UTRECHT UNIVERSITY

DEPARTMENT OF INFORMATION AND COMPUTING SCIENCES

MASTER'S THESIS

Divisorial gonality of graphs

Author:
Jelco M. Bodewes
ICA-3689719

Supervisors:
Hans L. Bodlaender
Gunther Cornelissen

A thesis submitted in partial fulfillment of the requirements for the degree of MSc

in

Computing Science

Utrecht University

November 16, 2017

Abstract

MSc

Divisorial gonality of graphs

by Jelco M. Bodewes
ICA-3689719

In this thesis the concept of divisorial gonality is explored. A triple of distinct definitions for divisorial gonality is presented and they are shown to be equivalent to each other. We explain a number of techniques for reasoning about divisorial gonality. We also consider several simple classes of graphs and calculate their divisorial gonality. We then present an upper bound on divisorial gonality based on minimum cuts between a set of vertices and a single other vertex in the graph.

The main result of this thesis is a set of reduction rules that can be used to recognize the graphs with divisorial gonality at most 2. We prove the rule set is both *safe* and *complete*, properties required for it to be usable. We also show that there exists a polynomial time algorithm based on this rule set.

Afterwards we answer three questions about the divisorial gonality of minors and subgraphs by making use of this set of reduction rules. These questions are then also answered for the closely related concepts of stable divisorial gonality and stable gonality.

Keywords: Chip-firing game, Divisor, Gonality, Parametrized complexity, Reduction rules

Acknowledgements

I would like to give my thanks to both of my supervisors Hans L. Bodlaender and Gunther Cornelissen. Their advice, both during the thesis, especially on what directions of research to pursue, and during the writing process of this thesis, has been very valuable. Working on the overlap between two research fields and under supervisors from two different departments has been an interesting and fun experience.

I would also like to thank my fellow student Marieke van der Wegen, who finished her thesis, also on gonality, some months before me. Our discussions led to many of the ideas and results in this thesis. Her feedback on the first version of this thesis was also much appreciated.

Finally I would like to thank my roommates, who suffered through occasional boring theoretical talks without too much complaint, but always were fun and relaxing to hang out with.

Contents

Chapter 1

Introduction

Graphs are a mathematical structure commonly used in both computer science and mathematics. Many problems can be modeled using graphs and algorithms on graphs can therefore often be used to solve a variety of problems. In many cases it turns out however that the more interesting problems are NP-hard for general graphs. One approach to still obtain usable algorithms has been through the field of parametrized complexity.

An important and commonly used parameter in this field is treewidth, which describes the complexity of a tree-decomposition of the graph. A recent survey on treewidth and its applications can be found in [5]. While treewidth has in many cases been successfully applied to create fast algorithms for graphs with low treewidth, there remain problems that are hard even for bounded treewidth. For this reason several other width-measures have been proposed, including amongst others branchwidth and cliquewidth; for an overview see [12].

Meanwhile in the seemingly unrelated field of algebraic curves there is the theory of divisors. It turns out that this field can actually be related to graphs: graphs can be seen as the discrete analogue of algebraic curves. Using this relation it is possible to apply ideas and theories from the theory of divisors to graphs. This was successfully done in [3] and has led to the current definition of divisorial gonality.

Divisorial gonality is a new measure that, in some sense, describes the complexity of a graph. Though it originated in theory from algebraic geometry, divisorial gonality can actually be described in terms of a chip-firing game. This often allows for intuitive reasoning about the divisorial gonality of graphs. At the same time it can be described more formally in the theory of divisors.

Compared to treewidth divisorial gonality has two main properties that we hope make it suitable to parametrize some hard problems: first, it is influenced by the complete structure of a multigraph, whereas treewidth is only influenced by the underlying simple graph; second, it is a finer measure than treewidth in the sense that it is lower bounded by treewidth, see [9], and there exist simple graphs of arbitrarily high divisorial gonality but with bounded treewidth, recently proven in [11].

As for the computability of divisorial gonality, there are already some interesting results. Calculating divisorial gonality for a general graph is NP-hard, a recent result from [10]. There exists an algorithm for calculating the divisorial gonality with running time $\mathcal{O}(n^4m^2n!)$ presented in [8] based on reduced divisors, a concept we will also use extensively in this thesis.

Interestingly, divisorial gonality is not the only new measure on graphs referred to as *gonality*. Two other variants are stable divisorial gonality and stable divisorial gonality. While stable divisorial gonality is based on ideas from [2], stable divisorial and stable gonality were both recently introduced in [6]. They are both based on refining of the graph: stable divisorial gonality still uses the same divisor based theory, but stable gonality is based on so called *finite harmonic morphisms*. While stable divisorial gonality and stable gonality will also both be defined in this thesis and we present a set of new results for both, the main focus will be on divisorial gonality.

The remainder of this thesis is split in several chapters. In Chapter 2 we introduce three different definitions of divisorial gonality and show they are equivalent. In addition we also introduce stable divisorial gonality and stable gonality. In Chapter 3 we explain several techniques useful for reasoning about divisorial gonality. The main result of this work is presented in Chapter 4 : a set of reduction rules to recognize graphs with divisorial gonality at most 2. In Chapter 5 new proofs for three questions answered in [11] are given. The same questions are then also answered for the cases of stable divisorial gonality and stable gonality. Finally in Chapter 6 we finish with a short conclusion and some interesting open problems.

New results of this thesis include the following:

- Theorem 3.3.1, a new upper bound for the divisorial gonality of a graph, based on minimum cuts.
- Several answers to questions about the stable and stable divisorial gonality of graphs, found in Theorems 5.2.1, 5.2.3, 5.3.1 and 5.3.2.
- The main new result is the set of reduction rules for recognizing graphs with divisorial gonality at most 2 and Theorem 4.2.1, stating that this set has the desired properties.

While working on this thesis the author collaborated and discussed often with fellow master student Marieke van der Wegen. While both theses are about gonality, this thesis is mostly focused on divisorial gonality, while her thesis focuses more on stable and stable divisorial gonality. For more information on these subjects we therefore recommend [15]. Much of the theory behind the reduction rule set is also joint work and the full results, including two sets of reduction rules for stable and stable divisorial gonality, can be found in [4].

Chapter 2

Preliminaries

In this chapter we introduce the main concept of this work, divisorial gonality, as well as two other variants of gonality that we will use in Chapter 5.

Both in this chapter and the rest of this thesis we will often refer to graphs. Unless otherwise mentioned these will be connected multi-graphs, allowing for multiple edges between any pair of vertices and loops from a vertex to itself. We will later see that loops do not actually have any influence on the divisorial gonality of a graph. For this reason graphs considered in the case of divisorial gonality will also be assumed to be loopless unless mentioned otherwise.

2.1 Divisorial gonality

The main subject of this work is divisorial gonality, but what exactly is divisorial gonality? On the most basic level the divisorial gonality of a graph is a natural number assigned to that graph, that attempts to describe how complex that graph is. A tree graph is the simplest graph in this measure. There are several different ways to define how to assign such a natural number to each graph.

In this section we start by giving three definitions of divisorial gonality. These are then shown to be equivalent to each other. The first definition is based on the idea of chip-firing games and most easily understood on a intuitive level. The second definition makes use of the concept of divisors and rank of divisors, and provides a nice formal notation. The final definition also makes use of divisors, but instead introduces the concept of *reaching* vertices. This turns out to often be useful when reasoning about the divisorial gonality of a graph.

We first consider the following chip-firing game that is played on a graph. It is similar to that of [3], but slightly different in the demand that any vertex can be reached with at least one chip.

Definition 2.1.1 (The chip-firing game)**.** The game is based on chips that move between vertices of the graph. A configuration is an assignment of an integer to each vertex in the graph that describes the number of chips on that vertex. Any vertex with a negative number of chips is called in debt. Firing is done by choosing a subset of the vertices, called the firing set, and then moving one chip along each outgoing edge out of the firing set.

A configuration is called a winning configuration, if for each vertex in the graph there exists a sequence of firings, such that the vertex ends with at least one chip on it and no other vertex ends in debt. The degree of a configuration is the sum of the integers assigned to each vertex or, in other words, the total number of chips on the graph.

We observe that if a configuration a can be turned into configuration b by firing a subset A, then b can be turned into a by firing A^c, the complement of A. This follows since the outgoing edges of A and its complement are the same. Note then that if configuration a can be turned into configuration b by firing a sequence of subsets $(A_1, \dots, A_k)$, then b can be turned into a by firing the sequence $(A_k^c, \dots, A_1^c)$. In other words, the transformations by firing are symmetric.

Another thing to note is that firing a subset has the same result as firing each vertex in that subset separately. When firing each vertex individually, any edge between two vertices in the firing set causes a chip to move one way and then back the other way. So only along edges with exactly one incident vertex in the firing set a chip is moved. The same would happen if we fired the entire firing set at once. For a similar reason loops have no influence on the chip-firing game, after all a chip moved along a loop simply stays on the same vertex.

Using this chip-firing game we can now define the divisorial gonality of any graph. The divisorial gonality of a graph is based on the following simple question: what is the minimum number of chips needed for a winning configuration?

Definition 2.1.2 (Divisorial gonality)**.** Let G be a graph. The divisorial gonality of G or $\mathrm{dgon}(G)$ is the lowest degree of all winning configurations on G for the chip-firing game of definition 2.1.1.

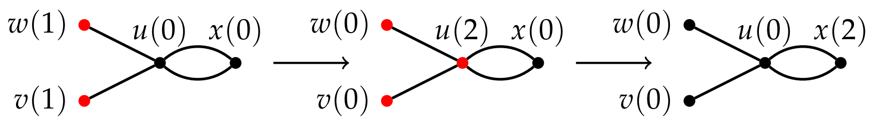

FIGURE 2.1: An example of the chip-firing game on a graph G

In Figure 2.1 an example of how the chip-firing game works is demonstrated. The numbers in brackets describe the number of chips on each vertex and the red vertices are those in the firing set. On the left there is an initial configuration with one chip on w and v. After firing the firing set $\{v, w\}$ the result is the middle configuration. From this configuration firing the firing set $\{v, w, u\}$ results in the rightmost configuration. Note that from the initial configuration we reached each vertex with a chip at least once, without having vertices go into debt, so the initial configuration is a winning configuration and in fact each of the configurations reached is. Since the configuration has a degree of 2 and, as we will see later, a non-tree graph can not have divisorial gonality 1, it follows this example graph has divisorial gonality 2.

While the definition given here is intuitive, it does not give a great basis to work with for reasoning and proofs on divisorial gonality. For that reason we now move on to the second, more formal, definition of divisorial gonality, based on the concepts of [3] and notation from [8].

Definition 2.1.3. Let G be a graph. A divisor D on G is an element of $\bigoplus_{V(G)} \mathbb{Z}$. We use $D(v)$ to denote the integer assigned to vertex v. We call a divisor D effective, denoted $D \geq 0$, if $D(v) \geq 0$ for all $v \in V(G)$. We denote the set of divisors on G by $\mathrm{Div}(G)$ and the set of effective divisors by $\mathrm{Div}_+(G)$. The degree $\deg(D)$ of a divisor is the sum over $D(v)$ for all $v \in V(G)$. By $\mathrm{Div}^k(G)$ we denote all divisors with degree k.

Definition 2.1.4. Let G be a graph with n vertices. The Laplacian matrix $\mathcal{L}$ is given by $\mathcal{L} = D - A$, where D is the diagonal matrix with $D_{v,v} = \deg(v)$ and A is the adjacency matrix of G.

We call a divisor P a principal divisor if there exists a divisor D such that $P = \mathcal{L}D$ and we denote the set of principal divisors by $\mathrm{Prin}(G)$.

Definition 2.1.5. We call two divisors D and D' equivalent or in notation $D \sim D'$, if there exists a principal divisor P such that $D' = D - P$. In this case we will also call $D' - D$ the transformation of D into D'. Given a divisor D, we have a class of equivalent effective divisors $|D| = \{D' \in \mathrm{Div}_+(G) \mid D \sim D'\}$.

While we will go more into the relation with definition 2.1.2 later, we want to note that the divisors introduced here are very similar to the configurations in the chip-firing game. Similarly the principal divisors here play the same role as the firing of subsets in the game.

Definition 2.1.6. The rank of a divisor D is denoted by $r(D)$ and defined as follows: $r(D) = \max\{k \mid |D - E| \neq \emptyset \quad \forall E \in \mathrm{Div}_+^k(G)\}$ if $|D| \neq \emptyset$ and $r(D) = -1$ if $|D| = \emptyset$.

Definition 2.1.7 (Divisorial gonality)**.** The divisorial gonality, dgon, of a graph G is the lowest degree for which there exists an effective divisor of rank greater or equal to one, ie. $\mathrm{dgon}(G) = \min\{\deg(D) \mid D \in \mathrm{Div}_+(G), r(D) \geq 1\}$.

Already the similarity of this definition with the earlier chip-firing game may be noticed. The concept of rank of a divisor is less intuitive though than that of the chip-firing game and also often is hard to reason about. For that reason we introduce one final definition of divisorial gonality. Also based on divisors, this definition asks what vertices in the graph can be *reached* by a divisor.

Definition 2.1.8. Let G be a graph, D a divisor on G and v a vertex of G. The divisor D *reaches* v if there exists an effective divisor D' on G with $D \sim D'$ and $D'(v) \geq 1$.

Definition 2.1.9 (Divisorial gonality)**.** The divisorial gonality, dgon, of a graph G is the lowest degree for which there exists an effective divisor that reaches each vertex of G.

As an example we again consider the graph G in Figure 2.1. To denote the divisors and Laplacian of G we use the ordering (w, v, u, x) here. The three configurations displayed in the figure then correspond with the following divisors:

$$c_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, c_2 = \begin{pmatrix} 0 \\ 0 \\ 2 \\ 0 \end{pmatrix}, c_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \end{pmatrix}$$

The Laplacian of G is:

$$\mathcal{L} = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ -1 & -1 & 4 & -2 \\ 0 & 0 & -2 & 2 \end{pmatrix}$$

Then by taking the divisors

$$f_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, f_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, f_3 = \begin{pmatrix} 2 \\ 2 \\ 1 \\ 0 \end{pmatrix}$$

and applying the Laplacian to them it can be seen that the differences $c_1 - c_2$, $c_2 - c_3$ and $c_1 - c_3$ are given by $\mathcal{L}f_1$, $\mathcal{L}f_2$ and $\mathcal{L}f_3$ respectively, making them principal divisors. From this it follows the divisors c_1, c_2 and c_3 are equivalent to each other. Note that f_3 is simply the sum of f_1 and f_2. It makes sense then that is also a principal divisor, after all the Laplacian is a linear map, so any sum of principal divisors must also be a principal divisor.

So there are three equivalent effective divisors and, for each vertex in the graph, one of them assigns at least one chip to that vertex. It follows c_1 reaches all vertices in the graph, and so $\text{dgon}(G) \leq 2$.

Clearly this definition of divisorial gonality is quite similar to that of Definition 2.1.7. In fact it is not very hard to show that the two notions are equivalent:

Theorem 2.1.1. *The Definitions 2.1.7 and 2.1.9 of divisorial gonality are equivalent.*

Proof. Let G be a graph and denote its divisorial gonality by Definition 2.1.7 as $\text{dgon}_{rank}(G)$ and its divisorial gonality by Definition 2.1.9 as $\text{dgon}_{reach}(G)$. We first show $\text{dgon}_{rank}(G) \leq \text{dgon}_{reach}(G)$. Let D be a divisor of degree $\text{dgon}_{reach}(G)$ that reaches each vertex in G. We claim $\text{rank}(D) \geq 1$.

Let E be a divisor in $\text{Div}_+^1(G)$. Note that this divisor assigns 1 to exactly one vertex v and 0 to all other vertices. Since D reaches all vertices in G, D reaches v. Let D_v be an equivalent effective divisor with $D_v(v) \geq 1$. Then $D_v - E$ is an effective divisor, since $D_v(v) \geq 1$ and D_v is effective, so it follows $|D - E| \neq \emptyset$. Since this holds for any $E \in \text{Div}_+^1(G)$, it follows that $\text{rank}(D) \geq 1$. We conclude $\text{dgon}_{rank}(G) \leq \text{dgon}_{reach}(G)$.

Now to show $\text{dgon}_{reach}(G) \leq \text{dgon}_{rank}(G)$: let D be an effective divisor with $\text{rank}(D) \geq 1$ and $\deg(D) = \text{dgon}_{rank}(G)$. We claim D reaches every vertex of G. Let $v \in V(G)$ and E the divisor with $E(v) = 1$ and $E(w) = 0$ for $w \neq v$. Since $\text{rank}(D) \geq 1$, we know $|D - E| \neq \emptyset$. Choose a divisor $D_E \in |D - E|$ and let $D' = D_E + E$. Then D' is an effective divisor equivalent to D such that $D' - E$ is effective, but from this it follows $D'(v) \geq 1$. This exactly means that D reaches v and since this works for every $v \in V(G)$, we know that D reaches every vertex of G. We conclude $\text{dgon}_{reach}(G) \leq \text{dgon}_{rank}(G)$ and finally combined with the previous result get $\text{dgon}_{reach}(G) = \text{dgon}_{rank}(G)$. □

Not only does the above proof show that the two definitions are equivalent, it even shows that the same divisors of lowest degree work for both definitions. This fact will be used often in proofs.

Corollary 2.1.2. *Let G be a graph and D a divisor. Then* $\text{rank}(D) \geq 1$ *if and only if D reaches all vertices of G.*

Proof. Follows from the proof of Theorem 2.1.1. □

Moving back to the original chip-firing game, note that the divisors introduced here are simply a more formal way of describing the configurations from the chip firing game of Definition 2.1.1. Similarly the Laplacian matrix describes how chips can be moved by firing certain subsets.

If G is a graph and $v \in V(G)$, consider the row of the Laplacian corresponding to v. This row in fact describes the change in chip totals that follows from firing the subset $V(G) - \{v\}$ in the chip-firing game. But the result of firing the subset $V(G) - \{v\}$ is simply the reverse of firing its complement $\{v\}$, in terms of the resulting divisor this is represented by the integer assigned to each vertex being multiplied by -1. So if we have a divisor P, it follows that $\mathcal{L}P$ gives the amount of chips each vertex loses, by the firing of each vertex $v \in V(G)$ exactly $P(V)$ times.

In other words the image of the Laplacian matrix, the set of principal divisors, is exactly the set of divisors that describe a change of chips that can be achieved by a finite sequence of firings in the chip-firing game. This leads to the following:

Lemma 2.1.3. *Let D and D' be two divisors on a graph G. Then the configurations corresponding with D and D' in the chip-firing game can be turned into each other by firings exactly when $D \sim D'$.*

Proof. If the configuration corresponding to D can be turned into that of D', there exists a sequence of subset firings that has the difference in their chip assignments as a result. This difference is exactly $D' - D$. If we take the divisor f that describes how often each vertex is fired in this sequence, then $\mathcal{L}f$ results exactly in this difference times -1. So $D' - D = -\mathcal{L}f$ and it follows $D \sim D'$.

Assuming on the other hand that $D \sim D'$, there must then exist a divisor f such that $D' - D = -\mathcal{L}f$. But now f exactly describes how often each vertex should be fired to transform the configuration corresponding to D into that of D'. We conclude the divisors are equivalent if and only if their corresponding configurations are related by a sequence of firings. $\square$

Now that we know that equivalence of divisors and the firing rules of our chip-firing game describe the same relation, we can show the Definitions 2.1.2 and 2.1.9 are the same:

Theorem 2.1.4. *The Definitions 2.1.2 and 2.1.9 of divisorial gonality are equivalent.*

Proof. Let G be a graph, $\text{dgon}_{chip}(G)$ its divisorial gonality under Definition 2.1.2 and $\text{dgon}_{reach}(G)$ its divisorial gonality under Definition 2.1.9. Then there must exist a winning configuration for the chip-firing game of degree $\text{dgon}_{chip}(G)$. Let W then be the divisor describing this configuration and let W_v describe the configuration with at least one chip on v and no vertices in debt reachable from W for a $v \in V(G)$. By Lemma 2.1.3 we know that $W \sim W_v$, and, since this holds for every $v \in V(G)$, conclude that W reaches each vertex in G. So $\text{dgon}_{reach}(G) \leq \text{dgon}_{chip}(G)$.

There also must exist an effective divisor D with $\deg(D) = \text{dgon}_{reach}(G)$ that reaches each vertex in v. So if $v \in V(G)$ we know there exists an effective divisor D_v with $D_v(v) \geq 1$ and $D \sim D_v$. But note now that the configuration in the chip-firing game described by D is a winning configuration, the divisors D_v describe the configurations with at least one chip on v that can be reached by firings. We conclude $\text{dgon}_{chip}(G) \leq \text{dgon}_{reach}(G)$ and finally $\text{dgon}_{chip}(G) = \text{dgon}_{reach}(G)$. $\square$

Not only are the definitions equivalent, as in Corollary 2.1.2 we also see in the proof that a configuration is a winning configuration exactly when the divisor describing it reaches all vertices:

Corollary 2.1.5. *Let G be a graph and D a divisor on G. D reaches all vertices of G exactly when the configuration represented by D is a winning configuration in the chip firing game of Definition 2.1.1.*

Proof. See the proof of Theorem 2.1.4. $\square$

So we now have three equivalent definitions of divisorial gonality, each with their own advantages. In many cases a combination of the definitions is used during proofs, in these situations the Lemma 2.1.3 and corollaries 2.1.2, 2.1.5 are especially useful. For this reason it will be convenient to be able to discuss firing sets directly on divisors.

Definition 2.1.10. *Firing* a firing set A, $A \subset V(G)$, on a divisor D on a graph G, results in the divisor $D' = D - \mathcal{L}I_A$, where $I_A(v) = 1$ if $v \in A$ and $I_A(v) = 0$ otherwise.

Note that this is the same as the firing set being fired in the chip-firing game on the configuration represented by D and then representing the resulting configuration by D'. It also will be useful to be able to discuss the impact a given firing set has on the divisor and specifically if it keeps all vertices out of debt.

Definition 2.1.11. Given a graph G and a firing set A, we define the *out degree* of a vertex $v \in A$ under A as follows: $\text{outdeg}_A(v) = \mathcal{L}I_A$, where $I_A(v) = 1$ if $v \in A$ and $I_A(v) = 0$ otherwise.

Definition 2.1.12. Given a graph G and a divisor D, we call a non-empty firing set A *valid* if $D(a) \geq \text{outdeg}_A(a)$ for all $a \in A$.

The out degree of a vertex v for a firing set A tells us how many chips v will lose by the firing of A. A firing set A then is valid if no vertex will go into debt by firing it (each vertex loses at most the number of chips it had). The final notation we introduce here will be useful when discussing the divisorial gonality of minors of graphs:

Definition 2.1.13. Let G be a graph and H a minor of G with $V(H) \subset V(G)$. Let D be a divisor on G and D' a divisor on H. The *restriction* of D to H is the divisor D_H on H with $D_H(v) = D(v)$ for all $v \in H$. On the other hand the *expansion* of D' to G is the divisor D'_G with $D'_G(v) = D'(v)$ for $v \in H$ and $D'_G(v) = 0$ for $v \notin H$.

2.2 Stable divisorial gonality

In this section we introduce stable divisorial gonality as a variant on divisorial gonality. The idea is that by adding the possibility of *refinements* stable divisorial gonality better describes the global structure of a graph. The concept was proposed in the appendix of [6]. We start by introducing refinements and then use these to define stable divisorial gonality:

Definition 2.2.1. Let G be a graph. A refinement G' of G is a graph created from G by a finite number of applications of the following two operations:

1. Attaching a new leaf to any vertex in the graph.
2. Subdividing an edge by adding a vertex in its middle.

Definition 2.2.2. Let G and H be graphs, such that H is a refinement of G. The *original vertices* of H are the vertices not created by refinements, i.e. $\{w \in V(H) | w \in V(G)\}$.

Definition 2.2.3. Let G be a graph. The stable divisorial gonality of G is the minimum of the divisorial gonality over all refinements of G, i.e. $\text{sdgon}(G) = \min\{\text{dgon}(G') | G' \text{ a refinement of } G\}$.

Since any graph is a refinement of itself it can quickly be seen that the divisorial gonality of a graph is an upper bound for its stable divisorial gonality. In addition, while the definition of refinements allows for two operations, it turns out only the second operation is required to obtain a refinement with minimum divisorial gonality.

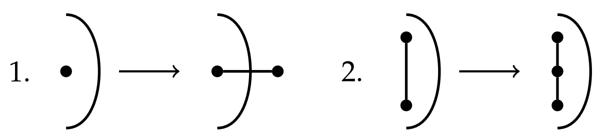

FIGURE 2.2: The two operations for the refinements of graphs.

Theorem 2.2.1. *Let G be a graph. There exists a refinement G' of G created only by subdividing edges such that* $\text{dgon}(G') = \text{sdgon}(G)$.

This theorem will be proven in Section 3.4 when the required techniques have been introduced. The result is quite useful though, since it significantly reduces the complexity of refinements that need to be considered for stable divisorial gonality. For this reason, when talking about stable divisorial gonality, we will only consider refinements created by subdividing edges.

2.3 Stable gonality

The final variant of gonality we introduce here is stable gonality. Its definition differs significantly from that of divisorial gonality, though it bears some similarity to stable divisorial gonality in the use of refinements. While this variant does not play a major role in this work, we give here a definition for the sake of several results in Chapter 5.

The definition given here has been taken from [4], while the concept was originally proposed in [6]. To start, we first need to define a class of functions between graphs, known as *finite harmonic morphisms*:

Definition 2.3.1. Let G and H be graphs. A *finite morphism* is a map $\phi : G \to H$ such that

(i) $\phi(V(G)) \subseteq V(H)$,

(ii) $\phi(uv) = \phi(u)\phi(v)$ for all $uv \in E(G)$,

together with, for every $e \in E(G)$, an "index" $r_\phi(e) \in \mathbf{N}$.

Definition 2.3.2. We call a finite morphism $\phi : G \to H$ *harmonic* if for every $v \in V(G)$ it holds that for all $e, e' \in E_{\phi(v)}(H)$

$$\sum_{d \in E_v(G), \phi(d)=e} r_\phi(d) = \sum_{d' \in E_v(G), \phi(d')=e'} r_\phi(d').$$

We write $m_\phi(v)$ for this sum.

Definition 2.3.3. The *degree* of a finite harmonic morphism $\phi : G \to H$ is

$$\sum_{d \in E(G), \phi(d)=e} r_\phi(e) = \sum_{u \in V(G), \phi(u)=v} m_\phi(u),$$

for $e \in E(H)$, $v \in V(H)$. This is independent of the choice of e or v ([3], Lemma 2.4).

Combining these finite harmonic morphisms with the definition as given for stable divisorial gonality gives the following definition of stable gonality. Here the leaf-adding operation of refinements can actually be necessary.

Definition 2.3.4. The *stable gonality* of a graph G is

$$\begin{aligned}\operatorname{sgon}(G) = \min\{\deg(\phi) \mid \phi : G' \to T \text{ a finite harmonic morphism,}\\ G' \text{ a refinement of } G, T \text{ a tree}\}.\end{aligned}$$

Chapter 3

Several techniques and results for gonality

We start this chapter by discussing some example graphs and their gonality. An obvious family of graphs to start with is that of the tree graphs and in fact tree graphs turn out to be the simplest graphs for gonality. Both of the following results will be proven later in section 3.4:

Theorem 3.0.1. *Let G be a graph. The divisorial gonality of G is 1 if and only if G is a tree. The same holds for stable divisorial gonality.*

While tree graphs are the simplest graphs in terms of divisorial gonality, the complete graph turns out to be the simple (not multi-) graph with highest divisorial gonality for its number of vertices:

Theorem 3.0.2. *The complete graph K_n with n vertices has divisorial gonality $n-1$.*

As mentioned in the introduction, treewidth is a commonly used measure of complexity of graphs. When considering a new measure such as divisorial gonality it is therefore interesting to check if it relates to treewidth. In 2014 Dobben de Bruyn and Gijswijt showed in [9] that such a relation exists and it also holds for stable divisorial gonality and stable gonality:

Theorem 3.0.3 ([9])**.** *For any connected graph G it holds that $\mathrm{dgon}(G) \geq \mathrm{tw}(G)$. In addition $\mathrm{sdgon}(G) \geq \mathrm{tw}(G)$ and $\mathrm{sgon}(G) \geq \mathrm{tw}(G)$.*

Let us consider now one additional example class of graphs with this theorem in mind: the grid graphs. Suppose we have a $n \times m$ grid graph G, what is the divisorial gonality of G? It turns out that by using the treewidth of the graph, which is known to be $\min(m, n)$, it becomes fairly easy to establish the divisorial gonality of G. This result has been proven previously by a different method in [8].

Theorem 3.0.4. *The $n \times m$ grid graph has divisorial gonality $\min(n, m)$.*

Proof. Let G be the $m \times n$ grid graph. We assume without loss of generality that $m \leq n$. Note that the vertices of G can be split into n columns of m vertices each, we will call these columns $(C_1, \ldots, C_k)$, ordering them by adjacency. We claim the divisor D_1 with one chip on each vertex of C_1 and no chips elsewhere reaches each vertex of G. As an example of D_1 see Figure 3.1, where the red vertices are those in C_1.

Consider the firing set A_1 containing exactly the vertices of C_1, the outgoing edges of this firing set connect each vertex of C_1 to an adjacent vertex of C_2. Firing this firing set on D_1 therefore results in a divisor D_2 with one chip one each vertex in C_2 and no chips elsewhere. Similarly we can create the firing set A_2 consisting of the vertices in C_1 and C_2, which, when fired, moves a chip from each vertex in C_2 to an adjacent vertex in C_3. By repeating this process we can create a series of equivalent divisors $(D_1, \dots, D_k)$, where D_i has a chip on each vertex in C_i and nowhere else. Since the columns $(C_1, \dots, C_k)$ together cover all vertices of G, it follows D_1 reaches all vertices of G.

So we know now that $\mathrm{dgon}(G) \leq m$, since $\deg(D_1) = m$. But since the treewidth of G is $\min(m, n) = m$, by Theorem 3.0.3 we also know $\mathrm{dgon}(G) \geq m$. We conclude $\mathrm{dgon}(G) = m$. □

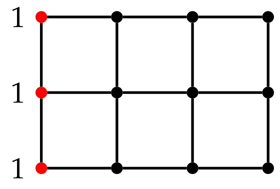

FIGURE 3.1: An example of the divisor D_1 for a grid graph

3.1 Level set decomposition

In this section we introduce a technique to be used in proofs on divisorial and stable divisorial gonality. It can be used to reason about transformations between equivalent divisors. We use the definition from [8].

We recall that two divisors D and D' are defined as equivalent if there exists a principal divisor P such that $D' = D - P$. A principal divisor then is a divisor that can be expressed as an image of the Laplacian matrix, in other words there is a divisor f, such that $P = \mathcal{L}f$. Intuitively this can be understood as follows: $-P$ is a vector over all vertices that for each vertex describes the change in number of chips. Then f is the vector that describes for each vertex how often it should be fired, in the terms of the chip-firing game, to produce the result $-P$.

Therefore we have some divisor f that describes how often vertices should be fired to produce D' from D, $D' = D - \mathcal{L}f$. We now consider the idea of splitting this divisor f into layers, where each layer gives a subset of the graph that should be fired once. We shall see that with the right definition these layers or level sets have certain helpful properties. The level sets are given by the following definition:

Definition 3.1.1. Let D and D' be two equivalent divisors, then there must exist a divisor f such that $D' = D - \mathcal{L}f$, where $\mathcal{L}$ is the Laplacian matrix. Let $m = \max\{f(v) \mid v \in V(G)\}$ and let $k = m - \min\{f(v) : v \in V(G)\}$. We then define the *level sets* as follows:

$$A_i = \{v \in V(G) : f(v) \geq m - i\} \quad \text{for } i \in \{0, \dots, k\}.$$

Note that while there are multiple such f, these can only differ by a multiple of the all ones vector and they therefore result in the same level set decomposition. Because of this we will often refer to the level set decomposition of a transformation of D into D', without mentioning the divisor f.

We can then also consider the sequence of divisors that is created by firing each level set in the given order:

Definition 3.1.2. A level set decomposition $A_0, \dots, A_k$ belonging to a transformation of D into D' produces a *sequence of divisors* as follows:

$$\begin{aligned} D_0 &= D, \\ D_{i+1} &= D_i - \mathcal{L} I_{A_i} \quad \text{for } i \in \{0, \dots, k-1\}. \end{aligned}$$

Here, I_{A_i} is the divisor with $I_{A_i}(v) = 1$ for $v \in A_i$ and $I_{A_i}(v) = 0$ otherwise.

Remark. *Note that for a level set decomposition $A_0, \dots, A_k$ we have that $A_0 \subseteq A_1 \subseteq \dots \subseteq A_k$.*

We continue with an useful property of level set decompositions, namely the fact that each divisor in the associated sequence is lower bounded by the pointwise minimum of D and D':

Theorem 3.1.1. *Let D and D' be two equivalent divisors and $A_0, \dots, A_k$ the level set decomposition of the transformation of D into D'. Let $D_0, \dots, D_k$ be the associated sequence of divisors. We then have that:*

$$D_i(v) \geq \min(D(v), D'(v)) \quad \forall i \in \{0, \dots, k\},\ \forall v \in V(G).$$

Proof. Choose an $i \in \{0, \dots, k\}$ and a vertex $v \in V(G)$. If $D_i(v) \geq D(v)$ we are done, so assume that $D_i(v) < D(v)$. Since $D_i(v) < D(v)$, v must have been fired at least once before D_i (the only way a vertex can loose chips is by firing), so there is an A_j with $v \in A_j$ and $j < i$. But since $A_0 \subseteq \dots \subseteq A_k$, we then have that $v \in A_m$ for all $m \geq j$ and specifically for all $m \geq i$.

So v is fired in every subset starting from A_i, but the number of chips on v cannot increase if v is part of the fired subset, so we have that $D_i(v) \geq D_m(v)$ for all $m \geq i$ and specifically, $D_i(v) \geq D_k(v) = D'(v)$.

Therefore either $D_i(v) \geq D(v)$ or $D_i(v) \geq D'(v)$ for all $v \in V(G)$ and $i \in \{0, \dots, k\}$. We conclude that $D_i(v) \geq \min(D(v), D'(v))$ for all $v \in V(G)$ and $i \in \{0, \dots, k\}$. □

Theorem 3.1.1 gives a nice result for the transformation between two equivalent effective divisors:

Corollary 3.1.2. *Let D and D' be two equivalent effective divisors. Let $A_0, \dots, A_k$ be the level set decomposition of the transformation of D into D' and $D_0, \dots, D_k$ the associated sequence of divisors. Then all divisors D_i are effective and all firing subsets A_i are valid with respect to D_i.*

Proof. This follows simply from Theorem 3.1.1: since $D_i(v) \geq \min(D(v), D'(v))$, $D(v) \geq 0$ and $D'(v) \geq 0$, we conclude that $D_i(v) \geq 0$ and so D_i is effective. To see that all firing sets are valid, note that a non-valid firing set would result in a non-effective divisor. □

3.2 Reduced divisors

When working with divisors it will often be useful to consider the equivalent divisor that moves chips 'as much as possible' towards a given vertex. This somewhat vague idea can be formalized in the following definition of *reduced divisors*. These were originally introduced in [3], though some notation used here was introduced in [8].
s

Definition 3.2.1. A divisor D on a graph G is called *v-reduced*, for $v \in V(G)$, if the following two conditions hold:

- $D(w) \geq 0 \; \forall w \in V(G) - \{v\}$
- If A is a valid firing set on D, then $v \in A$.

Any divisor for which only the first condition holds is called *v-semireduced*.

These reduced divisors are only really useful if any divisor can be turned into one. Luckily, it turns out this is possible and in fact any divisor is equivalent to a single unique v-reduced divisor, for any vertex v in the graph. The following proof is based on that in [3] and the proof in [8].

Theorem 3.2.1 (Proposition 3.1 in [3])**.** *Let G be a graph and D a divisor on G. For any $v \in V(G)$ there exists an unique v-reduced divisor D_v equivalent to D.*

For the proof of this theorem we require the following lemma:

Lemma 3.2.2. *Let G be a graph and D a divisor on G. For any $v \in V(G)$ there exists a v-semireduced divisor D' with $D' \sim D$.*

Proof. We assume D itself is not v-semireduced, otherwise $D' = D$ suffices. Let $l_v(w)$ be the length of the shortest path between any vertex w and v. Let $d = \max\{l_v(w) : w \in V(G)\}$ and $S_i = \{w \in V(G) | l_v(w) = i\}$ for $i \in [0, \dots, d]$. Note what happens if we fire the firing set $C_k = \{w \in V(G) | l_v(w) \leq k\}$ for $k \in \mathbb{N}$: clearly every vertex at distance $k+1$ from v is adjacent to at least one vertex in C_k and itself not in C_k, thus it receives at least one chip when C_k is fired. Meanwhile only vertices within at most k distance of v lose chips by the firing of C_k.

Starting with divisor D, by firing the firing set C_{d-1} enough times we can ensure that all vertices in S_d are out of debt. The number of firings required for this is at most the largest debt among vertices in S_d. Then by firing the set C_{d-2} enough times we can do the same for the vertices of S_{d-1} without affecting the number of chips on any vertices farther away. By repeating this process with firing sets $(C_{d-3}, \dots, C_0)$ we can ensure no vertex with distance at least 1 from v is in debt. Since the only vertex at distance 0 from v is v itself, we conclude the final divisor D' is v-semireduced, and since we achieved it by a sequence of firings starting in D, $D' \sim D$. □

We continue with the proof of the theorem:

Proof of Theorem 3.2.1. We start by proving that for every vertex v and divisor D there exists an equivalent v-reduced divisor D_v. Note that by Lemma 3.2.2 there exists at least one v-semireduced divisor D' equivalent to D. Let $l_v(w)$ be the length of the shortest path between any vertex w and v. Let $d = \max\{l_v(w) : w \in V(G)\}$ and $S_i = \{w \in V(G) | l_v(w) = i\}$ for $i \in [0, \dots, d]$. We introduce the following function on divisors:

$$\beta(D) = \Big(\sum_{v \in S_0} D(v), \sum_{v \in S_1} D(v), \dots, \sum_{v \in S_d} D(v)\Big)$$

Let now F be a v-semireduced divisor equivalent to D that maximizes the function $\beta(F)$ with regards to the lexicographical ordering. We claim F is actually v-reduced.

Suppose it is not, then, since F is v-semireduced, the second condition must be violated. So there exists a valid firing set A on F, with $v \notin A$. But consider the divisor F' created by firing A from F. Let $w \in A$ be any vertex in A with minimal $l_v(w)$ amongst the vertices of A. Note that $v \notin A$, so $l_v(w) \geq 1$. Then w must have at least one neighbor w' with $l_v(w') < l_v(w)$ and we know $w' \notin A$, since then w would not be at minimal distance.

But if $w \in A$ and $w' \notin A$, it follows the firing of A moves at least one chip from w to w'. And since no vertex at distance smaller than $l_v(w)$ can be in A, this implies $\beta(F') > \beta(F)$ with regards to the lexicographical ordering. But, because A was valid on F, we know that F' is still v-semireduced. But now we have another v-semireduced divisor F' equivalent to D, but with $\beta(F') > \beta(F)$, while we chose F to maximize β. By this contradiction we conclude that no such valid A can exist, and thus F is v-reduced.

It remains to show that this equivalent v-reduced divisor is unique for a given D and v. Suppose instead that there are two distinct v-reduced divisors D' and D'', both equivalent to D. Then we also know $D' \sim D''$. Let then $(A_0, \ldots, A_k)$ be the level set decomposition of the transformation from D' to D''.

Note that by Lemma 3.1.2 we know A_0 is valid, so, because D' is v-reduced it follows $v \in A_0$ and so $v \in A_i$, for all $i \in [0, \ldots, k]$. Since $D' \neq D''$, $k > 1$ and $A_{k-1} \neq V(G)$. Let D'_{k-1} be the divisor that is turned into D'' by firing A_{k-1}. By Lemma 3.1.1 we know that for all $w \neq v$, $D'_{k-1}(w) \geq 0$, since D' and D'' are both v-reduced.

Consider the firing set $V(G) - A_{k-1}$ on D''. Firing it results in the divisor D'_{k-1}. Since $v \in A_{k-1}$ it is clear that $v \notin A_{k-1}$. Then because $D_{k-1}(w) \geq 0$ for all $w \in V(G)$ $w \neq v$, it follows $V(G) - A_{k-1}$ is a valid firing set on D'' not containing v. This contradicts D' being v-reduced and we conclude no such two distinct divisors D' and D'' can exist and the v-reduced divisor is unique.

□

For this reason, if we specify the divisor D and the vertex v, it is possible to talk about the v-reduced divisor of D. The fact that this reduced divisor is unique and representative for a class of equivalent divisors is already quite useful, but the following result from [8] shows that reduced divisors are also related to the rank of a divisor:

Theorem 3.2.3. *Let D be a divisor on a graph G.* $\operatorname{rank}(D) \geq 1$ *if and only if $D_v(v) \geq 1$ for every $v \in V(G)$, where D_v is the v-reduced divisor equivalent to D.*

Proof. Let D be a divisor on a graph G. Suppose that $\operatorname{rank}(D) \geq 1$. Let v be a vertex in G and let D_v be the v-reduced divisor equivalent to D. Since $\operatorname{rank}(D) \geq 1$ we know that D is equivalent to an effective divisor D' with $D'(v) \geq 1$. Because $D \sim D_v$ and $D \sim D'$ it follows $D_v \sim D'$. Consider then the transformation of D_v to D' and its level set decomposition $(A_0, \ldots, A_k)$. By Corollary 3.1.2 we know A_0 must be valid on D_v, but since D_v is v-reduced that implies that $v \in A_0$. This means $v \in A_i$, for all $i \in [0, \ldots, k]$, from which it follows that $D_v(v) \geq D'(v)$, because v can never increase its chips when it is in the firing set. Since we know that $D'(v) \geq 1$, we conclude $D_v(v) \geq 1$ and this holds for all $v \in V(G)$.

Suppose now on the other hand that $D_v(v) \geq 1$ for all $v \in V(G)$. We fix one such $v \in V(G)$ and show that D reaches v. We know $D_v \sim D$, and also that D_v is effective everywhere except v by the definition of a reduced divisor. But we also know that $D_v(v) \geq 1$. So D is equivalent to an effective divisor with at least one chip on v and so reaches v. Since this holds for all $v \in V(G)$, with Corollary 2.1.2, we conclude $\operatorname{rank}(D) \geq 1$. □

3.3 Lower bound by minimal cuts

In this section a lower bound on the divisorial gonality of a given graph is proven. This lower bound is based on the idea that the number of chips fired out of a firing set is equal or greater to the minimal cut between any vertex in the fired subset and any vertex outside it. Based on this idea, the following lower bound can be created:

Theorem 3.3.1. *Let G be a connected graph. Let $v \in V(G)$ and $W \subset V(G)$, with $v \notin W$. If W contains at least $n-1$ elements, such that for each $w \in W$ the minimal cut between v and w is at least n, then $\mathrm{dgon}(G) \geq n$.*

Proof. We assume instead that the divisorial gonality of G is lower than n. Then there must exist an effective divisor D with $deg(D) = n-1$ and $r(D) \geq 1$. We now introduce a function that counts the number of elements of W with at least one chip on them for a given divisor:

$$\varphi(F) = |\{w \in W | F(w) \geq 1\}|$$

We note that since $r(D) \geq 1$, there must be at least one equivalent effective divisor that assigns one or more chips to v. We choose such a divisor that maximizes our function φ and name it D'. So $D' \geq 0$, $D' \sim D$, $D'(v) \geq 1$ and $\varphi(D') = max\{\varphi(F) : F \in Div(G), F \sim D, F \geq 0, F(v) \geq 1\}$.

Since $deg(D') = deg(D) = n-1$ and $D'(v) \geq 1$ it follows $\varphi(D') < n-1$. Because $|W| \geq n-1$ there exists a vertex $w' \in W$ with $D'(w') = 0$. However from $r(D) \geq 1$ we know that $r(D') \geq 1$, so there exists an effective divisor D'' with $D''(w') \geq 1$ and $D'' \sim D'$.

Given that $D' \sim D''$, there must be some divisor f such that $D'' = D' - Qf$. We now consider the level set decomposition $A_0, \ldots, A_k$ corresponding with this f and the associated sequence of divisors $D_0, \ldots, D_k$. By Corollary 3.1.2 we know that all the divisors D_i are effective. This implies that $outdeg(A_i) \leq D_i(A_i)$, where we use the notation $outdeg(A_i) = \sum_{a \in A_i} outdeg_{A_i}(a)$ and $D_i(A_i) = \sum_{a \in A_i} D_i(a)$.

Observe now that the total outdegree of a subset is equal to the weight of the cut between that subset and its complement. First consider now the case that $v \in A_i$ for some $A_i \subseteq V(G)$. If there is a $w \in W$ with $w \notin A_i$ then clearly A_i defines a cut between v and w. We know however that the minimum cut between v and w is at least n. So it follows that $outdeg(A_i) \geq n$.

Since $deg(D_i) = n-1$ for all i, $D_i(A_i) \leq n-1$ for all i. Combining this with $outdeg(A_i) \leq D_i(A_i)$, we conclude the following for all A_i: if $v \in A_i$ then $W \subset A_i$. Since the outdegree of any subset and its complement are the same, it also follows that if $v \notin A_i$ then $W \cap A_i = \emptyset$.

We move back to considering the level set decomposition $A_1, \ldots, A_k$ from D' to D''. Since $D''(w') > D'(w')$ it follows that there must exist an integer j such that $D_j(w') > D_{j-1}(w')$. This implies that $w' \notin A_{j-1}$, but by our result above then $v \notin A_{j-1}$ and $W \cap A_{j-1} = \emptyset$. Therefore for all $w \in W/\{w'\}$ it follows that $D_j(w) \geq D_{j-1}(w)$, combining this with $D_j(w') > D_{j-1}(w')$ we get that $\varphi(D_j) > \varphi(D_{j-1})$.

But since $W \cap A_{j-1} = \emptyset$, it follows that $W \cap A_h = \emptyset \; \forall h \leq j-1$. If no vertex in W fires until at least subset A_{j-1} it follows that $\varphi(D_{j-1}) \geq \varphi(D')$ and thus $\varphi(D_j) > \varphi(D')$.

Note however that D_j fulfills all conditions over which we maximized φ ($D_j \sim D$, $D_j \geq 0$ and $D_j(v) \geq 1$), yet it has higher value under φ than D'. From this contradiction we conclude that the divisorial gonality of G must be at least n.

□

3.4 Proofs of earlier claims

Earlier in this chapter we claimed that a graph has divisorial gonality 1 exactly when it is a tree and that the same holds for stable divisorial gonality. Now that we have the technique of level set decompositions we can prove this:

Proof of Theorem 3.0.1. We start by considering divisorial gonality. First let G be a tree and let v be any vertex in G. We claim the divisor D with one chip on v and no other chips reaches all vertices of G. Let w be any vertex in G with $w \neq v$. Note that there exists an unique path P from v to w. Let v_1 be the first vertex on this path from v, so v_1 is adjacent to v. Let G_v be the subtree attached to v_1 that contains v. If we fire the subset G_v from D, note the only outgoing edge is the one between v and v_1, so the one chip of D moves to v_1. By repeating this construction, we can move the chip along the entirety of P and end with a divisor D' with the chip on w. We conclude D reaches all vertices of G and so $\text{dgon}(G) = 1$.

Assume on the other hand then that $\text{dgon}(G) = 1$. Suppose there exists a cycle C in G and let v, w be two different vertices on C. Let D be the effective divisor with $D(v) = 1$, this divisor should reach all vertices and so should be equivalent to an effective divisor D' with $D'(w) = 1$. Let A_0 then be the first set of the level set decomposition of the transformation of D to D'. Note that $v \in A_0$ and $w \notin A_0$. But the minimum cut between v and w is at least 2, they are part of the same cycle after all. This implies however that firing A_0 would move two chips, since D only contains one chip this leads to at least one vertex going into debt, contradicting Corollary 3.1.2. We conclude that G cannot contain any cycles and must be a tree.

Now we consider stable divisorial gonality. Note that any graph is a refinement of itself. Therefore any tree graph has a tree as a refinement and it follows the stable divisorial gonality of a tree is 1. Assume on the other that a given graph G has $\text{sdgon}(G) = 1$, then there exists a refinement G' of G with $\text{dgon}(G') = 1$. But by the previous part of our proof then G' must be a tree, and the only graphs that have a tree as refinement are trees themselves. □

We also claimed that the complete graph is the simple graph with the highest divisorial gonality amongst the graphs of n vertices.

Lemma 3.4.1. *Let G be a simple graph with at least 2 vertices. The divisorial gonality of G is at most $|V(G)| - 1$.*

Proof. Choose a vertex $v \in V(G)$. Consider now the divisor D with $D(w) = 1$ if $w \neq v$ and $D(v) = 0$. Clearly the degree of D is $|V(G)| - 1$ and D reaches all vertices $w \neq v$. If we now fire the subset $V(G) - \{v\}$ from D, note that v receives at least 1 chip from a neighbor (we assume G is connected), while any other vertex loses at most 1 chip to v. It follows the resulting divisor is an effective divisor with at least one chip on v and so D can reach v. We conclude that $\text{dgon}(G) \leq |V(G)| - 1$. □

Proof of Theorem 3.0.2. By Lemma 3.4.1 K_n has divisorial gonality at most $n - 1$. Suppose that $\text{dgon}(K_n) < n - 1$, then there exists a divisor D with degree at most $n - 2$ that reaches each vertex of K_n. By Corollary 3.1.2 this requires the existence of valid firing sets on K_n, note though that each partition of K_n has a cut of at minimum $n - 1$. Since there are only $n - 2$ chips in D, it follows no valid firing set can exist for D. But this means D cannot have any equivalent divisors and so cannot reach every vertex. We conclude that K_n must have divisorial gonality exactly $n - 1$. □

In Section 2, Theorem 2.2.1, we stated that subdividing edges is the only necessary operation for stable divisorial gonality. We now have the techniques to prove this statement:

Proof of Theorem 2.2.1. We call any vertex in a refinement created by a leaf-adding operation an external added vertex. Note that the number of external added vertices in a refinement is exactly the number of times the leaf-adding operation was performed to create the refinement. Let now H be a refinement of G with $\mathrm{dgon}(H) = \mathrm{sdgon}(G)$ and minimal number of external added vertices. If H has no external added vertices, H is a valid candidate for G' in the theorem and we are done.

Suppose then that H has at least one external added vertex. Choose one such vertex v with degree 1 and let v' be its neighbor. Let D be an effective divisor on H with $\deg(D) = \mathrm{dgon}(H)$ that reaches all vertices in H. If $D(v) \geq 1$ we create the equivalent effective divisor D' by firing $\{v\}$ until $D'(v) = 0$, otherwise let $D' = D$. Let H' be H, but with v removed. We claim D' restricted to H', denoted here as $D'_{H'}$, reaches all vertices of H'.

Let w be a vertex in H' with $D'_{H'}(w) = 0$. We know D' reaches all vertices of H, so there exists an effective divisor D'_w on H with $D'_w \sim D'$ and $D'_w(w) \geq 1$. Let $A_0, \ldots, A_k$ be the level set decomposition of the transformation of D' into D'_w. Since $D'(v) = 0$ and each firing set of the level set decomposition is valid by Corollary 3.1.2, it follows that if $v \in A_i$ then $v' \in A_i$ for all $i \in [1, \ldots, k]$.

Consider now what happens when we fire each set of the level set decomposition after restricting to H'. Since only v is missing compared to H and $v \in A_i$ only if $v' \in A_i$, each vertex in H' ends up with at least as many chips as it would in H'. Specifically this means we end up with an effective divisor with at least one chip on w. So $D'_{H'}$ reaches w. This works for any $w \in V(H')$ and so it follows that $\mathrm{dgon}(H') \leq \mathrm{dgon}(H)$. But H' is an refinement of G that contains one external added vertex less than H, contradicting the minimality of H. We conclude H must have no external added vertices and so there always exists at least one refinement of minimal divisorial gonality created by only subdividing edges. □

Chapter 4

Reduction rules for divisorial gonality

As mentioned before, the general problem of finding the divisorial gonality of a graph is NP-hard [10]. In this section we therefore try to solve the smaller problem of checking whether a given graph has divisorial gonality at most 2. The method we take to solve this problem is similar to work on treewidth.

It is also known that the general problem of finding the treewidth of graph is NP-hard. Checking if a graph has treewidth at most k can be done in polynomial time though for any given k. For the cases of $k = 2$, $k = 3$ and $k = 4$ this can be done in linear time by making use of a set of reduction rules [1][14]. The idea is that the set of rules can reduce a graph to the empty graph exactly when it is in the class of graphs recognized. The question we answer here, at least partially, is if such sets of reduction rules can also be found for divisorial gonality and if they result in efficient recognition algorithms.

In this chapter we explore a set of reduction rules for graphs with divisorial gonality at most 2. We also show that these rules lead to a polynomial time algorithm to recognize graphs with divisorial gonality at most 2. These results are also part of a larger joint work [4] which presents similar sets of reduction rules for stable divisorial gonality and stable gonality. In addition in [4] it is shown that these reduction rule sets lead to $\mathcal{O}(n \log n + m)$ time algorithms, where n is the number of vertices and m the number of edges.

We start this chapter by formalizing what exactly a set of reduction rules is and what properties we want it to have. After that an extra layer of structure on top of the graph is introduced, called constraints. After this preparation in Section 4.1, the actual set of reduction rules is presented in Section 4.2. We will the prove that the set of reduction rules has the properties we require of it, which will be referred to as *safeness* and *completeness*. Finally in Section 4.3 we present proof that the set of reduction rules allows for a polynomial time algorithm.

4.1 Reduction rules

We will be talking a lot about reduction rules in this section. By a reduction rule we mean a rule that can be applied to a graph to produce a different graph. The following notation shows when a graph can be produced by the application of reduction rules starting from another graph:

Definition 4.1.1. Let G and H be graphs and $\boldsymbol{S}$ be some set of reduction rules. We use $G\boldsymbol{S}H$ to denote that H can be produced by some application of a reduction rule from $\boldsymbol{S}$. We use $G\boldsymbol{S}^*H$ to denote that H can be produced from G by some finite sequence of applications of reduction rules from $\boldsymbol{S}$.

If $\boldsymbol{U}$ is a single rule, we write $G\boldsymbol{U}H$ as shorthand for the application of the singleton set containing $\boldsymbol{U}$.

Our final goal with the set of reduction rules is to show that it can be used to characterize the graphs in a certain class by reduction to the empty graph. For this we need to make sure that membership of the class is invariant under our reduction rules.

Definition 4.1.2. Let $\boldsymbol{U}$ be a rule and $\boldsymbol{S}$ be a set of reduction rules. Let $\mathcal{A}$ be a class of graphs. We call $\boldsymbol{U}$ *safe* for $\mathcal{A}$ if from $G\boldsymbol{U}H$ it follows that $H \in \mathcal{A} \iff G \in \mathcal{A}$. We call $\boldsymbol{S}$ *safe* for $\mathcal{A}$ if every rule in $\boldsymbol{S}$ is safe for $\mathcal{A}$.

Note that if $\boldsymbol{S}$ is safe for a class $\mathcal{A}$ then $G\boldsymbol{S}^*H$ implies that $H \in \mathcal{A} \iff G \in \mathcal{A}$.

Apart from our rule sets being safe, we also need to know that, if a graph is in our class, it is always possible to reduce it to the empty graph. In other words, there must always be some sequence of rule applications that results in the empty graph.

Definition 4.1.3. Let $\boldsymbol{S}$ be a set of reduction rules, $\mathcal{A}$ a class of graphs. We call $\boldsymbol{S}$ *complete* for $\mathcal{A}$ if for any graph $G \in \mathcal{A}$ it holds $G\boldsymbol{S}^*\emptyset$.

For any rule set that is both complete and safe for $\mathcal{A}$ it then follows the rule set is suitable for characterization of $\mathcal{A}$. Additionally it is not possible to make a wrong choice early on that would prevent the graph from being reduced to the empty set.

Lemma 4.1.1. *Let $\boldsymbol{S}$ be a set of rules that is safe and complete for $\mathcal{A}$, with $\emptyset \in \mathcal{A}$, then we have the following for all graphs G, H:*

(i) $G\boldsymbol{S}^*\emptyset$ if and only if $G \in \mathcal{A}$;

(ii) if $G \in \mathcal{A}$ and $G\boldsymbol{S}^*H$, then $H\boldsymbol{S}^*\emptyset$.

Proof. For property *i*: Let G be a graph, such that $G\boldsymbol{S}^*\emptyset$. Note that by the safeness of $\boldsymbol{S}$ and the fact that $\emptyset \in \mathcal{A}$ it follows that $G \in \mathcal{A}$. Assume on the other hand that $G \in \mathcal{A}$, then by the completeness of $\boldsymbol{S}$ it follows that $G\boldsymbol{S}^*\emptyset$.

For property *ii*: Let G be a graph in $\mathcal{A}$ and H a graph such that $G\boldsymbol{S}^*H$. Note that by the safeness of $\boldsymbol{S}$ we have that $H \in \mathcal{A}$, then by completeness of $\boldsymbol{S}$ it follows that $H\boldsymbol{S}^*\emptyset$. □

4.1.1 Constraints

During the reduction of the graph we will need to keep track of certain restrictions otherwise lost by the removal of vertices and edges. We will maintain these restrictions in the form of a set of pairs on the vertices of the graph:

Definition 4.1.4. Given a graph $G = (V, E)$ the *set of constraints* $\mathscr{C}$ is a set of pairs (v, w), where $v, w \in V$. This set can contain pairs of a vertex with itself, but can contain each pair only once.

Checking whether a graph has gonality two or lower can be seen as checking whether there exists a divisor with degree two and rank greater or equal to one on our graph. Constraints in this case are used to restrict which divisors and transformations we consider after reduction. The pairs in the constraints place the following restrictions on what divisors and firing sets are allowed:

Definition 4.1.5. Given a constraint $r = (v, w)$ a divisor *satisfies* r if it is equivalent to an effective divisor after removing one chip from v and one chip from w. In addition any firing set used in transformations should either contain both v and w or neither.

In terms of principal divisors, this is the same as a constraint only allowing those principal divisors $P = \mathcal{L}f$, where $f(v) = f(w)$. Note that in the case that $v = w$ the first part means a divisor should be equivalent to an effective divisor after removing two chips from v and the second condition is fulfilled trivially. It will also be useful to define when constraints are non-conflicting on a cycle. For this we introduce the idea of compatibility on cycles:

Definition 4.1.6. Let C be a cycle. Let $\mathscr{C}_C \subseteq R$ be the set of constraints that contain a vertex in C. We call the constraints $\mathscr{C}_C$ *compatible* if the following hold.

(i) If $(v, w) \in \mathscr{C}_C$ then both $v \in C$ and $w \in C$.

(ii) For each $(v, w) \in \mathscr{C}_C$ and $(v', w') \in \mathscr{C}_C$, the divisor given by assigning a chip to v and w must be equivalent to the one given by assigning a chip to v' and w' on the subgraph consisting of C.

Note that a divisor of degree 2 can only satisfy all constraints on a cycle if they are compatible. Now we are interested in the existence of a divisor that has rank greater or equal to one, while satisfying all constraints.

Definition 4.1.7. Given a graph $G = (V, E)$ and its constraints $\mathscr{C}$, we will call a divisor D *suitable* if it has degree 2 and $r(D) \geq 1$ while also satisfying all constraints in $\mathscr{C}$.

Given a graph with constraints we will say that the graph has divisorial gonality 2 or lower if there exists a suitable divisor. Note that for a graph with no constraints this formulation is equivalent to the usual definition of divisorial gonality 2 or lower. We will denote the class of graphs with constraints that has divisorial gonality two or lower as $\mathcal{G}_2^d$.

4.2 The reduction rules

We are given a graph $G = (V, E)$ and a still empty set of constraints $\mathscr{C}$. We note here that it is assumed we start with a connected graph and all the reduction rules maintain connectivity. The following rules are illustrated in Figure 4.1, where a constraint is represented by a red dashed edge.

We start by covering the two possible end states of our reduction:

Rule E_1^d. *Given a graph consisting of exactly one vertex, remove that vertex.*

Rule E_2^d. *Given a graph consisting of exactly two vertices, u and v, connected to each other by a single edge, and $\mathscr{C} = \{(u, v)\}$, remove both vertices.*

Next are the reduction rules to get rid of vertices with degree equal to one. These rules are split by what constraint applies to the vertex:

Rule T_1^d. *Let v be a leaf, such that v has no constraints in $\mathscr{C}$. Remove v.*

Rule T_2^d. *Let v be a leaf, such that its only constraint in $\mathscr{C}$ is (v, v). Let u be its neighbor. Remove v and add the constraint (u, u) if it does not exist yet.*

Rule T_3^d. *Let v_1 be a leaf, such that its only constraint in $\mathscr{C}$ is (v_1, v_2), where v_2 is another leaf, whose only constraint is also (v_1, v_2). Let u_1 be the neighbor of v_1 and u_2 be the neighbor of v_2 (these can be the same vertex). Then remove v_1 and v_2 and add the constraint (u_1, u_2) if it does not exist yet.*

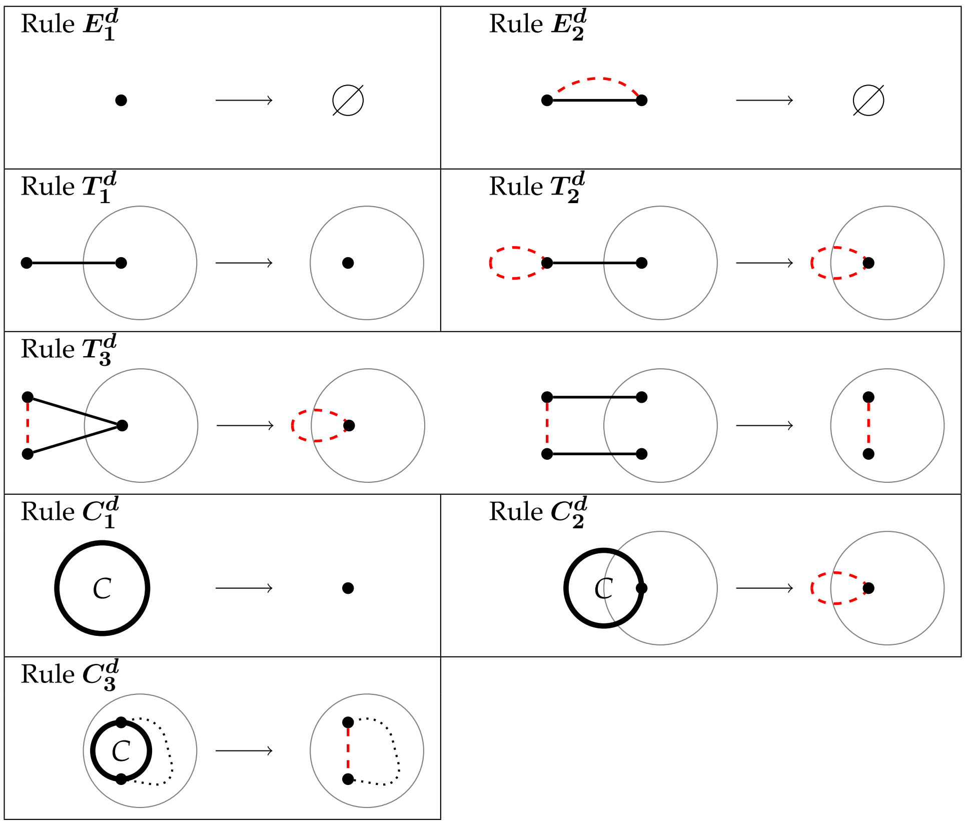

FIGURE 4.1: The reduction rules for divisorial gonality

Finally we have a set of reduction rules that apply to cycles containing at most 2 vertices with degree greater than two. The rules themselves are split by the number of vertices with degree greater than two.

Rule C_1^d. *Let C be a cycle of vertices with degree two. If the set of constraints $\mathscr{C}_C$ on C is compatible, then replace C by a new single vertex.*

Rule C_2^d. *Let C be a cycle with one vertex v with degree greater than two. If the set of constraints $\mathscr{C}_C$ on C plus the constraint (v, v) is compatible, then remove all vertices except v in C and add the constraint (v, v) if it does not exist yet.*

Rule C_3^d. *Let C be a cycle with two vertices v and u of degree greater than two. If there exists a path from v to u that does not share any edges with C and the set of constraints $\mathscr{C}_C$ on C plus the constraint (v, u) is compatible, then remove all vertices of C except v and u, remove all edges in C and add the constraint (v, u) if it does not exist yet.*

We shall use $\mathcal{R}^d$ for the set consisting of all the above reduction rules: E_1^d, E_2^d, T_1^d, T_2^d, T_3^d, C_1^d, C_2^d and C_3^d.

We will now state the main theorem declaring that this set of reduction rules has the desired properties. After this we will build up the proof.

Theorem 4.2.1. *The set of rules $\mathcal{R}^d$ is safe and complete for $\mathcal{G}_2^d$.*

4.2.1 Safeness

We assume here that there is a graph G and H such that H follows from G by applying a reduction rule. To start, we note that all reduction rules maintain connectivity of our graph:

Lemma 4.2.2. *Let G and H be graphs. If G is connected and $G\mathcal{R}^{d*}H$ then H is connected.*

Proof. We observe that the only rule that removes a path between two remaining vertices is $\boldsymbol{C_3^d}$. In the case of $\boldsymbol{C_3^d}$ however we demand that there is a path between v and w outside of C so this path will still exist and it follows that H is still connected. □

Since G is assumed to be connected it follows that each produced graph H is also connected. Now we show for each of the rules in $\mathcal{R}^d$ that it is safe.

Lemma 4.2.3. *Rules $\boldsymbol{E_1^d}$ and $\boldsymbol{E_2^d}$ are safe.*

Proof. For both rules it can easily be seen that their starting states as well as the empty graph have divisorial gonality two or lower. From this it follows they both are safe. □

Lemma 4.2.4. *Rules $\boldsymbol{T_1^d}$ and $\boldsymbol{T_2^d}$ are safe.*

Proof. Let v be our vertex with degree 1 and u its neighbor. We know that the only constraint on v can be the constraint (v, v).

Note that if $H \in \mathcal{G}_2^d$ then there is a divisor on H that puts at least one chip on u. Moving this divisor to G, note that we can move chips to v by firing $G - \{v\}$, it follows that this divisor is also suitable for G.

Given that $G \in \mathcal{G}_2^d$ note that we can find a suitable divisor that has no chips on v by firing v until it contains no chips. This divisor will also be suitable on H.

For $\boldsymbol{T_2^d}$ the proof is analogous, except with two chips on v. □

Lemma 4.2.5. *Rule $\boldsymbol{T_3^d}$ is safe.*

Proof. Let v_1 and v_2 be the vertices with degree one, such that their only constraint is (v_1, v_2) and let u_1 and u_2 be their neighbors. We first assume that $H \in \mathcal{G}_2^d$, then there is a suitable divisor on H with one chip on u_1 and another chip on u_2. Note that we can move this divisor to G. Then by firing $V(G) - \{v_1, v_2\}$ we can move a chip to v_1 and v_2. Therefore this divisor is also suitable on G.

Assume then that $G \in \mathcal{G}_2^d$, then there is a suitable divisor on G with one chip on v_1 and v_2. By firing $\{v_1, v_2\}$ we can create a divisor with a chip on u_1 and u_2 (or two on u_1 if $u_1 = u_2$). Note that this divisor is suitable on H. □

Lemma 4.2.6. *Rule $\boldsymbol{C_1^d}$ is safe.*

Proof. We start by assuming that $H \in \mathcal{G}_2^d$. Note that by Lemma 4.2.2 we have that H is connected. Therefore it follows that H must consist of a single vertex, therefore G consists of a single cycle and it follows that $G \in \mathcal{G}_2^d$, since all constraints are compatible.

Assume then that $G \in \mathcal{G}_2^d$ instead. Since G is connected it must consist exactly of the cycle C, thus H consists of a single point and $H \in \mathcal{G}_2^d$. □

Lemma 4.2.7. *Rule $\boldsymbol{C_2^d}$ is safe.*

Proof. Let C be our cycle with one vertex v with degree greater than 2. Assume that $H \in \mathcal{G}_2^d$; then there is a suitable divisor on H with two chips on v. Move this divisor to G. Note that if we fire $V(G) - C + \{v\}$ then we move the two chips onto the two neighbors of v in C. Since all constraints on C are compatible with the constraint (v, v) it follows that we can move the chips along C while satisfying the constraints on C. From this it follows that our divisor is suitable on G.

Assume now that $G \in \mathcal{G}_2^d$. Since all constraints on C are compatible with (v, v) it follows that we can find a suitable divisor with two chips on v. Moving this divisor to H gives a suitable divisor there. Thus, $H \in \mathcal{G}_2^d$. □

Lemma 4.2.8. *Rule $\boldsymbol{C_3^d}$ is safe.*

Proof. Let C be our cycle and v, w the two vertices with degree greater than two in C. We first assume that $H \in \mathcal{G}_2^d$. From this it follows that there exists a suitable divisor on H with a chip on v and a chip on w. Note that in G all constraints on C plus (v, w) are compatible. From this we see that if we move the divisor from H to G it will be able to satisfy all constraints on C. It is also clear that from v and w we can move chips along either of the arcs that form C together with v and w. Therefore the divisor is also suitable on G and thus $G \in \mathcal{G}_2^d$.

Let us then assume that instead $G \in \mathcal{G}_2^d$. Clearly there exists a suitable divisor D on G that has a chip on v. We will show that there is a suitable divisor that has a chip on both v and w: Assume that $D(w) = 0$, then there should be a suitable divisor D' with $D'(w) = 1$ and $D \sim D'$. This implies there is a level set decomposition of the transformation from D to D'.

If none of the subsets contain v then it follows that $D'(v) = 1$ and we are done. Otherwise let A_i be the first subset that contains v and D_i the divisor before firing A_i. Note that we should have $D_i(a) \geq \text{outdeg}_{A_i}(a)$ for all $a \in A_i$, since all firing sets should be valid by Corollary 3.1.2. Since $\deg(D_i) = 2$ it follows that $\sum_{a \in A_i} \text{outdeg}_{A_i}(a) \leq 2$. This is the same as the cut induced by A_i having size two or lower. Note that the minimum cut between v and w is at least three, since they are both part of C and there exists an additional path outside of C between them. Therefore it follows that A_i can only induce a cut of size two or lower if $w \in A_i$. But this implies that $D_i(w) \geq 1$, since a vertex can not receive a chip after entering the firing set. We conclude that $D_i(v) = 1$ and $D_i(w) = 1$.

Also by the fact that the minimum cut between v and w is at least three it follows that a subset firing can only be valid if the subset contains either both v and w or neither. Since, by Corollary 3.1.2, any transformation between effective divisors can be done by a series of valid subset firings it follows that any transformation can be done while adhering to the constraint (v, w).

Therefore the divisor D_i gives us a suitable divisor on H. We conclude that $H \in \mathcal{G}_2^d$. □

Since we have shown that each of the rules in $\mathcal{R}^d$ is safe, we conclude:

Theorem 4.2.9. *The ruleset $\mathcal{R}^d$ is safe for $\mathcal{G}_2^d$.*

4.2.2 Completeness

By the previous section we now have that membership in $\mathcal{G}_2^d$ is invariant under the reduction rules in $\mathcal{R}^d$. For the reduction rules to be useful however we will also need to confirm that any graph can be reduced to the empty graph by a finite sequence of rule applications, or in other words, the rule set is complete. We will first prove several lemmas required for this.

Lemma 4.2.10. *Let G be a graph and $v \in V(G)$ a vertex. If there are two different constraints on v, so $(v, w), (v, w') \in R(G)$, with $w \neq w'$, then $G \notin \mathcal{G}_2^d$.*

Proof. We first check the possibility where $v = w'$. Then any suitable divisor must be equivalent to the divisor D with $D(v) = 2$, but also equivalent to the divisor

D' with $D'(v) = 1$ and $D'(w) = 1$. But this means these divisors are equivalent to each other. Note however that since we have the constraint (v, w) any firing set containing v must also contain w. Starting with divisor D any valid firing set must contain v (it is the only vertex with chips), which means it must also contain w. This implies no level set decomposition from D to D' can exist, from which it follows that there is no transformation of D into D', so $G \notin \mathcal{G}_2^d$.

The other possibility then is that $v \neq w$ and $v \neq w'$. This means any suitable divisor should be equivalent to the divisor D with $D(v) = 1$, $D(w) = 1$, and equivalent to the divisor D' with $D'(v) = 1$, $D'(w') = 1$. Note that any firing set that contains v also contains both w and w' by our constraints, moreover any firing set containing w contains w' by our constraints. Since starting in D any valid firing set must contain either v or w (they are the only vertices with chips), it follows that any valid firing set must contain w'. Again this implies no level set decomposition from D to D' exists, so D and D' cannot be equivalent. We conclude no suitable divisor can exist and therefore $G \notin \mathcal{G}_2^d$. □

Lemma 4.2.11. *Let $G \in \mathcal{G}_2^d$ be a graph where none of the rules $\boldsymbol{E_1^d}$, $\boldsymbol{E_2^d}$, $\boldsymbol{T_1^d}$, $\boldsymbol{T_2^d}$ or $\boldsymbol{T_3^d}$ can be applied. Then G contains no vertices of degree 1.*

Proof. Assume on the contrary that G does contain a vertex v with degree 1. By Lemma 4.2.10 and $G \in \mathcal{G}_2^d$ we have that at most one constraint contains v. If there is no constraint on v, we could apply Rule $\boldsymbol{T_1^d}$ to it, therefore there is exactly one constraint on v. If this constraint is (v, v) we would be able to apply Rule $\boldsymbol{T_2^d}$ to v. If the constraint is (v, w), where w is another vertex of degree 1, Rule $\boldsymbol{T_3^d}$ could be applied to v. The only remaining possibility is that the constraint on v is the constraint (v, w) where w is a vertex with degree greater than 1. We will use D to denote the divisor with $D(v) = D(w) = 1$. Since we have the constraint (v, w) and $G \in \mathcal{G}_2^d$, D is a suitable divisor.

We first consider the case where w is not a cut-vertex. Let u be the neighbor of v. Consider the transformation from D to a divisor D' with $D'(u) = 1$. Let A_0 be the first firing set in the level decomposition of this transformation. Note that we have $v, w \in A_0$ and $u \notin A_0$. Since w is not a cut-vertex, it follows for each neighbor w_i of w that there is a path from w_i to u that does not contain w or $w_i = u$. Note that if a neighbor $w_i \neq u$ is in A_0, then somewhere on its path to u must be an edge that crosses between A_0 and A_0^c. But such a crossing edge would imply the firing set is not valid, since no vertex on this path contains a chip. Since w has degree at least two, and all its neighbors are not in A_0, it follows the firing set is not valid, since w would lose at least two chips. Since no valid firing set exists to start the transformation, it follows that no transformation from D to D' exists, but this implies that $r(D) < 1$. Since D should be suitable by constraint (v, w) we have a contradiction.

We proceed with the case where w is a cut-vertex. Let C_x be a connected component not containing v after removing w. Consider the subset C_x in G. Note that from D we can never obtain an equivalent divisor with two chips on C_x. Since the chip from v would have to move through w to get to C_x, this would require D to be equivalent to a divisor with two chips on w, which is impossible if $G \in \mathcal{G}_2^d$ by Lemma 4.2.10. Since D has rank greater than zero it then follows C_x must be a tree. This means C_x must contain a vertex x of degree one, we know however that since we cannot apply rules $\boldsymbol{T_1^d}$, $\boldsymbol{T_2^d}$ or $\boldsymbol{T_3^d}$ to G, x must have a constraint (x, y) where y is a vertex with degree greater than one. We now consider the possible locations of y.

If $y \in C_x$, then D must be equivalent to a divisor with a chip on x and a chip on y. As mentioned before D cannot be equivalent to a divisor with two chips on C_x, so it follows $y \notin C_x$.

Since $y \notin C_x$, D has to be equivalent to the divisor D'' with $D''(x) = D''(y) = 1$. Let C_y be the component containing y. Let A_0 be the first subset of the level set decomposition of the transformation of D into D''. Note that $v, w \in A_0$ and $x, y \notin A_0$. But this implies that w has at least one neighbor w_1 in C_y, with $w_1 \notin A_0$, namely the first vertex on the path from w to y. But w also has at least one neighbor w_2 in C_x, with $w_2 \notin C_x$, namely the first vertex on the path from w to x. But this means w has two neighbors that it will send a chip to, but w only has one chip. By Corollary 3.1.2 then no transformation from D to D'' can exist and thus $y \notin C_y$, giving a contradiction.

We conclude there can be no such constraint (x, y) and from this we conclude that no vertices with degree 1 can exist in G. $\square$

Lemma 4.2.12. *Let G be a graph with a set of constraints $\mathscr{C}$ and let C be a cycle in G with $\mathscr{C}_C$ the set of constraints that contain a vertex in C. If $G \in \mathcal{G}_2^d$ then the constraints $\mathscr{C}_C$ are compatible.*

Proof. We first show the first property of a compatible constraint set holds. Let $(v, w) \in \mathscr{C}_C$ be a constraint and let $v \in C$ without loss of generality. We show that $w \in C$. Assume on the contrary that $w \notin C$, then let D be the divisor with $D(v) = D(w) = 1$, D should be suitable. Let x be a vertex in C with $x \neq v$. Let D' be any divisor with $D'(x) \geq 1$. Let A_0 be the first firing set of the level set decomposition of the transformation of D into D'. Note that $v, w \in A_0$ and $x \notin A_0$. But note there are two disjoint paths from v to x, since they are on the same cycle. But this implies a chip will be sent along both these paths by A_0, but since $w \notin C$, both these chips must come from v, but v only has one chip. We conclude no transformation can exist and thus D is not suitable, a contradiction.

For the second property, let $(v, w), (v', w') \in \mathscr{C}_C$ be two constraints on C. By our first property we have that $v, w, v', w' \in C$. Let D be the divisor with $D(v) = D(w) = 1$ and D' the divisor with $D'(v') = D'(w') = 1$. Let A_0 be the first firing set of the level set decomposition of the transformation of D into D'. Note that $v, w \in A_0$ and $v', w' \notin A_0$. We observe that v and w split C into two arcs. Note that both v' and w' must be on the same arc: if they are not on the same arc, there exists disjoint paths from v to v' and to w' that do not contain w. This implies that A_0 sends two chips along these paths, but v only has one chip.

Now note that C is biconnected, this implies that for a firing set A with $w \in A$ and $w' \notin A$ to be valid there must be at least two chips on vertices in C. This follows since there are at least two edges crossing between A and A^C in C. We know D and D' must be equivalent, since $G \in \mathcal{G}_2^d$ and both correspond to constraints on G, so let $A_0, \dots, A_k$ be the level set decomposition of the transformation of D into D'. Since each of these firing sets is valid by Lemma 3.1.2, it follows this transformation leaves two chips on C at each intermediate divisor. It follows that if we restrain these firing sets to C, we have a sequence of firing sets that transforms D into D' on C. Therefore D and D' are equivalent on C, so our second property is also fulfilled. $\square$

Lemma 4.2.13. *Let G be a simple graph of treewidth 2 or lower and containing at least 4 vertices, then G has at least two vertices with degree 2 or lower.*

Proof. Let T be a tree decomposition of G, such that we can remove no vertex of T while still keeping a valid tree decomposition. Then note that T must contain at least two leaves t_1 and t_2. Note that t_1 must contain at least one vertex v from G that is sent to no vertex of T, since otherwise we would be able to remove t_1. Then note that all neighbors of v must be sent to t_1, but t_1 contains at most three elements of G, so v has at most two neighbors. Applying the same argument to t_2 we find another vertex with at most two neighbors. $\square$

Lemma 4.2.14. *Given a non-empty graph $G \in \mathcal{G}_2^d$ there is a rule in $\mathcal{R}^d$ that can be applied to G.*

Proof. Let $G \in \mathcal{G}_2^d$ be such a graph and assume that instead no rule in $\mathcal{R}^d$ can be applied to G. By Lemma 4.2.11 we have that such a graph can contain no vertices of degree one. Therefore our graph consists of vertices of degree 2 and of a set T of vertices of degree 3 or greater. Consider the minor H of G created by contracting each path of degree 2 vertices to an edge. Note that any edge in H therefore represents a path consisting of a single edge or a path with any number of vertices with degree 2 in G.

Now assume we have a loop in H, note this loop corresponds to a path of degree 2 vertices in G going from a degree 3 or greater vertex to itself, so this path plus the vertex forms a cycle with one vertex of degree 3 or greater. By Lemma 4.2.12 the constraints on this cycle are compatible, so we are able to apply Rule $\boldsymbol{C_2^d}$ to it. Since we assumed no rules can be applied, it follows H contains no loops.

Now we attempt to find a subgraph H' of H with no multiple edges. If H contains no multiple edges, simply let $H' = H$. Otherwise let v and w be vertices such that there are at least two edges between v and w. If v and w are still connected to each other after removing two edges e_1, e_2 between them, note that these edges correspond to two disjoint paths of degree 2 vertices in G. Thus v, w plus the paths corresponding to e_1 and e_2 form a cycle C in G with exactly two vertices of degree 3 or greater, where v and w have a path that does not share any edges with C. Again by Lemma 4.2.12 we have that the constraints on this cycle are compatible and so we are able to apply Rule $\boldsymbol{C_3^d}$ to C. From this it follows that v and w must be disconnected after removing e_1 and e_2. So any multiple edge in H consists of a double edge, whose removal splits the graph in two connected components. Let H' be the connected component of minimal size over all removals of a double edge in H. Note that H' cannot contain any double edge, since this would imply a smaller connected component.

Note we now have a minor H' of G, where each vertex has degree at least 3 with at most one exception, which has no loops or multiple edges and therefore is a simple graph. Since H' only has at most one vertex with degree lower than three, by Lemma 4.2.13 it follows that $\operatorname{tw}(H') \geq 3$ and since treewidth is closed under taking minors we get $\operatorname{tw}(G) \geq 3$. But then by Lemma 3.0.3 it follows that $\operatorname{dgon}(G) \geq 3$, creating a contradiction, since $G \in \mathcal{G}_2^d$. We conclude our assumption must be wrong and there is a rule in $\mathcal{R}^d$ that can be applied to G. □

Now we have everything required to prove our main theorem:

Proof of Theorem 4.2.1. By Theorem 4.2.9 we have that $\mathcal{R}^d$ is safe. It remains to prove that $\mathcal{R}^d$ is also complete.

Assume that $G \in \mathcal{G}_2^d$. By Lemma 4.2.14 and Theorem 4.2.9 we have that we can keep applying rules from $\mathcal{R}^d$ to G as long as G has not been turned into the empty graph yet. Now observe that each rule removes at least one vertex or in the case of $\boldsymbol{C_3^d}$ at least two edges, while never adding more vertices or edges. Since G starts with a finite number of vertices and edges it follows that rules from $\mathcal{R}^d$ can be only applied a finite number of times to the graph. When no more rules can be applied to the graph, it follows the graph has been reduced to the empty graph. Therefore $G\boldsymbol{\mathcal{R}}^{d*}\emptyset$ and it follows that $\mathcal{R}^d$ is complete. □

By Lemma 4.1.1 it follows $\mathcal{R}^d$ has the properties we want it to have so that we are able to use it for characterization of the graphs with divisorial gonality two or lower.

4.3 An algorithm based on the reduction rules

So we now have a set of reduction rules that reduces exactly the graphs with divisorial gonality 2 or lower to the empty graph. We will argue here that by using this set of reduction rules we can create a polynomial time algorithm that determines if a graph has divisorial gonality at most 2. We use n in this section to denote the number of vertices and m the number of edges.

For our proofs we assumed the graph was connected and loopless. Suppose our algorithm gets a non-connected graph as input, note that this graph has divisorial gonality at most 2 exactly if it consists of two trees. This can easily be checked in $\mathcal{O}(n+m)$ time. Suppose our input graph has loops in it. We know that loops have no influence on the divisorial gonality of the graph, so we can remove them in $\mathcal{O}(m)$ time before running the rest of the algorithm.

The idea of the algorithm is to repeatedly check if we can apply any of the rules and if so apply it. If at any point no rules can be applied to the graph, we check if we ended in the empty graph. If so, we conclude the input graph had divisorial gonality at most 2, otherwise we conclude it did not.

We start by bounding the number of rule applications that can be performed before the algorithm terminates:

Lemma 4.3.1. *The number of rules that can be applied to any graph in sequence is bounded by $n+m$.*

Proof. Note that every rule in $\mathcal{R}^d$ removes at least one edge or vertex from the graph. Since no rule adds any edges or vertices, it follows at most $n+m$ rules can be applied to any graph. □

Next is to show that every rule can be checked and performed in polynomial time:

Lemma 4.3.2. *It can be checked in $\mathcal{O}(n^2(n+m))$ time if any of the rules from $\mathcal{R}^d$ can be applied to a graph. The application of any rule can be performed in linear time.*

Proof. Rules $\boldsymbol{E_1^d}$ and $\boldsymbol{E_2^d}$ can only be applied if there are exactly 1 or 2 vertices left, and thus can be checked and performed in constant time. By keeping track of the degree of each vertex, we can check for leaves in linear time and the same holds for removal. Note that rule $\boldsymbol{C_1^d}$ can only be applied if the entire graph is a single cycle, this can easily be checked in $\mathcal{O}(n)$ time. $\boldsymbol{C_1^d}$ can also be applied in linear time.

To check for rules $\boldsymbol{C_2^d}$ and $\boldsymbol{C_3^d}$ we need to find all cycles containing 1 or 2 vertices with degree greater than 2. These can be found by performing a depth first search from each vertex with degree at least 3, passing only through vertices with degree 2 and noting which vertices with degree at least 3 are reachable from the vertex in this way and by how many disjoint paths. Any vertex with degree at least 3 that can reach itself through such a path is a candidate for rule $\boldsymbol{C_2^d}$. Any pair of vertices with degree at least 3 with two such paths between them is a possible candidate for rule $\boldsymbol{C_3^d}$, the second condition of which can be checked in time $\mathcal{O}(n+m)$ by another depth first search ignoring the two paths.

For any vertex with degree at least 3 this depth first search can be performed in $\mathcal{O}(n+m)$ time and the possible candidates checked in $\mathcal{O}(n(n+m))$ time. Since there are at most n vertices we need to perform this search from, we can check if there are any candidates for rules $\boldsymbol{C_2^d}$ and $\boldsymbol{C_3^d}$ in $\mathcal{O}(n^2(n+m))$ time. Both of these rules can be applied in time linear in the number of removed vertices, which is bounded by n.

We conclude that checking if there is a rule in $\mathcal{R}^d$ that can be applied can be done in $\mathcal{O}(n^2(n+m))$ time. The application of any rule can also be done in at most $\mathcal{O}(n)$ time. □

Theorem 4.3.3. *The set of reduction rules $\mathcal{R}^d$ allows for a polynomial time algorithm to determine whether a given graph has divisorial gonality at most 2 or not.*

Proof. The algorithm is as proposed before: first make sure the graph is connected and loopless; then apply rules until none can be applied and finally check if the empty graph was reached.

Taking care of disconnected graphs and loops in the graph can be done in $\mathcal{O}(n+m)$ time. By Lemma 4.3.1 we know that we need to check for and apply a rule at most $n+m$ times. By Lemma 4.3.2 we know each of these steps takes at most $\mathcal{O}(n^2(n+m))$ time. It follows that all application of rules takes at most $\mathcal{O}(n^2(n+m)^2)$ time. Checking if the remaining graph is the empty graph can be done in constant time.

Since each of the phases of the algorithm takes at most polynomial time, the entire algorithm terminates in polynomial time. □

While algorithm proposed here runs in polynomial time, it is not actually very efficient. By making use of Courcelle's Theorem, see [7], and Theorem 3.0.3 it is actually possible to design an algorithm that runs in $\mathcal{O}(n \log n + m)$ time. For details on this algorithm we refer to [4]. In addition similar sets of reduction rules are presented for stable divisorial gonality and stable gonality in the paper.

Chapter 5

Gonality of minors

Given a graph G of which we know the gonality, what can we say about the gonality of a minor (or subgraph) H of G? In 2015 Norin proposed this question in a survey about graph-minor theory [13]. In this chapter the following three questions will be considered:

Question 5.0.1. *If G and H are graphs, where H is a connected minor of G, is $\text{gon}(H) \leq \text{gon}(G)$?*

Question 5.0.2. *If G and H are graphs, where H is a connected subgraph of G, is $\text{gon}(H) \leq \text{gon}(G)$?*

Question 5.0.3. *If G and H are graphs, where H is a connected subgraph of G with a universal vertex, is $\text{gon}(H) \leq \text{gon}(G)$?*

Where a universal vertex is defined as follows:

Definition 5.0.1. A universal vertex of a graph G is a vertex that is adjacent to each other vertex of G.

Recent work has already answered these questions for the case of divisorial gonality [11]. In this chapter a new, simpler proof for the answers is presented. In addition similar techniques and examples are used to also answer the three questions for the cases of stable divisorial and stable gonality. The answers to the questions for stable divisorial and stable gonality are new results.

5.1 Divisorial gonality

We start by constructing an example that shows that the answer to Question 5.0.2 is *No*. This example also answers Question 5.0.1, since any subgraph of a graph is also a minor of that graph.

Theorem 5.1.1. *There exist graphs G and H, where H is a connected subgraph of G, such that $\text{dgon}(H) > \text{dgon}(G)$.*

Proof. Let G be as shown in Figure 5.1 and let H be obtained from G by removing vertex v. To see that G has divisorial gonality 2, first assign 2 chips to vertex u. Then by firing u we can move the chips to its two neighbors. Now to continue moving our chips to the left, we repeatedly add the current vertices containing chips to the firing set. This results in the chips passing through all vertices and ending up in y and x. We conclude that G has divisorial gonality at most 2. Since G is not a tree, it follows it has divisorial gonality exactly 2.

Now we move on to proving that H has divisorial gonality at least 3. For this we make use of the set of reduction we introduced in Chapter 4. By applying rules from

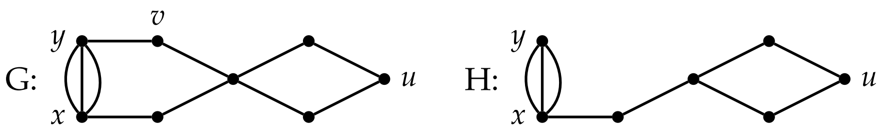

FIGURE 5.1: The graph G on the left and its subgraph H on the right.

this set we can reduce H to a graph with a vertex that is incident to two different constraints. From this graph we cannot reduce to the empty graph by Lemma 4.2.10 and it follows that H must have divisorial gonality at least 3. □

This example shows that for divisorial gonality the answer to Questions 5.0.1 and 5.0.2 is *No*. Luckily it turns out that at least the answer to Question 5.0.3 is *Yes*. The original proof of this in [11] is fairly complex, so we present a new shorter proof here.

Theorem 5.1.2. *If G and H are graphs, such that H is a subgraph of G with a universal vertex, then* $\text{dgon}(H) \leq \text{dgon}(G)$.

Proof. Let D be an effective divisor on G with $\text{rank}(D) \geq 1$ and $\deg(D) = \text{dgon}(G)$. Let D_v be the v-reduced divisor of D, where v is the vertex that is universal in H. Also consider the divisor D_v^H on H which is the restriction of D_v to H. We will show that $\text{rank}(D_v^H) \geq 1$.

Let u be a vertex of H other than v. Our aim is to find an equivalent effective divisor to D_v^H that has a chip on u. If $D_v^H(u) \geq 1$ we are done, so assume $D_v^H(u) = 0$. It follows $D_v(u)$ is also 0.

Since $\text{rank}(D_v) \geq 1$ there exists an effective divisor D' with $D'(u) \geq 1$ and $D_v \sim D'$. Let $A_0, A_1, \ldots, A_k$ be the level set decomposition of the transformation from D_v to D'. Note that since D_v is v-reduced, it follows that $v \in A_0$. Let A_0^H be the restriction of A_0 to H.

We claim firing A_0^H from D_v^H results in an effective divisor with at least one chip on u. First, since $v \in A_0^H$ and v is universal in H, it follows that at least one chip will move to u by firing A_0^H. It then remains to be proven that A_0^H is valid. For this we use the fact that A_0 is valid on D_v. Since a vertex in H has at most as many neighbors not in A_0^H as it had in G, it follows $\text{outdeg}_{A_0^{H'}}(u) \leq \text{outdeg}_{A_0}(u) \leq D_v(u) = D_v^H(u)$, for all $u \in V(H)$. So A_0^H is also valid.

This construction shows there exists an equivalent effective divisor to D_v^H with at least one chip on a vertex u. Because this holds for each $u \in H$, we conclude that $\text{rank}(D_v^H) \geq 1$. Since D_v^H has degree at most that of D_v, we see that $\text{dgon}(H) \leq \text{dgon}(G)$. □

To summarize, we have shown here the following for divisorial gonality:

- Is $\text{dgon}(H) \leq \text{dgon}(G1)$ when H is a connected minor of G? *No*
- Is $\text{dgon}(H) \leq \text{dgon}(G1)$ when H is a connected subgraph of G? *No*
- Is $\text{dgon}(H) \leq \text{dgon}(G1)$ when H is a connected subgraph of G with a universal vertex? *Yes*

5.2 Stable divisorial gonality

The next case we answer the questions for is that of stable divisorial gonality. It turns out that the results for stable divisorial gonality are quite similar to those of divisorial gonality. We start again by answering Questions 5.0.1 and 5.0.2.

Theorem 5.2.1. *There exist graphs G and H, where H is a connected subgraph of G, such that* $\operatorname{sdgon} H > \operatorname{sdgon} G$*.*

Proof. We again make use of the graph G as displayed in Figure 5.1. For the proof of Theorem 5.1.1 we showed that G has divisorial gonality 2. From this it follows G has stable divisorial gonality at most 2. Since G is not a tree, it has stable divisorial gonality exactly 2 by Theorem 3.0.1.

Let H again be the minor of G created by removing the vertex v. The goal now is to show that H has stable divisorial gonality at least 3. For this we will make use of a set of reduction rules that can be used to recognize the graphs with stable divisorial gonality at most two [4, figures 4, 7]. Similar to the set of reduction rules introduced in this work, a graph has stable divisorial gonality at most 2 exactly when it can be reduced to the empty graph by the reduction rules. By applying rules from this set we can reduce H to a graph with a vertex that is incident to two green edges. From this point it cannot be reduced to the empty graph and it follows H has stable divisorial gonality at least 3. We conclude $\operatorname{sdgon}(H) > \operatorname{sdgon}(G)$. □

By the same example used for the case of divisorial gonality we have shown that the answer to Questions 5.0.1 and 5.0.2 is *No*. We move on to answering Question 5.0.3 for stable divisorial gonality. The proof used here is similar to that of Theorem 5.1.2, though the addition of refinements adds complexity.

Lemma 5.2.2. *Let D be an effective divisor on a graph G with $D(v) \geq 1$ and $D(w) \geq 1$, where v and w are distinct vertices. If P is a path between v and w consisting of interior vertices of degree* 2*, then D can reach each vertex on P with at least one chip.*

Proof. We proof by induction on the number of vertices between v and w in P. Let D be the divisor and v, w vertices as mentioned in the lemma.

If P contains 1 vertex p_1 between v and w, then firing $V(G) - \{p_1\}$ results in one chip from v and one chip from w going to p_1. If P contains 2 vertices p_1 and p_2 between v and w, then firing $V(G) - \{p_1, p_2\}$ results in one chip from v going to p_1 and one chip from w going to p_2. In both cases this is enough to reach all vertices in P from D.

Assume now that all vertices on any such path containing of at most n vertices between its ends can be reached by D. If we have a path $P = \{v, p_1, \ldots, p_{n+1}, w\}$ containing $n+1$ vertices between v and w firing $V(G) - \{p_1, \ldots, p_{n+1}\}$ results in one chip moving from v to its neighbor p_1 and one chip moving from w to its neighbor p_{n+1}. Note that D has now reached p_1 and p_{n+1} with a chip. After this $P - \{v, w\}$ is a path of degree 2 vertices with a chip on both ends containing $n - 1$ vertices between its ends. By our assumption we know all vertices on such a path can be reached by the divisor D. We conclude all vertices on P can be reached by D. □

Theorem 5.2.3. *If G and H are graphs, such that H is a subgraph of G with a universal vertex, then* $\operatorname{sdgon}(H) \leq \operatorname{sdgon} G$*.*

Proof. Suppose we have such graphs G and H, where H has a universal vertex v. Let G' be a refinement of G such that $\operatorname{dgon}(G') = \operatorname{sdgon}(G)$. Let H' be the refinement of

$$\begin{tikzcd} G \arrow[r, "\text{subgraph}"] \arrow[d, "\text{refinement}"'] & H \arrow[d, "\text{refinement}"] \\ G' & H' \end{tikzcd}$$

FIGURE 5.2: An overview of the different graphs used in the proof of theorem 5.2.3

H created by applying the refinements that turned G into G' when they apply to an edge contained in H, except for those applying to an edge incident to v. Note that v is still universal in H' with regard to the original vertices. See Figure 5.2 for an overview of the relations of the graphs.

Let D be a divisor on G' with $\mathrm{rank}(D) \geq 1$ and $\deg(D) = \mathrm{dgon}(G')$ and D_v the v-reduced divisor equivalent to D. We then construct the divisor $D_v^{H'}$ by taking D_v, moving all chips, from vertices created by refinements of an edge incident to v, to v and then restricting the divisor to vertices in H' (the vertices of H' are a subset of those of G'). We will now show that $\mathrm{rank}(D_v^{H'}) \geq 1$ on H', we will do this by showing that $D_v^{H'}$ reaches all vertices and using Corollary 2.1.2.

We start by showing that $D_v^{H'}$ can reach each original vertex in H'. Let u be such a vertex, if $D_v^{H'}(u) \geq 1$ we are done, so assume $D_v^{H'}(u) = 0$. Then it follows that $D_v(u) = 0$, so there must be a transformation from D_v to an effective divisor that assigns a chip to u on G'. Let A_0 be the first subset of the level set decomposition of this transformation. Note that since D_v is v-reduced, $v \in A_0$ and $u \notin A_0$ because $D_v(u) = 0$. Now let $A_0^{H'}$ be the restriction of A_0 to vertices in H'. We claim firing $A_0^{H'}$ from $D_v^{H'}$ results in an effective divisor with at least one chip on u.

To show this, we first observe that, if we fire $A_0^{H'}$, we move at least one chip from v to u, since they are adjacent. So the resulting divisor has at least one chip on u. The next step is to prove that the resulting divisor is also effective. We start by checking what happens to v. Let e be an edge incident to v along which v sends a chip by firing $A_0^{H'}$. There are two possibilities: firing A_0 in G' also results in a chip being sent along this edge or there was a refinement of the edge and the vertex created by this refinement is also in A_0. In the second case somewhere among the refinements of e is a border between A_0 and its complement. Since A_0 is valid, there must then be a chip on one of these vertices created by refinements in D_v. Which means this chip was moved to v when we created $D_v^{H'}$ based on D_v. So for each chip v loses by firing $A_0^{H'}$, either v also loses this chip when firing A_0 from D_v or v has an additional chip in $D_v^{H'}$ compared to D_v. From this it follows that $\mathrm{outdeg}_{A_0^{H'}}(v) \leq \mathrm{outdeg}_{A_0}(v) + k$, where k is the number of additional chips v has in $D_v^{H'}$ compared to D_v. So v will not go into debt by firing $A_0^{H'}$.

We also need to check no other vertex of H' will go into debt by firing $A_0^{H'}$. Let w be a vertex in $A_0^{H'}$, $w \neq v$. Note that the neighbors of w in H' are a subset of its neighbors in G', with the potential addition of v. But since $v \in A_0^{H'}$, it follows $\mathrm{outdeg}_{A_0^{H'}}(w) \leq \mathrm{outdeg}_{A_0}(w)$. Because $D_v^{H'}(w) = D_v(w)$, no other vertex will go into debt by firing A_0. We conclude that firing A_0 results in an effective divisor with at least one chip on u.

We know now that the divisor $D_v^{H'}$ can reach all original vertices in H'. It remains now to be shown that $D_v^{H'}$ can also reach vertices in H' created by refinements of edges.

Let r_i be such a vertex. Let $e(u_1, u_2)$ be the original edge in H that was refined to create r_i and potentially more new vertices on the same edge. If the edge was refined into k additional vertices, then the situation is as in Figure 5.3. We first observe that if both $\{u_1, r_1, \ldots, r_{i-1}\}$ and $\{r_{i+1}, \ldots, r_k, u_2\}$ contain at least one vertex with a chip on it in $D_v^{H'}$ then by Lemma 5.2.2 r_i can be reached by $D_v^{H'}$. Assume then without loss of generality that $\{u_1, r_1, \ldots, r_{i-1}\}$ contains no vertex with a chip on it in $D_v^{H'}$.

By the previous part of the proof there exists a subset $A_0^{H'}$ with $v \in A_0^{H'}$ and $u_1 \notin A_0^{H'}$ that by firing from $D_v^{H'}$ results in an effective divisor with a chip on u_1. Now if $u_2 \notin A_0^{H'}$ firing this subset also results in a chip being moved onto u_2 in which case again by Lemma 5.2.2 it follows r_i can be reached. So assume instead that $u_2 \in A_0^{H'}$, then there must be an edge somewhere in between u_1 and u_2 that crosses the border of this firing set. But this means that the vertex incident to this edge outside the firing set contains a chip after firing $A_0^{H'}$. Since there were no vertices with chips on them in $\{u_1, r_1, \ldots, r_{i-1}\}$, it follows this vertex with a new chip must be within $\{r_i, \ldots r_k\}$. But now either r_i has a chip on it after firing $A_0^{H'}$ or by Lemma 5.2.2 r_i can be reached (since both this vertex in $\{r_{i+1}, \ldots r_k\}$ and u_1 have a chip).

We conclude that $D_v^{H'}$ can always reach r_i and since this also holds for the original vertices u it follows that $D_v^{H'}$ reaches all vertices. By how $D_v^{H'}$ was constructed we know that $\deg(D_v^{H'}) \leq \deg(D_v)$, so it follows $\text{dgon}(H') \leq \text{dgon}(G')$. Since H' is a refinement of H and we chose G' such that $\text{dgon}(G') = \text{sdgon}(G)$ we conclude $\text{sdgon}(H) \leq \text{sdgon}(G)$. □

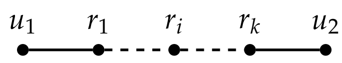

FIGURE 5.3: The situation surrounding r_i in H'.

To summarize, we have shown that the same results hold for stable divisorial gonality as for divisorial gonality, even though refining the graph adds more possibilities:

- Is $\text{sdgon}(H) \leq \text{sdgon}(G1)$ when H is a connected minor of G? *No*
- Is $\text{sdgon}(H) \leq \text{sdgon}(G1)$ when H is a connected subgraph of G? *No*
- Is $\text{sdgon}(H) \leq \text{sdgon}(G1)$ when H is a connected subgraph of G with a universal vertex? *Yes*

5.3 Stable gonality

Finally we will consider the three questions in the case of stable gonality. Before we saw that the answers to the three questions were the same for divisorial and stable divisorial gonality. For stable gonality however the result is different. Again Questions 5.0.1 and 5.0.2 are answered first. Here the result is still the same as that of divisorial and stable divisorial gonality.

Theorem 5.3.1. *There exist graphs G and H, where H is a connected subgraph of G, such that* $\text{sgon}(H) > \text{sgon}(G)$.

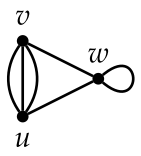

FIGURE 5.4: An example for theorem 5.3.2

Proof. Again the graph depicted in Figure 5.1 is used as graph G. We also create the same graph H again by removing vertex v from G. To see that $\text{sgon}(H) > \text{sgon}(G)$ we will make use of a set of reduction rules created to recognize graphs with stable gonality at most 2 [4, figure 4]. A graph has stable gonality at most 2 exactly when it can be reduced to the empty graph by these reduction rules. By repeatedly applying rules we can reduce G to the empty graph and it follows that $\text{sgon}(G) \leq 2$. On the other hand, when applying the reduction rules to graph H, we run into a vertex with two distinct incident green edges. Such a graph cannot be reduced to the empty graph and so $\text{sgon}(H) > 2$. We conclude $\text{sgon}(H) > \text{sgon}(G)$. □

We see that for Questions 5.0.1 and 5.0.2 the answers are the same as in the cases of divisorial and stable divisorial gonality. Even the counterexample that can be used is the same. Surprising then is the following result, showing this is not the case for Question 5.0.3.

Theorem 5.3.2. *There exist graphs G and H, where H is a connected subgraph of G with a universal vertex, such that* $\text{sgon}(H) > \text{sgon}(G)$

Proof. Let graph G be as the graph displayed in Figure 5.4, consisting of three connected vertices. Then we create the connected subgraph H by removing the edge between v and w in G, note that u is universal in H. As in the previous proof we shall make use of the set of reduction rules for stable gonality at most 2 [4, figure 4]. By repeatedly applying rules to G, we can quickly reduce it to the empty graph. From this it follows that $\text{sgon}(G) \leq 2$. When attempted with graph H however, we again run into a vertex with two distinct incident green edges. Since this means the graph cannot be reduced to the empty graph it follows that $\text{sgon}(H) > 2$. We conclude $\text{sgon}(H) > \text{sgon}(G)$. □

It is interesting to consider why stable gonality behaves differently here than divisorial or stable divisorial gonality. An important observation is that the counterexample used here is reliant on there being a loop in the graph. Loops have no influence on the divisorial or stable divisorial gonality of a graph, while they can have significant influence of the stable gonality of that same graph. During this research no counterexample was found that does not make use of a loop in the subgraph H. The following interesting question therefore remains open:

Question 5.3.1. *If G and H are graphs, where H is a connected loopless subgraph of G with a universal vertex, is* $\text{sgon}(H) \leq \text{sgon}(G)$*?*

Chapter 6

Conclusion

In this thesis a variety of aspects of divisorial gonality have been studied, in addition some questions relating to minors and subgraphs were also answered for the related cases of stable divisorial gonality and stable gonality. To start we introduced three different definitions of divisorial gonality, one using a chip-firing game and two using divisor theory, and showed that they are equivalent to each other. This allowed for a mixture of the more intuitive thinking about chip-firing with the formality of divisors.

Then in Chapter 3 we proved and reviewed several techniques and results for divisorial gonality. In terms of results, we saw how divisorial gonality is related to treewidth, the divisorial gonality of several example classes of graphs and an upper bound on the divisorial gonality of any graph. In this chapter we also introduced the techniques of level set decomposition and reduced divisors, both of which have proven to be quite useful for reasoning about divisorial gonality and divisor equivalence.

The main results of this thesis were presented in Chapters 4 and 5. We introduced a set of reduction rules that can be used to recognize the graphs with divisorial gonality at most 2. Using these rules the graphs can be recognized in polynomial time, even though the general problem of determining divisorial gonality is NP-hard. As mentioned before, more details on an fast algorithm using the rules and similar results for the cases of stable divisorial gonality and stable gonality can be found in a joint work with Marieke van der Wegen, Hans L. Bodlaender and Gunther Cornelissen [4].

After this we answered a set of questions about the gonality of subgraphs and minors. The reduction rules from Chapter 4 acted as an useful tool for checking the divisorial gonality of graphs. It turns out that at least in the cases of divisorial gonality and stable divisorial gonality having an universal vertex in a subgraph is a sufficient condition to bound the divisorial gonality.

Open problems and possible future research

A variety of open problems still remains in the field of gonality. In Chapter 5 we already mentioned the open question 5.3.1, regarding the stable gonality of loopless subgraphs with an universal vertex. As a more general question it remains open if there are other sufficient conditions for a subgraph to have bounded gonality.

Another possible area of future study concerns reduction rules. The motivation for finding the set of rules presented in this work was the similar existing result for treewidth. In that case however, existing work has already shown that sets of rules also exist for the cases of $\text{tw} \leq 3$ and $\text{tw} \leq 4$, see [1] and [14]. Perhaps there are then also sets of reduction rules that can recognize graphs with divisorial gonality at most 3 or 4?

While the work in this thesis was focused mostly on divisorial gonality itself and its properties, the main motivation behind the research is the field of parametrized complexity. Already treewidth has allowed for many problems on graphs to be solved in fast polynomial time, assuming the treewidth of the graph is bounded. Even for bounded treewidth though, many problems remain hard. An important open question then is if there problems among these that can be solved efficiently for graphs with bounded gonality. One such problem considered during the process of this thesis is the $L(2,1)$-labeling problem, but no parametrized algorithm was found yet.

In addition there has not been much research into general algorithms to calculate the divisorial gonality of graphs. In [8] an algorithm based on Theorem 3.2.3 is presented, but we suspect faster algorithms may be possible using the more recently found lower bound using treewidth from [9]. Overall divisorial gonality, and more generally gonality, is a fairly new and not yet widely researched subject. It still has a large number of interesting open questions, both related to the measure itself and to its applications.

Bibliography

[1] Stefan Arnborg and Andrzej Proskurowski. “Characterization and recognition of partial 3-trees”. In: *SIAM Journal on Algebraic Discrete Methods* 7.2 (1986), pp. 305–314.

[2] Matthew Baker. “Specialization of linear systems from curves to graphs”. In: *Algebra & Number Theory* 2.6 (2008), pp. 613–653.

[3] Matthew Baker and Serguei Norine. “Riemann-Roch and Abel-Jacobi theory on a finite graph”. In: *Adv. Math.* 215.2 (2007), pp. 766–788.

[4] Jelco M. Bodewes, Hans L. Bodlaender, Gunther Cornelissen, and Marieke van der Wegen. “Recognizing hyperelliptic graphs in polynomial time”. arXiv: 1706.05670. 2017.

[5] Hans L. Bodlaender. “Treewidth: characterizations, applications, and computations”. In: *Proceedings of the 32nd international conference on Graph-Theoretic Concepts in Computer Science*. 2006, pp. 1–14.

[6] Gunther Cornelissen, Fumiharu Kato, and Janne Kool. “A combinatorial Li–Yau inequality and rational points on curves”. In: *Mathematische Annalen* 361.1-2 (2015), pp. 211–258.

[7] Bruno Courcelle. “The monadic second-order logic of graphs. I. Recognizable sets of finite graphs”. In: *Information and computation* 85.1 (1990), pp. 12–75.

[8] Josse van Dobben de Bruyn. “Reduced divisors and gonality in finite graphs”. Bachelor thesis, Leiden University. 2012. URL: https://www.math.leidenuniv.nl/nl/theses/316/.

[9] Josse van Dobben de Bruyn and Dion Gijswijt. “Treewidth is a lower bound on graph gonality”. arXiv: 1407.7055. 2014.

[10] Dion Gijswijt. “Computing divisorial gonality is hard”. arXiv: 1504.06713. 2015.

[11] Kevin Hendrey. “Sparse graphs of high gonality”. arXiv: 1606.06412. 2016.

[12] Petr Hliněny, Sang-il Oum, Detlef Seese, and Georg Gottlob. “Width parameters beyond tree-width and their applications”. In: *The Computer Journal* 51.3 (2007), pp. 326–362.

[13] Sergey Norin. “New tools and results in graph minor structure theory.” In: *Surveys in Combinatorics* 424 (2015), pp. 221–260.

[14] Daniel P. Sanders. “On linear recognition of tree-width at most four”. In: *SIAM Journal on Discrete Mathematics* 9.1 (1996), pp. 101–117.

[15] Marieke van der Wegen. “Stable gonality of graphs”. Utrecht University. MA thesis. 2017. URL: https://dspace.library.uu.nl/handle/1874/354700.