# Stable gonality of graphs

Author:
Marieke van der Wegen

Supervisors:
Gunther Cornelissen
Hans L. Bodlaender

Master's thesis
Department of Mathematics
Utrecht University
February-August 2017

# Acknowledgements

First, I would like to thank Gunther Cornelissen and Hans Bodlaender for being my supervisors. They gave me an interesting and enjoyable topic, we had weekly meetings where they gave me useful advice and they gave comments on this thesis to improve it.

Second, I would like to thank Jelco Bodewes, who is writing his thesis about gonality as well. I thank him for the many fruitful conversations we had. Together with Gunther and Hans we wrote a joint preprint. I consider this a very successful cooperation.

Last, I would like to thank all fellow students who work in the library of the mathematical institute, with whom I enjoyed lunch breaks and with whom I talked about my thesis. Especially I would like to thank Sophie Huiberts for the ‘sandpit seminars’ and Merlijn Staps and Sven Bosman for reading parts of my thesis.

# Contents


**Introduction** **1**

**1 Preliminaries** **3**

1.1 Divisorial gonality . . . . . 3
1.2 Stable divisorial gonality . . . . . 9
1.3 Geometric and stable gonality . . . . . 10
1.4 Basic results . . . . . 13

**2 Bounds for gonality** **19**

2.1 Inequalities relating different notions of gonality . . . . . 19
2.2 Treewidth is a lower bound for gonality . . . . . 23

**3 Recognizing hyperelliptic graphs** **27**

3.1 Reduction rules . . . . . 27
3.2 Reduction rules for stable gonality . . . . . 28
3.3 Reduction rules for stable divisorial gonality . . . . . 38
3.4 Algorithms using the reduction rules . . . . . 44

**4 Complexity of computing gonality** **46**

4.1 Complexity theory . . . . . 46
4.2 Stable divisorial gonality is NP-hard . . . . . 47

**Conclusion** **52**

**Bibliography** **53**

# Introduction

Gonality is a concept that is motivated by algebraic geometry, and that measures the complexity of a multigraph. In this sense it is comparable to treewidth. But unlike treewidth, gonality does not only depend on the underlying simple graph of a multigraph.

There are four different notions of gonality for graphs. Divisorial gonality can be defined using a chip-firing game (Section 1.1). Geometric gonality is defined as the smallest degree of a harmonic morphism from a graph to a tree (Section 1.3). For both these versions there exists a stable variant, which asks for the minimal gonality over all refinements of the graph (Sections 1.2 and 1.3)).

In 2007, Baker and Norine [7] defined the notion of divisors on graphs. Based on this concept, Baker [6] introduced divisorial gonality in 2008. In 2000, Urakawa [31] defined harmonic morphisms of simple graphs; in 2009, Baker and Norine [8] extended this to multigraphs. Caporaso [17] extended this to indexed harmonic morphisms in 2014, and used it to define geometric gonality. The stable versions were introduced by Cornelissen, Kato and Kool in 2015 [20].

We mention some known results about gonality. Firstly, there are relations between the different notions of gonality and treewidth [20, 24]. Specifically, if $G$ is a graph and we use $\mathrm{dgon}(G)$ to denote its divisorial gonality, $\mathrm{sdgon}(G)$ to denote its stable divisorial gonality, $\mathrm{gon}(G)$ to denote its geometric gonality and $\mathrm{sgon}(G)$ to denote its stable geometric gonality, then we know that:

$$\begin{cases} \mathrm{dgon}(G) \geq \mathrm{sdgon}(G) \geq \mathrm{tw}(G) \\ \mathrm{gon}(G) \geq \mathrm{sgon}(G) \geq \mathrm{sdgon}(G). \end{cases}$$

However, the divisorial gonality of a graph cannot be bounded from above by a function of treewidth, since there are graphs $G$ with $\mathrm{tw}(G) = 2$ and arbitrarily high divisorial gonality [27]. Secondly, computing the divisorial gonality of a graph is NP-complete and in XP [26].

Stable versions of gonality are important in number theory to say something about the finiteness of the number of solutions of polynomial equations, cf. [20]. All notions are interesting in theoretical computer science. It is possible that there are hard problems that are not tractable for bounded treewidth, but are tractable for bounded gonality.

In this thesis we study various aspects of these notions of gonality. In Chapter 1, we give definitions and show some basic results. In the second chapter, we study the relation between these notions of gonality and we give proofs for the relations mentioned earlier. In the third chapter, we show our original work:

**Theorem.** *There are sets of reduction rules to recognize graphs of stable gonality 2 and to recognize graphs of stable divisorial gonality 2.*

These sets can be used for an algorithm to decide in $O(n\log(n)+m)$ time whether the stable (divisorial) gonality of a graph equals 2 [14]. In the last chapter we prove a new theorem on the complexity of gonality:

**Theorem.** *Computing the stable divisorial gonality of a graph is NP-hard.*

Part of my thesis work is contained in a joint preprint [14]. I reused some parts of that paper for this thesis. As a consequence the following results in this thesis were originally written by Jelco Bodewes: the definition of divisorial gonality, section 3.1 and the proofs of Lemma 3.3.2, 3.3.3 and 3.3.13.

## Prerequisites

Whenever we write "graph" we refer to a finite undirected multigraph. Recall that a multigraph $G = (V, E)$ can have parallel edges and loops, i.e., $V$ is a set of vertices and $E$ is a multiset of edges. In this thesis, we will count a loop $vv$ twice in the degree of the vertex $v$. We call a graph without loops a *loopless* graph.

For a vertex $v$, we write $E_v$ for the set of edges that are incident to $v$. For a set $A \subset V$, we write $E(A)$ for the set of edges with both endpoints in $A$, and $E(A, V\backslash A)$ for the set of edges with one endpoint in $A$ and one endpoint in $V\backslash A$. For a set $A \subseteq V$ and a vertex $v \in A$, we write $\text{outdeg}_A(v)$ for the number of neighbours of $v$ outside $A$.

Some definitions that we will use in this thesis are the following:

**Definition.** Let $G = (V, E)$ be a graph, and $U \subseteq V$. The *induced subgraph* on $U$ is the graph $H = (U, F)$, where for any $u, v \in U$ we have $uv \in F$ if and only if $uv \in E$.

**Definition.** Let $G = (V, E)$ be a graph and $v, u \in V$. When we remove $v$, $G$ can fall apart in several connected components, let $U$ be the set of vertices in the connected component that contains $u$. By $G_v(u)$ we denote the induced subgraph of $G$ on $U \cup \{v\}$.

**Definition.** Let $G = (V, E)$ be a graph. Let $U \subseteq V$, we write $G\backslash U$ for the induced subgraph on $V\backslash U$. Let $H = (W, F)$ be a subgraph of $G$, we write $G\backslash H$ for the induced subgraph on $V\backslash W$.

**Definition.** Let $G$ be a graph. The adjacency matrix of $G$ is the $n \times n$-matrix $A$ where $A_{v,v}$ is equal to two times the number of loops $vv$, and $A_{u,v}$ is the number of edges between $u$ and $v$ if $u \neq v$.

# 1 Preliminaries

In this chapter we introduce the main concepts of this thesis: four different notions of gonality. We first give the definitions of all notions, and after that we will show some basic results about gonality.

## 1.1 Divisorial gonality

Divisorial gonality can be understood using a chip firing game. A chip-firing game starts with a distribution of chips over the vertices of a graph: to each vertex $v$ a non-negative number of chips, $D(v)$, is assigned. We can *fire* a vertex $v$ by moving chips from this vertex to its neighbours. We move one chip for each incident edge, so the number of chips that a neighbour $w$ of $v$ receives is equal to the number of edges $vw$. When the number of chips on a vertex is negative, we consider this vertex to be in debt.

In 1991, Björner, Lovász and Shor [12] introduced a chip-firing game where no vertex is allowed to be in debt, so we can only fire a vertex when it has more chips than its degree. There are some variants of this game, for example a game on directed graphs [11] or a game where one specific vertex is allowed to be in debt [10].

Chip-firing games appear in various fields of research. Independently, in statistical physics a similar chip-firing game was defined by Bak, Tang and Wiesenfeld [5] and Dhar [22]; they called it the abelian sandpile. The complexity of some problems related to these abelian sandpiles is studied in [16] and [28]. Chip-firing games have links with spanning trees in a graph [18]. There are also links with potential theory, where graphs are considered as electrical networks [9].

In 2007, Baker and Norine [7] studied the similarity between Riemann surfaces and graphs. They introduced a new chip-firing game that led to the definition of divisorial gonality. In this game, all vertices are allowed to be in debt. The divisorial gonality is the minimum number $k$ of chips, such that there is an initial configuration with $k$ chips with the following property: for any vertex $v$ in the graph, there exists a sequence of firings that results in vertex $v$ having at least one chip and no other vertex being in debt.

### Formal definition

We now give a formal definition for divisorial gonality, based on the concepts of [7]:

**Definition 1.1.1.** Let $G$ be a graph. A *divisor* $D$ on $G$ is an element of $\bigoplus_{V(G)} \mathbb{Z}$, so it is a vector consisting of $n$ integers indexed by the vertices of $G$. We use $D(v)$ to denote the integer assigned to vertex $v$ by the divisor $D$. We call a divisor $D$ *effective* 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 of $D(v)$ over all $v \in V(G)$. By $\mathrm{Div}^k(G)$ we denote the set of all divisors with degree $k$.

**Definition 1.1.2.** Let $G$ be a graph with $n$ vertices. The *Laplacian matrix* $\mathcal{L}$ of $G$ is defined as the $n \times n$-matrix $\mathcal{L} = D - A$, 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 1.1.3.** We call two divisors $D$ and $D'$ *equivalent*, denoted by $D \sim D'$, if there exists a principal divisor $P$ such that $D' = D - P$. We call $P$ the *transformation* of $D$ into $D'$. Given a divisor $D$, we have a class of equivalent effective divisors $|D| = \{D' \in \mathrm{Div}_+(G) \mid D \sim D'\}$.

Notice that this relation indeed is an equivalence relation. We see that the class $|D|$ is the intersection of the equivalence class of $D$ with the set of all effective divisors. If $D$ is effective then $|D| \neq \emptyset$, but if $D$ is non-effective, then possibly $|D| = \emptyset$.

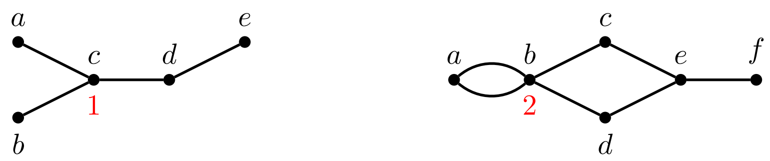
**Definition 1.1.4.** The *rank* of a divisor $D$ is denoted by $r(D)$ and defined as follows:

$$r(D) = \begin{cases} \max\{k \mid |D - E| \neq \emptyset \text{ for all } E \in \mathrm{Div}_+^k(G)\} & \text{if } |D| \neq \emptyset; \\ -1 & \text{if } |D| = \emptyset. \end{cases}$$

**Definition 1.1.5.** The *divisorial gonality* of a graph $G$, denoted by $\mathrm{dgon}(G)$, is the lowest degree for which there exists an effective divisor of rank at least one, i.e.,

$$\mathrm{dgon}(G) = \min\{\deg(D) \mid D \in \mathrm{Div}_+(G), r(D) \geq 1\}.$$

This definition is indeed equivalent to the more intuitive definition that we gave: the set $\mathrm{Div}(G)$ describes all possible distributions of chips over the vertices and $\mathrm{Div}^k(G)$ consists of all possible distributions of $k$ chips. A divisor $D$ can also describe how often we fire each vertex, the Laplacian matrix then describes how chips are moved: every vertex $v$ receives $-(\mathcal{L}D)(v)$ chips. We see that the set $\mathrm{Prin}(G)$ is the set of all transformations that can be obtained by firing vertices. The rank of a divisor tells us how many chips can be removed such that it is still possible to reach a distribution where no vertex is in debt. If the rank is at least 1, this means that we can remove a chip from any vertex and still reach a distribution in which no vertex is in debt. This is equivalent to the condition 'for any vertex $v$ in the graph, there exists a sequence of firings that results in vertex $v$ having at least one chip and no other vertex being in debt' that we mentioned earlier.




(a) The tree of example 1.1.6.

(b) The graph of example 1.1.8.

Figure 1.1: The numbers show divisors with rank at least 1.

**Example 1.1.6.** Let $G$ be the tree in Figure 1.1(a). Suppose that we have one chip on vertex $c$, and that we want to have one chip on vertex $e$. The edges on the unique path from $c$ to $e$ are $cd$ and $de$. We can move a chip along the edge $cd$ by firing $a$, $b$ and $c$ once. After that, we can move the chip along the edge $de$ by firing $a$, $b$, $c$ and $d$ once. In total we have fired $a$, $b$ and $c$ twice and $d$ once. This corresponds with the divisor $(2,2,2,1,0)^t$. The Laplacian of this tree is

$$\mathcal{L} = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}.$$

And we see that

$$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 2 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

So we see that firing $(2,2,2,1,0)^t$ indeed yields a transformation from the divisor with a chip on $c$ to the divisor with a chip on $e$.

**Example 1.1.7.** Let $G$ be a tree. Then $G$ has divisorial gonality 1. Notice that we can move a chip over an edge $e = uv$ from $u$ to $v$ as follows: fire every vertex in $A_u$ exactly once, where $A_u$ is the connected component of $u$ of the induced cut of $e$, as in Example 1.1.6.

Let $v$ be a vertex of $G$ and let $D$ be the divisor with one chip on $v$. Let $u \neq v$ be a vertex of $G$ and let $D'$ be the divisor with one chip on $u$. The divisors $D$ and $D'$ are equivalent: There is a unique path from $v$ to $u$ in $G$, and we can move a chip over every edge of this path. We see that $D$ has rank at least 1. Since $\deg(D) = 1$, we conclude that $\mathrm{dgon}(G) = 1$.

**Example 1.1.8.** Let $G$ be the graph in Figure 1.1(b). Consider the divisor $D = (0,2,0,0,0,0)^t$ with two chips on vertex $b$. For each vertex $v$ we can find vertices such that firing them yields a divisor with at least one chip on $v$ and all other vertices not in debt. To get a chip on $a$ we can fire each of the vertices $b, c, d, e, f$ once. To get a chip on $c$ we can fire $a$ and $b$ once. We then have the divisor $D' = (0,0,1,1,0,0)^t$. Notice that in this divisor we also have a chip on $d$. To reach a divisor with a chip on $e$ we can fire $a$, $b$, $c$ and $d$ once, starting from $D'$, to obtain the divisor $D'' = (0,0,0,0,2,0)^t$. Lastly, to reach a divisor with a chip on $f$ from $D''$, we can fire $a, b, c, d$ and $e$. So we see that $D$ has rank at least 1, thus $\text{dgon}(G) \leq 2$.

We can prove that the only graphs with divisorial gonality 1 are trees, thus the graph $G$ in Example 1.1.8 has divisorial gonality 2. To show this, we will first introduce an alternative definition of divisorial gonality.

## An equivalent definition of divisorial gonality

We will now look at a slightly different chip firing game, which yields us an alternative definition of divisorial gonality. Again we have a graph and an initial distribution of chips such that every vertex has a non-negative number of chips. In this game we will fire subsets $A$ of vertices by moving one chip along each outgoing edge of $A$, provided that there are no vertices going in debt. If a vertex in $A$ has more neighbours outside $A$ than it has chips, we are not allowed to fire $A$. The divisorial gonality is the minimum number of chips such that there is an initial configuration with $k$ chips with the following property: for any vertex $v$ in the graph, there exists an increasing sequence of sets, such that firing these sets results in vertex $v$ having at least one chip.

Formally, we can write the following, where we use $\mathbf{1}_A$ to denote the divisor with $\mathbf{1}_A(v) = 1$ if $v \in A$ and $\mathbf{1}_A(v) = 0$ otherwise:

**Definition 1.1.9.** We call two effective divisors $D, D'$ *s-equivalent*, denoted by $D \sim_s D'$, if there are sets $A_0 \subseteq \ldots \subseteq A_k \subseteq V(G)$ and divisors $D_1 = D - \mathcal{L}\mathbf{1}_{A_0}$, $D_i = D_{i-1} - \mathcal{L}\mathbf{1}_{A_{i-1}}$, $i \in \{2, \ldots, k+1\}$, such that

- $D_i$ is effective for all $i \in \{1, \ldots, k\}$,
- $D_{k+1} = D'$.

**Definition 1.1.10.** Let $G$ be a graph and $D$ an effective divisor, we call a non-empty firing set $A$ *valid for* $D$ if $D(a) \geq \text{outdeg}_A(a)$ for all $a \in A$. If it is clear from the context what divisor $D$ we use, we simply call $A$ *valid*.

This equivalence relation tells us which divisors we can transform into each other by firing an increasing sequence of sets of vertices such that in all intermediate steps the divisor is effective. A set is valid exactly when we can fire it according to this chip firing game.

We will show that, on the set of effective divisors, this equivalence relation is equal to the one we gave earlier, in Definition 1.1.3, i.e., for two effective divisors $D, D'$, it holds that $D \sim D'$ if and only if $D \sim_s D'$. In order to prove this we will first show a construction, called level set decomposition, and some properties of it.

**Definition 1.1.11** ([23, Definition 3.7])**.** Let $G$ be a graph and let $D, D'$ be two equivalent divisors. Then, by definition, there exists a divisor $C$ such that $D' = D - \mathcal{L}C$, where $\mathcal{L}$ is the Laplacian matrix. Let $m = \max\{C(v) \mid v \in V(G)\}$ and let $k = m - \min\{C(v) \mid v \in V(G)\}$. We then define the *level sets* of $C$ as follows:

$$A_i = \{v \in V(G) \mid C(v) \geq m - i\} \text{ for } i \in \{0, \ldots, k\}.$$

Notice that for level sets $A_0, A_1, \ldots, A_k$ of a divisor $C$, the set $A_{k-i}$ contains exactly the vertices that are fired at least $i + \min\{C(v) \mid v \in V(G)\}$ times. It follows that $A_0 \subseteq A_1 \subseteq \cdots \subseteq A_k$. Moreover, for the transformation of a divisor $D$ into a divisor $D'$ the level sets $A_0, \ldots, A_k$ are unique. Before we prove this, we show a lemma about the null space of the Laplacian matrix.

**Lemma 1.1.12** ([15, Proposition 1.3.7])**.** *Let $G$ be a connected graph. The null space of $\mathcal{L}$ is spanned by the all-ones vector, i.e.* $\ker(\mathcal{L}) = \langle \mathbf{1} \rangle$*, where* $\mathbf{1}$ *is the all-ones vector.*

*Proof.* First, we introduce the directed incidence matrix. For every edge of $G$, choose an orientation. The directed incidence matrix $N$ is a matrix, where the rows are indexed by the vertices and the columns are indexed by the edges without the loops, defined by:

$$N_{v,e} = \begin{cases} -1 & \text{if } v \text{ is the head of } e, \\ 1 & \text{if } v \text{ is the tail of } e, \\ 0 & \text{otherwise.} \end{cases}$$

Now look at $NN^t$. We see that the value of $(NN^t)_{v,v}$ is the degree of $v$ minus two times the number of loops $vv$. The value of $(NN^t)_{u,v}$ for $u \neq v$ equals minus the number of edges $uv$. Thus $NN^t = \mathcal{L}$.

Let $x$ be a vector in the null space of $\mathcal{L}$. Then it follows that $\mathcal{L}x = 0$, thus $x^t\mathcal{L}x = 0$ and $x^tNN^tx = 0$. We can rewrite this as $(N^tx)^t(N^tx) = 0$. It follows that $N^tx = 0$. Let $e = uv$, $u \neq v$, be an edge of $G$. Suppose that we oriented $uv$ from $u$ to $v$. It follows that $(N^tx)_e = x_u - x_v = 0$. Thus $x_u = x_v$. Since $G$ is connected, it follows that $x = \alpha\mathbf{1}$ for some $\alpha \in \mathbb{R}$, where $\mathbf{1}$ is the all-ones vector.

Since every row of $\mathcal{L}$ sums up to 0, we see that the all-ones vector is indeed in the null space of $\mathcal{L}$, thus the null space of $\mathcal{L}$ is spanned by the all-ones vector. □

**Lemma 1.1.13** ([23, Remark 3.8])**.** *Let $G$ be a connected graph and let $D, D'$ be two equivalent divisors. The level set decomposition of the transformation from $D'$ to $D$ is unique. That is, if there are two divisors $C_1, C_2$ such that $D' = D - \mathcal{L}C_1 = D - \mathcal{L}C_2$, then the level sets of $C_1$ and $C_2$ are equal.*

*Proof.* Let $C_1, C_2$ be two divisors such that $D' = D - \mathcal{L}C_1 = D - \mathcal{L}C_2$. We can rewrite this to $\mathcal{L}(C_1 - C_2) = 0$, it follows that $C_1 - C_2$ is in the null space of $\mathcal{L}$.

Now we see, by Lemma 1.1.12, that $C_1 = C_2 + \alpha\mathbf{1}$ for some $\alpha$. By the construction of the level sets it is clear that the level sets of $C_1$ and $C_2$ are equal. □

*Remark.* Let $D \sim D'$ be two equivalent divisors. By Lemma 1.1.13, the level sets of the transformation from $D$ into $D'$ are well defined. We will often call the level sets the *level set decomposition* of the transformation.

**Definition 1.1.14** ([23, Definition 3.7])**.** A level set decomposition $A_0, \ldots, A_k$ belonging to a transformation of $D$ into $D'$ has an associated *sequence of divisors*

- $D_0 = D$,
- $D_i = D_{i-1} - \mathcal{L}\mathbf{1}_{A_{i-1}}$ for all $i \in \{0, \ldots, k+1\}$.

Since $A_0, \ldots, A_k$ are the level sets of the transformation from $D'$ to $D$, it follows that $D_{k+1} = D'$. We continue with a useful property of level set decompositions, namely the fact that each divisor in the associated sequence is bounded from below by the pointwise minimum of $D$ and $D'$.

**Lemma 1.1.15** ([23, Theorem 3.10])**.** *Let $G$ be a graph, $D, D'$ be two equivalent divisors and $A_0, \ldots, A_k$ the level set decomposition of the transformation of $D$ into $D'$. Let $D_0, \ldots, D_k$ be the associated sequence of divisors. We then have that, for all $v \in V(G)$ and all $i \in \{0, \ldots, k\}$,*

$$D_i(v) \geq \min(D(v), D'(v)).$$

*Proof.* Let $i \in \{0, \ldots, k\}$ and $v \in V(G)$. If $D_i(v) \geq D(v)$ we are done, so assume that $D_i(v) < D(v)$. Since $D_i(v) < D(v)$ and since the only way a vertex can lose chips is by firing, $v$ must have been fired at least once before $D_i$. So there is an $A_j$ with $j < i$ such that $v \in A_j$. But since $A_0 \subseteq \ldots \subseteq A_k$, it follows 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 $A_i$. However, 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 $m \geq i$. In particular, we have that $D_i(v) \geq D_k(v) = D'(v)$. We conclude that $D_i(v) \geq \min(D(v), D'(v))$ for all $v \in V(G)$ and $i \in \{0, \ldots, k\}$. □

Lemma 1.1.15 immediately gives a result for the transformation between two equivalent effective divisors.

**Corollary 1.1.16** ([23, Corollary 3.11])**.** *Let $G$ be a graph and $D, D'$ be two equivalent effective divisors. Let $A_0, \ldots, A_k$ be the level set decomposition of the transformation of $D$ into $D'$ and $D_0, \ldots, D_k$ the associated sequence of divisors. Then all divisors $D_i$ are effective.* □

Now we are ready to prove that the two equivalence relations are the same on effective divisors.

**Lemma 1.1.17.** *Let $G$ be a graph and $D, D'$ two effective divisors on $G$. Then $D \sim D'$ if and only if $D \sim_s D'$.*

*Proof.* Suppose that $D \sim_s D'$. Then there exist sets $A_0 \subseteq \ldots \subseteq A_k$ and divisors $D_1 = D - \mathcal{L}\mathbf{1}_{A_0}$, $D_i = D_{i-1} - \mathcal{L}\mathbf{1}_{A_{i-1}}$ such that $D_{k+1} = D'$. We see that $D' = D_{k+1} = D - \mathcal{L}(\mathbf{1}_{A_0} + \ldots + \mathbf{1}_{A_{k+1}})$. We conclude that $D \sim D'$.

Suppose that $D \sim D'$. Let $A_0, \dots, A_k$ be the level set decomposition of the transformation of $D$ into $D'$ and let $D_0, \dots, D_k$ be the associated sequence of divisors. We have seen that $A_0 \subseteq \dots \subseteq A_k$, that $D_{k+1} = D'$ and that the divisors $D_0, \dots, D_k$ are effective. We conclude that $D \sim_s D'$. □

From now on we can use the level set decomposition to argue about divisors and divisorial gonality. This will turn out to be useful for giving proofs concerning divisorial gonality.

**Example 1.1.18.** Let $G$ be a tree. We have seen that we can move a chip along an edge $e = uv$ from $u$ to $v$ by firing every vertex in $A_u$ exactly once, where $A_u$ is the connected component of $u$ of the cut induced by $e$. Using our new definition we can just fire $A_u$ instead of firing every vertex separately.

**Example 1.1.19.** We have seen that trees have divisorial gonality 1. Now we can argue that all other graphs have divisorial gonality at least 2. Let $G$ be a graph with a cycle $c_1, c_2, \dots, c_k$ with $k > 1$. Suppose that $G$ has divisorial gonality 1. Let $D_1$ be the divisor with a chip on $c_1$, and $D_k$ the divisor with a chip on $c_k$. Then $D_1 \sim D_k$. Thus there exists a level set decomposition $A_0, \dots A_l$ of the transformation of $D_1$ into $D_k$. Since $c_1$ is the only vertex with a chip in $D_1$, it follows that $c_1 \in A_0$. The number of chips on vertex $c_k$ increases by firing the sets $A_0, \dots A_l$, thus $c_k \notin A_0$. Now look at the greatest index $j$ such that $c_j \in A_0$. Then we see that firing $A_0$ moves a chip along the edge $c_j c_{j+1}$, and moves a chip along the edge $c_1 c_k$ as well. Because there is only one chip and no vertex is allowed to go in debt, this yields a contradiction. So we conclude that $\mathrm{dgon}(G) \geq 2$.

## 1.2 Stable divisorial gonality

Now we move on to the definition of stable divisorial gonality. In order to do so we need one additional definition.

**Definition 1.2.1** ([20, Definition 3.4])**.** A graph $G'$ is a *refinement* of $G$ if $G'$ can be obtained by applying the following operations finitely many times to $G$.

(i) Add a leaf, i.e. a vertex of degree 1;

(ii) subdivide an edge by adding a vertex.

We call a vertex of $G' \backslash G$ from which there are two disjoint paths to vertices of $G$, *internal added vertices*, we call the other vertices of $G' \backslash G$ *external added vertices*.

**Definition 1.2.2** ([20, A.6])**.** The *stable divisorial gonality* of $G$ is

$$\mathrm{sdgon}(G) = \min\{\mathrm{dgon}(G') \mid G' \text{ a refinement of } G\}.$$

Notice that for all graphs $G$ it holds that $\mathrm{sdgon}(G) \leq \mathrm{dgon}(G)$.

**Example 1.2.3.** A tree has divisorial gonality 1; thus, its stable divisorial gonality is 1 as well. Notice that refining a graph does not change whether a graph is a tree or not. Thus if a graph $G$ is not a tree, then it follows that $\mathrm{dgon}(G') \geq 2$ for all refinements $G'$. We conclude that $\mathrm{sdgon}(G) \geq 2$; and the stable divisorial gonality of a graph $G$ is 1 if and only if $G$ is a tree.


Figure 1.2: The graph $G$ of Example 1.2.5 with $\mathrm{dgon}(G) = 4$ and $\mathrm{sdgon}(G) \leq 3$.

**Example 1.2.4.** Let $G$ be a graph with divisorial gonality 2, for example the graph in Figure 1.1(b). It follows that $\mathrm{sdgon}(G) = 2$.

**Example 1.2.5** ([17, Examples 4 and 5])**.** Consider the graph $G$ in Figure 1.2. We claim that $\mathrm{dgon}(G) = 4$ and $\mathrm{sdgon}(G) \leq 3$.

First we determine the divisorial gonality of $G$. Suppose that $\mathrm{dgon}(G) \leq 3$. Let $D$ be a divisor with rank at least 1 and degree 3. We know that there is a divisor $D_a \sim D$ with $D_a(a) \geq 1$. Suppose that $D_a(d) = 0$. There is a divisor $D_d \sim D_a$ such that $D_d(d) \geq 1$. Let $A_0, \ldots, A_k$ be the level set decomposition of the transformation of $D_a$ into $D_d$ and $D_1, \ldots, D_{k+1}$ the associated sequence of divisors. The number of chips on vertex $d$ increases by firing the sets $A_0, \ldots, A_k$, so we see that $d \notin A_0$. We distinguish three cases.

Suppose that $D_a(a) < 3$ and $D_a(b) < 2$. We can easily check that all sets $A \subseteq \{a, b, c\}$ are not valid; thus $A_0$ is not valid. This yields a contradiction.

Suppose that $D_a(a) < 3$ and $D_a(b) = 2$. We can easily check that $\{a, b\}$ is the only valid set, so $A_0 = \{a, b\}$. We see that $D_1(d) = 0$, thus $d \notin A_1$. Since $A_0 \subseteq A_1$, there are only two possibilities left: $A_1 = \{a, b\}$ or $A_1 = \{a, b, c\}$. Both sets are not valid, this yields a contradiction.

Suppose that $D_a(a) = 3$. Analogously we find that $A_0 = \{a\}$, $A_1 = \{a, b\}$ and that $A_2$ is not valid. This yields a contradiction.

We conclude that $D_a(d) > 0$. It follows that $D_a(b) = 0$ or $D_a(c) = 0$. Without loss of generality, we assume that $D_a(b) = 0$. Then there is a divisor $D_b \sim D_a$ with $D_b(b) \geq 1$. Let $A_0$ be the first set in the level set decomposition of the transformation of $D_a$ into $D_b$. Since the number of chips on $b$ increases by this transformation, it follows that $b \notin A_0$. We can easily check that there is no valid firing set $A_0 \subseteq \{a, c, d\}$. This yields a contradiction.

We conclude that $\mathrm{dgon}(G) > 3$. Starting with a chip on every vertex yields a divisor with rank at least 1 and degree 4, so $\mathrm{dgon}(G) = 4$.

The stable divisorial gonality of this graph is at most 3. Add a vertex $e$ on one of the edges from $b$ to $c$. Call the graph we obtained $G'$. Now assign 3 chips to vertex $a$. This is an effective divisor. It has rank at least 1: firing the sets $\{a\}$, $\{a, b\}$, $\{a, b, e\}$, $\{a, b, c, e\}$ consecutively leads to divisors $D_1, \ldots, D_4$ such that every vertex has at least a chip once. Thus $\mathrm{sdgon}(G) \leq 3$.

## 1.3 Geometric and stable gonality

In [17] and [20] the relation between algebraic curves and graphs is further studied and new notions of gonality are defined. We will define geometric gonality as in [17] using the terminology of [20].

**Definition 1.3.1.** Let $G$ and $H$ be loopless graphs. A *morphism* is a map $\phi : G \to H$ such that

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

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

together with, for every $e \in E(G)$, an *index* $r_\phi(e) \in \mathbb{N} \cup \{0\}$, such that $r_\phi(e) = 0$ if and only if $\phi(e) \in V(H)$.

**Definition 1.3.2.** We call a morphism $\phi : G \to H$ *non-degenerate* if for every vertex $v \in V(G)$ there is an edge $e \in E_v(G)$ such that $\phi(e) \neq \phi(v)$.

**Definition 1.3.3.** We call a 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.

Notice that a harmonic morphism is non-degenerate exactly when $m_\phi(v) \neq 0$ for all $v \in V(G)$.

**Definition 1.3.4.** The *degree* of a harmonic morphism $\phi : G \to H$ is

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

for $e \in E(H)$, $v \in V(H)$. This is independent of the choice of $e$ or $v$ [8, Lemma 2.4].

**Definition 1.3.5.** Let $G$ be a loopless graph. The *geometric gonality* of $G$ is

$$\begin{aligned}\mathrm{gon}(G) = \min\{\deg(\phi) \mid \phi : G \to T \text{ a non-degenerate harmonic}\\ \text{morphism from } G \text{ to a tree } T\}.\end{aligned}$$

*Remark.* Let $G$ be the graph consisting of a single vertex. Notice that there does not exist a non-degenerate morphism from $G$ to a tree $T$. We define the geometric gonality of $G$ to be 1.

We define stable geometric gonality, or stable gonality for short, as in [20, Definition 3.6].

**Definition 1.3.6.** We call a morphism $\phi : G \to H$ a *finite morphism* if $\phi(E(G)) \subseteq E(H)$.

It follows that for a finite morphism $\phi$ it holds that $r_\phi(e) \neq 0$ for all edges $e \in E(G)$.

**Definition 1.3.7.** The *stable gonality* of a graph $G$ is

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

Notice that, although a finite morphism is only defined for loopless graphs, we define stable gonality for all graphs, since we can add a vertex to every loop to obtain a loopless refinement.

For a disconnected graph, we map all components to the same tree; it follows that the stable gonality of a disconnected graph is the sum of the stable gonality of its components. One could argue that it is better to map disconnected graphs to a forest instead. Then one could define the degree of such a map to be the maximum of the degrees of this map restricted to each component. In that case the stable gonality of a disconnected graph is the maximum of the stable gonality of its components. Nevertheless, we use the definition that maps graphs to trees.

**Example 1.3.8.** For a tree $G$ we can assign to every edge index 1 and use the identity map to see that $\mathrm{gon}(G) = \mathrm{sgon}(G) = 1$.

On the other hand, we can show that if $G$ is not a tree, then $\mathrm{gon}(G) \neq 1$ and $\mathrm{sgon}(G) \neq 1$: Let $G$ be a graph, containing a cycle $c_1, \ldots, c_k$. Let $\phi : G \to T$ be a non-degenerate harmonic morphism to a tree $T$. If $c_1$ and $c_k$ are mapped to the same vertex, then the degree of $\phi$ is at least 2. Suppose that $c_1$ and $c_k$ are not mapped to the same vertex. Notice that by definition $\phi$ maps paths from $u$ to $v$ to walks from $\phi(u)$ to $\phi(v)$. There are two disjoint paths $P_1$ and $P_2$ from $c_1$ to $c_k$. In $T$ there is a unique path $P$ from $\phi(c_1)$ to $\phi(c_k)$, thus any walk from $\phi(c_1)$ to $\phi(c_k)$ contains $P$. Thus there is at least one edge from $P_1$ and one edge from $P_2$ mapped to each edge of this path $P$. So the degree of $\phi$ is at least two. We see that $\mathrm{gon}(G) \geq 2$.

Let $G$ be a graph, not a tree, and let $G'$ be a refinement of $G$. Then $G'$ contains a cycle. Let $\phi : G' \to T$ be a finite harmonic morphism to a tree $T$. Analogously, we see that $\deg(\phi) \geq 2$. Thus $\mathrm{sgon}(G) \geq 2$.

**Example 1.3.9.** Consider the graph in Figure 1.3 (this is the same graph as in Figure 1.1(b)). We can map this graph to the path graph on five vertices as follows: $\phi(a) = p_1$, $\phi(b) = p_2$, $\phi(c) = \phi(d) = p_3$, $\phi(e) = p_4$ and $\phi(f) = p_5$, see Figure 1.3 for an illustration. Give the edge $ef$ index 2, and all other edges index 1. This is a finite morphism, since no edge is mapped to a vertex.

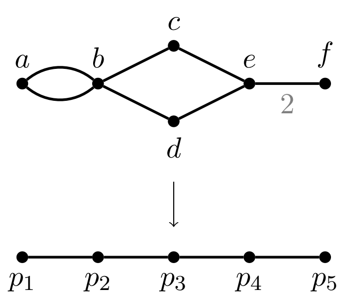



Figure 1.3: The graph $G$ of Example 1.3.9 and a finite harmonic morphism of degree 2.

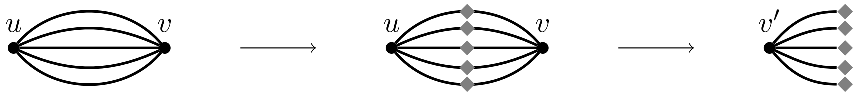



Figure 1.4: The banana graph has gonality $m$ and stable gonality 2.

We can also check that $\phi$ is harmonic. Consider, for example, vertex $e$. There are two edges incident to $\phi(e)$, namely $p_3p_4$ and $p_4p_5$. We can compute

$$\sum_{x \in E_e(G), \phi(x)=p_3p_4} r_\phi(x) = r_\phi(ce) + r_\phi(de) = 2,$$
$$\sum_{x \in E_e(G), \phi(x)=p_4p_5} r_\phi(x) = r_\phi(ef) = 2.$$

We see that these sums are indeed equal, and $m_\phi(e) = 2$. Analogously, we can check that $m_\phi(a) = m_\phi(b) = m_\phi(f) = 2$ and $m_\phi(c) = m_\phi(d) = 1$.

The degree of $\phi$ is $\sum_{x \in V(G), \phi(x)=p_4} m_\phi(x) = m_\phi(e) = 2$. So we conclude that $\mathrm{sgon}(G) \leq 2$. Since $G$ is not a tree, we see that $\mathrm{sgon}(G) \geq 2$, thus $\mathrm{sgon}(G) = 2$.

**Example 1.3.10** ([20, Example 3.9])**.** The banana graph $B_m$ is a graph with 2 vertices $u$ and $v$ and $m \geq 2$ edges, see Figure 1.4. Let $\phi : B_m \to T$ be a non-degenerate harmonic morphism to a tree $T$. Then $\phi(u) \neq \phi(v)$, otherwise $\phi$ is degenerate. It follows that all $m$ edges are mapped to the edge $\phi(u)\phi(v)$, thus $\deg(\phi) \geq m$. Let $T$ be a tree with two vertices $u', v'$ that are connected by an edge. The map $\phi : \phi(u) = u', \phi(v) = v'$ is a non-degenerate harmonic morphism of degree $m$. We conclude that $\mathrm{gon}(B_m) = m$.

The stable gonality of $B_m$ is much lower, namely 2. Consider the refinement $G'$ where every edge is subdivided. Let $T$ be a tree with a vertex $v'$ and $m$ leaves, see Figure 1.4. Let $\phi : G' \to T$ be the map such that $\phi(u) = \phi(v) = v'$ and all other vertices are mapped to a unique leaf. Give every edge of $G'$ index 1. Now we can see that $\phi$ is a finite harmonic morphism of degree 2. And it is clear that $B_m$ is not a tree, thus $\mathrm{sgon}(B_m) = 2$.

## 1.4 Basic results

In this section we mention some basic properties and results of gonality. In the previous sections, we have seen that for a disconnected graph, the stable gonality is equal to the sum of the stable gonality of each of the connected components. The same holds true for (stable) divisorial gonality, as chips can never move from one connected component to another. We have also seen a result on trees:

**Proposition 1.4.1.** *Let $G$ be a graph. The following are equivalent:*

(i) *$G$ is a tree,*

(ii) $\mathrm{dgon}(G) = 1$*,*

(iii) $\mathrm{sdgon}(G) = 1$*,*

(iv) $\mathrm{gon}(G) = 1$*,*

(v) $\mathrm{sgon}(G) = 1$*.*

*Proof.* This follows from Examples 1.1.7, 1.1.19, 1.2.3 and 1.3.8. □

For a graph $G$ that is not a tree, it follows that all gonalities are at least 2. We can also give simple upper bounds in the number of vertices and edges.

**Proposition 1.4.2.** *Let $G$ be a graph. It holds that* $\mathrm{dgon}(G) \leq n$ *and* $\mathrm{sdgon}(G) \leq n$*.*

*Proof.* It is easy to see that the (stable) divisorial gonality of a graph is at most $n$ by placing a chip on every vertex. □

**Proposition 1.4.3.** *Let $G$ be a graph. Let $l$ be the number of vertices with a loop. It holds that* $\mathrm{sgon}(G) \leq n + l$*.*

*Proof.* For every vertex $v$, let $l_v$ be the number of loops of $v$. Consider the graph $G'$, obtained by subdividing every edge once and adding $m - \deg(v) + l_v$ leaves to every vertex $v$ in $G$. See Figure 1.5 for an illustration. Notice that every vertex has $m$ added neighbours. Let $T$ be a tree consisting of a vertex $v$ with $m$ leaves. Then we can map this graph $G'$ to the tree $T$ by mapping every original vertex of $G$ to $v$, mapping all vertices that are added to an edge to a unique leaf, and mapping all added leaves such that for each original vertex its $m$ neighbours are mapped to the $m$ leaves. Assign index 2 to all edges $vx$, where $v$ is a vertex of $G$ with a loop $vv$ in $G$ and $x$ is not added to a loop, and index 1 to all other edges. We see that this is a finite harmonic morphism. Every vertex $v \in V(G)$ has index two if there is a loop $vv$ in $G$ and index 1 otherwise. Thus the degree of this morphism is $n + l$. □

The geometric gonality of a graph can be greater than $n$, consider the banana graph for example (see Figure 1.4 and Example 1.3.10).

**Proposition 1.4.4.** *Let $G$ be a connected graph. It holds that* $\mathrm{gon}(G) \leq m$*.*

*Proof.* Let $T$ be a tree consisting of two vertices $u, v$ and an edge $uv$. Pick a vertex $w \in V(G)$. We define the following map $\phi : G \to T$:

$$\phi(x) = \begin{cases} u & \text{if the shortest path from } w \text{ to } x \text{ has even length,} \\ v & \text{if the shortest path from } w \text{ to } x \text{ has odd length,} \end{cases}$$

here the length is the number of edges of the path. Assign index 1 to all edges that are mapped to $uv$ and 0 to all other edges.

For a vertex $x \in V(G)$, let $w = x_0, x_1, \ldots, x_k = x$ be a shortest path from $w$ to $x$. Then the edge $x_{k-1}x$ is mapped to $uv$. It follows that $\phi$ is non-degenerate. Since $T$ has only one edge, $\phi$ is harmonic. The degree of $\phi$ is at most $m$, since there are at most $m$ edges mapped to the edge $uv$. We conclude that $\mathrm{gon}(G) \leq m$. □

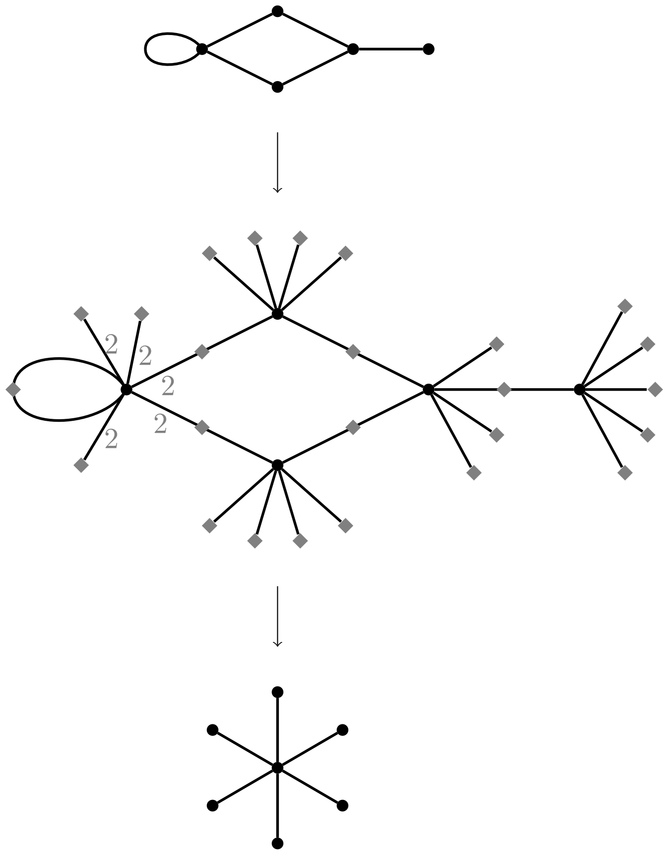



Figure 1.5: The stable gonality of a graph is at most $n + |\{v \mid \text{there is a loop } vv\}|$.

The stable divisorial gonality of a graph $G$ is defined as the minimum of the divisorial gonality over all refinements of $G$. We can prove that the stable gonality of a graph $G$ is equal to the minimum of the geometric gonality over all refinements of $G$ as well [20, Lemma A.3]:

**Proposition 1.4.5.** *Let $G$ be a graph. Then*

$$\operatorname{sgon}(G) = \min\{\operatorname{gon}(G') \mid G' \text{ a refinement of } G\}.$$

*Proof.* It is clear that $\operatorname{sgon}(G) \geq \min\{\operatorname{gon}(G') \mid G' \text{ a refinement of } G\}$, since every finite harmonic morphism is a non-degenerate harmonic morphism.

We will show that for any non-degenerate harmonic morphism $\phi : G' \to T$ from a refinement $G'$ of $G$ to a tree $T$, there exists a finite harmonic morphism $\phi'$ of the same degree from a refinement $G''$ of $G'$ to a tree $T'$. For every edge $e = uv$ such that $\phi(e) \in V(T)$, subdivide $e$ by adding a vertex $w_e$ and add a leaf $l_{e,x}$ to every vertex $x \in \phi^{-1}(\phi(e))$, $x \neq u, v$. Write $G''$ for the graph thus obtained. For every such $e$, add a leaf $l_e$ to $\phi(e)$ in $T$, to obtain a tree $T'$. Now set

$$r_{\phi'}(e') = \begin{cases} m_\phi(x) & \text{if } e' = xl_{e,x}, \\ m_\phi(u) & \text{if } e' = uw_e, \\ m_\phi(v) & \text{if } e' = w_e v, \\ r_\phi(e) & \text{otherwise.} \end{cases}$$

We define $\phi' : G'' \to T'$ by

$$\phi'(y) = \begin{cases} l_e & \text{if } y = l_{e,x}, w_e \\ \phi(y) & \text{otherwise.} \end{cases}$$

It is easy to check that $\phi'$ is a finite harmonic morphism with $\deg(\phi') = \deg(\phi)$. We conclude that $\operatorname{sgon}(G) = \min\{\operatorname{gon}(G') \mid G' \text{ a refinement of } G\}$. □

It is easy to see that $\operatorname{sdgon}(G') \geq \operatorname{sdgon}(G)$ for all refinements $G'$ of $G$. We will now prove that it also holds that $\operatorname{sdgon}(G') \leq \operatorname{sdgon}(G)$.

**Proposition 1.4.6.** *Let $G$ be a loopless graph and $H$ a refinement of $G$. Then* $\operatorname{sdgon}(G) = \operatorname{sdgon}(H)$.

*Proof.* Every refinement of $H$ is a refinement of $G$ as well, so $\operatorname{sdgon}(H) \geq \operatorname{sdgon}(G)$.

For the other direction we prove that the stable divisorial gonality of a graph does not increase by adding a leaf or subdividing an edge. Let $G'$ be a refinement of $G$ such that there exists an effective divisor $D$ with degree $\operatorname{sdgon}(G)$ and rank at least 1.

Let $\widetilde{G}$ be $G$ with a leaf $v$ added to a vertex $u$. We will show that there is a refinement $\widetilde{G}'$ of $\widetilde{G}$ such that there is an effective divisor $D'$ with degree $\operatorname{sdgon}(G)$ and rank at least 1. If there is a leaf added to $u$ in $G'$, then $G'$ is a refinement of $\widetilde{G}$, so we are done. If there is no leaf added to $u$ in $G'$, then add a leaf $l$ to vertex $u$ in $G'$. Write $\widetilde{G}'$ for this graph. It is clear that $\widetilde{G}'$ is a refinement of $\widetilde{G}$. There exists an effective divisor $D' \sim D$ on $G'$ such that $D'(u) \geq 1$. Now consider the divisor $D'$ on $\widetilde{G}'$. We claim that $D'$ has rank at least 1. By adding $l$ to every firing set that contains $u$, we see that we can still reach a divisor with one chip on $w$ for every vertex $w \in G'$. And we can reach a divisor with a chip on $l$ by firing $G'$ in $\widetilde{G}'$. Thus $D'$ is an effective divisor with rank at least 1 on $\widetilde{G}'$. We conclude that $\operatorname{sdgon}(\widetilde{G}) \leq \operatorname{sdgon}(G)$.

Let $\widetilde{G}$ be $G$ where an edge $e = u_1u_2$ is subdivided by a vertex $v$. We will prove that there is a refinement $\widetilde{G}'$ of $\widetilde{G}$ and an effective divisor on $\widetilde{G}'$ of rank at least 1 and with degree $\operatorname{sdgon}(G)$. If the edge $u_1u_2$ is subdivided in $G'$, then $G'$ is a refinement of $\widetilde{G}$ too, so we are done.

Assume that the edge $u_1u_2$ is not subdivided in $G'$. For every vertex $w \in V(G')$, let $D_w \sim D$ be a divisor with $D_w(w) \geq 1$ and $B_{w,0}, \dots B_{w,r_w}$ the level set decomposition of the transformation of $D$ into $D_2$. Suppose that there are $w$ and $i$ such that $u_1 \in B_{w,i}$ and $u_2 \notin B_{w,i}$ or such that $u_1 \notin B_{w,i}$ and $u_2 \in B_{w,i}$. Thus, suppose that there is a chip fired along the edge $u_1u_2$.

For all $w$, $i$, let $E_{w,i}$ be the set of all edges along which a chip is fired by the set $B_{w,i}$. Subdivide the following edges once to obtain a refinement $\widetilde{G}'$ of $G'$: all edges of $G'$ that occur in some set $E_{w,i}$. Notice that $\widetilde{G}'$ is a refinement of $\widetilde{G}$. Let $V_{w,i}$ be the set of vertices that are added to the edges in $E_{w,i}$. Define $B'_{w,i}$ as $B_{w,i}$ together with all added vertices that are added to an edge with both endpoints in $B_{w,i}$. We can replace every set $B_{w,i}$ by two sets $B'_{w,i}, B'_{w,i} \cup V_{w,i}$ in the level set decomposition $B_{w,0}, \dots B_{w,r_w}$ to see that for every vertex $w$ we can still reach a divisor with at least one chip on $w$. For every vertex in $V_{w,i}$ we will encounter a divisor with a chip on that vertex when we transform $D$ into $D_w$. We conclude that $\text{sdgon}(\widetilde{G}) \leq \deg(D) = \text{sdgon}(G)$.

Now suppose that for all $w$, $i$ either $u_1, u_2 \in B_{w,i}$ or $u_1, u_2 \notin B_{w,i}$. We claim that there is a divisor $D' \sim D$ such that $D'(u_1) \geq 1$ and $D'(u_2) \geq 1$. Suppose that such a divisor does not exist. Let $D_{u_1} \sim D$ be a divisor with $D_{u_1}(u_1) = 0$ and $D_{u_2} \sim D$ be a divisor with $D_{u_2}(u_2) = 0$. It follows that $D_{u_1}(u_2) = 0$ and $D_{u_2}(u_1) = 0$. Let $A_0, \dots, A_r$ be the level set decomposition of the transformation of $D_{u_1}$ into $D_{u_2}$ and let $D_1, \dots, D_{r+1}$ be the associated sequence of divisors. Let $i$ be the smallest index such that $D_i(u_2) \geq 1$. Then we know that $u_2 \notin A_{i-1}$. We also know that $D_i(u_1) = 0$ by assumption, thus $u_1$ is fired once. It follows that $u_1 \in A_{i-1}$. This yields a contradiction. We conclude that there is a divisor $D' \sim D$ such that $D'(u_1) \geq 1$ and $D'(u_2) \geq 1$.

Now subdivide the edge $u_1u_2$ by adding a vertex $v$ in $G'$ to obtain a refinement $\widetilde{G}'$ of $\widetilde{G}$. We can obtain a divisor with a chip on $v$ by firing all vertices of $\widetilde{G}$ starting from the divisor $D'$. Let $w \in \widetilde{G}$. Add $v$ to all sets $B_{w,i}$ for which holds that $u_1, u_2 \in B_{w,i}$. Now we can reach the divisor $D_w$ by firing these sets $B_{w,0}, \dots, B_{w,r_w}$. Thus $\text{sdgon}(\widetilde{G}) \leq \deg(D) = \text{sdgon}(G)$.

Since $H$ can be obtained from $G$ by adding some leaves and subdividing edges, and each of these operations does not increase the stable divisorial gonality, we can conclude that $\text{sdgon}(H) = \text{sdgon}(G)$. □

We can do the same for stable gonality [20, Lemma 5.4].

**Proposition 1.4.7.** *Let $G$ be a graph and $H$ a refinement of $G$. Then* $\text{sgon}(G) = \text{sgon}(H)$*.*

*Proof.* Every refinement of $H$ is a refinement of $G$ as well, so $\text{sgon}(H) \geq \text{sgon}(G)$.

For the other direction we prove that the stable gonality of a graph does not increase by adding a leaf or subdividing an edge. Let $G'$ be a refinement of $G$ and $T$ a tree such that there exists a finite harmonic morphism $\phi : G' \to T$ of degree $\text{sgon}(G)$.

Let $\widetilde{G}$ be $G$ with a leaf $v$ added to a vertex $u$. We will show that there is a refinement $\widetilde{G}'$ of $\widetilde{G}$, a tree $T'$ and a morphism $\phi' : \widetilde{G}' \to T'$ with degree $\text{sgon}(G)$. If there is a leaf added to $u$ in $G'$, then $G'$ is a refinement of $\widetilde{G}$, so we are done. If there is no leaf added to $u$ in $G'$, then add a leaf $l_x$ to every vertex $x$ in $\phi^{-1}(\phi(u))$. Write $\widetilde{G}'$ for the graph we obtained. It is clear that $\widetilde{G}'$ is a refinement of $\widetilde{G}$. Add a leaf $l$ to $\phi(u)$ in $T$ to obtain the tree $T'$. Set the indices of all new edges as $r_{\phi'}(xl_x) = m_\phi(x)$

and all other indices $r_{\phi'}(e) = r_\phi(e)$. Define $\phi' : \widetilde{G}' \to T'$ as

$$\phi'(y) = \begin{cases} l & \text{if } y = l_x, \\ \phi(y) & \text{otherwise.} \end{cases}$$

We see that $\phi'$ is a finite harmonic morphism with degree $\deg(\phi') = \deg(\phi) = \mathrm{sgon}(G)$, thus $\mathrm{sgon}(\widetilde{G}) \leq \mathrm{sgon}(G)$.

Subdivide the edge $e = u_1u_2$ of $G$ by a vertex $v$, write $\widetilde{G}$ for this graph. If the edge $e$ is subdivided in $G'$, then $G'$ is a refinement of $\widetilde{G}$ as well. Otherwise, we find, analogous to the previous case, by subdividing all edges of $G'$ in $\phi^{-1}(\phi(e))$, a refinement $\widetilde{G}'$ of $\widetilde{G}$. By subdividing $\phi(e)$ in $T$, we obtain a tree $T'$. Again, there is a finite harmonic morphism $\phi' : \widetilde{G}' \to T'$ of degree $\mathrm{sgon}(G)$.

Since $H$ can be obtained from $G$ by adding some leaves and subdividing edges, and each of these operations does not increase the stable gonality, we can conclude that $\mathrm{sgon}(H) = \mathrm{sgon}(G)$. □

# 2 Bounds for gonality

## 2.1 Inequalities relating different notions of gonality

From the definition of stable divisorial gonality it is clear that $\mathrm{sdgon}(G) \leq \mathrm{dgon}(G)$ for any graph $G$. It is known that $\mathrm{sgon}(G) = \min\{\mathrm{gon}(H) \mid H$ a refinement of $G\}$, see Proposition 1.4.5 (or [20, Lemma A.3]), thus it holds true that $\mathrm{sgon}(G) \leq \mathrm{gon}(G)$ for all (loopless) graphs $G$. As mentioned in [20, A.6] there is also a relation between the stable geometric gonality and the stable divisorial gonality of a graph: $\mathrm{sgon}(G) \geq \mathrm{sdgon}(G)$ for all graphs $G$.

The idea of the proof is as follows: Let $G$ be a graph, $G'$ a refinement of $G$, and $\phi : G' \to T$ a finite harmonic morphism of degree $k$ to a tree $T$. We want to show that there is a refinement $H$ of $G'$ and a divisor $D$ with degree $k$ and rank at least 1. The idea is to pick a vertex $v' \in T$, and assign $m_\phi(v)$ chips to every vertex $v$ in $\phi^{-1}(v')$. Notice that there are exactly $k$ chips in total. For any edge $e'$ in $T$ the sum of the indices of the edges that are mapped to $e'$ is equal to $k$. So if we can fire $r_\phi(e)$ chips along each edge $e$, then we can maintain the following property: there is a vertex $u' \in T$ such that the vertices $u \in \phi^{-1}(u')$ have exactly $m_\phi(u)$ chips.

Before we give the proof, we illustrate this idea by means of an example.

**Example 2.1.1.** Let $G$ be the graph in Figure 2.1. We have a finite harmonic morphism $\phi : G \to T$ of degree 4 by mapping all vertices to the vertex below it and we assign index 3 to the edges $a_1a_4$, $a_4a_5$, $a_{10}a_{11}$ and $a_{11}a_{12}$, and index 1 to all other

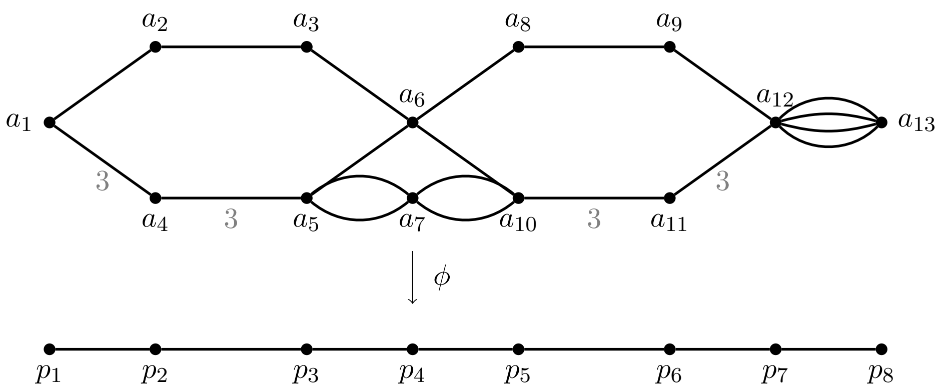



Figure 2.1: The graph $G$ of Example 2.1.1 with a finite harmonic morphism of degree 4.

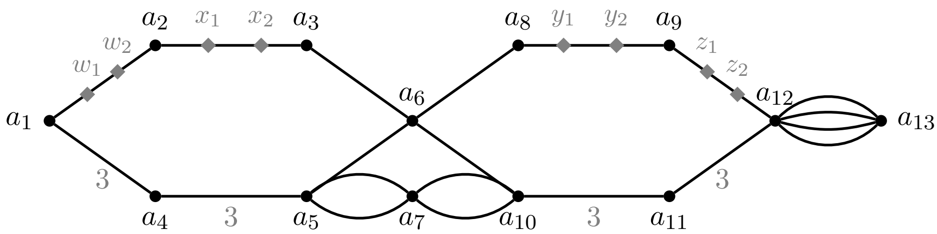



Figure 2.2: The graph $H$ of Example 2.1.1.

edges.

We pick a vertex, for example $p_4$, and distribute 4 chips over the vertices in $\phi^{-1}(p_4)$ according to their index $m_\phi$. Let $D_{p_4}$ be the divisor of this distribution, then $D_{p_4}(a_7) = D_{p_4}(a_6) = 2$.

Now we can fire vertices such that all vertices $u \in \phi^{-1}(p_5)$ have exactly $m_\phi(u)$ chips as follows: the edge $p_4p_5$ induces a cut $(X', Y')$ in $T$. This cut corresponds to a cut $(X, Y)$ in $G$, where $X = \phi^{-1}(X')$ and $Y = \phi^{-1}(Y')$. Thus $X = \{a_i \mid 1 \leq i \leq 7\}$ and $Y = \{a_i \mid 8 \leq i \leq 13\}$. By firing $X$ we get the distribution $D_{p_5}$: $D_{p_5}(a_8) = 1$ and $D_{p_5}(a_{10}) = 3$.

To obtain a distribution where all vertices $u \in \phi^{-1}(p_6)$ have exactly $m_\phi(u)$ chips, we have to fire three chips along the edge $a_{10}a_{11}$, while we fire only one chip along the edge $a_8a_9$. To do this, we have to slow this last chip down. We can do this by adding two vertices $y_1, y_2$ to the edge $a_8a_9$. Again, the edge $p_5p_6$ induces a cut $(X', Y')$ in $T$. Define $X = \phi^{-1}(X')$. Now firing the sets $X$, $X \cup \{y_1\}$ and $X \cup \{y_1, y_2\}$ leads to the desired divisor $D_{p_6}$ with $D_{p_6}(a_{11}) = 3$ and $D_{p_6}(a_9) = 1$.

Analogously, we can add two vertices $z_1$ and $z_2$ to the edge $a_9a_{12}$, and set $X = \{a_i \mid 1 \leq i \leq 11\} \cup \{y_1, y_2\}$. Now we can fire $X$, $X \cup \{z_1\}$ and $X \cup \{z_1, z_2\}$ to obtain a divisor $D_{p_7}$ with $D_{p_7}(a_{12}) = 4$. Firing the set $\{a_i \mid 1 \leq i \leq 12\} \cup \{y_1, y_2, z_1, z_2\}$ yields a distribution $D_{p_8}$ with $D_{p_8}(a_{13}) = 4$.

Analogously, we add two vertices to the edges $a_1a_2$ and $a_2a_3$. Write $H$ for the graph thus obtained. See Figure 2.2 for an illustration. This graph $H$ is a refinement of $G$, and the divisor $D_{p_4}$ on $G''$ has degree 4 and rank at least 1.

We use the idea of Example 2.1.1 for the proof of the following proposition.

**Proposition 2.1.2.** *Let $G$ be a graph. Then* $\mathrm{sgon}(G) \geq \mathrm{sdgon}(G)$.

*Proof.* Suppose that $\mathrm{sgon}(G) = k$. Then there exist a refinement $G'$ of $G$, a tree $T$ and a finite harmonic morphism $\phi : G' \to T$ such that $\phi$ has degree $k$. We will construct a refinement $H$ of $G$, such that $\mathrm{dgon}(H) \leq k$.

For every edge $e' \in E(T)$, determine $l_{e'} = \mathrm{lcm}\{r_\phi(e) \mid \phi(e) = e'\}$. Now subdivide every edge $e \in E(G')$ in $l_{\phi(e)}/r_\phi(e)$ edges by adding $a_e = l_{\phi(e)}/r_\phi(e) - 1$ vertices. Write $H$ for this refinement of $G'$; it is clear that $H$ is a refinement of $G$ too.

For a vertex $v' \in V(T)$, we define the divisor $D_{v'}$ by

$$D_{v'}(v) = \begin{cases} m_\phi(v) & \text{if } v \in \phi^{-1}(v') \\ 0 & \text{otherwise.} \end{cases}$$

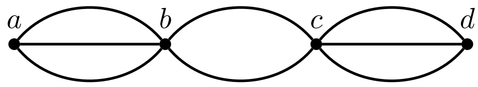


Figure 2.3: The graph $G$ of Example 2.1.3 with $\mathrm{dgon}(G) = 4$ and $\mathrm{gon}(G) \leq 3$.

Let $e' = \{x', y'\}$ be an edge in $T$. We will show that $D_{x'} \sim D_{y'}$. Removing $e$ induces a cut $(X', Y')$ in $T$, where $x' \in X'$ and $y' \in Y'$. This corresponds to a cut $(X, Y)$ in $G'$, where $X = \phi^{-1}(X')$ and $Y = \phi^{-1}(Y')$. Define $\widetilde{X} \subseteq V(H)$ as $X \cup \{w_{e,i} \mid e \in E(X)\}$. Now we will construct $l(e')$ sets $A_0, A_1, \ldots, A_{l(e')-1}$. For every $0 \leq i \leq l(e') - 1$, set $A_i = \widetilde{X}$.

Notice that the edges that cross the cut $(X, Y)$ are exactly the edges for which holds that $\phi(e) = e'$. For every edge $e = xy \in E(G')$, such that $\phi(e) = e'$, $x \in X$, $y \in Y$, let $w_{e,1}, \ldots, w_{e,a_e}$ be the vertices that are added to $e$ in order from $x$ to $y$. Add vertex $w_{e,i}$ to all sets $A_j$ with $j \geq i \cdot r_\phi(e)$. Notice that we have added the set $\{w_{e,1}, w_{e,2}, \ldots, w_{e,i}\}$ to exactly $r_\phi(e)$ sets.

Firing the sets $A_0, \ldots, A_{l(e')-1}$ moves exactly $r_\phi(e)$ chips along edge $e$ from $x$ to $y$. It follows that every $x$ with $\phi(x) = x'$ loses exactly $m_\phi(x)$ chips, and every $y$ with $\phi(y) = y'$ receives exactly $m_\phi(y)$ chips. We conclude that firing the sets $A_0, \ldots, A_{l(e')-1}$, leads to the divisor $D_{y'}$. Thus $D_{x'} \sim D_{y'}$.

Pick a vertex $v' \in V(T)$ and consider the divisor $D_{v'}$; this divisor has degree $k$. We claim that this divisor has rank at least 1. Let $u \in V(G')$ and $u' = \phi(u)$. There is a unique path $v', v_1, \ldots, v_r, u'$ in $T$ from $v'$ to $u'$. We have seen that $D_{v'} \sim D_{v_1} \sim \ldots \sim D_{v_r} \sim D_{u'}$. We know that $D_{u'}(u) \geq 1$. Thus for all vertices $u \in V(G')$, there is a divisor $D' \sim D_{v'}$ with $D'(u) \geq 1$. Now let $w_{e',i}$ be a vertex that is added to $G'$. There is a path in $T$ starting in $v'$ that contains $e'$. Now we can move chips in the same way as described above. Notice that there is some intermediate situation where there is at least 1 chip on $w_{e',i}$ and at least 0 on all other vertices.

We conclude that $\mathrm{dgon}(H) \leq k$, so $\mathrm{sdgon}(G) \leq k$.

□

So far we have the following relations between the different notions of gonality:

$$\begin{cases} \mathrm{dgon}(G) \geq \mathrm{sdgon}(G) \\ \mathrm{gon}(G) \geq \mathrm{sgon}(G) \geq \mathrm{sdgon}(G) \end{cases}$$

The following examples show that divisorial gonality is incomparable with stable gonality and with geometric gonality.

**Example 2.1.3** ([17, Example 4])**.** Let $G$ be the graph in Figure 2.3. In Example 1.2.5, we have seen that $\mathrm{dgon}(G) = 4$. This graph has stable gonality at most 3, as the following morphism shows. Give one of the edges $bc$ index 2 and all other edges index 1. Let $T$ be a path on four vertices $p_1, p_2, p_3, p_4$. The map $\phi : G \to T$, defined by $\phi(a) = p_1$, $\phi(b) = p_2$, $\phi(c) = p_3$, $\phi(d) = p_4$ is a non-degenerate harmonic morphism of degree 3. Thus we see that $\mathrm{gon}(G) \leq 3$.

**Example 2.1.4** ([2, Example 5.13])**.** Let $G$ be as in Figure 2.4(a). We claim that $\mathrm{sgon}(G) = 4$ and $\mathrm{dgon}(G) \leq 3$. First we look at the divisorial gonality. Let $D$ be the divisor with $D(a) = D(b) = D(c) = 1$ and $D(d) = D(e) = D(f) = 0$. It is easy to check that this divisor has rank at least 1, thus $\mathrm{dgon}(G) \leq 3$.

We will now show that $\mathrm{sgon}(G) = 4$. First look at the refinement and map in Figure 2.4(b). One can easily check that this is a finite harmonic morphism of degree 4. We see that $\mathrm{sgon}(G) \leq 4$.

Now we will show that $\mathrm{sgon}(G) \geq 4$. Suppose that there exists a refinement $G'$ of $G$ and a finite harmonic morphism $\phi : G' \to T$ of degree 3, where $T$ is a tree. Suppose that $\phi(a) \neq \phi(d)$, $\phi(b) \neq \phi(e)$ and $\phi(c) \neq \phi(f)$, then

$$\sum_{x \in G'_a(d), \phi(x) = \phi(a)} m_\phi(x) = 3,$$
$$\sum_{x \in G'_b(e), \phi(x) = \phi(b)} m_\phi(x) = 3,$$
$$\sum_{x \in G'_c(f), \phi(x) = \phi(c)} m_\phi(x) = 3.$$

It follows that $\phi(a) \neq \phi(b)$, $\phi(b) \neq \phi(c)$ and $\phi(c) \neq \phi(a)$. Write $P_{ab}$ for the path from $a$ to $b$ in $G'$ and $P_{ac}$ for the path from $a$ to $c$ and $P_{bc}$ for the path from $b$ to $c$. We distinguish two cases:

Suppose that $\phi(a) \in \phi(P_{bc})$. Then there is a vertex in $P_{bc}$ that is mapped to $\phi(a)$. But then it follows that

$$\sum_{x, \phi(x) = \phi(a)} m_\phi(x) \geq 4.$$

This yields a contradiction.

The cases $\phi(b) \in \phi(P_{ac})$ and $\phi(c) \in \phi(P_{ab})$ are analogous.

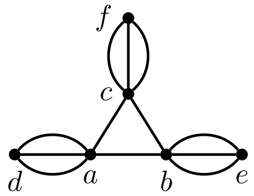
Suppose that $\phi(a) \notin \phi(P_{bc})$, $\phi(b) \notin \phi(P_{ac})$ and $\phi(c) \notin \phi(P_{ab})$. Then there is a vertex $v' \in T$ such that $v' \in \phi(P_{ab})$, $v' \in \phi(P_{ac})$ and $v' \in \phi(P_{bc})$. Let $v$ be the vertex on $P_{bc}$ such that $\phi(v) = v'$. It follows that there is an edge $e_1 = vw_1$ incident to $v$ that is mapped to the first edge of the path from $v'$ to $\phi(a)$. There is an edge $e_2 = w_1w_2$ that is mapped to the second edge of the path from $v'$ to $\phi(a)$. We can iterate this to see that there is a vertex $w$ added to $v$ such that $\phi(w) = \phi(a)$. It follows that

$$\sum_{x, \phi(x) = \phi(a)} m_\phi(x) \geq 4.$$

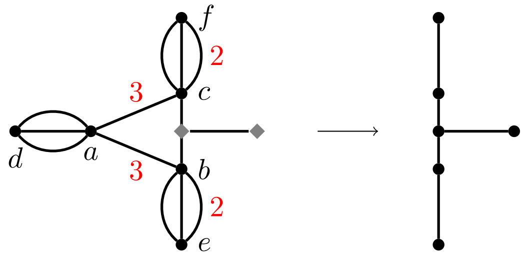
This yields a contradiction.

Now suppose that $\phi(a) = \phi(d)$. Suppose that $\phi(b) \neq \phi(e)$. There is an edge $e_1 = dw_1$ incident to $d$ that is mapped to the first edge of the path from $\phi(a)$ to $\phi(b)$. There is an edge $e_2 = w_1w_2$ that is mapped to the second edge of the path from $\phi(a)$ to $\phi(b)$. We can iterate this to see that there is a vertex $w$ added to $d$ such that $\phi(w) = \phi(b)$. It follows that

$$\sum_{x, \phi(x) = \phi(b)} m_\phi(x) \geq 4.$$




(a) The graph $G$ of example 2.1.4.




(b) A refinement of $G$ and a finite harmonic morphism of degree 4.

Figure 2.4: The graph $G$ of Example 2.1.4 with $\mathrm{sgon}(G) = 4$ and $\mathrm{dgon}(G) \leq 3$.

This yields a contradiction.

Now suppose that $\phi(a) = \phi(d)$, $\phi(b) = \phi(e)$ and $\phi(c) = \phi(f)$. It follows that

$$\sum_{x \in G'_a(d), \phi(x)=\phi(a)} m_\phi(x) \geq 2,$$
$$\sum_{x \in G'_b(e), \phi(x)=\phi(b)} m_\phi(x) \geq 2,$$
$$\sum_{x \in G'_c(f), \phi(x)=\phi(c)} m_\phi(x) \geq 2.$$

We see that $\phi(a) \neq \phi(b)$, $\phi(b) \neq \phi(c)$ and $\phi(c) \neq \phi(a)$. We can distinguish the same cases as before, and we see that there are vertices $w_1$ and $w_2$ added to $e$ and $f$ that are mapped to $\phi(a)$. But then it follows that

$$\sum_{x, \phi(x)=\phi(a)} m_\phi(x) \geq 4.$$

This yields a contradiction.

We conclude that such a morphism $\phi$ does not exist, and $\mathrm{sgon}(G) \geq 4$.

## 2.2 Treewidth is a lower bound for gonality

Now we look at the relation between gonality and treewidth. First we give one of the equivalent definitions of treewidth. After that, we will prove that $\mathrm{dgon}(G) \geq \mathrm{tw}(G)$ for all $G$. We follow the proof of [24, Section 2]. We will conclude that treewidth is a lower bound for all notions of gonality. We start with introducing brambles.

**Definition 2.2.1.** Let $G$ be a connected graph. A *bramble* $\mathcal{B}$ is a subset of $\mathcal{P}(V(G))$ such that $\emptyset \notin \mathcal{B}$, $\bigcup_{B \in \mathcal{B}} B = V(G)$ and for any two $B, B' \in \mathcal{B}$ it holds that the induced graph on $B \cup B'$ is connected.

**Definition 2.2.2.** Let $\mathcal{B}$ be a bramble. We call a set $S \subseteq V(G)$ a *hitting set* if for every $B \in \mathcal{B}$ it holds that $B \cap S \neq \emptyset$.

**Definition 2.2.3.** Let $\mathcal{B}$ be a bramble. The *order* $\|\mathcal{B}\|$ of $\mathcal{B}$ is the minimum size of a hitting set: $\|\mathcal{B}\| = \min\{|S| \mid S \text{ is a hitting set for } \mathcal{B}\}$.

The following definition is one of the definitions of treewidth:

**Definition 2.2.4.** Let $G$ be a connected graph. Then the treewidth of $G$, in notation $\mathrm{tw}(G)$, is $\mathrm{tw}(G) = \max\{\|\mathcal{B}\| \mid \mathcal{B} \text{ a bramble}\} - 1$.

A better known definition of treewidth uses the notion of tree-decomposition.

**Definition 2.2.5.** Let $G$ be a graph. A *tree decomposition* is a pair $(T, W)$, where $T$ is a tree and $W = (W_t)_{t \in V(T)}$ is a family of subsets of $V(G)$ such that:

(i) $\bigcup_{t \in V(T)} W_t = V(G)$;

(ii) for every edge $e = uv \in E(G)$ there is a $t \in V(T)$ such that $u, v \in W_t$;

(iii) for every vertex $v \in V(G)$, the induced subgraph on $\{t \in V(T) \mid v \in W_t\}$ is connected.

**Definition 2.2.6.** Let $G$ be a graph and $(T, W)$ a tree decomposition. The *width* of a $(T, W)$ is $\max\{|W_t| - 1 \mid t \in T(V)\}$.

**Lemma 2.2.7.** *Let $G$ be a graph. The treewidth of $G$ is at most $k$ if and only if there is a tree decomposition of width $k$.*

*Proof.* See [30, 1.4]. □

Now before we look at the relation between gonality and treewidth, we will prove two lemmas.

**Lemma 2.2.8.** *Let $D$ be an effective divisor and let $U \subseteq V(G)$ be a valid firing set. Let $D'$ be the divisor after firing the set $U$. Let $B \subseteq V(G)$ be such that the induced graph on $B$ is connected. If there is some vertex in $B$ with chips in $D$, but there are no chips on $B$ in $D'$, then it holds that $B \subseteq U$, so all vertices in $B$ are fired.*

*Proof.* Let $v \in B$ be a vertex such that $D(v) > 0$. It is clear that $v$ can only lose chips if $v$ is fired. Thus $v \in U$. Now suppose that $B \nsubseteq U$. Then there exist vertices $x, y$ in $B$ such that $x$ and $y$ are neighbours and $x \in U$, $y \notin U$, since the induced graph on $B$ is connected. Then we see that $y$ receives a chip by firing $U$, thus $D'(y) > 0$. This yields a contradiction. We conclude that $B \subseteq U$. □

**Lemma 2.2.9.** *Let $\mathcal{B}$ be a bramble and $U \subseteq V(G)$. Suppose that there exist $B, B' \in \mathcal{B}$ such that $B \subseteq U$ and $B' \subseteq V \backslash U$. Then $\|\mathcal{B}\| \leq |E(U, V\backslash U)| + 1$.*

*Proof.* We will construct a set $S$ of size $E(U, V\backslash U) + 1$ and prove that it is a hitting set for $\mathcal{B}$. Define $X = \{x \in U \mid \mathrm{outdeg}_U(x) \geq 1\}$ and $Y = \{y \in V\backslash U \mid \mathrm{outdeg}_{V\backslash U}(y) \geq 1\}$. Let $B'' \in \mathcal{B}$, $B'' \subseteq U$ be a set such that $B'' \cap X$ is minimal with respect to inclusion. Notice that $B'' \cap X \neq \emptyset$, since $B \cup B'$ is connected and $B' \subseteq V\backslash U$. Pick $v \in B'' \cap X$, add $v$ to $S$. Now for every edge $e = xy \in E(U, V\backslash U)$, where $x \in X$ and $y \in Y$, add one of its endpoints to $S$: if $x \in B$, add $y$ to $S$, otherwise add $x$ to $S$. Now $S$ consists of $v$ together with an endpoint of every edge in $E(U, V\backslash U)$, thus $|S| = |E(U, V\backslash U)| + 1$.

Let $A \in \mathcal{B}$. To prove that $S$ hits $A$, we distinguish three cases.

- Suppose that $A \subseteq U$. Since $B'' \cap X$ is minimal, it holds that $A \cap X = B'' \cap X$ or there exists a vertex $x \in (A \cap X) \backslash (B'' \cap X)$. In the first case $v \in A \cap S$. In the second case we have $x \in A \cap S$.
- Suppose that $A \subseteq V \backslash U$. We see that there are neighbours $x$ and $y$ with $x \in B''$ and $y \in A$, since the induced graph on $A \cup B''$ is connected. It follows that $y \in A \cap S$.
- Suppose that $A \cap U \neq \emptyset$ and $A \cap (V \backslash U) \neq \emptyset$. Since the induced graph on $A$ is connected, it is clear that $A$ contains an edge $e = xy$ with $x \in X$ and $y \in Y$. Thus $S \cap A$ contains at least one of $x$ and $y$.

We conclude that $S$ is a hitting set of size $|E(U, V\backslash U)|+1$, thus $\|\mathcal{B}\| \leq |E(U, V\backslash U)|+1$. □

Now we are ready to prove that treewidth is a lower bound for divisorial gonality. For a divisor $D$ the *support* $\text{supp}(D)$ of $D$ is the set of vertices with chips: $\text{supp}(D) = \{v \mid D(v) > 0\}$.

**Lemma 2.2.10.** *Let $G$ be a connected graph. Then* $\text{dgon}(G) \geq \text{tw}(G)$.

*Proof.* Suppose that $\text{dgon}(G) = k$. Let $\mathcal{B}$ be a bramble of maximum order, say $l$. Then $\text{tw}(G) = l - 1$. Let $D$ be a divisor of rank at least 1 and degree $k$, such that $\text{supp}(D) \cap B$ is non-empty for a maximal number of $B \in \mathcal{B}$. If $\text{supp}(D)$ is a hitting set for $\mathcal{B}$, then it holds that $\|\mathcal{B}\| \leq k$. Thus $l \leq k$ and $\text{tw}(G) = l - 1 < k = \text{dgon}(G)$.

Now suppose that $\text{supp}(D)$ is not a hitting set. Let $B \in \mathcal{B}$ be a set such that $B \cap \text{supp}(D) = \emptyset$ and pick $v \in B$. Let $D_v \sim D$ be a divisor with a chip on $v$. Let $A_0 \subseteq A_1 \subseteq \ldots \subseteq A_r$ be the level set decomposition of the transformation of $D$ into $D_v$, and let $D_1, \ldots, D_{r+1}$ be the associated sequence of divisors.

Let $j$ be the least index such that $\text{supp}(D_j) \cap B \neq \emptyset$. Since we chose $D$ to be such that it intersects as much elements of $\mathcal{B}$ as possible, there is a $B'$ that we do not hit any more, i.e. there exists a $B' \in \mathcal{B}$ such that $B' \cap \text{supp}(D) \neq \emptyset$ and $B' \cap \text{supp}(D_i) = \emptyset$ for some $i \leq j$. Let $i$ be the least index such that such a $B'$ exists.

By Lemma 2.2.8 we see that $B' \subseteq A_{i-1}$. Notice that we can obtain $D_{j-1}$ from $D_j$ by firing $V \backslash A_{j-1}$. Thus by Lemma 2.2.8 it follows that $B \subseteq (V \backslash A_{j-1})$. Since $A_{i-1} \subseteq A_{j-1}$, we see that $B \subseteq (V \backslash A_{i-1})$. By Lemma 2.2.9 we see that $\|\mathcal{B}\| \leq |E(A_{i-1}, V \backslash A_{i-1})| + 1$. We conclude that:

$$\begin{aligned} \text{tw}(G) &= \|\mathcal{B}\| - 1 \\ &\leq |E(A_{i-1}, V\backslash A_{i-1})| \\ &\leq k = \text{dgon}(G), \end{aligned}$$

where the second inequality holds since $A_{i-1}$ is valid for $D_{i-1}$. □

It is known that treewidth does not change under taking refinements, that is $\text{tw}(G) = \text{tw}(H)$ for any refinement $H$ of $G$. Thus we see that

$$\begin{aligned} \text{sdgon}(G) &= \min\{\text{dgon}(H) \mid H \text{ a refinement of } G\} \\ &\geq \min\{\text{tw}(H) \mid H \text{ a refinement of } G\} \\ &= \text{tw}(G). \end{aligned}$$

Summing up the results of this chapter, we conclude:

**Theorem 2.2.11.** *Let $G$ be a graph. The following relations hold:*

$$\begin{cases} \mathrm{dgon}(G) \geq \mathrm{sdgon}(G) \geq \mathrm{tw}(G), \\ \mathrm{gon}(G) \geq \mathrm{sgon}(G) \geq \mathrm{sdgon}(G). \end{cases}$$

□

# 3 Recognizing hyperelliptic graphs

In [8], hyperelliptic graphs $G$ are defined as graphs with $\mathrm{dgon}(G) = 2$. We call graphs of stable or stable divisorial gonality 2 respectively *stable* or *stable divisorial hyperelliptic* graphs.

In this chapter we will give two algorithms to recognize stable hyperelliptic graphs and stable divisorial hyperelliptic graphs. These algorithms are inspired by the reductions rules to recognize simple graphs with treewidth 2 or 3 [4].

## 3.1 Reduction rules

The main algorithmic technique that we will use is based on reduction rules. By a reduction rule we mean a rule that can be applied to a graph to produce a different, preferably smaller, graph. A set of reduction rules $\mathcal{R}$ can be used to recognize a graph class $\mathcal{G}$ if there is a finite set of graphs $\mathcal{H}$, such that a graph $G$ is an element of $\mathcal{G}$ if and only if it can be reduced using the rules from $\mathcal{R}$ to one of the graphs in $\mathcal{H}$. There are, for example, such rules to recognize graphs with treewidth 2 or 3 [4] or to recognize series-parallel graphs [25]. All classes of graphs with bounded treewidth that can be defined by a formula in monadic second order logic, can be recognized using reduction rules in linear time [3]. In parametrized complexity theory, reduction rules are used to find polynomial kernels, see for example [1] for such rules for the dominating set problem.

In this chapter we use reduction rules to recognize stable (divisorial) hyperelliptic graphs. We use the following notation to denote that a graph can be produced by the application of reduction rules starting from another graph:

**Definition 3.1.1.** 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 from $G$ 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 $\boldsymbol{R}$ is a single rule, we write $G\boldsymbol{R}H$ as shorthand for $G\{\boldsymbol{R}\}H$.

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 of vertices:

**Definition 3.1.2.** Given a graph $G = (V, E)$ a *set of constraints* $\mathcal{C}$ is a set of unordered pairs $(v, w)$, where $v, w \in V$. This set can contain pairs of the form $(v, v)$.

Notice that $\mathcal{C}$ is a set, thus it can contain every pair $(u, v)$ at most one.

Though the different notions of gonality use the same concept of a set of constraints, the restrictions given by a constraint differ between them. What a constraint means for each type of gonality will be explained in their respective sections.

Constraints are, like edges, pairs of vertices, so we can consider them as an extra set of edges. Similar to the notation $E_v$, we will use $\mathcal{C}_v$ to denote all constrains that contain a vertex $v$.

Our final goal with each set of reduction rules is to show that they 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 3.1.3.** Let $\boldsymbol{R}$ be a rule and $\boldsymbol{S}$ be a set of reduction rules. Let $\mathcal{A}$ be a class of graphs. We call $\boldsymbol{R}$ *safe* for $\mathcal{A}$ if for any two graphs $G, H$, if $G\boldsymbol{R}H$ then $H \in \mathcal{A} \iff G \in \mathcal{A}$. 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.

**Definition 3.1.4.** Let $\boldsymbol{S}$ be a set of reduction rules and let $\mathcal{A}$ be a class of graphs. We call $\boldsymbol{S}$ *complete* for $\mathcal{A}$ if for any graph $G \in \mathcal{A}$ it holds $G\boldsymbol{S}^*\emptyset$.

Any rule set that is both complete and safe for $\mathcal{A}$, 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 3.1.5.** *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 $G \in \mathcal{A}$, note that 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 $H\boldsymbol{S}^*\emptyset$. □

## 3.2 Reduction rules for stable gonality

In this section, we give a complete set of safe reduction rules to recognize stable hyperelliptic graphs, i.e. graphs with stable gonality 2. We will first introduce some extra notation and then we will state all rules. Next we will show that all rules are safe for graphs with stable gonality at most 2 and that those graphs can be reduced to the empty graph, i.e., that we have a safe and complete set of rules. It is not hard to see that the set of rules implies a polynomial time algorithm to test if a graph has stable gonality at most 2; in [14, Section 7], it is discussed how we can obtain an algorithm with a running time of $O(m + n \log n)$ by paying attention to the implementation of the reduction rules.

## Notation

For a given graph $G$, we want to determine whether there exists a finite harmonic morphism of degree 2 from a refinement of $G$ to a tree. We will do this by reducing $G$ to the empty graph. During this process we sometimes add constraints to our graph. The set of constraints gives restrictions to which morphisms we allow.

**Definition 3.2.1.** Let $G$ be a graph, $G'$ a refinement of $G$ and $T$ a tree. Let $\phi : G' \to T$ be a map. We call $\phi$ a *suitable morphism* if it is a finite harmonic morphism of degree 2 and it satisfies the following conditions.

(i) For all pairs $(v, v) \in \mathcal{C}$ it holds that $m_\phi(v) = 2$.

(ii) For all pairs $(u, v) \in \mathcal{C}$ with $u \neq v$ it holds that $\phi(u) = \phi(v)$ and $m_\phi(u) = m_\phi(v) = 1$.

We say that a graph with constraints has stable gonality at most 2 if there exists a suitable morphism from a refinement of $G$ to a tree. Let $\mathcal{G}_2^s$ be the class of graphs with constraints that have stable gonality at most 2. We define the empty graph to have stable gonality 0 and thus $\emptyset \in \mathcal{G}_2^s$.

Now we can prove some lemmas about constraints.

**Lemma 3.2.2.** *Let $G$ be a graph with constraints. If there is a vertex $v$ with $|\mathcal{C}_v| > 1$, then* $\operatorname{sgon}(G) \geq 3$.

*Proof.* Let $G$ be a graph with $\operatorname{sgon}(G) = 2$. Suppose that $|\mathcal{C}_v| > 1$. Let $(u, v)$ and $(v, w)$ be two constraints that contain $v$. We know that $u \neq w$. Suppose that $\phi$ is a suitable morphism of degree 2 from a refinement of $G$ to a tree. We distinguish two cases.

Suppose that $u = v$. Then we know that $m_\phi(v) = 2$. On the other hand we have that $m_\phi(v) = m_\phi(w) = 1$. This yields a contradiction.

Now suppose that $u \neq v$ and $w \neq v$. Notice that $\phi(u) = \phi(v) = \phi(w)$, thus there are at least three vertices mapped to $\phi(v)$. We conclude that $\deg(\phi) \geq 3$. This yields a contradiction.

We conclude that $|\mathcal{C}_v| \leq 1$. □

**Lemma 3.2.3.** *Let $G$ be a graph, and $\phi : G \to T$ a finite harmonic morphism of degree 2. If $\phi(u) = \phi(v)$, then* $\deg(u) = \deg(v)$.

*Proof.* Notice that $m_\phi(u) = m_\phi(v) = 1$. Let $e$ be an edge incident to $\phi(u)$. We see that there is exactly one edge $e'$ such that $e'$ is incident to $u$ and $\phi(e') = e$. On the other hand every edge that is incident to $u$ is mapped to an edge that is incident to $\phi(u)$. So we conclude that $\deg_G(u) = \deg_T(\phi(u))$. Analogously we find that $\deg_G(v) = \deg_T(\phi(v))$. Since $\phi(u) = \phi(v)$, it follows that $\deg(u) = \deg(v)$. □

**Lemma 3.2.4.** *Let $G$ be a graph where every leaf is incident to a constraint, so if* $\deg(u) = 1$ *then* $\mathcal{C}_u \neq \emptyset$ *for all* $u$*. Suppose that* $(u, v) \in \mathcal{C}$*. If* $\deg(u) \neq \deg(v)$*, then* $\operatorname{sgon}(G) \geq 3$.

*Proof.* Suppose that $\deg(u) \neq \deg(v)$. Assume without loss of generality that $\deg(u) > \deg(v)$. Suppose that $\operatorname{sgon}(G) = 2$. Let $G'$ be a refinement of $G$ with a minimal number of vertices such that there exists a suitable morphism of degree 2. Let $\phi : G' \to T$ be such a morphism.

We know that $\phi(u) = \phi(v)$, thus $\deg_{G'}(u) = \deg_{G'}(v)$. So there is a neighbour $x$ of $v$ which is an external added vertex. Now we look at $\phi(x)$. Notice that there is a neighbour $y$ of $u$ such that $\phi(x) = \phi(y)$. It is clear that $y \neq x$, since $x$ is an external added vertex. Thus $m_\phi(x) = m_\phi(y) = 1$.

Let $x'$ be a neighbour of $x$, not equal to $v$. Suppose that $m_\phi(x') = 2$. We know that the edge $e = \{x, x'\}$ has index 1, so there exists another neighbour of $x'$ that is mapped to $\phi(x)$. We know that $y$ is the unique vertex other than $x$ that is mapped to $\phi(x)$, hence $y$ is a neighbour of $x'$. This yields a contradiction, since $x'$ is an external added vertex. We conclude that $m_\phi(x') = 1$. Inductively we find that $m_\phi(x'') = 1$ for all vertices $x'' \in G_v(x)$.

Let $x' \neq v$ be a leaf in $G_v(x)$; then $m_\phi(x') = 1$. Let $y'$ be the vertex such that $\phi(x') = \phi(y')$. Now it follows that $\deg(x') = \deg(y')$, thus $y'$ is a leaf. Since $x'$ is an added vertex, it also follows that $\mathcal{C}_y = 0$. Since every leaf in $G$ is incident to a constraint, we conclude that $y'$ is added to $G$. It follows that $G' \backslash \{y', x'\}$ is a refinement of $G$ and that $\phi' : G' \backslash \{y', x'\} \to T \backslash \{\phi(y')\}$ is a suitable morphism of degree 2. This yields a contradiction with the minimality of $G'$.

We conclude that $\operatorname{sgon}(G) \geq 3$. □

## Reduction rules

We will now state all rules. Figure 3.1 shows all rules in pictures; constraints are showed as green dashed edges. From now on we will refer to the constraints as green edges. When a rule adds a constraint $uv$, and there already exists such a constraint, then the set of constraints does not change.

**Rule $T_1^s$.** *Let $v$ be a leaf with $\mathcal{C}_v = \emptyset$. Let $u$ be the neighbour of $v$. Contract the edge $uv$.*

**Rule $T_2^s$.** *Let $v$ be a leaf with $\mathcal{C}_v = \{(v, v)\}$. Let $u$ be the neighbour of $v$. Contract the edge $uv$.*

**Rule $S_1^s$.** *Let $v$ be a vertex of degree 2 with $\mathcal{C}_v = \emptyset$. Let $u_1$, $u_2$ be the neighbours of $v$ (possibly $u_1 = u_2$). Contract the edge $u_1v$.*

**Rule $T_3^s$.** *Let $G$ be a graph where every leaf and every degree 2 vertex is incident to a green edge. Let $v_1$ and $v_2$ be two leaves that are connected by a green edge. Let $u_1$ and $u_2$ be their neighbours (possibly $u_1 = u_2$). Contract the edges $u_1v_1$ and $u_2v_2$.*

**Rule $S_2^s$.** *Let $G$ be a graph where every leaf and every degree 2 vertex is incident to a green edge. Let $v$ be a vertex of degree 2 with a green loop, such that there exists*

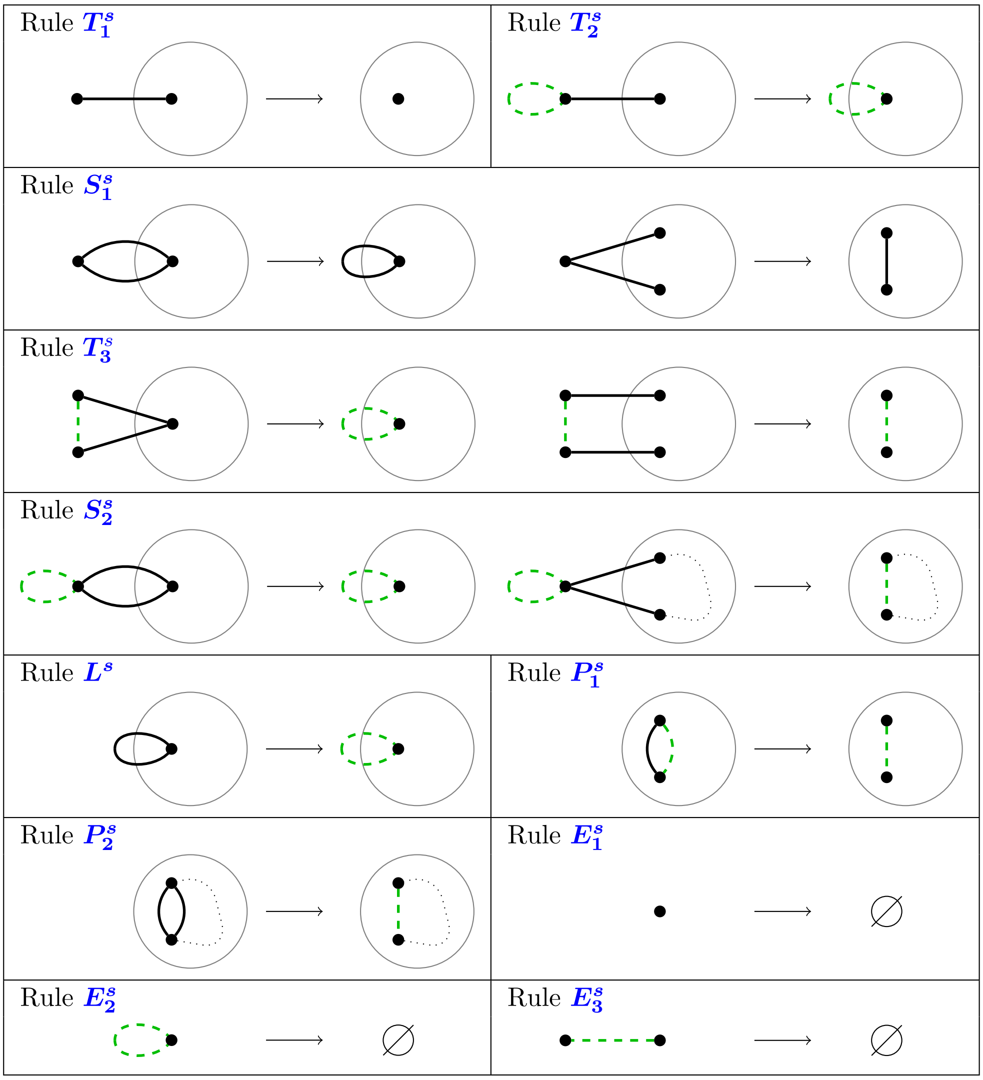



Figure 3.1: The reduction rules for recognizing stable hyperelliptic graphs.

*a path from $v$ to $v$ in $G$ (possibly containing green edges). Let $u_1$ and $u_2$ be the neighbours of $v$ (possibly $u_1 = u_2$). Remove $v$ and connect $u_1$ and $u_2$ with a green edge.*

**Rule $L^s$.** *Let $v$ be a vertex with a loop. Remove this loop from $v$ and add a green loop to $v$.*

**Rule $P_1^s$.** *Let $uv$ be an edge such that there also exists a green edge $uv$. Remove the black edge $uv$.*

**Rule $P_2^s$.** *Let $u, v$ be vertices, such that $|E(u, v)| > 1$. Let $e_1$ and $e_2$ be two of those edges. If there exists another path, possibly containing green edges, from $u$ to $v$, then remove $e_1$ and $e_2$ and add a green edge from $u$ to $v$.*

**Rule $E_1^s$.** *Let $G$ be the graph consisting of a single vertex $v$ with $\mathcal{C}_v = \emptyset$. Remove $v$.*

**Rule $E_2^s$.** *Let $G$ be the graph consisting of a single vertex $v$ with a green loop. Remove $v$.*

**Rule $E_3^s$.** *Let $G$ be the graph consisting of two vertices $u$ and $v$ that are connected by a green edge. Remove $u$ and $v$.*

We will write $\mathcal{R}^s$ for this set of reduction rules. We can now state the main theorem; in the next subsections we will prove this theorem.

**Theorem 3.2.5.** *The set of rules $\mathcal{R}^s$ is safe and complete for $\mathcal{G}_2^s$.*

## Safeness

First, we prove that the rules $\mathcal{R}^s$ are safe for $\mathcal{G}_2^s$, i.e., if $G$ a is graph, and $H$ is obtained from $G$ by applying one of the rules in $\mathcal{R}^s$, then $\text{sgon}(G) \leq 2$ if and only if $\text{sgon}(H) \leq 2$. In all proofs we assume that the original graph is called $G$ and the graph obtained by applying a rule is called $H$.

**Lemma 3.2.6.** *Rule $T_1^s$ is safe.*

*Proof.* This follows from Proposition 1.4.7. □

**Lemma 3.2.7.** *Rule $T_2^s$ is safe.*

*Proof.* Let $v$ be the vertex in $G$ to which the rule is applied.

Suppose that $\text{sgon}(G) \leq 2$. Then there exists a refinement $G'$ of $G$ and a suitable morphism $\phi : G' \to T$. Let $u$ be the neighbour of $v$ in $G$. We distinguish two cases:

Suppose that $m_\phi(u) = 2$. Define $H'$ as the graph $G'$ with a green loop at vertex $u$ and without the green loop at $v$, then $H'$ is a refinement of $H$. Now we see that $\phi : H' \to T$ is a suitable morphism, so $\text{sgon}(H) \leq 2$.

Suppose that $m_\phi(u) = 1$. Let $v_0 = v, v_1, \ldots, v_k = u$ be the vertices that are added to the edge $uv$ of $G$. Let $i$ be the largest integer such that $m_\phi(v_i) = 2$. Notice that $i < k$. Then there exists another vertex $x_1$ in $G'$ such that $\phi(v_{i+1}) = \phi(x_1)$. If $v_{i+1} \neq u$, it follows that there is an edge $x_1x_2$ that is mapped to $\phi(v_{i+1}v_{i+2})$. And since $m_\phi(v_{i+2}) = 1$, we see that $x_2 \neq v_{i+2}$. It follows that there exists $x_1 \neq v_{i+1}$, $x_2 \neq v_{i+2}$, …, $x_{k-i} \neq v_k$ such that $\phi(v_{i+j}) = \phi(x_j)$. Write $x = x_{k-i}$, then $\phi(x) = \phi(u)$ and $m_\phi(u) = m_\phi(x) = 1$. See Figure 3.2 for an illustration of this.

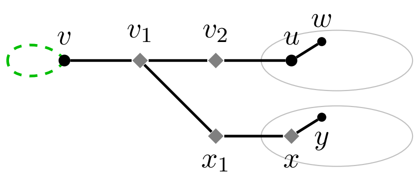



Figure 3.2: Proof of Lemma 3.2.7.

Notice that $x$ is an external added vertex. Let $w$ be a neighbour of $u$ not equal to $v_{k-1}$. Then we see that there exists a vertex $y$ such that $\phi(uw) = \phi(xy)$. Since $x$ is an external added vertex, we see that $w \neq y$. We conclude that $m_\phi(w) = 1$. Inductively we see that for every vertex $w'$ in $G_{v_i}(v_{i+1})\backslash\{v_i\}$ it holds that $m_\phi(w') = 1$. Define $H'$ as $G_{v_i}(v_{i+1})\backslash\{v_i\}$, with a green loop at vertex $u$. Notice that $H'$ is a refinement of $H$. Now we can restrict $\phi$ to $H'$ and assign to every edge index $r_{\phi'}(e) = 2$ to obtain a suitable morphism: $\phi' : H' \to T'$, where $T' = \phi(G_{v_i}(v_{i+1})\backslash\{v_i\})$. We conclude that $\mathrm{sgon}(H) \leq 2$.

Suppose that $\mathrm{sgon}(H) \leq 2$. Then there exists a refinement $H'$ of $H$ and a suitable morphism $\phi : H' \to T$. Write $u$ for the neighbour of $v$ in $G$. We know that $m_\phi(u) = 2$. Then add a leaf with a green loop to $u$ in $H'$ to obtain $G'$. Now we see that $G'$ is a refinement of $G$. Give the edge $uv$ index $r_{\phi'}(uv) = 2$, and give all other edges $e$ index $r_{\phi'}(e) = r_\phi(e)$. Add a leaf $v'$ to $\phi(u)$ in $T$ to obtain $T'$. Then we can extend $\phi$ to $\phi' : G' \to T'$ as follows:

$$\phi'(x) = \begin{cases} \phi(x) & \text{if } x \in H' \\ v' & \text{if } x = v. \end{cases}$$

It is easy to check that $\phi'$ is a suitable morphism, so we conclude that $\mathrm{sgon}(G) \leq 2$. □

**Lemma 3.2.8.** *Rule* $\boldsymbol{S_1^s}$ *is safe.*

*Proof.* This follows from Proposition 1.4.7. □

**Lemma 3.2.9.** *Rule* $\boldsymbol{T_3^s}$ *is safe.*

*Proof.* Let $v_1$ and $v_2$ be the vertices in $G$ to which the rule is applied.

"$\Longrightarrow$": Suppose that $\mathrm{sgon}(G) \leq 2$. Let $G'$ be a minimum refinement of $G$ such that there exists a suitable morphism $\phi : G' \to T$, i.e. for every refinement $G''$ with less vertices than $G'$ there is no suitable morphism $\phi' : G'' \to T'$ to a tree $T'$. Let $u_1$ and $u_2$ be the neighbours of $v_1$ and $v_2$ in $G$. We distinguish three cases.

Case 1: Suppose that $u_1 \neq u_2$, and that there does not exist a path from $v_1$ to $v_2$, except the green edge $v_1v_2$. Let $a_0 = v_1, a_1, \ldots, a_k = u_1$ be the subdivision of the edge $u_1v_1$ and $b_0 = v_2, b_1, \ldots, b_l = u_2$ the subdivision of the edge $u_2v_2$. We know that there exists an edge $v_2c$ such that $\phi(a_0a_1) = \phi(v_2c)$. It is clear that $c \neq a_1$, thus $m_\phi(a_1) = 1$. (See Figure 3.3(a).) Inductively we find that for every vertex $a'$ in


(a) Case 1. (b) Case 2.

Figure 3.3: Proof of Lemma 3.2.9.

$G'_{a_0}(a_1)$ it holds that $m_\phi(a') = 1$. We conclude that $G'_{a_0}(a_1)$ is a tree. Analogously we find that $G'_{b_0}(b_1)$ is a tree. Thus $G_{v_1}(u_1)$ and $G_{v_2}(u_2)$ are trees. Hence $H$ consists of two black trees connected by a green edge.

Now we can construct a refinement $H'$ of $H$, a tree $T'$ and a suitable morphism $\phi' : H' \to T'$. Copy every branch of $u_1$ and add them to $u_2$ and copy every branch of $u_2$ and add them to $u_1$. Write $H'$ for this graph. Now we see that the two trees of $H'$ are identical up to isomorphism, say they are isomorphic to $T'$. Now we can define $\phi' : H' \to T'$ as the identity map on each of the components, where $\phi'(u_1) = \phi'(u_2)$. Thus $\phi'$ is a suitable morphism. We conclude that $\mathrm{sgon}(H) \leq 2$.

Case 2: Suppose that $u_1 \neq u_2$ and that there exists a path (possibly containing green edges) from $v_1$ to $v_2$. Assume that $\phi(u_1) \neq \phi(u_2)$. Let $a_0 = v_1, a_1, \ldots, v_k = u_1$ be the added vertices on the edge $v_1u_1$ and let $b_0 = v_2, b_1, \ldots, b_l = u_2$ be the added vertices on the edge $v_2u_2$. Assume without loss of generality that $k \leq l$. It is clear that all vertices $a_0, \ldots, a_k, b_0, \ldots, b_l$ lie on the path from $v_1$ to $v_2$. If $\phi(a_1) \neq \phi(b_1)$, then there is a vertex $x$ in the path from $a_1$ to $b_1$ that is mapped to $\phi(v_1)$. This yields a contradiction. Thus $\phi(a_1) = \phi(b_1)$. Inductively we find that $\phi(a_i) = \phi(b_i)$ for all $i \leq k$. We conclude that $\phi(b_k) = \phi(u_1)$. It follows that $\deg(b_k) = \deg(u_1)$. We again distinguish two cases.

Suppose that $\deg(u_1) > 2$. Then $b_k$ has an external added neighbour $w$. We see that $u_1$ has a neighbour $x$ such that $\phi(b_kw) = \phi(u_1x)$. Since $w$ is an external added vertex, it follows that $w \neq x$. Thus $m_\phi(w) = 1$. (See Figure 3.3(b).) Iteratively we see that for every vertex $w'$ in $G'_{b_k}(w)$, it holds that $m_\phi(w') = 1$. Notice that $G'_{b_k}(w)$ is a tree, since $w$ is an external added vertex. Now let $y$ be a leaf in $G'_{b_k}(w)$, and let $y'$ be such that $\phi(y) = \phi(y')$. Then $y'$ is a leaf. It is clear that $y'$ has no green edge incident to it, thus $y'$ is an added vertex. We conclude that we can remove $y$ and $y'$ from $G'$ and $\phi(y)$ from $T$ and still have a suitable morphism. This yields a contradiction with the minimality of $G'$.

Suppose that $\deg(u_1) = 2$ in $G'$. Then the degree of $u_1$ in $G$ is also 2. It follows that $\mathcal{C}_{u_1} \neq \emptyset$. Let $u_1c$ be a green edge. If $c = u_1$, so if $u_1$ has a green loop, then $m_\phi(u_1) = 2$. This yields a contradiction. It is clear that $c \neq b_l$, since $b_l$ is an added vertex. It follows that there are 3 distinct vertices that are mapped to $\phi(u_1)$. This yields a contradiction.

Altogether we conclude that $\phi(u_1) = \phi(u_2)$. Define $H'$ as $G'$ with a green edge $u_1u_2$. Now $H'$ is a refinement of $H$, and $\phi : H' \to T$ is a suitable morphism. We conclude that $\mathrm{sgon}(H) \leq 2$.

Case 3: Suppose that $u_1 = u_2$. Analogous to the second case, we can prove that $m_\phi(u_1) = 2$. Define $H'$ as $G'$ with a green loop at vertex $u_1$. Now $H'$ is a refinement of $H$, and $\phi : H' \to T$ is a suitable morphism. We conclude that $\mathrm{sgon}(H) \leq 2$.

"$\Longleftarrow$": Suppose that $\mathrm{sgon}(H) \leq 2$. Then there exists a refinement $H'$ of $H$ and a suitable morphism $\phi : H' \to T$. Write $u_1$ and $u_2$ for the neighbours of $v_1$ and $v_2$ in $G$. We know that $\phi(u_1) = \phi(u_2)$. Then add leaves $v_1$ and $v_2$ to $u_1$ and $u_2$ and a green edge $v_1v_2$ in $H'$ to obtain $G'$. Now we see that $G'$ is a refinement of $G$. Give the edges $u_1v_1$ and $u_2v_2$ index $r_{\phi'}(u_1v_1) = r_{\phi'}(u_2v_2) = 1$, and give all other edges $e$ index $r_{\phi'}(e) = r_\phi(e)$. Add a leaf $v'$ to $\phi(u_1)$ in $T$ to obtain $T'$. Then we can extend

$\phi$ to $\phi' : G' \to T'$ as follows:

$$\phi'(x) = \begin{cases} \phi(x) & \text{if } x \in H' \\ v' & \text{if } x = v_1, v_2. \end{cases}$$

It is easy to see that $\phi'$ is a suitable morphism, so we conclude that $\mathrm{sgon}(G) \leq 2$. □

**Lemma 3.2.10.** *Rule $S_2^s$ is safe.*

*Proof.* This proof is analogous to the proof of the second and third case in the proof of Lemma 3.2.9, so we omit it. □

**Lemma 3.2.11.** *Rule $L^s$ is safe.*

*Proof.* Let $v$ be the vertex in $G$ to which the rule is applied.

Suppose that $\mathrm{sgon}(G) \leq 2$. Then there exists a refinement $G'$ of $G$ and a suitable morphism $\phi : G' \to T$. Let $u$ be a vertex that is added to the loop $vv$. We distinguish two cases.

Suppose that $m_\phi(v) = 2$. Define $H'$ as the graph $G' \backslash (G'_v(u) \backslash \{v\})$ with a green loop at vertex $v$, then $H'$ is a refinement of $H$. Let $T' = T \backslash \phi(G'_v(u) \backslash \{v\})$. We see that the restricted morphism $\phi : H' \to T'$ is a suitable morphism, so $\mathrm{sgon}(H) \leq 2$.

Suppose that $m_\phi(u) = 1$. Let $v_0 = v, v_1, \dots, v_k = v$ be the vertices that are added to the loop $vv$ of $G$. Let $i$ be such that $\phi(v_i) = \phi(v)$. Let $w$ be a neighbour of $v$ not equal to $v_1$ or $v_{k-1}$. Then there is a neighbour $x$ of $v_i$, not equal to $v_{i-1}$ and $v_{i+1}$, such that $\phi(w) = \phi(x)$. Notice that $x$ is an external added vertex, thus $m_\phi(w) = m_\phi(x) = 1$. Inductively we see that for every vertex $w'$ in $G'_v(w)$ it holds that $m_\phi(w') = 1$. We conclude that $G'_v(w)$ is a tree.

Define $H'$ as $G'_v(w)$, with a green loop at vertex $v$. Notice that $H'$ is a refinement of $H$. Now we can restrict $\phi$ to $H'$ and give every edge $e$ index $r_{\phi'}(e) = 2$ to obtain a suitable morphism: $\phi' : H' \to T'$, where $T' = \phi(G'_v(w))$. We conclude that $\mathrm{sgon}(H) \leq 2$.

Suppose that $\mathrm{sgon}(H) \leq 2$. Then there exists a refinement $H'$ of $H$ and a suitable morphism $\phi : H' \to T$. We know that $m_\phi(v) = 2$. Then add a vertex $u$ to $H'$ with two black edges to $v$ to obtain $G'$. Now we see that $G'$ is a refinement of $G$. Give both edges $uv$ index $r_{\phi'}(uv) = 1$, and give all other edges $e$ index $r_{\phi'}(e) = r_\phi(e)$. Add a leaf $v'$ to $\phi(u)$ in $T$ to obtain $T'$. Then we can extend $\phi$ to $\phi' : G' \to T'$ as follows:

$$\phi'(x) = \begin{cases} \phi(x) & \text{if } x \in H' \\ v' & \text{if } x = u. \end{cases}$$

It is clear that $\phi'$ is a suitable morphism, so we conclude that $\mathrm{sgon}(G) \leq 2$. □

**Lemma 3.2.12.** *Rule $P_1^s$ is safe.*

*Proof.* Let $uv$ be the edge in $G$ to which the rule is applied.

Suppose that $\mathrm{sgon}(G) \leq 2$. Let $G'$ be a refinement of $G$ and $\phi : G' \to T$ a suitable morphism. Let $G_{uv}$ be the set of all vertices that are internal added to the edge $uv$, together with all external added vertices for which all paths to $v$ contain such an internal added vertex. Thus $G_{uv}$ contains all vertices that are internal or external added to the edge $uv$. Now define $H' = G' \backslash G_{uv}$ and $T' = T \backslash (\phi(G_{uv}))$. Write $\phi'$ for the restriction of $\phi$ to $H'$. Notice that $\phi'$ is a suitable morphism and that $H'$ is a refinement of $H$. Thus $\text{sgon}(H) \leq 2$.

Suppose that $\text{sgon}(H) \leq 2$. Let $H'$ be a refinement of $H$ and $\phi : H' \to T$ a suitable morphism. Add an edge $uv$ and a vertex $w$ on this edge to $H'$, to obtain a refinement $G'$ of $G$. Add a vertex $w'$ to $T$ with an edge to $\phi(u)$, to obtain a tree $T'$. Give the edges $uw$ and $vw$ index $r_{\phi'}(uw) = r_{\phi'}(vw) = 1$, and give all other edges $e$ index $r_{\phi'}(e) = r_{\phi}(e)$. Consider the morphism $\phi' : G' \to T'$, defined as

$$\phi'(x) = \begin{cases} \phi(x) & \text{if } x \in H' \\ w' & \text{if } x = w. \end{cases}$$

Notice that $\phi(uw) = \phi(vw)$, since there is a green edge $uv$. We conclude that $\phi'$ is a suitable morphism, thus $\text{sgon}(G) \leq 2$. □

**Lemma 3.2.13.** *Rule* $\boldsymbol{P_2^s}$ *is safe.*

*Proof.* Let $u$ and $v$ be the vertices in $G$ to which the rule is applied.

Suppose that $\text{sgon}(G) \leq 2$. Let $G'$ be a refinement of $G$ and $\phi : G' \to T$ a suitable morphism. If $\phi(u) \neq \phi(v)$, then there are at least three paths that are mapped to the path from $\phi(u)$ to $\phi(v)$ in $T$. This yields a contradiction. Thus $\phi(u) = \phi(v)$. Now we see, analogous to the proof of Lemma 3.2.12, that $\text{sgon}(H) \leq 2$.

Suppose that $\text{sgon}(H) \leq 2$. Then we find analogous to the proof of Lemma 3.2.12, that $\text{sgon}(G) \leq 2$. □

**Lemma 3.2.14.** *Rules* $\boldsymbol{E_1^s}$, $\boldsymbol{E_2^s}$ *and* $\boldsymbol{E_3^s}$ *are safe.*

*Proof.* All these graphs have stable gonality at most 2, so the statement holds true. □

Now we have proven that all rules are safe, thus we have the following lemma:

**Lemma 3.2.15.** *The set of rules* $\mathcal{R}^s$ *is safe for* $\mathcal{G}_2^s$. □

## Completeness

Now we will prove that the set of rules $\mathcal{R}^s$ is complete. Let $G$ be a connected graph. Suppose we obtain $H$ by applying rules from $\mathcal{R}^s$ until no rule is applicable any more. Notice that the graph, formed by the vertices and the black and green edges of $H$ is connected.

We define the graphs $H_1$, $H_2$ and $H_3$ as a single vertex, a vertex with a green loop and two vertices connected by a green edge respectively. These are exactly the graphs that can be reduced to the empty graph by Rules $\boldsymbol{E_1^s}$, $\boldsymbol{E_2^s}$ and $\boldsymbol{E_3^s}$.

**Lemma 3.2.16.** *The set of rules* $\mathcal{R}^s$ *is complete for* $\mathcal{G}_2^s$.

*Proof.* Let $G$ be a graph with $\mathrm{sgon}(G) \leq 2$. Suppose that $H$ is obtained by reducing $G$ and that we cannot apply any rule to $H$ any more. By Lemma 3.2.15 it follows that $\mathrm{sgon}(H) \leq 2$.

Suppose that $H \neq \emptyset$. First we make an observation on the structure of $H$. If there is a double edge between two vertices $u$ and $v$, then removing these two edges yields a disconnected graph, otherwise we could apply Rule $\boldsymbol{P_2^s}$. Let $u_1v_1, \ldots, u_kv_k$ be all double edges in $H$. Let $H_{i,1}, H_{i,2}$ be the two different connected components after removing the edges $u_iv_i$. If there is a degree 2 vertex with a green loop, then removing this vertex yields a disconnected graph, otherwise we could apply rule $\boldsymbol{S_2^s}$. Let $v_1, \ldots, v_l$ be all degree two vertices with a green loop. Let $H'_{i,1}, H'_{i,2}$ be the two different connected components after removing $v_i$. Let $H'$ be the element of

$$\{H_{i,j} \mid 1 \leq i \leq k, j \in \{1,2\}\} \cup \{H'_{i,j} \mid 1 \leq i \leq l, j \in \{1,2\}\}$$

with the minimum number of vertices. Notice that there is at most one vertex $v$ in $H'$ with $\deg_{H'}(v) \neq \deg_H(v)$. Now we can say the following about $H'$.

- If $H'$ contains only one vertex, then we could have applied Rule $\boldsymbol{T_1^s}$, $\boldsymbol{T_2^s}$, $\boldsymbol{S_1^s}$, $\boldsymbol{S_2^s}$, $\boldsymbol{E_1^s}$ or $\boldsymbol{E_2^s}$. Thus $H'$ contains at least 2 vertices.
- If there is a vertex that is incident to more than one green edge, then $\mathrm{sgon}(H) \geq 3$ by Lemma 3.2.2. So we can assume that no vertex is incident to more than one green edge.
- If $H'$ contains a vertex $u \neq v$ of degree 0, then $\deg_H(u) = \deg_{H'}(u) = 0$. We see that $\mathcal{C}_u = \{(u, w)\}$ with $u \neq w$, because $H'$ is connected and contains at least two vertices. By Lemma 3.2.4 it follows that $\deg_H(u) = \deg_H(w) = 0$. Since $H$ is connected it follows that $H = H_3$, so we can apply Rule $\boldsymbol{E_3^s}$. This yields a contradiction. So we can assume that $H$ does not contain vertices with degree 0.
- If $H$ contains a leaf $u \neq v$, then $\deg_{H'}(u) = \deg_H(u) = 1$. We see that $u$ is incident to a green edge $uw$, with $\deg_H(w) \neq 1$, otherwise we could have applied Rule $\boldsymbol{T_1^s}$, $\boldsymbol{T_2^s}$ or $\boldsymbol{T_3^s}$. By Lemma 3.2.4, it follows that $\mathrm{sgon}(H) \geq 3$. So we can assume that $H$ does not contain leaves.
- If $H$ contains a vertex $u \neq v$ of degree 2, then we see that $\mathcal{C}_u = \{u, w\}$ with $u \neq w$, by Rules $\boldsymbol{S_1^s}$ and $\boldsymbol{S_2^s}$ and by the choice of $H'$.
- We see that $H'$ does not contain black loops because of Rule $\boldsymbol{L^s}$.
- By Rules $\boldsymbol{P_1^s}$ and $\boldsymbol{P_2^s}$ and by the choice of $H'$ it follows that $H'$ has no multiple edges.

Write $H''$ for the graph obtained from $H'$ by removing all green loops and colouring all green edges of $H'$ black. Altogether we see that $H''$ is a simple graph with at least two vertices and every vertex, except at most one, has degree at least 3. It follows that $H''$ has treewidth at least 3.

If we change the colour of all green edges to black, we see that all rules are deletions of vertices or edges, contractions of edges and/or additions of loops. Since the set of graphs with treewidth at most $k$ is closed under these operations, we see that $\mathrm{tw}(G) \geq \mathrm{tw}(H'') \geq 3$. But then it follows that $\mathrm{sgon}(G) \geq \mathrm{tw}(G) \geq 3$. This yields a contradiction.

We conclude that $H = \emptyset$, and thus $G\mathcal{R}^{s*}\emptyset$. $\square$

Lemma 3.2.15 says that $\mathcal{R}^s$ is safe for $\mathcal{G}_2^s$ and from 3.2.16 it follows that $\mathcal{R}^s$ is complete for $\mathcal{G}_2^s$. So together this proves Theorem 3.2.5. We conclude that we can use this set of rules to recognize graphs with stable gonality at most 2.

## 3.3 Reduction rules for stable divisorial gonality

In this section we show a set of reduction rules to decide whether stable divisorial gonality is at most two, this set is similar to the set of rules for stable gonality. In this section, we use a different concept of constraints. These are based on the constraints used by Jelco Bodewes for recognizing graphs of divisorial gonality 2 [13, 14].

### Notation

To check whether a graph has stable divisorial gonality two or lower, we have to check whether there is a refinement of our graph such that there exists a divisor with degree 2 and rank at least 1. Constraints in this case are used to restrict which divisors we consider and what sets we are allowed to fire after reduction. The pairs in the constraints place the following restrictions on what divisors and firing sets are allowed:

**Definition 3.3.1.** Given a constraint $r = (u, v)$, we say that a divisor $D$ *satisfies* $r$ if it is equivalent to an effective divisor after removing one chip from $v$ and one chip from $w$. In addition, any set that we fire should contain either both $u$ and $v$ or neither.

Note that in the case that $u = v$ 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.

We will refer to constraints as red edges. We call an effective divisor of degree 2, rank greater than or equal to 1, and that satisfies all conditions given by the constraints a *suitable* divisor. A graph with constraints has stable divisorial gonality at most 2 if there exists a refinement such that there is a suitable divisor. Again, let $\mathcal{G}_2^{sd}$ be the set of all graphs with constraints with stable divisorial gonality at most 2.

**Lemma 3.3.2.** *Let $G$ be a graph. If there is a vertex $v$ with $|\mathcal{C}_v| > 1$, then* $\mathrm{sdgon}(G) \geq 3$*.*

*Proof.* Let $(v, w)$ and $(v, w')$, with $w \neq w'$, be two constraints that contain $v$. Suppose that $\mathrm{sdgon}(G) = 2$. Let $G'$ be a refinement of $G$ such that $\mathrm{dgon}(G') = 2$. We first look at the possibility where $v = w'$. Then any suitable divisor on $G'$ must be equivalent to the divisor $D$ with $D(v) = 2$, but also equivalent to the divisor $D'$ with $D'(v) = 1$ and $D'(w) = 1$. It follows that these divisors are equivalent to each other. Let $A_0, \ldots, A_k$ be the level set decomposition of the transformation from $D$ into $D'$. Note that, since we have the constraint $(v, w)$, all sets $A_i$ containing $v$ also contain $w$. Note that $v \in A_0$, since it is the only vertex with chips, which means that it must also contain $w$. But then the number of chips on $w$ cannot increase by firing $A_0, \ldots, A_k$, this yields a contradiction. We conclude that $G \notin \mathcal{G}_2^{sd}$.

The other possibility is that $v \neq w$ and $v \neq w'$. This means that any suitable divisor should be equivalent to the divisor $D$ with $D(v) = 1$ and $D(w) = 1$, and equivalent to the divisor $D'$ with $D'(v) = 1$ and $D'(w') = 1$. Let $A_0, \ldots, A_k$ be the level set decomposition of the transformation of $D$ into $D'$. 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 in $D$ the only vertices that have chips are $v$ and $w$, it follows that either $v$ or $w$ is an element of $A_0$. It follows that $A_0$ contains $w'$. This means that the number of chips on $w'$ cannot increase, this yields a contradiction. We conclude that $G \notin \mathcal{G}_2^{sd}$.

We conclude that $\mathrm{sdgon}(G) > 2$. □

**Lemma 3.3.3.** *Let $G$ be a graph where every leaf is incident to a red edge, so if* $\deg(u) = 1$ *then* $|\mathcal{C}_u| > 0$ *for all $u$. Suppose that $v$ is a leaf and $(v, w)$ is a red edge. If* $\deg(w) \neq 1$*, then* $\mathrm{sdgon}(G) \geq 3$.

*Proof.* Assume on the contrary that $\deg(w) \neq 1$ and $\mathrm{sdgon}(G) = 2$, and let $G'$ be a refinement of $G$ with $\mathrm{dgon}(G') = 2$. Let $D$ be the divisor on $G'$ with $D(v) = D(w) = 1$. Since we have the constraint $(v, w)$ and $G' \in \mathcal{G}_2^{sd}$, $D$ is a suitable divisor.

Suppose that $\deg_G(w) = 0$. Then the graph $G'$ consists of two connected components, write $C_v$ for the component containing $v$ and $C_w$ for the component containing $w$. Notice that every vertex in $C_w \backslash \{w\}$ is an added vertex. The chip on $w$ will never leaf $C_w$, thus $G' \backslash C_w$ is a tree. It follows that there is a leaf $x$, with a constraint $(x, y)$. Since there will never be two chips on $C_v$, we see that $y \in C_w$. Since all vertices in $C_w$, except $w$, are added, it follows that $y = w$. This yields a contradiction with Lemma 3.3.2.

Suppose that $\deg_G(w) > 1$, then $\deg_{G'}(w) > 1$ too. We first consider the case where $w$ is not a cut-vertex in $G'$. Let $u$ be the neighbour of $v$. Since $w$ is not a cut-vertex, we see that $w \neq u$. Consider the transformation from $D$ to a divisor $D'$ with $D'(u) \geq 1$. Let $A_0$ be the first firing set in the level set 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 neighbour $w_i$ 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 neighbour $w_i \neq u$ is in $A_0$, then somewhere on its path to $u$ must be an edge that crosses between $A_0$ and its complement $A_0^c$. But such a crossing edge would imply that we are not allowed to fire $A_0$, since no vertex on this path contains a chip. It follows that none of the neighbours of $w$ are in $A_0$. Since $w$ has degree at least two, and only one chip, it follows that we are not allowed to fire the set $A_0$. This yields 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 by Lemma 3.3.2. Since $D$ has rank greater than zero it then follows that $C_x$ must be a tree. This means $C_x$ must contain a vertex $x$ of degree one. We know however that $x$ must have a constraint $(x, y)$ where $y$ is a vertex with degree greater than one. Thus $D$ is equivalent to $D''$ with $D''(x) = D''(y) = 1$. We now consider the possible locations of $y$.

Suppose that $y \in C_x$. As mentioned before, $D$ cannot be equivalent to a divisor with two chips on $C_x$, so it follows that $y \notin C_x$. 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 $x, y \notin A_0$. This implies that $w$ has at least one neighbour $w_1$ in $C_y$, with $w_1 \notin A_0$, namely the first vertex on a path from $w$ to $y$. But $w$ also has at least one neighbour $w_2$ in $C_x$, with $w_2 \notin C_x$, namely the first vertex on the path from $w$ to $x$. This means $w$ has two neighbours that it will send a chip to, but $w$ only has one chip. This yields a contradiction. We conclude there can be no such constraint $(x, y)$.

We conclude that if $\deg(w) \neq 1$, then $\mathrm{sdgon}(G) > 2$. □

## Reduction rules

We will now state all rules. When a rule adds a red edge $uv$, and there already exists such a red edge, then the set of constraints does not change. The reduction rules for stable divisorial hyperelliptic graphs are almost the same as the rules for stable hyperelliptic graphs. Instead of green edges we use red edges, and we replace Rule $\boldsymbol{S_1^s}$ and $\boldsymbol{L^s}$ by new Rules $\boldsymbol{S_{1a}^{sd}}$, $\boldsymbol{S_{1b}^{sd}}$ and $\boldsymbol{L^{sd}}$, see Figure 3.4 for the new rules.

**Rule $\boldsymbol{T_1^{sd}}$ (=$\boldsymbol{T_1^s}$).** *Let $v$ be a leaf with $\mathcal{C}_v = \emptyset$. Let $u$ be the neighbour of $v$. Contract the edge $uv$.*

**Rule $\boldsymbol{T_2^{sd}}$ (=$\boldsymbol{T_2^s}$).** *Let $v$ be a leaf with $\mathcal{C}_v = \{(v, v)\}$. Let $u$ be the neighbour of $v$. Contract the edge $uv$.*

**Rule $\boldsymbol{S_{1a}^{sd}}$.** *Let $v$ be a vertex of degree 2 with $\mathcal{C}_v = \emptyset$. Let $u$ be the only neighbour of $v$. Remove $v$ and add a red loop to $u$.*

**Rule $\boldsymbol{S_{1b}^{sd}}$.** *Let $v$ be a vertex of degree 2 with $\mathcal{C}_v = \emptyset$. Let $u_1$ and $u_2$ be the two neighbours of $v$, with $u_1 \neq u_2$. Contract the edge $u_1v$.*

**Rule $\boldsymbol{T_3^{sd}}$ (=$\boldsymbol{T_3^s}$).** *Let $G$ be a graph where every leaf and every degree 2 vertex is incident to a red edge. Let $v_1$ and $v_2$ be two leaves that are connected by a red edge. Let $u_1$ and $u_2$ be their neighbours. Contract the edges $u_1v_1$ and $u_2v_2$.*

Rule $\boldsymbol{S_{1a}^{sd}}$ Rule $\boldsymbol{L^{sd}}$

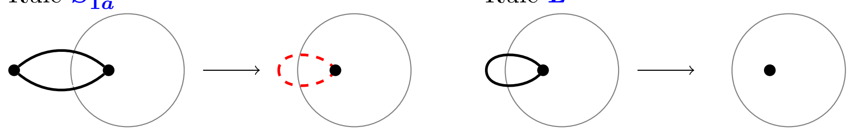


Figure 3.4: The reduction rules for stable divisorial gonality that are different from the rules for stable gonality.

**Rule $\boldsymbol{S_2^{sd}}$ (=$\boldsymbol{S_2^{s}}$).** *Let $G$ be a graph where every leaf and every degree 2 vertex is incident to a red edge. Let $v$ be a vertex of degree 2 with a red loop, such that there exists a path from $v$ to $v$ in the black and red graph $G$. Let $u_1$ and $u_2$ be the neighbours of $v$. Remove $v$ and connect $u_1$ and $u_2$ with a red edge.*

**Rule $\boldsymbol{L^{sd}}$.** *Let $v$ be a vertex with a loop. Remove all loops from $v$.*

**Rule $\boldsymbol{P_1^{sd}}$ (=$\boldsymbol{P_1^{s}}$).** *Let $uv$ be an edge. Suppose that there also exists a red edge from $u$ to $v$. Remove the black edge $uv$.*

**Rule $\boldsymbol{P_2^{sd}}$ (=$\boldsymbol{P_2^{s}}$).** *Let $u, v$ be vertices, such that $|E(u, v)| > 1$. Let $e_1$ and $e_2$ be two of those edges. If there exists another path, possibly containing red edges, from $u$ to $v$, then remove $e_1$ and $e_2$ and add a red edge from $u$ to $v$.*

**Rule $\boldsymbol{E_1^{sd}}$ (=$\boldsymbol{E_1^{s}}$).** *Let $G$ be the graph consisting of a single vertex $v$ with $\mathcal{C}_v = \emptyset$. Remove $v$.*

**Rule $\boldsymbol{E_2^{sd}}$ (=$\boldsymbol{E_2^{s}}$).** *Let $G$ be the graph consisting of a single vertex $v$ with a green loop. Remove $v$.*

**Rule $\boldsymbol{E_3^{sd}}$ (=$\boldsymbol{E_3^{s}}$).** *Let $G$ be the graph consisting of two vertices $u$ and $v$ that are connected by a green edge. Remove $u$ and $v$.*

We write $\mathcal{R}^{sd}$ for the set of these reduction rules. We can now state our main theorem.

**Theorem 3.3.4.** *The set of rules $\mathcal{R}^{sd}$ is safe and complete for $\mathcal{G}_2^{sd}$.*

## Safeness

We will show that the set $\mathcal{R}^{sd}$ is safe for $\mathcal{G}_2^{sd}$.

**Lemma 3.3.5.** *Rule $\boldsymbol{T_1^{sd}}$ is safe.*

*Proof.* This follows from Proposition 1.4.6. □

**Lemma 3.3.6.** *Rule $\boldsymbol{T_2^{sd}}$ is safe.*

*Proof.* This proof is analogous to the proof of Lemma 3.3.5. □

**Lemma 3.3.7.** *Rule $\boldsymbol{S_{1a}^{sd}}$ is safe.*

*Proof.* Let $v$ be the vertex in $G$ to which the rule is applied.

Let $u$ be the neighbour of $v$. Suppose that $\mathrm{sdgon}(G) \leq 2$. Let $G'$ be a refinement of $G$ such that there exists a suitable divisor $D$. Let $C$ be the cycle through $v$ and $u$. Notice that $D$ is equivalent to a divisor $D'$ with two chips on $C$. If $G \backslash (C \backslash \{u\})$ is a tree, then we are done. Otherwise we see that $D'$ is equivalent to $D''$, where $D''(u) = 2$. Let $H'$ be $G' \backslash (C \backslash \{u\})$ with a red loop at $u$. We see that $H'$ is a refinement of $H$ and $D''$ is a suitable divisor for $H'$, thus $\mathrm{sdgon}(H) \leq 2$.

Suppose that $\mathrm{sdgon}(H) \leq 2$. Then there exists a refinement $H'$ of $H$ such that $D$, with $D(u) = 2$, is a suitable divisor. Let $G'$ be $H'$ without the red loop on $u$ and with a vertex $v$ with two edges to $u$. Then $G'$ is a refinement of $G$. It is clear that $D$ is a suitable divisor for $G'$ too. We conclude that $\mathrm{sdgon}(G) \leq 2$. □

**Lemma 3.3.8.** *Rule* $S_{1b}^{sd}$ *is safe.*

*Proof.* This follows from Proposition 1.4.6. □

**Lemma 3.3.9.** *Rule* $T_3^{sd}$ *is safe.*

*Proof.* Let $v_1$ and $v_2$ be the vertices in $G$ to which the rule is applied.

Suppose that $\mathrm{sdgon}(G) \leq 2$. Let $G'$ be a minimum refinement of $G$ and $D$ a suitable divisor on $G'$. Let $u_1$ and $u_2$ be the neighbours of $v_1$ and $v_2$ in $G$. We distinguish three cases.

Case 1: Suppose that $u_1 \neq u_2$, and that there does not exist a path from $v_1$ to $v_2$, except the red edge $v_1v_2$. Then we can reach all vertices in $G_{v_1}(u_1)$ with only one chip, thus $G_{v_1}(u_1)$ is a black tree. So $G_{v_1}(u_1)$ contains a leaf that is not incident to a red edge. This yields a contradiction with the minimality of $G'$.

Case 2: Suppose that $u_1 \neq u_2$ and that there exists a path $P$, possibly containing red edges, from $v_1$ to $v_2$. Assume that $D \nsim D'$ where $D'$ is the divisor such that $D'(u_1) = D'(u_2) = 1$. Let $a_0 = v_1, a_1, \ldots, a_k = u_1$ be the added vertices on the edge $v_1u_1$ and let $b_0 = v_2, b_1, \ldots, b_l = u_2$ be the added vertices on the edge $v_2u_2$. Assume that $k < l$. It is clear that all vertices $a_0, \ldots, a_k, b_0, \ldots, b_l$ lie on $P$. Notice that firing the sets $\{a_i, b_i \mid i \leq j\}$ for all $j \leq k$ results in the divisor $D_k$ with $D_k(a_k) = D_k(b_k) = 1$. Thus $D_k(u_1) = 1$. Since $b_k$ is an internal added vertex and $G'$ is a minimum refinement, we see that $\deg(b_k) = 2$.

Suppose that $u_1$ is incident to a red edge $u_1x$. We know that $x \neq b_k$, since $b_k$ is an added vertex. Let $D''$ be the divisor with $D''(u_1) = D''(x) = 1$. Let $A_0, \ldots, A_s$ be the level set decomposition of the transformation from $D_k$ to $D''$. We see that $u_1$ cannot lose its chip. Thus $b_k$ fires its chip to one of its neighbours when we fire $A_0$. But then we see that the cut of $A_0$ is at least two, and we can only fire one chip. This yields a contradiction. We conclude that $u_1$ is not incident to a red edge.

By the conditions of the rule it follows that $\deg(u_1) \geq 3$. Let $w \notin P$ be a neighbour of $u_1$. Now we see that $G'_{u_1}(w)$ is a black tree. It follows that $G'_{u_1}(w)$ contains a leaf that is not incident to a red edge. Since $G'$ is a minimum refinement, this yields a contradiction. Altogether we conclude that $k = l$.

Let $P_1, P_2$ be the two arcs of $P$ between $u_1$ and $u_2$. Notice that, if there are two chips on $P$, then they are either on $u_1$ and $u_2$ or on the same arc $P_i$. Suppose that there are divisors $E, E'$ such that $E \sim E'$ and that there is a set $A$ in the level set decomposition of $E' - E$ such that $u_1 \in A$ and $u_2 \notin A$. It follows that there is a chip fired along each of the arcs $P_1$ and $P_2$. This yields a contradiction. We conclude that for every firing set it holds that either $u_1$ and $u_2$ are both fired or they are both not fired.

Now let $H'$ be $G'$ without the red edge $v_1v_2$ and with a red edge $u_1u_2$. We see that $D$ is a suitable divisor for $H'$ as well. Thus $\mathrm{sdgon}(H) \leq 2$.

Case 3: Suppose that $u_1 = u_2$. This case is analogous to case 2.

Suppose that $\mathrm{sdgon}(H) \leq 2$. Then it is clear that $\mathrm{sdgon}(G) \leq 2$. □

**Lemma 3.3.10.** *Rule* $S_2^{sd}$ *is safe.*

*Proof.* This proof is analogous to the proof of cases two and three in the proof of Lemma 3.3.9, so we omit it. □

**Lemma 3.3.11.** *Rule* $\boldsymbol{L^{sd}}$ *is safe.*

*Proof.* There will never be a chip fired over a loop, so loops do nothing for the stable divisorial gonality. Thus $\mathrm{sdgon}(G) \leq 2$ if and only if $\mathrm{sdgon}(H) \leq 2$. □

**Lemma 3.3.12.** *Rule* $\boldsymbol{P_1^{sd}}$ *is safe.*

*Proof.* Let $uv$ be the edge in $G$ to which the rule is applied.

Suppose that $\mathrm{sdgon}(G) \leq 2$. Let $G'$ be a refinement of $G$ and $D$ a suitable divisor with $D(u) = D(v) = 1$. Let $G_{uv}$ be the set of all vertices that are internal added to the edge $uv$, together with all external added vertices for which all paths to $v$ contain such an internal added vertex. Thus $G_{uv}$ contains all vertices that are internal or external added to the edge $uv$. Now define $H' = G' \backslash G_{uv}$. Look at the divisor $D$ on $H'$ and notice that $D$ is a suitable divisor. Observe that $H'$ is a refinement of $H$. Thus $\mathrm{sdgon}(H) \leq 2$.

Suppose that $\mathrm{sdgon}(H) \leq 2$. Let $H'$ be a refinement of $H$ and $D$ a suitable divisor. Add an edge $uv$ to $H'$, to obtain a refinement $G'$ of $G$. We see that $D'$ is a suitable divisor for $G'$, thus $\mathrm{sdgon}(G) \leq 2$. □

**Lemma 3.3.13.** *Rule* $\boldsymbol{P_2^{sd}}$ *is safe.*

*Proof.* Let $e_1$, $e_2$ be the edges from $u$ to $v$ in $G$ to which the rule is applied.

Suppose that $\mathrm{sdgon}(G) \leq 2$. Then there exists a refinement $G'$ of $G$ such that there exists a suitable divisor on $G'$. Let $D$ be a suitable divisor on $G'$ with a chip on $u$. We will show that there is a suitable divisor that has a chip on both $u$ and $v$: Assume that $D(v) = 0$, then there should be a suitable divisor $D'$ with $D'(v) = 1$ and $D \sim D'$. This implies there is a level set decomposition $A_0, \ldots, A_k$ of the transformation from $D$ to $D'$.

If none of the subsets contains $u$ then it follows that $D'(u) = 1$ and we are done. Otherwise let $i$ be the smallest index such that $A_i$ contains $u$ and let $D_i$ be the divisor before firing $A_i$. Suppose that $D_i(v) = 0$, then it follows that $v \notin A_i$. Notice that there are three disjoint paths from $u$ to $v$, so the minimum cut between $u$ and $v$ is at least 3. It follows that there are at least three chips fired by $A_i$. But we only have two chips, this yields a contradiction. We conclude that $D_i(v) = 1$.

Also by the fact that the minimum cut between $u$ and $v$ is at least three it follows that firing a subset $A$ can only be valid if $A$ contains either both $u$ and $v$ or neither.

Define $H'$ as $G'$ after removing the edges $e_1, e_2$ and adding a red edge $uv$. Notice that $H'$ is a refinement of $H$. We conclude that the divisor $D_i$ is a suitable divisor on $H'$, and $\mathrm{sdgon}(H) \leq 2$.

Now we assume that $\mathrm{sdgon}(H) \leq 2$. From this it follows that there exists a refinement $H'$ of $H$ and a suitable divisor $D$ on $H'$ with a chip on $u$ and a chip on $v$. Define $G'$ as $H'$ after adding the edges $e_1, e_2$ and notice that $G'$ is a refinement of $G$. Consider the divisor $D$ on $G'$. We see that $D$ is suitable on $G'$ and thus $\mathrm{sdgon}(G) \leq 2$. □

**Lemma 3.3.14.** *Rules* $\boldsymbol{E_1^{sd}}$, $\boldsymbol{E_2^{sd}}$ *and* $\boldsymbol{E_3^{sd}}$ *are safe.*

*Proof.* All those graphs have stable gonality at most 2, so the statement holds true. □

Now we have proven that all rules are safe, so we can conclude the following:

**Lemma 3.3.15.** *The set of rules $\mathcal{R}^{sd}$ is safe for $\mathcal{G}_2^{sd}$.* □

## Completeness

Now we will prove that $\mathcal{R}^{sd}$ is complete for $\mathcal{G}_2^{sd}$, i.e. if $G \in \mathcal{G}_2^{sd}$, then $G\mathcal{R}^{sd*}\emptyset$.

We define the graphs $H_1$, $H_2$ and $H_3$ as a single vertex, a vertex with a red loop and two vertices connected by a red edge respectively, these are the graphs that can be reduced to the empty graph by rules $\boldsymbol{E_1^{sd}}$, $\boldsymbol{E_2^{sd}}$ and $\boldsymbol{E_3^{sd}}$.

Let $G$ be a connected graph. Suppose we obtain $H$ by applying rules from $\mathcal{R}^s$ until no rule is applicable any more. Notice that the graph, formed by the vertices and the black and red edges of $H$ is connected.

**Lemma 3.3.16.** *The set of rules $\mathcal{R}^{sd}$ is complete for $\mathcal{G}_2^{sd}$.*

*Proof.* Let $G$ be a graph with $\text{sdgon}(G) \leq 2$. Suppose that $H$ is obtained by reducing $G$ and that no rule can be applied to $H$. By 3.3.15 it follows that $\text{sdgon}(H) \leq 2$.

Suppose that $H \neq \emptyset$. As in the proof of Theorem 3.2.16, if there are two edges between the vertices $u$ and $v$, then removing these edges leads to $H$ being disconnected. And if there is a degree 2 vertex $v$ with a red loop, then removing $v$ yields a disconnected graph. Let $H'$ be the smallest connected component that can be created by removing two parallel edges or a degree 2 vertex, as in the proof of Theorem 3.2.16.

Now we colour all red edges black and remove all loops to obtain $H''$, as in the proof of Theorem 3.2.16. Then we see that $H''$ is a simple graph that contains at least two vertices and all vertices, except at most one, have degree at least three. Thus $H''$ has treewidth at least three. It follows that $\text{sdgon}(G) \geq \text{tw}(G) \geq \text{tw}(H'') \geq 3$.

We conclude that if $\text{sdgon}(G) \leq 2$, then $G\mathcal{R}^{sd*}\emptyset$. □

Lemma 3.3.15 shows that the set of reduction rules $\mathcal{R}^{sd}$ is safe and Lemma 3.3.16 shows that this set is complete. So together this proves Theorem 3.3.4. Thus we can use the set $\mathcal{R}^{sd}$ to recognize graphs of stable divisorial gonality at most 2.

# 3.4 Algorithms using the reduction rules

We can use the reduction rules of Section 3.2 and 3.3 to obtain algorithms that decide in polynomial time whether a graph is stable (divisorial) hyperelliptic.

For this, we introduce a new rule, see Figure 3.5:

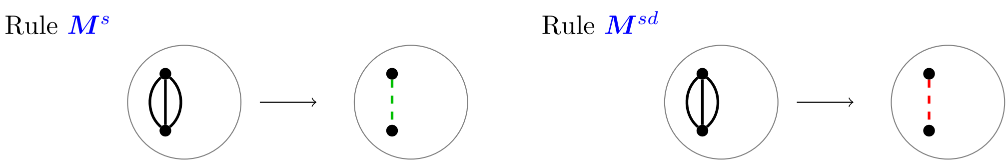



Figure 3.5: An extra reduction rule for stable and stable divisorial gonality.

**Rule $\boldsymbol{M^s}$.** *Let $u, v$ be vertices, such that $|E(u,v)| \geq 3$. Remove all edges in $E(u,v)$ and add a green edge from $u$ to $v$.*

It is clear that this rule is the same as first applying Rule $\boldsymbol{P_2^s}$ and then applying Rule $\boldsymbol{P_1^s}$ to all remaining edges $uv$. For stable divisorial gonality we introduce a similar rule.

**Rule $\boldsymbol{M^{sd}}$.** *Let $u, v$ be vertices, such that $|E(u,v)| \geq 3$. Remove all edges in $E(u,v)$ and add a red edge from $u$ to $v$.*

This is again the same as first applying Rule $\boldsymbol{P_2^{sd}}$ and then applying Rule $\boldsymbol{P_1^{sd}}$ to all remaining edges $uv$.

All applications of these rules can be done in $O(m)$ time at the start of the algorithm, after which we know that no pair of vertices can have more than two edges between them. By application of Rule $\boldsymbol{L^s}$ and $\boldsymbol{L^{sd}}$ we can also ensure in $O(m)$ time that no loops exist.

By Lemma 2.2.11 we know that treewidth is a lower bound for gonality. So if $\text{tw}(G) > 2$, then we know that $\text{sgon}(G) > 2$ and $\text{sdgon}(G) > 2$. So if the treewidth is at least 3, the algorithm can terminate. We can check in linear time whether the treewidth of a graph is at most 2. In the rest of the algorithm we can assume that the treewidth is at most 2.

If there is a vertex which is incident to more than one green or red edge, it follows by Lemma 3.2.2 and 3.3.2 that $\text{sgon}(G) > 2$ and $\text{sdgon}(G) > 2$. So if there is a vertex which is incident to more than one green or red edge, the algorithm can terminate. We can check in linear time whether there is such a vertex. In the rest of the algorithm we can assume that every vertex is incident to at most 1 green or red edge.

Now we can repeatedly apply a rule, until none is applicable. For each of the rules, one can test in polynomial time for a given graph (with green or red edges) if the rule can be applied to the graph, and if so, the rule can be applied in polynomial time.

We claim that we apply at most $O(n)$ rules. Consider the following potential function $f$: let $f(G) = n + 2m + g$ for a graph $G$ with $n$ vertices, $m$ (black) edges, and $g$ green or red edges. We know that every vertex is incident to at most 1 green or red edge, thus $g \leq n$. The number of edges is at most $4n$: Simple graphs of treewidth $k$ and $n$ vertices have at most $kn$ edges. It follows that the underlying simple graph has at most $2n$ edges. By our previous steps, there are at most 2 edges between a pair of vertices and no loops, so there are at most $4n$ edges left. We conclude that for a graph $G$ with treewidth at most 2, it holds that $f(G) = O(n)$. The application of each rule decreases $f(G)$ by at least one. So we apply at most $O(n)$ rules.

So we have polynomial time algorithms.

Using Courcelle's theorem [21], there is an implementation that leads to algorithms running in $O(m + n\log(n))$ time. For details, see [14, Section 7].

# 4 Complexity of computing gonality

Computing divisorial gonality is proven to be NP-hard by Gijswijt [26]. In this chapter, we extend his method to show that computing stable divisorial gonality is NP-hard.

## 4.1 Complexity theory

First we will give a short and informal introduction into complexity theory. This introduction is based on [19, Chapter 34] and [29, Chapter 9], which contain a more formal introduction.

The class NP is a class of *decision problems*; these are problems with as answer 'yes' or 'no'. An example of a decision problem is the following. An *independent set* is a subset $A \subseteq V$ of the vertices of a graph such that for any two vertices $u, v \in A$, there is no edge $uv$. By $\alpha(G)$ we denote the size of the largest independent set in a graph $G$. The *independent set problem* asks the following: given a simple graph $G$ and an integer $k$, is there a subset of $V(G)$ of size $k$ that is an independent set, i.e., does it hold that $\alpha(G) \geq k$?

The class NP consists of all decision problems that are 'verifiable' in polynomial time. We illustrate what we mean by this with an example. An *instance* of the independent set problem is a pair $(G, k)$ of a graph $G$ and an integer $k$. We call an instance $(G, k)$ a 'yes'-instance, if there is an independent set of size $k$ in $G$ and a 'no'-instance otherwise. For a 'yes'-instance there exists a *certificate*: an independent set $A \subseteq V(G)$ of size at least $k$. Notice that this certificate has polynomial size. For an instance $(G, k)$ and a set $B \subseteq V(G)$, we can check in polynomial time whether $B$ is a certificate for this instance. The class NP is the class of decision problems with the following property: for every 'yes'-instance there exists a certificate $A$ of polynomial size and it can be verified that $A$ is indeed a certificate in polynomial time.

A decision problem is NP-hard if it is 'at least as hard as any problem in NP'. To explain this, we introduce the notion of polynomial-time reductions. A *polynomial-time reduction* from a decision problem $\mathcal{A}$ to a decision problem $\mathcal{B}$ is a map that transforms every instance $A$ of $\mathcal{A}$ in polynomial time to an instance $B$ of $\mathcal{B}$ in such a way that $A$ is a 'yes'-instance if and only if $B$ is a 'yes'-instance. If there is a polynomial-time reduction from $\mathcal{A}$ to $\mathcal{B}$, then we write $\mathcal{A} \preceq \mathcal{B}$. Suppose that $\mathcal{A} \preceq \mathcal{B}$. We see that, if there is an algorithm that determines in polynomial time for any instance $B$ of $\mathcal{B}$ whether $B$ is a 'yes'-instance, then there is an algorithm that determines in polynomial time for any instance $A$ of $\mathcal{A}$ whether $A$ is a 'yes'-instance: first reduce $A$ to an instance of $\mathcal{B}$, and then use the algorithm for $\mathcal{B}$. In this sence $\mathcal{B}$ is 'harder' then $\mathcal{A}$. A problems $\mathcal{A}$ is NP-hard if it has the following property: for any problem $\mathcal{B} \in$ NP it holds that $\mathcal{B} \preceq \mathcal{A}$. Notice that to prove that a problem $\mathcal{A}$ is NP-hard, it suffices to prove that there is an NP-hard problem $\mathcal{B}$ such that $\mathcal{B} \preceq \mathcal{A}$.

A problem is NP-complete if it is in NP and NP-hard. The independent set problem is known to be NP-complete.

## 4.2 Stable divisorial gonality is NP-hard

We define the *divisorial gonality problem* to be the following: let $G$ be a graph and $k$ an integer, does it hold that $\operatorname{dgon}(G) \leq k$? And we define the *stable divisorial gonality problem* to be the following: let $G$ be a graph and $k$ an integer, does it hold that $\operatorname{sdgon}(G) \leq k$? It is proven that the divisorial gonality problem is NP-hard by Gijswijt [26]. In this section we extend this proof to a proof that the stable divisorial gonality problem is NP-hard.

For this proof, we will make a polynomial-time reduction from the independent set problem to the stable divisorial gonality problem. Let $G$ be a simple graph. Define $M = 3|V(G)| + 2|E(G)| + 2$. We will construct a *reduction graph $H$ of $G$* with $M - 1$ vertices such that $\operatorname{sdgon}(H) = 4|V(G)| + |E(G)| + 1 - \alpha(G)$.

Let $H$ be a graph with a single vertex $t$. For every vertex $v \in V(G)$, add three vertices $\tilde{v}, v', v_t$ to $H$. Add $M$ parallel edges from $t$ to $v_t$, three parallel edges from $v_t$ to $v'$ and $M$ parallel edges from $v'$ to $\tilde{v}$. For every edge $e = uv \in E(G)$, make two vertices $e_u, e_v$. Add an edge $e_ue_v$, $M$ parallel edges from $e_u$ to $\tilde{u}$ and $M$ parallel edges from $e_v$ to $\tilde{v}$ to $H$. Now we see that $H$ contains $M - 1$ vertices. See Figure 4.1 for an example.

We will now prove that $\operatorname{sdgon}(H) = 4|V(G)| + |E(G)| + 1 - \alpha(G)$ for a reduction graph $H$ of $G$.

**Lemma 4.2.1.** *Let $G$ be a simple graph and $H$ its reduction graph. Then* $\operatorname{sdgon}(H) \leq 4|V(G)| + |E(G)| + 1 - \alpha(G)$.

*Proof.* Let $A$ be an independent set in $G$ of size $\alpha(G)$. Order the vertices in $V \backslash A$: $u_1, \ldots, u_k$, where $k = |V(G)| - \alpha(G)$. For every edge $e = e_ue_v$ we indicate a head and a tail: if $e$ has an endpoint in $A$, call this endpoint tail and the other endpoint head. If $e$ has two endpoints $u_i, u_j \notin A$ with $i < j$, then call $u_i$ tail and $u_j$ head. Look at the divisor $D$ defined by:

$$D(x) = \begin{cases} 1 & \text{if } x = t, \\ 1 & \text{if } x = \tilde{v}, \\ 1 & \text{if } x = v' \text{ and } v \in A, \\ 0 & \text{if } x = v' \text{ and } v \notin A, \\ 1 & \text{if } x = v_t \text{ and } v \in A, \\ 3 & \text{if } x = v_t \text{ and } v \notin A, \\ 1 & \text{if } x = e_u \text{ and } u \text{ tail}, \\ 0 & \text{if } x = e_v \text{ and } v \text{ head}. \end{cases}$$

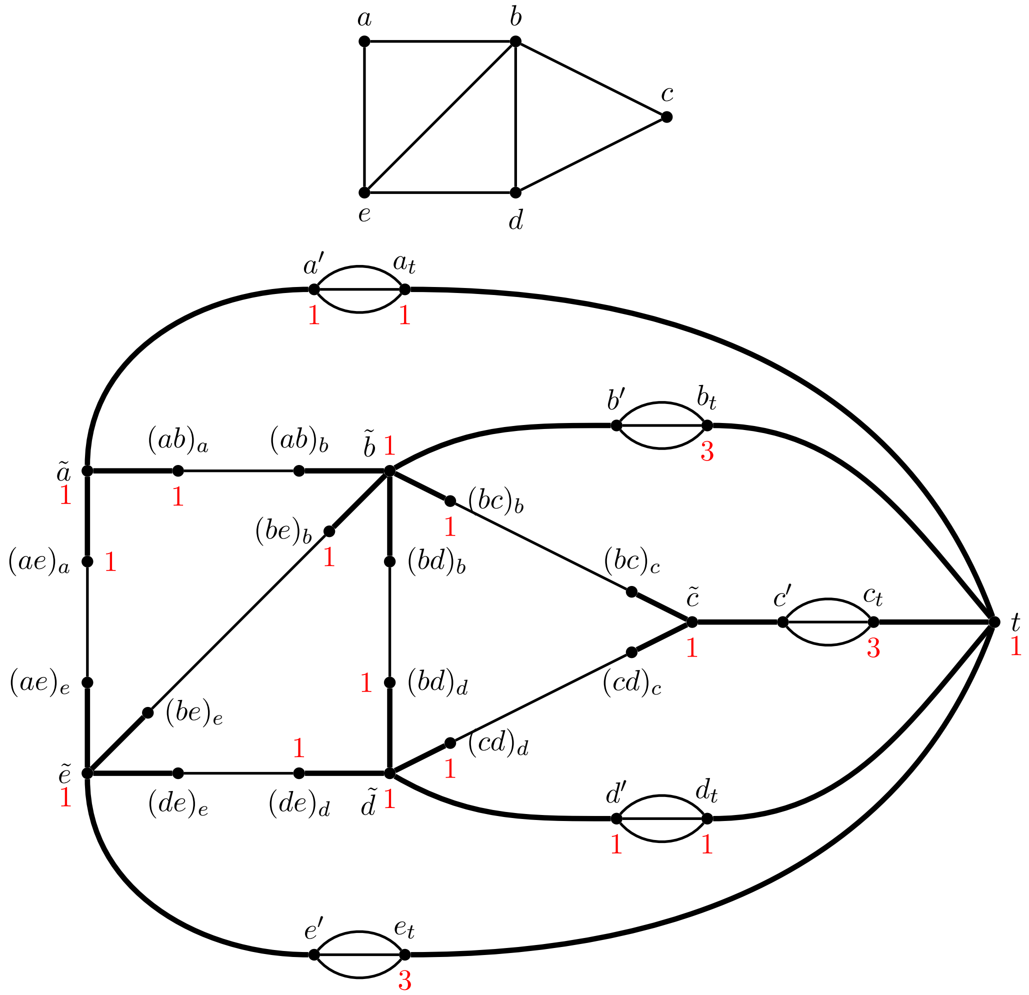



Figure 4.1: A graph $G$ and its reduction graph $H$. The bold edges are $M$ parallel edges, for this graph $M = 15 + 14 + 2 = 31$. The numbers show an effective divisor, as in Example 4.2.2

This divisor is effective and has degree $1 + |V(G)| + |A| + |A| + 3|V\backslash A| + |E(G)| = 1 + 4|V(G)| - \alpha(G) + |E(G)|$. We can show that it has rank at least 1.

The only vertices that have no chips are the vertices $v'$ with $v \notin A$ and the heads of the edges $e_u e_v$. For every vertex $v \in V(G)$ we define $B_v$ as the set $\{\tilde{v}, v'\} \cup \{e_v \mid e = uv \text{ for some } u\}$. Now consider the following firing sets for $l \in \{0, 1, \dots, k\}$:

$$C_l = \{t\} \cup \{v_t \mid v \in V(G)\} \cup \bigcup_{v \in A} B_v \cup \bigcup_{i=1}^{l} B_{u_i}$$

The outgoing edges of this set are:

- the three edges from $v_t$ to $v'$ for $v = u_i$, $i > l$;
- the edge from $e_v$ to $e_{u_i}$ for $v \in A$, $i > l$;

- the edge from $e_{u_i}$ to $e_{u_j}$ for $i \leq l$, $j > l$.

So we fire three chips from $v_t$ to $v'$ for $v = u_i$, $i > l$, which is possible since $D$ has three chips on $v_t$ if $v \notin A$. And we fire chips along edges $e_u e_v$ from tail to head. This is possible too, since we have a chip on every tail. We see that all sets $C_l$ are valid.

After firing $C_l$ we have three chips on $v'$ for $v = u_i$, $i > l$ and a chip on the head of every edge $e_u e_v$ with $u \in A$, $v = u_j$, $j > l$ and a chip on the head of every edge $e_u e_v$ with $u = u_i$, $i \leq l$, $v = u_j$, $j > l$. So for every vertex without a chip in $D$, we can fire one of the sets $C_l$ to obtain a divisor with a chip on that vertex. It follows that $D$ has rank at least 1.

We conclude that $\mathrm{dgon}(H) \leq 1 + 4|V(G)| + |E(G)| - \alpha(G)$, thus $\mathrm{sdgon}(H) \leq 1 + 4|V(G)| + |E(G)| - \alpha(G)$. □

**Example 4.2.2.** Let $G$ be the graph in Figure 4.1, and $H$ its reduction graph. We see that the vertices $\{a, d\}$ are an independent set. Order the vertices that are not in the independent set as follows: $u_1 = b$, $u_2 = c$, $u_3 = e$. Now we look at the divisor $D$ as in the proof of 4.2.1, see Figure 4.1. Suppose that we want to reach a divisor with a chip on $(bc)_c$. We see that we can reach such a divisor by firing the set

$$\begin{aligned} C_1 &= \{t\} \cup \{v_t \mid v \in V(G)\} \cup \bigcup_{v \in A} B_v \cup \bigcup_{i=1}^{1} B_{u_i} \\ &= \{t, a_t, b_t, c_t, d_t, e_t, a', \tilde{a}, (ab)_a, (ae)_a, \\ &\quad d', \tilde{d}, (cd)_d, (bd)_d, (de)_d, b', \tilde{b}, (ab)_b, (be)_b, (bd)_e, (bc)_b\}. \end{aligned}$$

Before proving that $\mathrm{sdgon}(H) \geq 4|V(G)| + |E(G)| + 1 - \alpha(G)$, we prove that if there are enough parallel edges in a graph $G$ with $\mathrm{sdgon}(G) = k$, then we do not have to subdivide these edges to obtain a refinement $G'$ with $\mathrm{dgon}(G') = k$. This allows us to assume that none of the $M$ parallel edges in the reduction graph of a graph $G$ is subdivided.

**Lemma 4.2.3.** *Let $G$ be a graph with $\mathrm{sdgon}(G) = k$. Suppose that there are $l > k$ edges from $u$ to $v$ in $G$. Then there is a refinement $G'$ of $G$, with $\mathrm{dgon}(G') = k$, such that the edges from $u$ to $v$ are not subdivided.*

*Proof.* Let $G'$ be a refinement of $G$ such that $\mathrm{dgon}(G') = k$. Let $D$ be a divisor of rank at least 1 and degree $k$. Suppose that the edge $e$ from $u$ to $v$ is subdivided in $G'$ by the vertices $w_1, \ldots, w_m$. Let $D' \sim D$ be such that $D'(u) + D'(v)$ is maximal. First we show that $\sum_{i=1}^{m} D'(w_i) \leq 1$.

Suppose that there are two chips on the vertices in $\{w_a \mid 1 \leq a \leq m\}$, one on $w_i$ and one on $w_j$, with $i \leq j$. Then we can fire $\{w_h \mid i \leq h \leq j\}$. We can repeat this, until one of those chips reaches $u$ or $v$. This yields a contradiction with the fact that $D(u) + D(v)$ was maximal. So there is at most one chip on the vertices $w_i$: $\sum_{i=1}^{m} D'(w_i) \leq 1$.

Let $G''$ be the refinement of $G$ such that adding the vertices $w_1, \ldots, w_m$ to $G''$ yields $G'$. We claim that the divisor $D'$ on $G''$ has rank 1 too. Let $w$ be a vertex in $G' \backslash \{u, v, w_1, \ldots, w_m\}$. Then there is a divisor $D_w \sim D'$ on $G'$ such that $D_w(w) \geq 1$.

Let $A_0, \ldots, A_r$ be the level set decomposition of the transformation from $D'$ into $D_w$ and $D_0, \ldots, D_{r+1}$ the associated sequence of divisors. Since there are $k$ chips and $l$ disjoint paths from $u$ to $v$, we know that for all $i$ it holds that $u \in A_i$ if and only if $v \in A_i$. Suppose that $w_i \in A_j$ for some $i$ and $j$. And suppose that $u, v \notin A_j$. It follows that there are two chips fired along the edge $e$, one in the direction of $u$ and one in the direction of $v$. Thus there are at least two chips on the vertices $w_i$ in $D_j$. This is only possible if $u$ or $v$ is already fired, so if $u \in A_h$ or $v \in A_h$ for some $h < j$. This yields a contradiction with the fact that $u, v \notin A_j$. We conclude that if $w_i \in A_j$ for some $i$ and $j$, then $u, v \in A_j$. Now we see that firing the sets $A_0 \backslash \{w_1, \ldots, w_m\}, \ldots, A_r \backslash \{w_1, \ldots, w_m\}$ in $G''$ yields a divisor with a chip on $w$. We conclude that $D'$ has rank at least 1 on $G''$ too.

We define $G'''$ as the refinement $G'$ without all vertices that are added to edges from $u$ to $v$. By the same argument it follows that $D'$ is an effective divisor on $G'''$ with rank at least 1 and degree $k$. Thus $G'''$ has the desired properties. □

Now we define an equivalence relation on the vertices of a graph. After this we can show that $\mathrm{sdgon}(H) \geq 4|V(G)| + |E(G)| + 1 - \alpha(G)$ for the reduction graph $H$ of a simple graph $G$.

**Definition 4.2.4.** Let $G$ be a graph and $D$ an effective divisor. We say that two vertices $u$ and $v$ are *$D$-equivalent*, or $u \sim_D v$, if for every effective divisor $D'$, for which it holds that $D \sim D'$, the following holds: let $A_0, \ldots, A_k$ be the level set decomposition of the transformation of $D$ into $D'$, then $u \in A_i$ if and only if $v \in A_i$.

So $u \sim_D v$ if and only if $u$ and $v$ are always fired together. If $u \sim_D v$, then there will never be chips fired along the edges from $u$ to $v$ (if those edges exists). And, if $D$ is a divisor of degree $k$, and there are $l > k$ edges between $u$ and $v$, then $u \sim_D v$.

**Lemma 4.2.5.** *Let $G$ be a simple graph and $H$ its reduction graph. Then* $\mathrm{sdgon}(H) \geq 4|V(G)| + |E(G)| + 1 - \alpha(G)$.

*Proof.* First notice that $\mathrm{sdgon}(H) < M$, since $H$ has only $3|V(G)|+2|E(G)|+1 < M$ vertices. Let $H'$ be a refinement of $H$ with $\mathrm{dgon}(H') = \mathrm{sdgon}(H)$. By Lemma 4.2.3 we know that we can choose $H'$ to be a refinement where all $M$ parallel edges are not subdivided. Let $D$ be a divisor on $H'$ of rank at least 1 and degree $\deg(D) = \mathrm{sdgon}(H)$. We will show that $D$ has degree at least $1 + 4|V(G)| + |E(G)| - \alpha(G)$.

Since we cannot move chips along $M$ parallel edges, we see that the number of chips on $t$ is constant, i.e. for all divisors $D' \sim D$ it holds that $D'(t) = D(t)$. Analogously, we see that for all $v \in V(G)$ and all divisors $D' \sim D$ it holds that $D'(v') = D(v')$. Let $A_v$ be the set of all vertices that are added to the edges from $v'$ to $v_t$, and let $A_e$ be the set of all vertices that are added to the edge from $e_u$ to $e_v$. Then we see that the number of chips on the sets $\{v', v_t\} \cup A_v$ and $\{e_u, e_v\} \cup A_e$

is constant as well. Moreover, we see that

$$\begin{aligned} D(t) &\geq 1 \\ D(\tilde{v}) &\geq 1 \\ D(v') + D(v_t) + \sum_{u \in A_v} D(u) &\geq \begin{cases} 3 & \text{if } v' \sim_D v_t \\ 2 & \text{if } v' \nsim_D v_t \end{cases} \\ D(e_u) + D(e_v) + \sum_{u \in A_e} D(u) &\geq \begin{cases} 2 & \text{if } e_u \sim_D e_v \\ 1 & \text{if } e_u \nsim_D e_v \end{cases} \end{aligned}$$

Since $v' \sim_D \tilde{v}$ and $v_t \sim_D t$, we can replace the condition $v' \sim_D v_t$ by the condition $\tilde{v} \sim_D t$, and since $e_u \sim u$ we can replace $e_u \sim_D e_v$ by $u \sim_D v$.

Consider the equivalence classes of $\sim_D$ on $\{\tilde{v} \mid v \in V(G)\} \cup \{t\}$, write $U_0 \cup \{t\}$, $U_1, \ldots, U_r$ for these classes. Now we see that we have $3|V(G)| - |U_0|$ chips on the vertices $v'$ and $v_t$:

$$\sum_{v \in V(G)} D(v') + D(v_t) + \sum_{u \in A_v} D(u) = 3|V(G)| - |U_0|.$$

And we have at least $|E(G)| + |E(U_0)|$ chips on the vertices $e_u$:

$$\sum_{e=uv \in E(G)} D(e_u) + D(e_v) \geq |E(G)| + |E(U_0)|.$$

Now it follows that

$$\begin{aligned} \deg(D) &\geq 1 + |V(G)| + 3|V(G)| - |U_0| + |E(G)| + |E(U_0)| \\ &\geq 1 + 4|V(G)| + |E(G)| - \alpha(G). \end{aligned}$$

The last inequality holds because $\alpha(G) + |E(U_0)| \geq |U_0|$: if $|E(U_0)| < |U_0|$, then there are at least $|U_0| - |E(U_0)|$ connected components. And we can pick a vertex from every connected component of $U_0$ to obtain an independent set. So then $\alpha(G) + |E(U_0)| \geq |U_0|$. $\square$

**Theorem 4.2.6.** *The stable divisorial gonality problem is NP-hard.*

*Proof.* Let $(G, k)$ be an instance of the independent set problem. We can construct the reduction graph $H$ in polynomial time. By Lemma 4.2.1 and 4.2.5, it follows that $G$ has an independent set of size at least $k$ if and only if $H$ has stable divisorial gonality at most $4|V(G)|+|E(G)|+1-k$. It follows that $(H, 4|V(G)|+|E(G)|+1-k)$ is a 'yes'-instance for the stable divisorial gonality problem if and only if $(G, k)$ is a 'yes'-instance for the independent set problem. So we conclude that the stable divisorial gonality problem is NP-hard. $\square$

# Conclusion

In this thesis, we have studied several notions of gonality. We have seen proofs of some relations between different notions of gonality and we have seen that treewidth is a lower bound for gonality. In Chapter 3, we gave a set of reduction rules to recognize stable hyperelliptic and stable divisorial hyperelliptic graphs. These rules lead to algorithms that recognize these graphs in $O(n \log n + m)$ time. In the last chapter, we have proven that computing stable divisorial gonality is NP-hard.

Many questions remain open. First of all, are there more relations between different notions of gonality? We have seen that in general $\mathrm{sgon}(G) \not\leq \mathrm{dgon}(G)$ and $\mathrm{dgon}(G) \not\leq \mathrm{sgon}(G)$. But it is possible that the stable gonality of a graph can be bounded from above by some function of the divisorial gonality of the graph, i.e., that there is a function $f$ such that $\mathrm{sgon}(G) \leq f(\mathrm{dgon}(G))$.

In Chapter 3 we gave a set of reduction rules to recognize stable hyperelliptic graphs in $O(n \log n + m)$ time. Can stable hyperelliptic graphs be recognized in linear time? Is there an algorithm to recognize graphs of stable gonality 3?

We have seen that computing stable divisorial gonality is NP-hard. It is open whether computing stable divisorial gonality is in NP, so whether it is NP-complete. What about stable and geometric gonality? And are some of the notions fixed parameter tractable? Or, are there problems that are fixed parameter tractable with gonality as parameter, while they are not fixed parameter tractable with treewidth as parameter?

# Bibliography


[1] Jochen Alber, Michael R. Fellows, and Rolf Niedermeier. Polynomial-time data reduction for dominating set. *Journal of the Association for Computing Machinery*, 51(3):363–384, 2004. doi:10.1145/990308.990309.

[2] Omid Amini, Matthew Baker, Erwan Brugallé, and Joseph Rabinoff. Lifting harmonic morphisms II: Tropical curves and metrized complexes. *Algebra & Number Theory*, 9(2):267–315, 2015. doi:10.2140/ant.2015.9.267.

[3] Stefan Arnborg, Bruno Courcelle, Andrzej Proskurowski, and Detlef Seese. An algebraic theory of graph reduction. *Journal of the Association for Computing Machinery*, 40(5):1134–1164, 1993. doi:10.1145/174147.169807.

[4] Stefan Arnborg and Andrzej Proskurowski. Characterization and recognition of partial 3-trees. *SIAM Journal on Algebraic and Discrete Methods*, 7(2):305–314, 1986. doi:10.1137/0607033.

[5] Per Bak, Chao Tang, and Kurt Wiesenfeld. Self-organized criticality. *Physical Review A*, 38:364–374, 1988. doi:10.1103/PhysRevA.38.364.

[6] Matthew Baker. Specialization of linear systems from curves to graphs. *Algebra & Number Theory*, 2(6):613–653, 2008. With an appendix by Brian Conrad. doi:10.2140/ant.2008.2.613.

[7] Matthew Baker and Serguei Norine. Riemann-Roch and Abel-Jacobi theory on a finite graph. *Advances in Mathematics*, 215(2):766–788, 2007. doi:10.1016/j.aim.2007.04.012.

[8] Matthew Baker and Serguei Norine. Harmonic morphisms and hyperelliptic graphs. *International Mathematics Research Notices*, 15:2914–2955, 2009. doi:10.1093/imrn/rnp037.

[9] Matthew Baker and Farbod Shokrieh. Chip-firing games, potential theory on graphs, and spanning trees. *Journal of Combinatorial Theory, Series A*, 120(1):164 – 182, 2013. doi:10.1016/j.jcta.2012.07.011.

[10] Norman L. Biggs. Chip-firing and the critical group of a graph. *Journal of Algebraic Combinatorics*, 9(1):25–45, 1999. doi:10.1023/A:1018611014097.

[11] Anders Björner and László Lovász. Chip-firing games on directed graphs. *Journal of Algebraic Combinatorics*, 1(4):305–328, 1992. doi:10.1023/A:1022467132614.

[12] Anders Björner, László Lovász, and Peter W. Shor. Chip-firing games on graphs. *European Journal of Combinatorics*, 12(4):283 – 291, 1991. doi:10.1016/S0195-6698(13)80111-4.

[13] Jelco M. Bodewes. Divisorial gonality of graphs. Master's thesis, Utrecht University, to appear, 2017.

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

[15] Andries H. Brouwer and Willem H. Haemers. *Spectra of Graphs*. Springer, 2012.

[16] Hannah Cairns. Some halting problems for abelian sandpiles are undecidable in dimension three. preprint, arXiv:1508.00161, 2015.

[17] Lucia Caporaso. Gonality of algebraic curves and graphs. In *Algebraic and Complex Geometry, In Honour of Klaus Hulek's 60th Birthday*, volume 71 of *Springer Proceedings in Mathematics & Statistics*, pages 77–108. Springer, 2014. doi:10.1007/978-3-319-05404-9.

[18] Melody Chan, Darren Glass, Matthew Macauley, David Perkinson, Caryn Werner, and Qiaoyu Yang. Sandpiles, spanning trees, and plane duality. *SIAM Journal on Discrete Mathematics*, 29(1):461–471, 2015. doi:10.1137/140982015.

[19] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT Press, third edition, 2009.

[20] Gunther Cornelissen, Fumiharu Kato, and Janne Kool. A combinatorial Li–Yau inequality and rational points on curves. *Mathematische Annalen*, 361(1-2):211–258, 2015. doi:10.1007/s00208-014-1067-x.

[21] Bruno Courcelle. The monadic second-order logic of graphs. I: Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

[22] Deepak Dhar. Self-organized critical state of sandpile automaton models. *Physical Review Letters*, 64:1613–1616, 1990. doi:10.1103/PhysRevLett.64.1613.

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

[24] Josse van Dobben de Bruyn and Dion Gijswijt. Treewidth is a lower bound on graph gonality. preprint, arXiv:1407.7055v2, 2014.

[25] Richard J. Duffin. Topology of series-parallel networks. *Journal of Mathematical Analysis and Applications*, 10(2):303 – 318, 1965. doi:10.1016/0022-247X(65)90125-3.

[26] Dion Gijswijt. Computing divisorial gonality is hard. preprint, arXiv:1504.06713, 2015.

[27] Kevin Hendrey. Sparse graphs of high gonality. preprint, arXiv:1606.06412, 2016.

[28] Cristopher Moore and Martin Nilsson. The computational complexity of sandpiles. *Journal of Statistical Physics*, 96(1):205–224, 1999. doi:10.1023/A:1004524500416.

[29] Guido Schäfer. Lecture notes of the master course: Discrete Optimization. MasterMath, fall 2015.

[30] Paul D. Seymour and Robin Thomas. Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory, Series B*, 58(1):22 – 33, 1993. doi:10.1006/jctb.1993.1027.

[31] Hajime Urakawa. A discrete analogue of the harmonic morphism and green kernel comparison theorems. *Glasgow Mathematical Journal*, 42(3):319–334, 2000.