Summer 2019

Taking a Canon to the Adjunction Formula

Paul M. Harrelson

TAKING A CANON TO THE ADJUNCTION FORMULA

by

PAUL HARRELSON

(Under the Direction of Jimmy Dillies)

ABSTRACT

In this paper, we show how the canonical divisor of a graph is related to the canonical divisor of its subgraph. The use of chip firing and the adjunction formula for graphs explains said relation and even completes it. We go on to show the difference between the formula for full subgraphs and that of non-full subgraphs. Examples are used to simplify these results and to see the adjunction formula in action. Finally, we show that though the adjunction formula seems simple at first glance, it is somewhat complex and rather useful.

INDEX WORDS: Graph theory, Algebraic geomety, Adjunction

2009 Mathematics Subject Classification: 15A15, 41A10

TAKING A CANON TO THE ADJUNCTION FORMULA

by

PAUL HARRELSON

Bachelor of Science, Middle Georgia State University, 2017

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial

Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

TAKING A CANON TO THE ADJUNCTION FORMULA

by

PAUL HARRELSON

Major Professor: Jimmy Dillies
Committee: Enkeleida Lakuriqi
Hua Wang

Electronic Version Approved:
July 2019

DEDICATION

This thesis is dedicated to my future wife and children whom I will feed and clothe with the money I make from the job I get with the help of the degree that I will receive once this thesis is accepted by this committee.

ACKNOWLEDGMENTS

I wish to acknowledge my fiancée Rachel Vogt, without whose help I could not have finished this. Thank you for the constant encouragement and for letting me bounce things off to you.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In this paper, we will show how the canonical divisor of a graph is related to the canonical divisor of its subgraph. We will use chip firing and the adjunction formula for graphs to explain said relation. We will even show the difference between the formula for full subgraphs and that of non-full subgraphs. We will use examples to simplify these results and to see the adjunction formula in action. Finally, we will show that though the adjunction formula seems simple at first glance, it can be complex and rather useful.

1.1 PRELIMINARIES

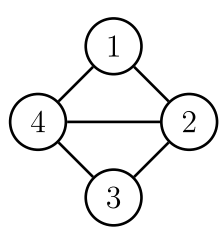
1.1.1 GRAPHS

Before getting into chip configuration, let us begin with some definitions. We will use the terminology of Diestel [2].

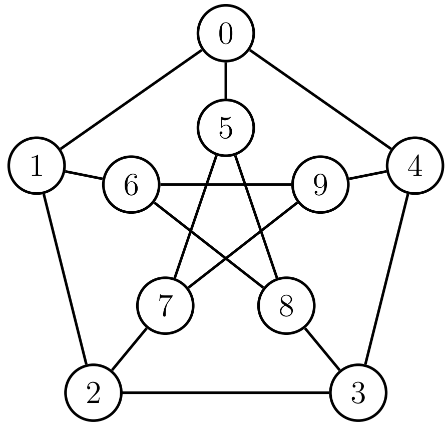
Definition 1. A graph G is a pair of sets (V, E) such that $E \subset V^2$. The elements of V are called vertices and those of E, edges.

Visually, one can represent a graph as a collection of 'dots' connected by 'segments'.

Example 1.1. *The graph* $G = (\{1, 2, 3, 4\}, \{\{1, 2\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\})$ *can be represented as*



Example 1.2. *The Peterson Graph*

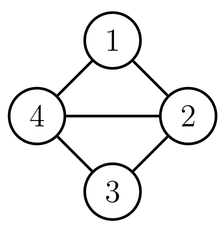


Definition 2. When an edge connects two vertices, these are said to be adjacent to one another.

Definition 3. The degree of a vertex is the number of edges it has, or the number of edges connected to it.

Definition 4. The canonical divisor of a graph, G, is just the sum of the degree minus two of each vertex and is denoted $K_G = \sum_{v \in V}(deg(v) - 2)(v)$ as will be shown in the example below.

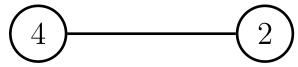
Example 1.3. *We will use graph G from Example 2.1 and we will continue using that graph throughout the rest of this chapter.*



The canonical divisor of G is:

$$\begin{aligned} K_G &= (2-2)[1] + (3-2)[2] + (2-2)[3] + (3-2)[4] \\ &= 0[1] + 1[2] + 0[3] + 1[4] \end{aligned} \tag{1.1}$$

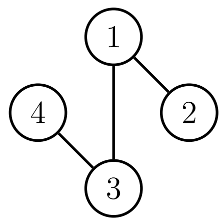
So vertex 1, $[1]$*, has 2 adjacencies but 2 minus 2 is 0, then the other vertices are denoted similarly. Zeroes are not formally used, but they are necessary. For instance, let us compare two graphs. First* G_1*:*



whose canonical divisor is:

$$\begin{aligned} K_{G_1} &= (1-2)[2] + (1-2)[4] \\ &= (-1)[2] - 1[4] \end{aligned} \tag{1.2}$$

Next, G_2*:*



whose canonical divisor is $K_{G_2} = (2-2)[1] + (1-2)[2] + (2-2)[3] + (1-2)[4]$ *which would equal* $(-1)[2] - 1[4]$ *if we did not include the zeroes. This would make the canonical divisors of the two graphs the same. This is just one simple example but it should show the importance of using the zeroes as placeholders.*

Definition 5. The genus of a graph is denoted $g = |E| - |V| + 1$

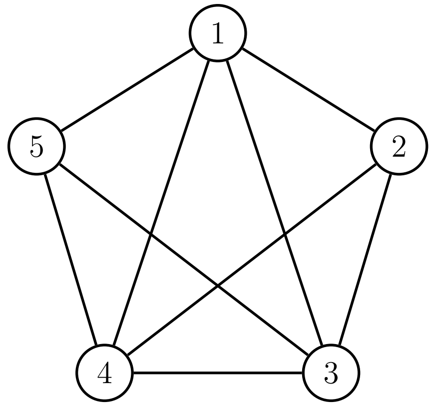
Example 1.4. *For the graph from Example 2.1, the genus is,*

$$\begin{aligned} g_G &= 5 - 4 + 1 \\ &= 2 \end{aligned} \tag{1.3}$$

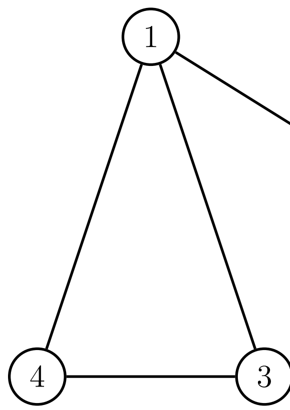
We know that a subgraph is supposed to be similar to a subset for graphs but we must definitively express what a subgraph is.

Definition 6. A subgraph X of a graph Y, is a graph such that any vertex and any edge in X is also in Y.

Example 1.5. *A graph Y*

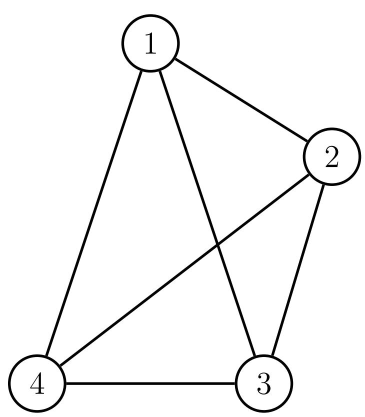


Has a subgraph X



Definition 7. An induced subgraph is a subgraph that contains every edge from the original graph between the vertices that are also in the subgraph.

Example 1.6. *Using that same Y, we can make an induced graph X with the vertices 1, 2, 3, and 4.*



Distinguishing induced and non-induced subgraphs allows us to even more clearly define the difference between graphs ad their subgraphs.

Definition 8. For any subgraph X of Y, vertices of the subgraph that were adjacent to vertices that are only in Y are called outervertices, while the other vertices of the subgraph are called innervertices.

Now we can play with this idea of induced subgraphs and non-induced subgraphs even further, and we can even "fill" non-induced subgraphs with what is called the closure of the graph.

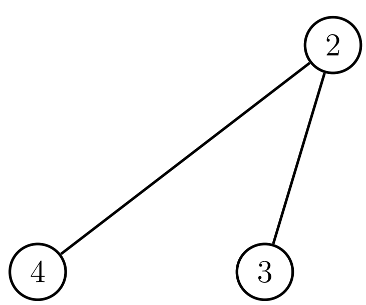
Definition 9. The closure of a subgraph, X, of Y, denoted $\bar{X}^Y \subset Y$, is the smallest induced subgraph of Y containing X. Furthermore, if X is already an induced subgraph of Y, Then $\bar{X}^Y = X$.

Remark 1.7. *So the graph in Example 3.2 is the closure of the one in Example 3.1.*

This makes things rather interesting, but what exactly is the difference of an induced subgraph and a non-induced subraph? Articulating that difference should help us with the difference between graphs and their subgraphs since non-induced subgraphs are actually subgraphs of induced subgraphs.

Definition 10. The boundary of a subgraph, X, on Y, denoted $\delta_Y X$, is the set of edges, with their attatched vertices, in $\bar{X}^Y$ that are not in X

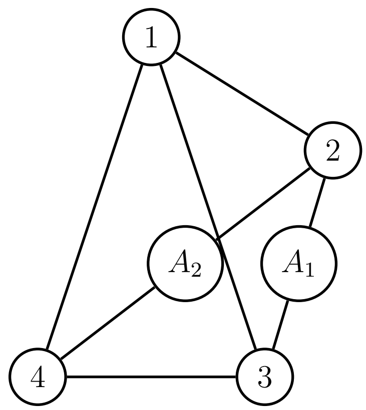
Example 1.8. *So the boundary of Example 4.1 would be*



So now we can pick apart these graphs and really work with everything underneath. All these differences between induced and non-induced subgraphs are a good starting point to figuring out the relation between graphs and their subgraphs.

Definition 11. Let X be a subgraph of Y. Let $Aug(\bar{X}^Y)$ be the augmentation of the closure of X on Y where the augmentation subdivides each edge in $\bar{X}^Y$ that is also in $\delta_Y X$.

Example 1.9. *Using the subgraph X of Y from Example 4.1,* $Aug(\bar{X}^Y)$ *is*



1.1.2 Matrices

Next we will look at a graph through its matrix representation and the matrix representation of chip configurations. We will start by defining the building blocks of all matrices. Since the values of the matrices are identified by their rows and columns, it is easy to construct bigger matrices with smaller matrices.

Definition 12. Elementary matrices are matrices that have only one non-zero value of “1”. They are denoted $e_{m,n}$ where “m” is the row and “n” is the column where the only non-zero value is.

Example 1.10.

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, e_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, e_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

And so on.

The elementary matrices in this example could have had more rows and columns, they were just shown with four rows and one column to show that they can be restricted. Normally they are restricted to the size that they are needed. For instance, if we wanted to construct a matrix with four columns and three rows, then that is how we would show the elementary matrices. The only problem is that the number of elementary matrices needed grows rather quickly as the constructed matrix grows. In that instance with a matrix of four columns and three rows, twelve different elementary matrices are needed since each one would have "1" in each of the different positions with zeroes everywhere else.

Definition 13. A square matrix is a matrix with the same number of rows and columns.

Pretty simple since they look square-ish but necessary since only square matrices are used to represent graphs. Now we also need to define diagonal matrices which are a special type of square matrix

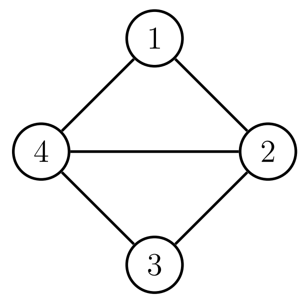
Definition 14. A diagonal matrix is a square matrix with all non-diagonal entries being zero.

To be clear, in a diagonal matrix, the diagonal entries can also be zero, it is only necessary that the non-diagonal entries are all zeroes.

Now, the reason square matrices are used to represent graphs is that the number of rows and columns correspond to the number of vertices in the graph, and the values are determined by how the vertices relate to one another. First there is the degree matrix.

Definition 15. The degree matrix of a graph G, denoted D_G is a matrix used to represent the degrees of each of the vertices of the graph. It is a diagonal matrix whose diagonal entries are equal to the degree of the corresponding vertex. For instance, the entry in row 1 column 1 is the degree of v_1, while the entry in row 2 column 2 is the degree of v_2.

Example 1.11. *We will use graph G from Example 2.1*

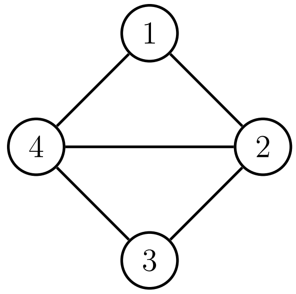


$$MatrixD_G = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

Next is the adjacency matrix which tells us what vertices are connected to each other.

Definition 16. The adjacency matrix of a graph G, denoted A_G is a matrix whose entries are all either “0” or “1”. The entry $(A_G)_{m,n} = 1$ if the vertices (m) and (n) are adjacent and “0” if they are not.

Example 1.12. *We will again use graph G from Example 2.1*



$$A_G = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

We use the degree matrix and the adjacency matrix of a graph to make up the graph Laplacian which is what is used in the matrix representation of chip firing.

Definition 17. The Graph Laplacian is the matrix that denotes as $Q_G = D_G - A_G$

Example 1.13. *For our graph G from Example 2.1, we use the diagonal matrix and the adjacency matrix we found in Example 2.11 and Example 2.12 respecively.*

$$Q_G = D_G - A_G = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix}$$

As stated in the definition, the columns of the graph laplacian are used to represent how each vertex fires chips. When v_1 in graph G from Example 2.1 fires its chips, it loses two chips and sends one chip to v_2, one chip to v_4, and zero chips to v_3. Column from the graph laplacian has "2" in it's first entry, "-1" in it's second entry, "0" in it's third entry, and "-1" in it's fourth and final entry, which corresponds exactly with firing chips from v_1. The only difference is that the values are actually reflected about 0 but why becomes clear very shortly.

CHAPTER 2

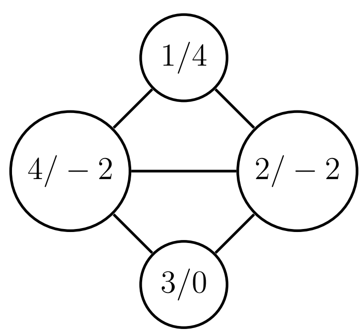
CHIP FIRING

In this chapter we discuss chip configurations and firing, and the many different ways it can be represented. Chip configurations are possibly easiest to grasp in context with graphs in graph theory, so we should start there. Many of these definitions, as well as the description of chip firing, come from Scott Corry's book "Divisors and Sandpiles" [1].

2.1 Graph Representation

Definition 18. A chip configuration is an amount of "chips," a weight, attributed to each vertex on the graph. Chips are currency and a chip configuration is assigning each vertex an amount of currency.

Example 2.1. *On the graph from Example 1.1 we will be adding a chip configuration. It will be denoted "vertex #/weight".*

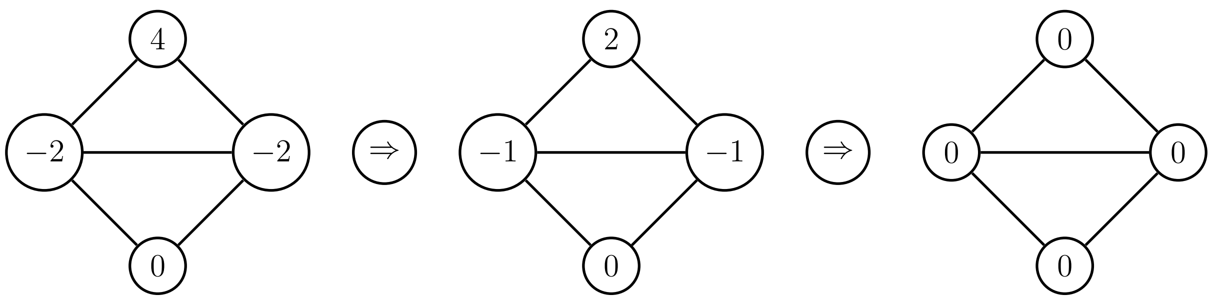


The algebraic representation of the chip configuration of G is:

$$G = 4[1] + (-2)[2] + 0[3] + (-2)[4].$$

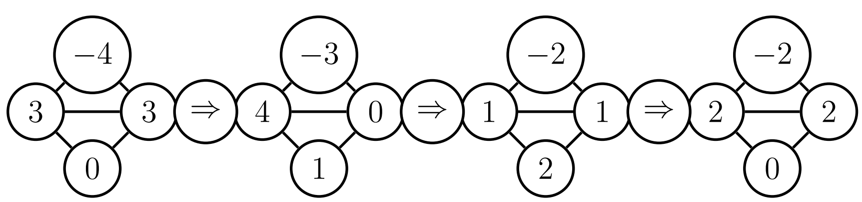
Chip configurations are used to play a game known as chip firing where a vertex fires chips according to its degree. A vertex fires its chips by passing along one chip to each other vertex it is adjacent to. A vertex may only fire once at a time. So if a vertex has four chips for example and a degree of three, when it is fired from, it will have only one chip left, but each of the vertices adjacent to it will have gained one chip. These are the rules to the game of chip firing, but the objective is to have non-negative weights on all of the vertices in the graph. In other words we do not want any vertices to be in debt. To be clear, you may fire from a vertex even if it will mean having a negative amount of chips attributed to it afterwards. Also, it is necessary to choose a vertex which will stay constant. That is, you must choose one vertex that will never be fired from. This is because if all vertices are fired from once exactly, then the graph returns to the same exact chip configuration it started with.

Example 2.2. *Here is an example of the chip firing game, using the chip configuration from Example 1.3. The vertices will fire until all have non-negative chip values, as the game would normally be played.*

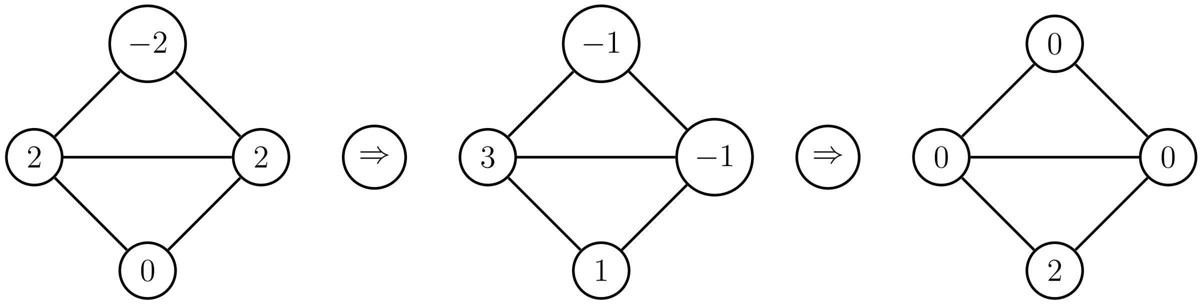


In this play through, v_1 was fired from twice to obtain a non-negative chip configuration.

Example 2.3. *In this example, a different chip configuration is used to better illustrate the rules:*



First, v_2 was fired from, then v_4, and finally from v_3. From here, we cannot fire from any vertex without leaving said vertex with a negative amount of chips, however, as stated in the rules, this does not stop us from being able to fire.



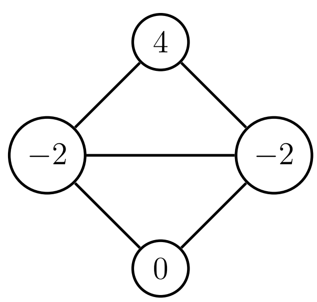
We next fire from v_2 once more, and finally we fire again from v_4 resulting in the non-negative chip configuration.

So we can already see hints as to how we can use the canonical divisor to show how the vertices and edges of the graph relate to eachother in how it can be used to help understand more about this chip firing game. However, that picture might not be quite clear, which is why we must look at this in another light.

2.2 Matrix Representation

Definition 19. The Chip Configuration Matrix is exactly that, it is the matrix representation of a graph's chip configuration. For a chip configuration G, it is denoted $\vec{x}_G$

Example 2.4. *We will use the chip configuration from Example 2.1*



$$\begin{pmatrix} 4 \\ -2 \\ 0 \\ -2 \end{pmatrix}$$

With the chip configuration matrix, we are now ready to fire chips using matrix representation.

Chip firing is represented by matrices with this equation: $\vec{x}_G - Q_G * e_n$ where "n" is the numbered vertex you want to fire from and e_n is an elementary matrix like the ones shown in Example 1.8. Notice that we are subtracting the graph laplacian, reflecting its values about 0 to how they need to be. It makes sense that when firing chips, we would subtract, which is why we made the graph laplacian as we did.

Example 2.5. *Continuing from Example 1.14, we will be firing from v_1 twice as in Example 1.5*

$$\begin{pmatrix} 4 \\ -2 \\ 0 \\ -2 \end{pmatrix} - \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 4 \\ -2 \\ 0 \\ -2 \end{pmatrix} - \begin{pmatrix} 2 \\ -1 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \\ 0 \\ -1 \end{pmatrix}$$

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

This matches the graphs from Example 2.2

Now we are ready to move on to our third and final field.

2.3 Algebraic Representation

Algebraic representation is the most useful one used. It allows us to put everything else we know from the other mediums all together and figure out how chip firing works and how it relates graphs with their subgraphs.

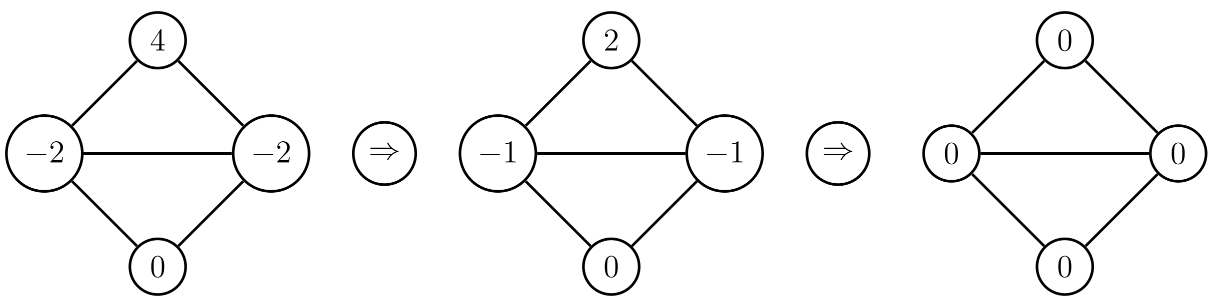
Remark 2.6. *Remember definition 4 of the canonical divisor and Example 2.4 when we showed the algebraic representation of the chip configuration.*

As was stated earlier in this chapter, the goal of the chip firing game is to make all the vertex values are non-negative. Next we will define the rank of the graph's chip configuration which tells us whether a chip configuration is winnable and how close it is to not being winnable.

Definition 20. The Rank of a graph's chip configuration is one less than the minimum number of chips needed to be removed so that there is no longer a winning strategy. It is denoted as $r(G)$ for any graph G.

To better understand this, let's use an example.

Example 2.7. *Let's use the chip configuration from example 2.1*



$r(G) = 0$

In this example, the chip configuration ends in all zeroes. In order to make this no longer winnable, we would just have to take one chip from any of the vertices. Since we just have to take one vertex and since the rank is defined as one less than the number of chips that have to be removed, then one less than one is zero, thus the rank is zero.

The degree of a graph, which we will define next, is not related to the degrees of the vertices of a graph.

Definition 21. The degree of a graph is merely the the sum of the chips of all of the vertices. It is denoted as $deg(G)$ for any graph G.

Example 2.8. *We will again use the chip configuration from Example 2.1*

$$deg(G) = 4 + (-2) + 0 + (-2) = 0$$

Theorem 2.9. $r(G) - r(K_G - G) = deg(G) - g_G + 1$ *This is known as the the Riemann-Roch Theorem for graphs.*

Example 2.10. *We will again use the graph and chip configuration from Example 2.4 whose canonical divisor we found in Example 2.3, algebraic representation of it's chip configuration in Example 2.1, rank we found in Example 1.14, genus we found in Example 1.15, and degree in Example 1.16*

$$r(G) - r(K_G - G) = deg(G) - g_G + 1$$

$$0 - r(K_G - G) = 0 - 2 + 1$$

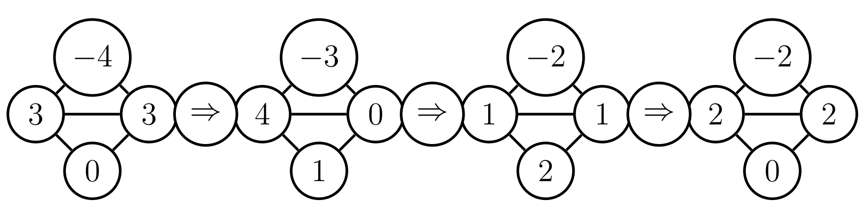
$$r(K_G - G) = 1$$

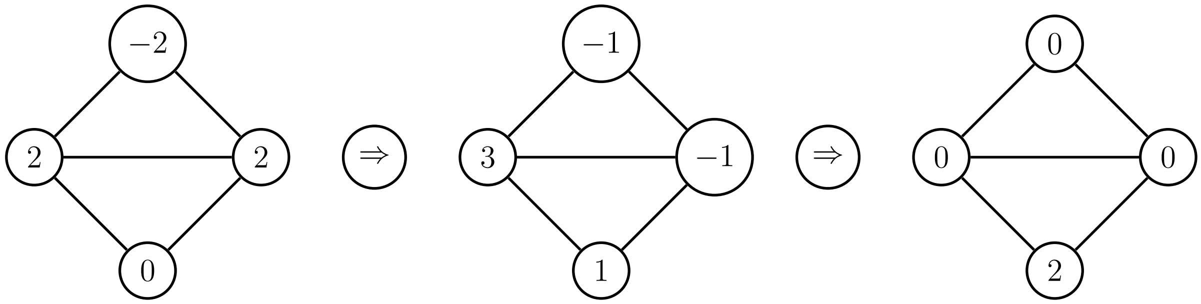
In the above example, we found that the rank of the chip configuration of $K_G - G$ is positive 1. This means that it should be a winnable chip configuration and that all it takes is to take two chips from it to make it no longer winnable.

We can find the chip configuration pretty easily.

$$\begin{aligned} K_G - G &= 0[1] + 1[2] + 0[3] + 1[4] - (4[1] + (-2)[2] + 0[3] + (-2)[4]) \\ &= (-4)[1] + 3[2] + 0[3] + 3[4] \end{aligned} \tag{2.1}$$

Now that we have the chip configuration, let's play the chip firing game to check if it is winnable.





Sure enough, we were able to get it down to all non-negative values for each of the vertices. But at first it looks like we would have to take three chips from it to make sure that there are no longer any winning strategies, but we do indeed have to take only two. If we take 2 chips from either v_2 or v_4, the resulting chip configuration no longer has any winning strategies.

And with this theorem, we see the beginning of the correlation between chip firing and the relationship between graphs and their subgraphs. This allows us to move on to the adjunction formula and find that relation between graphs and their subgraphs.

CHAPTER 3

ADJUNCTION FOR GRAPHS

The adjunction for graphs is a formula meant to show the relation between the canonical divisors of graphs and their subgraphs. To find this relation, we must build off what we know from chapters 1 and 2, but we must also continue introducing a few more definitions.

Remark 3.1. *Recall the definition of the canonical divisors of a graph from section 1.1, of the elementary matrices from section 1.2, and the Laplacian of a graph.*

We recall these defintions, because it is the canonical divisors which will give us the adjunction;
Letting X be a subraph of Y, $K_X = K_Y + \beta|_X$.

Definition 22. This $|_X$ just means we are compressing this equation to vertices on X since Y might include vertices in its canonical divisor that X does not. Thus, for any chip configuration, C, $|_C$ denotes the compression of an equation to only those vertices found also in C.

Returning to the adjuntion, "Letting X be a subraph of Y $K_X = K_Y + \beta|_X$" I will split this into two cases to better show what β is.

Case 1. *X is a non-induced subgraph of Y. Then* $K_X = K_{\bar{X}^Y} + \beta_1|_X$

Proof. Suppose that $\bar{X}^Y$ is just a graph of two vertices with a single adjoining edge between them, and that X is just two vertices without an edge. Let these vertices be called (1) and (2). Then $K_{\bar{X}^Y} = -1(1) - 1(2)$ and $K_X = -2(1) - 2(2)$. If we fire from one vertex, that chip goes to the other, thus one will always affect the other. However, if there was another vertex in-between the existing vertices to intercept the fired chips, firing from one of the two from the original graph will have no effect on the other. Thus, we use the augmentation graph. Let the chip configuration of $Aug(\bar{X}^Y)$ be all zeroes. Then, if we fire from the vertices that are also in X, we get

$$-\sum_{v \in X} Q_{Aug(\bar{X}^Y)} e_v[v] = (-1)[v] = (-1)[1] + (-1)[2].$$

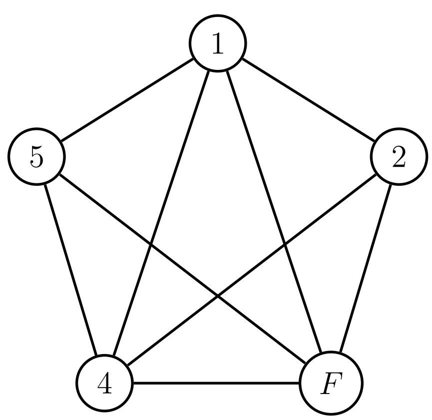
This looks almost familiar.

$$\begin{aligned} K_X - K_{\bar{X}^Y} &= (-2)[1] + (-2)[2] - ((-1)[1] + (-1)[2]) \\ &= (-1)[1] + (-1)[2]. \end{aligned} \tag{3.1}$$

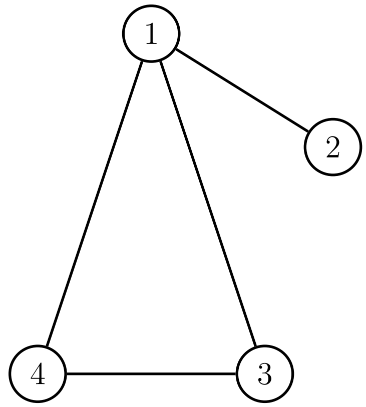
So this $-\sum_{v \in X} Q_{Aug(\bar{X}^Y)} e_v[v]$ is the β_1 that we are looking for in this example. The simplicity of this example allows us to expand it for all non-induced subgraphs X and their closures. As stated before, firing from all vertices in a chip configuration once returns the chip configuration to its original state. This means, that for the augmentation graph, firing from all of the vertices from the original graph will result in no difference of the amounts of chips for the vertices not adjacent to the added vertices in the augmentation graph, but the amounts of the chips of those vertices adjacent to the added vertices in the augmentation graph will be $-a_v$ for all $v \in V$ where $a_v = deg_{Aug(\bar{X}^Y)}[v] - deg_X[v]$.

Thus $\beta_1 = -\sum_{v \in X} Q_{Aug(\bar{X}^Y)} e_v[v]$ □

Example 3.2. *We will use the graphs from examples 1.5, 1.6, and 1.9 as an example of this case. As a reminder, here is our graph Y*

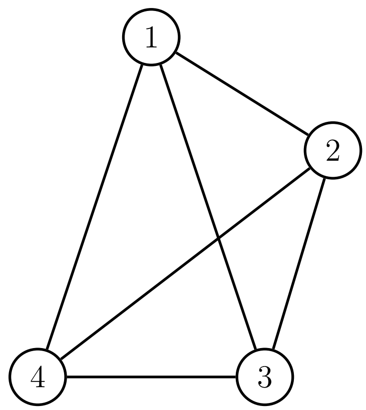


our subgraph X



Note that $K_X = 3[1] + 1[2] + 2[3] + 2[4]$

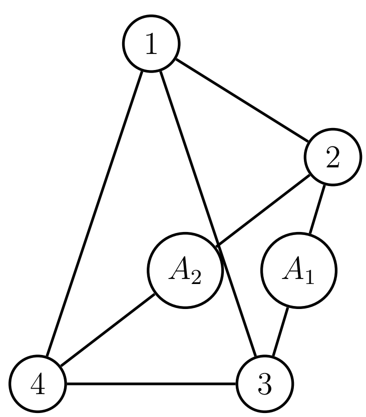
our closure graph $\bar{X}^Y$



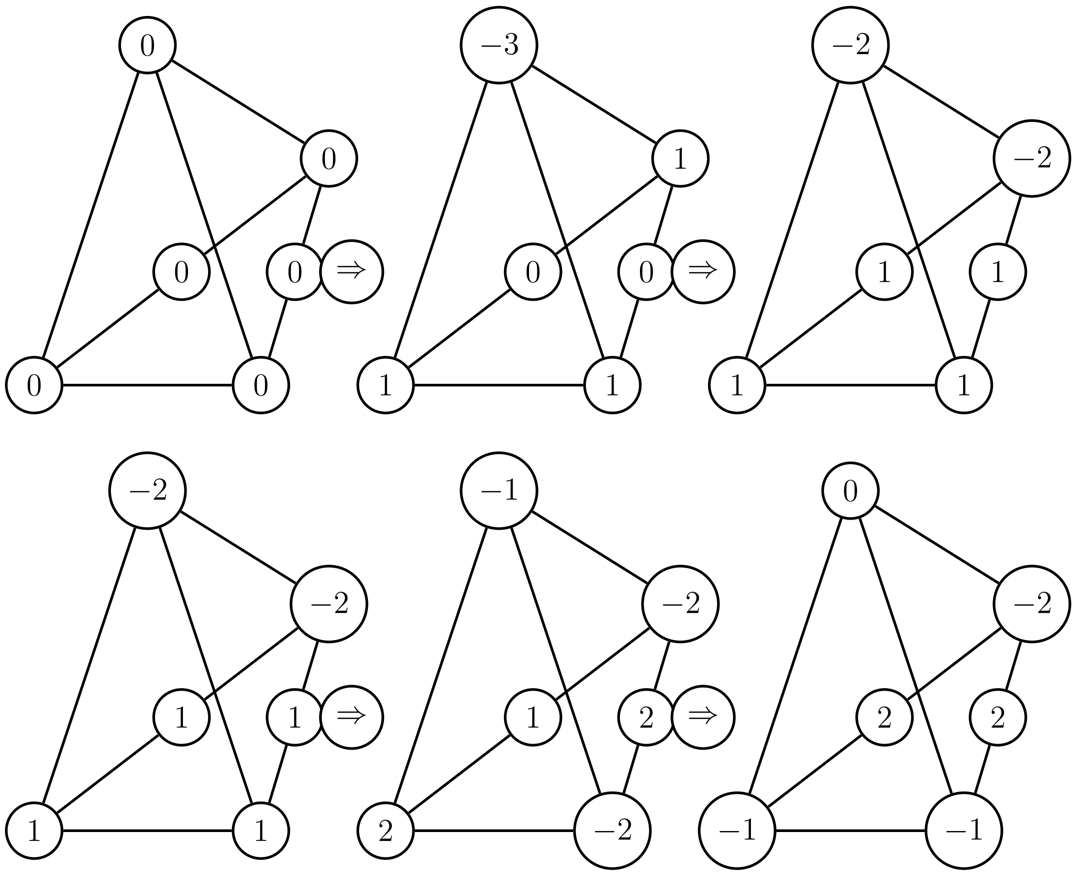
Note that $K_{\bar{X}^Y} = 3[1] + 3[2] + 3[3] + 3[4]$. *So*

$$\begin{aligned} K_X - K_{\bar{X}^Y} &= 3[1] + 1[2] + 2[3] + 2[4] - (3[1] + 3[2] + 3[3] + 3[4]) \\ &= 0[1] + (-2)[2] + (-1)[4] + (-1)[5] \end{aligned} \tag{3.2}$$

and our augmented graph $Aug(\bar{X}^Y)$



Next we give the augmented graph a chip configuration of all zeroes and fire from each of the vertices from the original graph X



The algebraic representation of the final chip configuration is

$$0[1]+(-2)[2]+(-1)[3]+(-1)[4]+2[A_1]+2[A_2] = 0[1]+(-2)[2]+(-1)[3]+(-2)[4]|_X$$

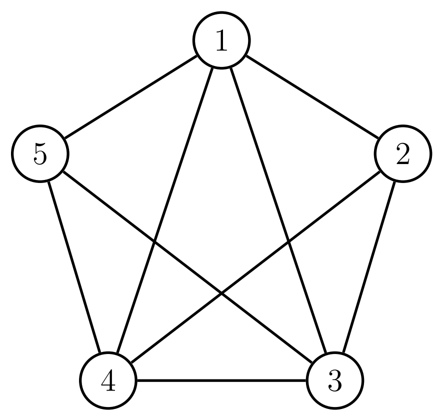
which is what we are looking for for β_1*! So*

$$\begin{aligned} K_X &= 3[1]+1[2]+2[3]+2[4] \\ &= 3[1]+3[2]+3[3]+3[4]+(0[1]+(-2)[2]+(-1)[3]+(-1)[4]) \\ &= K_{\bar{X}^Y} + -\sum_{v\in X} Q_{Aug(\bar{X}^Y)} e_v[v]|_X \end{aligned} \tag{3.3}$$

Case 2. *X is an induced subgraph of Y. Find* β_2 *such that* $K_X = K_Y + \beta_2|_X$

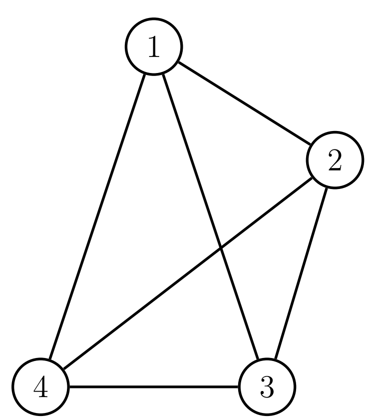
Proof. As previously stated for any graph, chip firing once from every vertex results in the chip configuration it started with. Then it is reasonable to say that firing from some collection of the vertices is a direct inverse of firing from the rest of the vertices. That is to say, if we choose one vertex not to fire from and fire the rest, firing from that last vertex would result in a reset of the chip configuration, thus it is the inverse chip firing. Also, firing once from all of a subgraph's vertces will result in a chip configuration such that the innervertices have the same value as they started with and that the outervertices have only reduced by the number of adjacencies to the rest of the graph and not the inner vertices. Thus, $\beta_2 = -\sum_{v \in X} Q_Y e_v[v]$. □

Example 3.3. *We will use the graphs from examples 1.5, and 1.6 as examples of this case. As a reminder, Here is our graph Y*



Note that $K_Y = 4[1] + 3[2] + 4[3] + 4[4] + 3[5]$

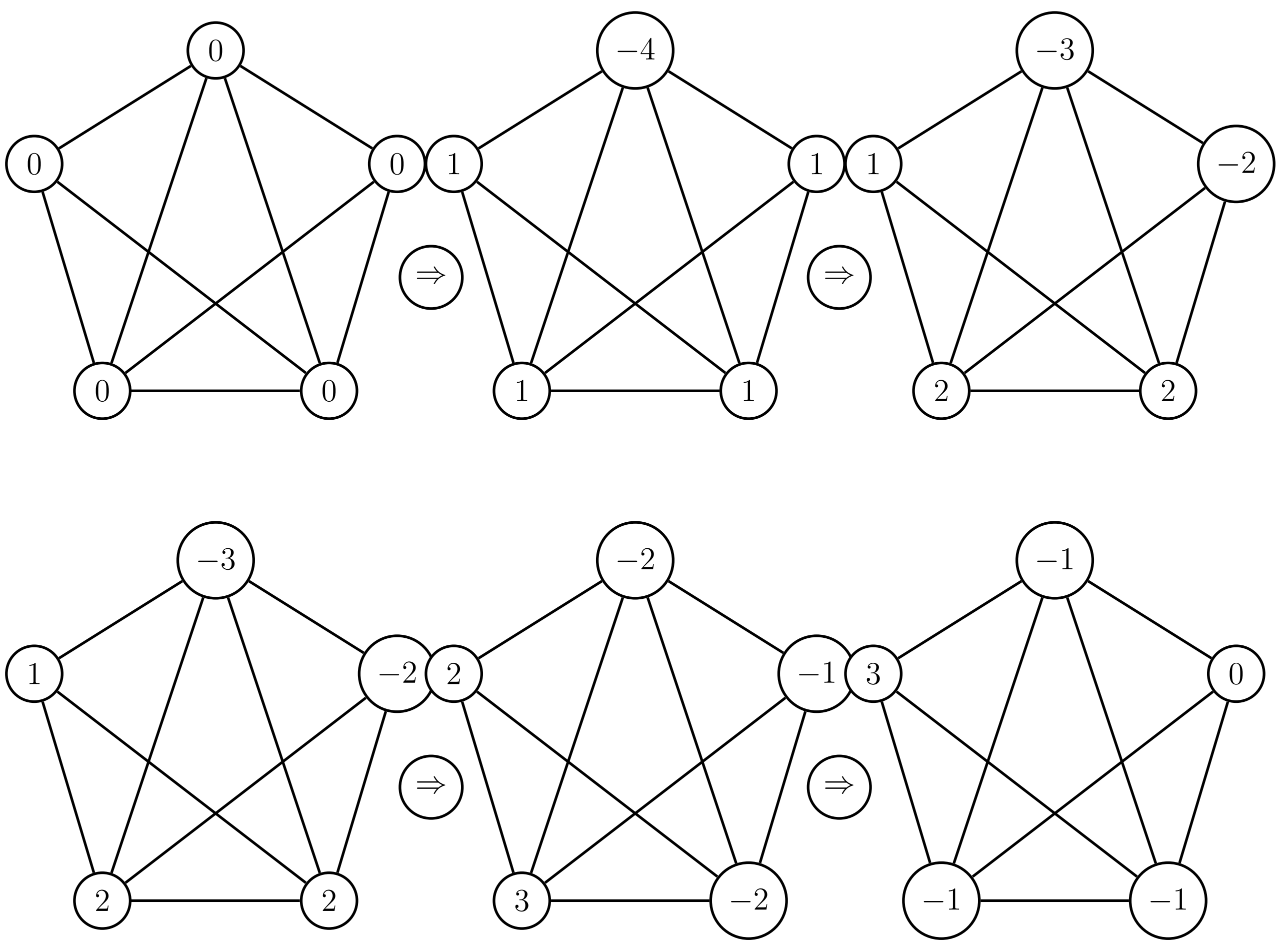
and our induced subgraph $\bar{X}^Y$



Note that $K_{\bar{X}Y} = 3[1] + 3[2] + 3[3] + 3[4]$ *so*

$$\begin{aligned} K_{\bar{X}Y} - K_Y|_X &= 3[1] + [2] + 3[3] + 3[4] - (4[1] + 3[2] + 4[3] + 4[4] + 3[5])|_X \\ &= 3[1] + 3[2] + 3[3] + 3[4] - (4[1] + 3[2] + 4[3] + 4[4]) \\ &= (-1)[1] + 0[2] + (-1)[3] + (-1)[4] \end{aligned} \tag{3.4}$$

Then we assign a zero chip configration on Y and fire from the vertices that are also in X or vertices 1, 2, 3, and 4



The algebraic representation of this chip configuration is

$$(-1)[1] + 0[2] + -1[3] + -1[4] + 3[5]$$

which looks a bit familiar.

$$(-1)[1] + 0[2] + (-1)[3] + (-1)[4] + 3[5]|_X = (-1)[1] + 0[2] + (-1)[3] + (-1)[4]$$

which is exactly what we are looking for yet again for our β_2*! So*

$$\begin{aligned} K_{\bar{X}^Y} &= 3[1] + 3[2] + 3[3] + 3[4] \\ &= 4[1] + 3[2] + 4[3] + 4[4] + 3[5] + ((-1)[1] + 0[2] + (-1)[3] + (-1)[4] + 3[5])|_X \\ &= K_Y + -\sum_{v \in X} Q_Y e_v[v] \end{aligned} \tag{3.5}$$

So returning now to the original adjunction equation we can put these two cases together to find our missing β.

Let the chip configuration of Y and of $Aug(\bar{X}^Y)$ be zero.

Then $\beta = \beta_1 + \beta_2$

So

$$\beta = -\sum_{v \in X} Q_Y e_v[v] - \sum_{v \in X} Q_{Aug(\bar{X}^Y)} e_v[v]$$

Thus

$$K_X = K_Y - \sum_{v \in X} Q_Y e_v[v] - \sum_{v \in X} Q_{Aug(\bar{X}^Y)} e_v[v]|_X$$

Example 3.4. *To use examples 1.11 and 1.12*

$$\begin{aligned} K_X &= 3[1] + 1[2] + 2[3] + 2[4] \\ &= 4[1] + 3[2] + 4[3] + 4[4] + 3[5] + ((-1)[1] + 0[2] + (-1)[3] + (-1)[4] + 3[5]) \\ &\qquad + (0)[1] + (-2)[2] + (-1)[3] + (-1)[4] + 2[A_1] + 2[A_2])|_X \\ &= K_Y - \sum_{v \in X} Q_Y e_v[v] - \sum_{v \in X} Q_{Aug(\bar{X}^Y)} e_v[v]|_X \end{aligned} \tag{3.6}$$

exactly as we proved above.

CHAPTER 4

CONCLUSION

In this paper, we showed that the canonical divisor of a graph is related to the canonical divisor of its subgraph. We explained that the adjunction formula for graphs can show this when chip firing is used along side of it. We even showed that there was a difference from the formula for full subgraphs versus that of non-full subgraphs. Examples allowed us to simplify these results and to see the adjunction formula in action. The adjunction formula for graphs though seemingly simple when we began turned out to be somewhat complex and quite useful for graphs.

REFERENCES

[1] Scott Corry and David Perkinson. *Divisors and Sandpiles: An Introduction to Chip Firing*. American Mathematical Society, 2018.

[2] Reinhard Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer, 2 edition, 2000.

[3] Miles Reid. Chapters on algebraic surfaces. In *Complex algebraic geometry (Park City, UT, 1993)*, volume 3 of *IAS/Park City Math. Ser.*, pages 3–159. Amer. Math. Soc., Providence, RI, 1997.

APPENDIX A

SOME GEOMETRY

In the previous chapters we made extensive use of canonical divisors and adjunction. These concepts seem to appear like a *deus ex machina* and their definition conceals their geometric origin. In this appendix, we will try to give a brief survey of the origin and importance of these two concepts. A good introduction can be found in Reid's *Chapters on Algebraic Surfaces* [3].

A.1 Canonical Class

One of the objectives when attempting to classify geometric objects is to find invariants that allow to discern between different spaces. For a smooth complex variety of dimension n, one can start by taking the tangent or cotangent bundle. While these bundles are invariants, they can sometimes be more complex than the original space. In order to find an invariant which is computed more readily, one can look at the n-th exterior power of the cotangent bundle which is a line bundle called the canonical bundle. In local coordinates, $z_1, \ldots, z_n$, a section of this bundle is an expression of the form

$$\sigma = f(z_1, \ldots, z_n) dz_1 \wedge \ldots \wedge dz_n.$$

Note that under the change of coordinates, f would be multiplied by the determinant of the Jacobian of an invertible function, i.e. it makes sense to talk about the zeroes and poles of σ as being those of f. We can thus define the divisor of σ as the integral linear combination of divisors, "essentially codimension 1 subvarieties" :

$$div\, \sigma = \sum_C v(\sigma, C) C$$

where $v(\sigma, C)$ is the order of vanishing of f along C. The divisor is an element of the free group generated by codimension one subvarieties.

Example A.1. *On the projective line, $\mathbb{P}^1$, a section of the cotangent bundle is given by*

$$\sigma = dz_1$$

on the chart around 0. On the chart around ∞, one can use the coordinate z_2 related to z_1 by $z_1 z_2 = 1$ outside $\{0, \infty\}$. In terms of z_2, σ takes the form

$$d\frac{1}{z_2} = -\frac{dz_2}{z_2^2}.$$

While σ has no zeroes nor poles around 0, we observe a pole of order 2 at infinity. We thus have $div\,\sigma = -2(\infty)$.

Another choice of section would have given us another divisor, however, it is not hard to see that these choices would be linearly equivalent, i.e. their difference is the divisor coming from a meromorphic function.

Example A.2. *In the above example, we could have taken the section $\sigma' = (z_1 - 1)dz_1$. Around infinity, $\sigma' = \frac{z_2 - 1}{z_2^3} dz_2$ and $div\,\sigma' = 1(1) - 3(\infty)$.*

Now, letting

$$\Delta := div(\sigma) - div(\sigma') = -(1) + (\infty)$$

we observe that

$$\Delta = div\,\frac{1}{z-1}$$

and hence that the two divisors are linearly equivalent.

Definition 23. The canonical class, K_X, of a smooth variety X is the divisor associated to a section of its canonical bundle.

Example A.3. *For the projective line $K_{\mathbb{P}^1} = -2(\infty)$ where (∞) is the class of ∞ (or any other point for that matter!).*

A.2 ADJUNCTION

One of the benefits of considering the canonical class is that in many instances, when the space is well understood, it is not too hard to compute. One of these scenarios is when we try to find the canonical divisor of the subvariety Y of the ambient space X for which the canonical divisor is known.

Proposition A.4 (Adjunction Formula)**.** *Let Y be a smooth hypersurface of a smooth variety X then*

$$K_Y = (K_X + Y)_Y \,.$$

The right hand side tells us to take a divisor which is equivalent to $K_X + Y$, transverse to Y and restrict it to Y.

Example A.5. *(i) Let (H) denote a hyperplane in $\mathbb{P}^n$. The Picard group of divisors is generated by $H = \mathbb{P}^{n-1}$ and the intersection pairing is a linear extension of $H^2 = 1$. Using adjunction, one can show recursively that $K_{\mathbb{P}^n} = -(n+1)(H)$. (Note that in the case of the projective line, this is the result which we obtained above, namely $K_{\mathbb{P}^1} = -2(\infty)$.*