix, so A is feasible in $\Gamma(M)$ if and only if it is feasible in $\Gamma(M')$. □

If Γ is a binary greedoid then it follows from Lemma 1.3.16 that there is a binary matrix M with linearly independent rows such that $\Gamma = \Gamma(M)$. In certain results concerning M and $\Gamma(M)$ it is necessary to assume that the rows of M are linearly independent. For instance if M has linearly independent rows, then a loop in $\Gamma(M)$ is represented by a column of zeros in M.

Lemma 1.3.17. *Let* N *be a binary matrix with linearly independent rows. Then the bases of* $\Gamma(N)$ *coincide with the bases of the vector matroid* $M(N)$*.*

Proof. Let m denote the number of rows of N. Then $\rho(N) = r(N) = m$. Let X be a subset of columns of N with $|X| = m$. A square matrix is non-singular if and only if its columns form a linearly independent set.

If X is a basis of $M(N)$ then the restriction of N to the columns of X is a non-singular square submatrix of N of rank m and is consequently a basis of $\Gamma(N)$.

If X is a basis of $\Gamma(N)$ then X has rank m and the columns of X are linearly independent. Consequently X is a basis of $M(N)$. □

We define the *simplification* of a rooted graph, or more generally a greedoid, to be that obtained by deleting all but one element from every parallel class. If G' is the simplification of G then G is said to *simplify* to give G'.

Lemma 1.3.18. *Let Γ and Γ' be greedoids such that Γ' is the simplification of Γ. Then*

1. *Γ is a branching greedoid if and only if Γ' is a branching greedoid.*

2. *Γ is a binary greedoid if and only if Γ' is a binary greedoid.*

Proof. Suppose that Γ is a branching greedoid and $\Gamma = \Gamma(G)$ for some rooted graph G. Then adding or removing parallel elements to or from Γ yields a greedoid $\Gamma' = \Gamma(G')$ where G' is obtained from G by adding or deleting parallel edges. Now suppose that Γ is a binary greedoid and $\Gamma = \Gamma(M)$ for some matrix M with linearly independent rows. Then adding or removing parallel elements to or from Γ yields a greedoid $\Gamma' = \Gamma(M')$ where M' is formed from M by adding or deleting duplicate columns. $\square$

Let M be an $m \times n$ binary matrix and let e be the label of a column with entry 1 in the first row of M. We let M_e denote the matrix obtained by adding the first row of M to every other row of M in which the entry in column e is equal to one. This leads to a matrix in which the only non-zero entry in column e is in the first row. We now show that the class of binary greedoids is closed under contraction.

Lemma 1.3.19. *Let $\Gamma = (E, \mathcal{F})$ where $\Gamma = \Gamma(M)$ for some $m \times n$ binary matrix M, and let $\{e\} \in \mathcal{F}$. Let N be the binary matrix obtained by deleting the first row and the column with label e of the matrix M_e. Then $\Gamma/e = \Gamma(N)$.*

Proof. Suppose matrix M has columns labelled $1, 2, \ldots, n-2, n-1, e$. Then M_e has the form

$$M_e = \begin{array}{c} \begin{array}{cc} & e \end{array} \\ \begin{pmatrix} x & 1 \\ N & 0 \end{pmatrix} \end{array}$$

where x is a $1 \times (n-1)$ matrix and N is a $(m-1) \times (n-1)$ matrix. By Lemma 1.3.16 we have $\Gamma(M_e) = \Gamma$. Let $\Gamma' = \Gamma(N)$. Then $A \subseteq E - e$ is feasible in Γ' if and only if $A \cup e$ is feasible in $\Gamma(M_e)$. Thus Γ' is binary and $\Gamma' = \Gamma/e$. $\square$

Providing the rows of M are linearly independent, note that when we delete the first row of M_e, all columns corresponding to elements parallel to e will now comprise entirely zeros and thus are now loops in $\Gamma(N)$. The following example illustrates this lemma.

Example 1.3.20. *We can reduce the matrix from our previous example using row operations to get*

$$M_1 = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \end{array}.$$

Now by deleting the first row and column 1 *we get*

$$N = \begin{array}{c} \begin{matrix} 2 & 3 & 4 \end{matrix} \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array}.$$

The binary greedoid of N is given by $\Gamma(N) = (\{2,3,4\}, \mathcal{F}')$ where

$$\mathcal{F}' = \{\emptyset, \{3\}, \{4\}, \{2,3\}, \{2,4\}\}.$$

These are precisely the sets $X \subseteq \{2,3,4\}$ such that $X \cup 1 \in \mathcal{F}$.

We now present several results on the relationship between binary greedoids and branching greedoids. We show that all binary greedoids of rank 2 are branching greedoids, and then state the requirements for a branching greedoid to be a binary greedoid.

Proposition 1.3.21. *All binary greedoids of rank* 2 *are branching greedoids.*

Proof. Let Γ be a binary greedoid of rank two. By the remarks after Lemma 1.3.16 we may assume that $\Gamma = \Gamma(M)$ where M has two rows. Now let N be the matrix obtained by deleting all duplicate columns and any columns comprising entirely zeros in M.

Then N must be of one of the following forms (obviously the columns of these matrices can be rearranged without affecting the associated binary greedoid).

$$\begin{array}{c} \begin{matrix} 1 & 2 \end{matrix} \\ \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \end{array}, \quad \begin{array}{c} \begin{matrix} 1 & 2 \end{matrix} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}, \quad \begin{array}{c} \begin{matrix} 1 & 2 \end{matrix} \\ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \end{array}, \quad \begin{array}{c} \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \end{array}.$$

The respective binary greedoids associated to these matrices are $(\{1,2\}, \mathcal{F}_i)$ for $i = 1,2,3,4$ where $F_1 = F_2 = \{\emptyset, \{1\}, \{1,2\}\}$, $F_3 = \{\emptyset, \{1\}, \{2\}, \{1,2\}\}$ and $F_4 = \{\emptyset, \{1\}, \{2\}, \{1,2\}, \{1,3\}, \{2,3\}\}$. It is easy to check that all of these are branching greedoids. Their corresponding graphs are given in Figure 1.1.

Note that by adding duplicates of existing columns or columns comprising entirely zeros to any of these matrices would create parallel edges and loops in the rooted graphs associated to these branching greedoids respectively. □

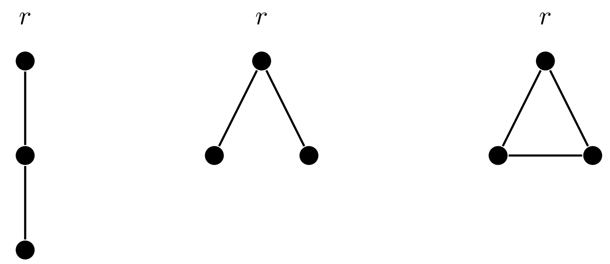

Figure 1.1: The three graphs referred to in the proof of Proposition 1.3.21.

We now show that the branching greedoid associated to a rooted graph that possesses a particular property is not binary. This is unlike matroids since every graphic matroid is binary.

Lemma 1.3.22. *Let $G = (r, V, E)$ be a rooted graph in which the root is adjacent to at least three vertices. Then $\Gamma(G)$ is not binary.*

Proof. Let M be a binary matrix such that $\Gamma(G) = \Gamma(M)$ and let i, j and k be three non-parallel edges that are incident to r in G. The first row of M must contain ones in the columns corresponding to i, j and k. Each of $\{i,j\}$, $\{i,k\}$ and $\{j,k\}$ is feasible in Γ. In order that $\{i,j\}$ and $\{i,k\}$ are feasible, the second row of M must contain either a zero in column i and ones in columns j and k, or a one in column i and zeros in columns j and k. But in either case $M_{[2],\{j,k\}}$ is singular giving a contradiction. $\square$

We now determine when a branching greedoid whose associated rooted graph has a root with degree 1 is binary.

Lemma 1.3.23. *Let $G = (r, V, E)$ be a rooted graph and let $G' = (r', V', E')$ be the graph obtained from G by attaching a path of length k to r for some $k \in \mathbb{N}$ and making the other end of the path r'. Then $\Gamma(G)$ is binary if and only if $\Gamma(G')$ is binary.*

Proof. Suppose that $\Gamma(G) = \Gamma(M)$ for some binary matrix M. Let

$$M' = \begin{pmatrix} I_k & 0 \\ 0 & M \end{pmatrix}.$$

We claim that $\Gamma(G') \cong \Gamma(M')$. To prove this we show that they both have the same feasible sets. Let A be a feasible set of $\Gamma(M')$ and let $|A| = t$ for $t > 0$. If $t \leq k$ then $A = \{1, \ldots, t\}$. If $t > k$ then $A = \{1, \ldots, k\} \cup F$ where F is a feasible set of $\Gamma(M)$. A set A with $|A| = t$ for $t > 0$ is feasible in $\Gamma(G')$ if and only if $G'|A$ is a rooted tree. Therefore if we label the path $1, 2, \ldots, k$ such that 1 is incident to r' and edge i is adjacent to edges $i-1$ and $i+1$ for $2 \leq i \leq k-1$, then we similarly have $A = \{1, \ldots, t\}$ if $t \leq k$ and $A = \{1, \ldots, k\} \cup F$ for some feasible set F of $\Gamma(G)$ if $t > k$. Thus $\Gamma(G') \cong \Gamma(M')$.

Now suppose that G'' is a rooted graph in which the root has degree 1 and that $\Gamma(G'') = \Gamma(M'')$ for some binary matrix M''. Let e denote the edge incident to the root in G''. Then $\{e\}$ is a singleton feasible set in $\Gamma(G'')$. By Lemma 1.3.19, $\Gamma(M'')/e$ is binary and furthermore $\Gamma(G''/e) = \Gamma(G'')/e = \Gamma(M'')/e$. Using induction it follows that if $\Gamma(G')$ is binary then so is $\Gamma(G)$. $\square$

We can now completely determine which branching greedoids, whose associated rooted graphs are connected, are binary greedoids.

Proposition 1.3.24. *Let $G = (r, V, E)$ be a connected rooted graph. Then $\Gamma(G)$ is binary if and only if after simplifying and removing loops G is either:*

1. *The path of length $n \geq 0$ with the root r at one endpoint and two leaves attached at the other.*

2. *The path of length $n \geq 0$ with the root r at one endpoint and the triangle attached at the other.*

3. *The path of length $n \geq 0$ with the root r at one endpoint.*

These graphs are given in Figure 1.2.

Proof. It is easy to check that if G is simplified and has one of the forms described then $\Gamma(G)$ is binary with one of the following representations, where in each case n is the length of the path:

$$I_n \quad , \quad \begin{pmatrix} I_n & 0 \\ 0 & Q \end{pmatrix} \quad , \quad \begin{pmatrix} I_n & 0 \\ 0 & R \end{pmatrix},$$

such that

$$Q = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

By Lemma 1.3.18, it follows that if the simplification of G has one of the forms listed, then $\Gamma(G)$ is binary.

Now suppose that $G = (r, V, E)$ is a rooted graph and $\Gamma(G)$ is binary. Let G' denote the graph obtained after simplifying G and removing loops. If $deg(r) = 0$ in G' then G' belongs to the third category in the proposition. If $deg(r) \geq 3$ in G' then Lemma 1.3.22 implies that $\Gamma(G')$ is not binary. Suppose $deg(r) = 1$ in G' and let e be the edge incident to r. By Lemma 1.3.23 the greedoid $\Gamma(G')$ is binary if and only if $\Gamma(G'/e)$ is binary, so we continue to contract the edge incident to the root until we have $deg(r) \neq 1$. Now suppose $deg(r) = 2$ in G'. Let i, j be the edges incident to the root. Now if there are no more edges in G' (i.e. G' is two leaf edges) then $\Gamma(G')$ is binary with a representation given by

$$\begin{array}{c} \begin{matrix} i & j \end{matrix} \\ \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \end{array}.$$

If there is one more edge k in G' such that k is adjacent to both i and j (i.e. G' is a triangle), then $\Gamma(G')$ is binary with a representation given by

$$\begin{array}{c} \begin{matrix} i & j & k \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array}.$$

Now suppose there is an edge k adjacent to i but not j in G'. Let $e_1, \ldots, e_t$ be the remaining edges in G'. Then $\{i, k\} \in \mathcal{F}$ but $\{j, k\} \notin \mathcal{F}$. If $\Gamma(G')$ were binary then the first row in a representation of $\Gamma(G')$ would be of the form

$$\begin{array}{cccccc} i & j & k & e_1 & \dots & e_t \\ \end{array}$$
$$\begin{pmatrix} 1 & 1 & 0 & 0 & \dots & 0 \end{pmatrix}.$$

Now the second row must have a 1 in the column for k so that $\{i,k\} \in \mathcal{F}$. However this would imply $\{j,k\} \in \mathcal{F}$. Hence if $deg(r) = 2$ in G' then $\Gamma(G')$ is binary if G' is either two leaf edges or a triangle. $\square$

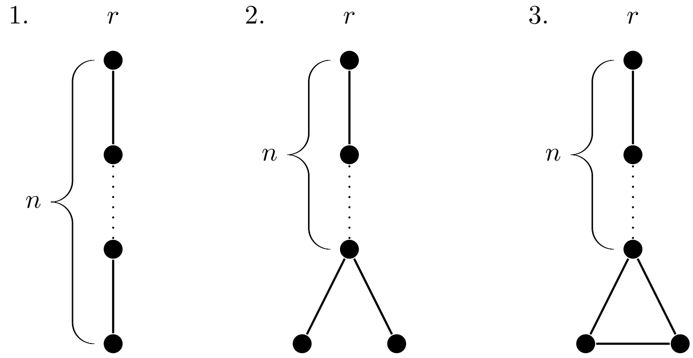

Figure 1.2: The graphs referred to in Proposition 1.3.24.

1.4 Complexity

Computational complexity theory determines the practical limits on the capabilities of a computer. The term 'computer' is quite ambiguous, which is why we commonly use the Turing machine as our chosen mathematical model of computation. It is a theoretical device that manipulates a finite set of symbols contained on a strip of tape. In Turing's thesis [54] he states that if a problem can be solved by an algorithm, then there exists a Turing machine that can solve it. We consider two particular types of Turing machine, namely a deterministic Turing machine (DTM) and a non-deterministic Turing machine (NDTM). A DTM is the most basic Turing machine which follows a fixed set of rules to determine all future actions, eventually halting with a "yes" or a "no". Before the DTM starts, the input or description of the instance of the problem to be solved is written on the tape. If the algorithm halts in the "yes" state for an instance x, it is said to accept x, otherwise it is said to reject x. A NDTM is composed of two separate stages, the first being a guessing stage, and the second a checking stage which proceeds to compute in a normal deterministic manner. For a problem instance I, the first stage merely guesses some string S. Both I and S are then provided as the input to the checking stage. So instead of having just one possible action on a given input, it has a different one for every possible guess. For a comprehensive introduction to complexity theory and a detailed description of the structure of these Turing machines along with examples, see [18] and [44].

The *time complexity function* of an algorithm expresses its time requirements by giving, for each possible input length, the largest amount of time needed by the algorithm to solve a problem instance of that size. We use the *Big 'O'* notation to describe the asymptotic behaviour of an algorithm. This excludes coefficients and lower order terms. Letting $T(n)$ denote the maximum time taken by the algorithm on any input of size n, we say that the algorithm runs in *constant-time* if $T(n) = O(1)$, *linear-time* if $T(n) = O(n)$ and *polynomial-time* if $T(n) = O(n^k)$ for some constant k. It is widely agreed that a problem has not been well solved until a polynomial-time algorithm is known for it. Time complexity is a worst-case measure: the fact that an algorithm has complexity n^3 say, means only that at least one problem instance of size n requires that much time. In [18] it is claimed that even though an algorithm having time complexity $O(n^{100})$ might not be considered likely to run quickly in practice, the polynomially solvable problems that arise naturally tend to be solvable with polynomial time bounds that have degree 2 or 3 at worst.

In practice, standard encoding schemes used to describe instances of a problem differ in size at most polynomially. Consequently any algorithm for a problem having polynomial-time complexity under one encoding scheme will have polynomial-time complexity under all other reasonable encoding schemes. For example in [18] they give three ways of encoding a graph and verify that the input lengths that each determines differ at most polynomially from one another.

Suppose we have two computational problems π_1 and π_2. We say that π_2 is *Turing reducible* to π_1 if there exists a deterministic Turing machine which solves π_2 in polynomial time using an oracle for π_1 which returns an answer in constant-time. When π_2 is Turing reducible to π_1 we write $\pi_2 \propto_T \pi_1$ and we say that solving problem π_1 is at least as hard as solving problem π_2.

The class of problems solvable in polynomial time is closed under Turing reduction.

A *decision problem* is one whose solution to an algorithm is either yes or no (or 1 or 0 in binary). The complexity class P (or PTIME) contains all decision problems that can be solved in polynomial time using a DTM. The problems in this class are said to be "efficient" or "easy". The complexity class NP contains all decision problems that can be verified in polynomial time using a NDTM. Consider the following subset-sum problem. Suppose we have a set of integers as our input, and we wish to know whether some of these integers sum to zero. As we increase the number of integers in our input set, the number of subsets grows exponentially and therefore so would the time of any algorithm which examines all the possible subsets. However, if we are given a particular subset, we can easily verify whether or not the integers in the subset sum to zero. Therefore the subset-sum problem is in NP because the guesses may be interpreted as encoding different subsets and verifying can be done in polynomial time.

A polynomial reduction from a decision problem π' to a decision problem π is a function f mapping instances of π' to instances of π such that f is computable in time polynomial in the size of its input and x is a "yes" instance of π' if and only if $f(x)$ is a "yes" instance of π. A decision problem π is NP-complete if it belongs to NP and $\pi' \propto_T \pi$ for every π' in NP. The NP-complete are often thought of as the hardest problems in NP. Because the $\propto_T$ relation is transitive, to prove that a decision problem π in NP is NP-complete it is only necessary to show that $\pi' \propto_T \pi$ for some NP-complete problem π'.

Although there are many more computational classes, we need only define one more that will be of significant importance in later sections, and will be useful when we need more than just a "yes" or "no" output. We define #P to be the counting analogue of NP, that is the class of all counting problems corresponding to the decision problems in NP. More specifically, a problem is in #P if it counts the number of accepting computations of a problem in NP. Consider the NP problem of determining whether a graph has an independent set of size k. The corresponding #P problem would be to determine how many independent sets of size k the graph has. A problem in #P must be at least as hard as the corresponding problem in NP. If it is easy to count the accepting computations, then it must be easy to tell if there is at least one.

A computational problem π is said to be *#P-hard* if $\pi' \propto_T \pi$ for all $\pi' \in$ #P, and *#P-complete* if, in addition, $\pi \in$ #P. These problems are thus said to be the "hardest" members of #P. Examples of #P-complete problems include counting the number of 3-colourings of a graph, and counting the number of subtrees of a graph, the latter is a result due to Jerrum [32].

To prove that a problem π is #P-hard we need only show that there exists one problem π' that is #P-hard such that $\pi' \propto_T \pi$, since by transitivity this would imply π is Turing reducible to every problem in #P.

1.5 The Tutte Polynomial of a Graph

The Tutte polynomial is a renowned tool for analyzing graphs or, more generally, matroids. It is an extensively studied, two-variable polynomial with rich theory and wide applications. Constructed by W. T. Tutte in 1947 [55], the Tutte polynomial has the universal property that essentially any graph invariant f with $f(G \cup H) = f(G)f(H)$ for any two graphs G and H that have no edges and at most one vertex in common, and with a deletion/contraction reduction, must be an evaluation of it. The Tutte polynomial has many different, yet equivalent, definitions. Below we give the rank generating function definition of the Tutte polynomial followed by its deletion/contraction recursion. In Chapter 4 we explore a convolution formula for the Tutte polynomial.

The *Tutte polynomial* of a graph $G = (V, E)$ is given by

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

The exponent $|A| - r(A)$ is often referred to as the *nullity* of A and essentially counts the minimum number of elements that need to be removed from A in order to make $G|A$ a forest (or more generally an independent set of a matroid).

If there exists $e \in E(G)$ which is neither a loop nor a coloop, then

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

Otherwise, G consists of i coloops and j loops and

$$T(G; x, y) = x^i y^j.$$

Suppose $G \cup H$ is the disjoint union of graphs G and H, and $G * H$ is the vertex-join of G and H, that is the graph formed by identifying a vertex of G with a vertex of H, then we have

$$T(G \cup H; x, y) = T(G * H; x, y) = T(G; x, y)T(H; x, y).$$

The Tutte polynomial encodes many interesting properties of a graph. Some interpretations for evaluations of the Tutte polynomial of a graph $G = (V, E)$ are given below:

- $T(G; 1, 1)$ = the number of spanning trees of G if G is connected, or the number of maximal spanning forests of G otherwise, where a maximal spanning forest comprises a spanning tree of each component.
- $T(G; 2, 1)$ = the number of subgraphs of G with vertex set $V(G)$ and containing no cycles.
- $T(G; 1, 2)$ = the number of spanning subgraphs of G having the same number of components as G.
- $T(G; 2, 2) = 2^{|E|}$ = the number of orientations of G = the number of spanning subgraphs of G.
- $T(G; 2, 0)$ = the number of acyclic orientations of G [51].
- $T(G; 0, 2)$ = the number of totally cyclic orientations of G [57].
- $T(G; 1, 0)$ = the number of acyclic orientations of G with one predefined source vertex per component of G [25].

For a full list of evaluations see [17].

The chromatic polynomial $P(G; \lambda)$ of a graph $G = (V, E)$ was first introduced by Birkhoff [4] in an attempt to solve the famous four-colour problem. Using a palette of λ colours, it counts the number of ways of vertex colouring a graph such that no two adjacent vertices are assigned the same colour. These are called the *proper colourings* of the graph. The chromatic polynomial is related to the Tutte polynomial of a graph as follows

$$P(G; \lambda) = (-1)^{r(G)} \lambda^{\kappa(G)} T(G; 1 - \lambda, 0).$$

Suppose that the edges of G are given an orientation. A nowhere zero λ-flow is a mapping from the edges of G to $\{1, \ldots, \lambda - 1\}$ such that at each vertex v the difference between the sum of the labels on edges entering v and the sum of the labels on edges leaving v is divisible by λ. It is not difficult to see that the number of nowhere zero λ-flows is independent of the choice of orientation. The *flow polynomial* $\chi^*(G; \lambda)$ of a graph G gives the number of nowhere zero λ-flows of G. It is a specialization of the Tutte polynomial given by

$$\chi^*(G;\lambda) = (-1)^{|E|-r(G)}T(G;0,1-\lambda).$$

If G is planar then $\chi^*(G;\lambda) = \frac{1}{\lambda^{\kappa(G)}}P(G^*;\lambda)$.

Before demonstrating a direct relationship between the Tutte polynomial and statistical physics, we first need to define a family of hyperbolae which will play a special role in the remaining chapters. For $\alpha \in \mathbb{Q} - \{0\}$ let $H_\alpha = \{(x,y) \in \mathbb{Q}^2 : (x-1)(y-1) = \alpha\}$, and let $H_0^x = \{(1,y) : y \in \mathbb{Q}\}$ and $H_0^y = \{(x,1) : x \in \mathbb{Q}\}$. Along H_q, for any positive integer q, the Tutte polynomial specializes to the partition function of the q-state Potts model and, in particular, the partition function of the Ising model when $q = 2$.

For a planar graph G with dual graph G^* it should be straightforward to check that

$$T(G;x,y) = T(G^*;y,x).$$

1.6 The Complexity of Computing the Tutte Polynomial of a Graph

Along the hyperbola H_1 the Tutte polynomial of a graph $G = (V,E)$ reduces to

$$T(G;x,y) = x^{|E|}(x-1)^{r(G)-|E|}$$

which is easily computed. The complete classification of the complexity of computing the Tutte polynomial at any point is given by Jaeger, Vertigan and Welsh in [31]. Here they consider points in an extension field of $\mathbb{Q}$ containing $i = \sqrt{-1}$ and $j = e^{2\pi i/3}$. We shall only present their results on the hardness of rational points because these will be what we work with in the following chapters.

Theorem 1.6.1 (Jaeger, Vertigan, Welsh)**.** *Evaluating the Tutte polynomial of a graph at any fixed point (a,b) in the rational xy-plane is #P-hard apart from when (a,b) lies on H_1, or when (a,b) equals $(-1,0),(0,-1),(-1,-1)$ or $(1,1)$, when there exists a polynomial-time algorithm.*

Vertigan strengthened this result to planar graphs in [58] and then together with Welsh in [59] to bipartite graphs. These results are summarized in the following theorem.

Theorem 1.6.2 (Vertigan, Welsh)**.** *Evaluating the Tutte polynomial of a bipartite planar graph at any fixed point (a, b) in the rational xy-plane is #P-hard apart from when (a, b) lies on H_1 or H_2, or when (a, b) equals $(-1, -1)$ or $(1, 1)$, when there exists a polynomial-time algorithm.*

Both cases of these results extend to the extension field of $\mathbb{Q}$ subject to including $(i, -i), (-i, i), (j, j^2)$ and (j^2, j) as easy points. Note that Theorem 1.6.1 implies that evaluating the partition function of the Ising model is #P-hard and similarly evaluating the chromatic polynomial $P(G; \lambda)$ and the flow polynomial $\chi^*(G; \lambda)$ are #P-hard for $\lambda \notin \{0, 1, 2\}$ and $\lambda \notin \{1, 2\}$ respectively.

The Tutte polynomial of a graph G can be expressed in the form

$$T(G; x, y) = \sum_{i,j \geq 0} b_{i,j}(G) x^i y^j.$$

We can lose the argument in $b_{i,j}(G)$ when the context is clear. Computing the complexities of the coefficients of the Tutte polynomial of a graph was considered by Annan in [2]. Here Annan reduces the problem of counting the number of 3-colourings of a graph (a well-known #P-complete problem) to that of computing $b_{1,0}$, i.e. the coefficient of x in the Tutte polynomial of the graph. One of the main theorems from [2] is presented below.

Theorem 1.6.3 (Annan)**.** *For all fixed nonnegative integers i, j, the coefficients $b_{i+1,j}$ and $b_{i,j+1}$ are #P-complete to compute.*

1.7 The Tutte Polynomial of an Arbitrary Greedoid, a Rooted Graph and a Rooted Digraph

Originally motivated by the Tutte polynomial of a matroid, McMahon and Gordon define the Tutte polynomial of a greedoid in [21]. In particular they define it of a rooted graph and of a rooted digraph. We begin by presenting the rank generating function definition and the deletion/contraction recursion of the Tutte polynomial of an arbitrary greedoid and then present results specifically on the Tutte polynomial of a rooted graph and of a rooted digraph.

The *Tutte polynomial* of a greedoid $\Gamma = (E, \mathcal{F})$ is given by

$$T(\Gamma; x, y) = \sum_{A \subseteq E} (x-1)^{\rho(\Gamma)-\rho(A)} (y-1)^{|A|-\rho(A)}. \tag{1.2}$$

For a rooted graph G we let $T(G; x, y) = T(\Gamma(G); x, y)$. Similarly to the Tutte polynomial of a matroid we can compute $T(\Gamma; x, y)$ recursively by contracting and deleting a feasible singleton in Γ, more specifically an edge incident to the root of G if $\Gamma = \Gamma(G)$ where G is a rooted graph.

Proposition 1.7.1. *Let $\Gamma = (E, \rho_\Gamma)$ be a greedoid specified by its rank function.*

1. *If e is an element of Γ with $\rho_\Gamma(e) = 1$ then*

$$T(\Gamma; x, y) = T(\Gamma/e; x, y) + (x-1)^{\rho(\Gamma)-\rho(\Gamma\setminus e)} T(\Gamma \setminus e; x, y).$$

2. *If e is a loop of Γ then*

$$T(\Gamma; x, y) = yT(\Gamma \setminus e; x, y).$$

Proof. 1. Let e be an element of Γ with $\rho_\Gamma(e) = 1$. The Tutte polynomial of Γ can be expressed in the following way.

$$\begin{aligned} T(\Gamma; x, y) = &\sum_{e \in A \subseteq E} (x-1)^{\rho(\Gamma)-\rho_\Gamma(A)}(y-1)^{|A|-\rho_\Gamma(A)} \\ &+ \sum_{e \notin A \subseteq E} (x-1)^{\rho(\Gamma)-\rho_\Gamma(A)}(y-1)^{|A|-\rho_\Gamma(A)}. \end{aligned} \tag{1.3}$$

Suppose that A is a subset of E containing e. Let $A' = A - e$. Then by Equation 1.1 $\rho_{\Gamma/e}(A') = \rho_\Gamma(A) - 1$. In particular $\rho(\Gamma) = \rho(\Gamma/e) + 1$. Therefore we can write the first summation in Equation 1.3 as follows.

$$\begin{aligned} \sum_{e \in A \subseteq E} (x-1)^{\rho(\Gamma)-\rho_\Gamma(A)}(y-1)^{|A|-\rho_\Gamma(A)} &= \sum_{A' \subseteq E-e} (x-1)^{\rho(\Gamma/e)-\rho_{\Gamma/e}(A')}(y-1)^{|A'|-\rho_{\Gamma/e}(A')} \\ &= T(\Gamma/e; x, y). \end{aligned}$$

Now suppose that A is a subset of $E - e$. Then $\rho_{\Gamma\setminus e}(A) = \rho_\Gamma(A)$. The Tutte polynomial of $\Gamma \setminus e$ is given by

$$T(\Gamma \setminus e; x, y) = \sum_{A \subseteq E-e} (x-1)^{\rho(\Gamma\setminus e)-\rho_{\Gamma\setminus e}(A)}(y-1)^{|A|-\rho_{\Gamma\setminus e}(A)}.$$

Therefore we can write the second summation in Equation 1.3 as follows.

$$\sum_{e \notin A \subseteq E} (x-1)^{\rho(\Gamma)-\rho_\Gamma(A)}(y-1)^{|A|-\rho_\Gamma(A)} = (x-1)^{\rho(\Gamma)-\rho(\Gamma\setminus e)}T(\Gamma \setminus e; x, y).$$

2. Let e be a loop of Γ. For any subset A of $E-e$, we have $\rho_\Gamma(A \cup e) = \rho_\Gamma(A) = \rho_{\Gamma\setminus e}(A)$. In particular $\rho(\Gamma) = \rho(\Gamma \setminus e)$. Thus proceeding in a similar way to the first part

$$\begin{aligned} T(\Gamma; x, y) &= \sum_{A \subseteq E-e} (x-1)^{\rho(\Gamma\setminus e)-\rho_{\Gamma\setminus e}(A)} \cdot \left((y-1)^{|A|-\rho_{\Gamma\setminus e}(A)} + (y-1)^{|A|+1-\rho_{\Gamma\setminus e}(A)}\right) \\ &= y \sum_{A \subseteq E-e} (x-1)^{\rho(\Gamma\setminus e)-\rho_{\Gamma\setminus e}(A)}(y-1)^{|A|-\rho_{\Gamma\setminus e}(A)} \\ &= yT(\Gamma \setminus e; x, y). \end{aligned}$$

□

Using the deletion/contraction recursion outlined in Proposition 1.7.1, we now derive the Tutte polynomial of a branching greedoid whose corresponding rooted graph is a path rooted at one of the endpoints.

Proposition 1.7.2. *Let $\Gamma = \Gamma(P_k)$ where P_k is the rooted path of length k such that the root is one of the leaves. Then*

$$T(\Gamma; x, y) = 1 + \sum_{i=1}^{k} (x-1)^i y^{i-1}.$$

Proof. We use induction on k. When $k = 0$ the graph P_0 comprises the root vertex and no edges. Thus $T(\Gamma(P_0); x, y) = 1$. Now assume the result holds for $k = j$. Let e be the edge incident to the root vertex in P_{j+1}, i.e. e is an element of $\Gamma(P_{j+1})$ with $\rho_{\Gamma(P_{j+1})}(e) = 1$. By Proposition 1.7.1 we have

$$\begin{aligned} T(\Gamma(P_{j+1}); x, y) &= T(\Gamma(P_{j+1})/e; x, y) + (x-1)^{j+1}T(\Gamma(P_{j+1}) \setminus e; x, y) \\ &= T(\Gamma(P_j); x, y) + (x-1)^{j+1}y^j \\ &= 1 + \sum_{i=1}^{j} (x-1)^i y^{i-1} + (x-1)^{j+1}y^j \\ &= 1 + \sum_{i=1}^{j+1} (x-1)^i y^{i-1}. \end{aligned}$$

$\square$

Note that if $\Gamma = \Gamma(P_k)$ then Γ will be isomorphic to a greedoid with ground set $[k]$ and feasible sets $\{\emptyset, [1], [2], \ldots, [k]\}$. The following proposition is also simple to prove using induction.

Proposition 1.7.3. *Let $\Gamma = \Gamma(S_k)$ where S_k is the rooted star graph with k edges emanating from the root. Then*

$$T(\Gamma; x, y) = x^k.$$

Unlike the Tutte polynomial of a matroid, the Tutte polynomial of a greedoid can have negative coefficents. For example by Proposition 1.7.2 we have $T(\Gamma(P_2); x, y) = x^2y - 2xy + x + y$.

Equation 1.2 can be specialized to rooted graphs in the following way. The *Tutte polynomial* of a rooted graph $G = (r, V, E)$ is given by

$$T(G; x, y) = \sum_{A \subseteq E} (x-1)^{\rho(G)-\rho(A)}(y-1)^{|A|-\rho(A)}. \tag{1.4}$$

Let T_1 and T_2 be rooted trees with no isolated vertices. It follows from Gordon and McMahon [21] that if $T(T_1; x, y) = T(T_2; x, y)$ then $T_1 \cong T_2$. This implies that a rooted tree without isolated vertices is completely determined by its Tutte polynomial.

Suppose we have two disjoint rooted graphs G_1 and G_2, rooted at vertices r_1 and r_2 respectively. Define the *direct sum* $G_1 \oplus G_2$ of G_1 and G_2 to be the rooted graph formed by glueing G_1 and G_2 together with new root vertex created by identifying r_1 and r_2. Gordon and McMahon [21] show that

$$T(G_1 \oplus G_2; x, y) = T(G_1; x, y)T(G_2; x, y). \tag{1.5}$$

The Tutte polynomial of a rooted graph shares many of the same evaluations as the Tutte polynomial of a graph. Let G be a rooted graph.

- $T(G; 1, 1) =$ the number of spanning trees of the connected component containing the root in G.
- $T(G; 2, 1) =$ the number of rooted subtrees of G.
- $T(G; 1, 2) =$ the number of spanning subgraphs of G in which the component containing the root in G is also a connected component, meaning every vertex connected to the root in G is also connected to the root in the spanning subgraph.

- $T(G;2,2) = 2^{|E|}$ = the number of spanning subgraphs of G.

- $T(G;1,0)$ = the number of acyclic orientations of G with a unique source. If G has more than one component containing an edge, then $T(G;1,0) = 0$. This result follows from Greene and Zaslavsky's interpretation [25] of the Tutte polynomial of an unrooted graph at $(1,0)$.

Let G be a connected rooted graph and let G' be the underlying unrooted graph of G. Then

$$T(G;1,y) = T(G';1,y).$$

That is, the Tutte polynomial of a connected rooted graph coincides with the Tutte polynomial of the corresponding unrooted graph along the line $x = 1$. This is easy to prove by noting that $\rho(G) = r(G')$ and that the subsets $A \subseteq E$ with non-zero terms in Equation 1.4 when $x = 1$ are precisely those with $\rho_G(A) = \rho(G)$. Moreover $r_{G'}(A) = r(G')$ if and only if $\rho_G(A) = \rho(G)$. This result will be of particular importance in the next chapter where we use the complexity of computing $T(G';1,y)$ to find the complexity of computing $T(G;1,y)$.

We now present a specialization of the Tutte polynomial of a rooted graph. The characteristic polynomial of a rooted graph (more generally a greedoid) was first introduced by Gordon and McMahon in [22] and is a generalization of the chromatic polynomial of a graph. For a rooted graph G, the one-variable *characteristic polynomial* $p(G;\lambda)$ is defined by

$$p(G;\lambda) = (-1)^{\rho(G)} T(G;1-\lambda,0). \tag{1.6}$$

The Tutte polynomial of a rooted digraph D is defined by $T(D;x,y) = T(\Gamma(D);x,y)$. Proposition 1.7.1 implies that $T(D;x,y)$ satisfies a delete/contract recurrence. The direct sum of two rooted digraphs is defined in exactly the same way as for rooted graphs and the analogue of Equation 1.5 holds. Let D be a rooted digraph. A subgraph T of D is said to be *full* if every vertex that is reachable by a directed path from the root in D is also reachable by a directed path from the root in T. If T is also a rooted arborescence then it is called a *full arborescence*. We now present some evaluations of the Tutte polynomial of the rooted digraph $D = (r, V, \overrightarrow{E})$.

- $T(D;1,1)$ = the number of spanning subgraphs of D that are full arborescences.

- $T(D;2,1)$ = the number of spanning subgraphs of D that are arborescences.

- $T(D; 1, 2) =$ the number of spanning subgraphs of D that are full.
- $T(D; 2, 2) = 2^{|\overrightarrow{E}|} =$ the number of spanning subgraphs of D.
- $T(D; 1, 0) = 1$ if G is acyclic and every vertex can be reached by a directed path in D, and 0 otherwise.

The last evaluation will be discussed in more detail in Section 2.4.

Proposition 1.7.4. *Let D be rooted digraph. Then $T(D; 1, 2) = 1$ if and only if D is a rooted arborescence.*

Proof. Let D be a rooted digraph with root vertex r. If D is a rooted arborescence then clearly $T(D; 1, 2) = 1$.

Now assume $T(D; 1, 2) = 1$ and D is not a rooted arborescence. Since D only has one spanning subgraph that is full, the underlying unrooted graph of D must not contain any cycles. Suppose there exists a vertex v such that there is no directed path from r to v in D. Since D is not a rooted arborescence we can let v be a vertex that is not isolated. Let $e_1, \ldots, e_t$ be the edges incident to v in D. These are all loops in the corresponding greedoid. Let D' be the subgraph of D obtained by deleting the edges incident with v. By Proposition 1.7.1 we have

$$T(D; x, y) = y^t T(D'; x, y).$$

Therefore $T(D; 1, 2) = 2^t T(D'; 1, 2)$. Since $t \geq 1$ we have $T(D; 1, 2) \neq 1$, which is a contradiction. This implies that there are no vertices in D that are not reachable by a directed path. Furthermore since the underlying unrooted graph of D is acyclic, D must be a rooted arborescence. $\square$

Chapter 2

The Computational Complexity of Evaluating the Tutte Polynomial at a Fixed Point

2.1 Introduction

By Theorem 1.6.1 we know that evaluating the Tutte polynomial of a general graph is #P-hard at most fixed points in the rational xy-plane. In this chapter we present analogous results which completely determine the complexity of evaluating the Tutte polynomial of a rooted graph, a rooted digraph and of a binary greedoid at any fixed point in the rational xy-plane. For the rooted graph case we also strengthen our results by restricting ourselves to planar, bipartite rooted graphs.

The following two graph operations play a significant role in the proof of Theorem 1.6.1. The *k-stretch* G_k of a graph G is obtained by replacing every edge of G by a path of length k, and the *k-thickening* G^k of G is obtained by replacing every edge of G by k parallel edges. Note that both of these operations can be generalized to matroids [11]. The following lemma illustrates the effect of the k-stretch and k-thickening operations on the Tutte polynomial of a graph (more generally a matroid) [10].

Lemma 2.1.1. *Let $G = (V, E)$ be a graph. The Tutte polynomial of the k-stretch G_k of G when*

$x \neq -1$ is given by

$$T(G_k; x, y) = (1 + x + \ldots + x^{k-1})^{|E|-r(G)} T\left(G; x^k, \frac{y + x + \ldots + x^{k-1}}{1 + x + \ldots + x^{k-1}}\right).$$

When $x = -1$ we have

$$T(G_k; -1, y) = \begin{cases} (y-1)^{|E|-r(G)} & \text{if } k \text{ is even;} \\ T(G; -1, y) & \text{if } k \text{ is odd.} \end{cases}$$

The Tutte polynomial of the k-thickening G^k of G when $y \neq -1$ is given by

$$T(G^k; x, y) = (1 + y + \ldots + y^{k-1})^{r(G)} T\left(G; \frac{x + y + \ldots + y^{k-1}}{1 + y + \ldots + y^{k-1}}, y^k\right).$$

When $y = -1$ we have

$$T(G^k; x, -1) = \begin{cases} (x-1)^{r(G)} & \text{if } k \text{ is even;} \\ T(G; x, -1) & \text{if } k \text{ is odd.} \end{cases}$$

Any encoding of the information specifying a graph can be transformed into the encoding of the k-stretch and k-thickening of G in polynomial time.

2.2 Greedoid Constructions

In this section we introduce three greedoid constructions and give expressions for the Tutte polynomial of greedoids resulting from these constructions.

The first construction is just the generalization of the k-thickening operation from matroids to greedoids. Given a greedoid $\Gamma = (E, \mathcal{F})$, its k-thickening is the greedoid Γ^k that, informally speaking, is formed from Γ by replacing each edge by k parallel edges. More precisely, Γ^k has element set $E' = E \times [k]$ and collection $\mathcal{F}'$ of feasible sets as follows. Define μ to be the projection operator $\mu : 2^{E \times [k]} \to 2^E$ so that element $e \in \mu(A)$ if and only if $(e, i) \in A$ for some i. Now a subset A is feasible in Γ^k if and only if $\mu(A)$ is feasible in Γ and $|\mu(A)| = |A|$. The latter condition ensures that A does not contain more than one element replacing a particular element of Γ.

It is clear that Γ^k is a greedoid and moreover $\rho_{\Gamma^k}(A) = \rho_\Gamma(\mu(A))$. In particular $\rho(\Gamma^k) = \rho(\Gamma)$.

For any element e of Γ the elements (e,i) and (e,j) are parallel. The effect of the k-thickening operation on the Tutte polynomial of a greedoid is given in the following theorem. In fact, the formula is consistent with that in Lemma 2.1.1.

Theorem 2.2.1. *Let $\Gamma = (E, \mathcal{F})$ be a greedoid. The Tutte polynomial of the k-thickening Γ^k of Γ when $y \neq -1$ is given by*

$$T(\Gamma^k; x, y) = (1 + y + \ldots + y^{k-1})^{\rho(\Gamma)} T\left(\Gamma; \frac{x + y + \ldots + y^{k-1}}{1 + y + \ldots + y^{k-1}}, y^k\right). \tag{2.1}$$

When $y = -1$ we have

$$T(\Gamma^k; x, -1) = \begin{cases} (x-1)^{\rho(\Gamma)} & \text{if } k \text{ is even;} \\ T(\Gamma; x, -1) & \text{if } k \text{ is odd.} \end{cases}$$

Proof. Let $\Gamma^k = (E', \mathcal{F}')$ be the k-thickened greedoid. Then $E' = E \times [k]$. Let μ be the mapping defined in the discussion at the beginning of this section. To ensure that we do not divide by zero in our calculations, we prove the case when $y = 1$ separately.

For each $A' \subseteq E'$ we have $\rho_{\Gamma^k}(A') = \rho_\Gamma(\mu(A'))$ and furthermore $\rho(\Gamma^k) = \rho(\Gamma)$. The Tutte polynomial of Γ^k when $y \notin \{-1, 1\}$ is thus given by

$$\begin{aligned}
T(\Gamma^k; x, y) &= \sum_{A' \subseteq E'} (x-1)^{\rho(\Gamma^k) - \rho_{\Gamma^k}(A')} (y-1)^{|A'| - \rho_{\Gamma^k}(A')} \\
&= \sum_{A \subseteq E} \sum_{\substack{A' \subseteq E': \\ \mu(A') = A}} (x-1)^{\rho(\Gamma) - \rho_\Gamma(\mu(A'))} (y-1)^{|A'| - \rho_\Gamma(\mu(A'))} \qquad (2.2) \\
&= \sum_{A \subseteq E} (x-1)^{\rho(\Gamma) - \rho_\Gamma(A)} (y-1)^{-\rho_\Gamma(A)} \sum_{\substack{A' \subseteq E': \\ \mu(A') = A}} (y-1)^{|A'|} \\
&= \sum_{A \subseteq E} (x-1)^{\rho(\Gamma) - \rho_\Gamma(A)} (y-1)^{-\rho_\Gamma(A)} (y^k - 1)^{|A|} \\
&= \sum_{A \subseteq E} (x-1)^{\rho(\Gamma) - \rho_\Gamma(A)} (y-1)^{-\rho_\Gamma(A)} (y^k - 1)^{|A| - \rho_\Gamma(A)} \left(\frac{1}{y^k - 1}\right)^{\rho(\Gamma) - \rho_\Gamma(A)} (y^k - 1)^{\rho(\Gamma)} \\
&= (1 + y + \ldots + y^{k-1})^{\rho(\Gamma)} \sum_{A \subseteq E} \left(\frac{(x-1)(y-1)}{y^k - 1}\right)^{\rho(\Gamma) - \rho_\Gamma(A)} (y^k - 1)^{|A| - \rho_\Gamma(A)} \\
&= (1 + y + \ldots + y^{k-1})^{\rho(\Gamma)} T\left(\Gamma; \frac{x + y + \ldots + y^{k-1}}{1 + y + \ldots + y^{k-1}}, y^k\right).
\end{aligned}$$

When $y = 1$ we get non-zero terms in Equation 2.2 if and only if $|A'| = \rho_\Gamma(\mu(A'))$, which implies that $|A'| = |A|$. For each $A \subseteq E$ there are $k^{|A|}$ choices for A' such that $|A'| = |A|$ (k choices for each of the elements in A). Therefore we have

$$\begin{aligned}
T(\Gamma^k; x, 1) &= \sum_{\substack{A \subseteq E: \\ \rho_\Gamma(A)=|A|}} (x-1)^{\rho(\Gamma)-\rho_\Gamma(A)} \sum_{\substack{A' \subseteq E': \\ \mu(A')=A, \\ |A'|=|A|}} 1 \\
&= \sum_{\substack{A \subseteq E: \\ \rho_\Gamma(A)=|A|}} (x-1)^{\rho(\Gamma)-\rho_\Gamma(A)} k^{\rho_\Gamma(A)} \\
&= \sum_{\substack{A \subseteq E: \\ \rho_\Gamma(A)=|A|}} \left(\frac{x-1}{k}\right)^{\rho(\Gamma)-\rho_\Gamma(A)} k^{\rho(\Gamma)} \\
&= k^{\rho(\Gamma)} T\left(\Gamma; \frac{x+k-1}{k}, 1\right)
\end{aligned}$$

which agrees with Equation 2.1 when $y = 1$.

When $y = -1$ we have

$$\begin{aligned}
T(\Gamma^k; x, -1) &= \sum_{A \subseteq E} \sum_{\substack{A' \subseteq E': \\ \mu(A')=A}} (x-1)^{\rho(\Gamma)-\rho_\Gamma(\mu(A'))} (-2)^{|A'|-\rho_\Gamma(\mu(A'))} \\
&= \sum_{A \subseteq E} (x-1)^{\rho(\Gamma)-\rho_\Gamma(A)} (-2)^{-\rho_\Gamma(A)} \sum_{\substack{A' \subseteq E': \\ \mu(A')=A}} (-2)^{|A'|} \\
&= \sum_{A \subseteq E} (x-1)^{\rho(\Gamma)-\rho_\Gamma(A)} (-2)^{-\rho_\Gamma(A)} ((-1)^k - 1)^{|A|} \\
&= \begin{cases} (x-1)^{\rho(\Gamma)} & \text{if } k \text{ is even;} \\ T(\Gamma; x, -1) & \text{if } k \text{ is odd.} \end{cases}
\end{aligned}$$

Note that the only contribution to $T(\Gamma^k; x, -1)$ when k is even is from the empty set. $\square$

The second construction is a little more involved. To motivate it we first describe a natural construction operation on rooted graphs. Let G and H be disjoint rooted graphs with G being connected. Then the *H-attachment* of G, denoted by $G \sim H$ is formed by taking G and $\rho(G)$ disjoint copies of H, and identifying each vertex of G other than the root with the root vertex of one of the copies of H. The root of $G \sim H$ is the root of G.

Let $V(G) = \{r, v_1, \ldots, v_{\rho(G)}\}$, where r is the root of G, let E_0 be the edge set of G and let E_i be the edge set of the copy of H attached at v_i. A set F is feasible in $\Gamma(G \sim H)$ if and only if each of the following conditions holds.

1. $F \cap E_0$ is feasible in $\Gamma(G)$.

2. For all i with $1 \leq i \leq \rho(G)$, $F \cap E_i$ is feasible in $\Gamma(H)$.

3. For all i with $1 \leq i \leq \rho(G)$, if v_i is not connected to the root in $G|(F \cap E_0)$ then $F \cap E_i = \emptyset$.

In order to extend these ideas to general greedoids, we first describe the notion of a closed set, which was first described for greedoids by Korte and Lovasz [35]. Let $\Gamma = (E, \rho)$ be a greedoid defined in terms of its rank function. Given a subset A of E, its *closure* $\sigma_\Gamma(A)$ is defined to be $\sigma_\Gamma(A) = \{e : \rho(A \cup e) = \rho(A)\}$. We will drop the dependence on Γ whenever the context is clear. Note that it follows from the definition that $A \subseteq \sigma(A)$. Moreover Lemma 1.3.4 implies that $\rho(\sigma(A)) = \rho(A)$. Furthermore if $e \notin \sigma(A)$, then $\rho(A \cup e) > \rho(A)$, so axiom (GR2) implies that $\rho(\sigma(A) \cup e) > \rho(\sigma(A))$ and hence $\sigma(\sigma(A)) = \sigma(A)$. A subset A of E satisfying $A = \sigma(A)$ is said to be *closed*. Every subset of E of the form $\sigma(X)$ for some X is closed.

We now introduce what we call an attachment function. Let $\Gamma = (E, \mathcal{F})$ be a greedoid with rank $\rho = \rho(\Gamma)$. A function $f : \mathcal{F} \to 2^{[\rho]}$ is called a Γ *attachment function* if it satisfies both of the following.

1. For each feasible set F, we have $|f(F)| = \rho(F)$.

2. If F_1 and F_2 are feasible sets and $F_1 \subseteq \sigma(F_2)$ then $f(F_1) \subseteq f(F_2)$.

These conditions ensure that if F_1 and F_2 are maximal feasible subsets of a set A, then $f(F_1) = f(F_2)$.

Given greedoids Γ_1 and Γ_2 with disjoint element sets, and Γ_1 attachment function f, we define the Γ_2-*attachment of* Γ_1, denoted by $\Gamma_1 \sim_f \Gamma_2$ as follows. The element set E is the union of the element set E_0 of Γ_1 together with $\rho = \rho(\Gamma_1)$ disjoint copies $E_1, \ldots, E_\rho$ of the element set of Γ_2. In the following we shall say that for $i > 0$, a subset of E_i is feasible in Γ_2 if the corresponding subset of the elements of Γ_2 is feasible. A subset F of E is feasible if and only each of the following conditions holds.

1. $F \cap E_0$ is feasible in Γ_1.

2. For all i with $1 \leq i \leq \rho$, $F \cap E_i$ is feasible in Γ_2.

3. For all i with $1 \leq i \leq \rho$, if $i \notin f(F \cap E_0)$ then $F \cap E_i = \emptyset$.

Proposition 2.2.2. *For any greedoids* Γ_1 *and* Γ_2*, and* Γ_1 *attachment function* f*, the* Γ_2-attachment of Γ_1 *is a greedoid.*

Proof. We use the notation defined above to describe the element set of $\Gamma_1 \sim_f \Gamma_2$. Clearly the empty set is feasible in $\Gamma_1 \sim_f \Gamma_2$. Suppose that F_1 and F_2 are feasible sets in $\Gamma_1 \sim_f \Gamma_2$ with $|F_2| > |F_1|$. If there is an element e of $F_2 \cap E_0$ which is not in $\sigma_{\Gamma_1}(F_1 \cap E_0)$ then $(F_1 \cap E_0) \cup e$ is feasible in Γ_1. Moreover $F_1 \cap E_0 \subseteq \sigma_{\Gamma_1}((F_1 \cap E_0) \cup e)$, so $f(F_1 \cap E_0) \subseteq f((F_1 \cap E_0) \cup e)$. Consequently $F_1 \cup e$ is feasible in $\Gamma_1 \sim_f \Gamma_2$.

On the other hand suppose that $F_2 \cap E_0 \subseteq \sigma_{\Gamma_1}(F_1 \cap E_0)$. Then $f(F_2 \cap E_0) \subseteq f(F_1 \cap E_0)$ and for some $i \in f(F_2 \cap E_0)$ we must have $|F_2 \cap E_i| > |F_1 \cap E_i|$. Thus there exists $e \in (F_2 - F_1) \cap E_i$ such that $(F_1 \cap E_i) \cup e$ is feasible in Γ_2. Hence $F_1 \cup e$ is feasible in $\Gamma_1 \sim_f \Gamma_2$. □

Every greedoid Γ has an attachment function formed by setting $f(F) = [|F|]$ for each feasible set F. However there are other examples of attachment functions. Let G be a connected rooted graph in which the vertices other than the root are labelled $v_1, \ldots, v_\rho$. There is an attachment function f defined on $\Gamma(G)$ as follows. For any feasible set F, define $f(F)$ so that $i \in f(F)$ if and only if v_i is connected to the root in the subtree $G|F$. It is straightforward to verify that f is indeed an attachment function. Furthermore if H is another rooted graph then $\Gamma(G \sim_f H) = \Gamma(G) \sim_f \Gamma(H)$.

We now consider the rank function of $\Gamma = \Gamma_1 \sim_f \Gamma_2$. We keep the same notation as above for the elements of Γ. Let A be a subset of $E(\Gamma)$ and let F be a maximal feasible subset of $A \cap E_0$. Then

$$\rho_\Gamma(A) = \rho_{\Gamma_1}(A \cap E_0) + \sum_{i \in f(F)} \rho_{\Gamma_2}(A \cap E_i). \tag{2.3}$$

Observe that the number of subsets of $E(\Gamma)$ with specified rank, size and intersection with E_0 does not depend on the choice of f. Consequently the Tutte polynomial of $\Gamma_1 \sim_f \Gamma_2$ does not depend on f. We now make this idea more precise by establishing an expression for the Tutte polynomial of an attachment.

Theorem 2.2.3. *Let* Γ_1 *and* Γ_2 *be greedoids, and let* f *be an attachment function for* Γ_1*. Then the Tutte polynomial of* $\Gamma_1 \sim_f \Gamma_2$ *is given by*

$$T(\Gamma_1 \sim_f \Gamma_2; x, y) = T(\Gamma_2; x, y)^{\rho(\Gamma_1)} T\Big(\Gamma_1; \frac{(x-1)^{\rho(\Gamma_2)+1} y^{|E(\Gamma_2)|}}{T(\Gamma_2; x, y)} + 1, y\Big),$$

providing $T(\Gamma_2; x, y) \neq 0$.

Proof. Let $\Gamma = \Gamma_1 \sim_f \Gamma_2$. We use the notation defined above to describe the element set of Γ. It is useful to extend the definition of the attachment function f to all subsets of E_0 by setting $f(A)$ to be equal to $f(F)$ where F is a maximal feasible set of A. The definition of an attachment function ensures that this is well-defined. It follows from Equation 2.3 that $\rho(\Gamma) = \rho(\Gamma_1)(\rho(\Gamma_2)+1)$. We have

$$\begin{aligned}
T(\Gamma; x, y) &= \sum_{A \subseteq E(\Gamma)} (x-1)^{\rho(\Gamma)-\rho_\Gamma(A)}(y-1)^{|A|-\rho(A)} \\
&= \sum_{A_0 \subseteq E_0} (x-1)^{\rho(\Gamma_1)-\rho_{\Gamma_1}(A_0)}(y-1)^{|A_0|-\rho_{\Gamma_1}(A_0)} \\
&\quad \cdot \prod_{i \in f(A_0)} \sum_{A_i \subseteq E_i} (x-1)^{\rho(\Gamma_2)-\rho_{\Gamma_2}(A_i)}(y-1)^{|A_i|-\rho_{\Gamma_2}(A_i)} \\
&\quad \cdot \prod_{i \notin f(A_0)} \sum_{A_i \subseteq E_i} (x-1)^{\rho(\Gamma_2)}(y-1)^{|A_i|} \\
&= \sum_{A_0 \subseteq E_0} (x-1)^{\rho(\Gamma_1)-\rho_{\Gamma_1}(A_0)}(T(\Gamma_2; x, y))^{\rho_{\Gamma_1}(A_0)} \\
&\quad \cdot \big((x-1)^{\rho(\Gamma_2)} y^{|E(\Gamma_2)|}\big)^{\rho(\Gamma_1)-\rho_{\Gamma_1}(A_0)}(y-1)^{|A_0|-\rho_{\Gamma_1}(A_0)} \\
&= (T(\Gamma_2; x, y))^{\rho(\Gamma_1)} \sum_{A_0 \subseteq E_0} (y-1)^{|A_0|-\rho_{\Gamma_1}(A_0)} \\
&\quad \cdot \Big(\frac{(x-1)^{\rho(\Gamma_2)+1} y^{|E(\Gamma_2)|}}{T(\Gamma_2; x, y)}\Big)^{\rho(\Gamma_1)-\rho_{\Gamma_1}(A_0)} \\
&= T(\Gamma_2; x, y)^{\rho(\Gamma_1)} T\Big(\Gamma_1; \frac{(x-1)^{\rho(\Gamma_2)+1} y^{|E(\Gamma_2)|}}{T(\Gamma_2; x, y)} + 1, y\Big).
\end{aligned}$$

□

The third construction is called the full rank attachment. Given greedoids $\Gamma_1 = (E_1, \mathcal{F}_1)$ and $\Gamma_2 = (E_2, \mathcal{F}_2)$ with disjoint element sets, the *full rank attachment of* Γ_2 *to* Γ_1 denoted by $\Gamma_1 \approx \Gamma_2$ has element set $E_1 \cup E_2$ and a set F of elements is feasible if either of the two following conditions holds.

1. $F \in \mathcal{F}_1$;
2. $F \cap E_1 \in \mathcal{F}_1$, $F \cap E_2 \in \mathcal{F}_2$ and $\rho_{\Gamma_1}(F \cap E_1) = \rho(\Gamma_1)$.

It is straightforward to prove that $\Gamma_1 \approx \Gamma_2$ is a greedoid.

Suppose that $\Gamma = \Gamma_1 \approx \Gamma_2$ and that A is a subset of $E(\Gamma)$. Then

$$\rho(A) = \begin{cases} \rho(A \cap E(\Gamma_1)) & \text{if } \rho(A \cap E(\Gamma_1)) < \rho(\Gamma_1), \\ \rho(A \cap E(\Gamma_1)) + \rho(A \cap E(\Gamma_2)) & \text{if } \rho(A \cap E(\Gamma_1)) = \rho(\Gamma_1). \end{cases}$$

This observation enables us to prove the following identity for the Tutte polynomial.

Theorem 2.2.4. *Let Γ_1 and Γ_2 be greedoids, and let $\Gamma = \Gamma_1 \approx \Gamma_2$. Let E, E_1 and E_2 denote the element sets of Γ, Γ_1 and Γ_2 respectively. Then*

$$T(\Gamma_1 \approx \Gamma_2; x, y) = T(\Gamma_1; x, y)(x-1)^{\rho(\Gamma_2)} y^{|E_2|} + T(\Gamma_1; 1, y)(T(\Gamma_2; x, y) - (x-1)^{\rho(\Gamma_2)} y^{|E_2|}).$$

Proof. We have

$$\begin{aligned} &T(\Gamma_1 \approx \Gamma_2; x, y) \\ &\quad= \sum_{A \subseteq E} (x-1)^{\rho(\Gamma) - \rho_\Gamma(A)} (y-1)^{|A| - \rho_\Gamma(A)} \\ &\quad= \sum_{\substack{A_1 \subseteq E_1: \\ \rho_{\Gamma_1}(A_1) < \rho(\Gamma_1)}} (x-1)^{\rho(\Gamma_1) - \rho_{\Gamma_1}(A_1)} (y-1)^{|A_1| - \rho_{\Gamma_1}(A_1)} \sum_{A_2 \subseteq E_2} (x-1)^{\rho(\Gamma_2)} (y-1)^{|A_2|} \\ &\qquad+ \sum_{\substack{A_1 \subseteq E_1: \\ \rho_{\Gamma_1}(A_1) = \rho(\Gamma_1)}} (y-1)^{|A_1| - \rho_{\Gamma_1}(A_1)} \sum_{A_2 \subseteq E_2} (x-1)^{\rho(\Gamma_2) - \rho_{\Gamma_2}(A_2)} (y-1)^{|A_2| - \rho_{\Gamma_2}(A_2)} \\ &\quad= \sum_{A_1 \subseteq E_1} (x-1)^{\rho(\Gamma_1) - \rho_{\Gamma_1}(A_1)} (y-1)^{|A_1| - \rho_{\Gamma_1}(A_1)} (x-1)^{\rho(\Gamma_2)} y^{|E_2|} \\ &\qquad+ \sum_{\substack{A_1 \subseteq E_1: \\ \rho_{\Gamma_1}(A_1) = \rho(\Gamma_1)}} (y-1)^{|A_1| - \rho_{\Gamma_1}(A_1)} \\ &\qquad\cdot \Big(\sum_{A_2 \subseteq E_2} (x-1)^{\rho(\Gamma_2) - \rho_{\Gamma_2}(A_2)} (y-1)^{|A_2| - \rho_{\Gamma_2}(A_2)} - (x-1)^{\rho(\Gamma_2)} y^{|E_2|} \Big) \\ &\quad= T(\Gamma_1; x, y)(x-1)^{\rho(\Gamma_2)} y^{|E_2|} + T(\Gamma_1; 1, y)\big(T(\Gamma_2; x, y) - (x-1)^{\rho(\Gamma_2)} y^{|E_2|}\big). \end{aligned}$$

□

One case where this operation will be useful later is when Γ_1 and Γ_2 are binary greedoids with $\Gamma_1 = \Gamma(M_1)$ and $\Gamma_2 = \Gamma(M_2)$ where M_1 has full row rank. Then $\Gamma_1 \approx \Gamma_2 = \Gamma(M)$ where M has the form

$$M = \left(\begin{array}{c|c} M_1 & 0 \\ \hline 0 & M_2 \end{array} \right).$$

2.3 Rooted Graphs

2.3.1 Introduction

In this section we restrict ourselves to rooted graphs that are connected. The main result of this section is stated below.

Theorem 2.3.1. *Evaluating the Tutte polynomial of a connected rooted graph at any fixed point (a, b) in the rational xy-plane is #P-hard apart from when (a, b) equals $(1, 1)$ or when (a, b) lies on H_1. In these exceptional cases we can evaluate the Tutte polynomial in polynomial time.*

Our results concern three computational problems. The most general of these is to compute the Tutte polynomial of a connected rooted graph. We present the problem below in the standard format of complexity theory. Let $\mathcal{G}$ be the class of connected rooted graphs.

$\pi_1[\mathcal{G}]$: #ROOTED TUTTE POLYNOMIAL
Input $G \in \mathcal{G}$.
Output The coefficients of $T(G; x, y)$.

The second problem is the evaluation of the Tutte polynomial of a connected rooted graph at a fixed point (a, b) such that $a, b \in \mathbb{Q}$. Due to computational problems caused by representing real numbers in a computer, we are unable to allow a, b to be arbitrary reals. For simplicity we stick with rationals.

$\pi_2[\mathcal{G}, a, b]$: #ROOTED TUTTE POLYNOMIAL AT (a, b)
Input $G \in \mathcal{G}$.
Output $T(G; a, b)$.

Finally we consider the computational problem of evaluating the Tutte polynomial of a connected rooted graph along a curve L in the xy-plane. We restrict our attention to the case where L is a rational curve given by the parametric equations

$$x(t) = \frac{p(t)}{q(t)} \quad \text{and} \quad y(t) = \frac{r(t)}{s(t)}$$

of L where p, q, r and s are polynomials over $\mathbb{Q}$.

$\pi_3[\mathcal{G}, L]$: #ROOTED TUTTE POLYNOMIAL ALONG L

Input $G \in \mathcal{G}$.

Output The coefficients of the rational function of t given by evaluating $T(G; x(t), y(t))$.

It should be straightforward to see that

$$\pi_2[\mathcal{G}, a, b] \propto_T \pi_3[\mathcal{G}, H_{(a-1)(b-1)}] \propto_T \pi_1[\mathcal{G}].$$

Our results will determine when the opposite reductions hold. We now restate Theorem 2.3.1 in terms of the computational problem $\pi_2[\mathcal{G}, a, b]$.

Theorem 2.3.2. *The problem $\pi_2[\mathcal{G}, a, b]$ is #P-hard for all (a, b) except when (a, b) equals $(1, 1)$ or when (a, b) lies on H_1. In each of these exceptional cases $\pi_2[\mathcal{G}, a, b]$ is easy.*

The remainder of this section will focus on proving Theorem 2.3.2.

2.3.2 Proof of Main Theorem

Here we begin by reviewing the exceptional points of Theorem 2.3.2. We first present a simple proof to show that evaluating the Tutte polynomial of a rooted graph along the hyperbola H_1 can be done in polynomial time. This is consistent with the situation in unrooted graphs.

If a point (a, b) lies on the hyperbola H_1 then we have $(a-1)(b-1) = 1$ by definition. Thus the

Tutte polynomial of a rooted graph G evaluated at such a point is given by

$$\begin{aligned}
T(G;a,b) &= \sum_{A\subseteq E}(a-1)^{\rho(G)-\rho(A)}(b-1)^{|A|-\rho(A)}\\
&= (a-1)^{\rho(G)}\sum_{A\subseteq E}\left(\frac{1}{a-1}\right)^{|A|}\\
&= (a-1)^{\rho(G)}\sum_{k=0}^{|E|}\binom{|E|}{k}\left(\frac{1}{a-1}\right)^{k}\\
&= (a-1)^{\rho(G)-|E|}a^{|E|},
\end{aligned}$$

which is easily computed.

By the previous chapter the Tutte polynomial of a connected rooted graph coincides with that of the corresponding connected unrooted graph along the line $x = 1$. Consequently the classifications of the complexity of the points $(1, y)$ also coincide. Evaluating the Tutte polynomial of a graph at the point $(1, 1)$ is easy by Theorem 1.6.1. Therefore the problem $\pi_2[\mathcal{G}, 1, 1]$ is easy.

Theorem 2.2.1 implies that for any $k \in \mathbb{N}$ computing $T(\Gamma^{2k}; x, -1)$ is easy for an arbitrary greedoid Γ provided its rank is specified or can be easily computed. That is, the Tutte polynomial of a greedoid in which every parallel class has an even number of elements is easy to evaluate along the line $y = -1$.

Note that we have $\Gamma(G^k) = (\Gamma(G))^k$, that is, the thickening operations in rooted graphs and greedoids are compatible. The k-thickening G^k of a rooted graph G is obtained by replacing every edge in G by k parallel edges. An example of the k-thickening operation on a rooted graph is given in Figure 2.1. By Theorem 2.2.1 the effect of the k-thickening operation on the Tutte polynomial of a rooted graph is given in the following lemma.

Lemma 2.3.3. *Let G be a rooted graph. The Tutte polynomial of the k-thickening G^k of G when $y \neq -1$ is given by*

$$T(G^k; x, y) = (1 + y + \ldots + y^{k-1})^{\rho(G)}T\left(G; \frac{x + y + \ldots + y^{k-1}}{1 + y + \ldots + y^{k-1}}, y^k\right). \tag{2.4}$$

When $y = -1$ we have

$$T(G^k; x, -1) = \begin{cases} (x-1)^{\rho(G)} & \textit{if } k \textit{ is even;} \\ T(G; x, -1) & \textit{if } k \textit{ is odd.} \end{cases}$$

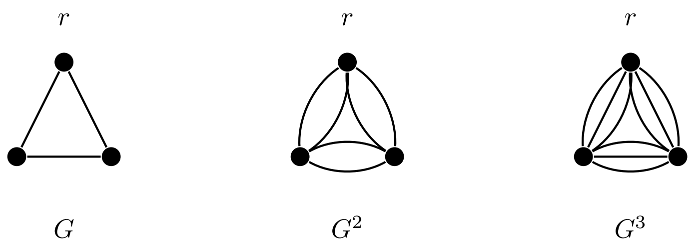

Figure 2.1: Example of the 2 and 3-thickening operation on G.

An example of the H-attachment operation on a connected rooted graph G is given in Figure 2.2. By Theorem 2.2.3 the effect of the H-attachment operation on the Tutte polynomial of a connected rooted graph is given in the following lemma.

Lemma 2.3.4. *Let $G = (r, V, E)$ and $H = (r', V', E')$ be disjoint rooted graphs with G connected. The Tutte polynomial of the H-attachment of G is given by*

$$T(G \sim H; x, y) = T(H; x, y)^{\rho(G)} T\left(G; \frac{(x-1)^{\rho(H)+1} y^{|E'|}}{T(H; x, y)} + 1, y\right),$$

providing $T(H; x, y) \neq 0$.

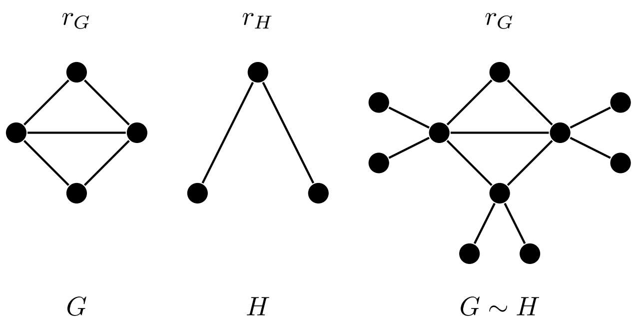

Figure 2.2: Example of the H-attachment operation on G.

For an arbitrary rooted graph G, the k-stretch G_k of G can be constructed in exactly the same way as in an unrooted graph. However we are unable to express the Tutte polynomial of G_k in terms of the Tutte polynomial of G because we cannot easily express the rank of a set of edges in G_k in terms of the corresponding set in G.

We will now review the hard points of Theorem 2.3.2. We begin by stating the following definition from linear algebra which we have generalized slightly compared with its standard form.

Definition 2.3.5 (Vandermonde Matrix $\mathbf{V}$)**.** A *Vandermonde* matrix is an $m \times n$ matrix $\mathbf{V}$ such that $\mathbf{V}_{i,j} = y_i z_i^{j-1}$, i.e. the terms in each row give a geometric progression. When $\mathbf{V}$ is a square matrix, the *determinant* $det(\mathbf{V})$ of $\mathbf{V}$ can be expressed as

$$\det(V) = \prod_{i=1}^{m} y_i \prod_{1 \le i < j \le m} (z_j - z_i).$$

It should be clear that if $z_i \neq z_j$ whenever $i \neq j$ and $y_i \neq 0$ for all i, then $\det(\mathbf{V}) \neq 0$. The determinant result is demonstrated in the following example.

Example 2.3.6. *When $\boldsymbol{V}$ is a 3×3 Vandermonde matrix we have*

$$\boldsymbol{V} = \begin{pmatrix} y_1 & y_1 z_1 & y_1 z_1^2 \\ y_2 & y_2 z_2 & y_2 z_2^2 \\ y_3 & y_3 z_3 & y_3 z_3^2 \end{pmatrix}.$$

Therefore

$$\begin{aligned} \det \boldsymbol{V} = y_1 y_2 y_3 \det \begin{pmatrix} 1 & z_1 & z_1^2 \\ 1 & z_2 & z_2^2 \\ 1 & z_3 & z_3^2 \end{pmatrix} \\ &= \prod_{i=1}^{3} y_i \left[(z_2 z_3^2 - z_2^2 z_3) - (z_1 z_3^2 - z_1^2 z_3) + (z_1 z_2^2 - z_1^2 z_2) \right] \\ &= \prod_{i=1}^{3} y_i \left[(z_3 - z_2)(z_3 - z_1)(z_2 - z_1) \right] \\ &= \prod_{i=1}^{3} y_i \prod_{1 \le i < j \le 3} (z_j - z_i). \end{aligned}$$

We now present three propositions which together show that at most fixed rational points (a, b), evaluating the Tutte polynomial of a connected rooted graph at (a, b) is just as hard as evaluating it along the curve $H_{(a-1)(b-1)}$. The first proposition considers the case when $a \neq 1$ and $b \notin \{-1, 0, 1\}$.

Proposition 2.3.7. *Let $L = H_\alpha$ for some $\alpha \in \mathbb{Q} - \{0\}$. Let $(a, b) \in L$ such that $b \notin \{-1, 0\}$. Then*

$$\pi_3[\mathcal{G}, L] \propto_T \pi_2[\mathcal{G}, a, b].$$

Proof. For a point $(x, y) \in L$ we have $y \neq 1$. Therefore $z = y - 1 \neq 0$ and so $\alpha/z = x - 1$. Let $G \in \mathcal{G}$. Along L the Tutte polynomial of G has the form

$$T(G; x, y) = T(G; 1 + \alpha/z, 1 + z) = \sum_{A \subseteq E} \left(\frac{\alpha}{z}\right)^{\rho(G)-\rho(A)} z^{|A|-\rho(A)} = \sum_{i=-\rho(G)}^{|E|} t_i z^i,$$

for some $t_{-\rho(G)}, \ldots, t_{|E|}$.

We will now show that we may determine all of the coefficients t_i in polynomial time from $T(G^k; a, b)$ for $k = 1, \ldots, |E| + \rho(G) + 1$. By Lemma 2.3.3 we have

$$T(G^k; a, b) = (1 + b + \ldots + b^{k-1})^{\rho(G)} T\left(G; \frac{a + b + \ldots + b^{k-1}}{1 + b + \ldots + b^{k-1}}, b^k\right).$$

Since $b \neq -1$ we have $1 + b + \ldots + b^{k-1} \neq 0$. Therefore we may compute $T\left(G; \frac{a+b+\ldots+b^{k-1}}{1+b+\ldots+b^{k-1}}, b^k\right)$ from $T(G^k; a, b)$. The point $\left(\frac{a+b+\ldots+b^{k-1}}{1+b+\ldots+b^{k-1}}, b^k\right)$ will also be on the curve L since

$$\left(\frac{a + b + \ldots + b^{k-1}}{1 + b + \ldots + b^{k-1}} - 1\right)(b^k - 1) = (a - 1)(b - 1).$$

In order to evaluate the one-variable Tutte polynomial of G along the curve L, we need the points $\left(\frac{a+b+\ldots+b^{k-1}}{1+b+\ldots+b^{k-1}}, b^k\right)$ to be pairwise distinct for $k = 1, 2, \ldots, |E| + \rho(G) + 1$. Since $b \notin \{-1, 0, 1\}$ we have b^k distinct for $k = 1, 2, \ldots, |E| + \rho(G) + 1$ and so all of the points must be pairwise distinct, regardless of a.

Therefore by evaluating $T(G^k; a, b)$ for $k = 1, 2, \ldots, |E| + \rho(G) + 1$ where $b \notin \{-1, 0, 1\}$ we obtain $\sum_{i=-\rho(G)}^{|E|} t_i z^i$ for $|E| + \rho(G) + 1$ distinct values of z. Denote these distinct values of z by z_k for $1 \leq k \leq |E| + \rho(G) + 1$. This gives us $|E| + \rho(G) + 1$ linear equations for the coefficients t_i. Let $T_k = T(G; 1 + \alpha/z_k, 1 + z_k)$ and $\underline{A} = (T_1\ T_2\ \ldots\ T_{|E|+\rho(G)}\ T_{|E|+\rho(G)+1})^T$. Now if $\underline{t} = (t_{-\rho(G)}\ t_{-\rho(G)+1}\ \ldots\ t_{|E|-1}\ t_{|E|})^T$ these linear equations can be represented in matrix form by $\underline{A} = \mathbf{V}\underline{t}$ where $\mathbf{V}$ is a Vandermonde matrix with $\mathbf{V}_{k,j} = z_k^{-\rho(G)+j-1}$ for $1 \leq k, j \leq |E| + \rho(G) + 1$. Since all z_k are distinct and non-zero, $\mathbf{V}$ is non-singular and we can solve $\underline{A} = \mathbf{V}\underline{t}$ to obtain the coefficients t_i in polynomial time [15]. $\square$

The next proposition considers the case when $a = 1$. Recall $H_0^x = \{(1, y) : y \in \mathbb{Q}\}$ and $H_0^y = \{(x, 1) : x \in \mathbb{Q}\}$.

Proposition 2.3.8. *Let $L = H_0^x$ and let $b \in \mathbb{Q} - \{-1, 0, 1\}$. Then*

$$\pi_3[\mathcal{G}, L] \propto_T \pi_2[\mathcal{G}, 1, b].$$

Proof. For a point $(x, y) \in L$ we have $x = 1$. Let $G \in \mathcal{G}$. Along L the Tutte polynomial of G has the form

$$T(G; 1, y) = \sum_{\substack{A \subseteq E: \\ \rho(A) = \rho(G)}} (y - 1)^{|A| - \rho(G)} = \sum_{i=-\rho(G)}^{|E|} t_i y^i,$$

for some $t_{-\rho(G)}, \ldots, t_{|E|}$.

The proof now follows in a similar way to that of Proposition 2.3.7 by determining all of the coefficients t_i in polynomial time from $T(G^k; 1, b)$ for $k = 1, 2, \ldots, |E| + \rho(G) + 1$. $\square$

The following proposition considers the case when $b = 1$.

Proposition 2.3.9. *Let $L = H_0^y$ and $a \in \mathbb{Q} - \{1\}$. Then*

$$\pi_3[\mathcal{G}, L] \propto_T \pi_2[\mathcal{G}, a, 1].$$

Proof. For a point $(x, y) \in L$ we have $y = 1$. Let $G \in \mathcal{G}$. Along L the Tutte polynomial of G has the form

$$T(G; x, 1) = \sum_{\substack{A \subseteq E: \\ \rho(A) = |A|}} (x - 1)^{\rho(G) - \rho(A)} = \sum_{i=0}^{\rho(G)} t_i x^i,$$

for some $t_0, \ldots, t_{\rho(G)}$.

We will now show that we may determine all of the coefficients t_i in polynomial time from $T(G^k; a, 1)$ for $k = 1, 2, \ldots, \rho(G) + 1$. By Lemma 2.3.3 we have

$$T(G^k; a, 1) = k^{\rho(G)} T\left(G; \frac{a + k - 1}{k}, 1\right).$$

Therefore we may compute $T\left(G; \frac{a+k-1}{k}, 1\right)$ from $T(G^k; a, 1)$. Since $a \neq 1$, the points $\left(\frac{a+k-1}{k}, 1\right)$ are pairwise distinct for $k = 1, 2, \ldots, \rho(G)+1$. By evaluating $T(G^k; a, 1)$ for $k = 1, 2, \ldots, \rho(G)+1$ where

$a \neq 1$ we obtain $\sum_{i=0}^{\rho(G)} t_i x^i$ for $\rho(G)+1$ distinct values of x. This gives us $\rho(G)+1$ linear equations for the coefficients t_i. Again the matrix corresponding to these equations is a Vandermonde matrix with non-zero entries, therefore we may recover the coefficients t_i in polynomial time. $\square$

We now summarize the preceding propositions.

Proposition 2.3.10. *Let L be either H_0^x, H_0^y, or H_α for $\alpha \in \mathbb{Q} - \{0\}$. Let $(a, b) \in L$ such that $(a, b) \neq (1, 1)$ and $b \notin \{-1, 0\}$. Then*

$$\pi_3[\mathcal{G}, L] \propto_T \pi_2[\mathcal{G}, a, b].$$

Proof. Follows directly from Propositions 2.3.7, 2.3.8 and 2.3.9. $\square$

We now consider the exceptional case when $b = -1$. For reasons that will soon become apparent, we recall $T(P_2; x, y) = x^2y - 2xy + x + y$ and $T(S_k; x, y) = x^k$ from Propositions 1.7.2 and 1.7.3 respectively.

Proposition 2.3.11. *Let L be the line $y = -1$. For $a \notin \{\frac{1}{2}, 1\}$ we have*

$$\pi_3[\mathcal{G}, L] \propto_T \pi_2[\mathcal{G}, a, -1].$$

Proof. Let $G = (r_G, V_G, E_G)$ be a rooted graph and let $z = x - 1$. Along L the Tutte polynomial of G has the form

$$T(G; x, -1) = \sum_{A \subseteq E_G} z^{\rho(G)-\rho(A)}(-2)^{|A|-\rho(A)} = \sum_{i=0}^{\rho(G)} t_i z^i$$

for some $t_0, \ldots, t_{\rho(G)}$.

We will now show that we may determine all of the coefficients t_i in polynomial time from $T(G \sim S_k; a, -1)$ for $k = 0, 1, \ldots, \rho(G)$, apart from at a few exceptional values of a. By Lemma 2.3.4 we have

$$T(G \sim S_k; a, -1) = a^{k\rho(G)} T\left(G; \frac{(a-1)^{k+1}(-1)^k}{a^k} + 1, -1\right).$$

Providing $a \neq 0$ we may compute $T\left(G; \frac{(a-1)^{k+1}(-1)^k}{a^k} + 1, -1\right)$ from $T(G \sim S_k; a, -1)$. For $a \notin \{\frac{1}{2}, 1\}$ we claim that the points $\left(\frac{(a-1)^{k+1}(-1)^k}{a^k} + 1, -1\right)$ are pairwise distinct for $k = 0, 1, 2, \ldots, \rho(G)$.

Suppose otherwise, i.e. that for some $m \neq n$ we have

$$\frac{(a-1)^{m+1}(-1)^m}{a^m} + 1 = \frac{(a-1)^{n+1}(-1)^n}{a^n} + 1.$$

This can be reduced to

$$\left(\frac{1-a}{a}\right)^{m-n} = \pm 1.$$

The case $\left(\frac{1-a}{a}\right)^{m-n} = -1$ is impossible, and it should be clear to see that $\left(\frac{1-a}{a}\right)^{m-n} = 1$ is only satisfied when $a = \frac{1}{2}$ or $m = n$ (or both).

Therefore by evaluating $T(G \sim S_k; a, -1)$ for $k = 0, 1, 2, \ldots, \rho(G)$ where $a \notin \{0, \frac{1}{2}, 1\}$, we obtain $\sum_{i=0}^{\rho(G)} t_i z^i$ for $\rho(G)+1$ distinct values of z. This gives us $\rho(G)+1$ linear equations for the coefficients t_i. Again the matrix corresponding to these equations is a Vandermonde matrix with non-zero entries, and so the coefficients may be recovered in polynomial time. Hence evaluating the Tutte polynomial of a connected rooted graph along the line $y = -1$ is Turing reducible to evaluating it at a point $(a, -1)$ for $a \notin \{0, \frac{1}{2}, 1\}$.

We now look at the case when $a = 0$. Note that $T(P_2; 0, -1) = -1$. Applying Lemma 2.3.4 to G and P_2 gives

$$\begin{aligned} T(G \sim P_2; 0, -1) &= (-1)^{\rho(G)} T\left(G; \frac{(-1)^3(-1)^2}{-1} + 1, -1\right) \\ &= (-1)^{\rho(G)} T(G; 2, -1). \end{aligned}$$

Therefore we have the reductions

$$\pi_3[\mathcal{G}, L] \propto_T \pi_2[\mathcal{G}, 2, -1] \propto_T \pi_2[\mathcal{G}, 0, -1].$$

Since the Turing reduction relation is transitive, this implies that evaluating the Tutte polynomial at the point $(0, -1)$ is at least as hard as evaluating it along the line $y = -1$. This completes the proof. $\square$

We now begin to classify the complexity of evaluating the Tutte polynomial of a connected rooted graph.

Proposition 2.3.12. *The computational problem $\pi_2[\mathcal{G}, 1, b]$ is #P-hard apart from when $b = 1$, in* *which case it is easy.*

Proof. This follows directly from Theorem 1.6.1 and the equivalence of evaluating the Tutte polynomial of a connected rooted graph and the Tutte polynomial of the corresponding connected unrooted graph along $x = 1$. $\square$

The next results will establish hardness for a few special cases, namely when $b \in \{-1, 0, 1\}$.

Proposition 2.3.13. *The computational problem $\pi_2[\mathcal{G}, a, -1]$ is #P-hard apart from when $a = 1/2$, in which case it is easy.*

Proof. First note that $(\frac{1}{2}, -1)$ is easy since it lies on H_1. Now let L be the line $y = -1$. By Proposition 2.3.11 we have

$$\pi_3[\mathcal{G}, L] \propto_T \pi_2[\mathcal{G}, a, -1]$$

for $a \notin \{\frac{1}{2}, 1\}$. This implies

$$\pi_2[\mathcal{G}, 1, -1] \propto_T \pi_2[\mathcal{G}, a, -1]$$

for $a \neq 1/2$. By Proposition 2.3.12 we know that $\pi_2[\mathcal{G}, 1, -1]$ is #P-hard. $\square$

Proposition 2.3.14. *The computational problem $\pi_2[\mathcal{G}, a, 0]$ is #P-hard apart from when $a = 0$, in which case it is easy.*

Proof. Let $G \in \mathcal{G}$. First note that evaluating the Tutte polynomial of G at the point $(0, 0)$ is easy since it lies on the hyperbola H_1. Applying Lemma 2.3.4 to G and S_k gives

$$T(G \sim S_k; a, 0) = a^{k\rho(G)} T(G; 1, 0).$$

Since $a \neq 0$ we may compute $T(G; 1, 0)$ from $T(G \sim S_k; a, 0)$. Therefore evaluating the Tutte polynomial of a connected rooted graph at any point on the line $y = 0$, apart from at $(0, 0)$, is just as hard as evaluating it at the point $(1, 0)$, which is #P-hard by Proposition 2.3.12. $\square$

Recall from Equation 1.6 that along $y = 0$ the Tutte polynomial of a rooted graph specializes to the characteristic polynomial. Therefore we have the following corollary.

Corollary 2.3.15. *Computing the characteristic polynomial $p(G; k)$ of a connected rooted graph G is #P-hard for all $k \in \mathbb{Q} - \{1\}$. When $k = 1$ the computation is easy.*

Proof. Let $k \in \mathbb{Q}$. Evaluating the characteristic polynomial of G when $\lambda = k$ gives

$$p(G;k) = (-1)^{\rho(G)}T(G;1-k,0).$$

By Proposition 2.3.14 evaluating $T(G;1-k,0)$ is #P-hard providing $k \neq 1$. Furthermore when $k = 1$ we have

$$p(G;1) = (-1)^{\rho(G)}T(G;0,0) = \begin{cases} 1 & \text{if } G \text{ is empty;} \\ 0 & \text{otherwise,} \end{cases}$$

and so it is easy to compute (as expected since $(0,0)$ lies on H_1). □

The following proposition determines the complexity of evaluating the Tutte polynomial of a connected rooted graph along the line $y = 1$. First we need to state a computational problem for finding the number of subtrees of an unrooted graph. Let $\overline{\mathcal{G}}$ be the class of connected unrooted graphs.

#SUBTREES
Input Graph $G \in \overline{\mathcal{G}}$.
Output The number of subtrees of G.

Jerrum considered the complexity of this problem in [32] and showed it to be #P-complete, settling an open problem of Valiant. He proves this result by reduction from the computational problem stated below.

#CUBICHAM
Input A cubic graph $G \in \overline{\mathcal{G}}$.
Output The number of Hamiltonian paths of G.

Jerrum also shows in [32] that #SUBTREES remains #P-complete when the input graph is restricted to being planar.

Proposition 2.3.16. *The computational problem $\pi_2[\mathcal{G}, a, 1]$ is #P-hard when $a \neq 1$.*

Proof. Let G be a connected unrooted graph with $V(G) = \{v_1, \ldots, v_t\}$. Now let G_j be the connected rooted graph obtained from G by choosing v_j to be the root where $1 \leq j \leq t$. Let ρ_j denote the rank function of G_j and $a_i(G_j)$ be the number of rooted subtrees of G_j with i edges. Then

$$\begin{aligned} T(G_j;x,1) &= \sum_{\substack{A\subseteq E:\\ \rho_j(A)=|A|}} (x-1)^{\rho(G_j)-|A|} = \sum_{\substack{A\subseteq E:\\ G_j|A \text{ is a rooted subtree}}} (x-1)^{\rho(G_j)-|A|} \\ &= \sum_{i=0}^{\rho(G_j)} a_i(G_j)(x-1)^{\rho(G_j)-i}. \end{aligned}$$

Let $a_i(G)$ denote the number of subtrees of G with i edges. Then

$$a_i(G) = \sum_{j=1}^{t} \frac{a_i(G_j)}{i+1}.$$

This is because every subtree T of G with $i > 0$ edges has $i+1$ vertices and corresponds to a rooted subtree in any of the rooted graphs where one of these $i+1$ vertices is the root. There are $|V(G)|$ subtrees with no edges and each of the rooted graphs has exactly one rooted subtree with no edges.

Therefore if we compute $a_i(G_j)$ for all j, then we can compute $a_i(G)$ in polynomial time. If we do this for all i where $0 \leq i \leq \rho(G)$ then we can recover the total number of subtrees of G in polynomial time.

Hence evaluating the Tutte polynomial of a connected rooted graph along the line $y = 1$ is at least as hard as counting the number of subtrees of the corresponding connected unrooted graph, i.e.

$$\#\text{SUBTREES} \propto_T \pi_3[\mathcal{G}, H_0^y].$$

By Proposition 2.3.10 we have

$$\#\text{SUBTREES} \propto_T \pi_3[\mathcal{G}, H_0^y] \propto_T \pi_2[\mathcal{G}, a, 1]$$

for $a \neq 1$. The transitivity of the reduction relation implies $\pi_2[\mathcal{G}, a, 1]$ is #P-hard for $a \neq 1$ by Jerrum's result. $\square$

We now summarize our results and prove Theorem 2.3.2.

Proof of Theorem 2.3.2. Let $(a,b) \in H_\alpha$ for some $\alpha \in \mathbb{Q} - \{0,1\}$. By Proposition 2.3.10 we have $\pi_3[\mathcal{G}, H_\alpha] \propto_T \pi_2[\mathcal{G}, a, b]$ providing $(a,b) \neq (1,1)$ and $b \notin \{-1,0\}$. The hyperbola H_α crosses the x-axis at the point $(1-\alpha, 0)$. By Proposition 2.3.14 the problem $\pi_2[\mathcal{G}, 1-\alpha, 0]$ is #P-hard to compute since $\alpha \neq 1$. This gives us a #P-hard point on each of these curves and therefore implies $\pi_3[\mathcal{G}, H_\alpha]$ is #P-hard to compute for $\alpha \in \mathbb{Q} - \{0, 1\}$. Hence $\pi_2[\mathcal{G}, a, b]$ is #P-hard for $(a, b) \in H_\alpha$ with $\alpha \in \mathbb{Q} - \{0, 1\}$ and $b \neq -1$. The rest of the proof now follows directly by Propositions 2.3.12, 2.3.13 and 2.3.16, and the discussion concerning the easy points at the beginning of the subsection. □

Following Vertigan and Welsh we now strengthen our result by restricting the class of input connected rooted graphs to be planar bipartite. By Theorem 1.6.2 the complexity of computing the Tutte polynomial of a connected planar bipartite graph along the line $x = 1$ is the same as that of an arbitrary graph. Note that all of the reductions we have used preserve the property of being planar and bipartite. It is therefore straightforward to see that we can find an analogous proof to Theorem 2.3.2 for connected planar bipartite rooted graphs, apart from when $b = 1$ which is not so obvious.

Along $y = 1$ we need extra consideration because we use Jerrum's result to classify the complexity along this line. In [32] Jerrum shows that the problem of counting the number of subtrees of a connected graph G remains #P-complete when the input graph is restricted to being planar. By using this we are only able to classify the complexity of evaluating the Tutte polynomial of connected planar rooted graphs along $y = 1$ as opposed to planar bipartite. We now aim to show that Jerrum's result can in fact be strengthened to connected planar bipartite graphs, allowing us to similarly strengthen our result along $y = 1$.

First we state a computational problem with the intention of showing #SUBTREES reduces to it. Let $\overline{\mathcal{B}}$ be the class of bipartite connected graphs.

#BISUBTREES
Input Bipartite graph $G \in \overline{\mathcal{B}}$.
Output The number of subtrees of G.

We say that an edge of a graph G is *external* in a subtree T if it is not contained in $E(T)$. Let $t_{i,j}(G)$ be the number of subtrees of G with i external edges having precisely one endpoint in the tree, and j external edges having both endpoints in the tree.

The following proposition gives a formula for counting the number of subtrees of the k-stretched graph G_k in terms of $t_{i,j}(G)$. Let $t(G_k)$ denote the number of subtrees of G_k.

Proposition 2.3.17. *The number of subtrees of the k-stretched graph G_k is given by*

$$t(G_k) = \left(\sum_{i,j \geq 0} t_{i,j}(G) k^i \binom{k+1}{2}^j \right) + \frac{k(k-1)|E|}{2}. \tag{2.5}$$

Proof. Let $E(G) = \{e_1, e_2, \ldots, e_m\}$. Let E_t be the set of edges replacing e_t in G_k for $1 \leq t \leq m$, thus $E(G_k) = \bigcup_t E_t$. Let f be the function mapping a subtree T in G_k to a subtree T' in G such that $V(T') = V(T) \cap V(G)$ and $e_t \in E(T')$ if and only if $E_t \subseteq E(T)$. Note that f may map a subtree to a graph with no vertices or edges. We will count such a graph as a subtree for the moment.

Let $T'_{i,j}$ be a subtree of G with at least one vertex, i external edges having precisely one endpoint in $T'_{i,j}$ and j external edges having both endpoints in $T'_{i,j}$.

If $T \in f^{-1}(T'_{i,j})$ then it must contain all of the edges in G_k that replace the edges in $E(T'_{i,j})$.

Suppose there exists an edge $e_t = v_1v_2$ in G that is external in $T'_{i,j}$ with $v_1 \in V(T'_{i,j})$ and $v_2 \notin V(T'_{i,j})$. There are k choices for the proper subsets of E_t in G_k such that T is a subtree (k possible paths of edges in E_t connected to v_1 in G_k).

Now suppose there exists an edge $e_t = v_1v_2$ in G that is external in $T'_{i,j}$ with $v_1, v_2 \in V(T'_{i,j})$. There are $\binom{k+1}{2}$ choices for the proper subsets of E_t in G_k such that T is a subtree ($\binom{k+1}{2}$ ways of choosing possible paths of edges in E_t connected to v_1, v_2 or both, providing they don't meet, in G_k). Therefore we have

$$|f^{-1}(T'_{i,j})| = k^i \binom{k+1}{2}^j.$$

It remains to count the subtrees of G_k mapped by f to a graph with no vertices. Such a subtree T satisfies $V(T) \cap V(G) = \emptyset$. There are $(k-1)|E(G)|$ subtrees of G_k comprising a single vertex not in $V(G)$, and $\binom{k-1}{2}|E(G)|$ subtrees of G_k with at least one edge not containing any vertex in $V(G)$. Hence

$$t(G_k) = \left(\sum_{i,j \geq 0} t_{i,j}(G) k^i \binom{k+1}{2}^j \right) + \frac{k(k-1)}{2}|E(G)|.$$

□

We have $\max_{i,j \geq 0}\{i + 2j : t_{i,j}(G) > 0\} \leq \max_{i,j \geq 0}\{i + 2j : i + j \leq |E(G)|\} = 2|E(G)|$. Therefore $t(G_k)$ is a polynomial of degree at most $2|E(G)|$. So we can write

$$t(G_k) = \sum_{p=0}^{2|E(G)|} a_p k^p.$$

Thus if we compute $t(G_k)$ for $2|E(G)|+1$ distinct values of k we can recover the polynomial and hence $t(G_1) = t(G)$ which is the number of subtrees of G. Note that $G_2, \dots, G_{4|E(G)|+2}$ are all bipartite. Therefore we have the required reduction

$$\#\text{SUBTREES} \propto_T \#\text{BISUBTREES}.$$

We have shown that counting the number of subtrees of a connected planar bipartite graph is #P-complete. Therefore we are able to determine the complexity along $y = 1$ when we restrict ourselves to connected planar bipartite graphs. The following theorem concludes this section and is a strengthening of Theorem 2.3.2.

Theorem 2.3.18. *Evaluating the Tutte polynomial of a connected planar bipartite rooted graph at any fixed rational point (a, b) in the xy-plane is #P-hard apart from when (a, b) equals $(1, 1)$ or when (a, b) lies on the hyperbola H_1. In these exceptional cases it is easy.*

2.4 Rooted Digraphs

We present analogous results to those in the previous section by finding the computational complexity of evaluating the Tutte polynomial of a rooted digraph at a fixed rational point. We say that a rooted digraph is *root connected* if every vertex is reachable by a directed path from the root. In this section we mainly restrict our attention to root connected digraphs.

The k thickening D^k of a root connected digraph D is obtained by replacing every edge e in D by k parallel edges that have the same orientation as e. Theorem 2.2.1 can be specialized to root connected digraphs in the following way.

Lemma 2.4.1. *Let D be a root connected digraph. The Tutte polynomial of the k-thickening D^k of D when $y \neq -1$ is given by*

$$T(D^k; x, y) = (1 + y + \dots + y^{k-1})^{\rho(D)} T\left(D; \frac{x + y + \dots + y^{k-1}}{1 + y + \dots + y^{k-1}}, y^k\right). \tag{2.6}$$

When $y = -1$ we have

$$T(D^k; x, -1) = \begin{cases} (x-1)^{\rho(D)} & \text{if } k \text{ is even;} \\ T(D; x, -1) & \text{if } k \text{ is odd.} \end{cases}$$

By Theorem 2.2.3 the effect of the H-attachment operation on the Tutte polynomial of a root connected graph is given in the following lemma.

Lemma 2.4.2. *Let $D = (r, V, \overrightarrow{E})$ and $H = (r', V', \overrightarrow{E}')$ be disjoint rooted digraphs with D root connected. The Tutte polynomial of the H-attachment of D is given by*

$$T(D \sim H; x, y) = T(H; x, y)^{\rho(D)} T\left(D; \frac{(x-1)^{\rho(H)+1} y^{|\overrightarrow{E}'|}}{T(H; x, y)} + 1, y\right),$$

providing $T(H; x, y) \neq 0$.

We now consider the computational complexity of evaluating the Tutte polynomial of a root connected digraph at a fixed point in the rational xy-plane. In parallel with Section 2.3 we draw our attention to the following three computational problems. Let $\mathcal{D}$ denote the class of all root connected digraphs, let $a, b \in \mathbb{Q}$ and let $x(t) = p(t)/q(t), y(t) = r(t)/s(t)$ be parametric equations of L where p, q, r and s are polynomials over $\mathbb{Q}$.

$\pi_4[\mathcal{D}]$: #ROOTED DIRECTED TUTTE POLYNOMIAL
Input $D \in \mathcal{D}$.
Output The coefficients of $T(D; x, y)$.

$\pi_5[\mathcal{D}, a, b]$: #ROOTED DIRECTED TUTTE POLYNOMIAL AT (a, b)
Input $D \in \mathcal{D}$.
Output $T(D; a, b)$.

$\pi_6[\mathcal{D}, L]$: #ROOTED DIRECTED TUTTE POLYNOMIAL ALONG L
Input $D \in \mathcal{D}$.
Output The coefficients of the rational function of t given by evaluating $T(D; x(t), y(t))$.

The main result from this section is as follows.

Theorem 2.4.3. *The problem $\pi_5[\mathcal{D}, a, b]$ is #P-hard for all (a, b) except when (a, b) equals $(1, 1)$, when (a, b) lies on H_1, or when $b = 0$. In these exceptional cases $\pi_5[\mathcal{D}, a, b]$ is easy.*

The proof of the following proposition is analogous to that of Proposition 2.3.10, thus we omit it from this section.

Proposition 2.4.4. *Let L be either H_0^x, H_0^y, or H_α for $\alpha \in \mathbb{Q} - \{0\}$. Let $(a,b) \in L$ such that $(a,b) \neq (1,1)$ and $b \notin \{-1,0\}$. Then*

$$\pi_6[\mathcal{D}, L] \propto_T \pi_5[\mathcal{D}, a, b].$$

We now define two root connected digraphs and state their Tutte polynomials.

- Let P_k be the root connected directed path of length k with the root being one of the leaves. Then $T(P_k; x, y) = 1 + \sum_{i=1}^{k}(x-1)^i y^{i-1}$.
- Let S_k be the root connected directed star with k edges emanating from the root. Then $T(S_k; x, y) = x^k$.

These are easy to prove in a similar way to Propositions 1.7.2 and 1.7.3 using induction on k. The proof of the following proposition is analogous to that of Proposition 2.3.11 and uses the Tutte polynomials of P_2 and S_k.

Proposition 2.4.5. *Let L be the line $y = -1$. For $a \notin \{\frac{1}{2}, 1\}$ we have*

$$\pi_6[\mathcal{D}, L] \propto_T \pi_5[\mathcal{D}, a, -1].$$

In a similar way to Section 2.3, we begin the proof of Theorem 2.4.3 by examining the easy points. Let $D = (r, V, \overrightarrow{E})$ be a rooted digraph, then for any point (a,b) lying on the hyperbola H_1 we have

$$T(D; a, b) = (a-1)^{\rho(D) - |\overrightarrow{E}|} a^{|\overrightarrow{E}|}.$$

This can be computed in linear time.

We now show that evaluating $T(D; a, 0)$ is easy for all $a \in \mathbb{Q}$. In [23] Gordon and McMahon define the following characteristic polynomial $p(D; \lambda)$ of a rooted digraph D and show that if D is root connected and has precisely s sinks, then

$$p(D; \lambda) = \begin{cases} (-1)^{\rho(D)}(1-\lambda)^s & \text{if } D \text{ is acyclic;} \\ 0 & \text{if } D \text{ has a directed cycle.} \end{cases}$$

Using the relation $T(D; 1-\lambda, 0) = (-1)^{\rho(D)} p(D; \lambda)$ we arrive at the result

$$T(D;x,0) = \begin{cases} x^s & \text{if } D \text{ is acyclic;} \\ 0 & \text{if } D \text{ has a directed cycle.} \end{cases}$$

This can easily be computed, therefore the problem $\pi_5[\mathcal{D}, a, 0]$ is easy for any $a \in \mathbb{Q}$.

Now suppose we have a rooted digraph $D = (r, V, \overrightarrow{E})$ with $|V| = n$. The $n \times n$ Laplacian matrix $Q(D)$ of D can be constructed as follows.

- Entry $q_{i,j}$ for distinct i, j equals $-m$ where m is the number of edges from i to j.

- Entry $q_{i,i}$ equals the in-degree of i minus the number of loops at i.

Although the original paper proves difficult to find, the following theorem is a result by Tutte and can be found in [56]. The proof of the theorem makes use of the so-called Kirchoff matrix.

Theorem 2.4.6. *Let D be a rooted digraph. The number of spanning full arborescences is equal to the determinant of the matrix obtained by removing the row and column of r in the Laplacian matrix $Q(D)$.*

Example 2.4.7. *Consider the graph given in Figure 2.3.*

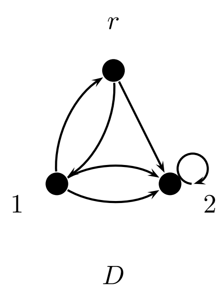

Figure 2.3

The Laplacian matrix $Q(D)$ of D is given by

$$Q(D) = \begin{array}{c} \\ r \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} r & 1 & 2 \end{array} \\ \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -2 \\ 0 & 0 & 3 \end{pmatrix} \end{array}.$$

The determinant of the matrix obtained by deleting the row and column corresponding to r is 3. The three subdigraphs of D that are spanning full arborescences are given in Figure 2.4.

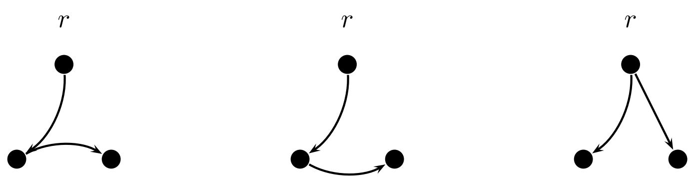

Figure 2.4

Computing the determinant of a matrix takes polynomial time [15]. Since $T(D;1,1)$ counts the number of spanning full arborescences of a rooted digraph D we have shown that computing $\pi_5[\mathcal{D},1,1]$ can be done in polynomial time and is therefore easy.

We now review the hard points of Theorem 2.4.3. We begin by classifying the complexity of evaluating $\pi_5[\mathcal{D},1,b]$ for $b \notin \{0,1\}$. Provan and Ball [45] define a reliability measure that computes the probability of a rooted digraph $D = (r, V, \overrightarrow{E})$ being root connected. Given $p \in \mathbb{R}$ with $0 \leq p \leq 1$ they impose a stochastic structure on D in which the edges of D are subject to random failure, independently and each with equal probability p. Edges that have not failed are said to be *operative*. They let

$$\begin{aligned} g(D;p) &= \mathrm{P}(\text{there is a path of operative edges from } r \text{ to every other vertex in } D) \\ &= \sum_{j=0}^{|\overrightarrow{E}|} g_j p^j (1-p)^{|\overrightarrow{E}|-j} \end{aligned}$$

where

$g_j =$ the number of sets of edges of cardinality j whose complement admits a path from r to every other vertex in D.

Example 2.4.8. *For D in Example 2.4.7 we have*

$$\begin{aligned} g(D;p) &= \sum_{j=0}^{6} g_j p^j (1-p)^{6-j} \\ &= (1-p)^6 + 5p(1-p)^5 + 10p^2(1-p)^4 + 9p^3(1-p)^3 + 3p^4(1-p)^2. \end{aligned}$$

In particular we have $g(D; \frac{1}{2}) = \frac{7}{16}$. Note that when $p = \frac{1}{2}$

$$g(D;p) = \frac{\textit{Number of spanning subdigraphs of } D \textit{ that are root connected}}{\textit{Total number of spanning subdigraphs of } D} = \frac{T(D;1,2)}{T(D;2,2)}$$

for any rooted digraph D.

Provan and Ball show that the following computational problem is #P-complete for fixed rational p with $0 < p < 1$, and easy when $p = 0$ or $p = 1$. Note that we have restricted the input digraph to being root connected in the problem which Provan and Ball did not, but this does not make a difference because if it is not root connected then clearly $g(D;p) = 0$.

#CONNECTEDNESS RELIABILITY
Input $D \in \mathcal{D}$.
Output $g(D;p)$.

We now use this result to classify a range of points along the line $x = 1$.

Proposition 2.4.9. *The computational problem $\pi_5[\mathcal{D}, 1, b]$ is #P-hard for $b > 1$.*

Proof. Let $D = (r, V, \overrightarrow{E})$ be a root connected digraph, and let

$$\begin{aligned} t_j &= \text{the number of sets of edges of cardinality } j \text{ that admit a path from the root of } D \text{ to every} \\ &\quad\ \text{other vertex in } D \\ &= \text{the number of sets of edges of cardinality } j \text{ with full rank.} \end{aligned}$$

Then for $0 < p < 1$ we have

$$\begin{aligned}
g(D;p) &= \sum_{j=0}^{|\overrightarrow{E}|} g_j p^j (1-p)^{|\overrightarrow{E}|-j} = \sum_{j=0}^{|\overrightarrow{E}|} t_j p^{|\overrightarrow{E}|-j}(1-p)^j \\
&= \sum_{\substack{A\subseteq \overrightarrow{E}: \\ \rho(A)=\rho(D)}} p^{|\overrightarrow{E}|-|A|}(1-p)^{|A|} = p^{|\overrightarrow{E}|-\rho(D)}(1-p)^{\rho(D)} \sum_{\substack{A\subseteq \overrightarrow{E}: \\ \rho(A)=\rho(D)}} \left(\frac{1-p}{p}\right)^{|A|-\rho(A)} \\
&= p^{|\overrightarrow{E}|-\rho(D)}(1-p)^{\rho(D)} T\left(D;1,\frac{1}{p}\right).
\end{aligned}$$

Evaluating $g(D;p)$ is therefore Turing-reducible to evaluating $T(D;1,\frac{1}{p})$ for $0 < p < 1$. Furthermore the problem $\pi_5[\mathcal{D},1,b]$ is #P-hard to compute for $b > 1$. $\square$

In order to determine the complexity of the point $(1,-1)$ we introduce a new operation on root connected digraphs called the k-digon-stretch. We define a tailed k-digon from u to v to be the digraph comprising a path of length 1 from vertex u to vertex w and a path of k digons (directed cycles of length 2) from vertex w to vertex v. Let D be a root connected digraph, the k-digon stretch D_k of D is constructed by replacing every directed edge uv in D by a tailed k-digon from u to v.

Theorem 2.4.10. *Let $D = (r, V, \overrightarrow{E})$ be a root connected digraph. The Tutte polynomial of the k-digon-stretched graph $D_k = (r, V_k, \overrightarrow{E}_k)$ of D when $x = 1$ is given by*

$$T(D_k;1,y) = (k+1)^{|\overrightarrow{E}|-\rho(D)} y^{k|\overrightarrow{E}|} T\left(D;1,\frac{k+y}{k+1}\right).$$

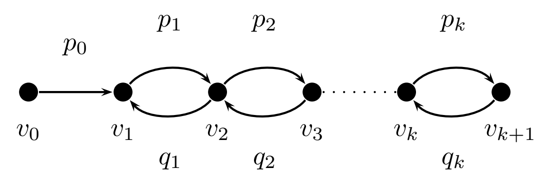

Figure 2.5: A tailed k-digon from vertex v_0 to vertex v_{k+1}.

Proof. Figure 2.5 shows a tailed k-digon in D_k replacing an edge e in D. Let S be a subset of edges of the tailed k-digon. If S contains all elements $p_0, p_1, p_2, \ldots, p_k$ then S is said to admit a *strong path* through the tailed k-digon. The vertices $v_1, v_2, \ldots, v_k$ are the *internal* vertices of the tailed k-digon.

Let $A \subseteq \overrightarrow{E}_k$ and $B(A)$ be the set of edges in D such that A contains a strong path through their corresponding tailed k-digons.

We have $\rho(A) = \rho(D_k)$ if and only if

(i) for each tailed k-digon and for each internal vertex v_i, A contains a path $p_0p_1p_2\dots p_{i-1}$ or $q_kq_{k-1}q_{k-2}\dots q_i$, and

(ii) $\rho(B(A)) = \rho(D)$.

Now

$$\begin{aligned}\rho(D_k) &= k|\overrightarrow{E}| + \rho(D)\\ &= k|\overrightarrow{E}| - k|B(A)| + (k+1)|B(A)| + \rho(D) - |B(A)|.\end{aligned}$$

We can write A as the disjoint union $A = \bigcup_{e\in\overrightarrow{E}} A_e$ where A_e is the intersection of A with the edges of the tailed k-digon replacing e. Let $\alpha = k|\overrightarrow{E}| - k|B(A)| + (k+1)|B(A)| + \rho(D) - |B(A)|$. The Tutte polynomial of D_k along the line $x = 1$ is therefore given by

$$\begin{aligned}T(D_k;1,y) &= \sum_{\substack{A\subseteq\overrightarrow{E}_k:\\ \rho(A)=\rho(D_k)}} (y-1)^{|A|-\rho(D_k)} = \sum_{\substack{B\subseteq\overrightarrow{E}:\\ \rho(B)=\rho(D)}} \sum_{\substack{A\subseteq\overrightarrow{E}_k:\\ \rho(A)=\rho(D_k)\\ B(A)=B}} (y-1)^{|A|-\rho(D_k)}\\
&= \sum_{\substack{B\subseteq\overrightarrow{E}:\\ \rho(B)=\rho(D)}} \sum_{\substack{A\subseteq\overrightarrow{E}_k:\\ \rho(A)=\rho(D_k),\\ B(A)=B}} \left(\prod_{e\in\overrightarrow{E}} (y-1)^{|A_e|}\right)(y-1)^{-\alpha}\\
&= \sum_{\substack{B\subseteq\overrightarrow{E}:\\ \rho(B)=\rho(D)}} \sum_{\substack{A\subseteq\overrightarrow{E}_k:\\ \rho(A)=\rho(D_k),\\ B(A)=B}} \left(\prod_{\substack{e\in\overrightarrow{E}\\ e\notin B(A)}} (y-1)^{|A_e|}\right)\left(\prod_{\substack{e\in\overrightarrow{E}:\\ e\in B(A)}} (y-1)^{|A_e|}\right)(y-1)^{-\alpha}\\
&= \sum_{\substack{B\subseteq\overrightarrow{E}:\\ \rho(B)=\rho(D)}} \sum_{\substack{A\subseteq\overrightarrow{E}_k:\\ \rho(A)=\rho(D_k),\\ B(A)=B}} \underbrace{\left(\prod_{\substack{e\in\overrightarrow{E}:\\ e\notin B(A)}} (y-1)^{|A_e|-k}\right)}_{(1)}\underbrace{\left(\prod_{\substack{e\in\overrightarrow{E}:\\ e\in B(A)}} (y-1)^{|A_e|-(k+1)}\right)}_{(2)}(y-1)^{|B(A)|-\rho(D)} \qquad (2.7)\end{aligned}$$

(1) Here $e \notin B(A)$. Therefore not all of the edges $p_0, p_1, \dots, p_k$ can belong to A_e. Suppose that $p_0, p_1, \dots, p_j \in A_e$ but $p_{j+1} \notin A_e$ for some $j < k$. Since $\rho(A) = \rho(D_k)$ the remaining vertices $v_{j+1}, \dots, v_k$ must be reachable by a directed path from v_{k+1}. Thus $q_k, q_{k-1}, \dots, q_{j+2} \in A_e$. The minimum number of edges in A_e is therefore k. Now we have the remaining edges

$p_{j+2}, \ldots, p_k, q_{j+1}, \ldots, q_1$, each of which could be included in our subset A_e or not. There are $k-(j+1)+(j+1) = k$ edges in this list. The presence of t of these edges in our set A_e would give $|A_e| = k+t$. There are $k+1$ choices of j. So summing $\prod_{\substack{e \in \overrightarrow{E}: \\ e \notin B(A)}} (y-1)^{|A_e|-k}$ over all choices of A_e gives

$$\left((k+1)\sum_{t=0}^{k}\binom{k}{t}(y-1)^t\right)^{|\overrightarrow{E}|-|B|} = \left((k+1)y^k\right)^{|\overrightarrow{E}|-|B|}.$$

(2) Here $e \in B(A)$. Therefore the edges $p_0, p_1, \ldots, p_k$ must belong to A_e. The minimum number of edges in A_e is therefore $k+1$. Now we have the remaining edges $q_1, q_2, \ldots, q_k$, each of which could be included in our subset A_e or not. There are k edges in this list. The presence of t of these edges in our set A_e would give $|A_e| = k+1+t$. So summing $\prod_{\substack{e \in \overrightarrow{E}: \\ e \in B(A)}} (y-1)^{|A_e|-(k+1)}$ over all choices of A_e gives

$$\left(\sum_{t=0}^{k}\binom{k}{t}(y-1)^t\right)^{|B|} = (y^k)^{|B|}.$$

Thus Equation 2.7 becomes

$$\begin{aligned}
&\sum_{\substack{B \subseteq \overrightarrow{E}: \\ \rho(B)=\rho(D)}} (y^k)^{|B|} \left((k+1)y^k\right)^{|\overrightarrow{E}|-|B|} (y-1)^{|B|-\rho(D)} \\
&= (y^k)^{|\overrightarrow{E}|} \sum_{\substack{B \subseteq \overrightarrow{E}: \\ \rho(B)=\rho(D)}} (k+1)^{\rho(B)-|B|+|\overrightarrow{E}|-\rho(D)} (y-1)^{|B|-\rho(B)} \\
&= (y^k)^{|\overrightarrow{E}|}(k+1)^{|\overrightarrow{E}|-\rho(D)} T\left(D; 1, \frac{y+k}{k+1}\right).
\end{aligned}$$

□

We now complete the classification of the complexities along the line H_0^x.

Proposition 2.4.11. *The computational problem $\pi_5[\mathcal{D}, 1, b]$ is #P-hard for $b \notin \{0, 1\}$.*

Proof. For $b \notin \{-1, 0, 1\}$ the proof follows immediately from Propositions 2.4.4 and 2.4.9. By Theorem 2.4.10 forming the 2-digon-stretch of a root connected digraph $D = (r, V, \overrightarrow{E})$ and setting $y = -1$ yields

$$T(D_2; 1, -1) = 3^{|\overrightarrow{E}|-\rho(D)} T\left(D; 1, \frac{1}{3}\right).$$

Therefore evaluating the Tutte polynomial of a root connected digraph at the point $(1, -1)$ is at least as hard as evaluating it at $(1, \frac{1}{3})$, which we have just shown to be #P-hard. $\square$

We now show that evaluating the Tutte polynomial of a root connected digraph at most points on the hyperbola H_α for $\alpha \neq 0$ is at least as hard as evaluating it at the point $(1 + \alpha, 2)$.

Proposition 2.4.12. *Let $(a, b) \in H_\alpha$ for $\alpha \in \mathbb{Q} - \{0\}$ with $b \notin \{-1, 0\}$, then*

$$\pi_5[\mathcal{D}, 1 + \alpha, 2] \propto_T \pi_5[\mathcal{D}, a, b].$$

Proof. The hyperbola H_α for $\alpha \in \mathbb{Q} - \{0\}$ crosses the line $y = 2$ at the point $(1 + \alpha, 2)$. By Proposition 2.4.4 we know that for any point $(a, b) \in H_\alpha$ with $b \notin \{-1, 0\}$ we have $\pi_5[\mathcal{D}, 1+\alpha, 2] \propto_T \pi_6[\mathcal{D}, H_\alpha] \propto_T \pi_5[\mathcal{D}, a, b]$. $\square$

We will now show that evaluating the Tutte polynomial of a root connected digraph at most of the points on the line $y = 2$ is #P-hard. This will enable us to classify the complexity of most points lying on the hyperbola H_α for all $\alpha \in \mathbb{Q} - \{0\}$.

Proposition 2.4.13. *The computational problem $\pi_5[\mathcal{D}, a, 2]$ is #P-hard for $a \neq 2$.*

Proof. We begin by proving that when L is the line $y = 2$ we have the reduction

$$\pi_6[\mathcal{D}, L] \propto_T \pi_5[\mathcal{D}, a, 2]$$

for $a \notin \{1, 2\}$. Let $D = (r, V, \overrightarrow{E})$ be a root connected digraph and let $z = x - 1$. Along L the Tutte polynomial of D has the form

$$T(D; x, 2) = \sum_{A \subseteq \overrightarrow{E}} z^{\rho(D) - \rho(A)} = \sum_{i=0}^{\rho(D)} t_i z^i$$

for some $t_0, t_1, \ldots, t_{\rho(D)}$. We will now show that we may determine all of the coefficients t_i in polynomial time from $T(D \sim S_k; a, 2)$ for $k = 0, 1, \ldots, \rho(D)$, apart from at some exceptional values of a. By Lemma 2.4.2 we have

$$T(D \sim S_k; a, 2) = a^{k\rho(D)} T\left(D; \frac{2^k (a-1)^{k+1}}{a^k} + 1, 2\right).$$

Therefore we may compute $T\left(D; \frac{2^k(a-1)^{k+1}}{a^k}+1, 2\right)$ from $T(D \sim S_k; a, 2)$ when $a \neq 0$. For $a \notin \{\frac{2}{3}, 1\}$ the values of $\left(\frac{2^k(a-1)^{k+1}}{a^k}+1, 2\right)$ are pairwise distinct for $k = 0, 1, \ldots, \rho(D)$. Suppose otherwise, that is for some $m \neq n$ we have

$$\frac{2^m(a-1)^{m+1}}{a^m}+1 = \frac{2^n(a-1)^{n+1}}{a^n}+1.$$

This can be reduced to

$$\left(\frac{2(a-1)}{a}\right)^{m-n} = 1.$$

It should be straightforward to see that this is only satisfied when $a = 2$, $m = n$ or $a = \frac{2}{3}$ and $m - n = 2t$ for some $t \in \mathbb{Z}$. Therefore by evaluating $T(D \sim S_k; a, 2)$ for $k = 0, 1, \ldots, \rho(D)$ where $a \notin \{0, \frac{2}{3}, 1, 2\}$ we obtain $\sum_{i=0}^{\rho(D)} t_i z^i$ for $\rho(D) + 1$ distinct values of z. This gives us $\rho(D) + 1$ linear equations for the coefficients t_i, and so they may be recovered in polynomial time. Hence evaluating the Tutte polynomial of a root connected digraph along the line $y = 2$ is Turing-reducible to evaluating it at the point $(a, 2)$ for $a \notin \{0, \frac{2}{3}, 1, 2\}$.

We now consider the cases where $a = 0$ and $a = \frac{2}{3}$. By Lemma 2.4.2 we have

$$\begin{aligned} T(D \sim P_2; 0, 2) &= 2^{\rho(D)} T\left(D; \frac{(-1)^3 2^2}{2}+1, 2\right) \\ &= 2^{\rho(D)} T(D; -1, 2). \end{aligned}$$

Therefore we have the reduction

$$\pi_5[\mathcal{D}, -1, 2] \propto_T \pi_5[\mathcal{D}, 0, 2].$$

Similarly we have

$$\begin{aligned} T\left(D \sim P_2; \frac{2}{3}, 2\right) &= 2^{\rho(D)} T\left(D; \frac{(-\frac{1}{3})^3 2^2}{2}+1, 2\right) \\ &= 2^{\rho(D)} T\left(D; \frac{25}{27}, 2\right). \end{aligned}$$

Therefore we have the reduction

$$\pi_5[\mathcal{D}, 25/27, 2] \propto_T \pi_5[\mathcal{D}, 2/3, 2].$$

By Proposition 2.4.11 we already know that evaluating the Tutte polynomial of a root connected digraph at the point $(1,2)$ is #P-hard. For $a \notin \{1,2\}$ we now have the following reductions

$$\pi_5[\mathcal{D},1,2] \propto_T \pi_6[\mathcal{D},L] \propto_T \pi_5[\mathcal{D},a,2].$$

This completes the proof. $\square$

Theorem 2.4.14. *For $(a,b) \in H_\alpha$ for $\alpha \in \mathbb{Q} - \{0,1\}$ with $b \neq 0$, the computational problem $\pi_5[\mathcal{D},a,b]$ is #P-hard.*

Proof. By Proposition 2.4.12 we have $\pi_5[\mathcal{D},1+\alpha,2] \propto_T \pi_5[\mathcal{D},a,b]$ for $b \notin \{-1,0\}$. By Proposition 2.4.13 we have $\pi_5[\mathcal{D},a,2]$ is #P-hard to compute for $a \neq 2$. Therefore since $\alpha \neq 1$ and $b \neq 0$ we have $\pi_5[\mathcal{D},a,b]$ is #P-hard for $b \neq -1$.

Now by Proposition 2.4.5 we have $\pi_5[\mathcal{D},1,-1] \propto_T \pi_5[\mathcal{D},a,-1]$ for $a \neq 1/2$. By Proposition 2.4.11 $\pi_5[\mathcal{D},1,-1]$ is #P-hard to compute. Therefore since $a \neq \frac{1}{2}$ $\pi_5[\mathcal{D},a,-1]$ is #P-hard. $\square$

The only remaining points we need to classify are those lying on the line $y=1$. To do this we prove that the problem of evaluating the Tutte polynomial of a root connected digraph at most fixed points along this line is at least as hard as the analogous problem for rooted graphs.

Theorem 2.4.15. *The computational problem $\pi_5[\mathcal{D},a,1]$ is #P-hard for $a \in \mathbb{Q} - \{1\}$.*

Proof. Let $G=(r,V,E)$ be a connected rooted graph. Suppose we construct a root connected digraph $D=(r,V,\overrightarrow{E})$ from G by replacing every edge in G by a digon with one edge oriented in each direction. We can define a natural map $f : 2^{\overrightarrow{E}} \to 2^E$ with $f(\overrightarrow{A}) = A$ such that $A \subseteq E$ is the set of edges in G that are replaced by edges of $\overrightarrow{A} \subseteq \overrightarrow{E}$ in D.

If $\rho_G(A) = |A|$ then $G|A$ is a rooted tree, possibly with isolated vertices. Similarly if $\rho_D(\overrightarrow{A}) = |\overrightarrow{A}|$ then $G|\overrightarrow{A}$ is a rooted arborescence, possibly with isolated vertices. For every subset A of E with $\rho_G(A)=|A|$, there is precisely one choice of $\overrightarrow{A} \subseteq \overrightarrow{E}$ with $\rho_D(\overrightarrow{A}) = |\overrightarrow{A}|$ and $f(\overrightarrow{A}) = A$. This is obtained by directing all the edges of A away from r. On the other hand if $\rho(\overrightarrow{A}) = |\overrightarrow{A}|$ then by removing the directions from the edges of $\overrightarrow{A}$, we obtain a set A with $\rho(A)=|A|$ and $f(\overrightarrow{A}) = A$. Thus there is a one-to-one correspondence between subsets A of E with $\rho_G(A)=|A|$ and subsets $\overrightarrow{A}$

of $\overrightarrow{E}$ with $\rho_D(\overrightarrow{A}) = |\overrightarrow{A}|$, and this correspondence preserves the sizes of the sets. Therefore we have

$$\begin{aligned}T(D;x,1) &= \sum_{\substack{\overrightarrow{A}\subseteq\overrightarrow{E}:\\ |\overrightarrow{A}|=\rho_D(\overrightarrow{A})}} (x-1)^{\rho(D)-|\overrightarrow{A}|} = \sum_{\substack{A\subseteq E:\\ |A|=\rho_G(A)}} (x-1)^{\rho(G)-|A|}\\ &= T(G;x,1).\end{aligned}$$

By the reduction $\pi_2[\mathcal{G},a,1] \propto_T \pi_5[\mathcal{D},a,1]$ and Proposition 2.3.16 we have proved that $\pi_5[\mathcal{D},a,1]$ is #P-hard for $a \neq 1$. □

2.5 Binary Greedoids

In this section we determine the computational complexity of evaluating the Tutte polynomial of a binary greedoid at a fixed rational point.

Let $\Gamma = \Gamma(M)$ for some binary matrix M. The k-thickening Γ^k of Γ is the binary greedoid $\Gamma(M')$ where M' is the matrix obtained by replacing each column of M by k copies of the column. We have $\Gamma(M') = (\Gamma(M))^k$, so Theorem 2.2.1 can be specialized to binary greedoids in the following way.

Lemma 2.5.1. *Let Γ be a binary greedoid. The Tutte polynomial of the k-thickening Γ^k of Γ when $y \neq 1$ is given by*

$$T(\Gamma^k;x,y) = (1+y+\ldots+y^{k-1})^{\rho(\Gamma)}T\left(\Gamma;\frac{x+y+\ldots+y^{k-1}}{1+y+\ldots+y^{k-1}},y^k\right).$$

When $y=-1$ we have

$$T(\Gamma^k;x,-1) = \begin{cases}(x-1)^{\rho(\Gamma)} & \textit{if } k \textit{ is even;}\\ T(\Gamma;x,-1) & \textit{if } k \textit{ is odd.}\end{cases}$$

We now determine the Tutte polynomial of the binary greedoid $\Gamma(I_k)$ where I_k is the $k\times k$ identity matrix.

Proposition 2.5.2. *The Tutte polynomial of the binary greedoid $\Gamma(I_k)$ is given by*

$$T(\Gamma(I_k);x,y) = 1+\sum_{j=1}^{k}(x-1)^j y^{j-1}.$$

Proof. We prove this result by performing induction on k. When $k = 1$ the matrix I_1 is given by a single entry 1. The Tutte polynomial of $\Gamma(I_1)$ is therefore given by $T(\Gamma(I_1); x, y) = 1 + (x - 1)$. Now assume the result holds for $k = t$. Let $\Gamma = \Gamma(I_{t+1})$. Let M' be obtained by deleting the first column of I_{t+1} and M'' be obtained by deleting the first row and column of I_{t+1}. Suppose that e is the element of Γ labelling the first column. Then $\Gamma(I_{t+1}) \setminus e = \Gamma(M')$ and by Lemma 1.3.19 $\Gamma(I_{t+1})/e = \Gamma(M'')$. Note that $M'' = I_t$ and every element of $\Gamma(M')$ is a loop. Thus

$$\begin{aligned} T(\Gamma; x, y) &= 1 + \sum_{j=1}^{t} (x-1)^j y^{j-1} + (x-1)^{t+1} y^t \\ &= 1 + \sum_{j=1}^{t+1} (x-1)^j y^{j-1} \end{aligned}$$

as required.

$\square$

We now present a special case of Theorem 2.2.4.

Proposition 2.5.3. *Let Γ be a binary greedoid and let $\Gamma' = \Gamma(I_k)$. Then*

$$T(\Gamma \approx \Gamma'; x, y) = T(\Gamma; x, y)(x-1)^k y^k + T(\Gamma; 1, y)\Big(1 + \sum_{j=1}^{k} (x-1)^j y^{j-1} - (x-1)^k y^k\Big).$$

Proof. The proof follows immediately from Theorem 2.2.4 and Proposition 2.5.2. $\square$

In a similar way to previous sections we concentrate on three particular computational problems. Let $\mathcal{B}$ denote the class of all binary greedoids, let $a, b \in \mathbb{Q}$ and let $x(t) = p(t)/q(t)$, $y(t) = r(t)/s(t)$ be parametric equations of L where p, q, r and s are polynomials over $\mathbb{Q}$.

$\pi_7[\mathcal{B}]$: #BINARY GREEDOID TUTTE POLYNOMIAL

Input $\Gamma \in \mathcal{B}$.

Output The coefficients of $T(\Gamma; x, y)$.

$\pi_8[\mathcal{B}, a, b]$: #BINARY GREEDOID TUTTE POLYNOMIAL AT (a, b)

Input $\Gamma \in \mathcal{B}$.

Output $T(\Gamma; a, b)$.

$\pi_9[\mathcal{B}, L]$: #BINARY GREEDOID TUTTE POLYNOMIAL ALONG L

Input $\Gamma \in \mathcal{B}$.

Output The coefficients of the rational function of t given by evaluating $T(\Gamma; x(t), y(t))$.

The main result from this section is as follows.

Theorem 2.5.4. *The problem $\pi_8[\mathcal{B}, a, b]$ is #P-hard for all (a, b) except when (a, b) lies on H_1. In this exceptional case $\pi_8[\mathcal{B}, a, b]$ is easy.*

The proof of the following proposition is analogous to that of Proposition 2.3.10, thus we omit it from this section.

Proposition 2.5.5. *Let L be either H_0^x, H_0^y, or H_α for $\alpha \in \mathbb{Q} - \{0\}$. Let $(a, b) \in L$ such that $(a, b) \neq (1, 1)$ and $b \notin \{-1, 0\}$. Then*

$$\pi_9[\mathcal{B}, L] \propto_T \pi_8[\mathcal{B}, a, b].$$

We begin by examining the easy points of Theorem 2.5.4. Let $\Gamma = (E, \mathcal{F})$ be a binary greedoid, then for any point (a, b) lying on the hyperbola H_1 we have

$$T(\Gamma; a, b) = (a-1)^{\rho(\Gamma)-|E|} a^{|E|},$$

which is easy to compute.

A binary matroid is a matroid that can be represented over the finite field $\mathbb{Z}_2$. Every graphic matroid is also binary, so Theorem 1.6.1 and Lemma 1.3.17 imply that $\pi_8[\mathcal{B}, 1, b]$ is #P-hard providing $b \neq 1$. By combining this with the following unpublished result by Vertigan, we are able to begin examining the hard points of Theorem 2.5.4.

Theorem 2.5.6 (Vertigan)**.** *Evaluating the Tutte polynomial of a binary matroid is #P-hard at the point* $(1, 1)$.

We now classify the complexity of evaluating the Tutte polynomial of a binary greedoid along the lines H_0^x and H_0^y. It is worth noting that the following two propositions are the only results that rely on Proposition 2.5.6.

Proposition 2.5.7. *The computational problem $\pi_8[\mathcal{B}, 1, b]$ is #P-hard for all b.*

Proof. This follows from Theorem 2.5.6 and the remarks before it. □

Proposition 2.5.8. *The computational problem $\pi_8[\mathcal{B}, a, 1]$ is #P-hard for all a.*

Proof. By Proposition 2.5.5 we have $\pi_9[\mathcal{B}, H_0^y] \propto_T \pi_8[\mathcal{B}, a, 1]$ for $a \neq 1$. The result now follows from Proposition 2.5.7. □

We now classify the complexity of evaluating the Tutte polynomial of a binary greedoid along $y = 0$ and $y = -1$.

Proposition 2.5.9. *The computational problem $\pi_8[\mathcal{B}, a, 0]$ is #P-hard for all $a \neq 0$.*

Proof. First note that $(0, 0)$ lies on H_1. Let Γ be a binary greedoid and let $\Gamma' = \Gamma(I_k)$. Now by Proposition 2.5.3 we have

$$T(\Gamma \approx \Gamma'; a, 0) = aT(\Gamma; 1, 0).$$

Therefore when $a \neq 0$ we have the reduction $\pi_8[\mathcal{B}, 1, 0] \propto_T \pi_8[\mathcal{B}, a, 0]$. The result now follows from Proposition 2.5.7. □

Proposition 2.5.10. *The computational problem $\pi_8[\mathcal{B}, a, -1]$ is #P-hard for all $a \neq \frac{1}{2}$.*

Proof. By applying Proposition 2.5.3 with $k = 1$, after a little rearrangement we obtain

$$(2a - 1)T(\Gamma; 1, -1) = T(\Gamma \approx \Gamma'; a, -1) + (a - 1)T(\Gamma; a, -1).$$

Thus, providing $a \neq \frac{1}{2}$, an algorithm solving $\pi_8[\mathcal{B}, a, -1]$ in polynomial time could be used to determine both $T(\Gamma \approx \Gamma'; a, -1)$ and $T(\Gamma; a, -1)$, and hence $T(\Gamma; 1, -1)$. The result follows by applying Proposition 2.5.7. □

The following final proposition of this section completes the proof of Theorem 2.5.4.

Proposition 2.5.11. *Let $(a, b) \in H_\alpha$ for $\alpha \in \mathbb{Q} - \{0, 1\}$ with $b \neq -1$, then the computational problem $\pi_8[\mathcal{B}, a, b]$ is #P-hard.*

Proof. The hyperbola H_α for $\alpha \in \mathbb{Q} - \{0, 1\}$ crosses the x-axis at the point $(1 - \alpha, 0)$. By Proposition 2.5.5 since $b \neq -1$ and $(a, b) \neq (1, 1)$ we have $\pi_8[\mathcal{B}, 1 - \alpha, 0] \propto_T \pi_8[\mathcal{B}, a, b]$. The result now follows from Proposition 2.5.9. □

Chapter 3

Polynomial-Time Algorithms for Evaluating the Tutte Polynomial

3.1 Introduction

From Chapter 2 we know that evaluating the Tutte polynomial of three particular classes of greedoids is #P-hard at most fixed rational points. Here we restrict each class to those that are of bounded tree-width and furthermore construct algorithms using a linear number of arithmetic operations to evaluate the Tutte polynomial of each of them at a fixed rational point. The theory of our algorithms closely follows the work of Noble in [42] in which he gives a polynomial-time algorithm to evaluate the Tutte polynomial of a graph of bounded tree-width. The main result from [42] is as follows.

Theorem 3.1.1 (Noble)**.** *For every $k \in \mathbb{N}$, there is an algorithm $\mathcal{A}_k$ that will input a graph G having tree-width at most k, and rationals $x = \frac{p_x}{q_x}, y = \frac{p_y}{q_y}$, and evaluate $T(G; x, y)$ in time at most*

$$O(f(k)(n+p)(n+m)s\log((n+m)s)\log\log((n+m)s))$$

where $s = \log(|p_x| + |q_x| + |p_y| + |q_y| + 1)$, p is the largest size of a set of mutually parallel edges in G and $f(k)$ is given by

$$f(k) = k^5(2k+1)^{(2k-1)}((4k+5)^{(4k+5)}(2^{(2k+5)}/3)^{(4k+5)})^{(4k+1)}.$$

Before we can discuss the notion of a graph having bounded tree-width, we must first define a tree-decomposition of a graph which essentially, as its name suggests, decomposes the graph into pieces connected in a tree-like fashion. The notion of a tree-decomposition of a graph was first developed by Halin in [26], and later rediscovered by Robertson and Seymour in [46] where they use it to find a polynomial-time algorithm to determine whether a graph has a subgraph contractible to a fixed planar graph. Our incentive is to represent any graph as a tree because many algorithms on graphs become easy when the input is restricted to being a tree. We will refer to the "vertices" of a tree-decomposition as *nodes* and to the edges as *branches*.

Definition 3.1.2 (Tree-Decomposition)**.** Let $G = (V, E)$ be a graph. A *tree-decomposition* of G is a pair $(\{S_i | i \in I\}, T = (I, B))$ such that T is a tree with branches B, and for every node i of T, we have a subset $S_i \subseteq V$, called the *bag* of i, satisfying the following axioms:

(TD1) $\bigcup_{i \in I} S_i = V$.

(TD2) for every edge $\{v, w\} \in E$, there exists an $i \in I$ such that $\{v, w\} \subseteq S_i$.

(TD3) for all $i, j, k \in I$, if j is on the path from i to k in T, then $S_i \cap S_k \subseteq S_j$.

Figure 3.1 illustrates an example of a tree-decomposition of a graph. Of course one could simply construct a trivial tree-decomposition of a graph with just one node containing all vertices of the graph, however this will not be of any computational interest. Note that a tree-decomposition of a graph is not unique and that two non-isomorphic graphs can share the same tree-decomposition. The property that we essentially want to carry over from trees is that the deletion of a very small set of vertices breaks the graph into disconnected components. A graph G is called a *k-tree* if and only if either G is the complete graph with k vertices, or G has a vertex v with degree k such that vertices adjacent to v form a complete graph and $G \setminus v$ is a k-tree. A *partial k-tree* is any subgraph of a k-tree. An example of a 2-tree is given in Figure 3.2.

The concept of tree-width was introduced by Robertson and Seymour in their work on graph minors [46], and almost simultaneously by Arnborg and Proskurowski in their work on partial k-trees [3]. We will exclusively focus on the definition in terms of tree-decompositions, however it is worth noting that a graph has tree-width at most k if and only if it is a partial k-tree, a result obtained independently by Wimer [63] and Scheffler [49].

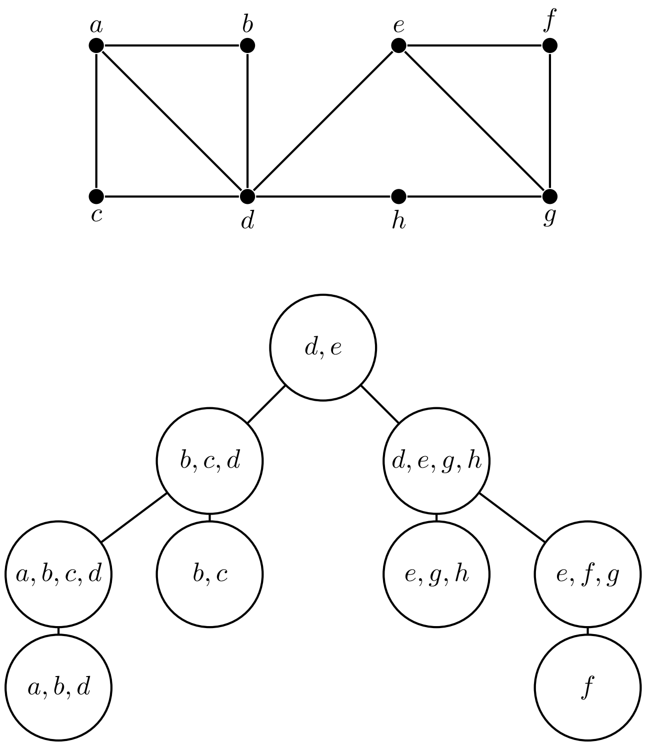

Figure 3.1: Graph with a corresponding tree-decomposition

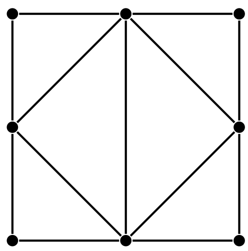

Figure 3.2: Example of a 2-tree

Definition 3.1.3 (Tree-Width)**.** Let $\tau = (\{S_i | i \in I\}, T = (I, B))$ be a tree-decomposition of a graph G. The *width* $w(\tau)$ of τ is given by

$$w(\tau) = \max_{i \in I} |S_i| - 1. \tag{3.1}$$

The *tree-width* $tw(G)$ of G is then said to be the minimum width taken over all possible tree-decompositions of G.

A graph G has tree-width 1 if and only if it is a tree or a forest. The -1 in the tree-width definition is somewhat arbitrary and ensures that a tree does in fact have tree-width 1. A graph $G = (V, E)$ has tree-width 0 if and only if $E = \emptyset$. Examples of other well-studied graphs along with their tree-width include series-parallel graphs which have tree-width 2, and the complete graphs K_n which have tree-width $n-1$ for $n \geq 1$. Intuitively, a graph has small tree-width if it can be recursively decomposed into small subgraphs that have small overlap. More precisely, it gives information about the connectivity of the graph. Therefore tree-width essentially measures the graph's deviation from a tree, i.e. the smaller the tree-width of G, the more "tree-like" the structure of G. Arnborg et al discovered that determining the tree-width of a graph is NP-complete [3].

The tree-width of a graph is a parameter which has proven to be very important in algorithmic graph theory. This is because many algorithmic problems that are intractable for arbitrary graphs, can be solved efficiently in polynomial and often linear time when restricted to the class of graphs of bounded tree-width. A well-known example of such a problem that becomes easy when the input graph is of bounded tree-width is given below.

#MAXIMUM INDEPENDENT SET
Input $G \in \overline{\mathcal{G}}$.
Output The size of a maximum independent set of vertices of G.

Before continuing with our approach to construct a fast algorithm to evaluate the Tutte polynomial of a rooted graph of bounded tree-width, we now discuss an independent approach due to Makowsky and coauthors using monadic second order logic (MSOL).

Let $G = (V, E)$ be a graph and let $R \subseteq V \times E$ be a binary relation such that $R(v, e)$ if and only if v is an endpoint of e in G. For technical reasons, in order to be able to incorporate polynomials such as the Tutte polynomial into the framework, we require that the graph comes with an arbitrary linear order on its vertices accessed by means of a successor relation S. More precisely $S(u, v)$ for vertices u and v if and only if u immediately precedes v in the linear order. The MSOL of graphs has variables v_i for vertices, e_i for edges, V_j for subsets of vertices, E_j for subsets of edges, and is built from

(i) the atomic formulae

$$v_i \in V_j, \quad e_i \in E_j, \quad R(v_i, e_i), \quad S(v_i, v_j), \quad v_i = v_j, \quad e_i = e_j,$$

(ii) the quantifiers $\forall$ and $\exists$ over the variables v_i, e_i, V_j, E_j, and

(iii) the standard logical connectives $\neg, \wedge, \vee, \rightarrow$ and $\leftrightarrow$.

Thus MSOL permits quantification over vertices, edges, subsets of vertices and subsets of edges. The monadic qualifier forbids quantification over functions or relations, which is, however, permitted in the full second order logic.

Suppose, for simplicity, that $G = (V, E)$ is loopless. Then the following MSOL formula states that G is 3-colourable.

$$\begin{gathered}
\exists A, B, C \subseteq V((\forall v \in V((v \in A) \vee (v \in B) \vee (v \in C))) \\
\wedge(\forall v_1, v_2 \in V, \forall e \in E \\
((R(e, v_1) \wedge R(e, v_2) \wedge (((v_1 \in A) \wedge (v_2 \in A)) \vee ((v_1 \in B) \wedge (v_2 \in B)) \vee ((v_1 \in C) \wedge (v_2 \in C)))) \\
\rightarrow (v_1 = v_2))))
\end{gathered}$$

In the formula the sets A, B and C denote the sets of vertices receiving each of the three colours. The first part of the formula ensures that every vertex belongs to at least one of A, B and C. The second part ensures that there is no edge joining two vertices belonging to the same set. The formula permits a vertex to belong to more than one of the sets A, B and C, or in other words to receive more than one colour. This does not create any difficulties because if the formula is true then there are certainly disjoint sets A, B and C that also satisfy it.

In [40] Makowsky and Mariño define MSOL-polynomials to be polynomials of the form

$$p(G) = c \sum_{\text{subgraphs } H} \left(\prod_{\text{case}_1} w_1 \cdot \ldots \cdot \prod_{\text{case}_\alpha} w_\alpha \right)$$

where the summation ranges over an MSOL-definable family of subgraphs H of G, the products range over all edges and vertices of H with an MSOL-definable finite case distinction where each case receives the same weight, and c is a constant.

The following result is proved in [38] using the spanning tree definition of the Tutte polynomial of a graph.

Lemma 3.1.4. *The Tutte polynomial of a graph is an MSOL-polynomial.*

In [38] Makowsky uses the results from [12] to prove the following theorem.

Theorem 3.1.5. *Let K be a class of graphs of tree-width at most k. Let $p(G)$ be an MSOL-polynomial. Then $p(G)$ can be computed on K in polynomial time.*

This gives a polynomial-time algorithm to evaluate the Tutte polynomial of a graph of bounded tree-width.

Makowsky [39] has informed us that the results of this section and the next can be obtained using techniques based on MSOL. There are many extensions of MSOL, for example, working with coloured graphs rather than ordinary graphs, as in [38]. But checking that the methods generalize to a specific case such as rooted graphs or the directed rooted graphs of the next section is very difficult as the proofs rely on several, long model-theoretic papers of Courcelle. The results presented here apparently do not follow directly from any published result using MSOL techniques of which we are aware. An advantage of our methods is that they give an explicit algorithm.

If there exists a branch $\{u, v\} \in B$ in a tree-decomposition $(\{S_i | i \in I\}, T = (I, B))$ such that $S_u \subseteq S_v$, then we can contract the branch $\{u, v\}$ and obtain a smaller tree-decomposition with the same width.

We say that a tree-decomposition is *good* if for some $k \in \mathbb{N}$ we have

1. $|S_i| = k + 1$ for all $i \in I$, and
2. $|S_i \cap S_j| = k$ if $\{i, j\} \in B$.

Figure 3.3 illustrates an example of a good tree-decomposition of a graph. Given a tree-decomposition of a graph, we can construct a good tree-decomposition of the graph with the same width in polynomial time.

Bodlaender gives a linear time algorithm for finding tree-decompositions of minimum width of a graph of bounded tree-width [7].

Theorem 3.1.6 (Bodlaender 1996)**.** *For all $k \in \mathbb{N}$, there exists a linear-time algorithm in size of G that tests whether a given graph $G = (V, E)$ has tree-width at most k and, if so, outputs a tree-decomposition of G with width at most k.*

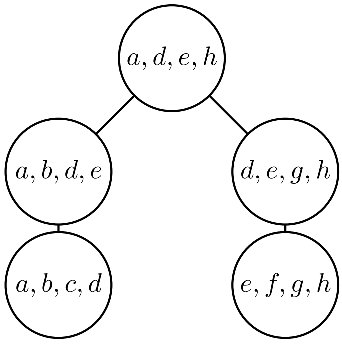

Figure 3.3: A good tree-decomposition of the graph given in Figure 3.1

Note that Bodlaender's algorithm can easily be modified to produce a good tree-decomposition of the input graph.

3.2 Rooted Graphs

In this section we deal exclusively with rooted graphs. The concept of a tree-decomposition and furthermore tree-width can naturally be defined for rooted graphs as opposed to unrooted graphs. The definitions remain the same and do not depend on the choice of root vertex. To simplify the presentation of our algorithm we borrow the concept of a nice tree-decomposition of a graph from [5], in which Blaser and Hoffman use it to construct a fast algorithm to evaluate the multivariate interlace polynomial of a graph of bounded tree-width. This will allow us to explain the intermediate steps of the algorithm in more detail. From now onwards assume that for every tree-decomposition $(\{S_i | i \in I\}, T = (I, B))$ of a rooted graph $G = (r, V, E)$ we let T be rooted with an arbitrarily chosen node ρ with $r \in S_\rho$ as the root and all branches directed away from ρ. If (i, j) is a branch of T then we say that i is the *parent* of j, and j is the *child* of i. If there is a directed path from node i to node j then we say j is a *descendant* of i.

Definition 3.2.1 (Nice Tree-Decomposition)**.** Let $G = (r, V, E)$ be a rooted graph. A *nice tree-decomposition* of G is a tree-decomposition $\tau = (\{S_i | i \in I\}, T = (I, B))$ of G with T rooted as described above, such that

(ND1) every node $i \in I$ must be one of the following types:

- Leaf: node i is a leaf of T.

- Join: node i has exactly two child nodes j and k in T and $S_i = S_j = S_k$.
- Introduce: node i has exactly one child j in T, and there is a vertex $a \in V - S_j$ with $S_i = S_j \cup a$.
- Forget: node i has exactly one child j in T, and there is a vertex $a \in V - S_i$ with $S_j = S_i \cup a$.

(ND2) for every node $i \in I$ which isn't a forget node, there exists a leaf l of T such that $S_i = S_l$.

Figure 3.4 illustrates an example of a nice tree-decomposition of a rooted graph.

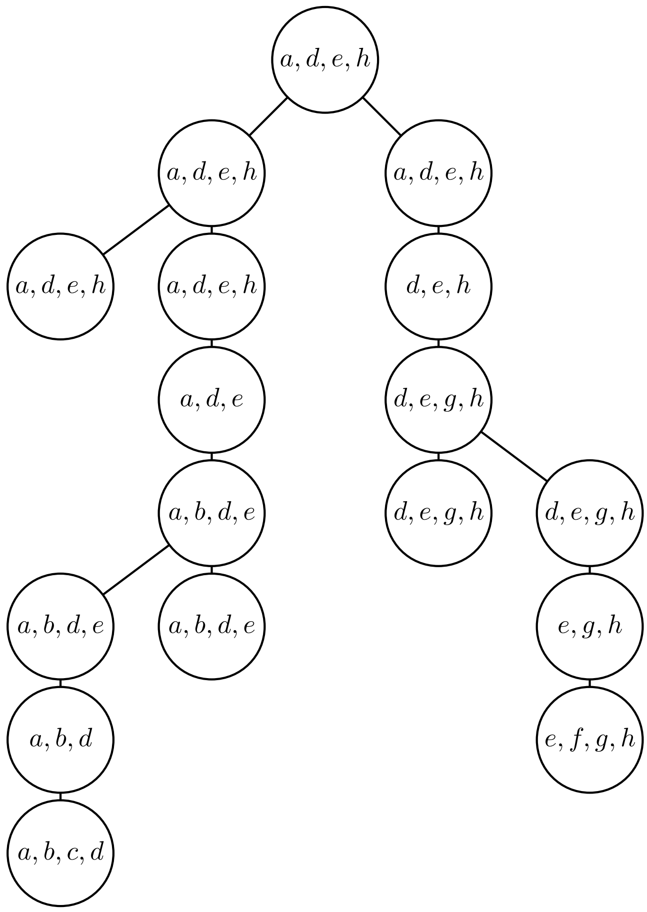

Figure 3.4: A nice tree-decomposition of the graph given in Figure 3.1 (suppose it is rooted at a)

Proposition 3.2.2. *Given a rooted graph $G = (r, V, E)$ of tree-width k, there exists a nice tree-decomposition $(\{S_i | i \in I\}, T = (I, B))$ of G of width k. Moreover, given a good tree-decomposition* *of G with width k, it is possible to construct a nice tree-decomposition of G with width k in time $O(nk)$.*

Proof. Given a rooted graph G of tree-width k, we can apply Bodlaender's algorithm to the underlying unrooted graph to find a good tree-decomposition of width k.

We show that $|I| = n - k$ for any good tree-decomposition $\tau = (\mathcal{S} = \{S_i | i \in I\}, T = (I, B))$ of an unrooted graph H of width k by using induction on $|I|$. When $|I| = 1$ we have one bag with $n = k + 1$ vertices in τ, so clearly $|I| = n - k$. When $|I| > 1$ let l be a leaf of T with parent node l'. Since τ is good we have $|S_l - S_{l'}| = 1$. Using $(TD3)$ the unique member of $S_l - S_{l'}$ is not contained in any other bag in τ. Let τ' be obtained from τ by deleting l from T and removing S_l from the collection $\mathcal{S}$.

Now τ' is a good tree-decomposition of $H - (S_l - S_{l'})$ with $w(\tau') = k$ and $|I| - 1$ nodes. Using induction we have $|I| - 1 = n - |S_l - S_{l'}| - k$, so $|I| = n - 1 - k + 1 = n - k$ as required.

We have shown that we can use Bodlaender's algorithm to return a good tree-decomposition $\tau = (\mathcal{S} = \{S_i | i \in I\}, T = (I, B))$ of width k and $n - k$ nodes.

Now choose ρ with $r \in S_\rho$ to be the root of T and orient the branches away from ρ. We now modify τ to get a new tree-decomposition $\tau' = (\mathcal{S}' = \{S'_i | i \in I'\}, T' = (I', B'))$ where T' is a rooted binary tree and for every $S_i \in \mathcal{S}$ there is a leaf l of T' such that $S_i = S'_l$.

Suppose we have a node 0 in I with children $1, 2, \ldots, d$ and corresponding bags $S_0, S_1, \ldots, S_d$. Replace the node 0 in I by nodes $0_1, 0_2, \ldots, 0_{d+1}$ such that nodes i and 0_{i+1} are the children of 0_i for $1 \leq i \leq d$, and 0_1 is the child of the parent of 0 if it exists. Let $S'_{0_i} = S_0$ and $S'_i = S_i$ for all $1 \leq i \leq d$. We apply this procedure to every node in I. Notice that in this procedure we double the number of branches. To see this note that the total outdegree of nodes replacing node 0 is twice the outdegree of 0 in τ. Hence τ' has $2(n - k) - 1$ nodes.

We have constructed a tree-decomposition $\tau' = (\mathcal{S}' = \{S'_i | i \in I'\}, T' = (I', B'))$ of H such that T' is a rooted tree in which every node has at most two children and for every distinct bag $S'_i \in \mathcal{S}'$ there exists a leaf node $l \in I'$ such that $S'_i = S'_l$ and $w(\tau') = k$.

We now modify τ' by inserting 'introduce' and 'forget' nodes to form a nice tree-decomposition of H. Suppose we have a node j in T' with parent node i such that $S'_i \neq S'_j$. Let $S'_i - S'_j = a$ and $S'_j - S'_i = b$. Insert nodes $\hat{\imath}, \hat{\jmath}$ into the branch (i, j) so that $i\hat{\imath}\hat{\jmath}j$ is a directed path from i to j. Let $S'_{\hat{\imath}} = S'_i$ and $S'_{\hat{\jmath}} = S'_j - b$. Note $S'_{\hat{\imath}} - a = S'_{\hat{\jmath}}$. This procedure adds $2(n - k - 1)$ nodes to T' since we add 2 nodes for every branch of T.

Hence the final tree-decomposition has at most $4(n-k)-3$ nodes. These steps can clearly be done in $O(nk)$ time.

The procedures discussed in this proof to construct a nice tree-decomposition of a rooted graph from a good tree-decomposition of the graph do not increase the width since we only ever add bags of size k or $k+1$. $\square$

Consider a nice tree-decomposition $(\{S_i | i \in I\}, T = (I, B))$ of a rooted graph $G = (r, V, E)$ with root node ρ. For each node $i \in I$ we let

$$V_i = \bigcup_{j \leq i} S_j$$

where the union is taken over every leaf j such that $j = i$ or j is a descendant of i in T.

By $(ND2)$ it is possible to partition the edges of G amongst the leaves of T such that an edge $\{x, y\} \in E$ is associated to a leaf i with $\{x, y\} \subseteq S_i$. Let D_i be the set of edges associated to leaf i. For graphs with parallel edges we ensure that if two edges are in the same parallel class then they belong to the same set D_i. If $e \in D_i$ let $m(e)$ denote the size of the parallel class containing e. For each node $i \in I$ we let

$$E_i = \bigcup_{j \leq i} D_j \tag{3.2}$$

where the union is again taken over every leaf j such that $j = i$ or j is a descendant of i in T.

Therefore for every node $i \in I$ in the nice tree-decomposition we have a corresponding subgraph $G_i = (V_i, E_i)$ and $G = \bigcup_{i \in I} G_i$.

Noble's algorithm begins by finding all partitions of a bag S_i, for some leaf node i, induced by subsets of edges of E_i. This allows it to know about the connectivity of each possible subgraph of G_i. Not only will we also need to know about the connectivity of our rooted subgraphs, but we will need to distinguish which vertices, if any, are connected to the root. For this we introduce the following definition of a state of a set which partitions the set into what we call "parts".

Definition 3.2.3 (State)**.** A *state* α of a set S is a partition of S with one distinguished part $B_0(\alpha)$ which, with a slight abuse of terminology, may be empty. We denote the other parts, which will always be non-empty, by $B_1(\alpha), \ldots, B_t(\alpha)$. Let $|\alpha|$ denote the number of parts of α not counting $B_0(\alpha)$, i.e. $|\alpha| = t$.

Let $\Upsilon(S)$ be the set of all states of S.

In the following example to differentiate between the parts, we place a "$\times$" adjacent to the part B_0. If there is no $\times$ then this signifies that B_0 is empty.

Example 3.2.4. *The states of the set $S = \{a, b, c\}$ are*

abc	$ab\vert c$	$ac\vert b$	$a\vert bc$	$a\vert b\vert c$
$abc^{\times}$	$ab\vert c^{\times}$	$ab^{\times}\vert c$	$ac\vert b^{\times}$	$ac^{\times}\vert b$
$a\vert bc^{\times}$	$a^{\times}\vert bc$	$a\vert b\vert c^{\times}$	$a\vert b^{\times}\vert c$	$a^{\times}\vert b\vert c$

Note that the set of partitions of a set S is a subset of the set of states of S. Below we present a formula to calculate the number of states of a set which will be useful when calculating the running time for the algorithm. Let $B(n)$ denote the n-th Bell number of a set with n elements.

Lemma 3.2.5. *The number of states of a set with t elements is $B(t + 1)$*

Proof. Let $\Pi(S)$ denote the set of all partitions of a set S. For $x \notin S$ we define $h : \Upsilon(S) \to \Pi(S \cup x)$ to be the function mapping the state α of S to a partition α' of $S \cup x$ formed by adding x to $B_0(\alpha)$. This is obviously a one-to-one correspondence between the states of S and the partitions of $S \cup x$. $\square$

Example 3.2.6. *In our previous example we had $S = \{a, b, c\}$. Therefore the number of states of S is $B(4) = 15$.*

We now give an informal idea of the role of states. The algorithm works up the tree from the leaves doing some computations on G_i only when the corresponding computations at the children of i have been done. Each subset A of the edges of G_i induces a partition of the vertices in S_i given by the connected components of $G_i|A$. We shall see that the contribution of A to $T(G; x, y)$ depends only on certain information concerning A, including this partition rather than the precise edges comprising A.

Given a rooted graph G, $S \subseteq V(G)$, $A \subseteq E(G)$ and $B_0 \subseteq S$, we will now define a state $\alpha(S, A, B_0)$ of S.

Let $C_1, \ldots, C_t$ be the connected components of $G|A$. Let $C_i' = V(C_i) \cap S$. If there exists i such that $C_i' \cap B_0$ and $C_i' \cap (S - B_0)$ are both non-empty, i.e. if there is a path in $G|A$ from a vertex in B_0 to a vertex in $S - B_0$, then $\alpha(S, A, B_0)$ is undefined. Otherwise let $B_0(\alpha) = B_0$ and $B_1(\alpha), \ldots, B_s(\alpha)$ be the sets C_i' that are non-empty and are contained in $S - B_0$. If $\alpha = \alpha(S, A, B_0)$ then let $f(A, \alpha)$

be the number of vertices in $V(G) - S$ that are not connected to a vertex in $S - B_0$ in $G|A$, and let $F(A, \alpha)$ be the set of such vertices. Now let $g(A, \alpha) = |A| - |V(G)| + f(A, \alpha) + |B_0| + |\alpha|$. Suppose without loss of generality that for $1 \leq i \leq s$, C_i is the connected component of $G|A$ such that $B_i(\alpha) = V(C_i) \cap S$. We claim that $g(A, \alpha)$ counts the sum over i of $|E(C_i)| - (|V(C_i)| - 1)$ and the number of edges that do not have both endpoints in $V(C_i)$ for $1 \leq i \leq s$. Note that

$$\begin{aligned} |V(G)| &= \sum_{i=1}^{s} |V(C_i)| + f(A, \alpha) + |B_0| \\ &= \sum_{i=1}^{s} (|V(C_i)| - 1) + f(A, \alpha) + |B_0| + |\alpha|. \end{aligned}$$

Hence

$$\begin{aligned} &\sum_{i=1}^{s} (|E(C_i)| - (|V(C_i)| - 1)) + |A| - \sum_{i=1}^{s} |E(C_i)| \\ &= |A| - |V(G)| + f(A, \alpha) + |B_0| + |\alpha| = g(A, \alpha). \end{aligned}$$

As $|E(C_i)| - (|V(C_i)| - 1) \geq 0$ for all i, it follows that $g(A, \alpha) \geq 0$. Informally $g(A, \alpha)$ is the number of edges that can be removed from A without changing the state α.

For each node i in a nice tree-decomposition of a rooted graph G and $\alpha \in \Upsilon(S_i)$, we define $T(G_i, \alpha; x, y)$ by

$$T(G_i, \alpha; x, y) = \sum_{\substack{A \subseteq E_i: \\ \alpha(S_i, A, B_0(\alpha)) = \alpha}} (x - 1)^{f(A, \alpha)} (y - 1)^{g(A, \alpha)}. \tag{3.3}$$

Note that we are summing over all subsets of edges of E_i that induce the state α of S_i.

We will now describe the computation done at a leaf node in our algorithm. Note that for a leaf node i in T we have $V(G_i) = S_i$, hence $f(A, \alpha) = 0$ for all $A \subseteq E_i$ and $\alpha \in \Upsilon(S_i)$. Let $\hat{E}_i$ contain one representative from each parallel class of edges contained in E_i. Therefore for each leaf node i in T, when $y \neq 1$ we can express $T(G_i, \alpha; x, y)$ in the form

$$\begin{aligned} T(G_i, \alpha; x, y) &= \sum_{\substack{A \subseteq \hat{E}_i: \\ \alpha(S_i, A, B_0(\alpha)) = \alpha}} (y - 1)^{|A| - |V(G_i)| + |B_0(\alpha)| + |\alpha|} \prod_{e \in A} \left(\sum_{j=1}^{m(e)} \binom{m(e)}{j} (y - 1)^{j-1} \right) \\ &= \sum_{\substack{A \subseteq \hat{E}_i: \\ \alpha(S_i, A, B_0(\alpha)) = \alpha}} (y - 1)^{-|V(G_i)| + |B_0(\alpha)| + |\alpha|} \prod_{e \in A} \left(y^{m(e)} - 1 \right). \end{aligned}$$

For the special case when $y = 1$ we have

$$T(G_i, \alpha; x, y) = \sum_{\substack{A \subseteq \hat{E}_i: \\ \alpha(S_i, A, B_0(\alpha)) = \alpha, \\ |A| - |V(G_i)| + |B_0(\alpha)| + |\alpha| = 0}} \prod_{e \in A} m(e).$$

If $m(e) = 1$ for all $e \in E_i$ then we simply have

$$T(G_i, \alpha; x, y) = \sum_{\substack{A \subseteq E_i: \\ \alpha(S_i, A, B_0(\alpha)) = \alpha}} (y-1)^{|A| - |V(G_i)| + |B_0(\alpha)| + |\alpha|}.$$

Suppose we have a tree-decomposition of a graph such that node i has children nodes j and k. There is a step in Noble's algorithm which finds all partitions of S_i that can occur by combining the partitions of S_j and S_k in the following way.

Definition 3.2.7 (Partition Join)**.** Consider two partitions π_1 and π_2 of a set S. Their *partition join* $\pi_1 \vee \pi_2$ is defined to be the partition of S for which the parts are minimal sets such that if two elements are in the same part of π_1 or π_2, then they are in the same part of $\pi_1 \vee \pi_2$.

Obviously the partition join is commutative and associative. We will now incorporate this into our definition on how to join two states. We say that two states α_1 and α_2 of a set S are *compatible* if $B_0(\alpha_1) = B_0(\alpha_2)$, and we denote their compatibility by $\alpha_1 \sim \alpha_2$.

Definition 3.2.8 (State Join)**.** The join of two compatible states α_1 and α_2 of a set S is given by $\alpha_1 \vee \alpha_2$ where the parts are labelled to ensure that $B_0(\alpha_1 \vee \alpha_2) = B_0(\alpha_1) = B_0(\alpha_2)$.

Example 3.2.9. *Let $\alpha_1 = ab|cd|e|f|gh^\times$ and $\alpha_2 = a|bcde|f|gh^\times$ be two states of the set $S = \{a, b, c, d, e, f, g, h\}$. Clearly $\alpha_1 \sim \alpha_2$ since $B_0(\alpha_1) = \{g, h\} = B_0(\alpha_2)$. Their state join is given by*

$$\alpha_1 \vee \alpha_2 = abcde|f|gh^\times.$$

The following analysis will allow us to describe the computation done at a join node in our algorithm.

Lemma 3.2.10. *Let i be a join node in a nice tree-decomposition with children nodes j and k. Let $A_j \subseteq E_j$, $A_k \subseteq E_k$ and $A_i = A_j \cup A_k$. Let B_0 be a subset of S_i. Suppose that the states $\alpha_j = \alpha(S_i, A_j, B_0)$ and $\alpha_k = \alpha(S_i, A_k, B_0)$ are defined. Then $\alpha = \alpha(S_i, A_i, B_0)$ is defined and*

1. $\alpha = \alpha_j \vee \alpha_k$*,*

2. $f(A_i, \alpha) = f(A_j, \alpha_j) + f(A_k, \alpha_k)$*, and*

3. $g(A_i, \alpha) = g(A_j, \alpha_j) + g(A_k, \alpha_k) + |\alpha| - |\alpha_j| - |\alpha_k| + |S_i| - |B_0|$*.*

Proof. Neither $G_i|A_j$ nor $G_i|A_k$ has a component containing vertices of both B_0 and $S_i - B_0$, so $G_i|A_i$ has no such component. Consequently α is defined.

1. Construct graphs H_j and H_k both having vertex set $S_i - B_0$ and such that vw is an edge of H_j if v and w are connected in $G_i|A_j$, and similarly for H_k. Thus H_j and H_k are both disjoint unions of cliques. The parts of α_j and α_k correspond to the vertex sets of the cliques of H_j and H_k respectively.

   Now let $H_i = H_j \cup H_k$. Then v and w are connected in H_i if and only if they are connected in $G_i|(A_j \cup A_k)$. Thus the parts of α other than B_0 correspond to the connected components of $G_i|(A_j \cup A_k)$ and are exactly the parts of $\alpha_j \vee \alpha_k$ other than B_0.

2. Let $S = S_i = S_j = S_k$. We claim that $(V_j - S) \cap (V_k - S) = \emptyset$. Suppose otherwise that there exists a vertex $v \in (V_j - S) \cap (V_k - S)$. Then v must be in the bags corresponding to a descendant node of j and a descendant node of k in the tree-decomposition. By (TD3) v must then also be in S, which is a contradiction. Therefore $F(A_j, \alpha_j) \cap F(A_k, \alpha_k) = \emptyset$.

   We now prove $F(A_i, \alpha) \subseteq F(A_j, \alpha_j) \cup F(A_k, \alpha_k)$. Let $v \in F(A_i, \alpha)$, then $v \in V_j - S$ or $v \in V_k - S$. Suppose without loss of generality $v \in V_j - S$. We know that $G_j|A_j \subseteq G_i|A_i$ so if v is not connected to $S - B_0$ in $G_i|A_i$ then it is not connected to $S - B_0$ in $G_j|A_j$. Therefore $v \in F(A_j, \alpha_j)$.

   We now prove $F(A_i, \alpha) \supseteq F(A_j, \alpha_j) \cup F(A_k, \alpha_k)$. Let $v \in F(A_j, \alpha_j) \cup F(A_k, \alpha_k)$. Suppose without loss of generality $v \in F(A_j, \alpha_j)$ and that there is a path from v to $S - B_0$ in $G_i|A_i$. Then there is a path $v \dots v_p v_q \dots s$ with $v_p \in V_j - S$, $v_q \in V_k - S$ and $s \in S - B_0$. This means that $\{v_p, v_q\} \subseteq S_t$ for some $t \in I$. However $t \neq i$ because neither v_p nor v_q is a member of S. Furthermore t cannot be a descendant of j because (TD3) would imply that $v_q \in S$. Similarly t cannot be a descendant of k. Hence we have a contradiction. Therefore $v \in F(A_i, \alpha)$. Since $F(A_j, \alpha_j) \cap F(A_k, \alpha_k) = \emptyset$ and $F(A_i, \alpha) = F(A_j, \alpha_j) \cup F(A_k, \alpha_k)$ we have $f(A_i, \alpha) = f(A_j, \alpha_j) + f(A_k, \alpha_k)$.

3. We have $g(A_i, \alpha) = |A_i| - |V(G_i)| + f(A_i, \alpha) + |B_0| + |\alpha|$ by definition. Now since $|V(G_i)| = |V(G_j)| + |V(G_k)| - |S_i|$ and $|A_i| = |A_j| + |A_k|$,

$$g(A_i, \alpha) = |A_j| + |A_k| - (|V(G_j)| + |V(G_k)| - |S_i|) + f(A_i, \alpha) + |B_0| + |\alpha|. \tag{3.4}$$

Using part 2, Equation 3.4 equals

$$|A_j| + |A_k| - |V(G_j)| - |V(G_k)| + |S_i| + f(A_j, \alpha_j) + f(A_k, \alpha_k) + |B_0| + |\alpha|.$$

Finally we can deduce the following equation using the formulae for $g(A_j, \alpha_j)$ and $g(A_k, \alpha_k)$:

$$g(A_i, \alpha) = g(A_j, \alpha_j) + g(A_k, \alpha_k) + |\alpha| - |\alpha_j| - |\alpha_k| + |S_i| - |B_0|.$$

□

Let i be a join node in T with children nodes j and k. For $\alpha, \alpha_j, \alpha_k \in \Upsilon(S_i)$, by Lemma 3.2.10 we can express $T(G_i, \alpha; x, y)$ in the form

$$\begin{aligned}
&T(G_i, \alpha; x, y) \\
&= \sum_{\substack{A_i \subseteq E_i: \\ \alpha(S_i, A_i, B_0(\alpha)) = \alpha}} (x-1)^{f(A_i,\alpha)}(y-1)^{g(A_i,\alpha)} \\
&= \sum_{\substack{A_j \subseteq E_j, A_k \subseteq E_k: \\ \alpha(S_i, A_j \cup A_k, B_0(\alpha)) = \alpha}} (x-1)^{f(A_j \cup A_k,\alpha)}(y-1)^{g(A_j \cup A_k,\alpha)} \\
&= \sum_{\substack{\alpha_j, \alpha_k: \\ \alpha_j \sim \alpha_k, \\ \alpha_j \vee \alpha_k = \alpha}} \sum_{\substack{A_j \subseteq E_j: \\ \alpha(S_j, A_j, B_0(\alpha_j)) = \alpha_j}} \sum_{\substack{A_k \subseteq E_k: \\ \alpha(S_k, A_k, B_0(\alpha_k)) = \alpha_k}} (x-1)^{f(A_j,\alpha_j)+f(A_k,\alpha_k)}(y-1)^{g(A_j \cup A_k,\alpha)} \\
&= \sum_{\substack{\alpha_j, \alpha_k: \\ \alpha_j \sim \alpha_k, \\ \alpha_j \vee \alpha_k = \alpha}} T(G_j, \alpha_j; x, y) T(G_k, \alpha_k; x, y)(y-1)^{|\alpha| - |\alpha_j| - |\alpha_k| + |S_i| - |B_0(\alpha)|}.
\end{aligned}$$

We now describe the computation done at a forget node in our algorithm. We let $\alpha \setminus a$ denote the state obtained by deleting a in α and removing any empty part created if a is a singleton part of α other than $B_0(\alpha)$.

Lemma 3.2.11. *Let i be a forget node in T and j be the child of i. Let a be the unique element of*

$S_j - S_i$. *Then for* $\beta \in \Upsilon(S_i)$ *we have*

$$T(G_i, \beta; x, y) = \sum_{\substack{\alpha: \\ \alpha \setminus a = \beta, \\ |\alpha| = |\beta|}} T(G_j, \alpha; x, y)(x-1)^{h(\alpha)} \tag{3.5}$$

where $h(\alpha) = 1$ *if* $a \in B_0(\alpha)$ *and* $h(\alpha) = 0$ *otherwise.*

Proof. Suppose $A \subseteq E_i$ is such that $\beta = \alpha(S_i, A, B_0(\beta))$ and $G_i|A$ has connected components $C_1, \ldots, C_t$. Without loss of generality let $a \in V(C_a)$ where $1 \leq a \leq t$.

- If $V(C_a) \cap (S_i - B_0(\beta)) = \emptyset$ then a is not connected to any vertex of $S_i - B_0(\beta)$ in $G_i|A$, so $a \in F(A, \beta)$. Let α be the state of S_j with $B_s(\alpha) = B_s(\beta)$ for $s \neq 0$ and $B_0(\alpha) = B_0(\beta) \cup a$. Clearly $|\alpha| = |\beta|$ and $\alpha \setminus a = \beta$. Since $G_j = G_i$ the graph $G_j|A$ has the same connected components as $G_i|A$. Hence $\alpha = \alpha(S_j, A, B_0(\alpha))$.

  Now $F(A, \beta) = F(A, \alpha) \cup a$ so we have $f(A, \beta) = f(A, \alpha) + 1$. Also $|\beta| = |\alpha|$ and $|B_0(\beta)| = |B_0(\alpha)| - 1$ so $g(A, \beta) = g(A, \alpha)$.

- If $V(C_a) \cap (S_i - B_0(\beta)) \neq \emptyset$ then a is connected to some other vertex b of $S_i - B_0(\beta)$ in $G_i|A$. Let $B_b(\beta)$ be the part containing b in β. Let α be the state of S_j with $B_s(\alpha) = B_s(\beta)$ for $s \neq b$ and $B_b(\alpha) = B_b(\beta) \cup a$. Again it is clear to see that $|\alpha| = |\beta|$, $\alpha \setminus a = \beta$ and that $G_j|A$ has the same connected components as $G_i|A$. Hence $\alpha = \alpha(S_j, A, B_0(\alpha))$.

  Now $F(A, \beta) = F(A, \alpha)$ so we have $f(A, \beta) = f(A, \alpha)$. Also $|\beta| = |\alpha|$ and $|B_0(\beta)| = |B_0(\alpha)|$ so $g(A, \beta) = g(A, \alpha)$.

We have shown that for $A \subseteq E_i$ with $\beta = \alpha(S_i, A, B_0(\beta))$, there exists $\alpha = \alpha(S_j, A, B_0(\alpha))$ such that $\alpha \setminus a = \beta$ and $|\alpha| = |\beta|$.

Now suppose instead we take $A \subseteq E_j$ with $\alpha = \alpha(S_j, A, B_0(\alpha))$ such that $\alpha \setminus a = \beta$ and $|\alpha| = |\beta|$. The last condition ensures that a is not a part of α unless $B_0(\alpha) = a$. We have $G_i = G_j$ so the connected components of $G_i|A$ are the same as the connected components of $G_j|A$. Hence $\beta = \alpha(S_i, A, B_0(\beta))$. We have shown that every term in the expansion of $T(G_i, \beta; x, y)$ appears as a term in the expansion of one of the summands on the right-hand-side of Equation 3.5, and vice-versa. $\square$

We now describe the computation done at an introduce node in our algorithm.

Definition 3.2.12 (Introduce States)**.** Let S be a set with $a \in S$ and $\alpha \in \Upsilon(S)$. We say that α is an *introduce state* of a in S if:

- $a \in B_0(\alpha)$, or
- $a \notin B_0(\alpha)$ and a is in a singleton part in α.

We denote the set of introduce states of a in S by $I(S, a)$.

Lemma 3.2.13. *Let i be an introduce node in T and j be the child node of i. Let a be the unique element of $S_i - S_j$. We can express $T(G_i, \alpha; x, y)$ in the form*

$$T(G_i, \alpha; x, y) = \begin{cases} T(G_j, \alpha \setminus a; x, y) & \text{if } \alpha \in I(S_i, a); \\ 0 & \text{otherwise.} \end{cases}$$

Proof. By (TD3) a cannot be in any bag corresponding to a descendant node of i in T. Therefore any edges incident to a in G cannot be in E_i. This means that a must be an isolated vertex in $G_i|A$ for any $A \subseteq E_i$. So if $\alpha \notin I(S_i, a)$ then $T(G_i, \alpha; x, y) = 0$.

If $\alpha \in I(S_i, a)$ and $A \subseteq E_i$ then $\alpha = \alpha(S_i, A, B_0(\alpha))$ if and only if $\alpha \setminus a = \alpha(S_j, A, B_0(\alpha) - a)$. Moreover $F(A, \alpha) = F(A, \alpha \setminus a)$ and

$$\begin{aligned} g(A, \alpha) &= |A| - |V(G_i)| + f(A, \alpha) + |B_0(\alpha)| + |\alpha| \\ &= |A| - (|V(G_j)| + 1) + f(A, \alpha \setminus a) + |B_0(\alpha) - a| + |\alpha \setminus a| + 1 \\ &= g(A, \alpha \setminus a). \end{aligned}$$

□

We now describe the computation done in our algorithm after we have computed $T(G_\rho, \alpha; x, y)$ for all $\alpha \in \Upsilon(S_\rho)$.

Lemma 3.2.14. *If $G = (r, V, E)$ is connected then the Tutte polynomial of G is given by*

$$T(G; x, y) = \sum_{\alpha : |\alpha| = 1} T(G_\rho, \alpha; x, y)(x - 1)^{|B_0(\alpha)|}. \tag{3.6}$$

Proof. Let $C_1, \ldots, C_t$ be the connected components of $G|A$ for some $A \subseteq E$. Let C_1 be the component containing the root vertex.

We claim that every subset A of E contributes to precisely one of the terms in the summation on the right-hand side of Equation 3.6. To see this note that the set A contributes to the term corresponding to the state α with $B_0(\alpha) = \bigcup_{s \neq 1}(V(C_s) \cap S_\rho)$ and $B_1(\alpha) = V(C_1) \cap S_\rho$. Any other state α satisfying $\alpha = \alpha(G_\rho, A, B_0(\alpha))$ would have $|\alpha| \geq 2$.

Since $G = G_\rho$ the connected components of $G_\rho|A$ are the same as the connected components of $G|A$.

Now since $|V(G_\rho)| = |V(C_1)| + f(A, \alpha) + |B_0(\alpha)|$ we have

$$\begin{aligned} \rho(G) - \rho(A) &= (|V(G)| - 1) - (|V(C_1)| - 1) \\ &= |V(G_\rho)| - (|V(G_\rho)| - f(A, \alpha) - |B_0(\alpha)|) \\ &= f(A, \alpha) + |B_0(\alpha)|, \end{aligned}$$

and

$$\begin{aligned} |A| - \rho(A) &= |A| - (|V(C_1)| - 1) \\ &= |A| - (|V(G_\rho)| - f(A, \alpha) - |B_0(\alpha)| - 1) \\ &= |A| - |V(G_\rho)| + f(A, \alpha) + |B_0(\alpha)| + |\alpha| \\ &= g(A, \alpha). \end{aligned}$$

□

Combining Lemmas 3.2.11, 3.2.13 and 3.2.14 and the discussion on the computations done at a leaf and join node, we see that Algorithm 1 correctly computes $T(G; x, y)$ when G is connected. If G is not connected then every edge which does not lie in the connected component containing the root is a loop in the greedoid $\Gamma(G)$. Thus if G is a rooted graph, G' is the connected component containing the root and L is the number of edges in components other than G', then

$$T(G; x, y) = y^L T(G'; x, y). \tag{3.7}$$

Example 3.2.15 (Evaluating $T(G; x, y)$)**.** *In Figure 3.5 we have a rooted connected graph $G = (a, V, E)$ with a nice tree-decomposition $\tau = (\{S_i | i \in I\}, T = (I, B))$ of G of width 2 with root node ρ and the corresponding partition $D_1 = \{1, 4, 7\}$ and $D_2 = \{2, 3, 5, 7\}$. In what follows if a state α*

Algorithm 1 Evaluating $T(G;x,y)$ where G is a rooted graph.

Require: $G=(r,V,E)$ a rooted connected graph of tree width at most k; $x,y\in\mathbb{Q}$; $\tau=(\{S_i,i\in I\},T=(I,B))$ a nice tree-decomposition of G; $\{D_i|i$ is a leaf node of $\tau\}$ the corresponding partition of E; $m:E\to\mathbb{N}$.

$\rho\leftarrow$ the root node of T
$T_\rho\leftarrow T$
$z\leftarrow\max_{e\in E}\{m(e)\}$
Compute $y,\dots,y^z$
while $T_\rho\neq\emptyset$ **do**
  $i\leftarrow$ a node at the greatest depth in T_ρ
  if i is a leaf in T **then**
    while $\alpha\in\Upsilon(S_i)$ **do**
      if $y\neq 1$ **then**
        $T(G_i,\alpha;x,y)\leftarrow\sum_{\substack{A\subseteq\hat{E}_i:\\ \alpha(S_i,A,B_0(\alpha))=\alpha}}(y-1)^{-|V(G_i)|+|B_0(\alpha)|+|\alpha|}\prod_{e\in A}\left(y^{m(e)}-1\right)$
      else
        $T(G_i,\alpha;x,y)\leftarrow\sum_{\substack{A\subseteq\hat{E}_i:\\ \alpha(S_i,A,B_0(\alpha))=\alpha}}(y-1)^{|A|-|V(G_i)|+|B_0(\alpha)|+|\alpha|}\prod_{e\in A}m(e)$
      end if
    end while
  end if
  if i is a forget node in T **then**
    $j\leftarrow$ the child of i
    $a\leftarrow$ the unique element of S_j-S_i
    while $\alpha\in\Upsilon(S_i)$ **do**
      $T(G_i,\alpha;x,y)\leftarrow\sum_{\substack{\beta:\\ \beta\setminus a=\alpha,\\ |\beta|=|\alpha|}}T(G_j,\beta;x,y)(x-1)^{h(\beta)}$
    end while
  end if
  if i is an introduce node in T **then**
    $j\leftarrow$ the child of i
    $a\leftarrow$ the unique element of S_i-S_j
    while $\alpha\in I(S_i,a)$ **do**
      $T(G_i,\alpha;x,y)\leftarrow T(G_j,\alpha\setminus a;x,y)$
    end while
  end if
  if i is a join node in T **then**
    $\{j,k\}\leftarrow$ the children of i
    while $\alpha\in\Upsilon(S_i)$ **do**
      $T(G_i,\alpha;x,y)\leftarrow\sum_{\substack{\alpha_j,\alpha_k:\\ \alpha_j\sim\alpha_k,\\ \alpha_j\vee\alpha_k=\alpha}}T(G_j,\alpha_j;x,y)T(G_k,\alpha_k;x,y)(y-1)^{|\alpha|-|\alpha_j|-|\alpha_k|+|S_i|-|B_0(\alpha)|}$
    end while
  end if
  delete i
end while
$T(G;x,y)\leftarrow\sum_{\substack{\alpha:\\ |\alpha|=1}}T(G_\rho,\alpha;x,y)(x-1)^{|B_0(\alpha)|}$
return $T(G;x,y)$

and the value of $T(G, \alpha; x, y)$ are not listed in a table below, then $T(G, \alpha; x, y) = 0$.

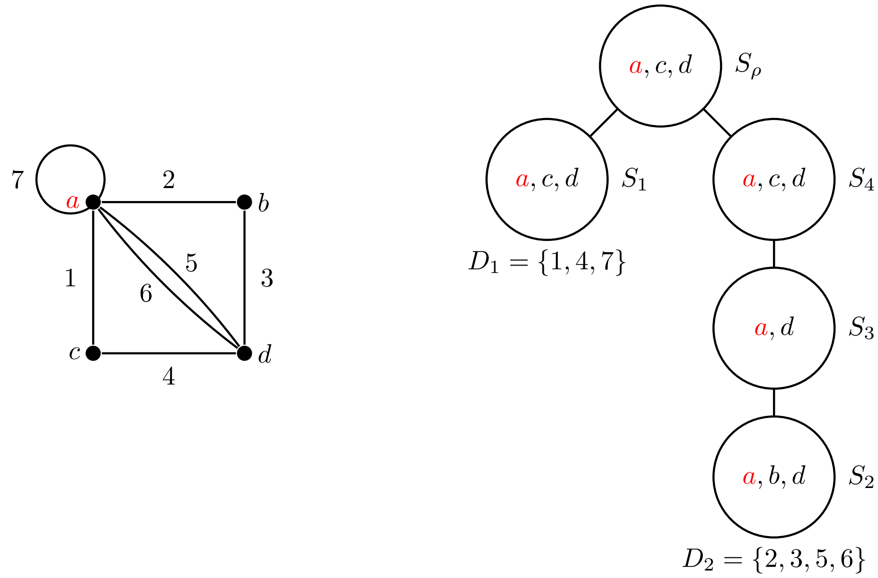

Figure 3.5: Rooted graph G with a corresponding nice tree-decomposition τ

We begin by computing $T(G_2, \alpha; x, y)$ for the leaf node 2 in T for all $\alpha \in \Upsilon(S_2)$.

α	**abd**	**ab|d**	**a|bd**	**ad|b**	**a|b|d**
$T(G_2, \alpha; x, y)$	$y^2 + 2y + 2$	1	1	$y + 1$	1
α	**ab|d**$^\times$	**a|bd**$^\times$	**ad|b**$^\times$	**a|b|d**$^\times$	**a|d|b**$^\times$
$T(G_2, \alpha; x, y)$	1	y	$y + 1$	1	1

Delete node 2 from T_ρ. We now compute $T(G_3, \beta; x, y)$ for the forget node 3 in T for all $\beta \in \Upsilon(S_3)$.

β	**ad**	**a|d**	**a|d**$^\times$
$T(G_3, \beta; x, y)$	$y^2+xy+x+y+1$	$x + 1$	$xy - y + 1$

Delete node 3 from T_ρ. We now compute $T(G_4, \gamma; x, y)$ for the introduce node 4 in T for all $\gamma \in I(S_4, c)$.

γ	**ad|c**	**ad|c**$^{\times}$	**a|d|c**
$T(G_4, \gamma; x, y)$	$y^2+xy+x+y+1$	$y^2+xy+x+y+1$	$x+1$
γ	**a|d|c**$^{\times}$	**a|c|d**$^{\times}$	**a|cd**$^{\times}$
$T(G_4, \gamma; x, y)$	$x+1$	$xy-y+1$	$xy-y+1$

Delete node 4 from T_ρ. We now compute $T(G_1, \delta; x, y)$ for the leaf node 1 in T for all $\delta \in \Upsilon(S_1)$.

δ	**acd**	**ac|d**	**a|cd**	**a|c|d**
$T(G_1, \delta; x, y)$	y	y	y	y
δ	**ac|d**$^{\times}$	**a|cd**$^{\times}$	**a|c|d**$^{\times}$	**a|d|c**$^{\times}$
$T(G_1, \delta; x, y)$	y	y^2	y	y

Delete node 1 from T_ρ. We now compute $T(G_\rho, \epsilon; x, y)$ for the join node ρ in T for all $\epsilon \in \Upsilon(S_\rho)$.

ϵ	**acd**	**ad|c**	**a|c|d**$^{\times}$	**a|cd**	**ac|d**
$T(G_\rho, \epsilon; x, y)$	$y^4 + xy^3 + 2y^3 + 2xy^2+2xy+2y^2+2y$	$y^3 + xy^2 + xy + y^2 + y$	xy^2-y^2+y	$xy+y$	$xy+y$
ϵ	**ad|c**$^{\times}$	**a|cd**$^{\times}$	**ac|d**$^{\times}$	**a|c|d**	**a|d|c**$^{\times}$
$T(G_\rho, \epsilon; x, y)$	$y^3+xy^2+xy+y^2+y$	$xy^3-y^3+y^2$	xy^2-y^2+y	$xy+y$	$xy+y$

Delete node ρ from T_ρ. We now compute $T(G; x, y)$.

$$\begin{aligned}
T(G; x, y) &= y^4 + xy^3 + 2y^3 + 2xy^2 + 2xy + 2y^2 + 2y + (x-1)(y^3 + xy^2 + xy + y^2 + y) \\
&\quad + (x-1)(xy^2 - y^2 + y) + (x-1)^2(xy^3 - y^3 + y^2) \\
&= y^4 + x^3y^3 - 3x^2y^3 + 3x^2y^2 + x^2y + 5xy^3 - 2xy^2 + 3xy + 3y^2.
\end{aligned}$$

3.2.1 Complexity of the Algorithm

Here we calculate the time complexity of our algorithm, that is, the maximum running time for each input length. For a rooted graph $G = (r, V, E)$ we let $t(n, m, k, x, y, z)$ be the maximum number of operations required to evaluate $T(G; x, y)$ if $n = |V|$, $m = |E|$, $z = \max_{e \in E}\{m(e)\}$ and G has tree-width at most k. We let $\delta = \delta(n, m, k, x, y, z)$ denote the maximum time taken for one arithmetical operation during the algorithm. Let

$$h(k) = k^5(2k+1)^{(2k-1)}((4k+5)^{(4k+5)}(2^{(2k+5)}/3)^{(4k+5)})^{(4k+1)}.$$

We now compute the complexity of the four preprocessing steps whose outputs are required by our algorithm.

1. Finding a tree-decomposition of width at most k can be done in time $O(h(k)n)$ using the algorithm given in [7].

2. Constructing a nice tree-decomposition from a tree-decomposition of width at most k can be done in time $O(nk)$ by Proposition 3.2.2

3. Computing the partition $\{D_i, i \text{ is a leaf node in the nice tree-decomposition}\}$ can be done in time $O(m + nk^2)$.

4. Computing $y, \ldots, y^z$ can be done in time $O(z\delta)$ since there are z values, each taking the maximum time δ to compute.

Therefore the combined maximum running time of the preprocessing steps is

$$O(h(k)n + m + z\delta).$$

In order to check whether two states α_1 and α_2 of a set S are compatible, we need to check that $B_0(\alpha_1) = B_0(\alpha_2)$. Each of $B_0(\alpha_1)$ and $B_0(\alpha_2)$ have at most $k+1$ elements, so the time required to do this is $O(k^2)$. To determine whether $\alpha_1 = \alpha_2$ we check that if u and v are in the same part in α_1, then they are in the same part in α_2. This takes time $O(k^2)$ since there are at most $k+1$ parts in each state. Suppose we construct graphs H_1 and H_2, both having vertex set S such that two vertices are connected in H_1 if and only if they are in the same part in α_1, and similarly for H_2. Now to compute the state $\alpha_1 \vee \alpha_2$ we can construct the graph $H_1 \cup H_2$ and perform a breadth-first search to find the connected components of it. There will be at most $k+1$ connected components, and these components will be the parts of $\alpha_1 \vee \alpha_2$. These operations can be done in $O(k^2)$ time. Therefore joining two states in our algorithm takes an overall time of $O(k^2)$.

Clearly the time taken to run the main part of the algorithm, that is omitting the preprocessing steps, is $O(nt'(n,m,k,x,y,z))$ where $t' = t'(n,m,k,x,y,z)$ is the maximum time required to compute $T(G_i, \alpha; x, y)$ for a join node i where $\alpha \in \Upsilon(S_i)$. Recall that the number of states of a set with t elements is $B(t+1)$. For a join node we combine at most $(B(k+2))^2$ states, with the contribution from each pair taking at most $O(k^2 + \delta)$ time to compute by the preceding discussion. Computing the running time for the main part of our algorithm in terms of δ therefore takes time

$$O(n(k^2 + \delta)(B(k+2))^2).$$

The following analysis allows us to calculate δ. To add, subtract, multiply or divide two l-bit integers takes time $O(l \log l \log \log l)$ by [1]. Therefore we need to find the largest possible integer in our algorithm.

Recall that for a node i in a nice tree-decomposition and for some state $\alpha \in \Upsilon(S_i)$ we define

$$T(G_i, \alpha; x, y) = \sum_{\substack{A \subseteq E_i: \\ \alpha(S_i, A, B_0(\alpha)) = \alpha}} (x-1)^{f(A,\alpha)}(y-1)^{g(A,\alpha)}.$$

Let

$$x = \frac{p_x}{q_x} \quad \text{and} \quad y = \frac{p_y}{q_y},$$

where p_x, q_x, p_y, and q_y are integers such that p_x and q_x are coprime, and p_y and q_y are coprime. We have

$$\begin{aligned} T(G_i, \alpha; x, y) &= \sum_{\substack{A \subseteq E_i: \\ \alpha(S_i, A, B_0(\alpha)) = \alpha}} (x-1)^{f(A,\alpha)}(y-1)^{g(A,\alpha)} \\ &= \sum_{\substack{A \subseteq E_i: \\ \alpha(S_i, A, B_0(\alpha)) = \alpha}} \left(\frac{p_x - q_x}{q_x}\right)^{f(A,\alpha)} \left(\frac{p_y - q_y}{q_y}\right)^{g(A,\alpha)} \\ &= \frac{\displaystyle\sum_{\substack{A \subseteq E_i: \\ \alpha(S_i, A, B_0(\alpha)) = \alpha}} (p_x - q_x)^{f(A,\alpha)}(p_y - q_y)^{g(A,\alpha)} q_x^{|V_i| - |S_i| - f(A,\alpha)} q_y^{|E_i| - g(A,\alpha)}}{q_x^{|V_i| - |S_i|} q_y^{|E_i|}}. \end{aligned}$$

For the denominator we have $q_x^{|V_i|-|S_i|}q_y^{|E_i|} \leq |q_x|^n|q_y|^m$. For the numerator we have

$$\sum_{\substack{A\subseteq E_i:\\ \alpha(S_i,A,B_0(\alpha))=\alpha}} (p_x-q_x)^{f(A,\alpha)}(p_y-q_y)^{g(A,\alpha)}q_x^{|V_i|-|S_i|-f(A,\alpha)}q_y^{|E_i|-g(A,\alpha)}$$

$$\leq 2^m|p_x-q_x|^n|p_y-q_y|^m|q_x|^n|q_y|^m \leq (|p_x|+|q_x|+|p_y|+|q_y|+2)^{2n+3m}.$$

We have shown that $\delta \leq l\log l\log\log l$ where $l=(2n+3m)\log(|p_x|+|q_x|+|p_y|+|q_y|+2)$. The running time for the main part of the algorithm is therefore

$$O(n(k^2+(n+m)\log(|p_x|+|q_x|+|p_y|+|q_y|+2)\log l\log\log l)(B(k+2))^2).$$

Suppose that our input graph has no parallel edges. Assume that we have at most one loop at any vertex (any additional loop is considered to be a parallel edge). We now find the maximum number of edges in our graph G in terms of its tree-width k. Suppose we have a good tree-decomposition $(\{S_i|i\in I\}, T=(I,B))$ as defined in the proof of Proposition 3.2.2. We count the number of edges in G by working down the nodes of T from the root. There are at most $\dfrac{k(k+1)}{2}+(k+1)$ edges in G between the vertices that are in S_ρ. For every other node $i\in I$ in T there is precisely one vertex in S_i which does not appear in any of the bags corresponding to its parent node. So between the vertices in S_i there are at most $k+1$ edges which have not been previously counted. Since $|I|=n-k$ the total number of edges in G is at most

$$(|I|-1)(k+1)+\frac{k(k+1)}{2}+(k+1)=\frac{(k+1)(2n-k)}{2}.$$

Therefore if the input graph to our algorithm has no parallel edges, the total running time is

$$O(n(k^2+nk\log(|p_x|+|q_x|+|p_y|+|q_y|+2)\log l\log\log l)(B(k+2))^2).$$

3.3 Rooted Digraphs

For this section all definitions and terminology relating to tree-decompositions are the same as those in the previous section, however we now impose an orientation on every edge. This will not affect the time it takes to construct a nice tree-decomposition or the bound on the number of nodes it has.

When we discussed the partitioning of the edges of our graph amongst the leaf nodes in our nice tree-decomposition, we spoke about only processing one edge from each parallel class. This will still be the case, however a parallel class is now defined to be the set of edges that are directed from and to the same vertices. The size of a parallel class is still denoted by $m(e)$ for a directed edge e. We have a different definition of a state in the directed case.

Definition 3.3.1 (State)**.** A *state* α of a set S is a partial order $\leq$ in which the elements of the partial order correspond to disjoint non-empty subsets $B_1(\alpha), \ldots, B_t(\alpha)$ of S. We let $B_0(\alpha)$, which may be empty, denote $S - \bigcup_i B_i(\alpha)$. We will refer to these subsets as parts of α.

To simplify notation we say that $x \leq y$ in some state α if $x \in B_i(\alpha)$, $y \in B_j(\alpha)$ and $B_i(\alpha) \leq B_j(\alpha)$ for some $1 \leq i, j \leq t$.

For any state α we can construct a digraph $H(\alpha)$ where the vertices correspond to the elements $v_1, \ldots, v_p$ of $S - B_0(\alpha)$ such that there exists a directed edge from v_i to v_j if $v_i \leq v_j$ in α.

Example 3.3.2. *Let $S = \{a, b, c, d, e\}$ and let α be the state of S with parts $B_1(\alpha) = \{a, b\}, B_2(\alpha) = \{c\}, B_3(\alpha) = \{d\}$ and $B_0(\alpha) = \{e\}$ such that $B_2(\alpha) \leq B_1(\alpha) \leq B_3(\alpha)$. Then $H(\alpha)$ is given in Figure 3.6.*

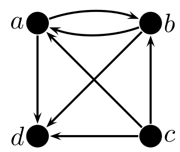

Figure 3.6

We say that two states α_1 and α_2 of a set S are *compatible* if $B_0(\alpha_1) = B_0(\alpha_2)$, and we denote their compatibility by $\alpha_1 \sim \alpha_2$.

As there is no known formula to compute the number of partially ordered sets with t elements, we now provide an upper bound for the number of states of a set.

Lemma 3.3.3. *The number of states of a set with t elements is at most 2^{t^2}.*

Proof. A state α is completely determined by the set $B_0(\alpha)$ and the digraph $H(\alpha)$. If α is a state of a set with t vertices then there are at most 2^t choices for $B_0(\alpha)$ and once $B_0(\alpha)$ has been chosen, the number of possibilities for $H(\alpha)$ is equal to the number of simple digraphs on $t - |B_0(\alpha)|$ vertices which is at most the number of digraphs on t vertices, namely $2^{t(t-1)}$. Hence the number of states is at most $2^t \cdot 2^{t(t-1)} = 2^{t^2}$. $\square$

These states play the same role in our algorithm as the states do in the previous section. Given a rooted digraph G, $S \subseteq V(G)$, $A \subseteq \overrightarrow{E}(G)$ and $B_0 \subseteq S$, a state $\alpha(S, A, B_0)$ of S is undefined if there is a directed path in $G|A$ from a vertex in $S - B_0$ to a vertex in B_0. Otherwise we let $B_0(\alpha) = B_0$ and $B_1(\alpha), \dots, B_t(\alpha)$ be the non-empty parts of $\alpha(S, A, B_0)$ such that two vertices $v, w \in S - B_0$ are in the same part of $\alpha(S, A, B_0)$ if and only if there are directed paths from v to w and from w to v in $G|A$. So the parts $B_1(\alpha), \dots, B_t(\alpha)$ are the non-empty intersections of the strong components of $G|A$ with $S - B_0$.

Now in the partial order for $i, j \neq 0$, we have $B_i(\alpha) \leq B_j(\alpha)$ if there is a directed path in $G|A$ from some vertex in $B_i(\alpha)$ to some vertex in $B_j(\alpha)$. It follows from the definition of the parts that this is equivalent to there being a directed path from every vertex in $B_i(\alpha)$ to every vertex in $B_j(\alpha)$.

Let $b(\alpha)$ denote the number of minimal parts of $\alpha(S, A, B_0)$, that is the number of parts $B_i(\alpha)$ in $S - B_0$ with $B_j(\alpha) \nleq B_i(\alpha)$ for all j with $j \neq i$.

If $\alpha = \alpha(S, A, B_0)$ then let $f(A, \alpha)$ be the number of vertices in $V(G) - S$ to which there is no directed path in $G|A$ from a vertex in $S - B_0$, and let $F(A, \alpha)$ be the set of such vertices. Now let $g(A, \alpha) = |A| - |V(G)| + f(A, \alpha) + |B_0| + b(\alpha)$.

Let $\Upsilon(S)$ be the set of all states of S. For each node i in a nice tree-decomposition of a rooted digraph G and $\alpha \in \Upsilon(S_i)$, we define $T(G_i, \alpha; x, y)$ by

$$T(G_i, \alpha; x, y) = \sum_{\substack{A \subseteq \overrightarrow{E}_i: \\ \alpha(S_i, A, B_0(\alpha)) = \alpha}} (x-1)^{f(A,\alpha)} (y-1)^{g(A,\alpha)}. \tag{3.8}$$

We will now describe the computation done at a leaf node in our algorithm. Note that for a leaf node i in T we have $V(G_i) = S_i$, hence $f(A, \alpha) = 0$ for all $A \subseteq \overrightarrow{E}_i$ and $\alpha \in \Upsilon(S_i)$. Let $\hat{E}_i$ contain one representative from each parallel class of edges contained in $\overrightarrow{E}_i$. Therefore for each leaf node i in T, when $y \neq 1$ we can express $T(G_i, \alpha; x, y)$ in the form

$$T(G_i,\alpha;x,y) = \sum_{\substack{A\subseteq \hat{E}_i: \\ \alpha(S_i,A,B_0(\alpha))=\alpha}} (y-1)^{|A|-|V(G_i)|+|B_0(\alpha)|+b(\alpha)} \prod_{e\in A}\left(\sum_{j=1}^{m(e)}\binom{m(e)}{j}(y-1)^{j-1}\right)$$
$$= \sum_{\substack{A\subseteq \hat{E}_i: \\ \alpha(S_i,A,B_0(\alpha))=\alpha}} (y-1)^{-|V(G_i)|+|B_0(\alpha)|+b(\alpha)} \prod_{e\in A}\left(y^{m(e)}-1\right).$$

For the special case when $y=1$ we have

$$T(G_i,\alpha;x,y) = \sum_{\substack{A\subseteq \hat{E}_i: \\ \alpha(S_i,A,B_0(\alpha))=\alpha, \\ |A|-|V(G_i)|+|B_0(\alpha)|+b(\alpha)=0}} \prod_{e\in A} m(e).$$

If $m(e)=1$ for all $e\in \overrightarrow{E}_i$ then we simply have

$$T(G_i,\alpha;x,y) = \sum_{\substack{A\subseteq \overrightarrow{E}_i: \\ \alpha(S_i,A,B_0(\alpha))=\alpha}} (y-1)^{|A|-|V(G_i)|+|B_0(\alpha)|+b(\alpha)}.$$

We now focus on the computation done at a join node in our algorithm.

Definition 3.3.4 (State Join)**.** The join of two compatible states α_1 and α_2 of a set S is given by the state $\alpha_1\vee\alpha_2$ of S where $B_0(\alpha_1\vee\alpha_2)=B_0(\alpha_1)=B_0(\alpha_2)$, and the sets $B_i(\alpha_1\vee\alpha_2)$ are given by the vertex sets of the strong components of $H(\alpha_1)\cup H(\alpha_2)$ for $1\le i\le t$. Now $B_i(\alpha_1\vee\alpha_2)\le B_j(\alpha_1\vee\alpha_2)$ if there exists a directed path from an element in $B_i(\alpha_1\vee\alpha_2)$ to an element in $B_j(\alpha_1\vee\alpha_2)$ in $H(\alpha_1)\cup H(\alpha_2)$.

Lemma 3.3.5. *Let i be a join node in a nice tree-decomposition with children nodes j and k. Let $A_j\subseteq \overrightarrow{E}_j$, $A_k\subseteq \overrightarrow{E}_k$ and $A_i=A_j\cup A_k$. Let B_0 be a subset of S_i. Suppose that the states $\alpha_j=\alpha(S_i,A_j,B_0)$ and $\alpha_k=\alpha(S_i,A_k,B_0)$ are defined. Then $\alpha=\alpha(S_i,A_i,B_0)$ is defined and*

1. *$\alpha=\alpha_j\vee\alpha_k$,*

2. *$f(A_i,\alpha)=f(A_j,\alpha_j)+f(A_k,\alpha_k)$, and*

3. *$g(A_i,\alpha)=g(A_j,\alpha_j)+g(A_k,\alpha_k)+b(\alpha)-b(\alpha_j)-b(\alpha_k)+|S_i|-|B_0|$.*

Proof. Suppose there is a directed path in $G_i|A_i$ from a vertex in S_i-B_0 to a vertex in B_0. In the shortest such path, the initial and final vertices are the only ones in S_i. As there are no edges between a vertex in $V(G_j) - S_i$ and a vertex in $V(G_k) - S_i$ either all the internal vertices of the path are in $V(G_j) - S_i$ or they are in $V(G_k) - S_i$. Consequently the edges are all from A_j or all from A_k. Thus there is either a path from a vertex in $S_i - B_0$ to a vertex in B_0 in $G_i|A_j$ or in $G_i|A_k$, giving a contradiction. Consequently we deduce that α is defined.

1. Construct directed graphs H_j and H_k both having vertex set $S_i - B_0$ and such that vw is a directed edge of H_j if there is a directed path from v to w in $G_i|A_j$, and similarly for H_k. The parts of α_j and α_k correspond to the vertex sets of the strongly connected components of H_j and H_k respectively.

   Now let $H_i = H_j \cup H_k$. There is a directed path from v to w in H_i if and only if there is a directed path from v to w in $G_i|(A_j \cup A_k)$. Thus the parts of α other than B_0 correspond to the strongly connected components of $G_i|(A_j \cup A_k)$ and are exactly the parts of $\alpha_j \vee \alpha_k$ other than B_0.

2. Showing that $F(A_j, \alpha_j) \cap F(A_k, \alpha_k) = \emptyset$ is identical to that in the proof of Lemma 3.2.10.

   We now prove $F(A_i, \alpha) \subseteq F(A_j, \alpha_j) \cup F(A_k, \alpha_k)$. Let $v \in F(A_i, \alpha)$, then $v \in V_j - S$ or $v \in V_k - S$. Suppose without loss of generality $v \in V_j - S$. We know that $G_j|A_j \subseteq G_i|A_i$ so if v is not reachable by a directed path from a vertex in $S - B_0$ in $G_i|A_i$ then it is not reachable by a directed path from a vertex in $S - B_0$ in $G_j|A_j$. Therefore $v \in F(A_j, \alpha_j)$.

   We now prove $F(A_i, \alpha) \supseteq F(A_j, \alpha_j) \cup F(A_k, \alpha_k)$. Let $v \in F(A_j, \alpha_j) \cup F(A_k, \alpha_k)$. Suppose without loss of generality $v \in F(A_j, \alpha_j)$ and that there is a directed path from a vertex $s \in S - B_0$ to v in $G_i|A_i$. Then there is a path $s \dots v_q v_p \dots v$ with $v_q \in V_k - S$ and $v_p \in V_j - S$. This means that $\{v_p, v_q\} \subseteq S_t$ for some $t \in I$. However $t \neq i$ because neither v_p nor v_q is a member of S. Furthermore t cannot be a descendant of j because (TD3) would imply that $v_q \in S$. Similarly t cannot be a descendant of k. Hence we have a contradiction. Therefore $v \in F(A_i, \alpha)$. Since $F(A_j, \alpha_j) \cap F(A_k, \alpha_k) = \emptyset$ and $F(A_i, \alpha) = F(A_j, \alpha_j) \cup F(A_k, \alpha_k)$ we have $f(A_i, \alpha) = f(A_j, \alpha_j) + f(A_k, \alpha_k)$.

3. We have $g(A_i, \alpha) = |A_i| - |V(G_i)| + f(A_i, \alpha) + |B_0| + b(\alpha)$ by definition. Now since $|V(G_i)| = |V(G_j)| + |V(G_k)| - |S_i|$ and $|A_i| = |A_j| + |A_k|$,

$$g(A_i, \alpha) = |A_j| + |A_k| - (|V(G_j)| + |V(G_k)| - |S_i|) + f(A_i, \alpha) + |B_0| + b(\alpha). \tag{3.9}$$

Using part 2, Equation 3.9 equals

$$|A_j| + |A_k| - |V(G_j)| - |V(G_k)| + |S_i| + f(A_j, \alpha_j) + f(A_k, \alpha_k) + |B_0| + b(\alpha).$$

Finally we can deduce the following required equation using the formulae for $g(A_j, \alpha_j)$ and $g(A_k, \alpha_k)$:

$$g(A_i, \alpha) = g(A_j, \alpha_j) + g(A_k, \alpha_k) + b(\alpha) - b(\alpha_j) - b(\alpha_k) + |S_i| - |B_0|.$$

□

Let i be a join node in T with children nodes j and k. For $\alpha, \alpha_j, \alpha_k \in \Upsilon(S_i)$, by Lemma 3.3.5 we can express $T(G_i, \alpha; x, y)$ in the form

$$\begin{aligned}
T(G_i, \alpha; x, y) &= \sum_{\substack{A_i \subseteq \overrightarrow{E}_i: \\ \alpha(S_i, A_i, B_0(\alpha)) = \alpha}} (x-1)^{f(A_i, \alpha)} (y-1)^{g(A_i, \alpha)} \\
&= \sum_{\substack{A_j \subseteq \overrightarrow{E}_j, A_k \subseteq \overrightarrow{E}_k: \\ \alpha(S_i, A_j \cup A_k, B_0(\alpha)) = \alpha}} (x-1)^{f(A_j \cup A_k, \alpha)} (y-1)^{g(A_j \cup A_k, \alpha)} \\
&= \sum_{\substack{\alpha_j, \alpha_k: \\ \alpha_j \sim \alpha_k, \\ \alpha_j \vee \alpha_k = \alpha}} \sum_{\substack{A_j \subseteq \overrightarrow{E}_j: \\ \alpha(S_j, A_j, B_0(\alpha_j)) = \alpha_j}} \sum_{\substack{A_k \subseteq \overrightarrow{E}_k: \\ \alpha(S_k, A_k, B_0(\alpha_k)) = \alpha_k}} (x-1)^{f(A_j, \alpha_j) + f(A_k, \alpha_k)} (y-1)^{g(A_j \cup A_k, \alpha)} \\
&= \sum_{\substack{\alpha_j, \alpha_k: \\ \alpha_j \sim \alpha_k, \\ \alpha_j \vee \alpha_k = \alpha}} T(G_j, \alpha_j; x, y) T(G_k, \alpha_k; x, y) (y-1)^{b(\alpha) - b(\alpha_j) - b(\alpha_k) + |S_i| - |B_0(\alpha)|}.
\end{aligned}$$

The following analysis will describe the computation done at a forget node in our algorithm. First we define a particular subset of states of a set.

Definition 3.3.6 (Forget States)**.** Let S be a set with $a \in S$ and $\alpha \in \Upsilon(S)$. We say that α is a *forget state* of a in S if:

- $a \in B_0(\alpha)$, or
- $a \notin B_0(\alpha)$ and a is not in a singleton part in α, or
- $a \notin B_0(\alpha)$ and a is in a singleton part $B_s(\alpha)$ in α with $B_t(\alpha) \leq B_s(\alpha)$ for some $t \neq s$.

We denote the set of forget states of a in S by $J(S,a)$ and we let $\alpha \setminus a$ denote the state α with a deleted.

Lemma 3.3.7. *Let i be a forget node in T and j be the child of i. Let a be the unique element of $S_j - S_i$. Then for $\beta \in \Upsilon(S_i)$ we have*

$$T(G_i, \beta; x, y) = \sum_{\substack{\alpha: \\ \alpha \setminus a = \beta, \\ \alpha \in J(S_j, a)}} T(G_j, \alpha; x, y)(x-1)^{h(\alpha)} \tag{3.10}$$

where $h(\alpha) = 1$ if $a \in B_0(\alpha)$ and $h(\alpha) = 0$ otherwise.

Proof. Suppose $A \subseteq \overrightarrow{E}_i$ is such that $\beta = \alpha(S_i, A, B_0(\beta))$ and $G_i|A$ has strongly connected components $C_1, \ldots, C_t$. Without loss of generality let $a \in V(C_a)$ where $1 \le a \le t$.

- Suppose $V(C_a) \cap (S_i - B_0(\beta)) = \emptyset$ and a is not reachable by a directed path from a vertex in $S_i - B_0(\beta)$ in $G_i|A$. Then $a \in F(A, \beta)$. Let α be the state of S_j with $B_s(\alpha) = B_s(\beta)$ for $s \ne 0$ and $B_0(\alpha) = B_0(\beta) \cup a$. Since $G_j = G_i$ the strongly connected components of $G_j|A$ are the same as those of $G_i|A$. Hence $\alpha = \alpha(S_j, A, B_0(\alpha))$. Clearly $\alpha \in J(S_j, a)$ and $\alpha \setminus a = \beta$.

  Now $F(A,\beta) = F(A,\alpha) \cup a$ so we have $f(A,\beta) = f(A,\alpha) + 1$. Also $b(\beta) = b(\alpha)$ and $|B_0(\beta)| = |B_0(\alpha)| - 1$ so $g(A,\beta) = g(A,\alpha)$.

- Suppose $V(C_a) \cap (S_i - B_0(\beta)) = \emptyset$ and a is reachable by a directed path from some vertex b in $S_i - B_0(\beta)$ in $G_i|A$. The existence of the path from b to a ensures that $a \notin B_0(\beta)$. As $V(C_a) \cap (S_i - B_0(\beta)) = \emptyset$ the vertex a must appear as a singleton part in α. Moreover the existence of the path from b to a ensures that this part is not minimal. Since $G_j = G_i$ the strongly connected components of $G_j|A$ are the same as those of $G_i|A$. Hence $\alpha = \alpha(S_j, A, B_0(\alpha))$. Clearly $\alpha \in J(S_j, a)$ and $\alpha \setminus a = \beta$.

  Now $F(A,\beta) = F(A,\alpha)$ so we have $f(A,\beta) = f(A,\alpha)$. Also $b(\beta) = b(\alpha)$ and $|B_0(\beta)| = |B_0(\alpha)|$ so $g(A,\beta) = g(A,\alpha)$.

- Suppose $V(C_a) \cap (S_i - B_0(\beta)) \ne \emptyset$. Then a belongs to the same strong component of $G_i|A$ as some other vertex b in $S_i - B_0(\beta)$. Let $B_b(\beta)$ be the part containing b in β. Let α be the state of S_j with $B_s(\alpha) = B_s(\alpha)$ for $s \ne b$ and $B_b(\alpha) = B_b(\beta) \cup a$. Since $G_j = G_i$ the strongly connected components of $G_j|A$ are the same as those of $G_i|A$. Hence $\alpha = \alpha(S_j, A, B_0(\alpha))$. Clearly $\alpha \in J(S_j, a)$ and $\alpha \setminus a = \beta$.

Now $F(A,\beta) = F(A,\alpha)$ so we have $f(A,\beta) = f(A,\alpha)$. Also $b(\beta) = b(\alpha)$ and $|B_0(\beta)| = |B_0(\alpha)|$ so $g(A,\beta) = g(A,\alpha)$.

We have shown that for $A \subseteq \overrightarrow{E}_i$ with $\beta = \alpha(S_i, A, B_0(\beta))$, there exists $\alpha = \alpha(S_j, A, B_0(\alpha))$ such that $\alpha \in J(S_j, a)$ and $\alpha \setminus a = \beta$.

Now suppose instead we take $A \subseteq \overrightarrow{E}_j$ such that $\alpha = \alpha(S_j, A, B_0(\alpha))$ with $\alpha \in J(S_j, a)$ and $\alpha \setminus a = \beta$. Since $G_i = G_j$ the strongly connected components of $G_i|A$ are the same as the strongly connected components of $G_j|A$. Hence $\beta = \alpha(S_i, A, B_0(\beta))$. Therefore we have shown that each subset A of $\overrightarrow{E}_i$ makes the same contribution to both sides of Equation 3.10. $\square$

We now describe the computation done at an introduce node in our algorithm.

Definition 3.3.8 (Introduce States)**.** Let S be a set with $a \in S$ and $\alpha \in \Upsilon(S)$. We say that α is an *introduce state* of a in S if:

- $a \in B_0(\alpha)$, or
- $a \notin B_0(\alpha)$ and a is in a singleton part B such that for each other part B' neither $B' \leq B$ nor $B \leq B'$.

We denote the set of introduce states of a in S by $I(S,a)$. Note that if $a \notin B_0(\alpha)$, then it follows immediately from the definition of an introduce state that a forms a minimal part of α.

Lemma 3.3.9. *Let i be an introduce node in T and j be the child node of i. Let $a \in S_i - S_j$. We can express $T(G_i, \alpha; x, y)$ in the form*

$$T(G_i, \alpha; x, y) = \begin{cases} T(G_j, \alpha \setminus a; x, y) & \textit{if } \alpha \in I(S_i, a) \\ 0 & \textit{otherwise.} \end{cases}$$

Proof. By (TD3) a cannot be in any bag corresponding to a descendant node of i in T. Therefore any edges incident to a in G cannot be in $\overrightarrow{E}_i$. This means that a must be an isolated vertex in $G_i|A$ for any $A \subseteq \overrightarrow{E}_i$. So if $\alpha \notin I(S_i, a)$ then $T(G_i, \alpha; x, y) = 0$.

If $\alpha \in I(S_i, a)$ and $A \subseteq \overrightarrow{E}_i$ then $\alpha = \alpha(S_i, A, B_0(\alpha))$ if and only if $\alpha \setminus a = \alpha(S_j, A, B_0(\alpha) - a)$.

Moreover $F(A,\alpha) = F(A, \alpha \setminus a)$ and

$$\begin{aligned} g(A,\alpha) &= |A| - |V(G_i)| + f(A,\alpha) + |B_0(\alpha)| + b(\alpha) \\ &= |A| - (|V(G_j)| + 1) + f(A, \alpha \setminus a) + |B_0(\alpha) - a| + b(\alpha \setminus a) + 1 \\ &= g(A, \alpha \setminus a). \end{aligned}$$

□

We now describe the computation done in our algorithm after we have computed $T(G_\rho, \alpha; x, y)$ for the root node ρ.

Lemma 3.3.10. *If $G = (r, V, \overrightarrow{E})$ is root-connected then the Tutte polynomial of G is given by*

$$T(G; x, y) = \sum_{\substack{\alpha: b(\alpha)=1 \\ r \text{ is in the unique} \\ \text{minimal part}}} T(G_\rho, \alpha; x, y)(x-1)^{|B_0(\alpha)|}. \tag{3.11}$$

Proof. We show that each subset A of E contributes to exactly one term on the right-hand side of Equation 3.11. Let $C_1, \ldots, C_t$ be the strongly connected components of $G|A$ for some $A \subseteq E$. Let J be the subset of $[t]$ defined so that $j \in J$ if and only if there is no directed path from r to any vertex in C_j. Let $B_0 = \bigcup_{j \in J}(V(C_j) \cap S_\rho)$ and let $\alpha = \alpha(S_\rho, A, B_0)$. Then α is defined and since there is a directed path in $G|A$ from r to each vertex that is not in B_0, the only minimal part of α other than B_0 is the one containing r. If α is any other state of S_ρ satisfying $\alpha = \alpha(S_\rho, A, B_0(\alpha))$, then α has at least two minimal parts besides $B_0(\alpha)$.

Now $|V(G_\rho)| = \cup_{j \notin J}|V(C_j)| + f(A,\alpha) + |B_0(\alpha)|$. Therefore

$$\begin{aligned} \rho(G) - \rho(A) &= (|V(G)| - 1) - \big(\cup_{j \notin J}|V(C_j)| - 1\big) \\ &= |V(G_\rho)| - (|V(G_\rho)| - f(A,\alpha) - |B_0(\alpha)|) \\ &= f(A,\alpha) + |B_0(\alpha)|, \end{aligned}$$

and

$$
\begin{aligned}
|A| - \rho(A) &= |A| - \big(\cup_{j \notin J} |V(C_j)| - 1\big) \\
&= |A| - (|V(G_\rho)| - f(A, \alpha) - |B_0(\alpha)| - 1) \\
&= |A| - |V(G_\rho)| + f(A, \alpha) + |B_0(\alpha)| + b(\alpha) \\
&= g(A_\rho, \alpha).
\end{aligned}
$$

Hence we have shown that each subset A of $\overrightarrow{E}$ makes the same contribution to both sides of Equation 3.11 □

Combining Lemmas 3.3.7, 3.3.9 and 3.3.10 and the discussion on the computations done at a leaf and join node, we see that $T(G; x, y)$ can be computed using an algorithm analogous to Algorithm 1, when G is root connected. If G is not root connected then every edge that is not contained in a directed path from the root to some vertex in G is a loop in the greedoid $\Gamma(G)$. Thus if G is a rooted digraph, $G' = G|A$ where A is the set of edges with both endpoints being reachable by a directed path from the root in G, and L is the number of edges in G that are not in G', then

$$T(G; x, y) = y^L T(G'; x, y).$$

3.3.1 Complexity of the Algorithm

For a rooted digraph $G = (r, V, \overrightarrow{E})$ we let $t(n, m, k, x, y, z)$ be the maximum number of operations required to evaluate $T(G; x, y)$ if $n = |V|$, $m = |\overrightarrow{E}|$, $z = \max_{e \in \overrightarrow{E}}\{m(e)\}$ and G has tree-width at most k. We let $\delta = \delta(n, m, k, x, y, z)$ denote the maximum time taken for one arithmetical operation during the algorithm. Let

$$h(k) = k^5 (2k+1)^{(2k-1)} ((4k+5)^{(4k+5)} (2^{(2k+5)}/3)^{(4k+5)})^{(4k+1)}.$$

The complexity for each preprocessing step is the same as that for the algorithm described in the previous section. Therefore the maximum running time for the preprocessing steps is $O(h(k)n + m + z\delta)$.

In order to check whether two states α_1 and α_2 are compatible, we need to check that $B_0(\alpha_1) = B_0(\alpha_2)$, each of which has at most $k+1$ elements. The time required to do this is $O(k^2)$. Let α_1

and α_2 be two compatible states. To determine whether $\alpha_1 = \alpha_2$ we simply check that they have the same parts which can be done in $O(k^2)$ time since there are at most $k+1$ parts in each state and check that they have the same partial order which can also be done in $O(k^2)$ time. To compute $\alpha_1 \vee \alpha_2$ for compatible states α_1 and α_2 we can construct the graph $H(\alpha_1) \cup H(\alpha_2)$ and then perform Tarjan's algorithm [52] to find the strongly connected components of the graph. This would give us the parts of $\alpha_1 \vee \alpha_2$. The edges between the strong components give us the partial order of $\alpha_1 \vee \alpha_2$. Again these operations can be done in $O(k^2)$ time. Therefore joining two states in our algorithm takes an overall time of $O(k^2)$.

Clearly the time taken to run the main part of the algorithm, that is omitting the preprocessing steps, is $O(nt'(n, m, k, x, y, z))$ where t' is the maximum time required to compute $T(G_i, \alpha; x, y)$ for a join node i and some state $\alpha \in \Upsilon(S_i)$. Recall that the number of states of a set with t elements is bounded above by 2^{t^2}. For a join node we combine at most $2^{2(k+1)^2}$ states, with the contribution from each pair taking at most $O(k^2 + \delta)$ time to compute by the preceding discussion. Computing the running time for the main part of our algorithm in terms of δ therefore takes time

$$O(n(k^2 + \delta)2^{2(k+1)^2}).$$

As in the previous section we can show that $\delta \leq l \log l \log\log l$ where $l = (2n + 3m)\log(|p_x| + |q_x| + |p_y| + |q_y| + 2)$. The running time for the main part of the algorithm is therefore

$$O(n(k^2 + (n + m)\log(|p_x| + |q_x| + |p_y| + |q_y| + 2)\log l \log\log l)2^{2(k+1)^2}).$$

Suppose that our input graph has no parallel edges. Then the running time for the algorithm becomes

$$O(n(k^2 + nk\log(|p_x| + |q_x| + |p_y| + |q_y| + 2)\log l \log\log l)2^{2(k+1)^2}).$$

3.4 Binary Greedoids

In this section we construct a polynomial-time algorithm to evaluate the Tutte polynomial of particular binary greedoids of bounded tree-width. We will use an important result by Hliněný that provides a polynomial-time algorithm to compute the Tutte polynomial of a representable matroid over a finite field of bounded branch-width. The notion of branch-width of a graph was first introduced by Robertson and Seymour as their main tool for proving Wagner's Conjecture in their pioneering work on graph minors [47]. Like tree-width, it has since then proven very useful in computational complexity theory as many intractable graph optimization problems may be solved efficiently for graphs of bounded branch-width. Although the graph parameter tree-width has undoubtedly proven to be more popular in the fields of graph theory and theoretical computer science, see [6] for example, it is branch-width that has drawn the attention of many matroid theorists. This is because, unlike tree-width, branch-width directly generalizes to matroids as its definition makes no explicit reference to graph vertices.

We begin by defining branch-width of a graph and more generally a matroid, and then present several results regarding this parameter. A graph is said to be a *binary tree* if it is a tree and every vertex has either degree 1 or 3.

Definition 3.4.1 (Branch-Decomposition of a Graph)**.** A *branch-decomposition* of a graph G is a pair (T, τ) where T is a binary tree and τ is a bijection from the set of edges of G to the vertices of degree 1 of T.

Let (T, τ) be a branch-decomposition of a graph G and let $e \in E(T)$. Let T_1 and T_2 be the connected components of $T \setminus e$. Let L_1 and L_2 denote the set of vertices of degree 1 of T_1 and T_2 respectively. This deletion induces a partition of $E(G)$ into sets $E_1 = \tau^{-1}(L_1)$ and $E_2 = \tau^{-1}(L_2)$. The *size* of e is the number of vertices that are an endpoint of both an edge in E_1 and an edge in E_2.

Definition 3.4.2 (Branch-Width of a Graph)**.** The *width* of a branch-decomposition (T, τ) of a graph G is the maximum size of any edge $e \in E(T)$. The *branch-width* of G, denoted by $bw(G)$, is the minimum width taken over all possible branch-decompositions of G.

The notion of a branch-decomposition can naturally be extended to matroids in the following way. A *branch-decomposition* of a matroid M is a pair (T, τ) where T is a binary tree and τ is a bijection from the elements of the ground set $E(M)$ to the vertices of degree 1 of T. Let (T, τ) be a branch-decomposition of a matroid M and let $e \in E(T)$. Let T_1 and T_2 be the connected components of $T \setminus e$. Let L_1 and L_2 denote the set of vertices of degree 1 of T_1 and T_2 respectively, and let $E_1 = \tau^{-1}(L_1)$ and $E_2 = \tau^{-1}(L_2)$. The size of e is given by $r(E_1) + r(E_2) - r(M) + 1$, and the *branch-width* of a matroid M, denoted by $bw(M)$, is defined analogously.

In [50] Seymour and Thomas show that computing the branch-width of a general graph is NP-hard, and computing that of a planar graph can be done in polynomial time. The latter result is somewhat surprising as the analogous computational problem for tree-width is still open. Kloks, Kratochvil and Muller consider the computational complexity of determining the branch-width of several classes of graphs in [33]. In particular they show that computing the branch-width of a bipartite graph is NP-complete.

Bodlaender and Thilikos [8] prove that, for any fixed $k \in \mathbb{N}$, one can construct a linear-time algorithm that checks whether a graph has branch-width $\leq k$ and, if so, outputs a branch-decomposition of minimum width. In [28] Hliněný shows that for each positive integer k and finite field $\mathbb{F}$, there is an algorithm which inputs a matrix A with entries from $\mathbb{F}$ such that $M(A)$ has branch-width at most k and outputs a branch-decomposition of $M(A)$ with width at most $3k$.

It is natural to ask whether the notion of tree-width can also be generalized to matroids. It is not immediately obvious that this can be done since the definition of the tree-width of a graph makes considerable use of the vertices. However, in [30] Hliněný and Whittle define tree-width of a graph without reference to the vertices of the graph in a way which we shall now discuss.

A *tree-decomposition* of a matroid M is a pair (T, τ) where T is a tree and $\tau : E(M) \to V(T)$ is an arbitrary mapping. For a node (vertex) x of T, denote the connected components of $T \setminus x$ by $T_1, T_2, \ldots, T_d$ and set $F_i = \tau^{-1}(V(T_i)) \subseteq E(M)$. The *node-width* of x is given by

$$r(M) - \sum_{i=1}^{d} \left[r(M) - r(M \setminus F_i) \right],$$

and the *width* of the tree-decomposition (T, τ) is the maximum width of any node of T. The *tree-width* of M is then said to be the minimum width taken over all possible tree-decompositions of M.

The main theorem from [30] is stated below.

Theorem 3.4.3. *Let G be a graph with at least one edge, and let $M = M(G)$ be the cycle matroid of G. Then the tree-width of G equals the tree-width of M.*

The following result was proven by Hicks and McMurray Jr [27] and independently by Mazoit and Thomassé [41].

Theorem 3.4.4. *Let G be a graph with a cycle of length at least 2 and $M = M(G)$ be the cycle matroid of G. Then the branch-width of G equals the branch-width of M.*

We now discuss the relationship between tree-width and branch-width of a graph and discover that the two parameters differ by a small linear factor. In [47] Robertson and Seymour give the following theorem.

Theorem 3.4.5. *Let G be a graph with tree-width k and branch-width $b > 1$. Then*

$$b - 1 \leq k \leq \left\lfloor \frac{3b}{2} \right\rfloor - 1.$$

This implies that a class of graphs has bounded tree-width if and only if it has bounded branch-width. Hliněný and Whittle extend this result to all matroids in [30].

Theorem 3.4.6. *Let M be a matroid of tree-width k and branch-width $b > 1$. Then*

$$b - 1 \leq k \leq max\ (2b - 2, 1).$$

Similarly this implies that a class of matroids has bounded tree-width if and only if it has bounded branch-width. Let M be a representable matroid over a finite field. In [29] Hliněný presents a recursive formula to compute the Tutte polynomial of M using a so-called parse tree of a tree-decomposition of M. This formula provides a polynomial-time algorithm with a fixed exponent to compute the Tutte polynomial of M when M is of bounded branch-width. An important result from [29] is given below.

Corollary 3.4.7. *Let $\mathbb{F}$ be a finite field, and let b be an integer constant. Suppose that $x, y \in \mathbb{Q}$ can be written as $x = \frac{p}{q}$ and $y = \frac{r}{s}$ such that the combined length of p, q, r, s is l bits, and that M is an n-element $\mathbb{F}$-represented matroid of branch-width at most b. Then the Tutte polynomial $T(M; x, y)$ can be evaluated in time $O(n^3 + n^2 l \log(nl) \log\log(nl))$.*

Note that this almost matches the performance of the algorithm to evaluate the Tutte polynomial of a graph of bounded tree-width given in [42]. By Theorem 3.4.6 this provides a polynomial-time algorithm to evaluate the Tutte polynomial of a representable matroid of bounded tree-width.

Let N be an $m \times n$ matrix. We let N_i denote the submatrix of N with columns $1, 2, \ldots, n$ and rows $1, 2, \ldots, i$ for some $1 \leq i \leq m$.

Theorem 3.4.8. *Let N be an $m \times n$ binary matrix with linearly independent rows. Let $\Gamma = \Gamma(N)$ and $M = M(N)$ be the binary greedoid and vector matroid of N respectively. Let $M_i = M(N_i)$ and let M_0 be the matroid comprising n loops. Then*

$$T(\Gamma; x, y) = T(M; 1, y) + \sum_{i=1}^{m} \Big[T(M_{m-i}; 1, y) - (y-1)T(M_{m-i+1}; 1, y) \Big] (x-1)^i. \tag{3.12}$$

Proof. By Lemma 1.3.17 the bases of Γ coincide with the bases of M. Moreover $\rho(\Gamma) = r(M) = m$. Let E denote the set of columns of N. We prove this theorem by showing that if each Tutte polynomial in Equation 3.12 is expressed as a sum over the subsets of its edges then the contribution of a subset A of E is the same on both sides of Equation 3.12. We let $[A]T(M; x, y)$ denote the contribution of A to the term $T(M; x, y)$. Now

$$\begin{aligned} T(\Gamma; x, y) &= \sum_{A \subseteq E} (x-1)^{\rho(\Gamma)-\rho(A)} (y-1)^{|A|-\rho(A)} \\ &= \sum_{\substack{A \subseteq E: \\ \rho(A)=\rho(\Gamma)}} (y-1)^{|A|-\rho(A)} + \sum_{\substack{A \subseteq E: \\ \rho(A)\neq\rho(\Gamma)}} (x-1)^{\rho(\Gamma)-\rho(A)} (y-1)^{|A|-\rho(A)}. \end{aligned}$$

The contribution of A to the LHS of Equation 3.12 is therefore

$$\begin{array}{ll} (y-1)^{|A|-\rho(A)} & \text{if } \rho(A) = \rho(\Gamma), \text{ and} \\ (x-1)^{\rho(\Gamma)-\rho(A)} (y-1)^{|A|-\rho(A)} & \text{if } \rho(A) \neq \rho(\Gamma). \end{array}$$

Now suppose $\rho(A) = \rho(\Gamma)$, then

$$[A]T(M; 1, y) = (y-1)^{|A|-\rho(\Gamma)}$$

and

$$[A]T(M_{m-i}; 1, y) = (y-1)^{|A|-\rho(\Gamma)+i}.$$

Therefore the contribution of A when $\rho(A) = \rho(\Gamma)$ to the RHS of Equation 3.12 is

$$\begin{aligned} &(y-1)^{|A|-\rho(\Gamma)} + \sum_{i=1}^{r(M)} \Big[(y-1)^{|A|-\rho(\Gamma)+i} - (y-1)(y-1)^{|A|-\rho(\Gamma)+i-1} \Big] (x-1)^i \\ &= (y-1)^{|A|-\rho(\Gamma)}. \end{aligned}$$

Now suppose $\rho(A) < \rho(\Gamma)$, then

$$[A]T(M_j;1,y) = \begin{cases} 0 & \text{if } j > \rho(A), \text{ and} \\ (y-1)^{|A|-j} & \text{if } j \leq \rho(A). \end{cases}$$

Therefore the contribution of A when $\rho(A) < \rho(\Gamma)$ to the RHS of Equation 3.12 is

$$\sum_{i=\rho(\Gamma)-\rho(A)}^{r(M)} (y-1)^{|A|-\rho(\Gamma)+i}(x-1)^i - (y-1)\sum_{i=\rho(\Gamma)-\rho(A)+1}^{r(M)} (y-1)^{|A|-\rho(\Gamma)+i-1}(x-1)^i$$
$$= (x-1)^{\rho(\Gamma)-\rho(A)}(y-1)^{|A|-\rho(A)}.$$

Hence A has the same contribution to both sides of Equation 3.12. $\square$

We now present the main theorem of this section.

Theorem 3.4.9. *For every $k \in \mathbb{N}$ there exists an algorithm $\mathcal{A}_k$ that will input an $m \times n$ binary matrix N with linearly independent rows such that for each i with $1 \leq i \leq m$ the vector matroid $M(N_i)$ has tree-width at most k, and rationals $x = \frac{p_x}{q_x}$, $y = \frac{p_y}{q_y}$, and evaluate $T(\Gamma(N);x,y)$ in time at most*

$$O(m(n^3 + n^2 l \log(nl) \log\log(nl)))$$

where $l = \log(|p_x| + |q_x| + |p_y| + |q_y| + 1)$ and $\Gamma(N)$ is the binary greedoid of N.

Proof. Let $M_i = M(N_i)$. By Corollary 3.4.7 and the discussion preceding Theorem 3.4.8 there exists a polynomial-time algorithm to evaluate $T(M_i;x,y)$ for all i with $1 \leq i \leq m$ such that M_i has bounded tree-width k. The result now follows by Equation 3.12. Moreover we can evaluate $T(\Gamma(N);x,y)$ by calling Hliněný's algorithm to evaluate $T(M_i;x,y)$ for each i, thus taking time at most $O(m(n^3 + n^2 l \log(nl) \log\log(nl)))$. $\square$

Note that we need the condition that every vector matroid M_i has tree-width at most k for all $1 \leq i \leq m$ in Theorem 3.4.8 otherwise we may have $tw(M_{i-1}) > tw(M_i)$. For example consider the following binary matrix representation of the path with three edges

$$N = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Clearly the path has tree-width 1. By Theorem 3.4.3 this implies $M(N)$ has tree-width 1. Now if we delete the bottom row of N we obtain a binary matrix representation of the triangle. The triangle has tree-width 2 and so by Theorem 3.4.3 $M(N_2)$ has tree-width 2.

Chapter 4

The Characteristic Polynomial and the Computational Complexity of the Coefficients of the Tutte Polynomial of a Rooted Graph

4.1 The Characteristic Polynomial

In this section we present a new expression for the characteristic polynomial of a rooted graph in terms of the Möbius function.

The total number of ways of vertex colouring a graph $G = (V, E)$ using a palette of λ colours is $\lambda^{|V|}$. For $A \subseteq E$ the number of ways we may assign colours to the vertices of $G|A$ so that adjacent vertices share the same colour is $\lambda^{\kappa(A)}$. Using the principle of inclusion/exclusion, Whitney [60] gives the following definition for the chromatic polynomial

$$P(G; \lambda) = \sum_{A \subseteq E} (-1)^{|A|} \lambda^{\kappa(A)}.$$

The *characteristic polynomial* of a matroid $M = (E, \mathcal{I})$ is defined by

$$p(M;\lambda) = \sum_{A \subseteq E} (-1)^{|A|} \lambda^{r(M)-r(A)}.$$

Let $M = M(G)$ be the cycle matroid of a graph $G = (V, E)$. Since $r(M) = r(G)$ we have

$$\begin{aligned} p(M;\lambda) &= \sum_{A \subseteq E} (-1)^{|A|} \lambda^{r(G)-r(A)} = \sum_{A \subseteq E} (-1)^{|A|} \lambda^{\kappa(A)-\kappa(G)} \\ &= \lambda^{-\kappa(G)} \sum_{A \subseteq E} (-1)^{|A|} \lambda^{\kappa(A)} = \frac{1}{\lambda^{\kappa(G)}} P(G;\lambda). \end{aligned}$$

We now show that the characteristic polynomial of a matroid $M = (E, \mathcal{I})$ is a specialization of the Tutte polynomial of M:

$$\begin{aligned} p(M;\lambda) &= \sum_{A \subseteq E} (-1)^{|A|} \lambda^{r(M)-r(A)} = \sum_{A \subseteq E} (-1)^{|A|-r(A)+r(A)-r(M)+r(M)} \lambda^{r(M)-r(A)} \\ &= (-1)^{r(M)} \sum_{A \subseteq E} (-\lambda)^{r(M)-r(A)} (-1)^{|A|-r(A)} = (-1)^{r(M)} T(M; 1-\lambda, 0). \end{aligned} \tag{4.1}$$

The characteristic polynomial therefore follows a deletion/contraction recursion. Let $e \in E$ such that e is neither a loop nor a coloop. We have $r(M \setminus e) = r(M)$ and $r(M/e) = r(M) - 1$. Therefore

$$\begin{aligned} p(M;\lambda) &= (-1)^{r(M)} \left[T(M \setminus e; 1-\lambda, 0) + T(M/e; 1-\lambda, 0) \right] \\ &= (-1)^{r(M \setminus e)} T(M \setminus e; 1-\lambda, 0) + (-1)^{r(M/e)+1} T(M/e; 1-\lambda, 0) \\ &= p(M \setminus e; \lambda) - p(M/e; \lambda). \end{aligned}$$

If $e \in E$ is a coloop then $M/e = M \setminus e$. It is easy to check that $T(M; x, y) = xT(M \setminus e; x, y)$. We have $r(M \setminus e) = r(M) - 1$. Therefore

$$p(M;\lambda) = (-1)^{r(M \setminus e)+1} (1-\lambda) T(M \setminus e; 1-\lambda, 0) = (\lambda - 1) p(M \setminus e; \lambda).$$

Similarly $p(M;\lambda) = (\lambda - 1) p(M/e; \lambda)$.

If there exists $e \in E$ that is a loop then $M/e = M \setminus e$. It is easy to check that $T(M; x, y) = yT(M \setminus e; x, y)$. Therefore

$$p(M;\lambda) = 0.$$

Suppose we have a partially ordered set P and let $x, y \in P$. The Möbius function μ_P of P is defined by

$$\mu_P(x,y) = \begin{cases} 1 & \text{if } x = y, \\ -\sum_{x \le z < y} \mu_P(x,z) & \text{if } x < y, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Let $M = (E, \mathcal{I})$ be a matroid and $A \subseteq E$. The set A is a *flat* of M if for any $x \in E - A$ we have $r(A \cup x) = r(A) + 1$. Let S be a finite set and $\le$ be a partial ordering on S. Given elements x and y of S, the element z of S is an *upper bound* for $\{x, y\}$ if $x \le z$ and $y \le z$. It is a *least upper bound* if $z \le z'$ for every upper bound z'. The notion of a *greatest lower bound* is defined similarly. A partial order is a *lattice* if every pair of elements has both a least upper bound and a greatest lower bound. In a lattice we say that an element X *covers* an element Y if $Y \le X$ and there is no element Z such that $Y \le Z \le X$.

We define the lattice $L(M)$ of a matroid M to be the lattice where the elements correspond to the flats of M, and $X \le Y$ if $X \subseteq Y$. The Möbius function μ_M of M is given by $\mu_M(X, Y) = \mu_{L(M)}(X, Y)$. Note that the Möbius function of a matroid can naturally be specialized to that of a graph. See [64] for a detailed discussion on the Möbius function of a matroid. Suppose F is an element of $L(M)$, then we define

$$\mu_M(F) = \begin{cases} \mu_M(\emptyset, F) & \text{if } M \text{ is loopless, and} \\ 0 & \text{otherwise.} \end{cases}$$

The following two propositions are originally due to Brylawski [9] and Rota [48] respectively.

Proposition 4.1.1. *Let $M = (E, \mathcal{I})$ be a matroid and $e \in E$. Then*

- $$\mu_M(E) = \begin{cases} \mu_{M\backslash e}(E-e) - \mu_{M/e}(E-e) & \textit{if } e \textit{ is not a coloop, and} \\ -\mu_{M\backslash e}(E-e) = -\mu_{M/e}(E-e) & \textit{otherwise.} \end{cases}$$

- *If $M = M_1 \oplus M_2$ where $M_1 = (E_1, \mathcal{I}_1)$ and $M_2 = (E_2, \mathcal{I}_2)$ are matroids, then $\mu_M(E) = \mu_{M_1}(E_1)\mu_{M_2}(E_2)$.*

Proposition 4.1.2. *The Möbius function of a matroid M is non-zero and alternates in sign. More*

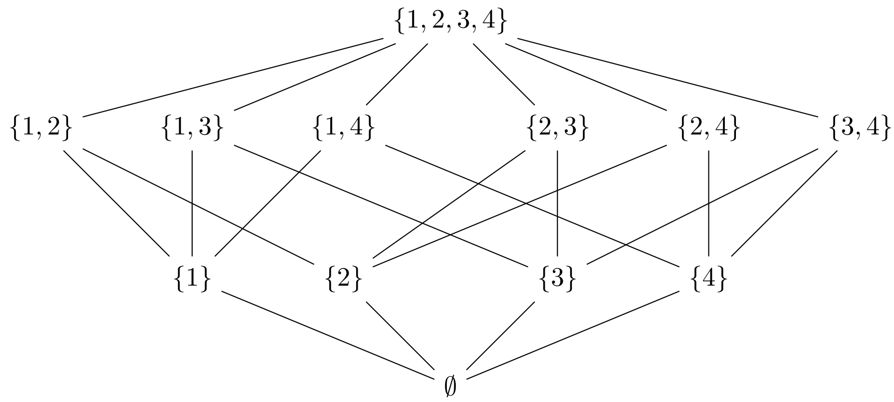

Figure 4.1: Lattice $L(M)$.

precisely for $X, Y \in L(M)$ with $X \leq Y$ we have

$$(-1)^{r(Y)-r(X)}\mu_M(X,Y) > 0.$$

Zaslavsky [64] gives the following expression for the characteristic polynomial of a matroid. Let M be a matroid, then

$$p(M;\lambda) = \sum_{F \in L(M)} \mu_M(F)\lambda^{r(M)-r(F)}. \tag{4.2}$$

Example 4.1.3. *Let $M = (E, \mathcal{I})$ be the matroid with $E = \{1,2,3,4\}$ and*

$$\mathcal{I} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\}, \{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}\}.$$

The lattice $L(M)$ is given in Figure 4.1. The flats of M together with their corresponding Möbius function are presented in the table below.

$F \in L(M)$	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$	$\{3,4\}$	$\{1,2,3,4\}$
$\mu_M(F)$	1	-1	-1	-1	-1	1	1	1	1	1	1	-3

Using Equation 4.2 we compute

$$p(M;\lambda) = \lambda^3 - 4\lambda^2 + 6\lambda - 3.$$

We now relate the Möbius function of a graphic matroid to the orientations of the corresponding graph. Let $\mathcal{A}(G)$ denote the collection of all acyclic orientations of a graph G with one predefined source vertex.

Proposition 4.1.4. *Let $M = M(G)$ be the cycle matroid of the connected graph $G = (V, E)$. Then*

$$\mu_M(E) = (-1)^{r(M)}|\mathcal{A}(G)|.$$

Proof. By Equations 4.2 and 4.1 we have

$$T(M; 1, 0) = (-1)^{r(M)} p(M; 0) = (-1)^{r(M)} \sum_{\substack{F \in L(M): \\ r(F)=r(M)}} \mu_M(F) = (-1)^{r(M)} \mu_M(E). \tag{4.3}$$

The last equality follows from E being the unique flat of M with full rank. The result now follows immediately by recalling from Section 1.5 that, since G is connected, $T(M; 1, 0)$ gives the number of acyclic orientations of G with one predefined source vertex. □

Gordon and McMahon generalize the characteristic polynomial to greedoids in [22]. Let Γ be a greedoid. Then the *characteristic polynomial* of Γ is given by

$$p(\Gamma; \lambda) = (-1)^{\rho(\Gamma)} T(\Gamma; 1 - \lambda, 0). \tag{4.4}$$

They also show that several of the matroidal results have direct greedoid analogues.

Using Equation 4.4 and Proposition 1.7.1, we now prove that the characteristic polynomial of a greedoid satisfies the following deletion/contraction recursion. This result is originally given in [22]. Let $\Gamma = (E, \mathcal{F})$ be a greedoid and let $e \in E$ such that $\{e\} \in \mathcal{F}$. We have $\rho(\Gamma/e) = \rho(\Gamma) - 1$. Therefore

$$\begin{aligned} p(\Gamma; \lambda) &= (-1)^{\rho(\Gamma)} T(\Gamma; 1 - \lambda, 0) \\ &= (-1)^{\rho(\Gamma)} \left[T(\Gamma/e; 1 - \lambda, 0) + (-\lambda)^{\rho(\Gamma) - \rho(\Gamma \setminus e)} T(\Gamma \setminus e; 1 - \lambda, 0) \right] \\ &= (-1)^{\rho(\Gamma)} \left[(-1)^{\rho(\Gamma/e)} p(\Gamma/e; \lambda) + (-\lambda)^{\rho(\Gamma) - \rho(\Gamma \setminus e)} (-1)^{\rho(\Gamma \setminus e)} p(\Gamma \setminus e; \lambda) \right] \\ &= \lambda^{\rho(\Gamma) - \rho(\Gamma \setminus e)} p(\Gamma \setminus e; \lambda) - p(\Gamma/e; \lambda). \end{aligned} \tag{4.5}$$

Another result from [22] is that the direct sum property holds for the characteristic polynomial of a greedoid. That is for greedoids Γ, Γ_1 and Γ_2 such that $\Gamma = \Gamma_1 \oplus \Gamma_2$, we have

$$p(\Gamma; \lambda) = p(\Gamma_1; \lambda)p(\Gamma_2; \lambda).$$

This follows immediately from Equation 4.4 and the direct sum property of the Tutte polynomial of a greedoid.

Also given in [22] is the result that the coefficients of the characteristic polynomial still alternate in sign when defined for a greedoid. That is, by expressing $p(\Gamma; \lambda)$ in the form

$$p(\Gamma; \lambda) = \sum_{k=0}^{\rho(\Gamma)} w_k \lambda^{\rho(\Gamma)-k},$$

the sign of w_k is $(-1)^k$.

Gordon and McMahon [22] find an analogous result to Equation 4.2 for antimatroids, a particular well-behaved class of greedoids.

Definition 4.1.5 (Antimatroid)**.** An *antimatroid* $\Gamma = (E, \mathcal{F})$ is a greedoid that satisfies the following axiom:

(AM) If $F_1, F_2 \in \mathcal{F}$ then $F_1 \cup F_2 \in \mathcal{F}$.

That is, a greedoid is an antimatroid if its feasible sets are closed under taking unions. Let $\Gamma = (E, \mathcal{F})$ be an antimatroid and let $A \subseteq E$. The set A is said to be *convex* if $E - A \in \mathcal{F}$. These are the sets whose complements are feasible. Let $\mathcal{C}$ denote the collection of convex sets of Γ. We can construct a lattice $L_{\mathcal{C}}(\Gamma)$ of Γ such that the elements correspond to the convex sets of Γ and an element X covers an element Y if $Y \subseteq X$ and $|X| = |Y| + 1$. Suppose A is an element of $L_{\mathcal{C}}(\Gamma)$, then we define

$$\mu_\Gamma(A) = \begin{cases} \mu_\Gamma(\emptyset, A) & \text{if } \Gamma \text{ is loopless, and} \\ 0 & \text{otherwise.} \end{cases}$$

Gordon and McMahon [22] prove the following theorem.

Theorem 4.1.6. *Let $\Gamma = (E, \mathcal{F})$ be an antimatroid, then*

$$p(\Gamma; \lambda) = (-1)^{|E|} \sum_{A \in L_{\mathcal{C}}(\Gamma)} \mu_\Gamma(A) \lambda^{|A|}. \tag{4.6}$$

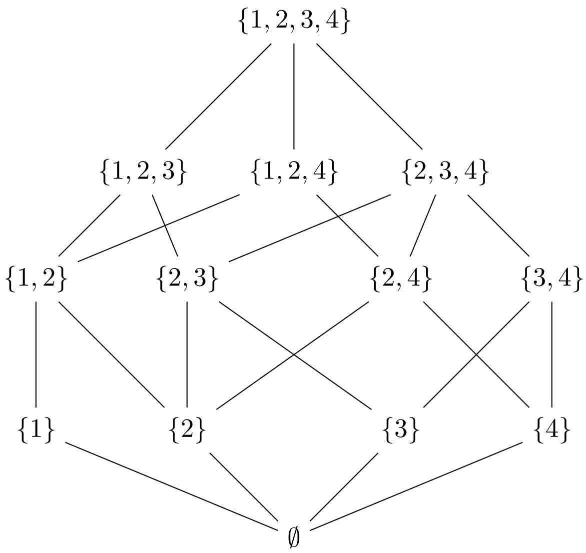

Figure 4.2: Lattice $L_{\mathcal{C}}(\Gamma)$

Example 4.1.7. *Let $\Gamma = (E, \mathcal{F})$ be the antimatroid with $E = \{1, 2, 3, 4\}$ and*

$$\mathcal{F} = \{\emptyset, \{1\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{3, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}.$$

The lattice $L_{\mathcal{C}}(\Gamma)$ of Γ is given in Figure 4.2. The convex sets of Γ together with their corresponding Möbius function are presented in the table below.

$A \in L_{\mathcal{C}}(\Gamma)$	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{2,3\}$
$\mu_{\mathcal{C}}(A)$	1	-1	-1	-1	-1	1	1
$A \in L_{\mathcal{C}}(\Gamma)$	$\{2,4\}$	$\{3,4\}$	$\{1,2,3\}$	$\{1,2,4\}$	$\{2,3,4\}$	$\{1,2,3,4\}$	
$\mu_{\mathcal{C}}(A)$	1	1	0	0	-1	0	

Using Equation 4.6 we compute

$$p(\Gamma; \lambda) = 1 - 4\lambda + 4\lambda^2 - \lambda^3.$$

For a rooted graph G we let $p(G; \lambda) = p(\Gamma(G); \lambda)$. The remainder of this section will focus on constructing a Möbius function formulation for the characteristic polynomial of a rooted graph. First we present and prove several results stated in [23].

Proposition 4.1.8. *Let $G = (r, V, E)$ be a rooted graph and let $H = (r, V, E')$ be the subgraph of G obtained by deleting all but one edge from each parallel class of G, then*

$$p(G; \lambda) = p(H; \lambda).$$

Proof. By Equation 4.4 we have $p(G; \lambda) = (-1)^{\rho(G)} T(G; 1 - \lambda, 0)$. It is not difficult to see that

$$T(G; x, y) = \sum_{A \subseteq E'} (x-1)^{\rho(H) - \rho_H(A)} (y-1)^{-\rho_H(A)} \prod_{e \in A} (y^{m(e)} - 1).$$

Now

$$T(G; 1-\lambda, 0) = \sum_{A \subseteq E'} (-\lambda)^{\rho(H) - \rho_H(A)} (-1)^{|A| - \rho_H(A)} = T(H; 1-\lambda, 0).$$

The result now follows since $\rho(G) = \rho(H)$. □

Proposition 4.1.9. *Let Γ be a greedoid with at least one loop, then*

$$p(\Gamma; \lambda) = 0.$$

Proof. This follows immediately from Proposition 1.7.1 and Equation 4.4. □

We now relate the characteristic polynomial of a rooted graph to the orientations of the graph. Let $\mathcal{O}(G)$ denote the collection of all acyclic orientations of a rooted graph G such that the root is the unique source. Note that an isolated vertex in a graph is considered to be neither a source nor a sink.

Proposition 4.1.10. *Let G be a connected rooted graph. Then*

$$|\mathcal{O}(G)| = (-1)^{\rho(G)} p(G; 0).$$

Proof. This follows immediately from Equation 4.4 and the interpretation of the evaluation of $T(G; 1, 0)$. □

A special case of a result of Tedford [53] states that if $G = (r, V, E)$ is a rooted graph then

$$p(G; \lambda) = (-1)^{\rho(G)} \sum_{k=0}^{\rho(G)} a_k(G)(1-\lambda)^k, \tag{4.7}$$

where $a_k(G)$ is the number of acyclic orientations of G with k sinks such that r is the unique source.

We now determine $p(G;\lambda)$ for three special classes of rooted graphs.

Proposition 4.1.11. *1. Let T be a rooted tree with l leaves (excluding the root if it is a leaf). Then $p(T;\lambda) = (-1)^{|E(T)|}(1-\lambda)^l$.*

2. Let C_n be the rooted cycle graph with n vertices. Then $p(C_n;\lambda) = (-1)^{n-1}(n-1)(1-\lambda)$.

3. Let K_n be the rooted complete graph with n vertices. Then $p(K_n;\lambda) = (-1)^{n-1}(n-1)!(1-\lambda)$.

Each evaluation can be shown using induction on the number of vertices in the graph. We omit the proof of the first two results and prove only the evaluation for the characteristic polynomial of a complete graph because this is considered to be the most complex.

Proof. 3. When $n = 2$ the complete graph K_2 comprises a single edge incident to the root vertex. The only orientation in which the root is the unique source is that where the edge is oriented away from the root. Therefore by Equation 4.7 we have $p(K_2;\lambda) = \lambda - 1 = (-1)1!(1-\lambda)$. Assume the result holds for $n = d-1$ where $d > 2$, i.e. $p(K_{d-1};\lambda) = (-1)^{d-2}(d-2)!(1-\lambda)$. Suppose $e_1, e_2, \ldots, e_{d-1}$ are the edges incident to the root vertex in K_d. Using Equation 4.5 and Proposition 4.1.8, we have

$$\begin{aligned} p(K_d;\lambda) &= p(K_d \setminus e_1;\lambda) - p(K_{d-1};\lambda) = \ldots \\ &= p(K_d \setminus \{e_1, e_2, \ldots, e_{d-2}\};\lambda) - (d-2)p(K_{d-1};\lambda) \\ &= \lambda^{d-1}p(K_d \setminus \{e_1, e_2, \ldots, e_{d-1}\};\lambda) - (d-1)p(K_{d-1};\lambda). \end{aligned}$$

Now $K_d \setminus \{e_1, \ldots, e_{d-1}\}$ is a disconnected graph with at least one edge not incident to the root since $d > 2$. By Proposition 4.1.9 we have $p(K_d \setminus \{e_1, \ldots, e_{d-1}\};\lambda) = 0$. Therefore

$$p(K_d;\lambda) = -(d-1)p(K_{d-1};\lambda) = (-1)^{d-1}(d-1)!(1-\lambda),$$

as required.

□

Lemma 4.1.12. *Let $G = (r, V, E)$ be a connected rooted graph and let $S \subseteq V - r$. Then there is an acyclic orientation of G with unique source r such that every vertex in S is a sink if and only if $G \setminus S$ is connected and S is independent.*

Proof. As G is connected, every vertex of G is reachable by a directed path from r in any acyclic orientation of G where r is the unique source.

Suppose that every vertex of S is a sink in an acyclic orientation $o \in \mathcal{O}(G)$. Clearly S is independent as we cannot have adjacent vertices both being sinks. Suppose $G \setminus S$ is disconnected. Let v be a vertex in $G \setminus S$ that is not in the same connected component as r. This means that there is a directed path $r \dots s \dots v$ in o for some $s \in S$. This cannot happen as s is a sink in o and thus does not have any incident edges directed away from s. Hence $G \setminus S$ must be connected.

Now let S be an independent set and $G \setminus S$ be connected. Since $G \setminus S$ is connected there exists an acyclic orientation o of $G \setminus S$ such that r is the only source. Now since S is independent we can orient the edges incident to vertices in S towards the vertices in S, obtaining an acyclic orientation in which r is the only source and every vertex of S is a sink. $\square$

Definition 4.1.13. Let $G = (r, V, E)$ be a connected rooted graph and let G' be its underlying unrooted graph. Let $M(G')$ be the cycle matroid of G'. A flat F of $M(G')$ is called *full* if there exists a set $S \subseteq V - r$ such that $F = E(G' \setminus S)$ where S is independent and $G \setminus S$ is connected.

Let $G = (r, V, E)$ be a rooted graph. Let $\mathcal{S}$ be the collection of subsets S of $V - r$ that are independent and such that $G \setminus S$ is connected. For every $S \in \mathcal{S}$ there is a one-to-one correspondence between $\mathcal{O}(G \setminus S)$ and $\mathcal{O}(G)$ where each vertex of S is a sink (with possibly other sinks) obtained by directing edges incident to vertices of S towards the vertices of S.

If G is connected and G' is its underlying unrooted graph, then $|\mathcal{O}(G)| = |\mathcal{A}(G')|$ since the number of acyclic orientations of G' with a predefined source vertex is independent of the choice of source for a connected graph.

Let $A \subseteq E$. Then $\mu_{M(G' \setminus S)}(A) = \mu_{M(G')}(A)$ if no element in A is incident to a vertex in S in G'. Also recall $r(G') = \rho(G)$ when G is connected.

We now present the main theorem of this section.

Theorem 4.1.14. *Let $G = (r, V, E)$ be a connected rooted graph and let G' be its underlying unrooted graph. Let $M = M(G')$ be the cycle matroid of G', then*

$$p(G;\lambda) = \begin{cases} \displaystyle\sum_{\substack{F \in L(M):\\ F \text{ is full}}} \mu_M(F)\lambda^{\rho(G)-\rho(F)} & \text{if } \Gamma(G) \text{ has no loops, and} \\ 0 & \text{otherwise.} \end{cases}$$

Proof. The result $p(G;\lambda)=0$ when $\Gamma(G)$ has greedoid loops is given by Proposition 4.1.9.

Let $S(o)$ be the set of sinks in the orientation $o \in \mathcal{O}(G)$. Let $\mathcal{S}$ be the collection of subsets S of $V-r$ that are independent and such that $G \setminus S$ is connected. We have

$$\begin{aligned}
p(G;\lambda) &= (-1)^{\rho(G)} \sum_{o\in\mathcal{O}(G)} (1-\lambda)^{|S(o)|} \\
&= (-1)^{\rho(G)} \sum_{o\in\mathcal{O}(G)} \sum_{U\subseteq S(o)} (-\lambda)^{|U|} \\
&= (-1)^{\rho(G)} \sum_{U\in\mathcal{S}} (-\lambda)^{|U|} \sum_{\substack{o\in\mathcal{O}(G):\\ U\subseteq S(o)}} 1 \\
&= (-1)^{\rho(G)} \sum_{U\in\mathcal{S}} (-\lambda)^{|U|} \sum_{o\in\mathcal{O}(G\setminus U)} 1 \\
&= (-1)^{\rho(G)} \sum_{U\in\mathcal{S}} (-\lambda)^{|U|} \sum_{o\in\mathcal{A}(G'\setminus U)} 1 \\
&= (-1)^{\rho(G)} \sum_{U\in\mathcal{S}} (-\lambda)^{|U|} (-1)^{r(G'\setminus U)} \mu_{M(G'\setminus U)}(E(G'\setminus U)) \\
&= (-1)^{\rho(G)} \sum_{U\in\mathcal{S}} (-\lambda)^{|U|} (-1)^{r(G')-|U|} \mu_{M(G')}(E(G'\setminus U)) \\
&= \sum_{U\in\mathcal{S}} \lambda^{|U|} \mu_{M(G')}(E(G'\setminus U)) \\
&= \sum_{\substack{F\in L(M):\\ F \text{ is full}}} \mu_{M(G')}(F)\lambda^{\rho(G)-\rho(F)}.
\end{aligned}$$

□

Example 4.1.15. *Let G be the rooted graph with root vertex r given in Figure 4.3 and let G' be its underlying unrooted graph. The lattice $L(M(G'))$ is given in Figure 4.4 in which the flats that are full in G are coloured red.*

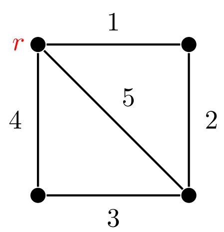

Figure 4.3: Rooted graph G

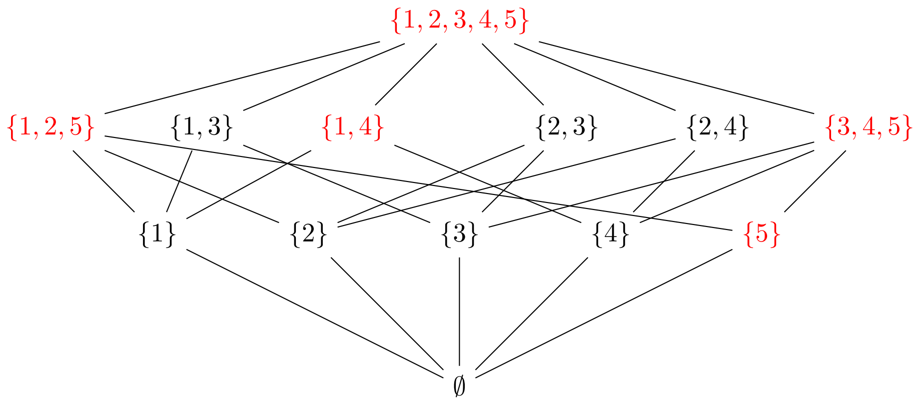

Figure 4.4: Lattice $L(M(G'))$

The full flats of G together with their corresponding Möbius function are presented in the table below.

Full $F \subseteq L(M(G'))$	$\{5\}$	$\{1,4\}$	$\{1,2,5\}$	$\{3,4,5\}$	$\{1,2,3,4,5\}$
$\mu_{M(G)}(F)$	-1	1	2	2	-4

Using Theorem 4.1.14 we compute

$$p(G;\lambda) = -\lambda^2 + 5\lambda - 4.$$

4.2 The Coefficients

In this section we determine the complexity of computing the coefficients of the Tutte polynomial of a rooted graph. We also find an expression for the minimum number of sinks taken over all acyclic orientations of a rooted graph in which the root is the unique source.

The Tutte polynomial of a graph can be expressed in the form

$$T(G;x,y) = \sum_{i,j\geq 0} b_{i,j}(G)x^i y^j.$$

We can omit the argument G when there is no risk of ambiguity.

Brylawski [9] discovered that the coefficients of the Tutte polynomial of a graph G (more generally a matroid) satisfy a collection of linear relations. In particular

$$\begin{aligned} b_{0,0}(G) &= 0 && \text{if } G \text{ has at least one edge, and} \\ b_{0,1}(G) &= b_{1,0}(G) && \text{if } G \text{ has at least two edges.} \end{aligned}$$

Note that the first relation states that there is no constant term in the Tutte polynomial of a graph with at least one edge.

The complexities of computing the coefficients of the Tutte polynomial of a graph were first considered by Annan in [2]. Annan reduces the well-known #P-complete 3-colouring problem to that of computing the coefficient $b_{1,0}$, and furthermore $b_{0,1}$ by Equation 4.8. He then deduces that for all fixed non-negative integers i, j the coefficients $b_{i,j+1}$ and $b_{i+1,j}$ are also #P-complete to compute.

We now present analogous results for the coefficients of the Tutte polynomial of a rooted graph.

Let $G = (r, V, E)$ be a rooted graph. We can similarly express the Tutte polynomial of G in the form

$$T(G; x, y) = \sum_{i,j \geq 0} b_{i,j}(G) x^i y^j.$$

Again we can omit the argument when the context is clear. By Equations 4.4 and 4.7 we have

$$T(G; x, 0) = \sum_{i \geq 0} b_{i,0}(G) x^i = \begin{cases} \sum_{o \in \mathcal{O}(G)} x^{|S(o)|} & \text{if } \mathcal{O}(G) \neq \emptyset, \text{ and} \\ 0 & \text{otherwise.} \end{cases} \tag{4.8}$$

Therefore the coefficients $b_{i,0}(G)$ give the number of acyclic orientations of G with i sinks in which r is the unique source. This implies $b_{i,0} \geq 0$ for all $i \geq 0$.

Example 4.2.1. *Figure 4.5 shows a rooted graph $G = (r, V, E)$ and the acyclic orientations of G where r is the unique source. The sinks in each orientation are coloured red. Therefore $b_{1,0}(G) = 3, b_{2,0}(G) = 1$ and $b_{i,0}(G) = 0$ for all $i \notin \{1, 2\}$. It is straightforward to check that $T(G; x, 0) = x^2 + 3x$.*

We now find an interpretation for the order of the characteristic polynomial of a rooted graph, and determine the computational complexity of computing it. Let $G = (r, V, E)$ be a rooted graph. From Equation 4.7 we may express the characteristic polynomial of G in the following form:

$$p(G; \lambda) = (-1)^{\rho(G)} \sum_{k=0}^{\rho(G)} a_k(G)(1 - \lambda)^k.$$

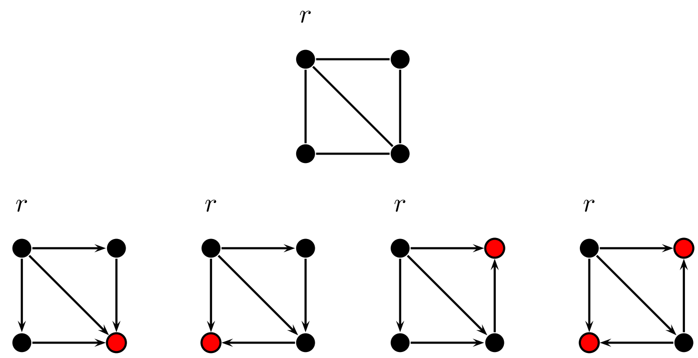

Figure 4.5

This implies that the order of $p(G;\lambda)$ is equal to the maximum number of sinks taken over all acyclic orientations in $\mathcal{O}(G)$. Recall that $\bar{\mathcal{G}}$ is the class of connected unrooted graphs. Consider the following well-known NP-complete computational decision problem.

INDEPENDENT SET

Input $G \in \bar{\mathcal{G}}$, integer k.

Question Does G have an independent set of vertices of size at least k?

We now determine the complexity of the following computational decision problem. Recall that $\mathcal{G}$ is the class of connected rooted graphs.

ORDER OF CHARACTERISTIC POLYNOMIAL

Input $G \in \mathcal{G}$, integer k.

Question Is the order of $p(G;\lambda)$ at least k?

Proposition 4.2.2. *The computational problem ORDER OF CHARACTERISTIC POLYNOMIAL is NP-complete.*

Proof. The problem is clearly in NP. Let $H \in \bar{\mathcal{G}}$. Let G be the rooted graph formed by adding a new vertex r adjacent to every vertex in H and making r the root. We claim that H has an independent set of size at least k if and only if the characteristic polynomial of G has degree at least k. If the characteristic polynomial of G has degree at least k, then by Equation 4.7, G has an acyclic orientation with r as the unique source and at least k sinks. These k sinks do not include r and, by Lemma 4.1.12, form an independent set. Thus H has an independent set of size at least k. Now suppose that H has an independent set S of size at least k. Then S is independent in G and as r is adjacent to every other vertex in G, the graph $G \setminus S$ is connected. Thus there is an acyclic orientation of G with unique source r having every vertex of S as a sink. So the degree of $p(G;\lambda)$ is at least k. Therefore the claim is true and the problem INDEPENDENT SET is reducible to ORDER OF CHARACTERISTIC POLYNOMIAL. □

We will now determine the complexity of computing the coefficient $b_{1,0}(G)$ where G is a connected rooted graph.

#COEFFICIENT OF x

Input $G \in \mathcal{G}$.

Output $b_{1,0}(G)$.

Theorem 4.2.3. *The computational problem #COEFFICIENT OF x is #P-hard to compute.*

Proof. Let $H \in \bar{\mathcal{G}}$. Create a rooted graph G from H by adding a vertex r adjacent to every vertex in H and making r the root. As H is connected, the number of acyclic orientations of H with a single predefined source is given by $T(H;1,0)$. By replacing any orientation by the one formed by reversing the direction of each edge we see that $T(H;1,0)$ also counts the number of acyclic orientations of H with a single predefined sink. Let s be a vertex of H. By orienting the edges incident to r away from r, we obtain a one-to-one correspondence between acyclic orientations of H in which s is the unique sink and acyclic orientations of G in which r is the unique source and s is the unique sink. There are $|V(H)|$ possibilities for s, so $b_{1,0}(G) = |V(H)|T(H;1,0)$. Therefore computing $b_{1,0}$ is #P-hard by Theorem 1.6.1. □

Note that when G is disconnected the problem of computing $b_{1,0}(G)$ becomes easy since $\mathcal{O}(G) = \emptyset$ and therefore $b_{1,0}(G) = 0$.

In [20] Gordon extends Brylawski's relations to rooted graphs. The first two affine relations are listed below:

$$b_{0,0}(G) = 0 \qquad \text{if } G \text{ has at least one edge,} \tag{4.9}$$

$$b_{0,1}(G) = b_{1,0}(G) \qquad \text{if } G \text{ has at least two edges.} \tag{4.10}$$

Note that Equation 4.9 agrees with the fact that there are no acyclic orientations of a rooted graph with no sinks and the root being the unique source. It is now straightforward to determine the complexity of computing the coefficient $b_{0,1}(G)$.

#COEFFICIENT OF y

Input $G \in \mathcal{G}$.

Output $b_{0,1}(G)$.

Theorem 4.2.4. *The computational problem #COEFFICIENT OF y is #P-hard to compute.*

Proof. This follows directly from Theorem 4.2.3 and Equation 4.10. $\square$

Again the problem of computing $b_{0,1}(G)$ becomes easy when G is disconnected.

Let S be the rooted graph obtained by attaching $j \geq 0$ loops to the root of the star graph S_i. By the direct sum property of the Tutte polynomial of a rooted graph we have

$$T(S; x, y) = x^i y^j.$$

Now let H be a rooted graph with Tutte polynomial

$$T(H; x, y) = \sum_{m,n \geq 0} b_{m,n}(H) x^m y^n.$$

Let G be the rooted graph obtained by identifying the root of S with the root of H. Then

$$T(G; x, y) = T(H; x, y)T(S; x, y) = \sum_{m,n \geq 0} b_{m,n}(H) x^{m+i} y^{n+j}. \tag{4.11}$$

We now extend our hardness results to include the remaining coefficients following Annan's approach [2]. Let i and j be fixed non-negative integers. Consider the following two computational problems.

#COEFFICIENT $b_{i,j+1}$

Input $G \in \mathcal{G}$.

Output $b_{i,j+1}(G)$.

#COEFFICIENT $b_{i+1,j}$

Input $G \in \mathcal{G}$.

Output $b_{i+1,j}(G)$.

Theorem 4.2.5. *#COEFFICIENT $b_{i,j+1}$ and #COEFFICIENT $b_{i+1,j}$ are #P-hard to compute.*

Proof. Given H, let G be as defined in the discussion after Theorem 4.2.4. By Equation 4.11 we have $b_{i+1,j}(H) = b_{1,0}(G)$ and $b_{i,j+1}(H) = b_{0,1}(G)$. Hence computing the coefficient $b_{i+1,j}$ and computing the coefficient $b_{i,j+1}$ are both #P-hard problems by Theorems 4.2.3 and 4.2.4. $\square$

We now present results for determining the complexity of the coefficients dependent on the input size of the rooted graph, i.e. the number of vertices and the number of edges of the graph. Let α and c be fixed constants with $0 \leq \alpha < 1$ and $0 < c \leq 1$. Consider the following two computational problems.

#COEFFICIENT $b_{\lfloor \alpha n(G)+1 \rfloor,0}$

Input $G \in \mathcal{G}$ with $n(G)$ vertices.

Output $b_{\lfloor \alpha n(G)+1 \rfloor,0}(G)$.

#COEFFICIENT $b_{\lfloor n(G)-n(G)^c+1 \rfloor,0}$

Input $G \in \mathcal{G}$ with $n(G)$ vertices.

Output $b_{\lfloor n(G)-(n(G))^c+1 \rfloor,0}(G)$.

Lemma 4.2.6. *#COEFFICIENT $b_{\lfloor \alpha n(G)+1 \rfloor,0}$ and #COEFFICIENT $b_{\lfloor n(G)-n(G)^c+1 \rfloor,0}$ are #P-hard to compute.*

Proof. Given a rooted graph H, let G be as in the proof of Theorem 4.2.5 with $i = \lfloor \frac{\alpha n(H)}{1-\alpha} \rfloor$ and $j = 0$. We have $b_{1+\lfloor \frac{\alpha n(H)}{1-\alpha} \rfloor, 0}(G) = b_{1,0}(H)$. Hence computing the coefficient $b_{1+\lfloor \frac{\alpha n(H)}{1-\alpha} \rfloor, 0}(G)$ is #P-hard by Theorem 4.2.3. We want to show that $1 + \left\lfloor \dfrac{\alpha n(H)}{1-\alpha} \right\rfloor = 1 + \lfloor \alpha n(G) \rfloor$. Let $\dfrac{\alpha n(H)}{1-\alpha} = t + k$ where $t \in \mathbb{Z}^+$ and $0 \leq k < 1$. We have

$$\begin{aligned}
\alpha n(G) &= \alpha n(H) + \alpha \left\lfloor \frac{\alpha n(H)}{1-\alpha} \right\rfloor \\
&= \frac{(1-\alpha)\alpha n(H)}{1-\alpha} + \alpha t \\
&= (t+k)(1-\alpha) + \alpha t \\
&= t + (1-\alpha)k.
\end{aligned}$$

Since $0 < 1 - \alpha \leq 1$ and $0 \leq k < 1$ we have $(1-\alpha)k < 1$. Therefore

$$\lfloor \alpha n(G) \rfloor = \lfloor t + (1-\alpha)k \rfloor = t = \left\lfloor \frac{\alpha n(H)}{1-\alpha} \right\rfloor.$$

Now repeat with $i = \lfloor (n(H))^{1/c} - n(H) \rfloor$ and $j = 0$. By Equation 4.11 we have $b_{1+\lfloor (n(H))^{1/c} - n(H) \rfloor, 0}(G) = b_{1,0}(H)$. Hence computing the coefficient $b_{1+\lfloor (n(H))^{1/c} - n(H) \rfloor, 0}(G)$ is #P-hard by Theorem 4.2.3. Since

$$n(H) = n(G) - \lfloor (n(H))^{1/c} - n(H) \rfloor$$

we have $n(G) = \lfloor (n(H))^{1/c} \rfloor$. We want to show that $1 + \lfloor (n(H))^{1/c} - n(H) \rfloor = 1 + \lfloor n(G) - (n(G))^c \rfloor$, i.e. that $\lfloor (n(H))^{1/c} - n(H) \rfloor = \lfloor n(G) - (n(G))^c \rfloor$. We have

$$\begin{aligned}
\lfloor n(G) - (n(G))^c \rfloor &= n(G) + \lfloor -(n(G))^c \rfloor = n(G) - \lceil (n(G))^c \rceil \\
&= n(G) - \left\lceil \lfloor (n(H))^{1/c} \rfloor^c \right\rceil = n(G) - n(H) \\
&= \lfloor (n(H))^{1/c} - n(H) \rfloor.
\end{aligned}$$

$\square$

Consider the following two computational problems:

#COEFFICIENT $b_{0,\lfloor \alpha m(G) \rfloor + 1}$

Input $G \in \mathcal{G}$ with $m(G)$ edges.

Output $b_{0,\lfloor \alpha m(G)\rfloor+1}(G)$.

#COEFFICIENT $b_{0,\lfloor m(G)-(m(G))^c\rfloor+1}$

Input $G \in \mathcal{G}$ with $m(G)$ edges.

Output $b_{0,\lfloor m(G)-(m(G))^c\rfloor+1}(G)$.

Lemma 4.2.7. *#COEFFICIENT* $b_{0,\lfloor \alpha m(G)\rfloor+1}$ *and #COEFFICIENT* $b_{0,\lfloor m(G)-(m(G))^c\rfloor+1}$ *are #P-hard to compute.*

Proof. The proof of this is similar to that of Lemma 4.2.6 by letting $j = \left\lfloor \dfrac{\alpha m(H)}{1-\alpha} \right\rfloor$ (respectively $\lfloor (m(H))^{1/c} - m(H)\rfloor$) in Equation 4.11 to show that the coefficient $b_{0,\lfloor \alpha m(G)\rfloor+1}$ (respectively $b_{0,\lfloor m(G)-(m(G))^c\rfloor+1}$) is #P-hard to compute. □

4.2.1 Coefficients of $T(G;x,0)$

Here we will study the minimum integer i such that for a rooted graph G, the coefficient of x^i in $T(G;x,0)$ is non-zero. First we discuss the corresponding situation for unrooted graphs.

We say that a connected graph G is *separable* if there exists a vertex v such that $G \setminus v$ is disconnected, and *nonseparable* otherwise. If such a vertex exists then it is called a *cut-vertex* of G. If a connected graph G contains a cut-vertex v then it is the vertex join of two connected subgraphs G_1 and G_2 such that $G_1 \cup G_2 = G$, $G_1 \cap G_2 = v$ and both G_1 and G_2 have at least one edge. Therefore $T(G;x,y) = T(G_1;x,y)T(G_2;x,y)$. Since $b_{0,0}(G_1) = b_{0,0}(G_2) = 0$ we must have $b_{1,0}(G) = 0$. Hence G must be nonseparable if $b_{1,0}(G) > 0$.

A *block* of a graph G is a maximal connected subgraph of G that is nonseparable. The Tutte polynomial of an unrooted graph is the product of the Tutte polynomials of its blocks. Moreover if G is nonseparable and is not a loop then $b_{1,0}(G) > 0$. So if G has no loops then the minimum integer i such that the coefficient of x^i in $T(G;x,0)$ is strictly positive equals the number of blocks of G.

The following graph tool was originally defined by Whitney in [61]. It can be used to characterize several important classes of graphs and improve the running time of some graph algorithms.

Definition 4.2.8 (Ear-Decomposition)**.** Let $G = (V,E)$ be a graph. An *ear-decomposition* of G is a partition of E into an ordered collection of edge-disjoint sets $P_0, P_1, \ldots, P_t$ such that P_0 is a cycle, and for $1 \leq i \leq t$, P_i is either a cycle with precisely one vertex in $P_0 \cup \ldots \cup P_{i-1}$ or a path with both endpoints but no internal vertices in $P_0 \cup \ldots \cup P_{i-1}$. The P_is are called the *ears* of the decomposition.

If P_i is a path for all $1 \leq i \leq t$, then the ear decomposition is said to be *open*.

Figure 4.6 shows a graph G alongside an open ear-decomposition of G.

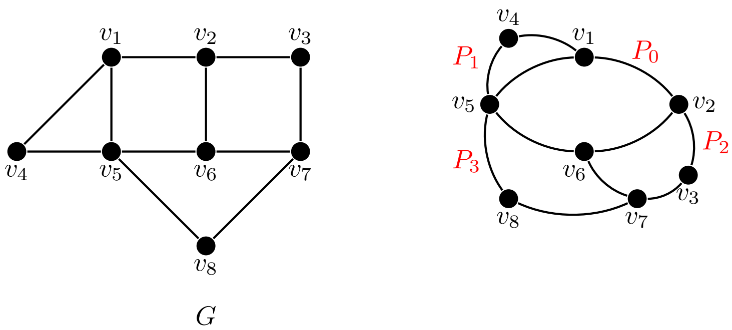

Figure 4.6

The definition of an open ear-decomposition can naturally be extended to rooted graphs.

The following is a result by Whitney [61].

Lemma 4.2.9. *A connected graph $G = (V, E)$ with $|E| \geq 2$ is nonseparable if and only if it has an open ear-decomposition.*

Let $G = (V, E)$ with $|E| \geq 2$ be a nonseparable graph. By adapting the proof from [61], if G has an open ear-decomposition, then it is possible to choose any cycle of G to be the first ear P_0. Moreover in a nonseparable graph, every pair of vertices is contained in a cycle. The next theorem is probably well-known, but we have been unable to find it stated anywhere.

Theorem 4.2.10. *Let $G = (V, E)$ be a nonseparable loopless, unrooted graph with at least one edge. For any pair r and s of vertices of G, there is an acyclic orientation of G for which the unique source is r and the unique sink is s.*

Proof. If G has only one edge, then the edge is not a loop and the result is obvious. So we may suppose that G has at least two edges. Now since G is nonseparable with $|E| \geq 2$ there exists an open ear-decomposition of G by Lemma 4.2.9. Let $P_0, P_1, \ldots, P_t$ be the ears of the open ear-decomposition. By the discussion immediately preceding the theorem, we may choose P_0 to be a cycle containing r and s.

We perform induction on the number of ears at a particular stage in the construction of our open ear-decomposition of G to show that there always exists an orientation in $\mathcal{O}(G)$ with precisely one sink.

At the stage when we have just one ear in the open ear-decomposition it will comprise solely P_0. We can orient the edges away from r and towards s in P_0 as it is a cycle. This orientation is clearly acyclic and has precisely one source r and one sink s.

Now assume that at the stage when we have ears $P_0, P_1, \ldots, P_w$ in our open ear-decomposition there exists an acyclic orientation o of the edges such that r is the unique source and s is the unique sink.

Suppose we were to attach the ear P_{w+1} to this open ear-decomposition with endpoints u and v. If either endpoint is r then we orient all edges in P_{w+1} away from it, similarly if either endpoint is s then we orient all edges in P_{w+1} towards it. If there already exists a directed path from u to v in o then we orient the edges from u to v in P_{w+1}. If none of these are the case then we orient the edges either from u to v or from v to u in P_{w+1}. In each of these cases it should be clear that we do not create any additional sources nor sinks in the orientation. Note also that orientating the edges in these ways will preserve the property of the orientation being acyclic. Suppose otherwise, that orientating the edges in P_{w+1} from u to v creates a cycle in the orientation. Then there exists a directed path from v to u already in o which means that we would have oriented the edges from v to u in P_{w+1}. If there exists a cycle between u and v regardless of which way we orient the edges in P_{w+1}, then there must already be a cycle between u and v in o, which is a contradiction as o is acyclic.

Therefore there exists an acyclic orientation of G with r being the unique source and s being the unique sink. $\square$

By Theorem 4.2.10 we know that for a nonseparable loopless, rooted graph G with at least one edge, we have $b_{1,0}(G) > 0$. If instead G is separable, the following graph tool will allow us to determine the smallest integer k such that $b_{k,0}(G) > 0$.

Definition 4.2.11 (Block Graph)**.** For a connected graph G let $B(G)$ and $C(G)$ be the set of blocks and cut-vertices of G respectively. The *block graph* $\mathcal{B}(G)$ of G is the graph with vertex set $B(G) \cup C(G)$ for which $c_i \in C(G)$ is adjacent to $b_j \in B(G)$ if and only if $c_i \in b_j$.

Figure 4.7 shows a connected graph G and the block graph $\mathcal{B}(G)$. It should be straightforward

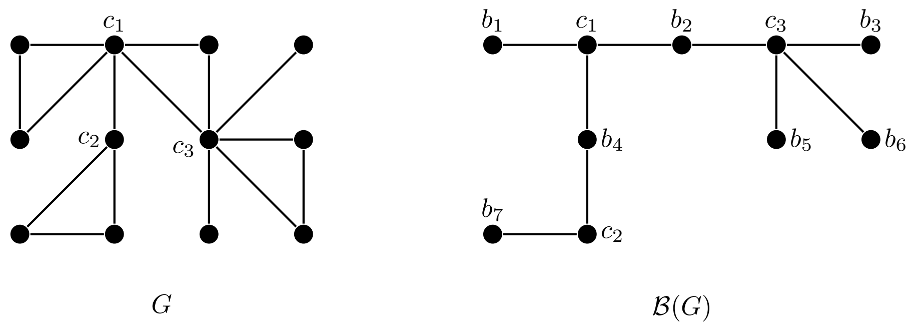

Figure 4.7

to see that the block graph $\mathcal{B}(G)$ is a bipartite tree with partite sets $B(G)$ and $C(G)$. We say that a block of G is a *leaf block* if it is a leaf vertex in $\mathcal{B}(G)$. Let $L_{\mathcal{B}}(G)$ be the set of leaf blocks of G.

Theorem 4.2.12. *Let $G = (r, V, E)$ be a connected rooted graph and let $\mathcal{B}(G)$ be the block graph of G. The smallest integer k such that $b_{k,0}(G) > 0$ is given by*

$$k = \begin{cases} |L_{\mathcal{B}}(G)| & \text{if } r \text{ is not in a leaf block of } G \text{ or } r \in C(G); \\ |L_{\mathcal{B}}(G)| - 1 & \text{if } r \text{ is in a leaf block of } G \text{ and } r \notin C(G). \end{cases}$$

Proof. We focus on the case when r is not a cut-vertex of G. The proof of the remaining case is similar and therefore omitted. An orientation of G is acyclic if and only if its restriction to each block is acyclic. In any acyclic orientation of a graph there is at least one source and one sink. Consequently in any acyclic orientation of G the restriction to each leaf block contains at least one source and one sink and either one or none of these is in $C(G)$. Hence the claimed value for k is a lower bound for the number of sinks. In a nonseparable graph we can acyclically orient the edges such that two vertices of our choice are the unique source and sink.

Let b_r be the block of G containing r. Regard $\mathcal{B}(G)$ as a tree rooted at vertex b_r with edges oriented away from b_r. We now discuss how we orient the edges for each type of block in G.

- Suppose b_r has adjacent vertices $c_1, \ldots, c_d$ in $\mathcal{B}(G)$. We acyclically orient the edges in b_r in G such that r is the unique source and any of the vertices $c_1, \ldots, c_d$ is the unique sink.

- Let b_i be a non-leaf block that does not contain r. By definition there must be at least two cut vertices in b_i. Suppose $c_1, \ldots, c_d$ are the cut vertices in b_i for $d \geq 2$ such that c_1 is the cut-vertex with the shortest path from b_r in $\mathcal{B}(G)$. We acyclically orient the edges in b_i in G such that c_1 is the unique source and any of the vertices $c_2, \ldots, c_d$ is the unique sink.

- Let b_i be a leaf block that does not contain r. By definition there must be exactly one cut-vertex c_1 in b_i. We acyclically orient the edges in b_i in G such that c_1 is the unique source and any other vertex in b_i is the unique sink.

Let o be an orientation of G defined as above. We claim that no cut-vertex is a source or a sink in o. Suppose that c is a cut-vertex. Let b denote the block adjacent to c on the shortest path from c to b_r in $\mathcal{B}(G)$. Then c is not a source in the restriction of o to b, so it is not a source in o. Moreover c is adjacent to at least one other block b' in $B(G)$ and c is a source in the restriction of o to b', so c is not a sink in o. The orientation o creates a sink in each leaf block not containing r at a vertex that is not a cut-vertex and a source at r, but otherwise, when restricted to a block only creates sources or sinks at the cut-vertices. Therefore r is the only source in o and the only sinks are those in the leaf blocks that do not contain r. If r is in a leaf block then there are $|L_{\mathcal{B}}(G)| - 1$ of these, otherwise there are $|L_{\mathcal{B}}(G)|$ of them.

□

4.3 A Convolution Formula for the Tutte Polynomial of an Interval Greedoid

Although this subsection may feel a little out of place, it would have been a missed opportunity to not include it in this thesis since all preliminary definitions have already been given in Chapter 1. As we shall see, Kook et al give a nice result for matroids so it was only natural for us to see if it holds for greedoids.

In [34] Kook, Reiner and Stanton give a convolution formula for the Tutte polynomial of a matroid. For a matroid $M = (E, \mathcal{I})$ and for all $A \subseteq E$, Kook et al express $T(M; x, y)$ as a convolution product of the flow polynomial and the chromatic polynomial of $M|A$ and M/A respectively.

Theorem 4.3.1 (Kook, Reiner, Stanton)**.** *The Tutte polynomial $T(M; x, y)$ of a matroid $M = (E, \mathcal{I})$ satisfies*

$$T(M; x, y) = \sum_{A \subseteq E} T(M|A; 0, y) T(M/A; x, 0).$$

Before presenting an overview of the proof of Theorem 4.3.1, we first define a convolution product of two functions on matroids into the ring $\mathbb{Z}[x,y]$. For a matroid $M=(E,\mathcal{I})$ let

$$(f\circ g)(M)=\sum_{A\subseteq E} f(M|A)g(M/A).$$

We now show that $\circ$ is associative, a result stated but not proved in [34]. We have

$$\begin{aligned}
[(f\circ g)\circ h](M) &= \sum_{A\subseteq E(M)} (f\circ g)(M|A)h(M/A)\\
&= \sum_{A\subseteq E(M)}\ \sum_{B\subseteq E(M|A)} f(M|A|B)g(M|A/B)h(M/A)\\
&= \sum_{B\subseteq E(M)}\ \sum_{C\subseteq E(M/B)} f(M|B)g(M|B\cup C/B)h(M/B\cup C) \quad \text{by writing } A=B\cup C\\
&= \sum_{B\subseteq E(M)}\ \sum_{C\subseteq E(M/B)} f(M|B)g(M/B|C)h(M/B/C) \qquad (4.12)\\
&= \sum_{B\subseteq E(M)} f(M|B)(g\circ h)(M/B)\\
&= [f\circ (g\circ h)](M).
\end{aligned}$$

Equation 4.12 follows from the property that contraction on a matroid is commutative.

The identity element δ of $\circ$ is defined by

$$\delta(M)=\begin{cases} 1 & \text{if } M=\emptyset;\\ 0 & \text{otherwise.}\end{cases}$$

Following results of Crapo [13], Kook et al let $\zeta(x,y)=x^{r(M)}y^{r(M^*)}$. They prove $\zeta(x,y)^{-1}=\zeta(-x,-y)$ using the property $r(M/A)=r(M)-r(M|A)$ for all subsets $A\subseteq E(M)$. They then go on to show $T(M;x+1,y+1)=(\zeta(1,y)\circ\zeta(x,1))(M)$ and subsequently

$$\begin{aligned}
\sum_{A\subseteq E(M)} T(M|A;0,y+1)T(M/A;x+1,0) &= ((\zeta(1,y)\circ\zeta(-1,1))\circ(\zeta(1,-1)\circ\zeta(x,1)))(M)\\
&= (\zeta(1,y)\circ(\zeta(-1,1)\circ\zeta(1,-1))\circ\zeta(x,1))(M) \qquad (4.13)\\
&= (\zeta(1,y)\circ\zeta(x,1))(M)\\
&= T(M;x+1,y+1).
\end{aligned}$$

Equation 4.13 follows from the associativity of $\circ$.

A natural question would be to ask if we could extend this convolution formula to greedoids. There are two important properties of a matroid that Kook et al use in the proof of Theorem 4.3.1. The first is that for a matroid $M = (E, \mathcal{I})$ the rank function satisfies $r(M/A) = r(M) - r(A)$ for all $A \subseteq E$. The second is that contraction on a matroid is commutative, that is for every pair X and Y of disjoint subsets of M, we have $M/(X \cup Y) = M/X/Y$.

We now show that for the first property to still hold in the more general greedoid setting we need M to be an interval greedoid satisfying a particular property. Let $\Gamma = (E, \mathcal{F})$ be a greedoid and let S be the set of all elements $x \in E$ such that $\{x\} \in \mathcal{F}$. If $\rho(\Gamma) = \rho(S)$ then Γ is said to be *singleton-full*.

Proposition 4.3.2. *Let $\Gamma = (E, \mathcal{F})$ be an interval greedoid. Then Γ is singleton-full if and only if $\rho(\Gamma/A) = \rho(\Gamma) - \rho_\Gamma(A)$.*

Proof. Suppose that Γ is not singleton-full. Let $A = \{e \in E : \rho(e) = 0\}$. Then A is non-empty. Note that by applying (G1′) repeatedly we see that every non-empty feasible set contains an element e with $\rho(e) = 1$. Thus $\rho_\Gamma(A) = 0$ and $\Gamma/A = \Gamma \setminus A$. Now every basis of Γ contains an element of A, so $\rho(\Gamma/A) = \rho(\Gamma \setminus A) < \rho(\Gamma)$. Thus $\rho(\Gamma/A) < \rho(\Gamma) - \rho_\Gamma(A)$.

Now suppose that Γ is singleton-full. Let A be a subset of E and let $X = \{e \in E : \rho(e) = 1\}$. As Γ is singleton-full, it has a basis B with $B \subseteq X$. Let Y be a maximal feasible set of A containing as many elements of X as possible. We have $\rho_\Gamma(A) = |Y|$. By applying (G1) repeatedly, there is a sequence of feasible sets $F_0 = Y \subseteq F_1 \subseteq \cdots \subseteq F_{\rho(\Gamma) - \rho_\Gamma(A)} = B'$ such that $|F_i - F_{i-1}| = 1$ and $B' \subseteq Y \cup B$. Suppose that $e \in (B' - Y) \cap A$. Then $\{e\}$ is feasible and for some i, $F_i - F_{i-1} = \{e\}$. We have $\emptyset \subseteq Y \subseteq F_{i-1}$, so the interval property implies that $Y \cup e$ is feasible, contradicting the choice of Y. Thus $(B' - Y) \cap A = \emptyset$. We have $\Gamma/A = \Gamma/Y \setminus (A - Y)$, so $B' - Y$ is a basis of Γ/A. Hence $\rho(\Gamma/A) = |B' - Y| = |B'| - |Y| = \rho(\Gamma) - \rho_\Gamma(A)$. ◻

We now show that if contraction on a singleton-full interval greedoid is commutative, then the greedoid is a matroid.

Proposition 4.3.3. *Let $\Gamma = (E, \mathcal{F})$ be a singleton-full interval greedoid. If for every pair X and Y of disjoint subsets of Γ, we have $\Gamma/(X \cup Y) = \Gamma/X/Y$ then Γ is a matroid.*

Proof. Suppose that Γ is not a matroid. Then by Theorem 1.3.13, there is an element x in E such that $\{x\}$ is not feasible, but x belongs to a basis B. Let $X = \{x\}$ and $Y = B - x$. Then every element of $\Gamma/(X \cup Y)$ is a loop, so $\rho(\Gamma/(X \cup Y)) = 0$. As $\{x\}$ is infeasible, $\Gamma/X = \Gamma \setminus X$. Because Γ is singleton-full, it has a basis that does not contain x, so $\rho(\Gamma/X) = \rho(\Gamma)$. Moreover Γ/X is singleton full, so $\rho(\Gamma/X/Y) = \rho(\Gamma/X) - \rho_{\Gamma/X}(Y) \geq \rho(\Gamma) - |B - x| \geq 1$. □

We have shown that if both of the properties of matroids used in Kook et al's proof hold for an interval greedoid then that interval greedoid is a matroid and the convolution formula must also hold. There may be an alternative way to prove the convolution formula but this seems unlikely, so we make the following conjecture.

Conjecture 4.3.4. *If Γ is an interval greedoid and the convolution formula of Theorem 4.3.1 holds for Γ, then Γ is a matroid.*

Bibliography

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Longman Publishing Co., Inc., 1st edition, 1974.

[2] J.D. Annan. The complexities of the coefficients of the Tutte polynomial. *Discrete Applied Mathematics*, 57(2):93 –103, 1995.

[3] S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987.

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

[5] M. Bläser and C. Hoffman. On the complexity of the interlace polynomial. *25th International Symposium on Theoretical Aspects of Computer Science*, pages 97–108, 2008.

[6] H. L. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11(1-2):1–21, 1993.

[7] H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.

[8] H. L. Bodlaender and D. Thilikos. Constructive linear time algorithms for branchwidth. In *Automata, Languages and Programming*, volume 1256, pages 627–637. Lecture Notes in Computer Science, 1997.

[9] T. Brylawski. A decomposition for combinatorial geometries. *Transactions of the American Mathematical Society*, 171:235–282, 1972.

[10] T. Brylawski. The Tutte polynomial part I: General theory. In A. Barlotti, editor, *Matroid Theory and its Applications*. Proceedings of the Third International Mathematical Summer Center (C.I.M.E. 1980), 1982.

[11] T. Brylawski and J. Oxley. The Tutte polynomial and its applications. In N. White, editor, *Matroid Applications*. Cambridge University Press, 1992.

[12] B. Courcelle, J. A. Makowsky, and U. Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discrete Applied Mathematics*, 108(1):23–52, 2001.

[13] H. H. Crapo. The Tutte polynomial. *Aequationes Mathematicae*, 3(3):211–229, 1969.

[14] R. Diestel. *Graph Theory*. Springer, 2006.

[15] J. Edmonds. Systems of distinct representatives and linear algebra. *Journal of Research of the National Bureau of Standards*, B 71(4):241–241, 1967.

[16] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971.

[17] J. Ellis-Monaghan and C. Merino. Graph polynomials and their applications I: The Tutte polynomial. In M. Dehmer, editor, *Structural analysis of complex networks*. Birkhäuser Boston, 2011.

[18] M. R. Garey and D. S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[19] O. Goecke. A greedy algorithm for hereditary set systems and a generalization of the Rado-Edmonds characterization of matroids. *Discrete Applied Mathematics*, 20(1):39 – 49, 1988.

[20] G. Gordon. Linear relations for a generalized Tutte polynomial. *Electronic Journal of Combinatorics*, 22(1), 2015.

[21] G. Gordon and E. McMahon. A greedoid polynomial which distinguishes rooted arborescences. *Proceedings of the American Mathematical Society*, 107(2):287–298, 1989.

[22] G. Gordon and E. McMahon. A greedoid characteristic polynomial. *Contemporary Mathematics*, 197:343–351, 1996.

[23] G. Gordon and E. McMahon. A characteristic polynomial for rooted graphs and rooted digraphs. *Discrete Mathematics*, 232(1):19–33, 2001.

[24] R. Gould. *Graph Theory*. Dover Publications, 2012.

[25] C. Greene and T. Zaslavsky. On the interpretation of Whitney numbers through arrangements of hyperplanes, zonotopes, non-radon partitions, and orientations of graphs. *Transactions of the American Mathematical Society*, 280(1):97–126, 1983.

[26] R. Halin. *S*-functions for graphs. *Journal of Geometry*, 8(1-2):171–186, 1976.

[27] I. V. Hicks and N. B. McMurray. The branchwidth of graphs and their cycle matroids. *Journal of Combinatorial Theory, Series B*, 97(5):681–692, 2007.

[28] P. Hliněný. A parametrized algorithm for matroid branch-width. *SIAM Journal on Computing*, 35(2):259–277, 2005.

[29] P. Hliněný. The Tutte polynomial for matroids of bounded branch-width. *Combinatorics, Probability and Computing*, 15(3):397–409, 2006.

[30] P. Hliněný and G. Whittle. Matroid tree-width. *European Journal of Combinatorics*, 27(7):1117–1128, 2009.

[31] F. Jaeger, D. L. Vertigan, and D. J. A. Welsh. On the computational complexity of the Jones and Tutte polynomials. *Mathematical Proceedings of the Cambridge Philosophical Society*, 108(1):35–53, 1990.

[32] M. Jerrum. Counting trees in a graph is #P-complete. *Information Processing Letters*, 51(3):111–116, 1994.

[33] T. Kloks, J. Kratochvl, and H. Müller. Computing the branchwidth of interval graphs. *Discrete Applied Mathematics*, 145(2):266–275, 2005.

[34] W. Kook, V. Reiner, and D. Stanton. A convolution formula for the Tutte polynomial. *Journal of Combinatorial Theory Series B*, 76(2):297–300, 1999.

[35] B. Korte and L. Lovász. Mathematical structures underlying greedy algorithms. In Ferenc Gécseg, editor, *Fundamentals of Computation Theory: Proceedings of the 1981 International FCT-Conference*, pages 205–209. Springer Berlin Heidelberg, 1981.

[36] B. Korte and L. Lovász. *Shelling structures, convexity and a happy end.* Inst. für Ökonometrie u. Operations-Research, 1983.

[37] B. Korte and L. Lovász. Structural properties of greedoids. *Combinatorica*, 3(3/4):359–374, 1983.

[38] J. A. Makowsky. Coloured Tutte polynomials and Kauffman brackets for graphs of bounded tree width. *Discrete Applied Mathematics*, 145(2):276–290, 2005.

[39] J. A. Makowsky. Private communication, 2015.

[40] J. A. Makowsky and J. P. Mariño. Farrell polynomials on graphs of bounded tree width. *Advances in Applied Mathematics*, 30(1):160–176, 2003.

[41] F. Mazoit and S. Thomassé. Branchwidth of graphic matroids. In *Surveys in Combinatorics 2007, London Mathematical Society Lecture Notes* 346, pages 275–286, 2007.

[42] S. Noble. Evaluating the Tutte polynomial for graphs of bounded tree-width. *Combinatorics, Probability and Computing*, 7(3):307–321, 1998.

[43] J. G. Oxley. *Matroid Theory.* Oxford University Press, 1992.

[44] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[45] J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4):777–788, 1983.

[46] N. Robertson and P. D. Seymour. Graph minors. III. planar tree-width. *Journal of Combinatorial Theory, Series B*, 36(1):49–64, 1984.

[47] N. Robertson and P. D. Seymour. Graph minors. X. obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991.

[48] G. Rota. On the foundations of combinatorial theory I. theory of Möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 2(4):340–368, 1964.

[49] P. Scheffler. *Linear-time algorithms for NP-complete problems restricted to partial k-trees.* Akademie der Wissenschaften der DDR, Karl-Weierstrass-Institut für Mathematik, 1987.

[50] P. D. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.

[51] R. P. Stanley. Acyclic orientations of graphs. *Discrete Mathematics*, 5(2):171–178, 1973.

[52] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[53] S. J. Tedford. A characteristic polynomial for rooted mixed graphs. *Discrete Mathematics*, 304(1):121–127, 2005.

[54] A. M. Turing. On computable numbers, with an application to the Entscheidungs problem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1936.

[55] W. T. Tutte. A ring in graph theory. *Mathematical Proceedings of the Cambridge Philosophical Society*, 43:26–40, 1947.

[56] W. T. Tutte. *Graph Theory*. Cambridge University Press, 2001.

[57] M. Las Vergnas. Acyclic and totally cyclic orientations of combinatorial geometries. *Discrete Mathematics*, 20:51–61, 1977.

[58] D. L. Vertigan. The computational complexity of Tutte invariants for planar graphs. *SIAM Journal on Computing*, 35(3):690–712, 2005.

[59] D. L. Vertigan and D. J. A. Welsh. The computational complexity of the Tutte plane: the bipartite case. *Combinatorics, Probability and Computing*, 1(2):181–187, 1992.

[60] H. Whitney. A logical expansion in mathematics. *Bulletin of the American Mathematical Society*, 38(8):572–580, 1932.

[61] H. Whitney. Non-separable and planar graphs. *Transactions of the American Mathematical Society*, 34(2):339–362, 1932.

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

[63] T. V. Wimer. *Linear Algorithms on K-terminal Graphs*. PhD thesis, Clemson University, 1987.

[64] T. Zaslavsky. The Möbius function and the characteristic polynomial. In N. White, editor, *Combinatorial Geometries*. Cambridge University Press, 1987.