Tesis Doctoral

Métodos algebraicos para problemas discretos

Tobis, Enrique Augusto

2009

Cita tipo APA:

Tobis, Enrique Augusto. (2009). Métodos algebraicos para problemas discretos. Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires.

Cita tipo Chicago:

Tobis, Enrique Augusto. "Métodos algebraicos para problemas discretos". Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires. 2009.

UNIVERSIDAD DE BUENOS AIRES

Facultad de Ciencias Exactas y Naturales

Departamento de Computación

Métodos Algebraicos para Problemas Discretos

Tesis presentada para optar al título de Doctor de la Universidad de Buenos Aires en el área Ciencias de la Computación.

Enrique Augusto Tobis

Directora de Tesis: Dra. Alicia Dickenstein

Consejera de Estudios: Dra. Verónica Becher

Lugar de Trabajo: Departamento de Matemática, FCEyN, UBA

Buenos Aires, 2009

Métodos Algebraicos para Problemas Discretos

Resumen

En esta tesis estudiamos tres problemas que relacionan Teoría de Grafos y Álgebra. En particular, consideramos el problema de contar el número de conjuntos independientes en un grafo, así como el problema relacionado de contar el número de anticadenas en un conjunto parcialmente ordenado, desde la perspectiva del álgebra computacional. También describimos los conjuntos independientes máximos de los grafos de de Bruijn $B(d, 3)$, vía el estudio de la acción del grupo simétrico en d elementos. Además, determinamos todos los etiquetamientos aditivos de aristas y de vértices módulo d en un grafo, por medio de una traducción combinatoria de los correspondientes problemas de álgebra lineal sobre el anillo de enteros módulo d.

Palabras clave: grafo, conjunto parcialmente ordenado, conjunto independiente, anticadena, serie de Hilbert, grafo de de Bruijn, etiquetamiento aditivo de aristas, etiquetamiento aditivo de vértices, complejidad

Algebraic Methods for Discrete Problems

Abstract

In this thesis we study three problems that link Graph Theory and Algebra. In particular, we consider the problem of counting independent sets in a graph, as well as the related problem of counting antichains in a finite partially ordered set, from the perspective of computational algebra. We also completely describe the maximum independent sets of the de Bruijn graphs $B(d, 3)$, via the study of the action of the symmetric group on d elements. Moreover, we determine all additive edge and vertex labelings modulo d on a graph, by combinatorially translating the corresponding linear algebra problems over the ring of integers modulo d.

Keywords: graph, poset, independent set, antichain, Hilbert series, de Bruijn graph, additive edge labeling, additive vertex labeling, complexity.

Agradecimientos

Alicia Dickenstein me condujo, a lo largo de todos estos años, por el camino que me trajo hasta acá. Su guía, sus conocimientos, y sobre todo su paciencia, hicieron posible que terminara este trabajo. ¡Gracias, Alicia!

Mi otra compañera (¡y coautora!) inflaqueable a lo largo de todo este proceso fue Angie. Estuvo siempre a mi lado, aún a la distancia, apoyándome en todo momento. Sin ella y su cariño, no habría podido. ¡Gracias, Angie!

A Verónica Becher, "for going above and beyond the call of duty" de una Consejera de Estudios. ¡Gracias, Vero!

Sin mi familia, puntualmente mis padres, no existiría. Además todos siempre me han apoyado en mis proyectos. ¡Gracias, Familia!

El único coautor a quien aún no agradecí es Dustin Cartwright, así que: ¡Gracias, Dustin, por la magia!

Desde 1998, me siento parte de una linda tribu: el Departamento de Computación. Me enseño muchas cosas, y me ayudó a crecer en muchos sentidos. Me dio amigos y me dio conocimientos. ¡Gracias, DC!

A lo largo del desarrollo de esta tesis, varias personas me ayudaron con sugerencias o comentarios atinados. Ellos son Jonathan Barmak, Anna Bigatti, Eduardo Cattani, María Chudnovsky, Jim Geelam, Marina Groshaus, Gabriela Jerónimo, Julia Kempe, Min Chih Lin, Jesús de Loera, Gabriel Minián, Hans Schönemann, Michael Stillman, Bernd Sturmfels, Mauricio Velasco, Josephine Yu y C.Q. Zhang. ¡Gracias a todos!

Tres personas leyeron esta Tesis con atención, y me hicieron comentarios invaluables. A mis tres Jurados, Min Chih Lin, Juan Sabia y Rafael Villarreal, ¡Gracias!

A todos mis amigos de las milongas. Ahí no importa si sale la demostración, o si aceptan el artículo. Cada tres minutos, la vida empieza de nuevo. ¡Gracias, Derrapados!

A mis amigos de la Compañía. Por tantas emociones compartidas, y por darme motivos para gritarles ¡Bravo! después de cada función. ¡Gracias, Coreutas!

Durante mi doctorado, realicé dos visitas científicas a centros del extranjero. A la gente del grupo Galaad de Francia, dirigidos por Bernard Mourrain, y a la gente del Institute for Mathematics and its Applications, ¡Gracias!

Durante estos años, el Departamento de Matemática se ocupó de que tuviera un escritorio para sentarme y pensar en mi trabajo. Y como si eso no fuera suficiente, me dio excelentes compañeros de oficina. ¡Gracias, DM!

Y por último, pero ciertamente no menos importante, le quiero agradecer a las tres instituciones que hicieron todo esto financieramente posible: el CONICET, la ANPCyT y la UBA. A las tres les estoy muy agradecido por el dinero que me dieron a cambio de esta Tesis. ¡Gracias!

A Angie.

Contents

The second one is that we can obtain the maximum independent sets of $B(d,3)$ recursively from those of $B(d-1,3)$ and those of $B(d-2,3)$. We give four explicit transformations (called f, f', g and g') to achieve this. In fact, all four transformations map orbits under the action of $\mathbb{S}_{d-1}$ or $\mathbb{S}_{d-2}$, respectively, into orbits under the action of $\mathbb{S}_d$.

These two observations combined allow us to prove Theorem 3.4.5, the main result of the chapter. That is, every maximum independent set of $B(d,3)$ can be obtained by an appropriate sequence of applications of the four functions we define, and permutations in $\mathbb{S}_d$, starting from the maximum independent sets of $B(1,3)$ and $B(2,3)$, which can be computed by an exhaustive procedure. As a consequence of this construction, we prove that the number a_d of maximum independent sets of $B(d,3)$ follows the recursion

$$\begin{cases} a_1 = 1, \\ a_2 = 6, \\ a_d = 2da_{d-1} + d(d-1)a_{d-2} & \text{for } d \geq 3. \end{cases}$$

This solves the problem completely. Furthermore, we discuss a way of generalizing some of our results to larger primes D, provided the size conjectured in [43] is correct.

The last chapter contains our study of additive graph labelings. Let $G = (V, E)$ be a simple graph. Let d be a positive integer, and note with $\mathbb{Z}_d$ the integers modulo d. We study two problems which are closely related.

Given a labeling $f_E \colon E \to \mathbb{Z}_d$ of the edges of G, is there a labeling $f_V \colon V \to \mathbb{Z}_d$ such that

$$f_E((u,v)) = f_V(u) + f_V(v),$$

for every edge $(u,v) \in E$? If so, how many are there? If there is a solution, we say that the labeling f_E is e-additive.

The second problem is dual to the first, and it starts with a vertex labeling $f_V \colon V \to \mathbb{Z}_d$, and asks whether there is a labeling $f_E \colon E \to \mathbb{Z}_d$ such that the label that f_V gives to each vertex is the sum of the labels that f_E gives to the edges incident ot it. That is,

$$f_V(u) = \sum_{(u,v)\in E} f_E((u,v)),$$

for every vertex $u \in V$. If so, how many are there? Likewise, we say that the labeling f_V is v-additive if there is a solution.

We give a definitive answer to both problems. A labeling f_E is e-additive if and only if the cycles of G have two properties that we state in Definition 4.1.11. On the other hand, a labeling f_V is v-additive if and only if the labels of the graph satisfy a more global property that we present in Definition 4.1.14. This theoretical characterization is presented in Theorem 4.1.16.

The theoretical results are interesting in themselves, but we also have the following complexity statement.

Theorem. *Let G be a graph.*

- *Given a labeling $f_E \colon E \to \mathbb{Z}_d$, we can decide in polynomial time whether f_E is e-additive.*
- *Given a labeling $f_V \colon V \to \mathbb{Z}_d$, we can decide in polynomial time whether f_V is v-additive.*

From our study of these two problems, we also obtain a description of the integer kernel, modulo d, of the incidence matrix A_G of G, as well as of its transpose A_G^t. This description allows us to compute the number of solutions to our two labeling problems, which we summarize in Theorem 4.1.17. In a way, our graphical conditions for additivity are a novel combinatorial interpretation of a linear algebra problem.

In fact, our results can be extended to labelings which assign labels not only in $\mathbb{Z}_d$, but in any arbitrary abelian group. We discuss this generalization at the end of the chapter.

Finally, we include two appendices. As some of them are not part of the standard CS curriculum, we summarize all the algebraic tools and techniques that we use in Appendix A. The exposition should be accessible to anyone with a modest background in Abstract Algebra. Appendix B contains a brief summary of the syntax for computing a Hilbert Series in a few widely used free Computer Algebra Systems.

Chapter 1

Known Methods for Counting and Enumerating Independent Sets in Graphs

1.1 Introduction

Counting independent sets in an arbitrary graph, or antichains in an arbitrary finite partially ordered set, is a #P-complete problem [49] with manifold applications. For example, the authors of [4] define a partially ordered set of gene mutations and work with the lattice of its antichains to predict the development of drug resistance in HIV.

In this chapter, we review some results concerning independent sets in graphs. Unless otherwise noted, we work with simple undirected graphs.

Let $G = (V, E)$ be a graph, an *independent set* of G is a set $S \subseteq V$ such that $S \times S \cap E = \emptyset$. Figure 1.1a shows the independent set $\{1, 3, 5\}$ highlighted. Independent sets are also called stable sets. The size of a maximum independent set of a graph G is called the *stability* or *independence* number of G, and is noted $\alpha(G)$.

There are two other objects intimately related to independent sets. Given a graph G, a *clique* is a maximal complete induced subgraph. An independent set in G is a clique in the complement of G. Figure 1.1b shows the set $\{1, 3, 5\}$ as a clique in the complement of the graph of Figure 1.1a. The size of a largest clique of G is its *clique number* $\omega(G)$. The other object is a *vertex cover*. A set of nodes C is a vertex cover of a graph G if every edge of G has at least one endpoint in C. A vertex cover is the complement of an independent set.

We are usually concerned with *maximal* (i.e. not included in another one) and *maximum* (i.e. of maximum cardinality) independent sets and

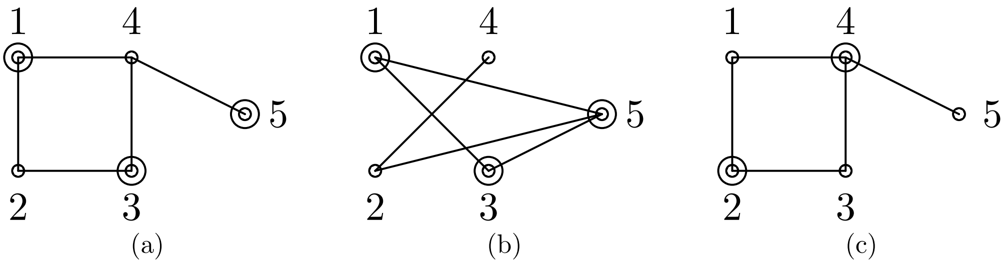


Figure 1.1: (a) An independent set in a graph. (b) The same set as a clique in its complement. (c) A cover of the original graph.

cliques, and with *minimal* and *minimum* vertex covers. The examples of Figure 1.1 are maximum (independent set and clique) and minimum (vertex cover). There are decision problems attached to these. Given a graph G and a positive integer k, one can ask whether there is a independent set of cardinality greater than k. This, and the equivalent clique and vertex cover versions, are NP-complete [26]. All three problems have their weighted versions.

These three objects arise in several applications. For example, a clique-detection algorithm is applied to a chemical problem in [22]. Independent sets play a relevant role in coding theory ([54]).

Even though the three decision problems stated above are NP-Complete, there are polynomial algorithms that solve them for several classes of graphs. For example, a polynomial algorithm for the maximum independent set problem (i.e. computing the size of a maximum independent set) in perfect graphs is presented in [28] (though it is not combinatorial in nature). Recall that a graph G is perfect if and only if the chromatic number $\chi(G')$ equal the clique number $\omega(G')$ for every induced subgraph G' of G. Perfect graphs include bipartite graphs, comparability graphs and interval graphs, among others.

1.2 Some methods for counting independent sets in a graph

In this work, we are interested in *counting* independent sets in a graph, which, as we said, is a #P-complete problem.

A useful object in this study is the *independence polynomial* of the graph. Given a graph G, its independence polynomial $I(G; x)$ has degree $\alpha(G)$, and the coefficient of x^i is the number of independent sets of size i in G. The survey [42] summarizes several known results about independence polynomials.

Clearly, evaluating the independence polynomial at 1 gives the total number of independent sets of G. In Chapter 2, we study some properties of an algebraic algorithm for computing the indepence polynomial of a graph. In Chapter 3, we find a closed formula for the number of maximum independent sets of a subfamily of de Bruijn graphs.

We start with a divide and conquer algorithm for computing the independence polynomial of a graph. The algorithm applies the following result.

Proposition 1.2.1 (see [29]). *Let $G = (V, E)$ be a graph, and let $v \in V$ be a node. Let $N[v]$ be the* closed neighborhood *of v (i.e. v and all of its adjacent nodes). Then*

$$I(G; x) = I(G \backslash v; x) + xI(G \backslash N[v]; x).$$

There is a simple way to derive an algorithm from this result: Choose a node v of G, call the algorithm recursively with $G \backslash v$ and with $G \backslash N[v]$, and then combine the two results. This is a rather crude algorithm, but we present it because it will resurface in Chapter 2. This algorithm can be modified to *enumerate* the independent sets.

Reverse Search is a method for combinatorial enumeration introduced in [1]. This method was applied to the enumeration of all maximal independent sets in a graph in [20].

There are other approaches to the counting problem. For example, the Belief Propagation heuristic was used in [10]. Binary Decision Diagrams ([38]) are applied to this problem in [63].

We close this chapter with an exposition of quantum algorithms for our problems.

1.3 Quantum Counting

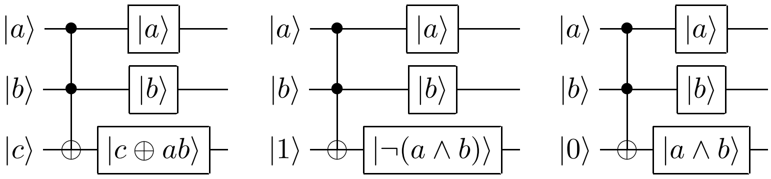
In this section, we show a quantum algorithm to *enumerate* all independent sets in a graph, and an algorithm to *count* the independent sets of a graph. We recommend [11] as a general reference on quantum algorithms.

Quantum computing promised to be a groundbreaking platform for computations. The best-known, and most spectacular, result is perhaps Peter Shor's polynomial time algorithm for the factorization of an integer [52]. The unstructured nature of our problem, however, prevents us from obtaining an efficient algorithm, as was shown in [5].

A classical circuit Given a set V of n elements, we can represent its subsets by strings of n 0s and 1s. Given a graph $G = (V, E)$, $|V| = n$, $|E| = m$ we are going to build a quantum circuit to recognize its independent sets. It will take as input an n-qubit register $|x_1 \dots x_n\rangle = |x\rangle$, which will represent a subset of V. If $|x\rangle$ represents an independent set of G, it will be transformed into $-|x\rangle$. Otherwise, it will remain untouched.

First, we discuss a classical circuit to decide whether a subset $x \subseteq V$ is an independent set of G. For every edge (i, j) of G, we can NAND the values of x_i and x_j. That will give us m boolean values, which will be 1 if and only if the corresponding edge is absent from x. We now have to AND all those values. The result will be 1 if and only if x is independent.

A quantum circuit Now we discuss the quantum circuit. Since all the operations have to be reversible, we will resort to the Toffoli gate (see Figure 1.2a).



(a) Toffoli gate (b) Logical NAND (c) Logical AND

Figure 1.2: Reversible logical gates

To build the quantum circuit, we will use n qubits corresponding to $|x\rangle$, m qubits corresponding to the edges, m qubits to summarize information, and one target qubit. The NAND operations of the classical circuit are translated into Toffoli gates (see Figure 1.2b) using nodes as control qubits and an ancillary $|1\rangle$. As pointed out above, that will leave us with the m ancillary qubits reflecting whether a certain edge is present in $|x\rangle$ or not. Let us call them $|e_1\rangle, \dots, |e_m\rangle$. The following step is to perform an AND operation between these $|e_i\rangle$. That will leave us with one qubit which will be $|1\rangle$ if and only if $|x\rangle$ is an independent set of G. We then use this qubit to control a NOT operation on the target qubit.

For example, given the graph $P_4 = (\{1, 2, 3, 4\}, \{(1, 2), (2, 3), (3, 4)\})$, a simple path of four nodes, the corresponding quantum circuit is illustrated in Figure 1.3. In this instance, $|q_1\rangle$ through $|q_4\rangle$ represent the nodes of P_4. After the first three gates have been applied, the state of $|q_5\rangle$, $|q_6\rangle$ and $|q_7\rangle$ will reflect the presence in $|x\rangle$ of the edges $(1, 2)$, $(2, 3)$ and $(3, 4)$, respectively. This information will be summarized in the state of $|q_9\rangle$, which will control the $CNOT$ on the target qubit.

We will call this quantum circuit O. It uses m Toffoli gates to perform NAND operations, $m - 1$ to perform AND operations, two Hadamard gates

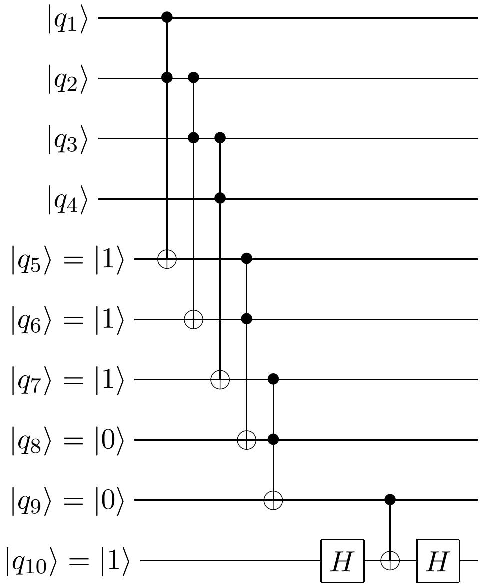

$|q_1\rangle$
$|q_2\rangle$
$|q_3\rangle$
$|q_4\rangle$
$|q_5\rangle = |1\rangle$
$|q_6\rangle = |1\rangle$
$|q_7\rangle = |1\rangle$
$|q_8\rangle = |0\rangle$
$|q_9\rangle = |0\rangle$
$|q_{10}\rangle = |1\rangle$
H
H

Figure 1.3: Oracle for P_4

and one $CNOT$ gate, for a total of $2m+1$ basic quantum gates.

The Grover iteration If we consider the circuit O as an oracle, we can build the box Gr for the Grover iteration

$$H^{\otimes n}(I - 2|0\rangle\langle 0|)H^{\otimes n}O$$

This box (see Figure 1.4) has one call to our oracle, two n-bit Hadamard transforms and one conditional phase shift. The latter operation can be realized with $O(n)$ gates. Therefore, Gr can be realized with $O(n+m)$ basic gates.

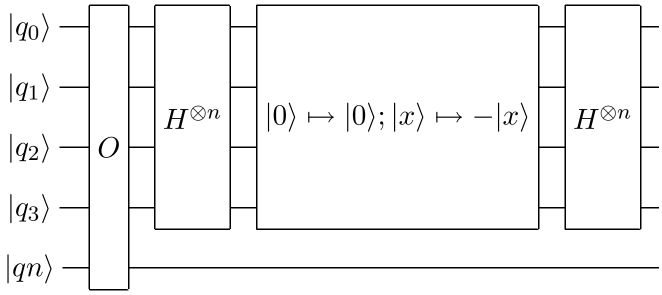


Figure 1.4: Grover iteration

Finding an Independent Set Let the number of independent sets of G be M, and let us assume that we know this number. Suppose that we want to find **one** of those M independent sets. Let $|\psi\rangle = H^{\otimes n}|0\rangle$ be the uniform superposition of all the base states. Applying the Grover iteration $O(\sqrt{\frac{2^n}{M}})$ times we will obtain an independent set of G with high probability. To see why, let us point out that

$$|\psi\rangle = \sqrt{\frac{2^n - M}{2^n}}|\alpha\rangle + \sqrt{\frac{M}{2^n}}|\beta\rangle$$

where

$$|\alpha\rangle = \frac{1}{\sqrt{2^n - M}} \sum_{x \text{ not independent}} |x\rangle$$

and

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{x \text{ independent}} |x\rangle$$

If we let $\theta/2$ be the initial angle between $|\psi\rangle$ and $|\alpha\rangle$, then each Grover iteration brings it an angle θ closer to $|\beta\rangle$. This can be proved using trigonometric identities. It follows that $O(\sqrt{\frac{2^n}{M}})$ iterations bring it closest to $|\beta\rangle$. Taking into account the complexity of evaluating Gr, the complexity of finding **1** independent set amounts to $O(\sqrt{\frac{2^n}{M}}(n+m))$.

Finding all the Independent Sets If we know M, and we want to enumerate all the independent sets, we can find one, say $|x\rangle$, and then change the oracle O so that neither $|x\rangle$ nor any of its subsets are solutions. This change involves $|x| = O(n)$ new gates. We then have to find one of the $M - 2^{|x|}$ solutions of a new function. Repeating this procedure, we can obtain all the independent sets.

Counting Quantumly If we want to estimate the number M of independent sets of the graph G, we can now use quantum counting ([7]) (see Figure 1.5). The procedure estimates an eigenvalue of the Grover iteration we have just described. Actually, the eigenvalue is of the form $e^{i\theta}$ and the algorithm approximates $\theta = 2 \arcsin \sqrt{\frac{M}{2^n}}$. If we want to approximate it with l bits of accuracy, and we want the algorithm to succeed with probability at least $1 - \epsilon$, we have to use $t = l + \lceil \log(2 + 1/2\epsilon) \rceil$ qubits, plus the qubits for G. In this circuit, the complexity is dominated by the $2^l - 1$ applications of the G iteration, resulting in an $O(2^l(m+n))$ algorithm.

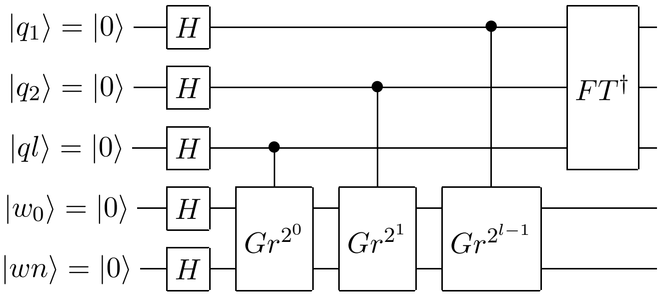


Figure 1.5: Quantum Counting via Phase Estimation

Chapter 2

Independent Sets from an Algebraic Perspective

2.1 Introduction

In this chapter, we study the problem of counting independent sets in a graph, as well as the related problem of counting antichains in a finite partially ordered set, both from an algebraic perspective. The connection between independent sets and Algebra is spearheaded by the following construction.

Definition 2.1.1. Let $G = (V, E)$ be an arbitrary graph, with $V = \{v_1, \ldots, v_n\}$. The *edge ideal* ([53, 62]) $I'_G \subseteq \mathbb{C}[x_1, \ldots, x_n]$ of G is defined as

$$I'_G = \langle x_i x_j, \text{ for all } (v_i, v_j) \in E \rangle. \tag{2.1.1}$$

Here, $\mathbb{C}$ can be any field.

This ideal links independent sets in G and polynomials: If x^α is a monomial *not* in I'_G, then it encodes an independent set S of G, with the encoding given by

$$v_i \in S \Leftrightarrow x_i \,|\, \mathbf{x}^\alpha. \tag{2.1.2}$$

This encoding is not one-to-one, though. For example, the monomials x_1 and x_1^2 represent the same independent set: $\{v_1\}$. In order to obtain a better encoding, we introduce a slightly modified version of I'_G.

Definition 2.1.2. Let $G = (V, E)$ be a graph. We define the *modified edge ideal* I_G of G as

$$I_G = I'_G + \langle x_i^2, \text{ for all } i \rangle. \tag{2.1.3}$$

Some results from this chapter appear in [18].

Notice that I_G is zero-dimensional (in fact, it has only one root: $\mathbf{0}$), and that the monomials *not* in I_G are square-free. The latter property turns the encoding in (2.1.2) into a bijection. The degree of a monomial now corresponds to the size of the independent set it represents. These monomials are relevant enough to have a special name.

Definition 2.1.3. Given a zero-dimensional polynomial ideal I, the monomials *not* in I are the *standard monomials of* I.

As an example of their usefulness, consider this: The quotient $\mathbb{C}[\mathbf{x}]/I$ is a finite-dimensional $\mathbb{C}$-vector space, and the standard monomials of I are a basis of this vector space ([14, Chapter 2, Section 2]).

If a zero-dimensional ideal I is generated by monomials, then we can define an object that gives us detailed information about the structure of its standard monomials.

Definition 2.1.4. Let I be a zero-dimensional ideal I. Let i be a non-negative integer. The *Hilbert Function* (HF_I) maps i into the number of standard monomials of I that have degree i.

As is often the case in combinatorics, we are also interested in the generating function of HF.

Definition 2.1.5. Let I be a zero-dimensional ideal I. We define the *Hilbert Series* ($HS_I(z)$) of I as the generating function

$$HS_I(z) = \sum_{0\leq i} HF_I(i)z^i.$$

Now, let G be any graph, and let I_G be its modified edge ideal. In light of the encoding in expression (2.1.2), the coefficient of z^i in HS_{I_G} is the number of independent sets of G that have i elements. Then, by definition, $HS_{I_G}(z)$ is precisely the *independence polynomial* of G (see [42] for a survey on the subject). Therefore, the Hilbert Series of the modified edge ideal of a graph G gives us a lot of information about the independent sets of G.

There is a standard algorithm for computing the Hilbert Series of I_G, and we explain it in detail in Section 2.2. One of our main results in this chapter concerns the running time of this algorithm when applied to ideals I_G with special algebraic properties. We now introduce these ideals.

A *partially ordered set* is a set P, together with a relation $\leq$ satisfying

- $a \leq a$, for all $a \in P$.
- $a \leq b$ and $b \leq a$ implies $a = b$, for all a and b in P.

- $a \leq b$ and $b \leq c$ implies $a \leq c$ for all a, b and c in P.

We say that two elements a and b of P are *comparable* if $a \leq b$ or if $b \leq a$. Otherwise, we say they are *incomparable*. A subset A of P is called an *antichain* if all the elements of A are pairwise incomparable in P. We write $A(P)$ for the set of antichains of P.

Let $(P, \leq)$ be a finite partially ordered set. For simplicity, we just write P when $\leq$ is understood from the context. Let $v_1, \ldots, v_r$ be the elements of P. We define an ideal J_P in $\mathbb{C}[x_1, \ldots, x_r]$:

$$J_P = \langle x_i - x_i \prod_{v_j \leq v_i} x_j, \text{ for all } v_i \in P \rangle.$$

The sets $A(P)$ and $V(J_P)$ are very closely related:

Theorem (2.3.1)**.** *Let P be a finite partially ordered set. Then J_P is a radical zero-dimensional ideal. That is, it has a finite number of (simple) zeros. Furthermore,*

$$|V(J_P)| = |A(P)|,$$

where $|\cdot|$ denotes cardinality.

We can think of a finite partially ordered set P as a directed acyclic transitive graph $\overrightarrow{G_P}$. The undirected graph underlying $\overrightarrow{G_P}$, minus the loops $a \to a$, is called the *comparability graph* of P (G_P). In Section 2.3, we prove that the initial ideal $LT_<(J_P)$ is precisely the modified edge ideal of G_P.

The ideal J_P is a complete intersection, it is radical and it is zero-dimensional. This is probably the best kind of ideal, from an algebraic point of view. One of our main results is that even in this optimal algebraic setting, computing a Hilbert Series is a difficult task.

Theorem (2.3.8)**.** *No algorithm can compute a Hilbert Series in polynomial time when applied to initial ideals of radical zero-dimensional complete intersections, unless $\#P = P$.*

The other main result of this chapter concerns the computation of the independence polynomial of Cohen-Macaulay graphs. A graph G is Cohen-Macaulay if and only if the quotient $\mathbb{C}[\mathbf{x}]/I_G$ is a Cohen-Macaulay ring. A Cohen-Macaulay ring is a particular type of commutative ring, possessing some of the algebraic-geometric properties of a nonsingular variety, such as local equidimensionality. In particular, complete intersection rings and quotients of the polynomial ring by a zero dimensional ideal are Cohen-Macaulay. Cohen-Macaulay rings have good homological (and duality) properties. By exploiting the link between bipartite Cohen-Macaulay graphs and finite partially ordered sets, we are able to derive the following result.

Theorem (2.4.2)**.** *There can be no polynomial algorithm to compute the independence polynomial of bipartite Cohen-Macaulay graphs unless* $\#P = P$*.*

The rest of this Chapter is organized as follows. In Section 2.2, we analyze the standard algorithm for computing a Hilbert Series. We present it in a more general setting, and study the meaning of each of its steps when applied to a modified edge ideal. In Section 2.3, we turn our attention to finite partially ordered sets and their antichains, in order to derive our first main result. In Section 2.4, we present some properties of bipartite Cohen-Macaulay graphs and prove our second main result. We close with some experimental observations in Section 2.5.

2.2 Counting independent sets via the computation of Hilbert Series

Let M be a positively graded finitely generated $\mathbb{C}[\mathbf{x}]$-module (e.g. the quotient $\mathbb{C}[\mathbf{x}]/I_G$ for some graph G). We can write

$$M = \bigoplus_{0 \leq i} M_i,$$

where M_i is the subspace of M of degree i. The Hilbert Function (HF_M) of M maps i onto $\dim_{\mathbb{C}}(M_i)$. The Hilbert Series (HS_M) of M is the generating function

$$HS_M(z) = \sum_{0 \leq i} HF_M(i)\, z^i. \tag{2.2.1}$$

If $M = \mathbb{C}[\mathbf{x}]/I$ for some monomial ideal I, then $HF_M(i)$ is the number of monomials of degree i which *are not* in I. If we take $I = I_G$ for some graph G, as we saw in the previous section, $HF_M(i)$ is then the number of independent sets of size i in G. In this case, the Hilbert Series of $\mathbb{C}[\mathbf{x}]/I_G$ is called the independence polynomial of G.

We can see in [39, Theorem 5.2.20] that in the case of the modified edge ideal I_G, the Hilbert Series of $M = \mathbb{C}[x_1, \ldots, x_n]/I_G$ has the form

$$HS_M = \frac{HN_M(z)}{(1-z)^n}, \tag{2.2.2}$$

where $HN_M(z)$ is called the *Hilbert Numerator*.

The problem of computing a Hilbert Series is NP-Complete ([2]). There is a standard algorithm (first proposed in [46]) for computing the Hilbert

Series of a quotient $\mathbb{C}[\mathbf{x}]/I$, where I is a homogeneous ideal in $\mathbb{C}[\mathbf{x}]$. Several computer algebra systems (CoCoA [12], Singular [25], Macaulay2 [24]) implement it in subtly different ways. There are some classes of ideals for which this algorithm finishes in time polynomial in the input, e.g. Borel ([2]) and Borel-type ideals ([31]). We refer to [39, Ch. 5] for a general reference on Hilbert Series.

The algorithm mentioned above hinges on the following property. If we have a homogeneous exact sequence of finitely generated graded $\mathbb{C}[\mathbf{x}]$-modules

$$0 \longrightarrow M' \longrightarrow M \longrightarrow M'' \longrightarrow 0, \tag{2.2.3}$$

then

$$HS_M(z) = HS_{M'}(z) + HS_{M''}(z). \tag{2.2.4}$$

We can use this additivity property for computations. Let M be a finitely generated graded $\mathbb{C}[\mathbf{x}]$-module and $f \neq 0$ a homogeneous polynomial of degree d. We have the following *multiplication sequence*

$$0 \longrightarrow [M/(0 :_M (f))](-d) \xrightarrow{\varphi} M \longrightarrow M/fM \longrightarrow 0, \tag{2.2.5}$$

where φ is induced by multiplication by f. Here, $(0 :_M (f)) = \{g \in M, \text{such that } gf = 0\}$, and $(-d)$ induces a degree shift, so that φ is a homogeneous map of degree 0. Rewriting equation (2.2.4) we obtain

$$HS_M(z) = HS_{M/fM}(z) + z^d\, HS_{(0:_M(f))}. \tag{2.2.6}$$

The polynomial f above is called a *pivot*.

Actually, only the Hilbert Numerator is computed by the algorithm, since the Series is obtained by dividing the Numerator by $(1-z)^n$. We reproduce the algorithm for computing the Hilbert Numerator of monomial ideal (see [39, Theorem 5.3.7]).

Algorithm 2.2.1. Algorithm to compute the Hilbert Numerator of a monomial ideal I (called HN).

Input: A set of minimal monomial generators for the ideal I.
Output: The Hilbert Numerator of $\mathbb{C}[\mathbf{x}]/I$.

```
if the minimal generators of $I$ are pairwise coprime then
    return $\prod_{i=1}^{s}(1 - z^{d_i})$, where $d_i$ is the degree of the $i$-th generator of $I$.
else
    Choose a monomial $p$ as pivot.
    $f_1 \leftarrow$ HN$(I : p)$.
```

```
6:     $f_2 \leftarrow \texttt{HN}(I+p)$.
7:     return  $z^{\deg(p)} f_1(z) + f_2(z)$.
8: end if
```

The choice of pivot must satisfy one condition. Namely,

$$\sum \deg(I:p) < \sum \deg(I) \quad \text{and} \quad \sum \deg(I+p) < \sum \deg(I). \qquad (2.2.7)$$

Here, $\sum \deg(I)$ denotes the sum of the degrees of all the minimal monomial generators of I. Intuitively, this condition says that the recursive calls are made on "smaller" ideals, and shows that the algorithm terminates.

The program CoCoA implements this algorithm, and uses a certain strategy for the choice of pivot in step 4. First, it chooses any variable x_i appearing in the most number of generators of I_G. Then it picks two random generators containing that variable. The pivot is the highest power of x_i that divides *both* random generators.

We present a specialized version of Algorithm 2.2.1, suited for the computation of the Hilbert Series of $\mathbb{C}[\mathbf{x}]/I_G$ for any graph G.

Theorem 2.2.2. *Let I_G be the modified edge ideal of a graph G. The general algorithm for computing the Hilbert Series of $\mathbb{C}[\mathbf{x}]/I_G$ has the specialized version presented in Algorithm 2.2.3.*

This algorithm has an obvious graphical interpretation. The choice of step 4 corresponds to choosing a node v of the graph. The recursive calls of step 7 correspond to counting the independent sets of G that contain v (HS_{Colon}) and those that do not contain v (HS_{Plus}).

Algorithm 2.2.3. Specialized algorithm to compute the *HS* of $\mathbb{C}[\mathrm{x}]/I_G$.

Input: The list L of minimal monomial generators of I_G described in (2.1.3).
Output: The Hilbert Series of $\mathbb{C}[\mathbf{x}]/I_G$.

```
1: if $L$ consists only of variables and squares of variables then
2:     return  $(1+z)^k$, where $k$ is the number of variables which appear
       squared in $L$.
3: else
4:     Choose a variable $x_i$ that appears squared in $L$.
5:     Colon $\leftarrow$ a minimal set of monomial generators of $(\langle L \rangle : x_i)$.
6:     Plus $\leftarrow$ a minimal set of monomial generators of $\langle L, x_i \rangle$.
7:     return  $z\mathrm{HS}_{\mathrm{Colon}}(z) + \mathrm{HS}_{\mathrm{Plus}}(z)$
8: end if
```

Proof. Algorithm 2.2.3 differs from Algorithm 2.2.1 in two key steps. In step 1, the special version does not check coprimality, as is done in Algorithm 2.2.1. The other difference is in step 4: The specialized version chooses a variable, instead of an arbitrary monomial.

We make a claim that helps us understand why this specialized version is correct. In every call to the algorithm, each of the n variables appears in L raised to the first or to the second power. Furthermore, in each call, L contains only the powers just mentioned and the "edge monomials" x_ix_j such that x_i and x_j appear squared in L. This leads to an obvious graphical interpretation: The list L represents the subgraph of G induced by those variables that appear squared in L.

We prove the correctness of the algorithm by showing that the choice of a pivot in Algorithm 2.2.1 must always yield a variable when applied to a modified edge ideal, and that the previous claim is true.

When the algorithm is originally invoked, every variable appears squared in L. Besides the squares of variables, L contains the "edge monomials" x_ix_j for every edge (i, j) of G. This proves that the claim above holds in the first call.

Assuming that the elements of L have the structure we claim, let us show that the any choice of pivot yields a variable. Suppose that we employ any conceivable strategy for the choice of pivot, always subject to condition (2.2.7). The pivot p cannot be a multiple of any monomial in L. If it is, then Plus $= L$, and the decreasing total degree condition (2.2.7) is not satisfied. The pivot p must then be a product of variables that appear squared in L, but it must not be divisible by any "edge monomial." Suppose that the pivot is the product of at least two variables. That is, $x_ix_j \,|\, p$, where x_i^2 and x_j^2 are in L, and x_ix_j is not in L. Then Plus violates the decreasing total degree condition (2.2.7), because it has the same generators that L has, plus p. If $p = 1$, then Colon $= L$, and this violates the decreasing total degree condition. The only valid choice is then $p = x_i$, for some x_i that appears squared in L.

Once we know that the pivot is always a variable, we can show that the claim above holds for Plus and for Colon. In doing so, we also explain the second part of the theorem.

The list of minimal monomial generators for Plus must contain all the variables that were raised to the first power in L. Furthermore, it must also contain the pivot x_i. The square of x_i is not in Plus, because Plus is minimal, and the "edge monomials" that contained x_i, are not present in Plus. The rest of the generators in L are unaffected. Therefore, we have that every variable appears in Plus either squared or raised to the first power, as we wanted to show. Plus corresponds to the graph obtained by removing the node that corresponds to x_i and all the edges incident with it.

The analysis of Colon is somewhat similar. To obtain a minimal set of monomial generators, we just cross out the pivot x_i from every generator in L that contains it, and then eliminate multiples. If we had an "edge monomial" $x_i x_j$, then x_j is in Colon. Therefore, the square of x_j is no longer a generator, and all the "edge monomials" containing x_j are also missing from Colon. Again, every variable appears either squared or raised to the first variable. In this case, we remove the node corresponding to x_i, all its adjacent nodes and all the edges incident with x_i or with any node adjacent to x_i.

Let v be the node of G associated with the pivot x_i. The combination step of the algorithm reflects the meaning of Colon and Plus: The independent sets of G are those of Plus (i.e. those *do not* that contain v) and those of Colon (i.e. those that contain v).

The algorithm terminates when there are no more "edge monomials". Since all the generators are variables, or squares of variables, then they are pairwise coprime and satisfy the stopping criterion of Algorithm 2.2.1.

A note is in order about the value returned in the base case. Algorithm 2.2.1 returns

$$\prod_{i=1}^{n}(1 - z^{d_i}), \tag{2.2.8}$$

where d_i is the degree of the i-th generator. Since in the specialized case the generators are of the form x_i or x_i^2, expression (2.2.8) has the form

$$(1 - z)^n (1 + z)^k, \tag{2.2.9}$$

where k is the number of variables that appear squared in L. According to the formula (2.2.2), the value return by Algorithm 2.2.3 is the Hilbert Series of $\mathbb{C}[x_1, \ldots, x_n]/I_G$.

All these observations show that the graphical interpretation is accurate and that the specialized version is indeed correct.

□

The execution time of this algorithm is determined by the choice of variable in step 4, and the CoCoA strategy appears to be a good heuristic. We show some experimental observations in Section 2.5.

2.3 Partially ordered sets and Gröbner Bases

In this section, we study a family of zero-dimensional radical complete intersection polynomial ideals first proposed in [9].

A *partially ordered set* is a set P, together with a relation $\leq$ satisfying

- $a \leq a$, for all $a \in P$.
- $a \leq b$ and $b \leq a$ implies $a = b$, for all a and b in P.
- $a \leq b$ and $b \leq c$ implies $a \leq c$ for all a, b and c in P.

We say that two elements a and b of P are *comparable* if $a \leq b$ or if $b \leq a$. Otherwise, we say they are *incomparable*. A subset A of P is called an *antichain* if all the elements of A are pairwise incomparable in P. We write $A(P)$ for the set of antichains of P.

Let $(P, \leq)$ be a finite partially ordered set. We recall the definition of the ideal $J_P \subset \mathbb{C}[x_1, \ldots, x_n]$:

$$J_P = \langle x_i - x_i \prod_{v_j \leq v_i} x_j, \text{ for all } v_i \in P \rangle. \tag{2.3.1}$$

We mentioned the following result in the Introduction of this chapter:

Theorem 2.3.1 ([9]). *Let P be a finite partially ordered set. Then J_P is a radical zero-dimensional ideal. That is, it has a finite number of (simple) zeros. Furthermore,*

$$|V(J_P)| = |A(P)|, \tag{2.3.2}$$

where $|\cdot|$ denotes cardinality.

We take the above theorem one step further, and give an explicit bijection between $A(P)$ and $V(J_P)$. We first need some results about the structure of $V(J_P)$.

Lemma 2.3.2. *Let P be a finite partially ordered set. Then the elements of $V(J_P)$ are strings of 0's and 1's.*

Proof. Suppose that an element $v_i \in P$ is minimal. Then $p_i = x_i - x_i^2 \in J_P$, hence x_i is 0 or 1. Now, take any x_i, and assume that for every $v_j \leq v_i$ we have shown that x_j is 0 or 1. Then if all the x_j are 1, x_i must also be 1. If any x_j is 0, then x_i must be 0 too. □

Similarly, we have the following remark.

Remark 2.3.3. Let P be a finite partially ordered set, and let $\mathbf{x}$ be an element of $V(J_P)$. If $x_i = 1$, then $x_j = 1$ for all j such that $v_j \leq v_i$, since the polynomial $x_i - x_i \prod_{v_j \leq v_i} x_j$ must be zero when evaluated at $\mathbf{x}$.

We can now present the bijection between $A(P)$ and $V(J_P)$.

Theorem 2.3.4. *Let P be a finite partially ordered set. We define $f : V(J_P) \to A(P)$ by*

$$(x_1, \ldots, x_r) \mapsto \{v_i \in P, \text{ such that } x_i = 1 \text{ and } x_j = 0 \text{ for all } v_j > v_i\}.$$

We define $g : A(P) \to V(J_P)$ by

$$A \mapsto (x_1, \ldots, x_r), \text{ where } x_i = 1 \text{ if and only if } \exists\, v_j \in A \text{ such that } v_i \leq v_j.$$

We then have

$$f \circ g = g \circ f = id. \tag{2.3.3}$$

Proof. Let $\mathbf{x} = (x_1, \ldots, x_r)$ be a point in $V(J_P)$. Let $A = f(\mathbf{x})$ and $\mathbf{x}' = (x'_1, \ldots, x'_r) = g(A)$. We want to show that $\mathbf{x} = \mathbf{x}'$. We know that $x'_i = 1$ if and only if $\exists\, v_j \in A$ such that $v_i \leq v_j$. By definition, v_j is in A if and only if $x_j = 1$ and $x_k = 0$ for all $v_k > v_j$. By Lemma 2.3.3, this happens if and only if $x_i = 1$. The other equality is proved similarly. □

We are going to relate the Hilbert Series algorithm of the previous section with the number of antichains of a finite partially ordered set P, but we need to state a few results concerning J_P before doing so.

Proposition 2.3.5. *The universal, reduced Gröbner Basis of J_P is the set Gb_P of polynomials*

$$\begin{aligned} gb_i &= x_i^2 - x_i \qquad \forall\, v_i \in P, \\ gb_{(j,i)} &= x_i x_j - x_i \qquad \forall\, v_j \leq v_i. \end{aligned}$$

Proof. We show that the ideal generated by Gb_P is radical, and that its variety coincides with that of J_P. By Hilbert's Nullstellensatz, this is enough to conclude that the ideals are equal. We then prove that Gb_P is in fact a Gröbner basis with the stated properties.

Lemma 2.3.2 shows that the elements of $V(J_P)$ are strings of 0's and 1's. The polynomials $x_i^2 - x_i$ are in Gb, and therefore the elements of $V(Gb_P)$ are also strings of 0's and 1's. Let $\mathbf{x} = (x_i)_{v_i \in P}$ be a string of 0's and 1's. $\mathbf{x} \in V(J_P)$ if and only if $\forall\, v_i \in P$, $(x_i = 0 \Leftrightarrow (\exists\, v_j \leq v_i \text{ such that } x_j = 0))$. But this is equivalent to $\mathbf{x} \in V(Gb_P)$.

The ideal J_P is radical (see [9]). Gb_P is zero-dimensional, and contains a reduced univariate polynomial in each variable (gb_i). Therefore it is also radical. This concludes the proof of equality.

We now prove that Gb_P is a Gröbner Basis. Given any two polynomials in the set Gb_P, we show that their S-polynomial is divisible by Gb_P. If we let $p = x_i x_j - x_i$ and $q = x_k x_\ell - x_k$ be two polynomials in Gb_P, all possible combinations of the indices i, j, k and ℓ boil down to the following non-trivial possibilities for (p, q) (with i, j, k, ℓ all different):

1. $(x_i^2 - x_i, x_k x_i - x_k)$ or $(x_i x_j - x_i, x_k x_j - x_k) \Rightarrow S(p,q) = 0$.
2. $(x_i^2 - x_i, x_i x_\ell - x_i) \Rightarrow S(p,q) = gb_i - gb_{(\ell,i)}$.
3. $(x_i x_j - x_i, x_i x_\ell - x_i) \Rightarrow S(p,q) = gb_{(j,i)} - gb_{(\ell,i)}$.
4. $(x_i x_j - x_i, x_k x_i - x_k) \Rightarrow S(p,q) = gb_{(j,k)} - gb_{(i,k)}$.
5. $(x_i x_j - x_i, x_j x_\ell - x_j) \Rightarrow S(p,q) = gb_{(\ell,i)} - gb_{(j,i)}$.
6. $(x_i^2 - x_i, x_k^2 - x_k)$ or $(x_i^2 - x_i, x_k x_\ell - x_k)$ or $(x_i x_j - x_i, x_k x_\ell - x_k)$. In all three cases, since the leading monomials of p and q are coprime, $S(p,q)$ is divisible by (p,q).
7. $(x_i x_j - x_i, x_j x_i - x_j)$. This can only hold if $v_i \leq v_j$ and $v_j \leq v_i$, that is, $v_i = v_j$.

In cases 4 and 5 above, we know that $gb_{(j,k)}$ and $gb_{(\ell,i)}$, respectively, are in Gb_P, because a partial order relation is *transitive*. Therefore, Gb_P is a Gröbner Basis.

All the polynomials in Gb_P are of the form $x_i x_\alpha - x_i$. The first term is greater than the second one for any monomial order, and thus Gb_P is universal.

Finally, none of the polynomials are redundant, all of the leading coefficients are one, and the "other" monomial in each polynomial of Gb_P has degree 1, so it cannot be divisible by any leading monomial of Gb_P. Therefore, Gb_P is also a reduced Gröbner Basis. □

We can count the antichains of P by studying J_P. We have seen that $|A(P)| = |V(J_P)|$. The following well-known result is helpful.

Theorem 2.3.6 ([14, Theorem 2.2.10]). *Let I be a radical zero-dimensional ideal (e.g. J_P). Then*

$$|V(I)| = \dim_{\mathbb{C}}(\mathbb{C}[\mathbf{x}]/I).$$

The Hilbert Series algorithm could help us to compute $\dim_{\mathbb{C}}(\mathbb{C}[\mathbf{x}]/J_P)$, but it requires that the ideal J_P be homogeneous, which it clearly is not. Fortunately, we can use another result.

Theorem 2.3.7 ([13, Chapter 5, Section 3]). *Let I be a zero-dimensional polynomial ideal, and let $<$ be any monomial ordering. Then*

$$\dim_{\mathbb{C}}(\mathbb{C}[\mathbf{x}]/I) = \dim_{\mathbb{C}}(\mathbb{C}[\mathbf{x}]/LT_<(I)).$$

We know, by the definition of a Gröbner Basis, that

$$LT_<(J_P) = \langle LT_<(g), g \in Gb_P \rangle.$$

Note that $\langle LT_<(g), g \in Gb_P \rangle$ has the same structure of the ideals I_G of the previous section. Actually, it is precisely I_{G_P}, where G_P is the *comparability graph* of P. This graph has one node per element of P, and two nodes are adjacent if and only if their corresponding elements are comparable. Notice that an antichain of P is exactly an independent set of G_P. As a consequence,

$$|A(P)| = \text{number of independent sets of } G_P = \dim_{\mathbb{C}}(\mathbb{C}[\mathbf{x}]/LT_<(J_P)).$$

Theorem 2.3.8. *No algorithm can compute a Hilbert Series in polynomial time when applied to initial ideals of radical zero-dimensional complete intersections, unless $\#P = P$.*

Proof. If an algorithm allowed us to compute the Hilbert Series of the initial ideal of any complete intersection, then we could apply it to the ideal generated by the leading monomials of Gb_P, for any finite partially ordered set P. But then we would be able to count the antichains of P in polynomial time, just evaluating the Hilbert Series at 1. □

We now study another ideal associated with a finite partially ordered set. An alternate way of dealing with a partially ordered set P is to look at the *cover* relation. Given a and b in P, we say that $a \prec b$ (read "b covers a") if and only if $a < b$ and there is no $c \in P$ such that $a < c < b$. Using this relation we define a new ideal

$$J'_P = \langle x_i - x_i \prod_{v_j \preceq v_i} x_j, \text{ for all } v_i \in P \rangle. \tag{2.3.4}$$

This ideal encodes the same information as J_P, as shown by the following

Lemma 2.3.9. *Let P be a finite partially ordered set. Then*

$$J_P = J'_P. \tag{2.3.5}$$

Proof. It is not hard to see that the varieties of J_P and J'_P coincide. They essentially encode the same partially ordered set. We show that J'_P is radical. Since we already know that J_P is radical, this proves the equality.

We prove radicality by showing that $x_i - x_i^2$ is in J'_P for all $v_i \in P$. We know this to be true for the minimal elements of P, by the very definition of J'_P. Suppose we have a non-minimal element v_i in P, and let $v_{j_1}, \ldots, v_{j_r}$ be the elements such that $v_{j_k} \prec v_i$, and assume that $x_{j_k} - x_{j_k}^2$ is in J'_P for all k. First, we observe that

$$(x_{j_l} - 1)(x_i - x_i^2 \prod_{k=1}^{r} x_{j_k}) - x_i^2 \prod_{\substack{k=1 \\ k\neq l}}^{r} x_{j_k}(x_{j_l} - x_{j_l}^2) = x_i x_{j_l} - x_i$$

is in J'_P for all l. Now, consider the following step.

$$(x_i - x_i^2 \prod_{k=1}^{r} x_{j_k}) - x_i \prod_{k=1}^{r-1}(x_i - x_i x_{j_k}) = x_i - \prod_{k=1}^{r-1} x_{j_k}.$$

Since $(x_i - x_i^2 \prod_{k=1}^{r} x_{j_k})$ and $(x_i - x_i x_{j_k})$ are in J'_P, we have that $x_i - \prod_{k=1}^{r-1} x_{j_k}$ is also in J'_P. If we apply this procedure repeteadly, we eliminate variables from the product, and eventually find that $x_i - x_i^2$ is in J'_P. Since we have shown that J'_P has a square-free univariate polynomial in each variable, it is a radical ideal. □

2.4 Independent sets in bipartite Cohen-Macaulay graphs

We are now going to refine Theorem 2.3.8. Let G be a graph, and I'_G its edge ideal. We say that G is a Cohen-Macaulay graph if $\mathbb{C}[\mathbf{x}]/I'_G$ is a Cohen-Macaulay ring. (The quotient $\mathbb{C}[\mathbf{x}]/I_G$, is always Cohen-Macaulay, because I_G is zero-dimensional.)

Not every graph is Cohen-Macaulay, of course. For example, the path of length three (see Figure 2.1) has the edge ideal $J_{P_3} = \langle x_1x_2, x_2x_3\rangle$, defined in $\mathbb{C}[x_1, x_2, x_3]$. The quotient $\mathbb{C}[x_1, x_2, x_3]/J_{P_3}$ is not Cohen-Macaulay. It is not even equidimensional, since the zero set of J_{P_3} consists of the plane $x_2 = 0$, together with the line $x_1 = x_3 = 0$.

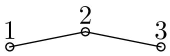


Figure 2.1: The path of length three P_3

One particularly interesting subfamily of Cohen-Macaulay graphs are *bipartite Cohen-Macaulay graphs.* These are characterized by the following result.

Theorem 2.4.1 ([34]). *Let $G = (V_1 \sqcup V_2, E)$ be a bipartite graph. Then G is a Cohen-Macaulay graph if and only if $|V_1| = |V_2|$, and the vertices $V_1 = \{x_1, \ldots, x_n\}$ and $V_2 = \{y_1, \ldots, y_n\}$ can be labeled in such a way that*

1. *$(x_i, y_i) \in E$ for all $i = 1, \ldots, n$;*
2. *if $(x_i, y_j) \in E$, then $i \leq j$;*
3. *if (x_i, y_j) and (x_j, y_k) are edges, then (x_i, y_k) is also an edge.*

There are two ways of seeing a bipartite Cohen-Macaulay graph G as a partially ordered set. One way is to view G as a poset. Suppose that the nodes of G are split into V_1 and V_2. Then we set $x \leq y$ if and only if $x = y$ or $x \in V_1$, $y \in V_2$ and (x, y) is an edge of G. That is, we choose one of the parts as the "upper" one.

The other way involves a different construction. Let $G = (V_1 \sqcup V_2, E)$ be a bipartite Cohen-Macaulay graph. We define a partially ordered set P_G as follows. The elements of P_G are those of V_1. Given x_i and x_j, we set $x_i \leq x_j$ if and only if the edge (x_i, y_j) is in E. From the transitivity of bipartite Cohen-Macaulay graphs, we see that P_G is a partially ordered set.

Conversely, let P be a partially ordered set, with elements $x_1, \ldots, x_r$. Consider a linear extension of P. That is, we assume that if $x_i \leq x_j$ then $i \leq j$. We build a bipartite graph $G_P = (V, E)$ as follows. We set $V = V_1 \sqcup V_2$, with $V_1 = \{x_1, \ldots, x_r\}$ and $V_2 = \{y_1, \ldots, y_r\}$. We put the edges (x_i, y_i) in E for all i, and we have the edge (x_i, y_j) if and only if $x_i \leq x_j$ in P. In this case, the transitivity of $\leq$ ensures that G_P is a bipartite Cohen-Macaulay graph.

The bijection we just oulined allows us to prove the following result.

Theorem 2.4.2. *There can be no polynomial algorithm to compute the independence polynomial of bipartite Cohen-Macaulay graphs unless $\#P = P$.*

Proof. Let G and P_G be a bipartite Cohen-Macaulay graph and its associated partially ordered set, respectively. The construction outlined above expands every element of a finite partially ordered set into a segment in a bipartite Cohen-Macaulay graph. Any antichain of size k in the partially ordered set is then automatically translated into 2^k independent sets of size k in the bipartite graph. Let f be the independence polynomial of G, and let g be the "antichain polynomial" of P. Then

$$f(x) = g(2x).$$

The result now follows from[49]. □

2.5 Some experimental observations

In this Section, we present some observations on the performance of the CoCoA strategy for computing Hilbert Numerators.

We used as a working example the partially ordered set given by the power set of $\{1, \ldots, n\}$. The number of independent sets in its comparability graph is called the n-th *Dedekind number* $D(n)$ ([17]).

We implemented Algorithm 2.2.3 and tried 1000 random choices of pivots for $n = 3$. They choices are plotted in Figure 2.2, next to the CoCoA choice. We can see that the heuristic is indeed good, but not optimal in this case.

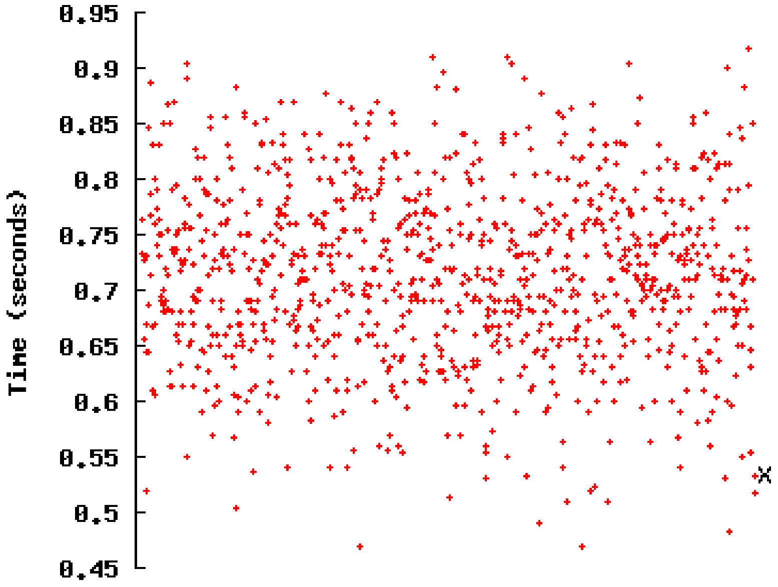


Figure 2.2: Times for $D(3)$. The CoCoA strategy is marked with an X.

We tested the three Computer Algebra Systems mentioned earlier, because we were able to discuss the implementation details with their developers. Singular and Macaulay 2 only managed to compute $D(6)$, and crashed when asked to compute $D(7)$. Only CoCoA was able to compute $D(7)$. Dedekind numbers are only known up to 13, but those computations required many hours of supercomputer time [32]. CoCoA was able to compute $D(7)$ in desktop hardware in under 7 minutes.

There is another popular free Computer Algebra System called SAGE ([58]). However, it delegates Hilbert Series computations to Singular. It does have a native implementation to enumerate the antichains of a poset, but it is implemented using a naive recursive algorithm.

Chapter 3

Maximum Independent Sets in de Bruijn Graphs

3.1 Introduction

We begin the chapter by defining our object of study. Let d and D be two positive integers. Throughout this chapter, we let $[\mathbf{d}]$ stand for the set $\{0, \ldots, d-1\}$. The *de Bruijn* graph $B(d, D)$ is a directed graph with d^D nodes, consisting of all the strings of lenght D, with each symbol taken from the alphabet $[\mathbf{d}]$. There is an edge in $B(d, D)$ from node $x = x_1 \ldots x_D$ to node $y = y_1 \ldots y_D$ if and only if $x_2 \ldots x_D = y_1 \ldots y_{D-1}$. We can think of each as a process: we append the last symbol of y to x to obtain $x_1 \ldots x_D y_D$, and then we discard the first symbol of this string. The edges of $B(d, D)$ can be labeled with the symbol appended to the source node (see $B(2, 2)$ in Figure 3.1).

These graphs were introduced in the paper "A Combinatorial Problem" by N. G. de Bruijn, under the name of *T-nets* [16]. Since then, de Bruijn graphs have been used in a wide range of contexts. For example, in the study of normal numbers [60, 3], network topology [6, 47], quantum computation [51], and sequence assembly [48].

This family of graphs has some interesting properties. For example, they are d-regular, and therefore Eulerian. Also, the graph $B(d, D+1)$ is the line graph of $B(d, D)$. The graph $B(4, 3)$ gained some relevance in the 1950's, in an attempt to explain the genetic code. The structure of DNA molecules had been recently discovered and it was known that DNA was translated into proteins, but the mechanism of this translation was not known. However, proteins were known to be made up of aminoacids. It was postulated (and

Some results from this chapter appear in [8]

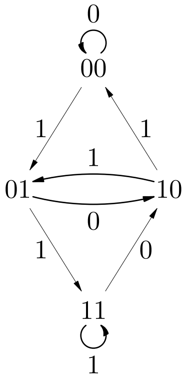


Figure 3.1: $B(2, 2)$.

later confirmed) that short sequences of nucleobases were directly translated into aminoacids by some mechanism. There were only 20 known aminoacids, and there are four nucleobases: adenine, thymine, cytosine and guanine. If the mechanism involved the translation of pairs of bases, then only $4^2 = 16$ aminoacids could be specified. If three bases were used, there were $4^3 = 64$ possible codewords. It was posited that "the wisest code" would be one in which codewords did not overlap. Otherwise, it was said, "genetic chaos would ensue." The maximum size of such a code, using three bases per codeword, is precisely 20. Interestingly, any possible comma-free code with words of length 3 and d available words is a loopless maximum independent set of $B(d, 3)$. We see later in this chapter that there are 408 such sets.

There is a family of sequences closely related to these graphs. A *de Bruijn Sequence* of order D using d symbols is a circular sequence *Seq* of symbols in $[\mathbf{d}]$, such that every possible sequence of length D consisting of symbols in $\{0, \ldots, d-1\}$ appears exactly once in *Seq*. All such sequences are obtained by reading the labels of the edges of the Eulerian circuits of $B(d, D-1)$.

In this chapter, we give a complete characterization of maximum independent sets of the subfamily $B(d, 3)$. The maximum independent sets of de Bruijn graphs have been previously studied ([36] and [43], for example). In particular, the size of a maximum independent set is determined for different values of d and D in [43].

The graph $B(d, D)$ contains d nodes of the form $x_1 = \cdots = x_D = i$, for $i \in [\mathbf{d}]$. We refer to such nodes as *loops*. Every loop is adjacent to itself, and hence cannot be present in any independent set. On account of this, we call any maximum independent set $B(d, D)$ a *loop-less maximum independent set*. We also consider the graph $B(d, D)$ with the d self-edges removed. We reserve the term *maximum independent set* for those of this modified version of $B(d, D)$. See Figure 3.2 for a maximum independent set of $B(3, 3)$.

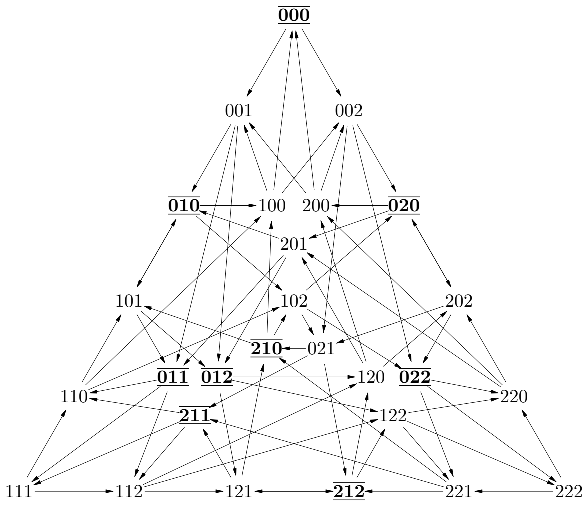


Figure 3.2: The nodes of a maximum independent set of $B(3,3)$ are highlighted. The self-edges of 000, 111, and 222 have been removed.

We write $\alpha(d,D)$ for the size of a maximum independent set with loops and $\alpha^*(d,D)$ for the size of a loop-less maximum independent set. From [43], one can show that for $d \geq 4$, $\alpha(d,2) = \alpha^*(d,2) = \lfloor d^2/4 \rfloor$ and the number of maximum independent sets of $B(d,2)$ is $\binom{d}{d/2}$ if d is even and $2\binom{d}{(d-1)/2}$ if d is odd.

In this chapter, we study the maximum independent sets of the graphs $B(d,3)$. Again, in [43], it was proved that

$$\alpha(d,3) = \frac{d^3-d}{3} + 1 \quad \text{and} \quad \alpha^*(d,3) = \frac{d^3-d}{3}. \tag{3.1.1}$$

The symmetric group $\mathbb{S}_d$ operates naturally on the independent sets of $B(d,D)$. We define four functions which, together with the action of the symmetric group $\mathbb{S}_d$ recursively generate all maximum independent sets in $B(d,3)$. From this, we deduce a recurrence relation for the number of maximum independent sets in $B(d,3)$.

From Theorem 3.4.5 in Section 3.4, our main result, we derive the following:

Theorem. *If we let a_d be the number of maximum independent sets of $B(d,3)$, then a_d has exponential generating function*

$$\sum_{d=1}^{\infty} \frac{a_d t^d}{d!} = \frac{t+t^2}{1-2t-t^2}.$$

We obtained the initial data for this chapter by computing the first few terms of a_d using the free computer algebra system CoCoA ([12]). We have computed all the maximum independent sets for $d \leq 5$ by an exhaustive procedure, and we used those sets in some of our proofs.

The rest of this chapter is organized as follows. In Section 3.2, we define two functions f and f' that take a maximum independent set of $B(d,3)$ and return maximum independent sets of $B(d+1,3)$. We construct another two functions g and g' that take a maximum independent set of $B(d,3)$ and return a maximum independent set of $B(d+2,3)$. We also prove some basic facts about the images of these four functions.

In Section 3.3 we present the interactions between the symmetric group and the functions defined in Section 3.2. We compute the stabilizers of the images of our four functions. We later prove that all four functions map orbits under the action of the symmetric group into orbits under that same action. Furthermore, we show that the images of all four functions extended to orbits are disjoint. These results are used to prove the main theorems of this chapter in Section 3.4.

In Section 3.5 we show the structure of the loop-less maximum independent sets of $B(d,3)$, and show that their number coincides with the number of maximum independent sets of $B(d,3)$ with loops. We end with some open questions about possible generalizations of our results to words of other lengths in Section 3.6.

3.2 Inductive Construction of Maximum Independent Sets

In this section, we present four combinatorial operations that transform a maximum independent set in the de Bruijn graph $B(d,3)$ into a maximum independent set in either $B(d+1,3)$ or $B(d+2,3)$.

We start by defining a function that is key to our proofs.

Definition 3.2.1 (Lichiardopol [43]). We define a function $\theta : B(d,D) \to B(d,D)$ that performs the following rotation on the digits of a node

$$\theta(x_1 \cdots x_D) = x_2 \cdots x_D x_1. \tag{3.2.1}$$

Given a node $\omega = x_1 \cdots x_D$, we define its θ-cycle as

$$\omega \xrightarrow{\theta} \theta(w) \xrightarrow{\theta} \theta^2(\omega) \xrightarrow{\theta} \cdots \xrightarrow{\theta} \theta^{D-1}(\omega) \xrightarrow{\theta} \omega. \tag{3.2.2}$$

The graph $B(d, D)$ can be decomposed into its disjoint θ-cycles. If we consider the graph $B(d, D)$, with D prime, then a total of d θ-cycles have size 1 (the loops), and the remaining $(d^D - d)/D$ θ-cycles have size D. Figure 3.3 shows this decomposition of $B(3, 3)$, with the θ-cycles indicated by darker arrows.

In the case at hand, $D = 3$, each of these disjoint θ-cycles contributes at most one node to any independent set of $B(d, 3)$. From this point onwards, when we speak of "cycles", we mean θ-cycles.

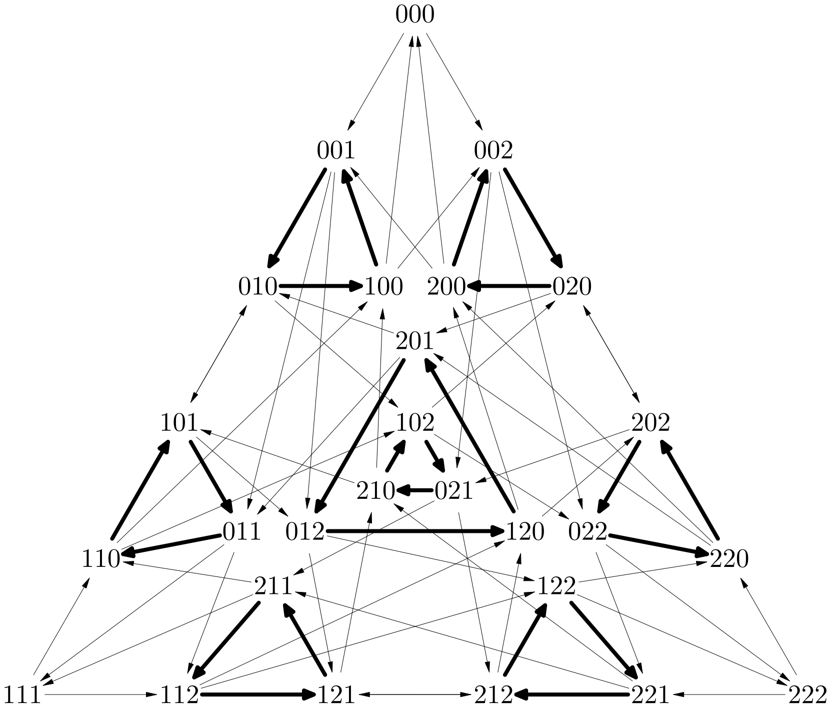


Figure 3.3: $B(3, 3)$ with the edges induced by θ in bold.

We start by defining a concept that greatly shortens our statements.

Definition 3.2.2. Let A be a set of nodes from $B(d, 3)$. Let x and y be two digits in $[\mathbf{d}]$. We say that *y appears between x in A* if the node $xyx \in A$. We define $\mathcal{M}_x(A)$ as the set of digits which do not appear between x in A. We define $m_x(A)$ as the number of digits which do not appear between x in A, i.e. $m_x(A) = |\mathcal{M}_x(A)|$.

The quantities $m_x(A)$ play a crucial role in proving that the functions we later define are surjective.

Proposition 3.2.3. *A maximum independent set S of $B(d,3)$ contains either one or two loops. S always contains one loop a such that $m_a(S) = 0$. If S contains another loop b, then $m_b(S) = 1$ and $\mathcal{M}_b(S) = \{a\}$. In particular, S contains aba, but not bab.*

Proof. Suppose that an independent set S contains two loops a and b. Consider the two cycles $aab \xrightarrow{\theta} aba \xrightarrow{\theta} baa$ and $bba \xrightarrow{\theta} bab \xrightarrow{\theta} abb$. These two cycles can contribute at most the nodes aba and bab, since aab, baa, bba and abb are adjacent to either aaa or bbb. However, aba and bab are adjacent to each other, so only one of them can be present in S. Therefore, for every pair of loops present in an independent set S, one of the $(d^3 - d)/3$ cycles contributes no nodes to S.

If l is the number of loops in S, there are $\binom{l}{2}$ pairs of loops, and thus $\binom{l}{2}$ cycles which cannot contribute any nodes to S. Therefore, $|S| \leq (d^3 - d)/3 + l - \binom{l}{2}$, so for $l > 2$, S cannot be a maximum independent set.

The rest of the proposition follows from the previous observations: S must have one loop, by cardinality. If S is a maximum independent set with only one loop a, then every cycle contributes one node. The cycles $aax \xrightarrow{\theta} axa \xrightarrow{\theta} xaa$ can only contribute axa. If S has two loops a and b, then only one of aba and bab can be present. □

Remark 3.2.4. The previous proof illustrates a tool we use throughout this chapter. When dealing with a maximum independent set S, we often choose a cycle $xyz \xrightarrow{\theta} yzx \xrightarrow{\theta} zxy$ and show that two of its nodes cannot be in S, therefore concluding that the third one is in S. Sometimes, we show that none of the three are in S, thus contradicting the fact that S is maximum.

Convention 3.2.5. From now on, we state things like "Let S be a maximum independent set with loops a and possibly b," when in fact S may have just one loop. In that case, everything said about b should be disregarded.

Definition 3.2.6. If w is a node, we denote by $w[x \rightarrow y]$ the node that results from replacing every occurrence of the digit x by the digit y in w. We write $x \in w$ to mean that x is one of the digits that appear in w.

We denote by a the loop of S such that $m_a(S) = 0$. If S has another loop we denote it b.

We start the definition of the functions that allow us to enlarge maximum independent sets. We first define two functions f and f' that take a maximum independent set of $B(d,3)$ into one of $B(d+1,3)$.

Definition 3.2.7. Let S be a maximum independent set of $B(d, 3)$. We define $f(S) \subset B(d+1, 3)$ to be the union of S with the sets

$$\begin{aligned}
U_1(S) &= \{w[a \to d] \mid w \in S, a \in w, w \neq aaa, w \neq aba\},\\
U_2(S) &= \{axd \mid x \in [\mathbf{d}]\backslash\{a, b\}\},\\
U_3(S) &= \{dxa \mid x \in [\mathbf{d}]\backslash\{a, b\}\},\\
U_4(S) &= \{udv \mid u, v \in \{a, b\}\},\\
U_5(S) &= \{udd \mid u \in \{a, b\}\}.
\end{aligned}$$

Using the example from Figure 3.2, $S = \{000, 010, 020, 011, 012, 210, 022, 211, 212\}$, and

$$\begin{aligned}
U_1(S) &= \{313, 323, 311, 312, 213, 322\},\\
U_2(S) &= \{013, 023\},\\
U_3(S) &= \{310, 320\},\\
U_4(S) &= \{030\},\\
U_5(S) &= \{033\}.
\end{aligned}$$

Remark 3.2.8. The loops of S are the same as the loops of $f(S)$. Notice that $m_d(f(S)) = l + 1$, where l is the number of loops of S.

Proposition 3.2.9. *If S is a maximum independent set of $B(d, 3)$, then $f(S)$ is a maximum independent set of $B(d+1, 3)$.*

Proof. $f(S)$ is made up of six disjoint sets. Let l be the number of loops of S. We have

$$\begin{aligned}
|S| &= \frac{d^3 - d}{3} + 1,\\
|U_1(S)| &= (d-1)(d-1) - (l-1) + (d-l) = d^2 - d + 2 - 2l\\
|U_2(S)| &= |U_3(S)| = (d-l),\\
|U_4(S)| &= l^2,\\
|U_5(S)| &= l.
\end{aligned}$$

Only the cardinality of $U_1(S)$ requires explanation. Notice that $B(d, 3)$ has $(d-1)(d-1)$ cycles which contain a once, with the exception of $abb \xrightarrow{\theta} bba \xrightarrow{\theta} bab$ in the case that $l = 2$. Each of these contributes one element to S and thus to $U_1(S)$. In addition, S contains one element from each of the $d - l$ cycles of the form $aax \xrightarrow{\theta} axa \xrightarrow{\theta} xaa$, where x is not a loop.

We add the six quantities to obtain

$$|f(S)| = \frac{(d+1)^3 - (d+1)}{3} + (l-1)(l-2) + 1.$$

Since l is either 1 or 2, $f(S)$ has the size of a maximum independent set.

We still have to prove that $f(S)$ is an independent set. This amounts to noticing that there are no arrows between the six sets defining $f(S)$. The only remark to bear in mind is that axa is in S for all x, and that bxb is also in S, except for $x = a$. We leave the details to the reader. □

We define another function very similar to f.

Definition 3.2.10. Let S be a maximum independent set of $B(d,3)$. We define $f'(S) \subset B(d+1,3)$ as the union of S, the sets $U_1(S)$, $U_2(S)$, $U_3(S)$, $U_4(S)$ from Definition 3.2.7, and

$$U'_5(S) = \{ddu \mid u \in \{a,b\}\}.$$

Using the same example from Figure 3.2 before, $U'_5(S) = \{330\}$.

Proposition 3.2.11. *If S is a maximum independent set of $B(d,3)$, then $f'(S)$ is a maximum independent set of $B(d+1,3)$.*

Proof. This proposition is proved analogously to Proposition 3.2.9. □

We now present two functions g and g' that take a maximum independent set of $B(d,3)$ into one of $B(d+2,3)$. The definitions are similar to those of the fs.

Definition 3.2.12. Let S be a maximum independent set of $B(d,3)$. We define $g(S) \subset B(d+2,3)$ to be the union of S with the sets

$$\begin{aligned}
V_1(S) &= \{w[a \to y] \mid y \in \{d, d+1\}, w \in S, a \in w, w \neq aaa, w \neq aba\},\\
V_2(S) &= \{axy \mid x \in [\mathbf{d}] \backslash \{a,b\}, y \in \{d, d+1\}\},\\
V_3(S) &= \{yxa \mid x \in [\mathbf{d}] \backslash \{a,b\}, y \in \{d, d+1\}\},\\
V_4(S) &= \{yxz \mid y, z \in \{d, d+1\}, y \neq z, x \in [\mathbf{d}] \backslash \{a,b\}\},\\
V_5(S) &= \{uyv \mid u, v \in \{a,b\}, y \in \{d, d+1\}\},\\
V_6(S) &= \{uyy \mid u \in \{a,b\}, y \in \{d, d+1\}\},\\
V_7(S) &= \{yzu \mid y, z \in \{d, d+1\}, y \neq z, u \in \{a,b\}\},\\
V_8(S) &= \{d(d+1)(d+1), (d+1)dd\}.
\end{aligned}$$

Using the example from Figure 3.2 once more, we obtain
$S = \{000, 010, 020, 011, 012, 210, 022, 211, 212\}$

$$\begin{aligned}
V_1(S) &= \{313, 414, 323, 424, 311, 411, 312, 412, 213, 214, 322, 422\},\\
V_2(S) &= \{013, 014, 023, 024\},\\
V_3(S) &= \{310, 410, 320, 420\},\\
V_4(S) &= \{314, 324, 413, 423\},\\
V_5(S) &= \{030, 040\},\\
V_6(S) &= \{033, 044\},\\
V_7(S) &= \{340, 430\},\\
V_8(S) &= \{344, 433\}.
\end{aligned}$$

Proposition 3.2.13. *If S is a maximum independent set of $B(d, 3)$, then $g(S)$ is a maximum independent set of $B(d + 2, 3)$.*

Proof. $g(S)$ is made up of nine disjoint sets. For each pair of sets, it is clear that there are no edges between them. We now show that $g(S)$ has the right size. If l is the number of loops of S,

$$\begin{aligned}
|S| &= \frac{d^3 - d}{3} + 1,\\
|V_1(S)| &= 2|U_1(S)| = 2(d^2 - d + 2 - 2l),\\
|V_2(S)| &= 2|U_2(S)| = 2(d - l),\\
|V_3(S)| &= 2|U_3(S)| = 2(d - l),\\
|V_4(S)| &= 2|U_4(S)| = 2l^2,\\
|V_5(S)| &= 2(d - l),\\
|V_6(S)| &= 2|U_5(S)| = 2l,\\
|V_7(S)| &= 2l,\\
|V_8(S)| &= 2.
\end{aligned}$$

The sum of these sizes is

$$|g(S)| = \frac{(d + 2)^3 - (d + 2)}{3} + 2(l - 1)(l - 2) + 1.$$

For $l = 1$ or 2, $g(S)$ is a maximum independent set of $B(d + 2, 3)$. □

We finish our function definitions with the analogous to g of f'.

Definition 3.2.14. Let S be a maximum independent set of $B(d, 3)$. We define $g'(S) \subset B(d + 2, 3)$ to be the union of S, the sets $V_1(S)$, $V_2(S)$, $V_3(S)$,

$V_4(S)$, $V_5(S)$ from Definition 3.2.12, and the sets

$$\begin{aligned}
V_6'(S) &= \{yyu,\ u \in \{a,b\}, y \in \{d, d+1\}\},\\
V_7'(S) &= \{uyz,\ y, z \in \{d, d+1\}, y \neq z, u \in \{a,b\}\},\\
V_8'(S) &= \{(d+1)(d+1)d, dd(d+1)\},
\end{aligned}$$

which are the reverses of $V_6(S)$, $V_7(S)$, and $V_8(S)$ respectively.

For the example of Figure 3.2, we have

$$\begin{aligned}
V_6'(S) &= \{330, 440\},\\
V_7'(S) &= \{034, 043\},\\
V_8'(S) &= \{334, 443\}.
\end{aligned}$$

Proposition 3.2.15. *If S is a maximum independent set of $B(d,3)$, then $g'(S)$ is a maximum independent set of $B(d+2,3)$.*

Proof. This proposition is proved analogously to Proposition 3.2.13. □

This concludes our definitions. In Section 3.6 we present a possible unifying framework for these functions.

3.3 Action of the Symmetric Group

In this section, we study the interaction between $\mathbb{S}_d$ and the four functions we defined in the previous section.

As we mentioned in the Introduction, the symmetric group $\mathbb{S}_d$ acts on the nodes of $B(d,3)$ by $\sigma(xyz) = \sigma(x)\sigma(y)\sigma(z)$. This action preserves the graph structure, and therefore permutes the maximum independent sets. We write $A \sim B$ to mean A and B are two sets in the same orbit under the action of $\mathbb{S}_d$. Notice that the functions f, f', g, and g' are defined in such a way that if $A \sim B$, then $f(A) \sim f(B)$ and so on. That is, all four functions map orbits into orbits.

We prove two propositions that show the effect of applying f, f', g or g' on the stabilizer of its image. We see that f and f' preserve stabilizers, while g and g' double their size by adding a transposition to it.

Proposition 3.3.1. *Let S be a maximum independent set of $B(d,3)$. Let H, H' and H'' be the stabilizers of S, $f(S)$ and $f'(S)$, respectively. Then*

$$H = H' = H'',$$

where we identify H with its image under the inclusion $\mathbb{S}_d \hookrightarrow \mathbb{S}_{d+1}$.

Proof. We know that $H \subseteq H'$, and we must prove the other inclusion.

Let $\sigma \in H'$, and let a and possibly b be the loops of S. The set of loops must be preserved by σ and moreover, by Proposition 3.2.3, σ fixes each loop. We want to show that $\sigma(d) = d$. Suppose that $\sigma(d) = z \neq d$ and then $\sigma(w) = d$, for some $w \neq d$. Since w is not a loop, the node awd then belongs to the set $U_2(S)$ from Definition 3.2.7, and so to $f(S)$. That means that $\sigma(awd) = adz$ must be in $f(S)$. Since it begins with a, and has d in the middle, it could only be in $U_4(S)$. But $z \neq a, b$, and so $adz \notin U_4(S)$. Therefore, $\sigma(d) = d$.

Now, since $\sigma(d) = d$, σ is also an element of $\mathbb{S}_d$. Furthermore, it must be in the stabilizer of S. Otherwise, it should map a node of S into a node having a d. Since this is not possible, $\sigma \in H$.

The proof for H'' is completely analogous. □

Proposition 3.3.2. *Let S be a maximum independent set of $B(d,3)$. Let H, H' and H'' be the stabilizers of S, $g(S)$ and $g'(S)$, respectively. Let τ be the transposition interchanging d and $d+1$. Then*

$$H' = H'' = \langle \tau, H \rangle,$$

where, again, we identify H with its image in $\mathbb{S}_{d+2}$. Notice that τ commutes with every element of H.

Proof. As in the proof of Proposition 3.3.1, we know that $\langle \tau, H \rangle \subseteq H'$. Now, let $\sigma \in H'$. Again, σ must preserve the set of loops in $g(S)$, and by Proposition 3.2.3, σ in fact fixes each loop. We will show that either σ or $\tau\sigma$ fixes d and $d+1$. Let x, y, z and w be such that

$$x \xmapsto{\sigma} d \xmapsto{\sigma} y \qquad \text{and} \qquad z \xmapsto{\sigma} d+1 \xmapsto{\sigma} w.$$

We know that $x, y, z, w \neq a, b$. Suppose that $x \neq d, d+1$. Then we must have $dxa \in V_3(S)$ from Definition 3.2.12. Consider $\sigma(dxa) = yda$. This node has to be in $g(S)$, but it can only be in $V_7(S)$. That means that $y = d+1$. Likewise, considering

$$\sigma((d+1)za) = w(d+1)a,$$

we have $w = d$. So $\sigma(d) = d+1$ and $\sigma(d+1) = d$. This contradicts our assumption about x, and implies that $x = d$ or $x = d+1$. Analogously, $z = d+1$ or $z = d$. That means that σ fixes d and $d+1$ or that it transposes them. Therefore, either σ or $\tau\sigma$ is in H, and so $\sigma \in \langle \tau, H \rangle$.

The other equality follows in much the same way. □

We now show the precise way in which our functions and $\mathbb{S}_d$ interact.

Lemma 3.3.3. *Let S and S' be maximum independent sets of $B(d,3)$. Then $f(S) \not\sim f'(S')$.*

Proof. We proceed by contradiction. Suppose that there is a permutation $\sigma \in \mathbb{S}_{d+1}$ such that

$$f(S) = \sigma f'(S').$$

Let a and possibly b be the loops of S and $a' = \sigma^{-1}(a)$ and $b' = \sigma^{-1}(b)$ be the corresponding loops in S'. Let $x \neq a', b'$, $y \neq a, b$ be such that

$$\begin{tikzcd} x \arrow[r, mapsto, "\sigma"] & d \arrow[r, mapsto, "\sigma"] & y. \end{tikzcd}$$

Suppose that $y \neq d$. Then the node ayd is in $U_2(S)$, and hence in $f(S)$. Therefore, $\sigma^{-1}(ayd)$ must be in $f'(S')$. But $\sigma^{-1}(ayd) = a'dx$, which cannot be in any of the sets that make up $f'(S')$. This implies that $\sigma(d) = d$. In other words, σ lies in the image of $\mathbb{S}_d$, and so $\sigma f'(S') = f'(\sigma S')$. However, $f(S)$ has at least one element of the form udd, and $f'(\sigma S')$ has none, so $f(S) \not\sim f'(S')$. □

A similar result holds for g and g':

Lemma 3.3.4. *Let S and S' be maximum independent sets of $B(d,3)$. Then $g(S) \not\sim g'(S')$.*

Proof. This proof is similar to that of Lemma 3.3.3, but somewhat more subtle. Suppose that there is $\sigma \in \mathbb{S}_{d+2}$ such that $g(S) = \sigma g'(S')$. Let a and possibly b be the loops of S and $a' = \sigma^{-1}(a)$ and $b' = \sigma^{-1}(b)$ be the corresponding loops of S'. Let $x, z \neq a', b'$, $y, w \neq a, b$ be such that

$$\begin{tikzcd} x \arrow[r, mapsto, "\sigma"] & d \arrow[r, mapsto, "\sigma"] & y & \text{and} & z \arrow[r, mapsto, "\sigma"] & d+1 \arrow[r, mapsto, "\sigma"] & w. \end{tikzcd}$$

Suppose that $y \neq d, d+1$. Then the node ayd is in $V_2(S)$, and therefore in $g(S)$. That means that $\sigma^{-1}(ayd) = a'dx$ must be in $g'(S')$. But such a node does not belong to any of the sets that make up $g'(S')$. This implies that either $\sigma(d) = d$ or $\sigma(d) = d+1$. Analogously, we can prove that $\sigma(d+1) = d+1$ or $\sigma(d+1) = d$.

Therefore, σ transposes d and $d+1$ or leaves them fixed. By Proposition 3.3.2, the transposition $(d, d+1)$ is in the stabilizer of $g'(S')$ and so by possibly multiplying σ on the right by this transposition, we can assume that σ fixes d and $d+1$ and so it lies in $\mathbb{S}_d$. Therefore, $\sigma g'(S') = g'(\sigma S')$, but $g(S)$ has at least one node of the form udd, and $g'(\sigma S')$ has none, so $g(S) \not\sim g'(S')$. □

We establish two invariants that completely characterize maximum independent sets in de Bruijn graphs with $D = 3$. This is useful to prove that our functions f, f', g, and g', together with the action of $\mathbb{S}_d$, allow us to construct all maximum independent sets of $B(d, 3)$. In order to reverse these functions, we need the following lemma:

Lemma 3.3.5. *If S is a (possibly loop-less) maximum independent set of $B(d, 3)$, with loops a and possibly b. Let d' be an integer such that $a, b < d' < d$. Then,*

$$S' = S \cap B(d', 3)$$

is a maximum independent set of $B(d', 3)$ with loops a and possibly b.

Proof. Since $B(d', 3)$ is a subgraph of $B(d, 3)$, S' is clearly an independent set. Furthermore, since S has one element from each cycle except possibly a cycle that only uses the digits a and b, then S' has the same property. Therefore, S' has the cardinality of a maximum independent set. □

We now present one of the keys to the surjectivity of f and f'.

Proposition 3.3.6. *Let S be a maximum independent set of $B(d, 3)$ with l loops, where d is at least 3. There exists a digit x such that $m_x(S) = l + 1$* if and only if *there exist $\sigma \in \mathbb{S}_d$ and S' a maximum independent set of $B(d-1, 3)$ such that $S = \sigma f(S')$ or $S = \sigma f'(S')$.*

Proof. One implication follows from the definitions of f and f', taking $x = \sigma(d - 1)$.

Suppose now that there is such an x. We know it is not a loop by Proposition 3.2.3. We define the transposition $\sigma = (d - 1, x)$ and the set $S' = \sigma S \cap B(d - 1, 3)$, which is a maximum independent set of $B(d - 1, 3)$ by Proposition 3.3.5.

Let a and possibly b be the loops of S. We know that the node $xax \notin S$. Therefore, either xxa or axx must be in S.

Suppose that $axx \in S$. We are going to show that $S = \sigma f(S')$. To do so, we consider each of the sets that make up $\sigma f(S')$, and show that they are included in S.

The nodes of $\sigma S'$ belong to S, because of the way we defined S'.

Let us consider the nodes of $\sigma U_1(S')$. The nodes of this set are of the form xyx, xyy, yyx, xyz or yzx, for $y, z \neq a, b, x$.

- The nodes of the form xyx are all in S. Otherwise, the hypothesis cannot be satisfied.

- If $xyy \in \sigma U_1(S')$, then $ayy \in S'$. This means that $ayy \in S$, and so yyx cannot be in S. The node yxy cannot be in S either, since xyx is. So, $xyy \in S$. Analogously, if $yyx \in \sigma U_1(S')$, then $yyx \in S$.

- If $xyz \in \sigma U_1(S')$, then $ayz \in S'$ and $ayz \in S$. Since neither zxy (adjacent to xyx) nor yzx (adjacent to ayz) can be in S, xyz must be in S. The same reasoning applies to yzx.

Let us consider the nodes of $\sigma U_2(S')$. These have the form ayx. The nodes yxa (adjacent to xyx) and xay (adjacent to aya) cannot be in S, which implies that $ayx \in S$. The same reasoning shows that $\sigma U_3(S') \subset S$.

Now, take a node from $\sigma U_4(S')$. That is a node of the form uxv, with u, v loops. The nodes xuv (adjacent to uxu) and uvx (adjacent to vxv) cannot be in S. Therefore, $uxv \in S$, and $\sigma U_4(S') \subset S$.

Finally, we know that $axx \in S$. The nodes xbx (adjacent to bxb) and xxb (adjacent to axx) cannot be in S. That implies that $bxx \in S$, which means $\sigma U_5(S') \subset S$.

This proves that $S \supseteq \sigma f(S')$. By cardinality, we conclude that equality holds.

If, instead of $axx \in S$ we have $xxa \in S$, an analogous procedure shows that $S = \sigma f'(S')$. □

The following technical lemma is used in the proof of the next Proposition.

Lemma 3.3.7. *Let S be a (possibly loop-less) maximum independent set of $B(d, 3)$, with $d \geq 3$. If there exist two different digits y and z, which are not loops, such that*

$$m_y(S) = m_z(S) = l + 2,$$

then $yzy \notin S$ and $zyz \notin S$.

Proof. Suppose that $yzy \in S$. Then, by the assumptions on $m_y(S)$, there must be some $v \neq y$ such that $yvy \notin S$. Suppose that $vyy \in S$. The node zyz cannot be in S, and by the assumption on $m_z(S)$, $zvz \in S$. Therefore, the nodes zvy (adjacent to vyy), vyz (adjacent to yzy) and yzv (adjacent to zvz) are not in S. But then the cycle $zvy \xrightarrow{\theta} vyz \xrightarrow{\theta} yzv$ contributes no nodes to S, which contradicts the fact that S is maximum. If we assume that $yyv \in S$, then the cycle $yvz \xrightarrow{\theta} vzy \xrightarrow{\theta} zyv$ cannot contribute any node to S.

In conclusion, our assumption that yzy is in S is inconsistent with S being a maximum independent set. By symmetry, the same holds if we assume $zyz \in S$. □

The following proposition is the equivalent of Proposition 3.3.6 for g and g', and is one of the keys to the surjectivity of those two functions.

Proposition 3.3.8. *Let S be a maximum independent set of $B(d,3)$, with $d \geq 3$. There are two different digits y and z such that*

$$m_y(S) = m_z(S) = l + 2$$

and no *digit x such that $m_x(S) = l + 1$,* if and only if *there exist $\sigma \in \mathbb{S}_d$ and S' a maximum independent set of $B(d-2,3)$ such that*

$$S = \sigma g(S') \quad \text{or} \quad S = \sigma g'(S').$$

Proof. One implication follows from the construction of g and g' taking $y = \sigma(d-1)$ and $z = \sigma(d-2)$.

The proof in the other direction is analogous to the proof of Proposition 3.3.6. We can safely assume that $y = d-1$ and $z = d-2$. By Lemma 3.3.7, either $(d-1)(d-2)(d-2)$ and $(d-2)(d-1)(d-1)$ are in S, or $(d-1)(d-1)(d-2)$ and $(d-2)(d-2)(d-1)$ are in S. In the former case, we find that there is an S' such that $S = \sigma g(S')$. In the latter case, we find that $S = \sigma g'(S')$. □

Corollary 3.3.9. *Let S and S' be maximum independent sets of $B(d-1,3)$ and $B(d-2,3)$, $d \geq 3$. Then for $\mathcal{F} = f, f'$ and $\mathcal{G} = g, g'$, we have*

$$\mathcal{F}(S) \not\sim \mathcal{G}(S').$$

Proof. This result follows from the invariants of $\mathcal{F}(S)$ and $\mathcal{G}(S')$ that are stated in Propositions 3.3.6 and 3.3.8. □

This Corollary, together with Lemmas 3.3.3 and 3.3.4 shows that all four functions give rise to essentially different (i.e. in different $\mathbb{S}_d$-orbits) maximum independent sets.

3.4 Characterization of Maximum Independent Sets

In this section, we show that the functions f, f', g, and g', together with the action of $\mathbb{S}_d$ are sufficient to construct every maximum independent set of $B(d,3)$. For the rest of this section L denotes the set of loops of S, and l denotes $|L|$. In Section 3.5, we take S to be a loop-less maximum independent set, i.e. L is empty.

Lemma 3.4.1. *Let S be a (possibly loop-less) maximum independent set of $B(d,3)$. There cannot be three different digits x, y, and z, with $x, y, z \notin L$, such that*

$$\mathcal{M}_x(S) = \mathcal{M}_y(S) = L \cup \{x, y, z\},$$
$$\mathcal{M}_z(S) = L \cup \{x, z\} \text{ or } L \cup \{x, y, z\}.$$

Proof. Suppose that S is a maximum independent set and x, y, and z satisfy the given condition. Without loss of generality, we can assume that x, y, z, and the loops are less than 5. Then $S' = S \cap B(5,3)$ is a maximum independent set in $B(5,3)$ by Proposition 3.3.5, that satisfies $\mathcal{M}_x(S') = \mathcal{M}_x(S)$, $\mathcal{M}_y(S') = \mathcal{M}_y(S)$, and $\mathcal{M}_z(S') = \mathcal{M}_z(S)$.

We can show that there cannot be such an independent set S'. Suppose we had $S' \subset B(5,3)$ with

$$\mathcal{M}_x(S) = \mathcal{M}_y(S) = L \cup \{x, y, z\},$$
$$\mathcal{M}_z(S) = L \cup \cup\{x, y, z\}.$$

Then either $\{xxy, yyx, xxz, zzx, zzy, yyz\} \subset S'$ or $\{yxx, xyy, zxx, xzz, yzz, zyy\} \subset S'$. In either case, the cycle $xyz \xrightarrow{\theta} yzx \xrightarrow{\theta} zxy$ can contribute no nodes to S', and hence it is not a maximum independent set.

Suppose now that

$$\mathcal{M}_x(S) = \mathcal{M}_y(S) = L \cup \{x, y, z\},$$
$$\mathcal{M}_z(S) = L \cup \{x, z\}.$$

Then either $\{xxy, xxz, yyx, yyz, zzx\} \subset S'$ or $\{yxx, zxx, xyy, zyy, xzz\} \subset S'$. In the first case, the cycle $xyz \xrightarrow{\theta} yzx \xrightarrow{\theta} zxy$ can contribute no nodes to S', and hence it is not a maximum independent set. In the second case, the same observation applies to the cycle $zyx \xrightarrow{\theta} yxz \xrightarrow{\theta} xzy$.

Therefore, there is no such independent set S. □

Lemma 3.4.2. *Let S be a (possibly loop-less) maximum independent set of $B(d,3)$. There cannot be three different digits x, y, and z, none of them loops, such that*

$$m_x(S) = m_y(S) = m_z(S) = l + 2.$$

Proof. We prove the result by contradiction. Suppose there are such x, y and z. We know that $L \cup \{x\} \subset \mathcal{M}_x(S)$ and $|\mathcal{M}_x(S)| = l + 2$. Therefore, at least one of y and z must appear between x. An analogous statement holds for y and z. Without loss of generality, suppose that y appears between x. Then yxy (adjacent to xyx) is not in S, which forces z to appear between

y. That, in turn, forces x to appear between z. That is, the nodes xyx, yzy and zxz are in S. But then, none of the nodes $xyz \xrightarrow{\theta} yzx \xrightarrow{\theta} zxy$ are in S, which cannot hold. □

Remark 3.4.3. Let S be a maximum independent set of $B(d,3)$ with loops a and possibly b. There cannot be two different digits x and y such that $m_x(S) = m_y(S) = l+1$. If there were, then xyx and yxy would have to be in S, leading to a contradiction.

Proposition 3.4.4. *Let S be a (possibly loop-less) maximum independent set of $B(d,3)$, $d \geq 3$. Suppose there is no digit z such that $m_z(S) = l+1$. Then, there must be exactly two digits x and y such that $m_x(S) = m_y(S) = l+2$. Moreover, $\mathcal{M}_x(S) = \mathcal{M}_y(S) = L \cup \{x,y\}$.*

Proof. We just need to show that $m_x(S) = m_y(S) = l+2$. Lemma 3.3.7 then implies that $\mathcal{M}_x(S) = \mathcal{M}_y(S) = L \cup \{x,y\}$.

Without loss of generality, we can assume that $m_{d-1}(S) \leq m_{d-2}(S)$ and $m_{d-2}(S) \leq m_i(S)$ for all $i < d-2$. We know that $m_{d-1}(S) \geq l+2$ and we want to prove that $m_{d-2}(S) = l+2$. By Lemma 3.4.2, this would imply that $d-2$ and $d-1$ are the only digits with this property.

We prove our result by induction on d. We manually check the result for all $d \leq 5$. Now, let d be greater than 5 and consider $S' = S \cap B(d-1,3)$. By the inductive hypothesis, we must have one of two possibilities:

Case 1: S' has exactly one digit z with $m_z(S') = l+1$. If $z = d-2$, we are done.

Suppose that $z \neq d-2$. By Remark 3.4.3, $m_{d-2}(S') > m_z(S')$. On the other hand, we have $m_{d-2}(S) \leq m_z(S)$, and $m_{d-2}(S')$ has to be at most $m_{d-2}(S)$. Since $m_z(S')$ is at most $m_z(S)$, we must have $m_z(S') = m_z(S) - 1$ and $m_{d-2}(S') = m_{d-2}(S)$. This means that $z(d-1)z \notin S$ and $(d-2)(d-1)(d-2) \in S$. We then have that $m_{d-2}(S) = l+2$, as we wanted to show.

Case 2: S' has exactly two digits x and y with $m_x(S') = m_y(S') = l+2$. We split this situation in two subcases.

Case 2.1: We suppose $x, y \neq d-2$. By an argument similar to that of Case 1, we know that

$$\mathcal{M}_x(S) = \mathcal{M}_y(S) = L \cup \{x, y, d-1\},$$

and that $d-1 \notin \mathcal{M}_{d-2}(S)$. On the other hand, we know that

$$m_{d-1}(S) \leq m_x(S) = l+3 = m_{d-2}(S).$$

By cardinality, one of x or y appears between $d-1$ in S. Without loss of generality, suppose $(d-1)x(d-1) \in S$. None of the nodes $x(d-2)(d-1)$,

$(d-2)(d-1)x$ and $(d-1)x(d-2)$ can be in S, because they are adjacent to $(d-2)(d-1)(d-2)$, $(d-1)x(d-1)$ and $x(d-2)x$, respectively. Thus, S cannot be a maximum independent set, a contradiction.

Case 2.2: Either x or y equals $d-2$. Suppose $y = d-2$. Since $m_{d-2}(S') = l+2$, if $(d-2)(d-1)(d-2) \in S$, then $m_{d-2}(S) = l+2$, and the result follows. Therefore, we assume $(d-2)(d-1)(d-2) \notin S$. This forces $x(d-1)x \notin S$ as well.

We know that

$$\mathcal{M}_x(S) = \mathcal{M}_{d-2}(S) = L \cup \{x, d-2, d-1\}. \tag{3.4.1}$$

Since $m_{d-1}(S) \leq m_{d-2}(S) = l+3$, there can be at most two digits, besides the loops and itself, which do not appear between $d-1$ in S. Call them u and v (potentially, $u = v$).

Case 2.2.1: Suppose $u \neq v$. We assume $u, v \neq d-2$. That means that $(d-1)(d-2)(d-1) \in S$. Since $u \neq v$, we can assume without loss of generality that $u \neq x$. Then $xux \in S$ and $(d-2)u(d-2) \in S$. The nodes $(d-1)u(d-2)$ and $(d-2)u(d-1)$ must be in S, because the rest of the nodes in their cycles are adjacent to something just shown to be in S. We know that $(d-1)u(d-1) \notin S$, because of the very definition of u. Plus, the nodes $u(d-1)(d-1)$ and $(d-1)(d-1)u$ are adjacent two one of the two nodes we just mentioned being in S. Therefore, neither of them belong to S. That gives us a contradiction.

Therefore, one of u, v must be $d-2$, and so we have (3.4.1) and

$$\mathcal{M}_{d-1}(S) = L \cup \{u, d-2, d-1\}.$$

We want to show that $u = x$. Assume the contrary. Then xux and $(d-1)x(d-1)$ are in S. Therefore, by inspecting their cycles we see that both $xu(d-1)$ and $(d-1)ux$ must be in S. On the other hand, either $u(d-1)(d-1) \in S$ or $(d-1)(d-1)u \in S$.

However, $u(d-1)(d-1) \in S$ implies $xu(d-1) \notin S$, and $(d-1)(d-1)u \in S$ implies $(d-1)ux \notin S$. Therefore, $u = x$. By Lemma 3.4.1 applied to x, $d-1$ and $d-2$, this is a contradiction.

Case 2.2.2 $u = v$. If we assume $u \neq x$ and $u \neq d-2$ and proceed as in the previous case, we get a contradiction. Therefore, $u = x$ or $u = d-2$. In either case, Lemma 3.4.1 applied to x, $d-1$ and $d-2$ leads to a contradiction. □

We now state our main result.

Theorem 3.4.5 (Characterization of the Maximum Independent Sets of $B(d,3)$). *For all positive d we have:*

1. *Any orbit of independent sets of $B(d,3)$ under the action of $\mathbb{S}_d$ is obtained from the orbit of $\{000\}$ under $\mathbb{S}_1$ and the orbit of $\{000, 010, 111\}$ under $\mathbb{S}_2$ by a* unique *sequence of applications of f, f', g, and g'.*

2. *Let S be a maximum independent set of $B(d,3)$. Then the stabilizer of S is generated by disjoint transpositions. In particular, this implies that the size of the stabilizer of S is a power of 2.*

3. *Let $b_{d,k}$ be the number of orbits of maximum independent sets in $B(d,3)$ whose elements have stabilizers of size 2^k. Then we have the recurrence relation*

$$\begin{cases} b_{1,0} = 1, \\ b_{2,0} = 3, \\ b_{d,k} = 2b_{d-1,k} + 2b_{d-2,k-1} & \textit{for } d \geq 3, \end{cases}$$

and the generating function

$$\sum_{d=1}^{\infty}\sum_{k=0}^{\infty} b_{d,k} t^d s^k = \frac{t+t^2}{1-2t-2t^2 s}.$$

4. *Let a_d be the number of maximum independent sets of $B(d,3)$. Then a_d satisfies*

$$\begin{cases} a_1 = 1, \\ a_2 = 6, \\ a_d = 2da_{d-1} + d(d-1)a_{d-2} & \textit{for } d \geq 3, \end{cases}$$

and has exponential generating function

$$\sum_{d=1}^{\infty} \frac{a_d t^d}{d!} = \frac{t+t^2}{1-2t-t^2}.$$

Proof. For $d = 1$, the only maximum independent set of $B(1,3)$ consists of the unique node $\{000\}$. For the case of $d = 2$, it can be checked manually that the three orbits of maximum independent sets under $\mathbb{S}_2$ are the orbits of $\{000, 010, 011\}$, $\{000, 010, 110\}$, and $\{000, 010, 111\}$. Note that the first two of these are $f(\{000\})$ and $f'(\{000\})$ respectively.

Thus, the existence statement in (1) follows from Propositions 3.4.4, 3.3.6, and 3.3.8. The uniqueness comes from Lemmas 3.3.3, 3.3.4, and Corollary 3.3.9.

The statements in (2) and (3) follow from the previous result and the description of the stabilizers in Propositions 3.3.1 and 3.3.2.

Finally, the generating function in (4) is obtained by substituting $s = 1/2$ into the previous generating function, because

$$a_d = \sum_{k=0}^{\infty} \frac{d! b_{d,k}}{2^k}.$$

The recurrence follows immediately. □

Remark 3.4.6. The sequence a_d is under A052608 in Sloane's Encyclopedia of Integer Sequences ([55]).

For illustrative purposes, we show the values of $b_{d,k}$, for all $d \leq 6$.

$k \backslash d$	1	2	3	4	5	6
0	1 (1,0)	3 (2,1)	6 (4,2)	12 (8,4)	24 (16,8)	48 (32,16)
1			2 (2,0)	10 (8,2)	32 (24,8)	88 (64,24)
2					4 (4,0)	28 (24,4)

In each entry, the first number between parentheses indicates the number of orbits whose elements have only one loop, whereas the second number indicates the number of orbits whose elements have two loops.

3.5 Loop-less Maximum Independent Sets

In this section, we analyze the number of *loop-less* maximum independent sets of $B(d, 3)$, for all d. Recall from the introduction that the size of a loop-less maximum independent set of $B(d, 3)$ is

$$\alpha^*(d, 3) = \frac{d^3 - d}{3} = \alpha(d, 3) - 1.$$

By maximum independent set, we will continue to mean a maximum independent set *with* loops.

We define a bijection between maximum independent sets and loop-less maximum independent sets of $B(d, 3)$.

Definition 3.5.1. Let S be a maximum independent set of $B(d, 3)$, $d \geq 3$ with loops a and possibly b. We define

$$h(S) = \begin{cases} S \backslash \{aaa\} & \text{if } S \text{ has only one loop,} \\ S \backslash \{aaa, bbb, aba\} \cup \{aab, bba\} & \text{if } S \text{ has two loops } a < b, \\ S \backslash \{aaa, bbb, aba\} \cup \{baa, abb\} & \text{if } S \text{ has two loops } a > b. \end{cases} \quad (3.5.1)$$

Proposition 3.5.2. *Let S be a maximum independent set of $B(d, 3)$. Then $h(S)$ is a loop-less maximum independent set of $B(d, 3)$.*

Proof. Let S be a maximum independent set of $B(d, 3)$. If S has only one loop, then eliminating it leaves us with an independent set of the correct size.

If S has two loops, then $h(S)$ is a set of the correct size, since the nodes we added were not already present in S. However, we must see that $h(S)$ is an independent set. Assume $a < b$. Suppose we have a node adjacent to aab. Then it is of the form abx or xaa. Since bxb and aaa are in S, then abx and xaa cannot be in S. Suppose now that we have a node adjacent to bba. Then it must be bax or xbb. Again, we know that aba and bbb are in S. Therefore, the nodes we add are not adjacent to any other nodes in the construction, and the result follows. The case $a > b$ is proved analogously. □

Proposition 3.5.3. *The function h is injective.*

Proof. Let S and S' be two different maximum independent sets of $B(d, 3)$. Then showing that $h(S) \neq h(S')$ is just a matter of analyzing all the possible combinations of loops and their relative order in S and S'. We leave the details to the reader. □

The following technical Lemma is used to prove the surjectivity of h.

Lemma 3.5.4. *Let S be a maximum independent set with two loops a and b. Let τ be the transposition of a and b. Let $S' = S \backslash \{aaa, bbb, aba\}$. Then $S' = \tau S'$.*

Proof. We must show that for every node $w \in S'$ such that $a \in w$, we have $w[a \to b] \in S'$ and vice versa.

Notice that any node of S' cannot contain a and b simultaneously. The nodes that contain two a's or two b's are axa and bxb, and they are in S' for all $x \neq a, b$. Thus, $xay \notin S'$ for all $x, y \neq a$, so the nodes that contain only one a are xya or axy for $x, y \neq a$. If $xya \in S'$, then $bxy \notin S'$, and so xyb must be in S' in order to have one element from its cycle. We can prove that $axy \in S'$ implies $bxy \in S'$ in a similar fashion. □

Proposition 3.5.5. *The function h is surjective.*

Proof. Let S be a loop-less maximum independent set of $B(d, 3)$. By Proposition 3.4.4, we have two possibilities:

If there is a digit x such that $m_x(S) = 1$, then there is no node of the form xxy or yxx. Therefore, $S' = S \cup \{xxx\}$ is a maximum independent set of $B(d, 3)$, and $S = h(S')$.

On the other hand, if there are two digits x and y such that $m_x(S) = m_y(S) = 2$, then we have either $xxy, yyx \in S$ or $yxx, xyy \in S$. In the first case, we construct

$$S' = S \cup \{xxx, yyy, xyx\} \backslash \{xxy, yyx\}.$$

If $x < y$, then $S = h(S')$. If $x > y$, then $S = h(\tau S')$, where τ is the transposition of x and y. The remaining case is dealt with analogously. □

Theorem 3.5.6. *Let a_d^* be the number of loop-less maximum independent sets of $B(d,3)$. Then $a_d^* = a_d$.*

Proof. Propositions 3.5.3 and 3.5.5 show that there is a bijection between the set of maximum independent sets and loop-less maximum independent sets of $B(d,3)$. □

3.6 Generalizations

In this chapter, we give a complete characterization of the structure of the maximum independent sets of $B(d,3)$. We believe that this paves the way for a characterization of the maximum independent sets of de Bruijn graphs with word length higher than 3. For example, Lichiardopol proved that

$$\alpha(d,D) = \frac{(D-1)(d^D-d)}{2D} + 1 \quad \text{and} \quad \alpha^*(d,D) = \frac{(D-1)(d^D-d)}{2D},$$

for $D = 3$, 5, and 7, and he conjectured that these hold for every prime $D \geq 3$ [43]. We think that our procedures can be generalized to higher prime numbers. For example, the statement of Proposition 3.2.3 also holds for $D = 5, 7$ and more generally for any D for which Lichiardopol's conjecture is true. However, we know that, remarkably, the number of maximum independent sets and loop-less maximum independent sets do not coincide for $D = 5$.

As en example, we present the proof of a generalized version of Proposition 3.2.3.

Remark 3.6.1. Let P be a simple path with $k > 0$ nodes. If k is even, any (of several possible) maximum independent set of P has $k/2$ nodes. If k is odd, the only maximum independent set of P has $\frac{k+1}{2}$ nodes, and it contains both endpoints of P.

We fix some notation that will help us prove the next result.

Definition 3.6.2. Let D be an odd prime number satisfying Lichiardopol's conjecture. Let x and y be two nodes in the same θ-cycle of $B(d, D)$. Then we define $\hat{\theta}(x, y)$ as the directed path from x to y obtained by applying θ successively, and *excluding* the nodes x and y.

For example, if we let $x = 00001$ and $y = 10000$, then $\hat{\theta}(x, y) = \{00010 \xrightarrow{\theta} 00100 \xrightarrow{\theta} 01000\}$.

Definition 3.6.3. Let D be an odd prime number sastisfying Lichiardopol's conjecture. Let a and b be two integers in $[\mathbf{d}]$. Let $1 \leq i \leq \frac{D-1}{2}$. Denote $C_D^i(a, b)$ the θ-cycle of $B(d, D)$ containing the node $a^{D-2i}(ab)^i$.

As an example, the θ-cycle $C_5^1(0, 1)$ contains the node 00001. The cycle $C_7^3(1, 0)$ contains the node 1101010.

Proposition 3.6.4. *Let D be an odd prime number sastisfying Lichiardopol's conjecture. Let S be a maximum independent set of $B(d, D)$. Then S contains at most two loops. Let a be a loop of S, and let $x \in [\mathbf{d}]$ be any digit which is not a loop of S. Then the node $(ax)^{(D-1)/2}a$ is in S.*

Proof. In order to prove this proposition, we assume that S contains two loops a and b, and we show that one of the θ-cycles that involve only a and b cannot contribute $\frac{D-1}{2}$ nodes to S. Therefore, having one pair of loops means that we lose one node from some cycle. If S has more than one pair of loops, then $|S| < \alpha(d, D) = \frac{(D-1)(d^D-d)}{2D}$. To analyze the proposition, we consider the cycles $C_D^i(a, b)$ and $C_D^i(b, a)$.

Set $i = 1$ and consider the cycle $C_D^1(a, b)$, which contains $a^{D-2}ab$ and ba^{D-1}. These two nodes are not in S, since they are adjacent to a^D. The path $\hat{\theta}(a^{D-2}ab, ba^{D-1})$ has $D-2$ nodes, and the path $\hat{\theta}(ba^{D-1}, a^{D-2}ab)$ has 0 nodes. We have an analogous consideration for the cycle $C_D^1(b, a)$. If any of the two cycles considered contribute less than $\frac{D-1}{2}$ nodes to S, we are done.

Suppose that both $C_D^1(a, b)$ and $C_D^1(b, a)$ contribute $\frac{D-1}{2}$ nodes to S. By Remark 3.6.1, all four endpoints of the paths $\hat{\theta}(a^{D-2}ab, ba^{D-1})$ and $\hat{\theta}(b^{D-2}ba, ab^{D-1})$ must be in S. We now set $i = 2$ and consider the cycles $C_D^2(a, b)$ and $C_D^2(b, a)$. From these cycles, the nodes $a^{D-4}abab$, $baba^{D-3}$, $b^{D-4}baba$ and $abab^{D-3}$ cannot be in S, since they are adjacent to endpoints of the paths $\hat{\theta}(a^{D-2}ab, ba^{D-1})$ and $\hat{\theta}(b^{D-2}ba, ab^{D-1})$ above. In the cycle $C_D^2(a, b)$, the path $\hat{\theta}(a^{D-4}abab, baba^{D-3})$ contains $D-4$ nodes, and the path $\hat{\theta}(baba^{D-3}, a^{D-4}abab)$ contains 2 nodes. An analogous consideration holds for the cycle $C_D^2(b, a)$. If any of these two cycles contributes less than $\frac{D-1}{2}$ nodes, then we are done. Otherwise we repeat this procedure.

The procedure outlined in the previous paragraph can be repeated until $i = \frac{D-1}{2}$. More precisely, by an inductive argument we shor for each $i < \frac{D-1}{2}$ that the nodes $a^{D-2i}(ab)^i$ and $(ba)^i a^{D-2i}$ are not in S, and that the endpoints of $\hat{\theta}(a^{D-2i}(ab)^i, (ba)^i a^{D-2i})$ are in S, and the analogous swapping a and b.

The cycle $C_D^{\frac{D-1}{2}}(a,b)$ contains the nodes $a(ab)^{\frac{D-1}{2}}$ and $(ba)^{\frac{D-1}{2}}a$, which cannot be in S. The path $\hat{\theta}(a(ab)^{\frac{D-1}{2}}, (ba)^{\frac{D-1}{2}}a)$ contains only the node $(ab)^{\frac{D-1}{2}}a$. If this cycle is to contribute $\frac{D-1}{2}$ nodes to S, then this node must be present. An analogous consideration shows that the node $(ba)^{\frac{D-1}{2}}b$ from the cycle $C_D^{\frac{D-1}{2}}(b,a)$ must be in S. But these two nodes are mutually adjacent. So, at least one of the cycles $C_D^{\frac{D-1}{2}}(a,b)$ and $C_D^{\frac{D-1}{2}}(b,a)$ contributes less than $\frac{D-1}{2}$ nodes to S.

Figure 3.4 illustrates this process for $D = 7$. The rightmost cycles correspond to $i = \frac{D-1}{2}$, and contain the two (boxed) nodes that are mutually adjacent. The nodes known not to be in S are crossed out. The nodes known to be in S are underlined. The nodes that remain unmarked may or may not be in S. □

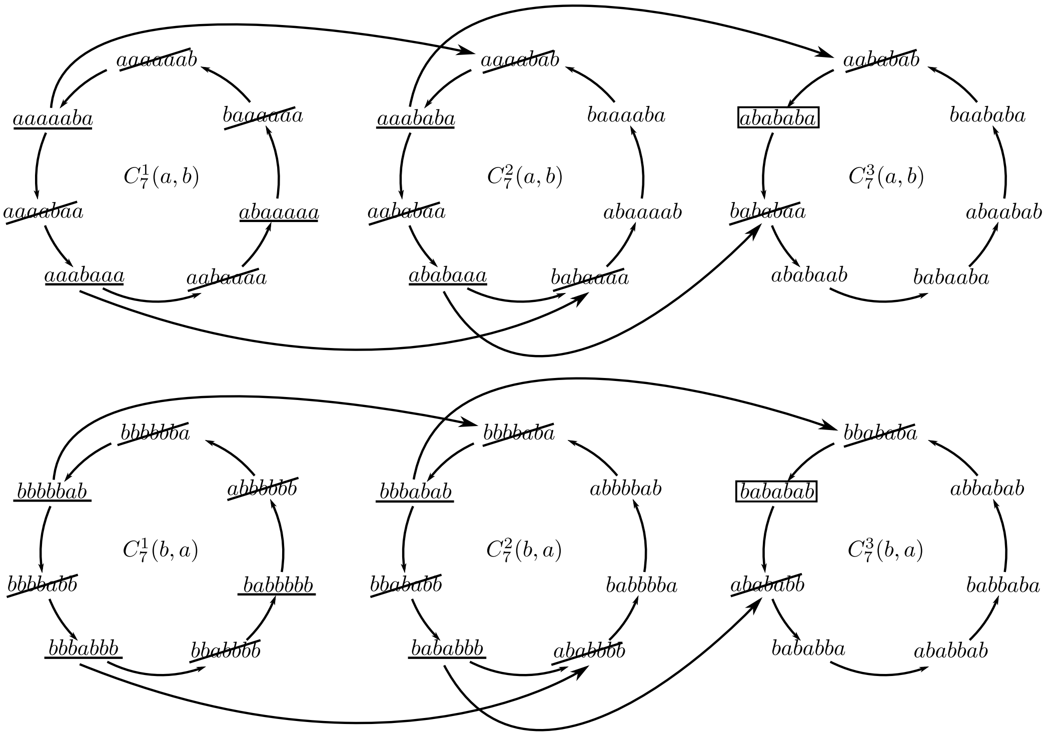


Figure 3.4: An example illustrating the process outlined in the proof of Proposition 3.6.4.

As we can see, the general version of the result requires a far more elaborate proof than was necessary for $D = 3$, but it is manageable.

Chapter 4

Additive Graph Labelings

4.1 Introduction

This chapter deals with two problems inspired by the study of biological networks. A biological network can be described using a directed graph $G = (V, E)$ with its edges labeled by a function $f_E : E \to \mathbb{Z}_2$, where $\mathbb{Z}_2 = \{0, 1\}$. The nodes of the graph represent concentrations of chemical species. A label 0 between two nodes indicates an *activates* relation, and a label 1 indicates an *inhibits* relation. A network is said to be *monotone* if and only if there exists a function $f_V : V \to \mathbb{Z}_2$ that satisfies

$$f_E((u, v)) = f_V(u) + f_V(v), \tag{4.1.1}$$

for every (u, v) in E. For technical reasons (see [15]), it suffices to study the undirected variant of this problem. The characterization of monotone systems is given by the following result.

Theorem 4.1.1 ([15])**.** *A network G is monotone if and only if every cycle in G has an even number of edges labeled with* 1.

We can generalize this problem to other sets of labels. The expression (4.1.1) remains valid if we replace $\mathbb{Z}_2$ by $\mathbb{Z}_d$, for any integer $d > 1$. In fact, we can replace $\mathbb{Z}_2$ by any abelian group H, with the understanding that the operation on the right-hand-side of (4.1.1) is the group operation in H (see Section 4.4). We study the particular case $H = \mathbb{Z}_d$, which is more transparent and hints at the solution of the general case.

We state our two problems. Let $G = (V, E)$ be a simple undirected graph. Let $d > 1$ be an integer. We call any function $f_V : V \to \mathbb{Z}_d$ a v-labeling of G. Analogously we call any function $f_E : E \to \mathbb{Z}_d$ an e-labeling of G.

Some of the results in this chapter appear in [19].

Problem 4.1.2. Let f_E be an e-labeling of G. Is there a v-labeling f_V of G, such that

$$f_E((u,v)) = f_V(u) + f_V(v), \tag{4.1.2}$$

for every edge $(u,v) \in E$? If so, how many are there?

There is a natural dual of this problem.

Problem 4.1.3. Let f_V be a v-labeling of G. Is there an e-labeling f_E of G, such that

$$f_V(u) = \sum_{(u,v)\in E} f_E((u,v)), \tag{4.1.3}$$

for every vertex $u \in V$. If so, how many are there?

In order to more explicitly understand this duality, we state the following result.

Lemma 4.1.4. *If applied to a graph that is actually a simple cycle Problems 4.1.2 and 4.1.3 are essentially the same.*

Proof. In a simple cycle, the roles of the nodes and the edges are interchangeable: every edge connects two nodes, and every node is reached by two edges. □

The subject of graph labeling encompasses a myriad variants, including harmonious labelings [23] and felicitous labelings [41]. A comprehensive survey of the subject can be found in [21].

We introduce two definitions that simplify the exposition.

Definition 4.1.5. Let $G = (V, E)$ be a graph. Let f_E be an e-labeling of G. If there exists a v-labeling f_V such that condition (4.1.2) is satisfied for every edge, we say that f_E is *e-additive*, and that f_V is *valid* for f_E. We define $\kappa(G, f_E)$ as the number of valid v-labelings for f_E.

Definition 4.1.6. Let $G = (V, E)$ be a graph, and let f_V be a v-labeling of G. If there exists an e-labeling f_E of G such that condition (4.1.3) is satisfied for every node, we say that f_V is *v-additive*, and that f_E is *valid* for f_V. We define $\kappa(G, f_V)$ as the number of valid e-labelings for f_V.

The similarity between the two definitions above is another indication of the duality of the problems studied. Figures 4.1 and 4.2 show instances of both problems.

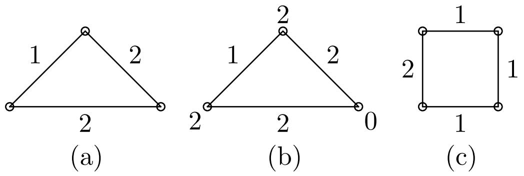


Figure 4.1: Different graphs e-labeled with $\mathbb{Z}_3$. (a) An additive e-labeling and (b) a valid v-labeling for it. (c) A non-additive e-labeling.

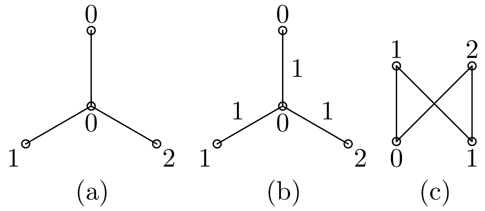


Figure 4.2: Different graphs v-labeled with $\mathbb{Z}_3$. (a) An additive v-labeling and (b) a valid e-labeling for it. (c) A non-additive v-labeling.

A graph G, v-labeled with f_V is v-additive if and only if every connected component of G is v-additive with the v-labeling induced by f_V. Furthermore, if $G_1, \ldots, G_r$ are the connected components of G, we have

$$\kappa(G, f_V) = \prod_{i=1}^{r} \kappa(G_i, f_V).$$

This analysis also holds for e-labelings. Therefore, from this point on we restrict our work to connected graphs.

The following two definitions are crucial to the solution of Problem 4.1.2.

Definition 4.1.7. We say that an e-labeled graph (G, f_E) has the *even-cycle property* if every cycle of even length in G, with edges $e_1, \ldots, e_{2k}$, satisfies

$$\sum_{l \text{ odd}} f_E(e_l) = \sum_{l \text{ even}} f_E(e_l). \tag{4.1.4}$$

Definition 4.1.8. We say that an e-labeled graph (G, f_E) has the *odd-cycle property* if for every cycle of odd length in G, with edges $e_1, \ldots, e_{2k+1}$, there exists $x \in \mathbb{Z}_d$ such that

$$\sum_{l=1}^{2k+1} f_E(e_l) = 2x. \tag{4.1.5}$$

Remark 4.1.9. Notice that if d is odd, then definition 4.1.8 imposes no restrictions on the cycles of G, since 2 is invertible modulo d.

Remark 4.1.10. If we let $d = 2$, then the even- and the odd-cycle properties mean the same: The number of 1's in every cycle of G has to be even.

We now combine the definitions presented above in order to characterize additive e-labelings.

Definition 4.1.11. Let (G, f_E) be an e-labeled graph. We say that (G, f_E) is *e-compatible* if either one of the following conditions holds

- d is odd and (G, f_E) has the even-cycle property;
- d is even and (G, f_E) has both the even- and the odd-cycle properties.

Remark 4.1.12. The preceding definitions take into account *all* the cycles of a graph, not just its simple cycles. The example in Figure 4.3 shows two simple cycles joined at a vertex. Both cycles are e-labeled, and each of them, considered as a graph, is additive. However, they assign different labels to the shared vertex. This incompatibility only appears if we check non-simple cycles too.

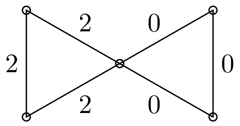


Figure 4.3: Two simple cycles joined at a vertex. Their edges are labeled with $\mathbb{Z}_3$.

We define a shorthand for a quantity that we encounter often when dealing with Problem 4.1.3.

Notation 4.1.13. Let $G = (V, E)$ be a graph, and f_V a v-labeling of G, and let $V' \subseteq V$ be any set of nodes of G. We write $\sum V'$ for the sum $\sum_{v \in V'} f_V(v)$.

The following definition states the properties that characterize for Problem 4.1.3.

Definition 4.1.14. Let (G, f_V) be a v-labeled graph. We say that (G, f_V) is *v-compatible* if either one of the following conditions holds

- G is bipartite, with $V = V_1 \sqcup V_2$, and
$$\sum V_1 = \sum V_2; \tag{4.1.6}$$

- G is not bipartite and there exists $x \in Z_d$ such that
$$\sum V = 2x. \tag{4.1.7}$$

Remark 4.1.15. If G is a bipartite graph, then the first condition implies the second condition. Furthermore, if d is odd, the second condition imposes no restriction, since 2 is then invertible modulo d.

We now have our main result, which once again shows the similarity betwen the two problems.

Theorem 4.1.16. *Let $G = (V, E)$ be a graph. Then*

- *A v-labeling f_V G is v-additive if and only if it is v-compatible.*
- *An e-labeling f_E of G is e-additive if and only if it is e-compatible.*

Proof. This Theorem is a consequence of Theorems 4.2.13 and 4.2.25, which we prove in the next section. □

We also compute the numbers $\kappa(G, f_V)$ and $\kappa(G, f_E)$.

Theorem 4.1.17. *Let f_E be an e-additive labeling of a graph G. Then*

- *If G is bipartite, $\kappa(G, f_E) = d$.*
- *If G is not bipartite, then*
    - *if d is odd, $\kappa(G, f_E) = 1$;*
    - *if d is even, $\kappa(G, f_E) = 2$.*

Let f_V be a v-additive labeling of a graph G. Then

- *If G is bipartite, $\kappa(G, f_V) = d^{m-n+1}$.*
- *If G is not bipartite, then*
    - *if d is odd, $\kappa(G, f_V) = d^{m-n}$;*
    - *if d is even, $\kappa(G, f_V) = 2d^{m-n+1}$.*

Proof. The result concerning e-additive labelings can be derived from Theorem 4.2.14. The result concerning v-additive labelings follows from Theorem 4.3.8. □

We give a characterization of our problems in terms of graphs, and present algorithms to solve them, in Section 4.2. In Section 4.3 we study the kernel of the incidence matrix of G modulo an integer $d > 1$, and we use this to derive the value of $\kappa(G, f_V)$. We apply Problem 4.1.2 to the theory of toric ideals, and we generalize our results to the setting of labels in an arbitrary abelian group in Section 4.4.

4.2 Graphical characterization

4.2.1 Additive e-labelings

In this section, we analyze Problem 4.1.2. We show that a given e-labeling is additive if and only if the cycles of (G, f_E) satisfy certain conditions. Additionally, we present polynomial algorithms to recognize whether an e-labeling is additive and to generate all valid v-labelings of it.

Throughout this chapter, we use the term cycle with a general meaning (i.e. they need not be simple cycles).

Notation 4.2.1. If $C = (V, E)$ is a cycle of length k in G, we number its nodes "consecutively" $v_1, \ldots, v_k$ and its edges $e_1, \ldots, e_k$, where $e_i = (v_i, v_{i+1})$ for all $i < k$, and $e_k = (v_k, v_1)$.

We first show that compatibility is a necessary condition for additivity.

Lemma 4.2.2. *If (G, f_E) is an additive e-labeled graph, then (G, f_E) has the even-cycle property.*

Proof. Let $e_1, \ldots, e_{2k}$ be the edges of a cycle of even length in G. Recall that $e_i = (v_i, v_{i+1})$. Let f_V be a v-labeling of G satisfying (4.1.2). We have

$$\sum_{l \text{ even}} f_E(e_l) = \sum_{l \text{ even}} (f_V(v_l) + f_V(v_{l+1}))$$
$$= \sum_{l \text{ odd}} (f_V(v_l) + f_V(v_{l+1})) = \sum_{l \text{ odd}} f_E(e_l).$$

□

Lemma 4.2.3. *If d is even, and (G, f_E) is an additive e-labeled graph, then G has the odd-cycle property.*

Proof. Let $e_1, \ldots, e_{2k+1}$ be the edges of a cycle of odd length in G. Let f_V be a v-labeling of G satisfying (4.1.2). We have

$$\sum_{l=1}^{2k+1} \frac{d}{2} f_E(e_l) = \sum_{l=1}^{2k+1} (\frac{d}{2} f_V(v_l) + \frac{d}{2} f_V(v_{l+1})) = \sum_{l=1}^{2k+1} d f_V(v_l) = 0.$$

□

We have just shown that compatibility is necessary for additivity. In fact, the compatibility conditions are sufficient for additivity. We break up the proof of this result into several smaller Lemmas. At the end of this section, we present a polynomial algorithm that allows us to decide whether an e-labeling is additive, and if so, gives us all its valid v-labelings.

Lemma 4.2.4. *Let (G, f_E) be a connected additive e-labeled graph, and suppose that f_V and f'_V are valid v-labelings of (G, f_E). If there is a vertex $v \in V$ such that $f_V(v) = f'_V(v)$, then $f_V = f'_V$.*

Proof. Let $v \in V$ be such that $f_V(v) = f'_V(v)$. Let $v' \in V$. We prove that $f_V(v') = f'_V(v')$ by induction on the distance k between v and v'. The conclusion holds for $k = 0$, so we assume now that $k > 0$ and f_V and f'_V are equal on all the vertices at distance less than k from v. Let v' be at distance k. Let $\tilde{v} \in V$ be such that $d(v, \tilde{v}) = k - 1$ and $d(\tilde{v}, v') = 1$. Then, by the induction hypothesis, $f_V(\tilde{v}) = f'_V(\tilde{v})$. Since f_V and f'_V are valid v-labelings, $f_V(v') = f_E(v', \tilde{v}) - f_V(\tilde{v}) = f_E(v', \tilde{v}) - f'_V(\tilde{v}) = f'_V(v')$ modulo d. □

The previous lemma is important because it says that, given a connected additive e-labeled graph, once we fix the label for one vertex, the rest of the vertex labels are fixed. Furthermore, it shows that $\kappa(G, f_E) \leq d$.

Remark 4.2.5. There is no equivalent version of Lemma 4.2.4 for Problem 4.1.3: Given a v-additive labeling f_V, several valid e-labelings can coincide in an edge and still be different.

The following Lemma shows that it is not necessary to inspect *every* possible cycle of odd length to check whether a graph satisfies the odd-cycle property.

Lemma 4.2.6. *Let (G, f_E), with d even, be a connected e-labeled graph satisfying the even-cycle property. Let C be any odd cycle in G. Then (G, f_E) satisfies the odd-cycle property if and only if (4.1.5) holds for C.*

Proof. We only need to prove one implication. Suppose that (G, f_E) satisfies the even-cycle property, and that (4.1.5) holds for C. Let C' be an odd cycle in G. Let $v \in C$ and $v' \in C'$. Since G is connected, there is a path P from v to v'. Let $e_1, \ldots, e_{2k+1}$, $e'_1, \ldots, e'_{2s+1}$ and $e^P_1, \ldots, e^P_r$ be the edges of C, C' and P, such that v is a vertex of e_1 and of e^P_1, and such that v' is a vertex of e'_1 and e^P_r (see Figure 4.4).

The even-cycle property of (G, f_E) applied to the even cycle made up of C, P from v to v', C' and then P from v' to v, implies that

$$\sum_{i=1}^{2k+1} (-1)^{i+1} f_E(e_i) + \sum_{i=1}^{r} (-1)^i f_E(e^P_i) + \sum_{i=1}^{2s+1} (-1)^{r+i} f_E(e'_i) + \sum_{i=1}^{r} (-1)^i f_E(e^P_i) = 0.$$

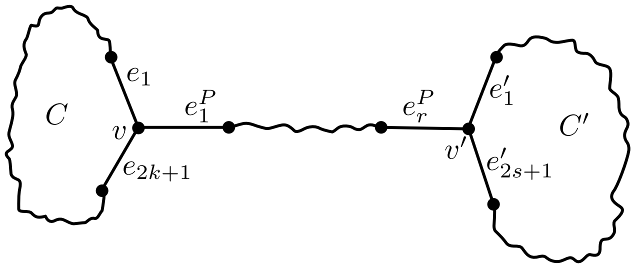


Figure 4.4: A path between v and v'.

If we multiply both sides by $d/2$, and since $d/2 = -d/2$ in $\mathbb{Z}_d$, we obtain

$$\frac{d}{2}\left(\sum_{i=1}^{2k+1} f_E(e_i) + \sum_{i=1}^{2s+1} f_E(e_i')\right) = 0.$$

Since (C, f_E) satisfies (4.1.5), we obtain

$$\frac{d}{2}\sum_{i=1}^{2s+1} f_E(e_i') = 0,$$

which means that (4.1.5) holds for C' too. □

Let C be a simple cycle, and v and v' two vertices in C. There are two ways to go from v to v' using a simple path made up of edges of C, and we often need to refer to them. The following note fixes a way of identifying those two paths.

Notation 4.2.7. Given a simple cycle C and three vertices v, v' and v'' in C, we define $C[v, v', v'']$ as the simple path in C from v to v' that contains v''. Conversely, $C[v, v', \overline{v''}]$ is the simple path from v to v' in C that *does not* contain v'' (see Figure 4.5).

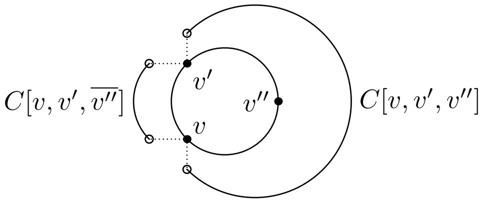


Figure 4.5: Two simple paths from v to v' in C.

Given a path P, with its edges labeled by some f_E, we can assign a label to one of its endpoints, and then propagate that label to the rest of the path. The following note makes this notion more precise.

Notation 4.2.8. Let P be a path with vertices $v_1, \dots, v_k$ and edges $e_1 = (v_1, v_2), \dots, e_{k-1} = (v_{k-1}, v_k)$. Let f_E be an e-labeling of P. We define a function $\varphi_P : \mathbb{Z}_d \to \mathbb{Z}_d$.

$$\varphi_P(c) = (-1)^{k-1} c + \sum_{l=1}^{k-1} (-1)^{k-1-l} f_E(e_l). \tag{4.2.1}$$

In other words, $\varphi_P(c)$ is the label that v_k would have if we assigned label c to v_1 and propagated it through P.

Remark 4.2.9. Let (C, f_E) be an additive e-labeled simple cycle. Let v, v', v'' be in C and set $C_1 = C[v, v', v'']$ and $C_2 = C[v, v', \overline{v''}]$. Let f_V be a valid v-labeling of (C, f_E). We have

$$\varphi_{C_1}(f_V(v)) = \varphi_{C_2}(f_V(v)) = f_V(v').$$

The following Lemma is a first approximation to a solution of Problem 4.1.2: It solves the problem for simple cycles of odd length.

Lemma 4.2.10. *If (C, f_E) is an e-compatible e-labeled simple cycle of odd length then it is e-additive. If d is odd, $\kappa(C, f_E) = 1$. If d is even, $\kappa(C, f_E) = 2$.*

Proof. Let $v_1, \dots, v_{2k+1}$ be the nodes of the cycle. Suppose that we have a valid v-labeling f_V. We want to see which are the possible values of $f_V(v_1)$. We need

$$\varphi_{C[v_1, v_{2k+1}, v_2]}(f_V(v_1)) = \varphi_{C[v_1, v_{2k+1}, \overline{v_2}]}(f_V(v_1)). \tag{4.2.2}$$

We have

$$\varphi_{C[v_1, v_{2k+1}, v_2]}(f_V(v_1)) = (-1)^{2k} f_V(v_1) + \sum_{l=1}^{2k} (-1)^{2k-l} f_E(e_l),$$

and

$$\varphi_{C[v_1, v_{2k+1}, \overline{v_2}]}(f_V(v_1)) = f_E(e_{2k+1}) - f_V(v_1).$$

Merging these two expressions with (4.2.2) we get

$$(-1)^{2k} f_V(v_1) + \sum_{l=1}^{2k} (-1)^{2k-l} f_E(e_l) = f_E(e_{2k+1}) - f_V(v_1).$$

Since $2k$ is even, this expression is equivalent to

$$2f_V(v_1) = \sum_{l=1}^{2k+1} (-1)^{l+1} f_E(e_l). \tag{4.2.3}$$

If d is odd, then 2 is invertible modulo d and equation (4.2.3) has a unique solution. That implies that there is at most one possible value for $f_V(v_1)$. Since this value gives a valid v-labeling, there is a unique valid v-labeling of (C, f_E).

If d is even, we use the odd-cycle property. Recall that this implies that the sum of the labels of the edges in the cycle is an even number. Since changing the sign of some summands does not alter the parity of a sum, the right-hand side of (4.2.3),

$$\ell := \sum_{l=1}^{2k+1} (-1)^{l+1} f_E(e_l),$$

is also even.

Equation (4.2.3) is then of the form

$$2X = 2b \pmod{2c}.$$

This equation has exactly two solutions: $X = b$ and $X = b + c$. This means that $f_V(v_1)$ is either $\ell/2$ or $(\ell + d)/2$. Since these two values for $f_V(v_1)$ give valid v-labelings, our proof is complete. □

The proof we just presented shows the relationship between the two possible ways of labeling a compatible simple odd cycle. We formalize this in the following

Corollary 4.2.11. *Let (C, f_E), with d even, be an e-additive e-labeled simple cycle of odd length. If f_V and f'_V are its two different valid v-labelings, then $f_V(v) = f'_V(v) + d/2 \pmod{d}$ for all $v \in V$.*

Let (G, f_E) be an e-labeled graph. In the following proofs, we abuse our notation. If C is a subgraph of G, then (C, f_E) stands for the graph C labeled with the restriction of f_E to the edges of C.

Lemma 4.2.12. *Let (G, f_E) be an e-compatible e-labeled connected graph. Let C and C' be two cycles of odd length in G. Let $e_1, \ldots, e_r$ be the edges of C and $e'_1, \ldots, e'_s$ be the edges of C'. Assume that C and C' share at least one vertex v_1, such that both e_1 and e'_1 are incident with v_1. Then*

$$\sum_{l=1}^{r} (-1)^{r-l} f_E(e_l) = \sum_{l=1}^{s} (-1)^{s-l} f_E(e'_l). \tag{4.2.4}$$

Proof. Consider the cycle obtained by traversing $e_1, \dots, e_r, e'_1, \dots, e'_s$. Since r and s are odd, this cycle has even length. The compatibility hypothesis implies that

$$f_E(e_1) - f_E(e_2) + \cdots + f_E(e_r) - f_E(e'_1) + f_E(e'_2) - \cdots - f_E(e'_s) = 0. \quad (4.2.5)$$

But this means

$$\sum_{l=1}^{r} (-1)^{r-l} f_E(e_l) - \sum_{l=1}^{s} (-1)^{s-l} f_E(e'_l) = 0, \quad (4.2.6)$$

which is what we wanted to prove. □

We now have all the elements we need to tackle a complete characterization of the solution of Problem 4.1.2. As one can anticipate from the preparatory Lemmas, e-additive e-labelings are *exactly* e-compatible e-labelings.

Theorem 4.2.13. *Let (G, f_E) be an e-labeled graph. Then (G, f_E) is e-additive if and only if it is e-compatible.*

Proof. Let (G, f_E) be a compatible e-labeled graph. We prove the theorem by constructing a valid v-labeling of G.

If G has odd simple cycles, call one of them C. By Lemma 4.2.10, we can choose a valid v-labeling f of (C, f_E). Pick a vertex v in C and set $\ell = f(v)$. If G has no odd cycles, choose any vertex v in G and label it with any ℓ in $\mathbb{Z}_d$.

We build a valid v-labeling f_V of (G, f_E) by propagating the label of v to the rest of the graph. For that, set $f_V(v) = \ell$. For any vertex $v' \in V$, choose a path P from v to v' and set

$$f_V(v') = \varphi_P(\ell),$$

where φ_P is as in (4.2.1). We have to prove that f_V is well-defined and that it is a valid v-labeling of (G, f_E).

Given v' and two simple paths P_1 and P_2 from v to v', we have to prove that

$$\varphi_{P_1}(\ell) = \varphi_{P_2}(\ell).$$

Let $e_1, \dots, e_r$ and $e'_1, \dots, e'_s$ be the edges of P_1 and P_2, respectively, and assume that v is an endpoint of e_1 and e'_1. We call C' the cycle formed by the union of P_1 and P_2.

If the sum of the lengths of P_1 and P_2 is even, we can use the even-cycle property of (G, f_E) applied to C'. That is,

$$f_E(e_1) - f_E(e_2) + \cdots + (-1)^{r+1} f_E(e_r) + (-1)^{r+2} f_E(e'_s) + \cdots - f_E(e'_1) = 0,$$

which can be rewritten as

$$\sum_{l=1}^{r}(-1)^{l+1}f_E(e_l)+\sum_{l=1}^{s}(-1)^{r+1+s-l+1}f_E(e'_l)=0.$$

This condition is equivalent to the identity

$$\sum_{l=1}^{r}(-1)^{l}f_E(e_l)=\sum_{l=1}^{s}(-1)^{l}f_E(e'_l). \tag{4.2.7}$$

We have

$$\varphi_{P_1}(\ell)=(-1)^r\ell+\sum_{l=1}^{r}(-1)^{r-l}f_E(e_l), \tag{4.2.8}$$

and

$$\varphi_{P_2}(\ell)=(-1)^s\ell+\sum_{l=1}^{s}(-1)^{s-l}f_E(e_l). \tag{4.2.9}$$

We must prove that $\varphi_{P_1}(\ell)=\varphi_{P_2}(\ell)$. Since the parity of r and s is the same, $(-1)^s\ell=(-1)^r\ell$, and we just need to prove that

$$\sum_{l=1}^{s}(-1)^{s-l}f_E(e_l)=\sum_{l=1}^{r}(-1)^{r-l}f_E(e_l). \tag{4.2.10}$$

If r and s are even, $(-1)^{r-l}=(-1)^{s-l}=(-1)^l$ for any integer l. Therefore, (4.2.7) shows that (4.2.10) holds. If r and s are odd, $(-1)^{r-l}=(-1)^{s-l}=(-1)^{l+1}$ for any integer l, and again (4.2.7), this time multiplied by -1, shows that (4.2.10) holds.

If r is odd and s is even, the cycle C' has odd length. We need to prove that $\varphi_{P_1}(\ell)=\varphi_{P_2}(\ell)$, which is equivalent to

$$-\ell+\sum_{l=1}^{r}(-1)^{l+1}f_E(e_l)=\ell+\sum_{l=1}^{s}(-1)^{l}f_E(e_l). \tag{4.2.11}$$

This is the same as proving that

$$2\ell=\sum_{l=1}^{r}(-1)^{l+1}f_E(e_l)+\sum_{l=1}^{s}(-1)^{l+1}f_E(e_l). \tag{4.2.12}$$

The right-hand side of (4.2.12) is the alternating sum of the labels of the edges of the odd cycle C', starting at v. By Lemma 4.2.12, this sum is equal to the alternating sum of the labels of the edges of C, starting at v.

By Lemma 4.2.10, this sum is equivalent to 2ℓ, which is what we needed to prove.

We now know that f_V is a well-defined labeling. We must show that it is also a valid v-labeling of (G, f_E). That is, for each edge (v', v''),

$$f_E((v', v'')) = f_V(v') + f_V(v''). \tag{4.2.13}$$

All the edges incident to v satisfy (4.2.13) by the previous argument. Let v' and v'' be two adjacent vertices in G, both different from v. Let e be the edge between v' and v''. Let P_1 and P_2 be paths from v to v' and v'', respectively. Let $e_1, \dots, e_r$ and $e'_1, \dots, e'_s$ be the edges of P_1 and P_2, respectively (see Figure 4.6). We must prove that

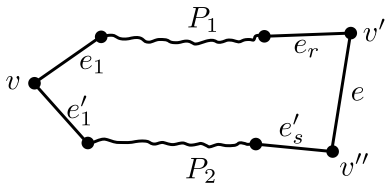


Figure 4.6: Paths between v, v' and v''.

$$\varphi_{P_1}(\ell) + \varphi_{P_2}(\ell) = f_E(e). \tag{4.2.14}$$

Consider the path $P'_2 = P_2 \cup \{e\}$. P'_2 and P_1 are two paths from v to v'. If we write $e'_{s+1} = e$, we have just proved that

$$(-1)^r \ell + \sum_{l=1}^{r} (-1)^{r-l} f_E(e_l) = (-1)^{s+1} \ell + \sum_{l=1}^{s+1} (-1)^{s+1-l} f_E(e'_l). \tag{4.2.15}$$

But the right-hand side of (4.2.15) can be split

$$\sum_{l=1}^{s+1} (-1)^{s+1-l} f_E(e'_l) = f_E(e) + \sum_{l=1}^{s} (-1)^{s+1-l} f_E(e'_l). \tag{4.2.16}$$

So joining (4.2.15) and (4.2.16), we obtain

$$(-1)^r \ell + \sum_{l=1}^{r} (-1)^{r-l} f_E(e_l) + (-1)^s \ell + \sum_{l=1}^{s} (-1)^{s-l} f_E(e'_l) = f_E(e), \tag{4.2.17}$$

which proves (4.2.14). □

Theorem 4.2.14. *The following algorithm decides whether an e-labeling is e-additive. Furthermore, the choice of label in steps 3 and 5 allows us to generate every valid v-labeling.*

Algorithm 4.2.15. Decide whether an e-labeled graph is additive and generate all valid v-labelings.

Input: A graph G, with its edges labeled by a function $f_E : E \to \mathbb{Z}_d$.
Output: **true** if and only if the (G, f_E) is an additive e-labeled graph.

```
1: Find a simple cycle $C$ of odd length in $G$, if there is one.
2: if the graph has odd cycles then
3:     Label the simple cycle $C$ using in one of the at most two ways prescribed by Lemma 4.2.10.
4: else
5:     Label any node $v$ of $G$ with any label $a \in \mathbb{Z}_d$.
6: end if
7: Propagate the labeling to the rest of the graph.
8: Check every edge. Return true if and only if equation (4.1.2) is satisfied for every edge.
```

Proof. The correctness of the algorithm follows from Theorem 4.2.13. □

Algorithm 4.2.15 runs in time $O(m + n)$, because steps 1 and 7 can be achieved using Breadth First Search (see [37]). As usual, we write $n = |V|$ and $m = |E|$.

4.2.2 Additive v-labelings

In this section, we characterize additive v-labelings. Whereas for additive e-labelings the key to the problem lies with the cycles of G, the characterization of v-labelings does not depende strongly on them.

As in the previous section, we begin our characterization with some necessary conditions for the additivity of a v-labeling. We then move on to proving that they are sufficient.

Lemma 4.2.16. *Let $G = (V, E)$ be a graph. Let f_V be a v-additive labeling of G. Then f_V is v-compatible.*

Proof. If G is bipartite, with $V = V_1 \sqcup V_2$, then we have

$$\sum V_1 = \sum_{e \in E} f_E(e) = \sum V_2.$$

Furthermore, we always have

$$\sum V = 2 \sum_{e \in E} f_E(e).$$

□

We now prove that if a labeling f_V is v-compatible, that is enough to show that f_V is v-additive. We prove it by presenting algorithms that find a solution to Problem 4.1.3, provided the v-compatibility conditions are satisfied. We present two algorithms, one for bipartite graphs and the other for non-bipartite graphs.

We start with an ancillary procedure. Let $G = (V, E)$ be a graph. Let v be a vertex of V. The *degree* d of v is the number vertices adjacent to v in G. We say that a node v is a *leaf* of G if $d(v) = 1$. Given a graph $G = (V, E)$, a v-labeling f_V and an e-labeling f_E, Algorithm 4.2.17 removes the leaves from a graph G and returns updated versions of f_V and f_E that reflect the deletions that took place.

The algorithm is quite intuitive. In each iteration, a leaf v is deleted. The e-labeling f_E is updated to assign the value $f_V(v)$ to the edge connecting v to the rest of the graph. The v-labeling f_V is updated to assign $f_V(u) - f_V(v)$ to the only vertex u adjacent to v in G.

Algorithm 4.2.17. Removing the leaves from a graph.

Input: A graph $G = (V, E)$, a v-labeling f_V and an e-labeling f_E.
Output: All the leaves of G are eliminated, and the labelings are updated to reflect these deletions.

```
while $G$ has leaves do
    $v \leftarrow$ a leaf of $G$
    $u \leftarrow$ the only node adjacent to $v$ in $G$
    $f_E((u, v)) \leftarrow f_V(v)$
    $f_V(u) \leftarrow f_V(u) - f_V(v)$
    $E \leftarrow E \backslash \{(u, v)\}$
    $V \leftarrow V \backslash \{v\}$
end while
```

The running time of this procedure is $O(n \lg n)$. This complexity can be attained by maintaining a priority queue of nodes, sorted by degree.

Remark 4.2.18. Notice that the graph and the labelings updated by Algorithm 4.2.17 satisfy conditions (4.1.6) and (4.1.7), provided the original graph satisfied it.

Armed with Algorithm 4.2.17, we can assume that our graphs are leafless. This helps us to prove that condition (4.1.6) is sufficient for bipartite graphs.

Proposition 4.2.19. *Let $G = (V = V_1 \sqcup V_2, E)$ be a bipartite graph v-labeled by f_V. If f_V is v-compatible, then there exists a solution f_E to Problem 4.1.3. Furthermore, all possible solutions can be obtained using Algorithm 4.2.20.*

Proof. Algorithm 4.2.20 shows how to construct a valid e-labeling f_E starting with a bipartite graph satisfying (4.1.6). In each iteration, an edge is removed from the graph, and the labelings are updated accordingly. If the removal causes leaves to appear, the following iteration removes them using Algorithm 4.2.17.

Algorithm 4.2.20. Labeling a bipartite graph.

Input: A bipartite graph $G = (V = V_1 \sqcup V_2, E)$ and a labeling $f_V : V \to \mathbb{Z}_d$ satisfying equation (4.1.6).
Output: A labeling $f_E : E \to \mathbb{Z}_d$ solving Problem 4.1.3.

```
while $E \neq \emptyset$ do
    Remove any leaves from $(V, E)$ using Algorithm 4.2.17
    $(u, v) \leftarrow$ any edge in $E$
    $f_E((u, v)) \leftarrow a$, for any $a \in \mathbb{Z}_d$
    $f_V(v) \leftarrow f_V(v) - a$
    $f_V(u) \leftarrow f_V(u) - a$
    $E \leftarrow E \backslash \{e\}$
end while
return $f_E$
```

The loop of this algorithm removes at least one edge from E in each iteration, and hence terminates after finitely many steps. Furthermore, condition (4.1.6) remains valid throughout the whole iteration. Therefore, when, after some time, E contains only one edge (u, v), we have $f_V(u) = f_V(v)$, and we can set $f_E((u, v)) = f_V(u)$ (this is performed by the leaf removal algorithm). □

The running time of Algorithm 4.2.20 is $O(mn \lg n)$.

Our characterization of Problem 4.1.3 for the non-bipartite case is also algorithmic. The base case and the iterative step of that algorithm are justified by the following two results, respectively.

Lemma 4.2.21. *Let $G = (V, E)$ consist only of a simple cycle of odd length. Suppose that f_V is a v-compatible labeling of G. Then, Problem 4.1.3 has* 1 *solution if d is odd and* 2 *solutions if d is even.*

Proof. This result is a corollary of the duality expressed in Lemma 4.1.4 and of Lemma 4.2.10. □

Lemma 4.2.22. *Let G be a connected, leafless, non-bipartite graph, such that $m > n$. Then, we can remove an edge of G to obtain another non-bipartite, connected graph G', such that $m' = m - 1$. Applying this Lemma repeteadly (and removing leaves), we obtain a simple cycle of odd length (i.e. $m = n$).*

Proof. Since G is not bipartite, it contains at least one simple cycle C of odd length. One of the vertices of C must have degree higher than two. Otherwise, G would be a simple cycle of odd length and we would have $m = n$.

Let v be a vertex of C with degree at least three. We move through the edges of G as follows. Start at v, and move through an edge not belonging to C. From then on, whenever we reach a vertex, we leave it by a different edge. Since G has no leaves, we can do this until we reach a vertex v' we had already visited. From this traversal, we can obtain a simple cycle C' that contains v'. Namely, the sub-path from v' to v' (see Figure 4.7). There are two edges e and e' in C that are incident to v. C' cannot have both e and e', because if we reach v during our traversal, we stop. Therefore, C' and C are different simple cycles, and there is an edge present in C' but not in C. We remove that edge.

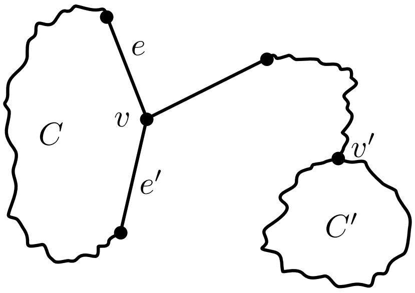


Figure 4.7: Two cycles C and C'.

The graph obtained in the previous paragraph contains the odd cycle C, and therefore is not bipartite. Furthermore, since we removed an edge from a simple cycle, the resulting graph remains connected.

The procedure outlined above allows us to remove at most $m - n$ edges. We then get a connected, leafless graph such that $m = n$. The leafless property means that the degree of each vertex is at least two. The additional property $m = n$ implies that the degree of all vertices is exactly two. Therefore, the graph is a simple cycle. Since it is not bipartite, it must have odd length. □

We now have all the tools we need to present the complete characterization of Problem 4.1.3 for non-bipartite graphs.

Proposition 4.2.23. *Let $G = (V, E)$ be a graph with an odd cycle, and $f_V : V \to \mathbb{Z}_d$ a v-labeling of G. If there exists $x \in \mathbb{Z}_d$ such that f_V satisfies condition* (4.1.7)*, then there is an e-labeling f_E that solves Problem 4.1.3.*

Proof. Algorithm 4.2.24 gives a procedure to label the edges of a non-bipartite graph satisfying the conditions of this proposition, thus proving them sufficient. Notice that the body of the loop (steps 2-8) preserves condition (4.1.7) by transforming the right-hand side from $2k$ into $2(k-a)$. The graph remains non-bipartite. When the execution reaches step 11, by Lemma 4.2.22 the graph is just a simple cycle of odd length, and we can apply Lemma 4.2.21.

Algorithm 4.2.24. Labeling a non-bipartite graph.

Input: A non-bipartite graph $G = (V, E)$ and a labeling $f_V : V \to \mathbb{Z}_d$ satisfying condition (4.1.7).
Output: A labeling $f_E : E \to \mathbb{Z}_d$ that solves Problem 4.1.3.

```
 1: while $m > n$ do
 2:    Remove the leaves from $G$ using Algorithm 4.2.17, recording the changes
       in $f_V$ and $f_E$
 3:    if $m > n$ then
 4:       Choose an edge $(u, v)$ as in the proof of Lemma 4.2.22
 5:       $f_E((u, v)) \leftarrow a$, for any $a \in \mathbb{Z}_d$
 6:       $f_V(v) \leftarrow f_V(v) - a$
 7:       $f_V(u) \leftarrow f_V(u) - a$
 8:       $E \leftarrow E \setminus \{(u, v)\}$
 9:    end if
10: end while
11: Label the edges of $G$ using Lemma 4.2.21 and complete $f_E$ with those
    labels
12: return $f_E$
```

□

Algorithm 4.2.24 has running time $O(mn \lg n)$.

The algorithms presented above implicitly shows how to obtain all possible solutions to Problem 4.1.3. Step 4 in Algorithm 4.2.20 and step 5 in

Algorithm 4.2.24 can use any element of $\mathbb{Z}_d$, and thus allows to compute any solution.

We summarize the results of this section.

Theorem 4.2.25. *Let G be a graph, with its vertices labeled with f_V. The labeling f_V is additive if and only if one of the following conditions holds:*

- *G is bipartite ($V = V_1 \sqcup V_2$) and $\sum V_1 = \sum V_2$;*
- *G is not bipartite and $\sum V$ is an even element of $\mathbb{Z}_d$.*

We still have not computed the number of solutions $\kappa(G, f_V)$. We defer that discussion until the next section.

4.3 The Kernel of A_G and of A_G^t modulo d

Problems 4.1.2 and 4.1.3 can be restated in matrix form. Set $n = |V|$ and $m = |E|$. Assume that the vertices and edges of G are numbered 1 through n and 1 through m, respectively. The *incidence matrix* of G is the $n \times m$ matrix A_G, defined by

$$A_{i,j} = \begin{cases} 1 & \text{if the vertex } i \text{ is incident with the edge } j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3.1}$$

Suppose we have an e-labeling f_E of G. We can restate Problem 4.1.2 using A_G as follows. We define $\mathbf{y} \in \mathbb{Z}_d^m$, with $y_j = f_E(j)$. Is there a vector $\mathbf{x} \in \mathbb{Z}_d^n$ such that the equation

$$A_G^t \mathbf{x} = \mathbf{y} \tag{4.3.2}$$

has a solution? If so, how many are there? Here A_G^t is the transpose of A_G.

Suppose we have a v-labeling f_V of G. The analogous restatement of Problem 4.1.3 follows. We define $\mathbf{x} \in \mathbb{Z}_d^n$, with $x_i = f_V(i)$. Is there a vector $\mathbf{y} \in \mathbb{Z}_d^m$ such that the equation

$$A_G \mathbf{y} = \mathbf{x} \tag{4.3.3}$$

has a solution? If so, how many are there?

These formulations bring to mind perfect b-matchings (see [50]), but the nature of the labels in that problem is different, as all are required to be positive integers (see [50]).

In this section, we study the kernel of the incidence matrix of G and of its transpose A_G^t, modulo some integer $d > 1$, which we define here.

Definition 4.3.1. Given $M \in \mathbb{Z}^{a\times b}$, we define $\ker_{\mathbb{Z}_d}(M) = \{\mathbf{x} \in \mathbb{Z}_d^b, M\mathbf{x} = 0 \pmod{d}\}$.

We use the Smith Normal Form (SNF) S of A_G together with the left and right multipliers U, V. Here, $U \in \mathbb{Z}^{n\times n}, V \in \mathbb{Z}^{m\times m}, S \in \mathbb{Z}^{n\times m}$ have the following properties:

1. U and V are unimodular,
2. S is a diagonal matrix, with $s_{i,i}|s_{i+1,i+1}$ for all i, and
3. $A_G = USV$.

Let $\mathbf{0}$ be the $n \times m - n$ matrix of 0's. Then the SNF S of A_G is ([27])

$$\begin{bmatrix} D & \mathbf{0} \end{bmatrix}, \tag{4.3.4}$$

where D is an $n \times n$ diagonal matrix, with $D_{i,i} = 1$ for $i \leq n-1$, and $D_{n,n} = \alpha$. Here, $\alpha = 0$ if G is bipartite and 2 otherwise.

We introduce some notation to simplify our statements.

Definition 4.3.2. Let $G = (V, E)$ be a graph and let C be any cycle of G. We associate a vector $\omega_C \in \mathbb{Z}^{|E|}$ with C. We index the coordinates of ω_C using the edges of G.

Label the consecutive edges of C

$$e_1, e_2, \ldots, e_{k-1}, e_k, \tag{4.3.5}$$

with e_1 any edge of the cycle. If C is an even cycle, we adjoin $(-1)^{i+1}$ to e_i:

$$e_1, -e_2, \ldots, (-1)^{i+1}e_i, \ldots, e_{k-1}, -e_k. \tag{4.3.6}$$

If C is an odd cycle and d is even, we adjoin $d/2$ to each edge:

$$\frac{d}{2}e_1, \ldots, \frac{d}{2}e_i, \ldots, \frac{d}{2}e_k. \tag{4.3.7}$$

Since C need not be a simple cycle, some edges may appear more than once in (4.3.5). Let $e'_1, \ldots, e'_r$ be the *distinct* edges of C. For each distinct edge e'_i, we define $\omega_{e'_i}$ to be the sum of the coefficients of each appearance of e'_i in (4.3.6) or (4.3.7). For example, if an edge e'_i appears twice, both times accompanied by a 1, then the corresponding $\omega_{e'_i}$ is 2. If one of the appearances has a 1 and the other one a (-1), then $\omega_{e'_i}$ is 0.

Given a cycle C, we define $\omega_C \in \mathbb{Z}^E$ as

$$(\omega_C)_{(u,v)} = \begin{cases} \omega_{(u,v)} & \text{if } (u,v) \text{ is in } C, \\ 0 & \text{otherwise.} \end{cases}$$

Notice that in (4.3.6), the choice of e_1 may swap the 1's and the -1's. This is not problematic, since it only changes ω_C into $-\omega_C$. The ω_C, with C of even length, are in the kernel of the incidence matrix of G and we only use them in that context.

Remark 4.3.3. Let C be a cycle of G. If the length of C is even, then the sum of the coordinates of ω_C is 0. If the length of C is odd, then the sum of the coordinates of ω_C is $d/2 \pmod{d}$.

Let (G, f_E) be an e-labeled graph and $\omega \in \mathbb{Z}^E$. We set

$$\langle \omega, f_E \rangle := \sum_{(u,v)\in E} \omega_{(u,v)} f_E((u,v)).$$

Let $\pi_d : \mathbb{Z}^{|E|} \to \mathbb{Z}_d^{|E|}$ denote the projection

$$\pi_d(x)_{(u,v)} = r_d(x_{(u,v)}),$$

where r_d is the remainder modulo d. Finally, we denote by $\mathcal{C}$ the set of even cycles in G.

The integer kernel of A_G was computed in [61], and was shown to be the submodule spanned by $\{\omega_C, C \in \mathcal{C}\}$:

$$\ker_{\mathbb{Z}}(A_G) = \langle \omega_C, C \in \mathcal{C} \rangle. \tag{4.3.8}$$

Proposition 4.3.4. *Let G be a connected graph, and let A_G be its incidence matrix. Then*

1. *If d is odd or if G is bipartite, then* $\ker_{\mathbb{Z}_d}(A_G) = \pi_d(\ker_{\mathbb{Z}}(A_G))$.

2. *If d is even and there is an odd cycle C' in G, then*

$$\ker_{\mathbb{Z}_d}(A_G) = \pi_d(\ker_{\mathbb{Z}}(A_G)) \oplus \langle \pi_d(\omega_{C'}) \rangle.$$

Proof. In this proof, $\{z_1, \dots, z_m\}$ denotes the canonical basis of $\mathbb{Z}^m$. That is, $(z_i)_i = 1$ and $(z_i)_j = 0$, for $j \neq i$. We also write $\{z_1, \dots, z_m\}$ to denote the canonical basis of $\mathbb{Z}_d^m$.

Let S be the SNF of A_G, and U,V such that $A_G = USV$, as described in (4.3.4). Equivalently, $U^{-1}A_G = SV$. Since U and V are both unimodular, they have integer inverses and hence they have integer inverses modulo d. Therefore $\ker_{\mathbb{Z}}(A_G) = \ker_{\mathbb{Z}}(SV)$ and $\ker_{\mathbb{Z}_d}(A_G) = \ker_{\mathbb{Z}_d}(SV)$, implying that

$$\ker_{\mathbb{Z}}(A_G) = V^{-1} \ker_{\mathbb{Z}}(S). \tag{4.3.9}$$

$$\ker_{\mathbb{Z}_d}(A_G) = \pi_d(V^{-1} \ker_{\mathbb{Z}_d}(S)). \tag{4.3.10}$$

Let $\mathbf{x} = (x_1, \ldots, x_m) \in \ker_{\mathbb{Z}_d}(S)$. That means that

$$S\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ \alpha x_n \end{pmatrix} = 0 \pmod{d}. \tag{4.3.11}$$

If $\alpha = 0$ (i.e. G has no odd cycles), equation (4.3.11) holds if and only if $x_i = 0 \pmod{d}$ for $i \in \{1, \ldots, n-1\}$. That means that

$$\ker_{\mathbb{Z}_d}(S) = \langle z_n, \ldots, z_m \rangle \quad \text{and} \quad \ker_{\mathbb{Z}}(S) = \langle z_n, \ldots, z_m \rangle.$$

Therefore, we have $\ker_{\mathbb{Z}_d}(S) = \pi_d(\ker_{\mathbb{Z}}(S))$, whence $\ker_{\mathbb{Z}_d}(A_G) = \pi_d(\ker_{\mathbb{Z}}(A_G))$.

If $\alpha = 2$ (i.e. G has an odd cycle) and d is odd, equation (4.3.11) holds if and only if $x_i = 0 \pmod{d}$ for $i \in \{1, \ldots, n\}$. That means that

$$\ker_{\mathbb{Z}_d}(S) = \langle z_{n+1}, \ldots, z_m \rangle.$$

Once more,

$$\ker_{\mathbb{Z}}(S) = \langle z_{n+1}, \ldots, z_m \rangle.$$

And again $\ker_{\mathbb{Z}_d}(S) = \pi_d(\ker_{\mathbb{Z}}(S))$, implying $\ker_{\mathbb{Z}_d}(A_G) = \pi_d(\ker_{\mathbb{Z}}(A_G))$.

We now assume that $\alpha = 2$ and that d is even. From equation (4.3.11) we now deduce that

$$\ker_{\mathbb{Z}_d}(S) = \langle z_{n+1}, \ldots, z_m \rangle \oplus \langle \frac{d}{2} z_n \rangle, \tag{4.3.12}$$

$$\ker_{\mathbb{Z}}(S) = \langle z_{n+1}, \ldots, z_m \rangle. \tag{4.3.13}$$

Notice that

$$\langle \frac{d}{2} z_n \rangle = \{0, \frac{d}{2} z_n\}. \tag{4.3.14}$$

Combining equations (4.3.9), (4.3.10), (4.3.12) and (4.3.13), we have

$$\ker_{\mathbb{Z}_d}(A_G) = \langle \pi_d(V^{-1} z_{n+1}), \ldots, \pi_d(V^{-1} z_m) \rangle \oplus \langle \pi_d(V^{-1} \frac{d}{2} z_n) \rangle. \tag{4.3.15}$$

$$\ker_{\mathbb{Z}}(A_G) = \langle V^{-1} z_{n+1}, \ldots, V^{-1} z_m \rangle. \tag{4.3.16}$$

Let C' be an odd cycle of G. Then $\pi_d(\omega_{C'}) \in \ker_{\mathbb{Z}_d}(A_G)$. To see why, recall that entry e_j of $\omega_{C'}$ is $d/2$ times the number of occurrences of the edge e_j in C'. For every vertex v_i of the cycle, the number of edges that enter and leave it must be the same. That means that the v_i-th entry of $A_G \omega_{C'}$ is an even number times $d/2$ (if vertex v_i is in the cycle) or 0. In both cases, $A_G \omega_{C'} = 0 \pmod{d}$.

Now, since $\pi_d(\omega_{C'}) \in \ker_{\mathbb{Z}_d}(A_G)$, we must have

$$\pi_d(\omega_{C'}) = \sum_{l=n+1}^{m} \gamma_l \pi_d(V^{-1} z_l) + \varepsilon \pi_d(V^{-1} \frac{d}{2} z_n), \tag{4.3.17}$$

where ε is 0 or 1 (see (4.3.14)). The first summand consists of multiples of the projections of even cycles (see (4.3.8)). That means that if we take the sum of the coordinates of both sides of equation (4.3.17), we get $\varepsilon = 1$ (see Remark 4.3.3). If we set

$$\gamma = \sum_{l=n+1}^{m} \gamma_l \pi_d(V^{-1} z_l),$$

we can write

$$\pi_d(V^{-1} \frac{d}{2} z_n) = \gamma - \pi_d(\omega_{C'}). \tag{4.3.18}$$

Now, take any $\mathbf{x} \in \ker_{\mathbb{Z}_d}(A_G)$. We have that

$$\mathbf{x} = \sum_{l=n+1}^{m} \beta_l \pi_d(V^{-1} z_l) + \beta \pi_d(V^{-1} \frac{d}{2} z_n).$$

Plugging in equation (4.3.18) we get

$$\mathbf{x} = \sum_{l=n+1}^{m} \beta_l \pi_d(V^{-1} z_l) + \beta(\gamma - \pi_d(\omega_{C'})).$$

If we set $\tilde{\beta}_l = \beta_l + \gamma_l$, we have

$$\mathbf{x} = \sum_{l=n+1}^{m} \tilde{\beta}_l \pi_d(V^{-1} z_l) + (-\beta) \pi_d(\omega_{C'}),$$

which shows that

$$\ker_{\mathbb{Z}_d}(A_G) = \pi_d(\ker_{\mathbb{Z}}(A_G)) \oplus \langle \pi_d(\omega_{C'}) \rangle.$$

□

We can combine this series of results to obtain the following result linking compatibility, additivity and the modular kernel of A_G.

Theorem 4.3.5. *Let (G, f_E) be an e-labeled connected graph. Let A_G be the incidence matrix of G. The following statements are equivalent:*

1. *(G, f_E) is an e-compatible e-labeled graph.*

2. *$\langle \pi_d(\omega_C), f_E \rangle = 0 \pmod{d}$, for every cycle C of G.*

3. *$\langle \omega, f_E \rangle = 0 \pmod{d}$, for all $\omega \in \ker_{\mathbb{Z}_d}(A_G)$.*

4. *If d is odd or if G is bipartite, $\langle \omega, f_E \rangle = 0 \pmod{d}$, for all ω belonging to the projection of a finite set of generators of $\ker_{\mathbb{Z}}(A_G)$. If d is even and has an odd cycle, $\langle \omega, f_E \rangle = 0 \pmod{d}$, for all ω belonging to a finite set of generators of $\ker_{\mathbb{Z}}(A_G)$ and for ω_C, for some odd cycle C.*

5. *(G, f_E) is an e-additive e-labeled graph.*

Proof. Clause 1. is equivalent to clause 5. by Theorem 4.2.13. Clause 2. is a restatement of clause 1. using a different notation. Clauses 2. and 3. are equivalent by Proposition 4.3.4. Clauses 3. and 4. also follow from that proposition: the finite sets described in clause 4. were shown to be generators of $\ker_{\mathbb{Z}_d}(A_G)$. □

Remark 4.3.6. Given a graph G, consider the *cycle space* of G ([30]). It is the $\mathbb{Z}_2$-vector space generated by the fundamental cycles of G. That is, the cycles obtained when adding an edge of G to a spanning tree.

One might be tempted to think that checking the compatibility conditions on these generators suffices to verify the compatibility of a graph with labels in $\mathbb{Z}_d$ for any d, as in the case $d = 2$. However, consider for instance the graph in Figure 4.8, in which we marked the spanning tree with edges $\{e_{14}, e_{23}, e_{24}\}$: The sum of the two fundamental triangle cycles C_1 and C_2 (represented by their $0, 1$ vectors) equals the square cycle C only when $d = 2$. This situation is depicted informally in Figure 4.9. However, if d is odd we do not impose any conditions on C_1 and C_2, and so this cannot ensure the even-cycle property we need to check. When $d \neq 2$ is even, we get $\frac{d}{2}$ times the even cycle condition, which again is not sufficient to ensure additivity. Consider for instance the labeling $f(e_{12}) = f(e_{24}) = f(e_{34}) = 1$, $f(e_{14}) = f(e_{23}) = 0$ and $d = 4$. The odd-cycle property is verified for C_1 and C_2 but the labeling is not additive.

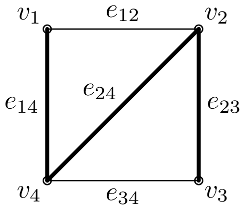


Figure 4.8: A spanning tree of a graph.

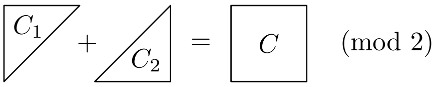


Figure 4.9: Adding two odd cycles to obtain an even one.

We now turn our attention to the modular kernel of A_G^t.

Proposition 4.3.7. *Let G be a graph, and let A_G be its incidence matrix. Then*

1. *If G is bipartite, with $V = V_1 \sqcup V_2$, then*

$$\ker_{\mathbb{Z}}(A_G^t) = \langle \omega_{V_1,V_2} \rangle,$$

where ω_{V_1,V_2} is a vector with 1 in the coordinates corresponding to vertices of V_1 and -1 in those corresponding to vertices of V_2. Furthermore,

$$\ker_{\mathbb{Z}_d}(A_G^t) = \pi_d(\ker_{\mathbb{Z}}(A_G^t))$$

2. *If G is not bipartite, then*

$$\ker_{\mathbb{Z}}(A_G^t) = \langle 0 \rangle.$$

If d is odd, then

$$\ker_{\mathbb{Z}_d}(A_G^t) = \pi_d(\ker_{\mathbb{Z}}(A_G^t)).$$

If d is even, then

$$\ker_{\mathbb{Z}_d}(A_G^t) = \langle \omega_{d/2} \rangle,$$

where $\omega_{d/2}$ is the vector that has $d/2$ in every entry.

Proof. 1. Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a vector. Then $\mathbf{x}$ is in $\ker_{\mathbb{Z}}(A_G^t)$ if and only if

$$A_G^t \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_{u_1} + x_{v_1} \\ \vdots \\ x_{u_m} + x_{v_m} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix},$$

where x_{u_i} and x_{v_i} are the variables that correspond to the vertices of edge i. This expression shows that $\omega_{V1,V2} \in \ker_{\mathbb{Z}}(A_G^t)$.

Now that we want to see when

$$\begin{pmatrix} x_{u_1} + x_{v_1} \\ \vdots \\ x_{u_m} + x_{v_m} \end{pmatrix} = 0$$

In each row of this vector we have the label of one element of V_1, and that of an element of V_2. They have to be the inverse of each other. Combined with the connectedness of G, this implies that the x_u must have the same value for every element of V_1, the x_v must have the same value for every element of V_2, and that $x_u = -x_v$. Therefore, $\mathbf{x}$ is a multiple of ω_{V_1,V_2}.

Moving into the modular setting maintains the validity of this argument.

2. Let $\mathbf{x}$ be in $\ker_{\mathbb{Z}}(A_G^t)$. An observation similar to the one just made shows that every coordinate x_v of $\mathbf{x}$ must satisfy $x_v = -x_v$, and they must all be equal to the same value.

□

Theorem 4.3.8. *Let f_V be a v-additive labeling of a graph G. We set $r = m - \operatorname{rank}_H(A_G^t)$ (i.e. the corank of A_G^T). Then*

- *If G is bipartite or if d is odd, $\kappa(G, f_V) = d^r$.*
- *If G is not bipartite and d is even, then $\kappa(G, f_V) = 2d^r$.*

Proof. The result follows from a straightforward linear algebra argument: The set of solutions to Problem 4.1.3 is a fiber of the linear map $\mathbf{x} \mapsto A_G\mathbf{x}$. □

In light of the results we have shown about the structure of the kernel of A_G, we can see that any solution to the e-labeling of the v-labeling problem can be expressed as the sum of a particular solution $\mathbf{x}_0$ and an element of $\ker_{\mathbb{Z}_d}(A_G)$ or $\ker_{\mathbb{Z}_d}(A_G^t)$, depending on the problem. The algorithms of the previous section provide an efficient combinatorial procedure to obtain such an $\mathbf{x}_0$.

4.4 Applications to Toric Varieties and Generalizations

The results of the previous sections remain valid if we swap $\mathbb{Z}_d$ for $\mathbb{G}_d$, the group of d-th roots of unity. In this case the solution to Problem 4.1.2 provides a modular version of a classical result on toric parametrizations. We refer the reader to [59] for a general introduction to toric ideals.

Let $G = (V, E)$ be a connected graph and d an integer greater than 1. Let $n = |V|$ and $m = |E|$. Let $v_1, \ldots, v_n$ be the vertices of G and let $e_1, \ldots, e_m$

be its edges. We work with complex variables x_{v_i} for each $v_i \in V$, and y_{e_i} for each $e_i \in E$. The value of x_{v_i} corresponds to the label of vertex v_i, and the value of y_{e_i} corresponds to the label of edge e_i. The statement of Problem 4.1.2 in this setting gives us

Problem 4.4.1. For which $\mathbf{y} \in \mathbb{G}_d^m$ are there $\mathbf{x} \in \mathbb{G}_d^n$ such that

$$y_{e_i} = x_{v_i} x_{v_j} \tag{4.4.1}$$

holds for every edge $e_i = (v_i, v_j) \in E$?

According to a classic result for toric parametrizations, given a vector $\mathbf{y} \in (\mathbb{C}^*)^m$ of complex nonzero numbers, there is a vector $\mathbf{x} \in (\mathbb{C}^*)^n$ satisfying (4.4.1) if and only if

$$y^{\mathbf{u}} = y_1^{u_1} \cdots y_m^{u_m} = 1, \tag{4.4.2}$$

for every $\mathbf{u} = (u_1, \ldots, u_m) \in \ker_{\mathbb{Z}}(A_G)$. Furthermore, when these conditions are satisfied, the number of such solutions is

$$g = \gcd(\{\text{maximal minors of } A_G\}), \tag{4.4.3}$$

provided that $g \neq 0$, in which case there are infinitely many solutions. We deduce from (4.3.4) that $g = 2$ or 0, depending on whether G has an odd cycle or not, respectively. It was this result which prompted us to study the incidence matrix of G in connection with Problem 4.1.2.

We now state a modular version of this toric result. We impose the additional restriction that

$$x_{v_i}^d = 1, \tag{4.4.4}$$

for all $v_i \in V$. This condition, together with (4.4.1), implies that the y_{e_i} are also in $\mathbb{G}_d$.

Theorem 4.4.2. *Let $G = (V, E)$ be a connected graph. Given $\mathbf{y} \in \mathbb{G}_d^m$, there exists $\mathbf{x} \in \mathbb{G}_d^n$ satisfying (4.4.1) if and only if*

$$y^{\mathbf{u}} = 1, \tag{4.4.5}$$

for every $\mathbf{u} \in \ker_{\mathbb{Z}_d}(A_G)$. If g is 0, there are d solutions to (4.4.1) and (4.4.4) simultaneously. If g is 2 and d is even, there are two solutions. Otherwise, there is a unique solution.

The result can be translated from Theorem 4.2.13. Alternatively, we could prove that given $\mathbf{y} \in \mathbb{G}_d^m$, there are as many solutions $\mathbf{x} \in \mathbb{G}_d^n$ as stated using the knowledge of g in (4.4.3), by checking how many of the complex solutions $\mathbf{x} \in (\mathbb{C}^*)^n$ consist of d-th roots of unity.

We can derive analogous results for v-labelings, switching A_G for A_G^t. Moreover, we could use any suitable integer matrix (e.g. the incidence matrix of a hypergraph).

As we mentioned, our results for $\mathbb{Z}_d$ or $\mathbb{G}_d$ can be extended to an arbitrary abelian group. The statement of Problems 4.1.2 and 4.1.3 are well-defined in this general setting. So are the e- and v-compatiblity conditions layed out it in Definitions 4.1.11 and 4.1.14. In Theorem 4.1.17 we stated the number of solutions. The move to an arbitrary abelian group H forces us to replace 1 and 2 in that theorem for the size of the set

$$0_H = \{x \in H, \text{ such that } 2x = 0\}.$$

This set has size 1 and 2 if we work in $\mathbb{Z}_d$, with d odd or even, respectively. The only other significant change we need to make is that whenever we speak of $d/2$, what we mean is "a non-zero element of 0_H." We can also study which results from Section 4.3 can be translated to the generalized setting.

Appendix A

Summary of Algebraic Tools and Techniques

In this appendix we present the algebraic definitions and results used throughout this thesis. We present definitions and results, but we omit all the proofs.

A.1 Groups

We refer the reader to [40] for a general introduction to the study of groups.

A.1.1 Basic definitions

Definition A.1.1. A set of elements H, together with an operation $\cdot : H \times H \to H$ and a distinguished element $e \in H$ is called a group if

1. The operation $\cdot$ is associative.
2. For every element $x \in H$, $xe = ex = x$.
3. For every element $x \in H$, there is an element $x^{-1} \in H$, called the *inverse* of x, such that $xx^{-1} = x^{-1}x = e$.

Definition A.1.2. A group $(H, \cdot)$ is called *abelian* if the operation $\cdot$ is commutative. In that case, the operation is usually written $+$, the unit element is written as 0 and the inverse of x is written $-x$.

Definition A.1.3. The cardinality of a group is called its *order*.

Definition A.1.4. Let H be a group. A subgroup M of H is a subset of H containing the unit element, and such that M is closed under the law of composition and inverse.

Theorem A.1.5. *If H is a finite group and M is a subgroup of H, then the order of M divides the order of H.*

A.1.2 The action of a group on a set

Definition A.1.6. Let H be a group and let S be a set. An *action* of H on S is a function $\cdot : H \times S \to S$ that satisfies

- $es = s$ for all $s \in S$.
- $(xy)s = x(ys)$ for all $x, y \in H$ and $s \in S$.

Definition A.1.7. Let H be a group acting on a set S. Let s be an element of S. The *stabilizer* of s, written H_s is the set of elements $x \in H$ that satisfy

$$xs = s.$$

The stabilizer of an element is always a subgroup of H.

Definition A.1.8. Let H be a group acting on a set S. Let s be an element of S. The *orbit* of s, written Hs is the subset of S defined as $\{xs \,|\, x \in H\}$.

Definition A.1.9. Let H be an abelian group. There is a canonical action of the integers $\mathbb{Z}$ on H, defined by

$$nx = \begin{cases} 0 & \text{if } n = 0 \\ (n-1)x + x & \text{if } n > 0 \\ (-n)(-x) & \text{if } n < 0 \end{cases}$$

A.1.3 Cyclic groups

Definition A.1.10. Let H be an abelian group, and let X be a subset of H. We define the *subgroup generated by* X as the group

$$\langle X \rangle = \{x_1 + \cdots + x_k \,|\, \text{for any finite combination of elements of X}\}$$

Definition A.1.11. A group H is *cyclic* if there is an element $x \in H$ such that $H = \langle x \rangle$.

Common examples of cyclic groups include the integers $\mathbb{Z}_D$ modulo d with addition, the group $\mathbb{G}_d$ of d-th roots of unity with multiplication and the integers with addition.

Remark A.1.12. All cyclic groups are abelian.

Theorem A.1.13. *Any cyclic group of finite order d is isomorphic to $\mathbb{Z}_d$.*

A.1.4 The Symmetric Group $\mathbb{S}_d$

Definition A.1.14. Let d be a positive integer. The *symmetric group* $\mathbb{S}_d$ is the group of bijections (also called permutations) of the set $\{0, \ldots, d-1\}$, together with function composition.

Definition A.1.15. An element τ of $\mathbb{S}_d$ is called a *transposition* if

$$|\{s \mid s \in H \wedge \tau s \neq s\}| = 2$$

In other words, a transposition is a permutation that only swaps two elements.

Remark A.1.16. In the literature, the symmetric group is usually defined to be the group of permutations of $\{1, \ldots, d\}$, but our choice is more suited to our work.

A.2 Rings and polynomials

A.2.1 Basic definitions

Definition A.2.1. A set of elements A, together with two operations $+ : A \times A \to A$ and $\cdot : A \times A \to A$, and two distinguished elements 0 and 1 is called a *commutative ring* if

1. The set A, with $+$ and 0 is an abelian group.

2. The operation $\cdot$ is associative and commutative.

3. For every element $x \in A$, $x1 = 1x = x$.

4. For every x, y and z in A, we have $(x + y)z = xz + yz$.

Definition A.2.2. Let A be a commutative ring. A subset $I \subseteq A$ is called an *ideal* of A if it is a subgroup of A and if $xy \in I$ for all $x \in A$ and all $y \in I$.

Remark A.2.3. Fields (e.g. rational, real or complex numbers) are special cases of commutative rings.

Definition A.2.4. Let A be a commutative ring, and let $X \subset A$ be any set of elements of A. We define the *ideal generated by X* as

$$\langle X \rangle = \{\sum_{p \in S} c_p p, \text{ with } S \subseteq X, |S| < \infty, c_p \in A\}. \tag{A.2.1}$$

A.2.2 Polynomials

As a general reference on the subject of polynomials, we refer the reader to the books [13] and [14].

Definition A.2.5. Given a field k, we can define a commutative ring $k[\mathbf{x}] = k[x_1, \ldots, x_n]$, called the ring of polynomials in n variables with coefficients in k.

Definition A.2.6. A *monomial ordering* $<$ is a relation on the set of monomials in n variables such that

1. The relation $<$ is a total ordering.

2. If $x^\alpha < x^\beta$, then $x^\alpha x^\gamma < x^\beta x^\gamma$, for any monomials x^α, x^β and x^γ.

3. The relation $<$ is a well-ordering. That is, every non-empty set of monomials has a smallest element.

Definition A.2.7. Let $p \in k[\mathbf{x}]$ be a non-zero polynomial, and let $<$ be a monomial ordering. We define the *leading term* of p with respect to $<$, $LT_<(p)$ as its maximum monomial with respect to the ordering $<$.

Remark A.2.8. It is not really relevant whether a coefficient is included in the leading term of a polynomial or not.

Definition A.2.9. Let I be a polynomial ideal. The *initial ideal* of I is defined as

$$LT_<(I) = \{LT_<(p), \text{ for all } p \in I\}. \tag{A.2.2}$$

Remark A.2.10. If a polynomial ideal I is generated by a finite set of polynomials X, it is not true, in general, the $LT_<(I)$ is generated by the leading terms of the elements of X.

Definition A.2.11. Let I be a polynomial ideal, and let $<$ be a monomial ordering. A finite subset Gb of I is a *Gröbner Basis* of I with respect to $<$, if

$$LT_<(I) = \langle LT_<(g), g \in GB \rangle. \tag{A.2.3}$$

Definition A.2.12. Let I be a polynomial ideal. We define the quotient $k[\mathbf{x}]/I$ as the partition of $k[\mathbf{x}]$ induced by the relation $p \sim q \iff p - q \in I$. The quotient $k[\mathbf{x}]/I$ has a k-vector space structure. The *dimension* of the quotient $k[\mathbf{x}]/I$ is its vector space dimension over k.

Definition A.2.13. Let I be a polynomial ideal. We define the *variety* $V(I)$ of I as the set of points y in k^n such that $p(y) = 0$ for all $p \in I$. We say that I is *zero-dimensional* if $V(I)$ has a finite number of points in the algebraic closure of k.

Theorem A.2.14. *Let I be a polynomial ideal. Then I is zero-dimensional if and only if $k[\mathbf{x}]/I$ is a finite-dimensional vector space over k. If that is the case, then $\dim_k(k[\mathbf{x}]/I)$ is the number of elements of $V(I)$, counted with multiplicity.*

Theorem A.2.15. *Let I be a polynomial ideal, and let $<$ be any monomial ordering. Then*

$$\dim_{\mathbb{C}}(\mathbb{C}[\mathbf{x}]/I) = \dim_{\mathbb{C}}(\mathbb{C}[\mathbf{x}]/LT_<(I)). \tag{A.2.4}$$

Definition A.2.16. Let I be a polynomial ideal. We define the *radical* $\sqrt{I}$ of I as the ideal given by

$$f \in \sqrt{I} \iff \exists n, \text{ such that } f^n \in I. \tag{A.2.5}$$

An ideal I is said to be radical if $I = \sqrt{I}$.

Definition A.2.17. A polynomial ideal I is a *complete intersection* if and only if it is generated by r equations, where r is its codimension.

For a definition of the Krull dimension of a polynomial ideal, see [13, Chapter 9].

Definition A.2.18. Let I be a polynomial ideal, and let f be a polynomial. Then the *colon ideal* is defined by

$$(I : f) = \{g \in \mathbb{C}[\mathbf{x}], \text{ such that } gf \in I\}. \tag{A.2.6}$$

We now describe Cohen-Macaulay rings. For the a comprehensve treatment of the subject, see [33].

Definition A.2.19. Let A be a ring. If there is exactly one proper ideal $\mathfrak{m}$ (different from A and from $\{0\}$), we say that A is a *local ring.* We usually refer to "the local ring" $(A, \mathfrak{m})$.

Definition A.2.20. Let A be a ring. An A-module M is an abelian group, together with a group action $\cdot : A \times M \to M$, satisfying

- $(a + b)x = ax + bx$, for all $a, b \in A$, $x \in M$.

- $a(x + y) = ax + ay$, for all $a \in A$, $x, y \in M$.

Definition A.2.21. Let $M \neq \{0\}$ be a module over a local ring $(A, \mathfrak{m})$. The *depth* of M is the length of any maximal regular sequence on M which is contained in $\mathfrak{m}$.

Definition A.2.22. Let M be an $(A, \mathfrak{m})$-module. Then M is a Cohen-Macaulay module if $\operatorname{depth}(M) = \dim(M)$.

Definition A.2.23. A local ring $(A, \mathfrak{m})$ is Cohen-Macaulay if it is a Cohen-Macaulay A-module.

The depth of a ring is always bounded above by the Krull dimension; equality provides interesting regularity conditions on the ring, enabling some powerful theorems to be proven in this rather general setting.

Definition A.2.24. A non-local ring A is Cohen-Macaulay if the localization $A_{\mathfrak{p}}$ is a Cohen-Macaulay local ring for every prime ideal $\mathfrak{p}$ of A.

Definition A.2.25. An ideal I of A is Cohen-Macaulay if A/I is a Cohen-Macaulay A-module.

A.3 Smith Normal Form

Definition A.3.1. A matrix $A \in \mathbb{Z}^{n \times n}$ is said to be *unimodular* if its determinant is 1 or -1 (i.e. it has an integer inverse).

Theorem A.3.2 (Smith Normal Form [56]). *Let A be an $n \times m$ integer matrix. There exist unimodular matrices $U \in \mathbb{Z}^{n \times n}$ and $V \in \mathbb{Z}^{m \times m}$, and a diagonal matrix $S \in \mathbb{Z}^{n \times m}$, such that $A = USV$, and $S_{i,i} \mid S_{i+1,i+1}$.*

Appendix B

Hilbert Series in CAS's

B.1 Introduction

In this appendix, we summarize the syntax used for working with Hilbert Series in three free Computer Algebra Systems. We work with the modified edge ideal I_G of the graph G shown in Figure B.1.

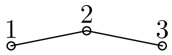


Figure B.1: A sample graph

In this case, $I_G = \langle x_1^2, x_2^2, x_3^2, x_1x_2, x_2x_3 \rangle$, and the Hilbert Series we are looking for is $1 + 3z + z^2$.

B.2 CoCoA

We first have to declare the polynomial ring we work in.

```
Use R ::= QQ[x[1..3]];
```

Next we define the ideal

```
I := Ideal(x[1]^2,x[2]^2,x[3]^2,x[1]*x[2],x[2]*x[3]);
```

And finally we ask for the Hilbert Series of `R/I`.

```
Hilbert(R/I);
```

obtaining as output

```
H(0) = 1
H(1) = 3
H(2) = 1
H(t) = 0   for t >= 3
-------------------------------
```

B.3 Macaulay2

We first have to declare the polynomial ring we work in.

```
R = QQ[x1,x2,x3]
```

Next we define the ideal

```
I = ideal(x1^2,x2^2,x3^2,x1*x2,x2*x3)
```

And finally we ask for the Hilbert Series of `R/I`.

```
numerator reduceHilbert hilbertSeries (R/I)
```

obtaining as output

```
                 2
o3 = 1 + 3T + T
```

B.4 Singular

We first have to declare the polynomial ring we work in.

```
ring r = 0,(x1,x2,x3),dp;
```

Next we define the ideal

```
ideal i = std(ideal(x1^2,x2^2,x3^2,x1*x2,x2*x3));
```

And finally we ask for the Hilbert Series of `R/I`.

```
hilb(i);
```

obtaining as output

```
//          1 t^0
//         -5 t^2
//          5 t^3
//         -1 t^5

//          1 t^0
//          3 t^1
//          1 t^2
// dimension (affine) = 0
// degree (affine)  = 5
```

We read the Hilbert Series from the second polynomial printed by the program.

Bibliography

[1] David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1–3):21–46, March 1996.

[2] Dave Bayer and Mike Stillman. Computation of Hilbert functions. *Journal of Symbolic Computation*, 14(1):31–50, July 1992.

[3] Verónica Becher. On the normality of Eulerian sequences. Manuscript, 2009.

[4] Niko Beerenwinkel, Nicholas Eriksson, and Bernd Sturmfels. Evolution on distributive lattices. *Journal of Theoretical Biology*, 242(2):409–420, September 2006.

[5] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, October 1997.

[6] Jean-Claude Bermond and Claudine Peyrat. De Bruijn and Kautz networks: a competitor for the hypercube? In F. André and J-P. Verjus, editors, *Proceedings of the 1st European Workshop on Hypercubes and Distributed Computers, Rennes*, pages 279–293. North Holland, 1989.

[7] Gilles Brassard, Peter Høyer, and Alain Tapp. *Automata, Languages and Programming*, volume 1443 of *Lecture Notes in Computer Science*, chapter Quantum counting. Springer, 1998.

[8] Dustin Alexander Cartwright, María Angélica Cueto, and Enrique Augusto Tobis. A characterization of maximum independent sets of de Bruijn graphs, 2009. `http://arxiv.org/abs/0905.3820`.

[9] Eduardo Cattani and Alicia Dickenstein. Counting solutions to binomial complete intersections. *Journal of Complexity*, 23(1):82–107, February 2007.

[10] V. Chandreskaran, M. Chertkov, D. Gamarnik, D. Shash, and J. Shin. Counting independent sets using the Bethe approximation, 2009. `http://www-math.mit.edu/~jinwoos/submit_bp.pdf`.

[11] Isaac L. Chuand and Michael Nielsen. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[12] CoCoATeam. CoCoA: a system for doing Computations in Commutative Algebra. Available at `http://cocoa.dima.unige.it`.

[13] David Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms.* Undergraduate Texts in Mathematics. Springer, second edition, 1997.

[14] David Cox, John Little, and Donal O'Shea. *Using Algebraic Geometry.* Number 185 in Graduate Texts in Mathematics. Springer, 1998.

[15] Bhaskar DasGupta, German Andres Enciso, Eduardo Sontag, and Yi Zhang. Algorithmic and complexity results for decompositions of biological networks into monotone subsystems. *Biosystems*, 90(1):161–178, July-August 2007.

[16] Nicolaas Govert de Bruijn. A combinatorial problem. *Koninklijke Nederlandse Akademie van Wetenschappen*, 49:758–764, 1946.

[17] Richard Dedekind. Über Zerlegungen von Zahlen durch ihre grössten gemeinsammen Teiler. In Robert Fricke, Emy Noether, and Öysten Ore, editors, *Gesammelte mathematische Werke*, volume 2, chapter XXVIII, pages 103–147. Friedrich Vieweg & Sohn AG, Braunschweig, 1931.

[18] Alicia Dickenstein and Enrique Augusto Tobis. Algebraic methods for counting antichains. In Jacob Scharcasnki and Vilmar Trevisan, editors, *Advances in Graph Theory and Applications.* UFRGS, Porto Alegre, 2007. ISBN: 85-88425-07-6.

[19] Alicia Dickenstein and Enrique Augusto Tobis. Additive edge labelings. *Discrete Applied Mathematics*, 2009. In press.

[20] David Eppstein. All maximal independent sets and dynamic dominance for sparse graphs. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 451–459. SIAM, 2005.

[21] Joseph A. Gallian. A dynamic survey of graph labeling, January 2009. Available at `http://www.combinatorics.org/Surveys/ds6.pdf`.

[22] Eleanor J. Gardiner and Peter Willett. Graph-theoretic techniques for macromolecular docking. *Journal of Chemical Information and Modeling*, 40(2):273–279, 2000.

[23] Ronald Lewis Graham and Neil James Alexander Sloane. On additive bases and harmonious graphs. *SIAM Journal of Algebraic and Discrete Methods*, 1(4):382–404, December 1980.

[24] Daniel R. Grayson and Michael E. Stillman. Macaulay 2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.

[25] Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann. SINGULAR 3.1.0 — A computer algebra system for polynomial computations. Available at http://www.singular.uni-kl.de, 2009.

[26] Michael R. Grey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[27] Jerrold W. Grossman, Devadatta M. Kulkarni, and Irwin E. Schochetman. On the minors of an incidence matrix and its Smith normal form. *Linear Algebra and its Applications*, 218(1–3):213–224, 1995.

[28] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.

[29] Ivan Gutman and Frank Harary. Generalizations of the matching polynomial. *Utilitas Mathematica*, 24:97–106, 1983.

[30] Frank Harary. *Graph Theory*. Addison-Wesley, 1969.

[31] Amir Hashemi. Polynomial complexity for Hilbert series of Borel type ideals. *Albanian Journal of Mathematics*, 1(3):145–155, September 2007.

[32] Jobst Heitzig and Jürgen Reinhold. The number of unlabeled orders on fourteen elements. *Order*, 17:333–341, 1999.

[33] Jürgen Herzog and Winfried Bruns. *Cohen-Macaulay Rings*, volume 39 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1993.

[34] Jürgen Herzog and Takayuki Hibi. Distributive lattices, bipartite graphs and Alexander duality. *Journal of Algebraic Combinatorics*, 22(3):289–302, November 2005.

[35] Mordechai Katzman. Counting monomials. *Journal of Algebraic Combinatorics*, 22(3):331–341, November 2005.

[36] Yosuke Kikuchi and Yukio Shibata. On the independent set of de Bruijn graphs. In *Topics in Applied and Theoretical Mathematics and Computer Science*, pages 117–128. WSEAS Press, 2001.

[37] Donald Ervin Knuth. *The Art of Computer Programming*, volume 4 Fascicle 0: Introduction to Combinatorial Algorithms and Boolean Functions. Addison-Wesley, 2008.

[38] Donald Ervin Knuth. *The Art of Computer Programming*, volume 4 Fascicle 1: Bitwise Tricks & Techniques. Binary Decision Diagrams. Addison-Wesley, 2009.

[39] Martin Kreuzer and Lorenzo Robbiano. *Computational Commutative Algebra 2*. Springer Verlag, Heidelberg, 2005.

[40] Serge Lang. *Algebra*. Number 211 in Graduate Texts in Mathematics. Springer, revised third edition, 2002.

[41] Sin-Min Lee, Edward F. Schmeichel, and Sze Chin Shee. On felicitous graphs. *Discrete Mathematics*, 93:201–209, 1991.

[42] Vadim E. Levit and Eugen Mandrescu. The independence polynomial of a graph — a survey. In *Proceedings of the 1st Internation Conference on Algebraic Informatics*, pages 233–254, Thessaloniki, 2005.

[43] Nicolas Lichiardopol. Independence number of de Bruijn graphs. *Discrete Mathematics*, 306(12):1145–1160, 2006.

[44] László Lovász. Stable sets and polynomials. *Discrete Mathematics*, 124:137–153, 1994.

[45] Ezra Miller and Bernd Sturmfels. *Combinatorial Commutative Algebra*, volume 277 of *Graduate Texts in Mathematics*. Springer, 2004.

[46] Ferdinando Mora and H. Michael Möller. The computation of the Hilbert function. In *EUROCAL 83*, number 162 in Lecture Notes in Computer Science, pages 157–167. Springer-Verlag, 1983.

[47] Biswanath Mukherjee. *Optical Communication Networks*. Series on Computer Communications. McGraw-Hill, New York, 1997.

[48] Pavel A. Pevzner, Haixu Tang, and Michael S. Waterman. An eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences*, 98(17):9748–53, August 2001.

[49] J. Scott Provan and Michael 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.

[50] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics.* Springer, 2003.

[51] Simone Severini. Universal quantum computation with unlabelled qubits. *Journal of Physics A: Mathematical and General*, 39(26):8507–8513, 2006.

[52] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[53] Aron Simis, Wolmer Vasconcelos, and Rafael Heraclio Villarreal Rodríguez. On the ideal theory of graphs. *Journal of Algebra*, 167(2):389–416, July 1994.

[54] Neil James Alexander Sloane. Challenge problems: Independent sets in graphs. `http://www2.research.att.com/~njas/doc/graphs.html`.

[55] Neil James Alexander Sloane. Sequence A052608 in The On-Line Encyclopedia of Integer Sequences. `http://www.research.att.com/~njas/sequences/index.html?q=A052608`.

[56] Henry J. Stephen Smith. On systems of linear indeterminate equations and congruences. *Philosophical Transactions*, 151:293–326, 1861.

[57] Richard P. Stanley. *Combinatorics and Commutative Algebra*, volume 41 of *Progress in Mathematics.* Birkhäuser Boston, Inc., Boston, MA, second edition, 1996.

[58] William Stein. Sage: open source mathematics. `http://www.sagemath.org/`.

[59] Bernd Sturmfels. *Gröbner Bases and Convex Polytopes*, volume 8 of *University Lecture Series.* American Mathematical Society, December 1995.

[60] Edgardo Ugalde. An alternative construction of normal numbers. *Journal de Théorie des Nombres de Bordeaux*, 12(1):165–177, 2000.

[61] Rafael Heraclio Villarreal Rodríguez. Rees algebras of edge ideals. *Communications in Algebra*, 23(9):3513–3524, 1995.

[62] Rafael Heraclio Villarreal Rodríguez. *Monomial Algebras*. Number 238 in Pure and Applied Mathematics. CRC, January 2001.

[63] Adam B. Yedidia. Counting independent sets and kernels of regular graphs, 2009. `http://arxiv.org/pdf/0910.4664v1`.

Index