

DELETION AND CONTRACTION GAMES: CHROMATIC POLYNOMIAL  
PROOFS AND MAKING SENSE OF AN APPLE TREE

HONORS THESIS  
ITHACA COLLEGE

DEPARTMENT OF MATHEMATICS

BY  
DONNY TANG

FACULTY MENTOR: THOMAS J. PFAFF

ITHACA, NY  
MAY 13, 2005

## **ABSTRACT**

Using the deletion contraction algorithm we can find the chromatic polynomials for graphs. Similar to combinatorial proofs, we can apply this algorithm in different ways to the same graph to derive polynomial identities. Also, we will be looking at a couple of results from a previous paper and provide alternate proofs. Lastly, we will give a formula for the chromatic polynomial of an Apple Tree. Roughly, an Apple Tree is a tree with cycles attached at its vertices.

## TABLE OF CONTENTS

ABSTRACT . . . . .	ii
1 INTRODUCTION . . . . .	1
2 CHROMATIC POLYNOMIAL PROOFS . . . . .	5
3 A NEW LOOK AT OLD RESULTS . . . . .	14
4 THE CHROMATIC POLYNOMIAL OF AN APPLE TREE . . . . .	18

# CHAPTER 1

## INTRODUCTION

In this paper, we will be proving three polynomial identities that were derived by using graphs and their chromatic polynomials. We do this by using chromatic polynomial proofs, which are similar to proving combinatorial identities, on three types of graphs: trees, cycles, and wheels. After, we will be going over a couple of results given by Meredith, [2]. We will reprove one of his theorems to get a better sense why the theorem works, and we will revise his last theorem, making it a little easier to understand. Finally, we will look at the chromatic polynomial of a graph called an Apple Tree.

Before we begin studying the polynomial identities, we will first define a chromatic polynomial, and then give the algorithm which is used to find the polynomial. Let  $\lambda \in \mathbb{N}$ . A chromatic polynomial of a graph  $G$ , denoted  $P_G(\lambda)$ , calculates the number of ways the vertices of  $G$  can be colored using  $\lambda$  colors, without adjacent vertices having the same color. One application for a chromatic polynomial is to help solve scheduling conflicts. For example, given a set of jobs, some of which cannot be done at the same time, we can find the least amount of time that is needed to complete all the tasks. We will let each job be represented by a vertex, and two vertices be adjacent to each other if the two jobs cannot be done concurrently. Finding the chromatic polynomial of this graph would then give us the solution to this problem.

One method to compute the chromatic polynomials of graphs is called the Deletion and Contraction Algorithm. This algorithm deletes and contracts edges of a graph

and reduces the graph down to simpler graphs, such as trees and null graphs, which have chromatic polynomials that are easier to compute. And at the end, we can then sum up all these polynomials, which would give us the chromatic polynomial of the whole graph.

To see how this algorithm works, we will define three graphs,  $G$ ,  $G_1$ , which is formed by deleting an edge  $\alpha$ , and  $G_2$ , which is formed by contracting  $\alpha$ . Let  $P_G$ ,  $P_{G_1}$ , and  $P_{G_2}$  be the chromatic polynomial of a graph  $G$ ,  $G_1$  and  $G_2$  respectively. Let the two vertices that are connected by  $\alpha$  be vertex  $a$  and vertex  $b$ .  $P_{G_1}$  represents the number of ways  $G_1$  can be colored when  $a$  and  $b$  have the same coloring and when they have different colorings. So we get that  $P_{G_1} = P_G + P_{G_2}$  because  $P_G$  represents a graph when  $a$  and  $b$  have different colors, and  $P_{G_2}$  represents a graph when  $a$  and  $b$  are the same color (since  $a = b$  when the edge connecting them is contracted). So this gives us that  $P_G = P_{G_1} - P_{G_2}$ .

The following, provided by Brualdi [1], is a formal algorithm for the Deletion and Contraction algorithm. Figure 1.1 shows a graph with four vertices being reduced using this algorithm. Let  $G$  be a graph.

- 1- Give  $G$  positive value.
- 2- While there is a signed graph, and an edge,  $\alpha$ , in the signed graph, do:
  - i- Choose a nonnull signed graph and an edge  $\alpha$ .
  - ii- Remove  $\alpha$  from the graph, while keeping its sign if  $\alpha$  was deleted, and negate the sign if  $\alpha$  was contracted.
- 3- Sum up all the chromatic polynomials of the null graphs with the appropriate signs.

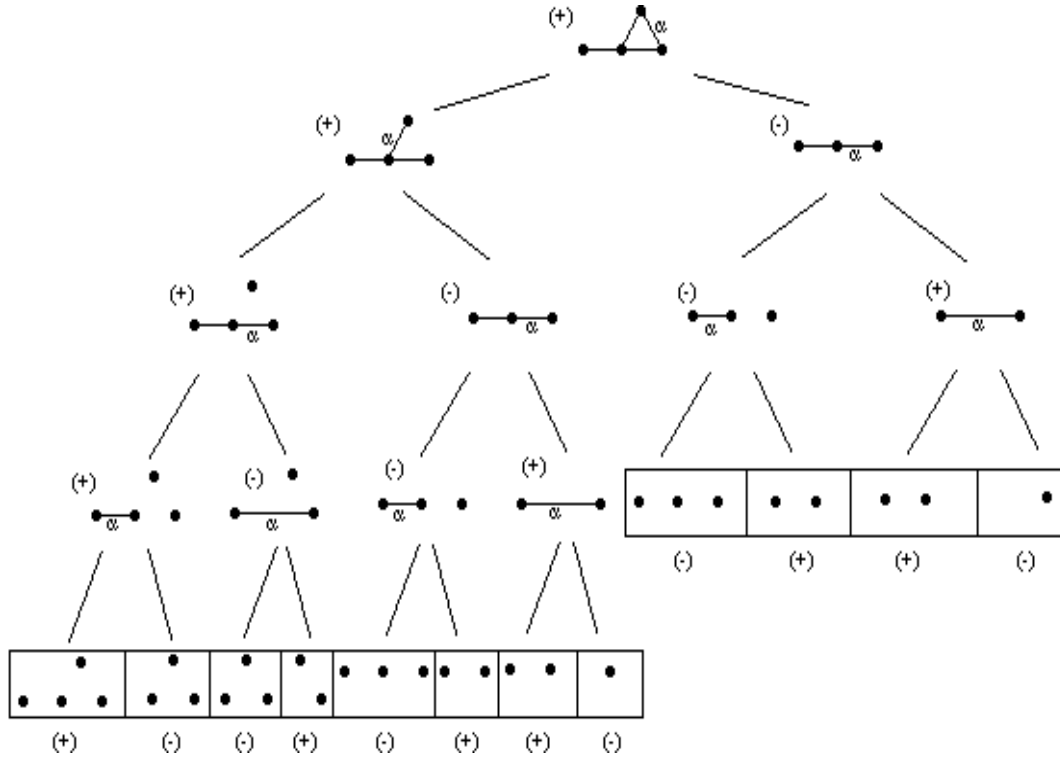


Figure 1.1: Reducing a graph with four vertices to its null graphs using the Deletion and Contraction Algorithm.

In Figure 1.1, each step to the left represents a deletion, and the step on the right represents a contraction. After each step, if the graph is not reduced down to a null graph, the algorithm is repeated. At the end of the algorithm, only null graphs remain. Since the chromatic polynomial of a null graph of order  $n$  is  $\lambda^n$ , we get that  $P_G(\lambda)$  for the graph in Figure 1.1 is  $\lambda^4 - \lambda^3 - \lambda^3 + \lambda^2 - \lambda^3 + \lambda^2 + \lambda^2 - \lambda^1 - \lambda^3 + \lambda^2 + \lambda^2 - \lambda^1 = \lambda^4 - 4\lambda^3 + 5\lambda^2 - 2\lambda$ .

There are three polynomial identities that we will prove by using their chromatic polynomials. For these identities, we will use the graphs of trees, cycles and wheels. A cycle of order  $n$ ,  $C_n$ , is a connected graph with  $n$  vertices that has its initial vertex equaling its ending vertex in a chain, with no other vertices being passed through more than once. A tree of order  $n$ ,  $T_n$ , is a connected graph with  $n$  vertices and

contains no cycles. And lastly, a wheel of order  $n$ ,  $W_n$ , is a cycle of order  $n - 1$  plus a vertex who is adjacent to every other vertex. We will derive our polynomial identities by using the deletion and contraction algorithm and looking at the different ways we can reduce the graphs. We call these proofs chromatic polynomial proofs since they are similar to combinatorial proofs, in that we will be computing the chromatic polynomial of a graph in two different ways.

After that, we will consider a paper written by Meredith, [2]. His paper gives three theorems, two of which we will be looking at, the first and the last. His first theorem gives us a bound for the value of the coefficients of the chromatic polynomial of any graph. His last theorem gives the coefficients of a graph that has exactly one cycle.

The final part of this paper will be on Apple Trees. An Apple Tree,  $A_k$  is a graph with exactly  $k$  cycles, with each cycle of order  $x_i$  for  $1 \leq i \leq k$ , and  $n$  vertices. Let  $x = \sum_1^k x_i$ . Each cycle may only share its vertices. The name Apple Tree is given because the graph is basically a tree with cycles attached to the graph at its vertices. In this section, we will derive the chromatic polynomial of an Apple Tree, which is

$$P_{A_k}(\lambda) = \sum_{j=1}^{x-2k+1} (-1)^{j-1} \left[ T_{\{x_i\}_1^k}(x - k + 2 - j)\lambda(\lambda - 1)^{n-j} \right],$$

where

$$T_{\{x_i\}_1^k}(j) = \sum_{h=1}^{x_k-1} T_{\{x_i\}_1^{k-1}}(j - h),$$

and

$$T_{\{x_1\}}(j) = \begin{cases} 1 & \text{if } 2 \leq j \leq x_1; \\ 0 & \text{otherwise.} \end{cases}$$

## CHAPTER 2

### CHROMATIC POLYNOMIAL PROOFS

In this chapter, we will be looking at three polynomial identities. Similar to combinatorial proofs, we will calculate the chromatic polynomials for trees, cycles and wheels in two ways (three for the cycles). Proofs for these identities were found mainly by using the deletion-contraction algorithm.

**Identity 2.0.1.** For  $n > 1$  and  $\lambda \in \mathbb{R}$ ,

$$\sum_{i=0}^{n-1} (-1)^i \binom{n-1}{n-1-i} \lambda^{n-i} = \lambda(\lambda-1)^{n-1},$$

which is  $P_{T_n}(\lambda)$ , the chromatic polynomial of a tree with  $n$  vertices.

*Proof.* One way to compute  $P_{T_n}(\lambda)$  is to directly count the number of ways to color a tree with  $\lambda$  colors. This proof is similar to the one given in [1]. Let  $T_n$  be a tree with  $n$  vertices,  $n \geq 2$ , and note that  $T_n$  has at least two pendent vertices. Starting at one of the pendent vertices, we can color that vertex with any of the  $\lambda$  colors. The vertex or vertices that are adjacent to the pendent vertex can then be colored in  $\lambda - 1$  ways, since adjacent vertices can't have the same coloring. If we continue coloring the adjacent vertices until all  $n$  vertices are colored, we would have all the vertices, other than the first colored in  $\lambda - 1$  ways. Hence  $P_{T_n}(\lambda) = \lambda(\lambda - 1)^{n-1}$ .

Another way to calculate  $P_{T_n}(\lambda)$  is by using the deletion-contraction algorithm and count the number of ways to delete edges from the graph  $G$  so it reduces to a null graph. Note that the chromatic polynomial of a null graph of order  $n$  is  $\lambda^n$ . Figure

2.1 shows a tree of order 5,  $T_5$  being reduced to its null graphs. To get the null graph of order  $n$  from a tree of order  $n$ , we must delete all  $n - 1$  edges, since there are no cycles, a contraction always removes exactly one edge. There is  $\binom{n-1}{n-1}$ , only one, way to delete all the edges, which gives us the polynomial  $\binom{n-1}{n-1}\lambda^n$ . To get the null graph of order  $n - 1$  from a tree of order  $n$ , we must delete  $n - 2$  edges and contract one. This gives us  $\binom{n-1}{n-2}$  ways to delete the edges, a polynomial of  $-\binom{n-1}{n-2}\lambda^{n-1}$ . We want to contract and delete the graph's edges until we get to the null graph of one vertex. In general, to get a null graph of order  $k$  from  $T_n$ , there are  $\binom{n-1}{k-1}$  ways to delete the edges, which has a polynomial of  $(-1)^{n-k}\binom{n-1}{k-1}\lambda^k$ . For instance in Figure 2.1, there are  $\binom{5}{k}$  null graphs of order  $k$ . Thus,

$$\begin{aligned} P_{T_n}(\lambda) &= \binom{n-1}{n-1}\lambda^n - \binom{n-1}{n-2}\lambda^{n-1} + \dots + \binom{n-1}{0}(-1)^{n-1}\lambda^1 \\ &= \sum_{i=0}^{n-1} (-1)^i \binom{n-1}{n-1-i} \lambda^{n-i}. \end{aligned}$$

□

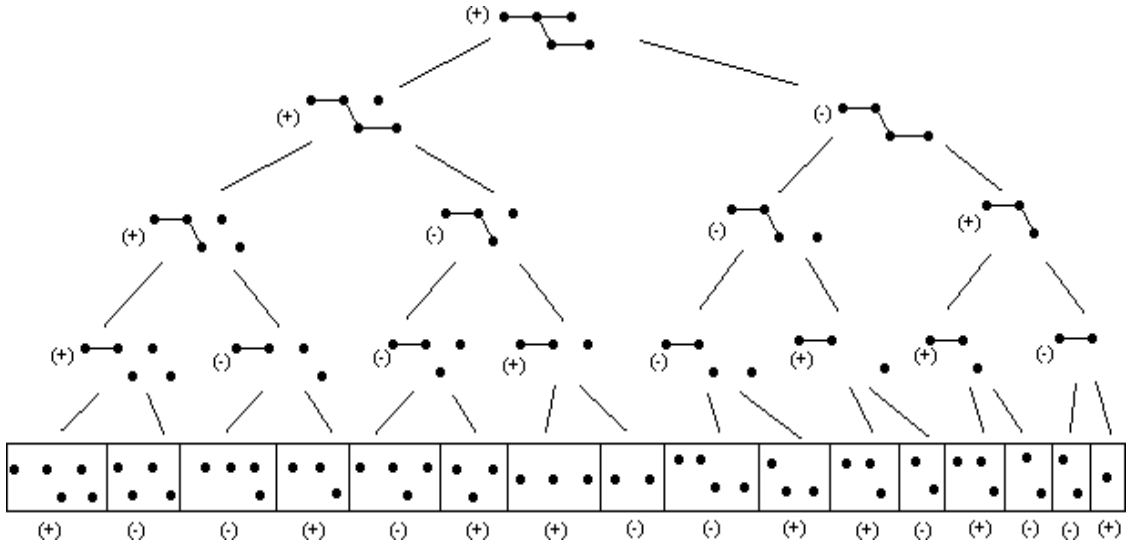


Figure 2.1: A tree of order 5,  $T_5$ , being broken down to its null graphs.

Of course, we can also derive Identity 2.0.1 by using the binomial theorem, we have that

$$\begin{aligned}\lambda(\lambda - 1)^{n-1} &= \lambda \left[ \binom{n-1}{n-1} \lambda^{n-1} - \binom{n-1}{n-2} \lambda^{n-2} + \dots + \binom{n-1}{0} (-1)^{n-1} \lambda^0 \right] \\ &= \sum_{i=0}^{n-1} (-1)^i \binom{n-1}{n-1-i} \lambda^{n-i}.\end{aligned}$$

**Identity 2.0.2.** For  $n > 2$  and  $\lambda \in \mathbb{R}$ ,

$$\left[ \sum_{i=0}^{n-1} (-1)^i \binom{n}{n-i} \lambda^{n-i} \right] + (-1)^n \lambda = \sum_{i=0}^{n-2} (-1)^i \lambda (\lambda - 1)^{n-i-1} \quad (2.1)$$

$$= (\lambda - 1)^n + (-1)^n (\lambda - 1), \quad (2.2)$$

which is  $P_{C_n}(\lambda)$ , the chromatic polynomial of a cycle with  $n$  vertices.

*Proof.* One way to calculate  $P_{C_n}(\lambda)$  is by counting the number of ways to delete edges, reducing the original graph to a null graph. To get a null graph of  $n$ -vertices, from an  $n$ -cycle, we need to delete all the edges, which we can do in  $\binom{n}{n}$  ways and contributes the polynomial term of  $\binom{n}{n} \lambda^n$  to  $P_{C_n}(\lambda)$ . To get a null graph of  $(n-1)$ -vertices, we can delete all but one edge, which we can do in  $\binom{n}{n-1}$  ways. Generalizing, if we want the null graph of  $k$  vertices,  $2 \leq k \leq n$ , we can delete in  $\binom{n}{k}$  ways, and have the polynomial  $(-1)^k \binom{n}{k} \lambda^k$ . However, when we want the null graph of one vertex, we can either delete 1 edge, which can be done in  $\binom{n}{1}$  ways, or contract all the edges, which gives a term of

$$(-1)^{n-1} \left[ \binom{n}{1} \lambda - \lambda \right].$$

Hence,

$$\begin{aligned} P_{C_n}(\lambda) &= \lambda^n - \binom{n}{n-1}\lambda^{n-1} + \binom{n}{n-2}\lambda^{n-2} + \dots + \binom{n}{1}(-1)^{n-1}\lambda + (-1)^n\lambda \\ &= \sum_{i=0}^{n-1} (-1)^i \binom{n}{n-i}\lambda^{n-i} + (-1)^n\lambda. \end{aligned}$$

On the other hand, we can derive  $P_{C_n}(\lambda)$  by contacting and deleting the cycle down to trees. Note that deleting and contracting an edge from an  $n$ -cycle gives us an  $n$ -tree and an  $(n-1)$ -cycle, which can be seen in Figure 2.2. By deleting an edge from the  $n$ -cycle, we end up with a  $n$ -tree, which has the chromatic polynomial of  $\lambda(\lambda-1)^{n-1}$ . Looking at the contraction step of the  $n$ -cycle, we get an  $(n-1)$ -cycle. We then want to take the  $(n-1)$ -cycle and delete and contract it down to a 3-cycle, which gives us the polynomial terms  $(-1)^{n-k}\lambda(\lambda-1)^{k-1}$ ,  $4 \leq k \leq n$ . The 3-cycle deletes to a 3-tree, a chromatic polynomial of  $\lambda(\lambda-1)^2$  and contracts to a 2-tree, a term of  $\lambda(\lambda-1)$ , as seen in Figure 2.2. If we add up all the polynomials we got from the trees of sizes two to  $n-1$ , we get that

$$\begin{aligned} P_{C_n}(\lambda) &= \lambda(\lambda-1)^{n-1} - \lambda(\lambda-1)^{n-2} + \lambda(\lambda-1)^{n-3} + \dots + (-1)^n\lambda(\lambda-1) \\ &= \sum_{i=0}^{n-2} (-1)^i \lambda(\lambda-1)^{n-i-1}. \end{aligned}$$

Lastly, we can show  $P_{C_n}(\lambda) = (\lambda-1)^n + (-1)^n(\lambda-1)$  by an induction proof, which is similar to Read's proof in [3]. We will let  $n=3$  for the base case since a cycle must have at least 3 vertices. By one deletion and contraction, we get a 3-tree

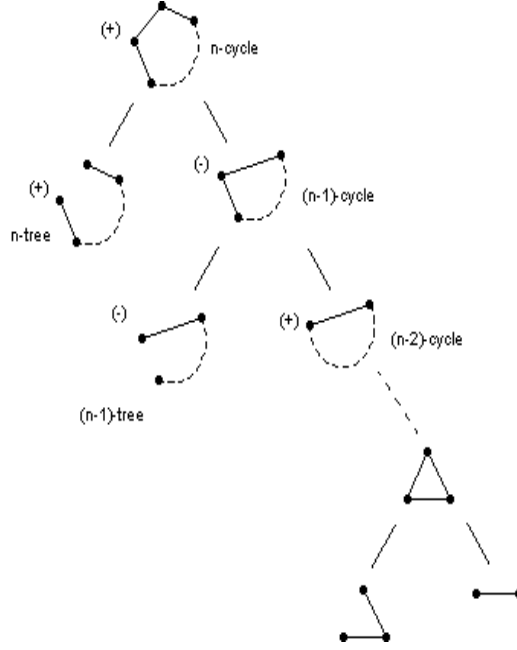


Figure 2.2: An  $n$ -cycle being broken down into its trees.

and a 2-tree, so

$$P_{C_3}(\lambda) = \lambda(\lambda - 1)^2 - \lambda(\lambda - 1) \quad (2.3)$$

$$= (\lambda - 1)^3 - (\lambda - 1). \quad (2.4)$$

Assuming that  $P_{C_n}(\lambda) = (\lambda - 1)^n + (-1)^n(\lambda - 1)$  is true, consider the  $n + 1$  case. By deleting and contracting one edge, we get an  $(n + 1)$ -tree and a  $C_n$ , so that

$$\begin{aligned} P_{C_{n+1}}(\lambda) &= \lambda(\lambda - 1)^n - P_{C_n}(\lambda) \\ &= \lambda(\lambda - 1)^n - [(\lambda - 1)^n + (-1)^n(\lambda - 1)] \\ &= (\lambda - 1)^{n+1} + (-1)^{n+1}(\lambda - 1). \end{aligned}$$

Thus,  $P_{C_n}(\lambda) = (\lambda - 1)^n + (-1)^n(\lambda - 1)$ . □

Of course, by the binomial theorem, we can show that the first part of 2.1 is equal to 2.2, since

$$\begin{aligned}
& (\lambda - 1)^n + (-1)^n(\lambda - 1) \\
&= \sum_{i=0}^n \left[ \binom{n}{i} (-1)^{n-i} \lambda^i \right] + (-1)^n(\lambda - 1) \\
&= \left[ \lambda^n - \binom{n}{n-1} \lambda^{n-1} + \binom{n}{n-2} \lambda^{n-2} + \dots + \binom{n}{0} (-1)^n \right] + (-1)^n(\lambda - 1) \\
&= \lambda^n - \binom{n}{n-1} \lambda^{n-1} + \binom{n}{n-2} \lambda^{n-2} + \dots + \binom{n}{1} (-1)^{n-1} \lambda + (-1)^n \lambda \\
&= \sum_{i=0}^{n-1} (-1)^i \binom{n}{n-i} \lambda^{n-i} + (-1)^n \lambda.
\end{aligned}$$

An induction proof can also be used to prove that  $P_{C_n}(\lambda) = \sum_{i=0}^{n-2} (-1)^i \lambda(\lambda - 1)^{n-i-1}$ . By (2.3), our base case of  $n = 3$  is true. Assume  $P_{C_n}(\lambda) = \sum_{i=0}^{n-2} (-1)^i \lambda(\lambda - 1)^{n-i-1}$  is true. To get the chromatic polynomial of an  $n + 1$ -cycle, we can follow the contract and delete algorithm. When we delete an edge we get a tree of length  $n + 1$ . By Identity 2.0.1, we get that the chromatic polynomial of an  $n+1$ -tree is  $\lambda(\lambda - 1)^n$ . When we contract, we get an  $n$ -cycle. By assumption, we know that an  $n$ -cycle produces  $\sum_{i=0}^{n-2} (-1)^i \lambda(\lambda - 1)^{n-i-1}$ . Combining the two we get that the chromatic polynomial of an  $n + 1$ -cycle, which is

$$\begin{aligned}
P_{C_{n+1}}(\lambda) &= \lambda(\lambda - 1)^n - \sum_{i=1}^{n-1} (-1)^{n-i+1} \lambda(\lambda - 1)^i \\
&= \sum_{i=0}^{n-1} (-1)^i \lambda(\lambda - 1)^{n-i}.
\end{aligned}$$

Therefore, the chromatic polynomial of an  $n$ -cycle is  $\sum_{i=0}^{n-2} (-1)^i \lambda(\lambda - 1)^{n-i-1}$ .

Many methods of counting were used to try to find a combinatorial proof for

$P_{C_n}(\lambda) = (\lambda - 1)^n + (-1)^n(\lambda - 1)$ , such as directly coloring the vertices, as in the tree case. But in the end, all of them needed some sort of recursive step starting at a 3-cycle, so the combinatorial proof was abandoned. However, it was a lot easier to formulate a proof for the  $n$ -cycle by the deletion and contraction algorithm.

**Identity 2.0.3.** For  $n > 3$  and  $\lambda \in \mathbb{R}$ ,

$$\lambda[(\lambda - 2)^{n-1} + (-1)^{n-1}(\lambda - 2)] = (-1)^{n-1}\lambda(\lambda - 1) + \sum_{i=1}^{n-1} (-1)^{n-i-1} \binom{n-1}{i} \lambda(\lambda - 1)^i,$$

which is  $P_{W_n}(\lambda)$ , the chromatic polynomial of a wheel with  $n$  vertices.

*Proof.* One way to find  $P_{W_n}(\lambda)$  is by looking first at the vertex which is adjacent to all other vertices, call it  $v$ , and see how many different ways we can color it. We can color  $v$  in  $\lambda$  ways. After coloring  $v$ , we can then color the rest of the graph, which is now a cycle, with  $\lambda - 1$  colors. From Identity 2.0.2, we get that the chromatic polynomial of the cycle is  $(\lambda - 2)^{n-1} + (-1)^{n-1}(\lambda - 2)$ . Hence  $P_{W_n}(\lambda) = \lambda[(\lambda - 2)^{n-1} + (-1)^{n-1}(\lambda - 2)]$ .

Another way to calculate  $P_{W_n}(\lambda)$  is by counting the number of ways to contract edges. Using the deletion-contraction algorithm, we will, one-by-one, contract the edges on the outside of the wheel, to produce trees, shown in Figure 2.3. Given an  $n$ -wheel, there are  $\binom{n-1}{0}$  ways to contract the edges to form an  $n$ -tree, because we would need to delete all  $n - 1$  of the outside edges. This gives us the term  $\binom{n-1}{0}\lambda(\lambda)^{n-1}$  in  $P_{W_n}(\lambda)$ . To get a  $(n - 1)$ -tree, we can contract one edge from the outside. There are  $\binom{n-1}{1}$  ways to do this, giving us the term  $-\binom{n-1}{1}\lambda(\lambda)^{n-2}$ . Note that when doing a contraction to an  $n$ -wheel, the result is an  $(n - 1)$  wheel, if  $n > 4$ . In general, if we want the tree of order  $k$ ,  $4 \leq k \leq n$  we can contract in  $\binom{n-1}{n-k}$  ways. Looking at when we want a 3-tree, we can contract  $n - 3$  edges. However, when we start with  $n - 3$

contractions, we get a 3-cycle. So instead of two deletions, we just do one. This still gives us  $\binom{n-1}{n-3}$  ways to contract to get a 3-tree. To get a 2-tree, we can contract  $n-2$  edges. This is shown in the right side of Figure 2.3. Looking at the cases where we start with  $n-3$  contractions (C,C,C,...,C,D,C and C,C,C,...,C,C,D), we see that the case with  $n-3$  contractions, one deletion, and one contraction actually gives us a 3-tree, without doing the last contraction, so we can take out that case. The second case where we start with  $n-2$  contractions and one deletion gives us a 2-tree without doing the last deletion. Counting all these combinations, we get  $\binom{n-1}{n-2} - 1$  ways to contract to a 2-tree. Combining the terms, we get that

$$\begin{aligned}
P_{W_n}(\lambda) &= \binom{n-1}{n-1} \lambda(\lambda-1)^{n-1} - (-1)^{n-(n-2)-1} \binom{n-1}{n-2} \lambda(\lambda-1)^{n-2} + \dots + \\
&\quad (-1)^{n-2-1} \binom{n-1}{2} \lambda(\lambda-1)^2 + (-1)^{n-1-1} \left[ \binom{n-1}{1} - 1 \right] \lambda(\lambda-1) \\
&= \sum_{i=1}^{n-1} \left[ (-1)^{n-i-1} \binom{n-1}{i} \lambda(\lambda-1)^i \right] - (-1)^{n-2} \lambda(\lambda-1).
\end{aligned}$$

□

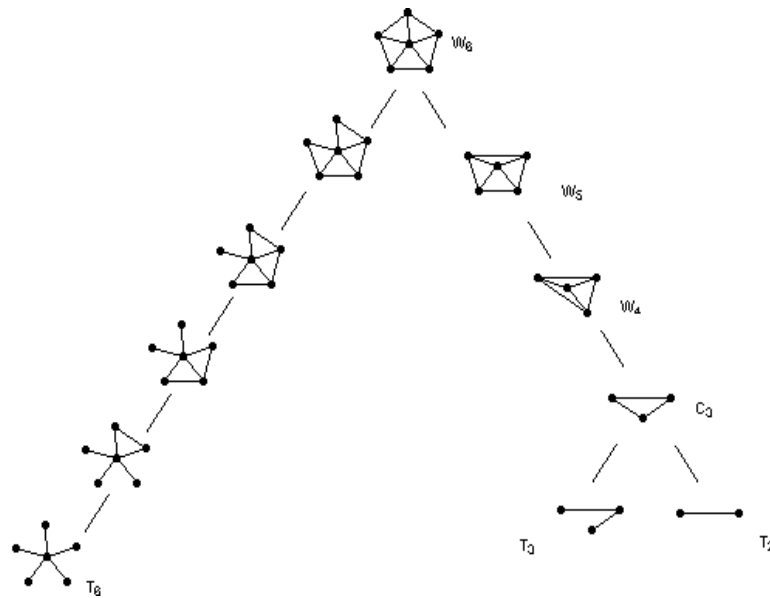


Figure 2.3: Partial reduction of a wheel of order 6,  $W_6$ . This shows how we can get the wheel down to its trees.

## CHAPTER 3

### A NEW LOOK AT OLD RESULTS

This chapter focuses on a paper written by Meredith, [2]. The paper gave a couple of theorems that give the coefficients of chromatic polynomials, one for a graph of  $n$  vertices and  $k$  edges (Theorem 1 in [2]), and the latter, for a graph with one cycle of a specific length (Theorem 3 in [2]). Since Meredith's first theorem was given an induction proof that is not very enlightening, and the third theorem gave only coefficients for graphs with a very specific structure, we will turn to the deletion and contraction algorithm to prove the first theorem, in place of the induction proof Meredith gave. We will use the deletion and contraction algorithm again to prove a similar version of Meredith's third theorem.

**Theorem 3.0.4.** *If a connected graph  $G$  has  $n$  vertices,  $k$  edges and if the coefficient of  $\lambda^r$  of  $P_G(\lambda)$ , the chromatic polynomial of  $G$ , is  $\alpha_r$ , then  $|\alpha_r| \leq \binom{k}{n-r}$ .*

*Proof.* Given a graph with  $n$  vertices and  $k$  edges, to find  $\alpha_r$ , we would need to contract and delete to an  $r$  null graph. To do so, we must contract exactly  $n - r$  edges, since a contraction is the only way to reduce vertices, which can be done in  $\binom{k}{n-r}$  ways. However, some graphs contract more than one edge when doing a single contraction step. For example, a 3-cycle contracts 2 edges in one step. Hence some graphs have fewer than  $\binom{k}{n-r}$  contractions that can be realized. Therefore,  $|\alpha_r| \leq \binom{k}{n-r}$ .

□

Unlike Meredith's proof by induction, we proved this theorem by simply counting the ways we can contract edges.

**Theorem 3.0.5.** *Let  $G$  be a connected graph with  $n$  vertices and one cycle of order  $x$ . Then  $P_{G_n}(\lambda) = \sum_{i=1}^{x-1} (-1)^{i-1} \lambda(\lambda-1)^{n-i}$ . Furthermore, if  $\alpha_r$  is the coefficient for  $\lambda^r$ , then  $|\alpha_r| = \binom{n}{r} - \binom{n-x+1}{r}$ .*

*Proof.* Let  $G$  be a graph with  $n$  edges and one cycle of length  $x$ . First, we want to calculate  $P_{G_n}(\lambda)$ , by deleting and contracting to trees from this graph. To do this, we can use the deletion-contraction algorithm on the  $x$ -cycle. A deletion of one of the edges on the  $x$ -cycle, would leave us with an  $n$ -tree, which has a chromatic polynomial of  $\lambda(\lambda-1)^{n-1}$ . Looking at the contraction step, we are left with a graph with  $n-1$  vertices and exactly one  $x-1$ -cycle. If we do a deletion, we would get an  $n-1$ -tree, with the chromatic polynomial of  $\lambda(\lambda-1)^{n-2}$ . A contraction would leave us with a graph with  $n-2$  vertices and exactly one  $x-2$ -cycle. If we do  $k$  contractions and a deletion,  $0 \leq k \leq n-3$ , we get a  $(n-k)$ -tree with the chromatic polynomial of  $\lambda(\lambda-1)^{n-k-1}$ . Looking at the 3 cycle now, a deletion from the 3-cycle would leave us with an  $(n-x+3)$ -tree,  $\lambda(\lambda-1)^{n-x+2}$ , and a contraction would give us an  $(n-x+2)$ -tree,  $\lambda(\lambda-1)^{n-x+1}$ . Combining the pieces of chromatic polynomials we found from the trees in the graph and remembering that contractions give negative addends, we get,

$$\begin{aligned} P_{G_n}(\lambda) &= \lambda(\lambda-1)^{n-1} - \lambda(\lambda-1)^{n-2} + \dots + \\ &\quad (-1)^{x-3} \lambda(\lambda-1)^{n-x+2} + (-1)^{x-2} \lambda(\lambda-1)^{n-x+1} \\ &= \sum_{i=1}^{x-1} (-1)^{i-1} \lambda(\lambda-1)^{n-i}. \end{aligned}$$

To get  $\alpha^r$ , we will look at our previous result. By the binomial theorem, we get that,

$$\sum_{i=1}^{x-1} (-1)^{i-1} \lambda (\lambda - 1)^{n-i} = \sum_{i=1}^{x-1} \sum_{j=0}^{n-i} (-1)^{n-j} \binom{n-i}{j} \lambda^{j+1}. \quad (3.1)$$

Using the Hockey Stick Lemma, which says  $\binom{m}{m} + \binom{m+1}{m} + \binom{m+2}{m} + \dots + \binom{n-2}{m} + \binom{n-1}{m} = \binom{n}{m+1}$ ,  $m \in \mathbb{N}$ , we can sum up the coefficients of each  $\lambda^r$ . However, the coefficients stop at a certain point, in our case, when we have  $\binom{n-x+1}{m}$ . Taking the coefficient of  $\lambda^i$  in 3.1, we have, by the Hockey Stick lemma,

$$\begin{aligned} |\alpha_r| &= \binom{n-1}{r-1} + \binom{n-2}{r-1} + \binom{n-3}{r-1} + \dots + \binom{n-x+2}{r-1} + \binom{n-x+1}{r-1} \\ &= \binom{n-1}{r-1} + \binom{n-2}{r-1} + \dots + \binom{2}{r-1} + \binom{1}{r-1} - \\ &\quad \left[ \binom{n-x}{r-1} + \binom{n-x-1}{r-1} + \dots + \binom{2}{r-1} + \binom{1}{r-1} \right] \\ &= \binom{n}{r} - \binom{n-x+1}{r}. \end{aligned}$$

So  $|\alpha_r| = \binom{n}{r} - \binom{n-x+1}{r}$ . □

In Meredith's paper, his third theorem states that if  $G$  has just one circuit, of length  $n - s + 1$  then

- (a)  $|\alpha_r| = \binom{k}{n-r}$ , if  $r > s$
- (b)  $|\alpha_r| = \binom{k}{n-r} - \binom{k-n+s}{s-r}$ , if  $r \leq s$ ,

where  $k$  is the number of edges and  $n$  is the number of vertices.

Note that in a graph with only one cycle, the total number of vertices,  $n$ , and edges,  $k$ , are the same (i.e. add an edge to a tree). So we can actually replace  $k$  by

$n$ . Looking at case “a”, we can change that to

$$|\alpha_r| = \binom{n}{n-r},$$

and case “b” to

$$|\alpha_r| = \binom{n}{n-r} - \binom{s}{s-r}.$$

Also note that in a combination,  $\binom{p}{q}$ , if  $p < q$ , then  $\binom{p}{q} = 0$ . With that, we can combine cases “a” and “b” to

$$\begin{aligned} |\alpha_r| &= \binom{n}{n-r} - \binom{s}{s-r}, \\ &= \binom{n}{r} - \binom{n-x+1}{r}, \end{aligned}$$

since  $n - s + 1 = x$ , we get that  $s = n - x + 1$ . So now we can see Meredith’s third theorem is the same as Theorem 3.0.5 in terms of  $|\alpha_r|$ , but we also give a formula for  $P_G(\lambda)$  in terms of  $\lambda(\lambda - 1)$ .

Meredith’s proof for his third theorem counts the number of spanning subgraphs in his original graph to get the  $|\alpha_r|$ . In doing so, he used five different variables which ended up very confusing to follow. Our proof is just a straight forward calculation using the contraction and deletion algorithm, and counting the number of ways we can do each.

## CHAPTER 4

### THE CHROMATIC POLYNOMIAL OF AN APPLE TREE

In this chapter, we will look at deriving the chromatic polynomial of Apple Trees. Recall an Apple Tree,  $A_k$  is a graph with exactly  $k$  cycles, with each cycle of order  $x_i$  for  $1 \leq i \leq k$ , and  $n$  vertices. Let  $x = \sum_1^k x_i$ . Each cycle may only share its vertices. This chapter is an extension to our Theorem 3.0.5 result, which gives the chromatic polynomial and its coefficient for a graph of one cycle.

**Theorem 4.0.6.** *Let  $A_n$  be an Apple Tree with  $n$  vertices, and cycles of order  $\{x_1, x_2, \dots, x_k\}$ . Let  $x = \sum_1^k x_i$ . Then,*

$$P_{A_k}(\lambda) = \sum_{j=1}^{x-2k+1} (-1)^{j-1} T_{\{x_i\}_1^k}(x-k+2-j) \lambda(\lambda-1)^{n-j}, \quad (4.1)$$

where

$$T_{\{x_i\}_1^k}(j) = \sum_{h=1}^{x_k-1} T_{\{x_i\}_1^{k-1}}(j-h),$$

and

$$T_{\{x_1\}}(j) = \begin{cases} 1 & \text{if } 2 \leq j \leq x_1; \\ 0 & \text{otherwise.} \end{cases}$$

*Proof.* We will use induction to prove this theorem. When  $k = 1$ , we can use Theorem 3.0.5, which gives us,

$$P_{A_1}(\lambda) = \sum_{j=1}^{x_1-1} (-1)^{j-1} \lambda(\lambda-1)^{n-j}. \quad (4.2)$$

To see that 4.2 agrees with 4.1, simply evaluate 4.1 at  $k = 1$ . Note that  $T_{\{x_1\}}(j) = 1$  when  $2 \leq j \leq x_1$ , and 0 otherwise, so 4.1 agrees with 4.2.

Assume  $P_{A_k}(\lambda) = \sum_{j=1}^{x-2k+1} (-1)^{j-1} T_{\{x_i\}_1^k}(x - k + 2 - j) \lambda(\lambda - 1)^{n-j}$ . Consider the case  $k + 1$ . Now we have a total of  $n + x_{k+1} - 1$  vertices, where  $n$  is the total number of vertices for case  $k$ . We will delete and contract out the  $x_{k+1}$ -cycle. Doing the deletion-contraction algorithm on the  $x_{k+1}$ -cycle, with a deletion we get an apple tree with  $k$  cycles and  $n + x_{k+1} - 1$  vertices. The polynomial obtained is

$$\sum_{j=1}^{x-2k+1} (-1)^{j-1} T_{\{x_i\}_1^k}(x - k + 2 - j) \lambda(\lambda - 1)^{n+x_{k+1}-1-j}.$$

Notice that when  $j = 0$ , we have  $\lambda(\lambda - 1)^{n+x_{k+1}-1}$ , which is the chromatic polynomial of a tree that has more vertices than the original graph. So when  $j < 1$ ,  $T_{\{x_i\}_1^k}(x - k + 2 - j) = 0$ . Also, when  $j > x - 2k + 1$ ,  $\lambda(\lambda - 1)^{n+x_{k+1}-1-j}$  would have fewer vertices than the whole graph when only performing one deletion on the  $x_{k+1}$ -cycle. Therefore when  $j > x - 2k + 1$ ,  $T_{\{x_i\}_1^k}(x - k + 2 - j) = 0$ .

Looking at the contraction step, we are left with an  $(x_{k+1} - 1)$ -cycle attached to the  $k$ -apple tree,  $A_k$ . Performing a deletion to the  $(x_{k+1} - 1)$ -cycle, we get a  $k$ -apple tree with  $(n + x_{k+1} - 2)$  vertices, with the polynomial

$$\begin{aligned} (-1) \sum_{j=1}^{x-2k+1} (-1)^{j-1} T_{\{x_i\}_1^k}(x - k + 2 - j) \lambda(\lambda - 1)^{n+x_{k+1}-2-j} \\ = \sum_{j=2}^{x-2k+2} (-1)^{j-1} T_{\{x_i\}_1^k}(x - k + 3 - j) \lambda(\lambda - 1)^{n+x_{k+1}-1-j}. \end{aligned}$$

Using the same reasoning as above, when  $j < 2$  the resulting polynomial of a tree would have more vertices than the total vertices of our original graph with one con-

traction and a deletion. Therefore, when  $j < 2$ ,  $T_{\{x_i\}_1^k}(x - k + 3 - j) = 0$ . Same goes for when  $j > x - 2k + 2$ , the resulting polynomial has fewer vertices than the whole graph when a contraction and then a deletion is done. Hence, when  $j > x - 2k + 2$ ,  $T_{\{x_i\}_1^k}(x - k + 3 - j) = 0$

Continuing this algorithm, we get to a point when a contraction is done, our  $k + 1^{\text{th}}$  cycle becomes a 3-cycle. A deletion from this 3-cycle would give us

$$\begin{aligned} & (-1)^{x_{k+1}-3} \sum_{j=1}^{x-2k+1} (-1)^{j-1} T_{\{x_i\}_1^k}(x - k + 2 - j) \lambda(\lambda - 1)^{n+x_{k+1}-x_{k+1}+2-j} \\ &= \sum_{j=x_{k+1}-2}^{x-2k+x_{k+1}-2} (-1)^{j-1} T_{\{x_i\}_1^k}(x - k + x_{k+1} - 1 - j) \lambda(\lambda - 1)^{n+x_{k+1}-1-j}. \end{aligned}$$

A contraction on the 3-cycle would give us

$$\begin{aligned} & (-1)^{x_{k+1}-2} \sum_{j=1}^{x-2k+1} (-1)^{j-1} T_{\{x_i\}_1^k}(x - k + 2 - j) \lambda(\lambda - 1)^{n+x_{k+1}-x_{k+1}+1-j} \\ &= \sum_{j=x_{k+1}-1}^{x-2k+x_{k+1}-1} (-1)^{j-1} T_{\{x_i\}_1^k}(x - k + x_{k+1} - j) \lambda(\lambda - 1)^{n+x_{k+1}-1-j} \end{aligned}$$

Again, using the same reasoning as before, if the  $j$  values go beyond their bounded values, the resulting  $T_{\{x_i\}_1^k}$  values are equal to 0.

Finally adding all the polynomials together would give us that

$$P_{A_{k+1}}(\lambda) = \sum_{i=1}^{x_{k+1}-1} \sum_{j=i}^{x-2k+i} (-1)^{j-1} T_{\{x_i\}_1^k}(x - k + i + 1 - j) \lambda(\lambda - 1)^{n+x_{k+1}-1-j}.$$

Remember that there are no  $T_{\{x_i\}_1^k}(j)$  terms with  $j > x - k + 1$ , and  $j < k + 1$ , because they are zero. However, calculation would be easier if we put them back

in. For example, let's look at the coefficients of  $\lambda(\lambda - 1)^{n-x+2k}$ . If we leave out the coefficients where  $j < k + 1$ , we have  $T_{\{x_i\}_1^k}(k + 1)$ . However, if we add in all the coefficients that precedes  $j = k + 1$ , which are of value 0, then we have,

$$\begin{aligned} T_{\{x_i\}_1^k}(k + 1) &= \sum_{h=0}^{x_k-2} T_{\{x_i\}_1^k}(k + 1 - h) \\ &= \sum_{h=1}^{x_k-1} T_{\{x_i\}_1^k}(k + 2 - h) \\ &= T_{\{x_i\}_1^{k+1}}(k + 2). \end{aligned}$$

So if we add in the coefficients that come out to 0 in the sum, we get that

$$\begin{aligned} &\sum_{i=1}^{x_{k+1}-1} \sum_{j=i}^{x-2k+i} (-1)^{j-1} T_{\{x_i\}_1^k}(x - k + i + 1 - j) \lambda(\lambda - 1)^{n+x_{k+1}-1-j} \\ &= \sum_{j=1}^{x+x_{k+1}-2(k+1)+1} (-1)^{j-1} T_{\{x_i\}_1^{k+1}}(x + x_{k+1} - k + 1 - j) \lambda(\lambda - 1)^{n+x_{k+1}-1-j}. \end{aligned}$$

Note that for  $P_{A_{k+1}}$ , the upper limit is suppose to be  $x + x_{k+1} - 2(k + 1) + 1$  since we went from  $x$  to  $x + x_{k+1}$  and  $k$  to  $k + 1$ . The power of  $\lambda(\lambda - 1)$  went from  $n + x_{k+1} - 1 - j$  to  $n - j$  because the  $k + 1$ -cycle is left out and and the shared vertex doesn't get taken out from  $A_k$ . Therefore, by induction,

$$P_{A_k}(\lambda) = \sum_{j=1}^{x-2k+1} (-1)^{j-1} T_{\{x_i\}_1^k}(x - k + 2 - j) \lambda(\lambda - 1)^{n-j}.$$

□

This proof provides us with a chromatic class. In particular, Apple trees with same vertices and cycle structure have the same chromatic polynomial even though

the graphs are not isomorphic. Graphs that are not isomorphic can have the same chromatic polynomial, as long as they have the same cycle structure and the same number of vertices, which are the criteria for this theorem.

Another quality of apple trees that is interesting, is that  $T_{\{x_i\}_1^k}(j)$  corresponds with coefficients in the expansion of products of  $(1 + x^2 + x^3 + \dots + x^k)$  where  $k \in \mathbb{N}$  and  $j$  is the order of the tree. The factors for the product depends on the size of the cycles in our graph. If we have an  $n$ -cycle, we use  $(1 + x + x^2 + \dots + x^{n-2})$  as our factor.

For example, given an apple tree with the cycle structure of  $\{3, 4, 5\}$  and 12 vertices, the values of  $T_{\{3,4,5\}}(j)$  for  $6 \leq j \leq 12$  are the same as the coefficients of the product of  $(1 + x)(1 + x + x^2)(1 + x + x^2 + x^3)$ , which is given by sequence A008302 in [4]. Specifically, the values are given in Table 4.1.

$j$	$T_{\{3,4,5\}}(j)$	Terms of Expansion
12	1	$x^6$
11	3	$x^5$
10	5	$x^4$
9	6	$x^3$
8	5	$x^2$
7	3	$x$
6	1	1

Table 4.1: The values of  $T_{\{3,4,5\}}(j)$  and its corresponding terms.

Hence, the chromatic polynomial for an Apple Tree with a cycle structure of  $\{3, 4, 5\}$  and 12 vertices, remembering the alternating signs, is

$$\lambda(\lambda - 1)^{10} - 3\lambda(\lambda - 1)^9 + 5\lambda(\lambda - 1)^8 - 6\lambda(\lambda - 1)^7 + 5\lambda(\lambda - 1)^6 - 3\lambda(\lambda - 1)^5 + \lambda(\lambda - 1)^4.$$

## REFERENCES

- [1] R. A. Brualdi, *Introductory Combinatorics*, 3rd ed. Prentice-Hall, Upper Saddle River, New Jersey 1999.
- [2] G. H. J. Meredith, Coefficients of Chromatic Polynomials, *Journal of Combinatorial Theory* B13 (1972), 14-17.
- [3] R. C. Read, An Introduction to Chromatic Polynomials, *Journal of Combinatorial Theory* 4 (1968), 52-71.
- [4] N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*  
<http://www.research.att.com/~njas/sequences>, published electronically at  
[www.research.att.com/~njas/sequences](http://www.research.att.com/~njas/sequences).