8-2022

An Introduction to Combinatorics via Cayley's Theorem

Jaylee Willis
Utah State University

AN INTRODUCTION TO COMBINATORICS VIA CAYLEY'S THEOREM

by

Jaylee Willis

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

of

MASTER OF SCIENCE

in

Mathematics

Approved:

David Brown, Ph.D. Major Professor	Matthew Young, Ph.D. Committee Member
Zhaohu Nie, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY

Logan, Utah

2022

ABSTRACT

An Introduction to Combinatorics via Cayley's Theorem

by

Jaylee Willis, Master of Science

Utah State University, 2022

Major Professor: Dr. David Brown
Department: Mathematics and Statistics

In 1889, Arthur Cayley published his *Theorem on Trees* which stated that the number of trees with $n+1$ labeled knots is $(n+1)^{n-1}$. In most modern literature, this result is rephrased as the number of trees on vertices labeled with $\{v_1, v_2, ..., v_n\}$ is n^{n-2}.

The purpose of this paper is to give the reader an introduction to some of the methods that are often used to solve combinatorial problems by proving Cayley's theorem via inductive proofs, bijective proofs, double counting proofs, generating functions, linear algebra, and probability theory.

The intended audience of this paper is undergraduate and graduate mathematics students with little to no experience in combinatorics. It is possible that this paper could be used as a supplementary text for an undergraduate combinatorics course.

(103 pages)

PUBLIC ABSTRACT

An Introduction to Combinatorics via Cayley's Theorem

Jaylee Willis

In this paper, we explore some of the methods that are often used to solve combinatorial problems by proving Cayley's theorem on trees in multiple ways. The intended audience of this paper is undergraduate and graduate mathematics students with little to no experience in combinatorics. This paper could also be used as a supplementary text for an undergraduate combinatorics course.

ACKNOWLEDGMENTS

I give special thanks to my parents for inspiring me to love mathematics from a young age and for their enthusiastic support of all of my dreams. I thank Jean Culbertson, Dr. Jim Cangelosi, and Dr. David Brown for reigniting my love of mathematics during my undergraduate studies. I also want to thank Dr. Brown for his support as my advisor and my committee members Dr. Zhaohu Nie and Dr. Matthew Young for their mentoring this year. Thank you all!

Jaylee Willis

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

1 Introduction

Cayley's Theorem is a basic enumerative result in combinatorics that may have applications when tree graphs are studied. For the purpose of this paper, the primary appeal of Cayley's Theorem is that it can be proved in so many different ways and is therefore an excellent theorem to illustrate the various methods of combinatorics.

In this introduction section, we will define some basic graph theory terms, verify Cayley's Theorem by hand for small values of n, and explain Cayley's original proof. In the following sections, we will explore some of the strategies that are used to solve combinatorics problems by studying proofs of Cayley's Theorem that illustrate each strategy.

Many of the proofs in this paper use various binomial coefficient identities to simplify formulas. This introduction includes combinatorial proofs for these identities since they are used frequently in this paper.

1.1 General Definitions

The following definitions are taken from Douglas West's *Introduction to Graph Theory* [17].

Graph

A graph G is a triple consisting of a vertext set $V(G)$, an edge set $E(G)$, and a relation that associates with each edge two vertices (not necessarily distinct) called its endpoints.

Directed Graph

A directed graph G is a triple consisting of a vertext set $V(G)$, an edge set $E(G)$, and a function assigning each edge to an ordered pair of vertices. The first vertex of the ordered pair is the tail of the edge and the second is the head. Together, they are the endpoints of the edge.

Simple Graph

A loop is an edge whose endpoints are equal. Multiple edges are edges having the same pair of endpoints. A simple graph is a graph having no loops or multiple edges.

Multigraph

A multigraph is a graph that allows but does not necessarily require multiple edges and loops.

Graphs are often visualized with diagrams where the vertices are represented with labeled points and the edges are represented with lines or arrows between the vertices.

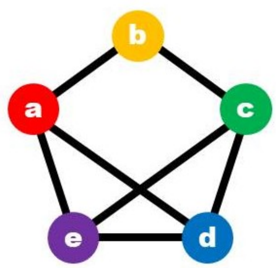
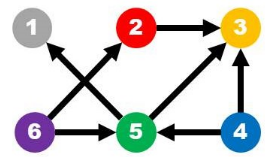
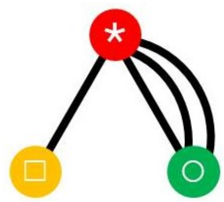
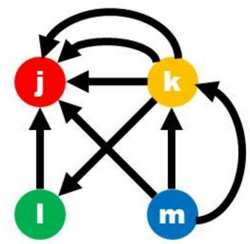
Undirected Simple Graph	Directed Simple Graph
	
Undirected Multigraph	Directed Multigraph
	

Table 1

Adjacent Vertices

When u and v are endpoints of an edge, they are adjacent.

Incident Edges and Vertices

If vertex v is an endpoint of an edge e, then v and e are incident.

Degree of a Vertex

The degree of a vertex v in a loopless graph G, notated as $d(v)$ is the number of edges incident with v.

Subgraph

A subgraph of a graph G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ and the assigment of endpoints to edges in H is the same as in G.

Path

A path is a simple graph whose vertices can be ordered so that two vertices are adjacent if and only if they are consecutive in the list.

Connected Graphs

A graph G is called connected if each pair of vertices in G belongs to a path. Otherwise, G is called disconnected.

Cycle

A cycle is a graph with an equal number of vertices and edges whos vertices can be placed around a circle so that two vetices are adjacent if and only if they appear consecutively along the circle.

Acylic Graphs

An acyclic graph is a graph with no cycle.

Forest

A forest is an acyclic graph.

Tree

A tree is a connected acyclic graph. The following forest has 5 subgraphs that are trees:

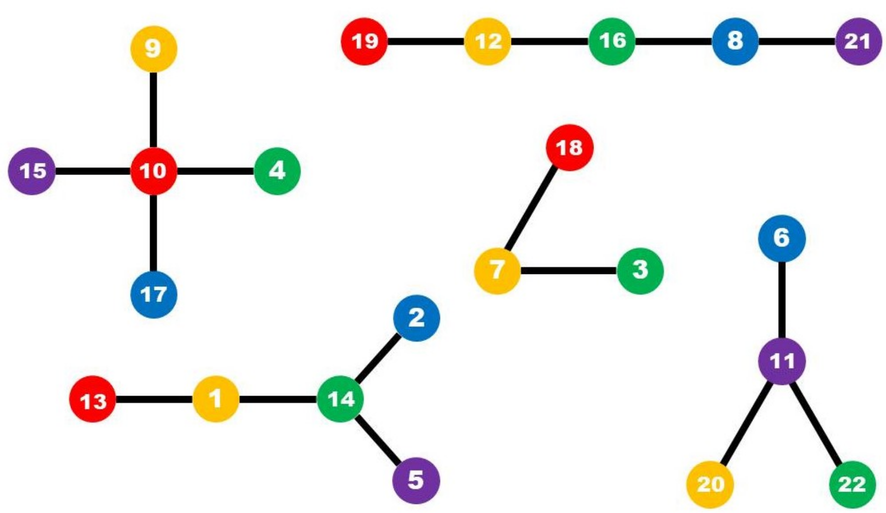


Figure 1

Leaf

A leaf is a vertex of degree 1. In the following tree, the vertices v_3, v_5, v_2 are leaves.

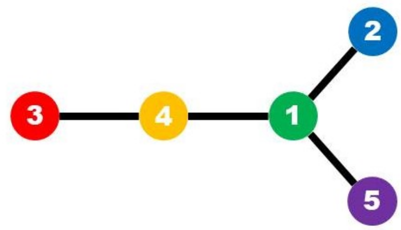


Figure 2

Rooted Trees

A rooted tree is a pair consisting of a tree and a specific vertex that has been designated as the root of the tree. Any vertex in the tree (including leaves) can be designated as the root.

Spanning Subgraph

A spanning tree of a graph G is a subgraph of G with the vertex set $V(G)$ that is a tree.

Complete Graph

A complete graph is a simple graph whose vertices are pairwise adjacent. The complete graph on n vertices is denoted by K_n.

For example, the complete graph K_5 looks like this:

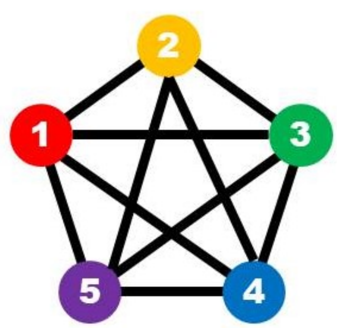


Figure 3

Notation for the Number of Trees

Let $n \in \mathbb{N}$ and let $[N] = \{1, 2, ..., n\}$.

Let T_n denote the set of all trees such that the vertex set of any tree $T \in T_n$ is $V(T) = \{v_1, v_2, ..., v_n\}$.

Let $T(G)$ denote the set of spanning trees in a graph G.

In all of the diagrams in this paper, a circle labeled with i represents the vertex v_i.

1.2 Verifying Cayley's Theorem for Small Values of n

Cayley's Theorem states that the number of labeled trees on n vertices is n^{n-2}.

$$T_n = n^{n-2}$$

It can also be interpretted as the number of spanning trees in K_n is n^{n-2}.

$$T(K_n) = n^{n-2}$$

It is relatively easy to verify Cayley's Theorem by hand for small values of n and doing so can help the reader understand and unpack the theorem statement.

Let $n = 1$.

$$T_1 = 1^{1-2} = 1^{-1} = 1$$

If $n = 1$, then Cayley's Theorem is vacuously true because there is only one tree with one vertex.

Figure 4

Let $n = 2$.

$$T_2 = 2^{2-2} = 2^0 = 1$$

If $n = 2$, Cayley's Theorem holds because there is only one way to construct a tree on two vertices.

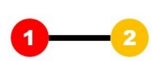


Figure 5

Let $n = 3$**.**

$$T_3 = 3^{3-2} = 3^1 = 3$$

If $n = 3$, Cayley's Theorem is easy to verify by drawing the trees.

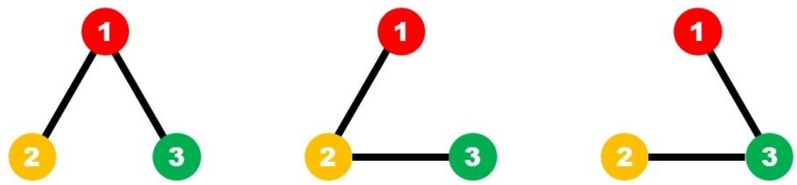


Figure 6

Let $n = 4$**.**

$$T_4 = 4^{4-2} = 4^2 = 16$$

If $n = 4$, it is harder to verify Cayley's Theorem by drawing all 16 trees, but it is doable.

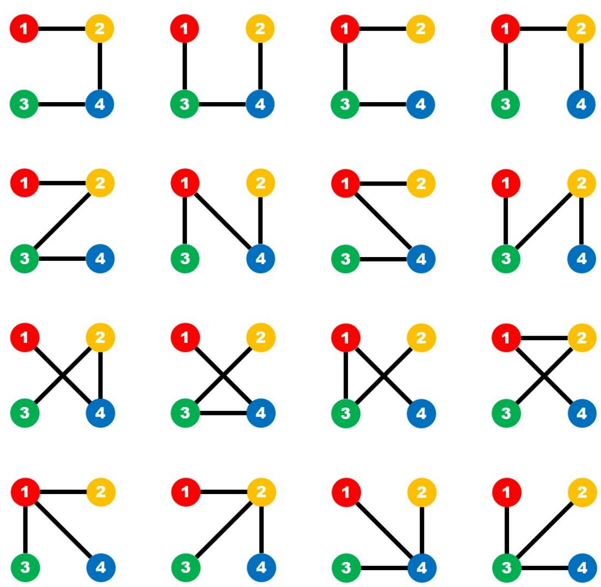


Figure 7

When $n = 4$, we begin to see how the labeled trees can be separated into isomorphism classes as unlabeled trees.

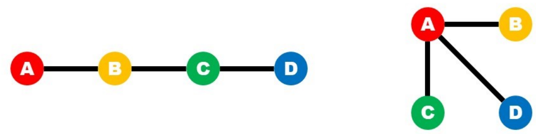


Figure 8

There are 12 trees in the ”straight line” isomorphism class because there are 4! ways to arrange 4 vertices in a straight line but the graph $1-2-3-4$ is the same graph as $4-3-2-1$ so there are $\frac{4!}{2} = 12$ 4-vertex trees that are ”straight line” trees. These trees correspond to the first 12 trees drawn above.

The ”single root” isomorphism class has 4 trees in it because the labeling of the leaves of the tree are indistinguishable and there are 4 ways to choose the root vertex from 4 vertices. These trees correspond to the last 4 trees drawn above.

Note that $12 + 4 = 16 = T(4)$.

Let $n = 5$.

$$T_5 = 5^{5-2} = 5^3 = 125$$

If $n = 5$, it is not economical to draw all 125 trees but we can draw the isomorphism classes:

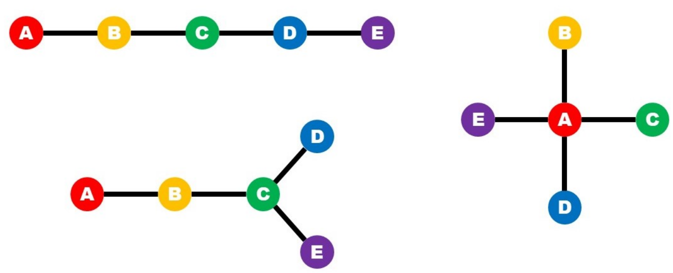


Figure 9

By the same reasoning as the 4-vertex case, there are $\frac{5!}{2} = 60$ trees in the 5-vertex ”straight line” isomorphism class. And there are 5 ways to choose the root in a ”single root” tree so there are 5 trees in the ”single root” isomorphism class.

There are $5(4)(3) = 60$ trees in the "straight line with a fork at the end" isomorphism class because there are 5 ways to choose the vertex that is placed in the A position, then 4 ways to choose the vertex that is placed in the B position, then 3 ways to choose the vertex that is placed in the C position. The last two positions (D and E) are indistinguishable, so once the first 3 positions are filled, there is only one unique way to fill positions D and E.

Note that $60 + 5 + 60 = 125 = T(5)$.

Let $n = 6$.

$$T_6 = 6^{6-2} = 6^4 = 1296$$

If $n = 6$, the isomorphism classes become even more complicated:

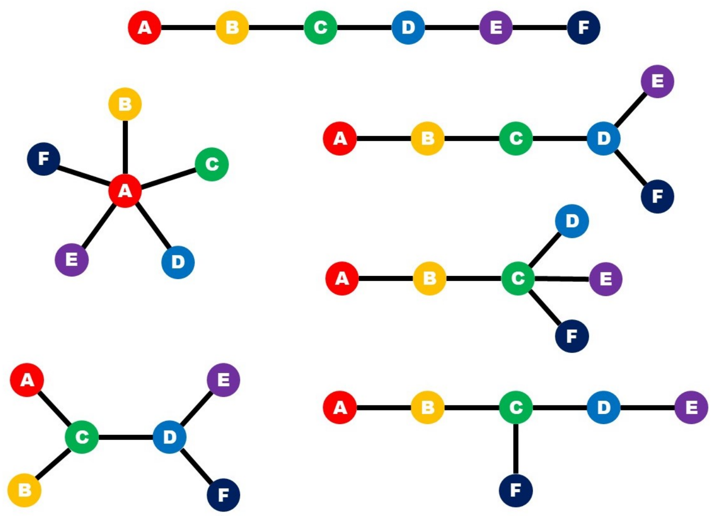


Figure 10

The "single root" isomorphism class has 6 trees in it because there are 6 ways to choose a root.

The "straight line" isomorphism class has $\frac{6!}{2} = 360$ trees.

The "straight line with a fork at the end" isomorphism class has $6(5)(4)(3) = 360$ trees.

The "straight line with a double fork at the end" isomorphism class has $6(5)(4) = 120$ trees.

The "straight line with a fork at both ends" isomorphism class has $\binom{6}{2}\binom{4}{2}\binom{2}{2} = 90$ trees.

The "straight line with a fork in the middle" isomorphism class has $\frac{6(5)(4)(3)(2)}{2} = 360$ trees.

Note that $6 + 360 + 360 + 120 + 90 + 360 = 1296 = T(6)$.

1.3 Cayley's Original Proof

When Cayley published his theorem, he actually did not prove it for all values of n.

He simply illustrated a 1-1 correspondence between the terms of the polynomial $(a+b+c+d+e+f)^4$ and the trees with 6 vertices and claimed that a similar argument could be made for all values of n.

To illustrate the idea of Cayley's proof, consider the trees on 4 vertices.

Cayley defined a 1-1 correspondence between these trees and the terms of the following polynomial by allowing the exponents in each term to describe the degree of each vertex.

$$(a + b + c + d)^2 abcd = (a^2 + 2ab + 2ac + 2ad + b^2 + 2bc + 2bd + c^2 + 2cd + d^2)abcd$$

So, if $a = v_1$, $b = v_2$, $c = v_3$, and $d = v_4$, then the term $2a^2b^2cd$ counts the following two trees because $a^2 \implies deg(v_1) = 2$ and $b^2 \implies deg(v_2) = 2$ and $c^1 \implies deg(v_3) = 1$ and $d^1 \implies deg(v_4) = 1$.

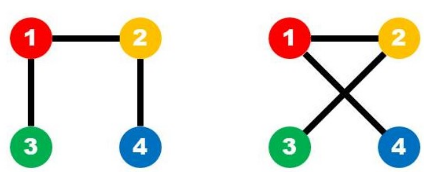


Figure 11

So, the 16 terms in $(a + b + c + d)^2 abcd$ correspond to the following trees:

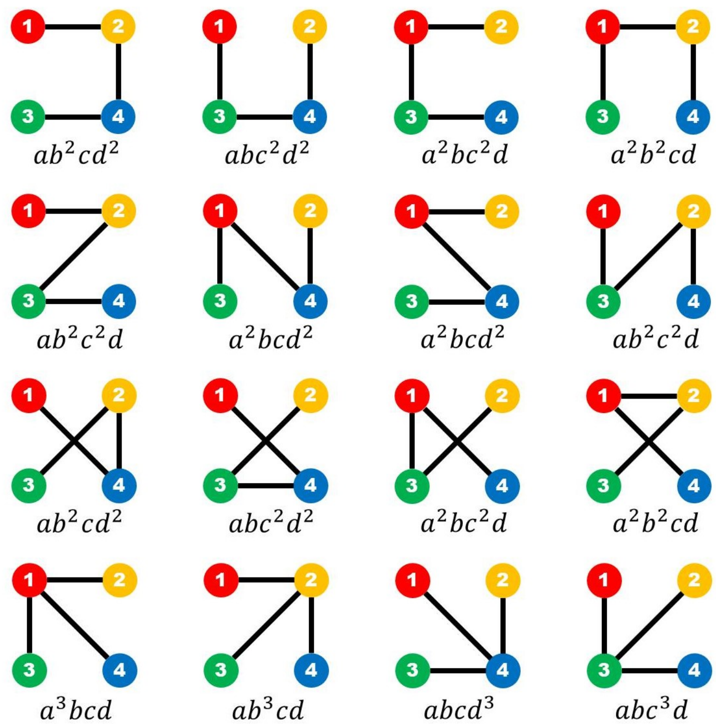


Figure 12

Notice that the structure of the polynomial terms wxy^2z^2 and $wxyz^3$ automatically separates the trees into isomorphism classes where wxy^2z^2 corresponds to the "straight line" isomorphism class and $wxyz^3$ corresponds to the "single root" isomorphism class.

For $n = 5$, the polynomial $(a+b+c+d+e)^3abcde$ has 125 terms that have the following forms:

Form	Number of Identical Terms	Number of Ways to Assign a, b, c, d, e Labels	Total
$vwxyz^4$	1	$\binom{5}{1} = 5$	$1(5) = 5$
$vwxy^2z^3$	3	$2\binom{5}{2} = 20$	$3(20) = 60$
$vwx^2y^2z^2$	6	$\binom{5}{2} = 10$	$6(10) = 60$
		Total	125

Table 2

And these three types of terms in the polynomial correspond to the following isomorphism classes:

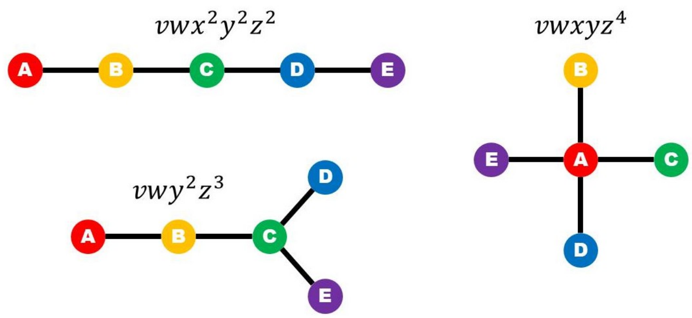


Figure 13

For $n = 6$, the polynomial $(a+b+c+d+e+f)^4abcdef$ has 1296 terms with the following forms:

Form	Number of Identical Terms	Number of Ways to Assign a, b, c, d, e, f Labels	Total
$uvwxyz^5$	1	$\binom{6}{1} = 6$	$1(6) = 6$
$uvwxy^2z^4$	4	$2\binom{6}{2} = 30$	$4(30) = 120$
$uvwxy^3z^3$	6	$\binom{6}{2} = 15$	$6(15) = 90$
$uvwx^2y^2z^3$	12	$3\binom{6}{3} = 60$	$12(60) = 720$
$uvw^2x^2y^2z^2$	24	$\binom{6}{4} = 15$	$24(15) = 360$
		Total	1296

Table 3

And these five types of terms in the polynomial correspond to the following six isomorphism classes:

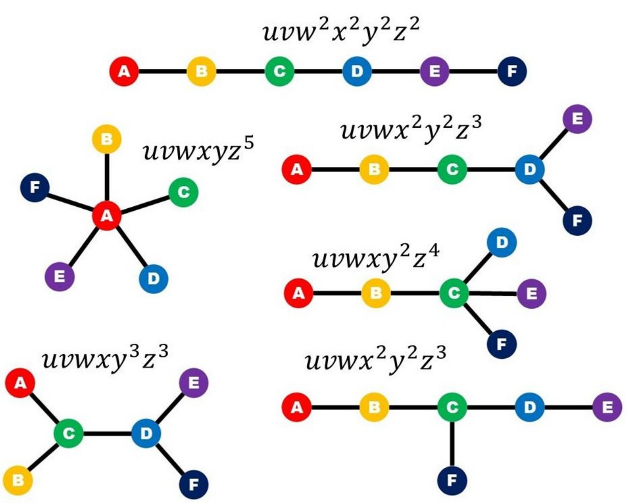


Figure 14

Notice that the "fork on the end" and "fork in the middle" isomorphism classes that we distinguished as separate isomorphism classes are counted together in Cayley's proof because their vertices have the same degrees.

1.3.1 Counting the Number of Terms in a Polynomial to a Power

Cayley's proof was incomplete because it only showed that the polynomial expansion $(a+b+c+d+e+f)^4$ and the trees on 6 vertices have a one-to-one correspondence. While it certainly seems reasonable that the same process could be extended for any n, it was not formally proven.

The first complete proof of Cayley's Theorem that included all values of n was published in 1918 by Prüfer. We will look at his proof in the Bijective Proof chapter. It is also interesting to note that Cayley's proof foreshadows the concept of generating functions which we will discuss in Chapter 5.

If Cayley's assumption was true that there is a one to one correspondence between the trees on n vertices and the terms of the expansion of $(x_1+x_2+...+x_n)^{n-2}$, then the number of trees on n vertices is n^{n-2} because of the way polynomial multiplication works.

This is because the expansion of any polynomial $(x_1+x_2+...+x_n)^r$ has n^r terms in it due to the way that polynomial multiplication works.

The expansion of the polynomial $(x_1+x_2+...+x_n)^r$ is the sum of all the ways that you can choose one term from each of the r polynomials and multiply those r terms together. Because there are n possibilities for each of the r choices, the total number of terms in $(x_1+x_2+...+x_n)^r$ is n^r.

For example, when you multiply $(x+y)^3$, you get $x^3+3x^2y+3xy^2+y^3$. But you could also think of this as the number of ways to choose either x or y three different times.

$$(x+y)^3 = xxx + xxy + xyx + xyy + yxx + yxy + yyx + yyy$$

If x and y are viewed as objects instead of variables, the terms of the polynomial expansion generate sets or lists of objects depending on whether xxy and xyx are considered to be identical or different.

1.4 Combinatorial Proofs of Binomial Coefficient Identities

The following identities can be proved algebraically but it is much more illustrative to prove them combinatorically. For these combinatorical proofs, define the following notation:

$\binom{n}{k}$ is the number of ways to choose k objects from a set of n objects.

$n!$ is the number of ways to arrange n objects in a line.

1.4.1 Factorial Expansion of the Binomial Coefficient

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Proof:

Let n, k be integers such that $n \geq k \geq 0$.

Rearrange the formula to read:

$$\binom{n}{k} k!(n-k)! = n!$$

Count the number of ways to arrange n objects in a line.

Right Hand Side: By definition.

Left Hand Side: Choose k number of objects from the set of n objects. Then arrange the k objects in a line and then arrange the remaining $n-k$ objects in a line.

The number of ways to choose k objects from the set of n objects is $\binom{n}{k}$ by definition. Also by definition, the number of ways to arrange the k objects is $k!$ and the number of ways to arrange $n-k$ objects is $(n-k)!$.

Therefore, by the multiplication principle, the number of ways to arrange n objects is $\binom{n}{k}k!(n-k)!$.

Therefore, because we counted the number of ways to arrange n objects in two different ways,

$$\binom{n}{k}k!(n-k)! = n!$$

1.4.2 Symmetry Identity

$$\binom{n}{k} = \binom{n}{n-k}$$

Proof:

Let n, k be integers such that $n \geq k \geq 0$.

Count the number of ways to choose a committee of k people from n people.

Left Hand Side: By definition.

Right Hand Side: To count the number of ways to choose a committee of k people from n people, we could choose $n-k$ NON-committee members from the group of n people and then let the remaining people be the committee of k people. Therefore, the number of ways to choose a committee of k people from n people is $\binom{n}{n-k}$.

Therefore, for $n \geq k \geq 0$,

$$\binom{n}{k} = \binom{n}{n-k}$$

1.4.3 Absorption Identity

$$\binom{r}{k} = \frac{r}{k}\binom{r-1}{k-1}$$

Proof:

Let r, k be positive integers.

Rearrange the formula to read:

$$\binom{r}{k}k = r\binom{r-1}{k-1}$$

Count the number of ways to choose a committee (with a president) of k people from r people.

Left Hand Side: Choose a committee of k people from r people. Then out of those k people choose a president. There are $\binom{r}{k}$ ways to choose a committee of k people from n people and there are k ways to choose a president within that committee. Therefore, the number of way to to choose a committee (with a president) of k people from r people is $\binom{r}{k}k$.

Right Hand Side: Choose a president from the group of r people. Then choose the remaining $k-1$ committee members from the remaining $r-1$ people. There are r ways to choose a president from r people. And there are $\binom{r-1}{k-1}$ ways to choose the remaining committee members from the remaining group of people after the president is chosen. Therefore, the number of ways to choose a committee (with a president) of k people from r people is $r\binom{r-1}{k-1}$.

Therefore, for positive integers r and k,

$$\binom{r}{k} = \frac{r}{k}\binom{r-1}{k-1}$$

1.4.4 Binomial Theorem

$$\sum_{k} \binom{r}{k} x^k y^{r-k} = (x+y)^r$$

Let $r \geq 0$ be an integer.

Because of the way polynomial multiplication works, all the terms in the expansion of $(x+y)^r$ will have degree r. Therefore, the term that contains x^k will be the same term that contains y^{r-k}.

Condition on the power of the x term being k. The coefficient of this term is $\binom{r}{k}$ because out of the r factors $(x+y)$, the x term was chosen k times and there are $\binom{r}{k}$ ways to do that.

When you sum the expression $\binom{r}{k}x^k y^{r-k}$ for all k, this will be equal to $(x+y)^r$ because it includes all of the terms from $x^0 y^r$ all the way to $x^r y^0$.

2 Inductive Proofs

Proofs by induction are useful when you want to prove a statement that involves non-negative integers. In the case of Cayley's theorem, we will use induction in multiple proofs because the number of vertices in a graph is always a natural number.

The Principle of Mathematical Induction is one of Peano's axioms that says if A is a set such that $0 \in A$ and $n \in A \implies n+1 \in A$, then $\mathbb{N} \subseteq A$.

A proof by induction makes use of this axiom by defining a statement in terms of a non-negative integer n and then defines the set A to be the set of all non-negative integers that makes the statement true.

The strategy for a proof by induction is to prove the statement is true for a base case (usually a small non-negative integer like $n = 0$, $n = 1$, $n = 2$, etc). Then for any non-negative integer k greater than your base case, you prove that the statement is true for k by assuming the statement is true for non-negative integers between your base case and k.

In a weak induction proof, you prove that if the statement is true for $k - 1$, then the statement is true for k. In a strong induction proof, you prove that if the statement is true for all non-negative numbers greater than or equal to your base case and less than k, then the statement is true for k. Strong and weak induction are logically equivalent so it doesn't matter which approach you use.

In the proofs that follow, we will define recursive formulas related to the number of trees on n vertices and then use those recursive formulas to prove the induction step.

2.1 Riordan and Renyi - Counting Labeled Forests

Our first inductive proof is found in *Proofs from the Book* [2] which summarizes the ideas presented by Riordan [14] and Rényi [13] in the 1960s.

In this proof, we first define a recursive formula for the number of labeled forests on n vertices where the forests contain k trees. Then we use induction to define a closed formula for the number of labeled forests on n vertices with k trees.

Then we conclude that the number of labeled trees on n vertices is the same as the number of labeled forests on n vertices with 1 tree in the forest because a forest with one tree is just a tree. So, we let $k = 1$ and use our closed formula to prove Cayley's Theorem.

2.1.1 Definitions and Notation

Let $n \in \mathbb{N}$ and $k \in \mathbb{N}$ such that $k \leq n$.

Labeled Forests

Let $T_{n,k}$ denote the set of labeled forests on n vertices such that there are k trees in the forest and all the vertices $\{v_1, v_2, ..., v_k\}$ are in different trees. Let $|T_{n,k}|$ denote the number of such forests.

2.1.2 Define a Recursive Relationship for $|T_{n,k}|$

To define a recursive relationship, we need to define our base cases $|T_{0,0}|$ and $|T_{0,n}|$.

$|T_{0,0}| = 1$ because there is only one way to form a forest on 0 vertices with 0 trees, namely the empty forest. And $|T_{0,n}| = 0$ for $n > 0$ because there is no way to form a forest on $n > 0$ vertices such that there are 0 trees in the forest.

Next, we can define a recursive relationship formula for $|T_{k,n}|$.

In each forest $F \in T_{n,k}$, the vertex v_1 is adjacent to i vertices where $i \in \{0, 1, ..., n-k\}$.

Therefore, we can count the total number of forests in $T_{n,k}$ by summing the numbers of the forests where v_1 is adjacent to i vertices from $i = 0$ to $n - k$.

For any forest $F \in T_{n,k}$, if we delete the vertex v_1, then we are left with a labeled forest $F \backslash v_1$ on the vertices $\{v_2, v_3, ..., v_n\}$ with $k - 1 + i$ trees because the tree that contained v_1 and the i vertices adjacent to v_1 is now i different trees.

So, for a fixed i, we can count the number of forests where v_1 is adjacent to i vertices by choosing i vertices from $\{k+1, k+2, ..., n\}$ and then choose a labeled forest $F \backslash v_1$ on the vertices $\{v_2, v_3, ..., v_n\}$ with $k-1+i$ trees such that the vertices $v_2, ..., v_k$ and the chosen i vertices are all in different trees.

Therefore,

$$|T_{n,k}| = \sum_{i=0}^{n-k} \binom{n-k}{i} |T_{n-1,k-1+i}|$$

2.1.3 Use Induction to Prove that $|T_{n,k}| = kn^{n-k-1}$.

Inductive Statement:

For any $n \in \mathbb{N}$ and $k \in \mathbb{N}$ such that $k \leq n$, then $|T_{n,k}| = kn^{n-k-1}$.

Base Cases:

Let $n = 1$ and $k \leq n$.

$$|T_{1,0}| = 0(1)^{1-0-1} = 0$$

As explained above, there is no way to form a forest on 1 vertex with 0 trees

$$|T_{1,1}| = 1(1)^{1-1-1} = 1$$

There is only one tree on one vertex with one tree in it.

Inductive Case:

Let $n \in \mathbb{N}$ and $k \in \mathbb{N}$ such that $1 \leq k \leq n$.

Assume that $|T_{n-1,m}| = m(n-1)^{(n-1)-m-1}$ for all $1 \leq m \leq n-1$.

By the recursion found above,

$$|T_{n,k}| = \sum_{i=0}^{n-k} \binom{n-k}{i} |T_{n-1,k-1+i}|$$

If $k \neq n$, then $k - 1 + i \leq n - 1$ for $0 \leq i \leq n - k$, therefore

$$|T_{n-1,k-1+i}| = (k-1+i)(n-1)^{(n-1)-(k-1+i)-1}$$

Substituting this result into the recursion gives

$$|T_{n,k}| = \sum_{i=0}^{n-k} \binom{n-k}{i} (k-1+i)(n-1)^{(n-1)-(k-1+i)-1}$$

By re-indexing the sum so that $i = n-k-j$, then

$$|T_{n,k}| = \sum_{j=n-k}^{0} \binom{n-k}{n-k-j} (k-1+n-k-j)(n-1)^{(n-1)-(k-1+n-k-j)-1}$$

By simplifying the expression we get:

$$|T_{n,k}| = \sum_{j=0}^{n-k} \binom{n-k}{n-k-j} (n-1-j)(n-1)^{j-1}$$

And applying the symmetry binomial identity 1.4.2 which states that $\binom{a}{a-b} = \binom{a}{b}$ gives:

$$|T_{n,k}| = \sum_{j=0}^{n-k} \binom{n-k}{j} (n-1-j)(n-1)^{j-1}$$

By splitting up the terms of $n-1-j$, we get:

$$|T_{n,k}| = \sum_{j=0}^{n-k} \binom{n-k}{j} (n-1)(n-1)^{j-1} - \sum_{j=0}^{n-k} \binom{n-k}{j} j(n-1)^{j-1}$$

And the fact that the $j = 0$ term does not contribute anything to the second sum and the fact that $(n-1)(n-1)^{j-1} = (n-1)^j$ allows us to simplify this to:

$$|T_{n,k}| = \sum_{j=0}^{n-k} \binom{n-k}{j} (n-1)^j - \sum_{j=1}^{n-k} \binom{n-k}{j} j(n-1)^{j-1}$$

By the binomial theorem 1.4.4, the first sum is equal to $(n-1+1)^{n-k} = n^{n-k}$,

$$|T_{n,k}| = n^{n-k} - \sum_{j=1}^{n-k} \binom{n-k}{j} j(n-1)^{j-1}$$

And because the absorption identity for binomial coefficients 1.4.3 states that $\binom{a}{b} = \frac{a}{b}\binom{a-1}{b-1}$, then

$$|T_{n,k}| = n^{n-k} - \sum_{j=1}^{n-k} \binom{n-1-k}{j-1} \frac{n-k}{j} j(n-1)^{j-1}$$

And because the sum doesn't depend on $n-k$ and the j's divide to 1, then

$$|T_{n,k}| = n^{n-k} - (n-k)\sum_{j=1}^{n-k}\binom{n-1-k}{j-1}(n-1)^{j-1}$$

By re-indexing the sum so $p = j-1$, then

$$|T_{n,k}| = n^{n-k} - (n-k)\sum_{p=0}^{n-1-k}\binom{n-1-k}{p}(n-1)^{p}$$

And by the binomial theorem 1.4.4, the sum is equal to $(n-1+1)^{n-1-k} = n^{n-1-k}$

$$|T_{n,k}| = n^{n-k} - (n-k)n^{n-k-1}$$

$$|T_{n,k}| = n^{n-k} - n^{n-k} + kn^{n-k-1}$$

$$|T_{n,k}| = kn^{n-k-1}$$

If $n = k$, then $|T_{n,n}| = 1$ because there is only one forest on n vertices with n distinct trees, namely the forest of n discrete points. This result is consistent with the calculation

$$|T_{n,n}| = n(n)^{n-n-1} = n(n)^{-1} = \frac{n}{n} = 1$$

Inductive Conclusion:

For any $n \in \mathbb{N}$ and $k \in \mathbb{N}$ such that $k \leq n$, then $|T_{n,k}| = kn^{n-k-1}$.

2.1.4 Prove Cayley's Theorem

If $k = 1$, then any labeled forest in $T_{n,1}$ is a labeled tree because there is only one tree in the forest.

Therefore, $|T_{n,1}| = |T_n|$ and the formula we found above simplifies to Cayley's Theorem

$$|T_n| = |T_{n,1}| = 1n^{n-1-1} = n^{n-2}$$

2.2 Ariannejad and Emami - Trees in Graphs with Contracted Edges

In 2014, Ariannejad and Emami published their paper called *A New Proof of Cayley's Formula for Labeled Spanning Trees* [3] where they used the idea of contracting edges in graphs to create another recursive formula for the number of trees in a graph.

Their formula is not very useful for most graphs but for a special type of complete multigraph that they define, it can be simplified to a closed formula. They use that formula to prove Cayley's theorem by applying it to a complete multigraph where all the edges have multiplicity 1.

2.2.1 Definitions and Notation

Trees in a Graph Trees with a Specified Vertex and Degree

Let G be a graph with a fixed vertex v such that $d(v) = d$ and $1 \leq i \leq d$.

Let $T_i(G(v))$ denote the set of trees where vertex v has degree i in the tree.

Deleted Vertices

Let G be a graph that contains a specified vertex v. Then we will use the notation $G - v$ to denote the graph where the vertex v and all of the edges incident to v are deleted.

Contracted Edges

Let G be a graph that contains an edge e. Then we will use the notation $G \cdot e$ to denote the graph where the edge e is contracted. In other words, the edge e is deleted and its endpoints are identified as the same vertex.

Contracting Multiple Edges of a Specified Vertex

Let G be a connected graph with a fixed vertex v with degree d. Then $\{e_1, e_2, ..., e_d\}$ is used to denote the set of all edges that are incident to v in the graph G.

And the notation $G \cdot e_{ij}$ where $1 \leq i \neq j \leq d$ is used to denote the graph created when all the edges incident to v in G are deleted except e_i and e_j and the edges e_i and e_j are both contracted.

Similarly, the notation $G \cdot e_{ijk}$ where $1 \leq i \neq j \neq\leq d$ is used to denote the graph created when all the edges incident to v in G are deleted except e_i, e_j, and e_k and all those edges are contracted.

And $G \cdot e_{12...d}$ is used to denote the graph created when none of the edges incident to v in G are deleted and all of the edges e_1, e_2,..., e_d are contracted.

Complete Multigraph with Specified Vertex

For any $n \in \mathbb{N}$ and $1 \leq r \leq n$, let the notation $K_n^{(r)}$ denote the complete multigraph with n vertices where there is one specific vertex v whose edges all have multiplicity r and all edges that do not have v as an endpoint have multiplicity 1.

2.2.2 Define a Recursive Formula

Define a Recursive Formula for the Number of Trees in any Graph

Let G be a connected graph with a fixed vertex v with degree d.

Each spanning tree in G is in exactly one of the following categories:

- Trees where v has degree 1
- Trees where v has degree 2
- ...
- Trees where v has degree d

Therefore, by conditioning on the degree of v, we get that

$$|T(G)| = |T_1(G(v))| + |T_2(G(v))| + ... + |T_d(G(v))|$$

We can calculate the values of each of these terms as follows:

1. If v has degree 1 in a tree in G, then that means that v is a leaf of the tree. And because the degree of v is d in the graph G, then there are d ways to attach a leaf to any tree in the graph $G - v$ to create a tree in G where v has degree 1.

$$|T_1(G(v))| = d|T(G - v)|$$

2. If v has degree 2 in a tree in G, then the two edges incident to v in the tree can be labeled as e_i and e_j in G and if those edges are contracted in G and the remaining edges incident to v in G are deleted then this tree is mapped to a specific tree in the graph $G \cdot e_{ij}$.

$$|T_2(G(v))| = \sum_{1 \leq i \neq j \leq d} |T(G \cdot e_{ij})|$$

3. By similar reasoning,

$$|T_3(G(v))| = \sum_{1 \leq i \neq j \neq k \leq d} |T(G \cdot e_{ijk})|$$

4. And this pattern can be computed recursively until

$$|T_d(G(v))| = |T(G \cdot e_{12 \ldots d})|$$

Therefore,

$$|T(G)| = d|T(G-v)| + \sum_{i \neq j} |T(G \cdot e_{ij})| + \sum_{i \neq j \neq k} |T(G \cdot e_{ijk})| + \ldots + \sum_{m_1 \neq m_2 \neq \ldots \neq m_{d-1}} |T(G \cdot e_{m_1 m_2 \ldots m_{d-1}})| + |T(G \cdot e_{12 \ldots d})|$$

This result is not easy to calculate for most graphs, but it is relatively easy to calculate for complete graphs and complete multigraphs.

Apply the formula to $K_n^{(r)}$ to calculate $|T(K_n^{(r)})|$

Let $G = K_n^{(r)}$

Let $v \in K_n^{(r)}$ be the specified vertex whose edges all have multiplicity r.

Then $K_n^{(r)} \cdot e_{ij}$ is a complete multigraph $K_{n-2}^{(2)}$ because contracting the edges e_i and e_j and deleting all the remaining edges attached to v means that the merged vertex $v = v_i = v_j$ has two edges incident to all of the other vertices, namely the edges that originally made v_i and v_j adjacent to all of the other vertices. And there are $n-2$ vertices in the contracted graph because v, v_i, and v_j were all contracted to a single point.

Because there are $\binom{n-1}{2}$ ways to choose two vertices that are not v and there are r ways to choose the each of the edges e_i and e_j that will make v adjacent to these vertices, then the second term in the recursive formula found above can be written as:

$$\sum_{i \neq j} |T(K_n^{(r)} \cdot e_{ij})| = \binom{n-1}{2} r^2 |T(K_{n-2}^{(2)})|$$

By similar reasoning,

$$\sum_{m_1 \neq \ldots \neq m_p} |T(K_n^{(r)} \cdot e_{m_1 \ldots m_p})| = \binom{n-1}{p} r^p |T(K_{n-p}^{(p)})|$$

Therefore, substituting these calculations into the recursive formula gives:

$$|T(K_n^{(r)})| = d|T(K_n^{(r)} - v)| + \binom{n-1}{2} r^2 |T(K_{n-2}^{(2)})| + \ldots + \binom{n-1}{n-1} r^{n-1} |T(K_{n-(n-1)}^{(n-1)})|$$

Because $K_n^{(r)}$ is a complete multigraph, then v is adjacent to to all the other $n-1$ vertices and because each of those edges has multiplicity r, then the degree of v is $d = r(n-1) = \binom{n-1}{1} r^1$.

And $K_n^{(r)} - v$ is the complete graph K_{n-1} which is equivalent to the complete multigraph $K_{n-1}^{(1)}$.

Therefore,

$$|T(K_n^{(r)})| = \sum_{i=1}^{n-1} \binom{n-1}{i} r^i |T(K_{n-i}^{(i)})|$$

2.2.3 Use induction to find a closed formula for $|T(K_n^{(r)})|$

Inductive Statement: For all $r \geq 1$ and $n \geq 1$, then $|T(K_n^{(r)})| = r(r+n-1)^{n-2}$.

Base Cases:

Let $n = 1$.

It is trivially true that $|T(K_1^{(r)})| = r(r+1-1)^{1-2}$ because $r(r+1-1)^{1-2} = r(r)^{-1} = 1$ and the complete multigraph $K_1^{(r)}$ is just a single point and therefore there is only one tree in it.

Let $n = 2$.

$$|T(K_1^{(r)})| = r(r+n-1)^{n-2} = r(r+1)^0 = r$$

For all $r \geq 1$, the complete multigraph $K_2^{(r)}$ is two vertices with r edges between them. There are r trees in the multigraph because there are r ways to choose a single edge that makes the vertices adjacent to each other.

Inductive Case:

Let $r \geq 1$.

Assume that for all $t \leq k$,

$$|T(K_t^{(r)})| = r(r+t-1)^{t-2}$$

We want to show that

$$|T(K_{k+1}^{(r)})| = r(r+k+1-1)^{k+1-2}$$

Or in simplified terms, that

$$|T(K_{k+1}^{(r)})| = r(r+k)^{k+1}$$

By the recursive formula we found above,

$$|T(K_{k+1}^{(r)})| = \sum_{i=1}^{k+1-1} \binom{k+1-1}{i} r^i |T(K_{k+1-i}^{(i)})| = \sum_{i=1}^{k} \binom{k}{i} r^i |T(K_{k+1-i}^{(i)})|$$

Because $k+1-i \leq k$ for all $1 \leq i \leq k$, then by assumption

$$|T(K_{k+1-i}^{(i)})| = i(i+k+1-i-1)^{k+1-i-2} = i(k)^{k-i-1}$$

Therefore,

$$|T(K_{k+1}^{(r)})| = \sum_{i=1}^{k} \binom{k}{i} r^i i(k)^{k-i-1}$$

Because the absorption identity for binomial coefficients states that $\binom{k}{i} = \frac{k}{i}\binom{k-1}{i-1}$, then

$$|T(K_{k+1}^{(r)})| = \sum_{i=1}^{k} \frac{k}{i} \binom{k-1}{i-1} r^i i(k)^{k-i-1}$$

Cancelling out the i's in the numerator and denominator and combining the k factors gives

$$|T(K_{k+1}^{(r)})| = \sum_{i=1}^{k} \binom{k-1}{i-1} r^i (k)^{k-i}$$

Finally, we will re-index the sum with $j = i - 1$.

$$|T(K_{k+1}^{(r)})| = \sum_{j=0}^{k-1} \binom{k-1}{j} r^{j+1} (k)^{k-(j+1)} = r \sum_{j=0}^{k-1} \binom{k-1}{j} r^{j} (k)^{k-1-j}$$

Therefore, by the binomial theorem,

$$|T(K_{k+1}^{(r)})| = r(r+k)^{k-1}$$

And this is equivalent to

$$|T(K_{k+1}^{(r)})| = r(r+k+1-1)^{k+1-2}$$

Inductive Conclusion: For all $r \geq 1$ and $n \geq 1$, then $|T(K_n^{(r)})| = r(r+n-1)^{n-2}$.

2.2.4 Prove Cayley's Theorem

If $r = 1$ and $n \geq 1$, the specified vertex v only has 1 edge incident to each of the other vertices.

Therefore, there are no multiple edges in the complete multigraph so

$$K_n^{(1)} = K_n$$

Therefore, the result we found via induction proves Cayley's Theorem

$$|T(K_n)| = |T(K_n^{(1)})| = 1(1+n-1)^{n-2} = n^{n-2}$$

3 Bijective Proofs

Bijective proofs are useful when you are trying to count a set that is hard to enumerate by itself but there is another set of the same size that is easy to count.

There is a theorem that says if two of the following conditions hold for a function $f : A \to B$, then the third condition also holds:

1. $|A| = |B|$.

2. f is onto

3. f is one-to-one

A bijection is defined as a one-to-one and onto function, so our bijective proofs will only use the fact that 2 and 3 imply 1. I will prove that part of the theorem and leave the other two parts of the proof to the reader.

Proof:

Suppose that f is a one-to-one and onto function.

By the definition of one-to-one, for any $x, y \in A$ such that $f(x) = f(y)$, then $x = y$.

Therefore, B has at least as many elements as A does.

$$|A| \leq |B|$$

By the definition of onto, for any $b \in B$, there exists an $a \in A$ such that $f(a) = b$.

Therefore, because functions require every element in the domain A to be mapped to exactly one element in B, then B cannot have more elements than A does.

$$|B| \leq |A|$$

These two results show that $|A| = |B|$.

The strategy of a bijective proof is to find a bijection between the set that is easy to count and the set you want to count. By the theorem we just proved, this will prove that the number of objects in the easy-to-count set is the same number as the number of objects in the set you want to count.

Even though it was incomplete, Cayley's original proof is an example of a bijective proof because he established a bijection between the set of terms in the expansion of $(a + b + c + d + e + f)^4$ and the trees on 6 vertices. Because the size of the first set was 6^4 the one to one correspondence implied that the number of of trees on 6 vertices was also 6^4.

In the proofs that follow we will establish bijections between sets of sequences of numbers and sets of functions (which are both easy to count) and sets of labeled trees (which are hard to count) to prove Cayley's Theorem.

3.1 Prüfer - Sequences of Length $n-2$ and Labeled Trees

In 1918, Prüfer [12] published the first complete proof of Cayley's Theorem. His proof established a bijection between the set of sequences of length $n-2$ and the set of labeled trees.

Because Prüfer's original paper was written German, the following proof is based on the summary of his proof found in Moon's book *Counting Labelled Trees* [9].

3.1.1 Definitions

Sequences of Length $n-2$

Let S_n be the set of sequences of length $n-2$ such that the terms of the sequences are in $[N]$.

3.1.2 Sizes of the Sets

For any $s \in S_n$, there are $n-2$ terms in the sequence and n ways to choose each of those terms.

Therefore,

$$|S_n| = n^{n-2}$$

We want to show that $|T_n| = n^{n-2}$ so we need to find a bijection between T_n and S_n.

3.1.3 Defining the Bijection

Let $t \in T_n$.

Every tree has at least one leaf.

Let $L_0 = \{\text{leaves of } t\}$ and define $l_1 = min(L_0)$ where the minimum of the set of vertices is defined as the vertex with the minimum index. Because l_1 is a leaf, it is only adjacent to one other vertex. Let s_1 be the index of this vertex.

Let $L_1 = \{\text{leaves of } t - \{l_1\}\}$ and define $l_2 = min(L_1)$. Let s_2 be the index of the only vertex that is adjacent to l_2.

Recursively, for $1 \leq i \leq n-3$, let $L_i = \{\text{leaves of } t - \{l_1, ..., l_i\}\}$ and define $l_{i+1} = min(L_i)$. Then define s_{i+1} to be the index of the vertex adjacent to l_{i+1}.

Define the sequence $s = \{s_1, s_2, ..., s_{n-2}\}$.

This function maps any tree to a unique sequence of length $n-2$, so it is a 1-1 function.

Example:

Let $t \in T_7$ be

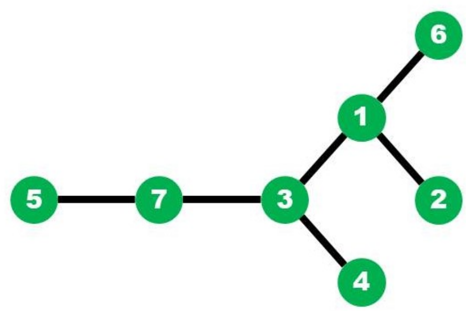


Figure 15

Then

$$\begin{aligned} L_0 = \{v_2, v_4, v_5, v_6\} \quad & l_1 = v_2 \quad s_1 = 1 \\ L_1 = \{v_4, v_5, v_6\} \quad & l_2 = v_4 \quad s_2 = 3 \\ L_2 = \{v_5, v_6\} \quad & l_3 = v_5 \quad s_3 = 7 \\ L_3 = \{v_6, v_7\} \quad & l_4 = v_6 \quad s_4 = 1 \\ L_4 = \{v_1, v_7\} \quad & l_5 = v_1 \quad s_5 = 3 \end{aligned}$$

Therefore, the sequence that corresponds to this tree is $s = \{1, 3, 7, 1, 3\}$

And this sequence can be converted back into the original tree by the following process:

3.1.4 Reversing the Bijection

Let $s = \{s_1, s_2, ..., s_{n-2}\} \in S_n$

Let the set $A_0 = [N]$ and let $B_0 = s$. Define $a_1 = min(A_0 - B_0)$. This minimum exists because the sequence s contains natural numbers between 1 and n and since there are only $n-2$ terms in the sequence, there are at least 2 numbers between 1 and n that are not in the sequence. There may be more than 2 if the sequence has duplicate terms.

Let the set $A_1 = [N] - \{a_1\}$ and let $B_1 = s - \{s_1\}$. Then define $a_2 = min(A_1 - B_1)$.

Recursively, for $1 \leq i \leq n-3$, let $A_i = N - \{a_1, ..., a_i\}$ and $B_i = s - \{s_1, ..., s_i\}$ and define $a_{i+1} = min(A_i - B_i)$.

This process defines $a_1, ..., a_{n-2}$ uniquely.

If we were to iterate this one more time, then $A_{n-2} = \{a, b : a, b \in \mathbb{N} \text{ and } a \neq b\}$ and $B_{n-2} = \emptyset$.

Define a tree T with the vertex set $V(T) = \{v_1, v_2, ..., v_n\}$ and the edge set $E(T)$ that is defined such that all vertices v_{a_i} and v_{s_i} are adjacent for all i and also that the vertices v_a and v_b are adjacent in T and no other pairs of vertices are adjacent.

Example:

Consider the sequence we found in our last example: $s = \{1, 3, 7, 1, 3\} \in S_7$.

$$A_0 = \{1,2,3,4,5,6,7\} \quad B_0 = \{1,3,7,1,3\} \quad A_0 - B_0 = \{2,4,5,6\} \quad a_1 = 2 \quad s_1 = 1$$

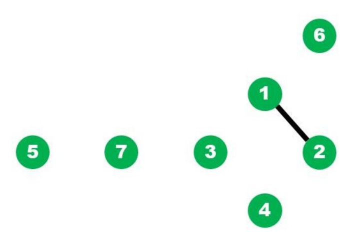


Figure 16

$A_1 = \{1,3,4,5,6,7\} \quad B_1 = \{3,7,1,3\} \quad A_1 - B_1 = \{4,5,6\} \quad a_2 = 4 \quad s_2 = 3$

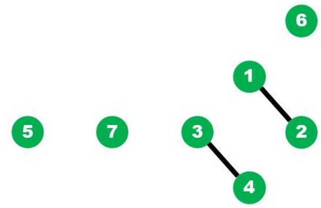


Figure 17

$A_2 = \{1,3,5,6,7\} \quad B_2 = \{7,1,3\} \quad A_2 - B_2 = \{5,6\} \quad a_3 = 5 \quad s_3 = 7$

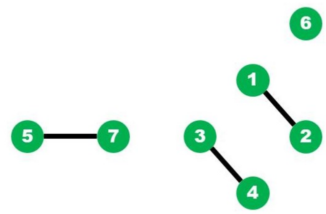


Figure 18

$A_3 = \{1,3,6,7\} \quad B_3 = \{1,3\} \quad A_3 - B_3 = \{6,7\} \quad a_4 = 6 \quad s_4 = 1$

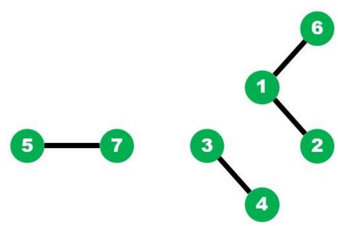


Figure 19

$A_4 = \{1,3,7\} \quad B_4 = \{3\} \quad A_4 - B_4 = \{1,7\} \quad a_5 = 1 \quad s_5 = 3$

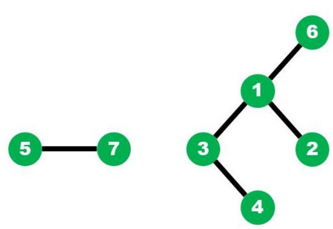


Figure 20

If we were to recursively define A_5 and B_5, then

$$A_5 = \{3, 7\} \quad B_4 = \{\}$$

Therefore, let $a = 3$ and $b = 7$.

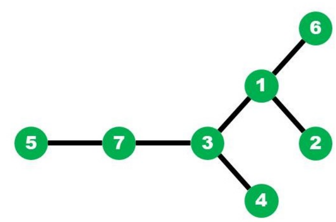


Figure 21

As we can see, this process does indeed generate the same tree that originally generated the sequence $\{1, 3, 7, 1, 3\}$.

3.1.5 Proving it is a Bijection

We know that the function defined from $T_n \to S_n$ is 1-1 so we only need to show that it is also onto or in other words that every sequence generates a tree.

Based on the example above, it seems like it should work but we still need to formally prove it to be so.

Inductive Claim: For any $n \geq 3$, a sequence in S_n will generate a tree on n vertices by the process given above.

Base Case: Let $n = 3$.

If $n = 3$ then $S_n = \{\{1\}, \{2\}, \{3\}\}$.

For each sequence in S_n, the process given above will yield the following trees respectively:

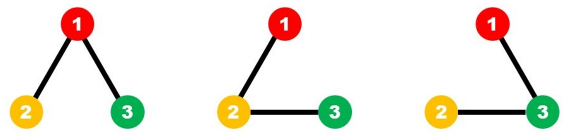


Figure 22

Inductive Step: Assume that all sequences in S_{k-1} generate trees on $k-1$ vertices and prove that a sequence in S_k generates a tree.

Let $\{s_1, s_2, ..., s_{k-2}\} \in S_k$ and let $m = min([k] - \{s_1, s_2, ..., s_{k-2}\})$.

Define a new sequence $\{b_2, ..., b_{k-2}\}$ such that $b_i = s_i - 1$ if $s_i > m$ and $b_i = s_i$ if $s_i \leq m$.

Because this new sequence has length $(k-1) - 2$ and all of its elements are less than or equal to $k-1$, then $\{b_2, ..., b_{k-2}\} \in S_{k-1}$ which implies by the inductive assumption that it generates a tree on $k-1$ vertices by the process listed above.

Generate this tree $T \in T_{k-1}$ and then relabel all of its vertices that have an index $i \geq m$ with $i+1$. Leave all the vertices with indices $i < m$ with the same index. This new tree T' now has vertices labeled $\{1, 2, ..., m-1, m+1, ...n\}$.

Add a new vertex v_m to the graph and attach it to the vertex indexed with s_1.

The resulting tree T'' is the same tree on n vertices that the reverse bijection process would generate from the sequence $\{s_1, s_2, ..., s_{n-2}\}$. Therefore, for any $n \geq 3$, a sequence in S_n will generate a tree on n vertices by the process given above.

3.1.6 Conclusion

Because there exists a bijection between T_n and S_n, then $|T_n| = |S_n| = n^{n-2}$.

3.2 Joyal - Functions $[N] \to [N]$ and Labeled Trees with 2 Special Vertices

In 1981, André Joyal [7] published a bijective proof of Cayley's theorem which establishes a bijection between the set of functions from $[N] \to [N]$ and the set of labled trees with two vertices designated as the beginning and ending of a path within the tree.

Joyal's approach to proving Cayley's Theorem is one of those highlighted in *Proofs from the Book* [2] and the following proof is based on the summary of Joyal's proof found on page 202 of that book.

3.2.1 Definitions

Let $E_n = \{(t, a, b) : t \in T_n, \quad a \in [N], \quad b \in [N]\}$ where t is a tree on n vertices, a is the index of a vertex that has been designated as the "beginning" and b is the index of a vertex that has been designated as the "ending" where it is possible for v_a and v_b to be the same vertex.

Let $F_n = \{f : [N] \to [N]\}$ be the set of functions that maps $[N]$ to itself.

3.2.2 Sizes of the Sets

Because v_a and v_b can be the same vertex, there are n ways to choose $a \in [N]$ and n ways to choose $b \in [N]$. Therefore,

$$|E_n| = n^2 |T_n|$$

When a function $f \in F_n$ is defined, there are n choices for $f(x)$ for each $x \in [N]$.

Therefore,

$$|F_n| = n^n$$

3.2.3 Defining the Bijection

Let $f \in F_n$.

This function can be converted into a directed graph by creating edges $v_x \to v_{f(x)}$ for all $x \in [N]$.

This directed graph may or may not be connected. Because each component of the graph has an equal number of edges and vertices and each vertex has an out-degree of exactly 1, then we know that each component has exactly 1 directed cycle.

Let M be the set that contains all indices of the vertices that are in the directed cycle of any of the components in the directed graph.

Order the elements of $M = \{x_1, x_2, ..., x_{k\}}$ from least to greatest and let $a = f(x_1)$ and $b = f(x_k)$.

Then build a tree on the vertices $v_1, v_2, ..., v_n$ such that $v_{f(x_i)}$ and $v_{f(x_{i+1})}$ are adjacent for all $1 \leq i \leq k-1$ and the remaining vertices with indices in $[N] - M$ are attached to the tree by the same edges as in the directed graph.

It is clear that every function will map to a unique ordered triple (t, a, b).

Therefore, this process that maps $F_n \to E_n$ is a one-to-one function.

Example:

Let $f : [9] \to [9]$ be defined by

x	1	2	3	4	5	6	7	8	9
f(x)	3	9	6	4	5	3	5	5	8

By mapping $x \to f(x)$ we get the following directed graph:

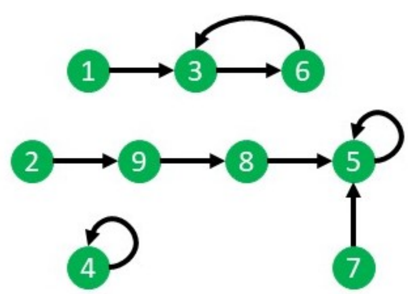


Figure 23

Therefore, the vertices in directed cycles are $\{3, 6, 5, 4\}$ and ordering this gives us the ordered set

$$M = \{3, 4, 5, 6\}$$

Therefore, $a = f(min(M)) = f(3) = 6$ and $b = f(max(M)) = f(6) = 3$ and

$$f(M) = \{6, 4, 5, 3\}$$

This generates the following tree t:

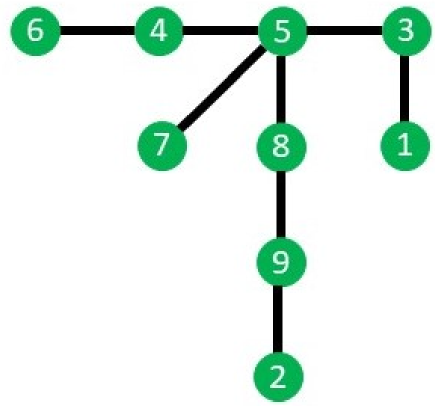


Figure 24

Therefore, the function f maps to $(t, 6, 3)$.

3.2.4 Reversing the Bijection

Let $(t, a, b) \in E_n$.

Because t is a tree, there exists a path from $v_a \to v_b$.

Let the indices of vertices in this path be labeled

$$a = y_1 \to y_2 \to ... \to y_k = b$$

And define $M = \{y_1, y_2, ..., y_k\}$.

Order the elements of M from least to greatest by their original indices

$$M = \{x_1, x_2, ..., x_k\}$$

Define a function from $f : M \to M$ such that $f(x_1) = y_1$.

For the remaining vertices $z_i \in [N] - M$, extend the function $f : [N] \to [N]$ by finding the path from z_i to the closest x_j and letting $f(z_i)$ be the first vertex in that path.

It is clear that every ordered triple $(t, a, b) \in E_n$ can be mapped to a unique function, so the process that maps $F_n \to E_n$ is an onto function.

Example:

Let $(t, 6, 3) \in E_9$ where t is the tree generated in the previous example:

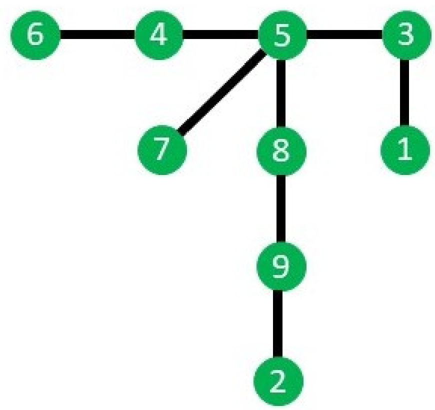


Figure 25

The "beginning" index 6 and the "ending" index 3 define the path

$$v_6 \to v_4 \to v_5 \to v_3$$

The remaining vertices $\{v_1, v_2, v_7, v_8, v_9\}$ have three leaves, namely $\{v_1, v_2, v_7\}$ that define three leaf paths.

$$v_1 \to v_3$$

$$v_2 \to v_9 \to v_8$$

$$v_7 \to v_5$$

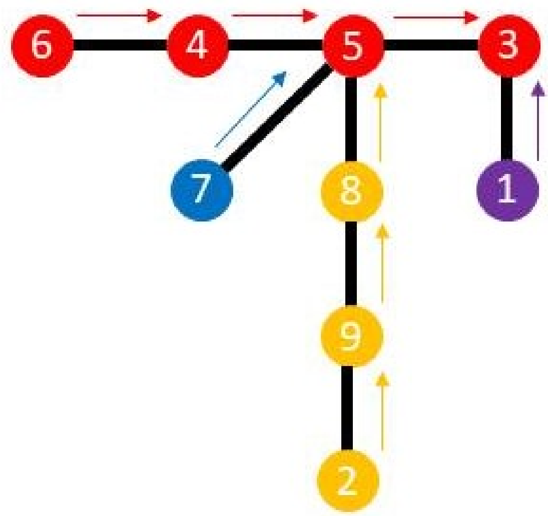


Figure 26

So, $M = \{y_1 = 6, y_2 = 4, y_3 = 5, y_4 = 3\}$.

Ordering these indices gives $M = \{x_1 = 3, x_2 = 4, x_3 = 5, x_4 = 6\}$.

Defining the function $f : M \to M$ such that $f(x_i) = y_i$ gives:

x	3	4	5	6
f(x)	6	4	5	3

By using the leaf paths, I can extend this function to be $f : [9] \to [9]$ defined by

x	1	2	3	4	5	6	7	8	9
f(x)	3	9	6	4	5	3	5	5	8

Notice that this is the same function that generated $(t, 6, 3)$ in the first place.

3.2.5 Proving it is a Bijection

Because the process that maps $F_n \to E_n$ is a one-to-one and onto function, then it is a bijection.

3.2.6 Conclusion

Because there is a bijection between E_n and F_n, then

$$|E_n| = |F_n|$$

Substituting in the sizes of the sets we found above gives

$$n^2|T_n| = n^n$$

And dividing by n^2 gives us the result of Cayley's Theorem

$$|T_n| = \frac{n^n}{n^2} = n^{n-2}$$

3.3 Abu-Sbeih - Trees in Special Bipartite Graphs and Labeled Trees

Sometimes, it is possible to count objects by constructing them and directly counting the number of ways to construct them. In the case of spanning trees in a complete graph K_n, this is not easy to do.

However, in 1988, Abu-Sbeih [1] published a simple proof of how to directly count the number of trees in a complete bipartite graph $K_{m,n}$.

Then he applied the result to a special type of complete bipartite graph and used it to prove Cayley's Theorem with an extremely simple bijection between the trees in that bipartite graph and the trees in a complete graph.

3.3.1 Definitions and Notation

Directed Trees with Specified Vertex

Let G be a connected graph with at least one vertex. For any tree $T \in T(G)$ with a specified vertex v, let $T' \in D(G; v)$ denote the directed tree that is formed when all of the edges in T are directed towards v. This is well defined because there is a unique path between any two vertices in a tree.

In and Out Degree of a Vertex

For a vertex v in a directed graph, let $d^+(v)$ denote the out-degree of v and $d^-(v)$ the in-degree.

Complete Bipartite Graph

A complete bipartite graph between two sets A and B is a simple graph such that a and b are adjacent for all $a \in A$ and $b \in B$ and no pair of vertices in A are adjacent nor are any pair of vertices in B.

Let $K_{m,n}$ denote the complete bipartite graph between the vertex sets $A = \{a_1, a_2, ..., a_m\}$ and

$B = \{b_1, b_2, ..., b_n\}$ where $m \geq 2$ and $n \geq 2$.

3.3.2 Prove that $|T(K_{m,n})| = m^{n-1}n^{m-1}$

There is a 1 to 1 correspondence between T and T' for every $T \in T(G)$ and $T' \in D(G, v)$, so

$$|T(G)| = |D(G, v)|$$

To prove that $T(K_{m,n})| = m^{n-1}n^{m-1}$, we will show that

$$|D(K_{m,n}, v)| = m^{n-1}n^{m-1}$$

Characterize Trees in $D(K_{m,n}; b_1)$ by the out-degree of the vertices

Consider a directed bipartite tree $T' \in D(K_{m,n}; b_1)$. Because all the edges are directed towards b_1 then $d^+(b_1) = 0$ and $d^-(v) = 1$ for all the other vertices in $K_{m,n}$.

And if there is a directed connected bipartite graph $G \in K_{m,n}$ such that $d^+(b_1) = 0$ and $d^-(v) = 1$ for all the other vertices then for any v_1 adjacent to b_1, the edge associated with them them must be directed towards b_1. Therefore, all the remaining edges incident to v must be directed towards v.

By similar reasoning, for any $w \neq b_1$ adjacent to v, the edge that makes v and w adjacent is directed towards v and the remaining vertices adjacent to w are directed towards w.

By iterating this process for all the vertices in G, we find that $G \in D(K_{m,n}; b_1)$ because it forms a directed tree where all the edges are directed towards b_1.

Therefore, a directed, connected bipartite graph $G \in D(K_{m,n}, b_1)$ if and only $d^+(b_1) = 0$ and $d^-(v) = 1$ for all $v \neq b_1 \in K_{m,n}$.

Construct all trees $T' \in D(K_{m,n}; b_1)$

Let $B' = \{b_2, b_3, ..., b_n\}$.

Let G denote any directed bipartite graph with vertex sets A and B' such that $d^+(b) = 1$ for all $b \in B'$ and $d^+(a) = 0$ for all $a \in A$.

Because B' has $n-1$ elements and each of these vertices have a single edge directed away from them into the set A which has m elements in it, there are m^{n-1} ways to choose the edges $B' \to A$. And because $d^+(a) = 0$ for all $a \in A$ then there are no edges from $A \to B'$.

Therefore, there are m^{n-1} ways to construct G and G has m connected components.

Specifically, each of the m connected components contains exactly one $a \in A$.

Let $0 \leq t \leq m-1$.

Let H_t denote the graph formed by adding t edges to G in the following way:

1. Choose a $b \in B'$ and then choose an $a \in A$ that is not in the same component as b.
2. Construct an edge from $a \to b$.
3. Repeat t times.

There are $(n-1)(m-1)$ ways to form the first edge because there are $n-1$ vertices in B' to choose b from and $m-1$ connected components that do not contain b. After the first edge is constructed, the graph now has $m-1$ connected components because two of the components become one component when you add the edge.

Therefore, there are $(n-1)(m-2)$ ways to construct the second edge because there are still $n-1$ vertices in B' to choose b from but now there are $m-2$ connected components that do not contain b.

By iterating this process t times, we find that the number of graphs H_t that can be created from a graph G by adding the edges in a specific order is:

$$(n-1)(m-1)(n-1)(m-2)...(n-1)(m-t) = (n-1)^t \frac{(m-1)!}{(m-t-1)!}$$

The order we add the edges doesn't affect the end result H_t, so the number of H_t that can be created from G by the process above is:

$$(n-1)^t \frac{(m-1)!}{(m-t-1)!t!} = \binom{m-1}{t}(n-1)^t$$

Any graph H_t constructed in this way has $m-t$ components where each component contains one specific vertex $a \in A$ with out-degree 0 and all the remaining vertices have out-degree 1.

Creating an edge $a \rightarrow b_1$ from the specific vertices a to the vertex b_1 creates a connected bipartite graph in $K_{m,n}$ where $d^+(b_1) = 0$ and $d^-(v) = 1$ for all the other vertices.

As shown above, this means that the resulting graph is a $T' \in D(K_{m,n}, b_1)$.

When t is ranged from 0 to $m-1$ for all possible graphs G, this process will generate all $T' \in D(K_{m,n}, b_1)$.

Therefore,

$$|D(K_{m,n}; b_1)| = m^{n-1} \sum_{t=0}^{m-1} \binom{m-1}{t}(n-t)^t = m^{n-1}n^{m-1}$$

3.3.3 Prove Cayley's Theorem

Let $B = \{x_1, x_2, ..., x_n\}$ where $B' = \{x_2, x_3, ..., x_n\}$.

Then let $A = \{x'_2, x'_3, ..., x'_n\}$ be a copy of B'.

The complete bipartite graph between A and B' is $K_{n-1,n}$.

Instead of allowing G to be any directed bipartite graph with vertex sets A and B' such that $d^+(b) = 1$ for all $b \in B'$ and $d^+(a) = 0$ for all $a \in A$, let it be the specific graph where $x_i \rightarrow x'_i$ for all $2 \leq i \leq n$.

Obviously, there is only 1 way to construct such a graph G and as shown above, the number of ways to construct H_t for this G for all $0 \leq t \leq m-1$ where $m = |A|$ is:

$$\sum_{t=0}^{m-1} \binom{m-1}{t} (n-t)^t = n^m - 1$$

And because $m = |A| = n-1$, then this number can be simplified to $n^{(n-1)-1} = n^{n-2}$.

The trees formed by connecting the components of H_t to x_1 are not trees in K_n because they have $n + n - 1 = 2n - 1$ vertices namely $\{x_1, x_2, x_2^*, ..., x_n, x_n^*\}$.

However, because every one of the trees has adjacent vertices $x_i \rightarrow x_i^*$, the trees can be converted to trees in K_n by deleting all of the edges (x_i, x_i^*) and merging x_i and x_i^* into a single vertex labeled v_i.

Every tree in K_n can be constructed with this process, therefore,

$$|T(K_n)| = n^{n-2}$$

4 Double Counting

Double counting proofs are useful when the thing you are trying to count can be used to count something else.

The strategy of a double counting proof is count the same thing twice in two different ways and then set the results equal to each other.

In the proofs that follow we will define mathematical objects that can be counted using labeled trees and then count those objects in two different ways.

4.1 Pitman - Counting Refining Sequences of Rooted Forests

In 1997, Pitman published his paper *Coalescent Random Forests* [10] which defined a concept he called refining sequences that end in a given forest and then he counted those sequences in two ways to prove Cayley's Theorem. His approach was also summarized in *Proofs from the Book* [2] and the following proof is based on both sources.

4.1.1 Definitions and Notation

Let $n \in \mathbb{N}$ and $k \in \mathbb{N}$ such that $k \leq n$.

Rooted Forests

Let $\mathscr{F}_{n,k}$ denote the set of rooted forests on $\{v_1, v_2, ..., v_n\}$ such that there are k trees in the forest.

Let $F_{n,k}$ denote the set of directed rooted forests on $\{1, 2, ..., n\}$ such that there are k trees in the forest, each tree has a designated "root", and all the edges in each tree are directed away from the root.

Refining Sequences of Forests

For $\mathscr{F} \in \mathscr{F}_{n,k}$ and $\mathscr{F}' \in \mathscr{F}_{n,j}$ where $0 \leq k < j \leq n$ with the corresponding directed forests $F \in F_{n,k}$ and $F' \in F_{n,j}$, let the notation $F' \subset F$ denote that F' can be built from F by deleting edges from F.

Let $\{F_1, F_2, ..., F_k\}$ be called a refining sequence of forests if $F_i \in F_{n,i}$ and $F_{i+1} \subset F_i$ for all $1 \leq i < k$.

Notation for Counting Sets

Let $\mathscr{F}_k \in \mathscr{F}_{n,k}$ be a fixed rooted forest with the corresponding directed rooted forest $F_k \in F_{n,k}$.

Let $R(F_k) = |\{F \in F_{n,1} : F_k \subset F\}|$ denote the number of directed rooted trees that contain F_k.

Let $S(F_k)$ denote the number of refining sequences that end in F_k.

4.1.2 Count $S(F_k)$ in Two Ways

Let $F_k \in F_{n,k}$ be a directed rooted forest with the edges of each tree directed away from the roots.

Counting Method 1:

Suppose that $F_1 \in \{F \in F_{n,1} : F_k \subset F\}$.

Forests with k trees have $n - k$ edges. So, the number of edges in F_1 is $n - 1$ and the number of edges in F_k is $n-k$. Therefore, F_1 has $k-1$ more edges than F_k.

We can define a refining sequence $\{F_1, ..., F_k\}$ by deleting the $k-1$ edges of $F_1 \backslash F_k$ one at a time.

There are $(k-1)!$ ways to choose the order of edge deletion to get the refining sequence, so

$$S(F_k) = R(F_k)(k-1)!$$

Counting Method 2:

The forest F_k has k trees in it.

We can create a forest F_{k-1} from F_k by attaching a directed edge from any vertex in the forest to the root of a tree that does not contain that vertex.

There are n vertices to choose in the forest and $k-1$ trees that do not contain that vertex so the number of ways to create the forest F_{k-1} is $n(k-1)$.

After we define F_{k-1}, then we can define F_{k-2} by choosing any of the n vertices in the forest and attaching a directed edge from it to the root of any of the $k-2$ trees that do not contain that vertex.

By iterating this process until there is only one tree in the forest F_1, we define a refining sequence $\{F_1, ..., F_k\}$. And the number of ways to generate the sequence is

$$n(k-1)n(k-2)...n(1) = n^{k-1}(k-1)!$$

Therefore,

$$S(F_k) = n^{k-1}(k-1)!$$

Conclusion:

Because $S(F_k) = R(F_k)(k-1)!$ and $S(F_k) = n^{k-1}(k-1)!$, then

$$R(F_k)(k-1)! = n^{k-1}(k-1)!$$

$$R(F_k) = n^{k-1}$$

4.1.3 Prove Cayley's Theorem

Let $k = n$.

$$R(F_n) = n^{n-1}$$

There is only one choice for $F_n \in F_{n,n}$ because the only way to get a forest of n trees on n vertices is to have a forest of n isolated vertices.

Therefore, $R(F_n)$ is the number of rooted directed trees on n labeled vertices.

And because there are n ways to choose a root for a tree on n labeled vertices, then

$$n|T_n| = R(F_n)$$

Substituting in the result we found above gives us Cayley's Theorem

$$n|T_n| = n^{n-1}$$

$$|T_n| = n^{n-2}$$

4.2 Clarke - Counting Linkages of Trees

In 1958, Clarke published his paper *On Cayley's Formula for Counting Trees* [5] which categorized trees by the number of vertices adjacent to the vertex n. Then he used that categorization to define a concept he called linkages of trees and counted the linkages in two ways to prove Cayley's Theorem.

4.2.1 Definitions

Trees of Type l

Let t be a tree with vertices $\{v_1, v_2, ..., v_n\}$.

t is considered to be a tree of type l if the vertex v_n shares edges with l other vertices.

The notation $N_{l,n}$ denotes the number of trees of type l.

Linked Trees

Let x be a tree of type $l-1$ and let y be a tree of type l.

x and y are considered to be linked trees x can be converted into y by choosing one of the vertices v that does not share an edge with n in x, identifying the path from $v \to n$, deleting the first edge in the path, and then reattaching v to the tree by creating an edge between v and n.

For example, in the diagram below, the tree x is of type 3 because there are 3 edges attached to the vertex v_8 and the tree y is of type 4 because there are 4 edges attached to the vertex v_8.

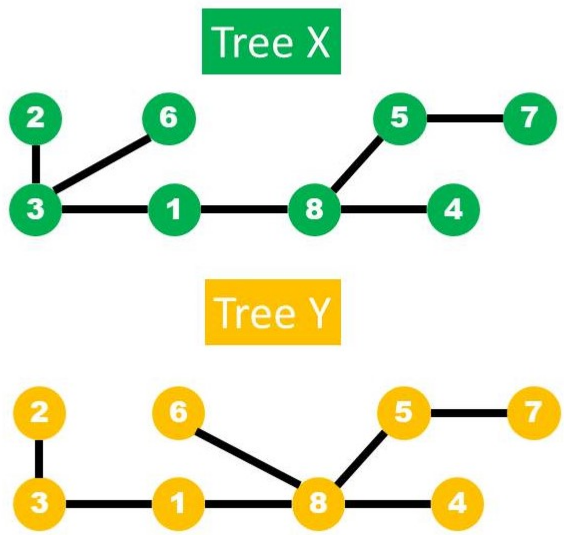


Figure 27

The trees x and y are linked trees because the vertex v_6 does not share an edge with vertex v_8 in x and if the first edge in the path from $v_6 \to v_8$ is deleted and an edge is created between v_6 and v_8, then the resulting graph is the same graph as y.

Linkages of Type l

A linkage of type l is an ordered pair (x, y) where x and y are linked trees and x is a tree of type $l-1$ and y is a tree of type l.

I will use the notation $M_{l,n}$ to denote the number of linkages of type l on n vertices.

4.2.2 Count the Number of Linkages of Type l in Two Ways

Let $2 \leq l \leq n-1$.

Counting Method 1:

Let x be a tree of type $l-1$.

By definition, x has $(n-1)-(l-1) = n-l$ vertices that do not share an edge with the vertex v_n because there are $n-1$ vertices that are not v_n itself and $l-1$ of those vertices are adjacent to v_n.

Therefore, there are $l-1$ vertices that can have an edge deleted and then reattached to n to form a linked tree y of type l.

This is true for every tree of type $l-1$, therefore,

$$M_{l,n} = (n-l)N_{l-1,n}$$

Counting Method 2:

Let y be a tree of type l.

Without loss of generality, assume that the vertices that are adjacent to v_n are the vertices $\{v_1, v_2, ..., v_l\}$.

Delete the edges that attach $\{v_1, v_2, ..., v_l\}$ to v_n.

The resulting graph is a forest of $l+1$ trees where one of the trees is the solitary point v_n.

Let p_i denote the number of vertices in the tree t_i that contains the vertex indexed by $1 \leq i \leq l$.

It is obvious that $p_1 + p_2 + ... + p_l = n-1$ because the forest has n vertices, one of which is v_n.

Choose one of the vertices with an index $1 \leq i \leq l$ and attach it to one of the $n-1-p_i$ vertices that is not v_n and then reattach the remaining vertices in $\{v_1, v_2, ..., v_l\}$ to v_n to create a linked tree of type $l-1$.

Each linked tree of type $l-1$ can be generated by this process as i ranges from 1 to l for every tree y of type l. Therefore,

$$M_{l,n} = N_{l,n} \sum_{i=1}^{l} n - 1 - p_i$$

Expanding this sum and rearranging the addends gives:

$$M_{l,n} = N_{l,n}[(n-1-p_1) + (n-1-p_2) + ... + (n-1-p_l)]$$

$$M_{l,n} = N_{l,n}[l(n-1) - (p_1 + p_2 + ... + p_l)]$$

$$M_{l,n} = N_{l,n}[l(n-1) - (n-1)]$$

$$M_{l,n} = N_{l,n}(l-1)(n-1)$$

Conclusion:

Because $(n-l)N_{l-1,n} = M_{l,n} = N_{l,n}(l-1)(n-1)$, then for $2 \leq l \leq n-1$

$$N_{l-1,n} = \frac{(n-1)(l-1)}{n-l} N_{l,n}$$

Changing the indexing so that $k = l-1$ yields

$$N_{k,n} = \frac{(n-1)k}{n-k-1} N_{k+1,n} \quad \text{for all } 1 \leq k \leq n-2$$

If $k = n-1$, then there is only one tree of type k, namely the tree where v_n shares edges with all the other vertices.

For any $k \leq 0$, $N_{k,n} \leq 0$ because there are no trees on n vertices that have the vertex v_n attached to 0 other vertices. And for any $k \geq n$, $N_{k,n} \leq 0$ because it is impossible to attach v_n to n other vertices when there are only n vertices in the graph.

Therefore, the number of trees of type k on n vertices are defined by the recursive function:

$$N_{k,n} = \begin{cases} 1 & \text{if } k = n-1 \\ \frac{(n-1)k}{n-k-1} N_{k+1} & \text{if } 1 \le k \le k-2 \\ 0 & \text{otherwise} \end{cases}$$

4.2.3 Apply Induction to the Recursive Formula

Our double-counting argument defined a recursive formula for $N_{l,n}$.

Now we will use induction to prove that

$$N_{l,n} = \binom{n-2}{l-1}(n-1)^{n-l-1}$$

Because our recursive formula is defined by a base case of $l = n-1$ and then the formula works backwards to $l = 1$, our base case will be $l = n-1$ and the inductive step will show that if the induction statement is true for all $l \ge k$, then it is true for k.

Inductive Statement:

For all $1 \le l \le n-1$,

$$N_{l,n} = \binom{n-2}{l-1}(n-1)^{n-l-1}$$

Base Case:

Let $l = n-1$.

As argued above, there is only one tree of type $n-1$, namely the tree where n shares edges with all the other vertices.

So, $N_{n-1,n} = 1$ which is consistent with the result

$$N_{n-1,n} = \binom{n-2}{(n-1)-1}(n-1)^{n-(n-1)-1} = \binom{n-2}{n-2}(n-1)^0 = 1$$

Inductive Step:

Let $1 \leq k \leq n-2$ and assume that $N_{l,n} = \binom{n-2}{l-1}(n-1)^{n-l-1}$ for all $l \geq k$.

By the definition of our recursive formula,

$$N_{k,n} = \frac{(n-1)k}{n-k-1} N_{k+1,n}$$

Because $k+1 \geq k$, then by the induction assumption,

$$N_{k+1,n} = \binom{n-2}{(k+1)-1}(n-1)^{n-(k+1)-1} = \binom{n-2}{k}(n-1)^{n-k-2}$$

Therefore,

$$N_{k,n} = \frac{(n-1)k}{n-k-1}\binom{n-2}{k}(n-1)^{n-k-2}$$

The factorial expansion of the binomial coefficient 1.4.1 says that $\binom{n-2}{k} = \frac{(n-2)!}{k!(n-2-k)!}$ so

$$N_{k,n} = \frac{(n-1)k}{n-k-1}\frac{(n-2)!}{k!(n-2-k)!}(n-1)^{n-k-2}$$

$$N_{k,n} = \frac{(n-1)k}{n-k-1}\frac{(n-2)!}{k(k-1)!(n-2-k)!}(n-1)^{n-k-2}$$

$$N_{k,n} = \frac{(n-2)!}{(k-1)!(n-k-1)(n-2-k)!}\frac{(n-1)k(n-1)^{n-k-2}}{k}$$

$$N_{k,n} = \frac{(n-2)!}{(k-1)!(n-k-1)!}(n-1)^{n-k-1}$$

$$N_{k,n} = \binom{n-2}{k-1}(n-1)^{n-k-1}$$

Inductive Conclusion:

For all $1 \leq l \leq n-1$,

$$N_{l,n} = \binom{n-2}{l-1}(n-1)^{n-l-1}$$

4.2.4 Prove Cayley's Theorem

Because all trees on n vertices will be of type l for some $1 \le l \le n-1$, then

$$|T_n| = \sum_{l=1}^{n-1} N_{l,n}$$

Re-indexing the sum so $j = l - 1$ gives us:

$$|T_n| = \sum_{j=0}^{n-2} N_{j+1,n}$$

Substituting the result from our induction gives:

$$|T_n| = \sum_{j=0}^{n-2} \binom{n-2}{j} (n-1)^{n-j}$$

The binomial theorem 1.4.4 says that

$$(n-1+1)^{n-2} = \sum_{j=0}^{n-2} \binom{n-2}{j} (n-1)^{n-j} 1^j$$

Substituting this into our equation for T_n gives us the result of Cayley's Theorem.

$$|T_n| = (n-1+1)^{n-2} = n^{n-2}$$

5 Generating Functions

Generating functions can be used to solve many types of combinatorial problems where the things you are trying to count depend on the natural numbers in some way.

The strategy of a generating function proof is to think of the number of objects you are trying to count as a sequence since they depend on the natural numbers. Then you define a formal power series that contains the sequence terms as coefficients and manipulate it to see if you can find a closed formula for the sequence.

For example, we could define a sequence with terms $|T_n|$ and then define a generating function

$$f(x) = \sum_{n\geq 1} |T_n| \frac{x^n}{n!}$$

Based on the cases we verified earlier for $|T_n|$, the first few terms of this power series should be

$$f(x) = 1x + 1\frac{x^2}{2!} + 3\frac{x^3}{3!} + 16\frac{x^4}{4!} + 125\frac{x^5}{5!} + 1296\frac{x^6}{6!} + ...$$

Through manipulating this formula or perhaps defining other generating functions for related objects we would hope to be able to show that the following is true to prove Cayley's Theorem.

$$f(x) = \sum_{n\geq 1} n^{n-2} \frac{x^n}{n!}$$

Defining a generating function as a formal power series allows us to use the terms of the power series as a sort of clothesline on which we hang the terms of the sequence we are trying to define. It also very convenient because in the ring of formal power series, questions of convergence are nonexistent. This concept is covered in more detail in Wilf's *generatingfunctionality* book [18].

There are two types of generating functions that are commonly used. Ordinary generating functions have the form $f(x) = \sum A_n x^n$ and exponential generating functions have the form $f(x) = \sum A_n \frac{x^n}{n!}$. Exponential generating functions are especially useful because their structure makes it easy to take the derivative of the entire sum as part of the manipulation process.

In the proofs that follow we will use exponential generating functions to prove Cayley's theorem.

5.1 Pólya - Labeled Rooted Trees and Forests

In 1937, Pólya [11] published a proof of Cayley's Theorem which relates two generating functions defined to count labeled rooted trees and labeled rooted forests and then uses the result to count the number of non-rooted labeled trees.

Because his original paper was written in German, the following proof is based on a section in Stanley's *Enumerative Combinatorics Volume 2* [16].

5.1.1 Definitions

Let t_n be the set of labeled rooted trees on n vertices.

Let f_n be the set of labeled rooted forests on n vertices.

5.1.2 Proof

Relate $|T_n|$, $|t_n|$, and $|f_n|$

Any labeled tree can be converted in to a labeled rooted tree by choosing one of the n vertices to be the root. Therefore,

$$|t_n| = n|T_n|$$

Any labeled rooted forest on n vertices can be converted into a labeled tree on $n+1$ vertices by adding a new vertex and attaching the roots of each tree in the forest to the new vertex. Therefore,

$$|f_n| = |T_{n+1}|$$

Define and Relate Exponential Generating Functions

Let $y = \sum_{n\geq 1} |t_n| \frac{x^n}{n!}$ be the exponential generating function for $|t_n|$.

Let $z = \sum_{n\geq 0} |f_n| \frac{x^n}{n!}$ be the exponential generating function for $|f_n|$.

Because a rooted forest is a collection of rooted trees, then we can use the exponential formula to count z as a function of y.

$$z = e^y$$

And because we know that $|f_n| = |T_{n+1}|$ and $|T_{n+1}| = \frac{|t_{n+1}|}{n+1}$, then

$$xz = \sum_{n\geq 0} |f_n| \frac{x^{n+1}}{n!} = \sum_{n\geq 0} \frac{t_{n+1}}{n+1} \frac{x^{n+1}}{n!} = \sum_{n\geq 0} t_{n+1} \frac{x^{n+1}}{(n+1)!} = y$$

Therefore,

$$xe^y = y$$

Then by applying the Lagrange Inversion Theorem to write the power series of y, we get

$$y = \sum_{n \geq 1} n^{n-1} \frac{x^n}{n!}$$

5.1.3 Conclusion:

Therefore, by the definition of the generating function,

$$|t_n| = n^{n-1}$$

And because $|t_n| = n|T_n|$, then we get Cayley's Theorem

$$n|T_n| = n^{n-1}$$

$$|T_n| = n^{n-2}$$

5.2 Shukla - Labeled Trees with Specified Edge

In 2018, Shukla [15] published a proof of Cayley's Theorem which relates the number of labeled trees to the the number of labeled trees with specified edges with a recurrence relation and then uses that relation to manipulate an exponential generating function for the number of labeled trees.

5.2.1 Definitions

Let $C_n = \{t \in T_n : \text{t contains a specific edge}\}$ be the set of trees that contain a specific edge.

5.2.2 Proof:

Counting C_n

Rearranging the labels of the vertices does not affect the value of $|T_n|$. Therefore, we know that each choice of specific edge contributes to the overall value of $|T_n|$ equally.

So, without loss of generality, let the specified edge be the edge between vertices v_1 and v_2.

Because the edge between v_1 and v_2 must be in the tree, then any tree in C_n can be built by attaching smaller trees to the vertices v_1 and v_2. To do this, choose k vertices from the vertices

$\{v_3, v_4, ..., v_n\}$. There are $\binom{n-2}{k}$ ways to do this.

These k vertices can be attached as a tree to vertex v_1 in $|T_{k+1}|$ ways because there are $|T_{k+1}|$ trees that contain all k vertices and vertex v_1. By a similar argument, the remaining $(n-2)-k$ vertices can be attached as a tree to vertex v_2 in $|T_{[(n-2)-k]+1}|$ ways.

The number k can be chosen as any integer from 0 to $n-2$. If $k = 0$, then vertex v_1 is a leaf of the tree and if $k = n-2$, then vertex v_2 is a leaf of the tree.

Therefore, the total number of trees such that v_1 and v_2 are adjacent in the tree is

$$|C_n| = \sum_{k=0}^{n-2} \binom{n-2}{k} |T_{k+1}| \, |T_{[(n-2)-k]+1}| = \sum_{k=0}^{n-2} \binom{n-2}{k} |T_{k+1}| \, |T_{n-k-1}|$$

Relating $|T_n|$ and $|C_n|$

Consider the set of all trees on n labeled vertices paired with a specific edge in that tree.

$$\{(t, e) : t \in T_n, e \in t\}$$

Because there are $|T_n|$ trees in T_n and each of these trees has $n-1$ edges,

$$|\{(t, e) : t \in T_n, e \in t\}| = (n-1)|T_n|$$

Another way to construct the set above is to choose two vertices from $\{v_1, v_2, ..., v_n\}$ and construct an edge between them. Then construct all the possible trees that contain that edge.

There are $\binom{n}{2}$ ways to choose a specific edge between n vertices and once that edge is chosen, there are $|C_n|$ trees that contain that edge.

Therefore,

$$|\{(t, e) : t \in T_n, e \in t\}| = \binom{n}{2}|C_n| = \frac{n!}{2!(n-2)!}|C_n| = \frac{n(n-1)}{2}|C_n|$$

Combining these two equations gives us:

$$(n-1)|T_n| = \frac{1}{2}n(n-1)|C_n|$$

$$|T_n| = \frac{1}{2}n|C_n|$$

By substituting in the result from counting C_n,

$$|T_n| = \frac{1}{2}n\sum_{k=0}^{n-2}\binom{n-2}{k}|T_{k+1}||T_{n-k-1}|$$

$$\frac{2|T_n|}{n} = \sum_{k=0}^{n-2}\binom{n-2}{k}|T_{k+1}||T_{n-k-1}|$$

Define and Manipulate an Exponential Generating Function

Let $f(x)$ be defined as the following exponential generating function:

$$f(x) = \sum_{n=1}^{\infty}|T_n|\frac{x^n}{(n-1)!}$$

Note that

$$f'(x) = \sum_{n=1}^{\infty}|T_n|\frac{nx^{n-1}}{(n-1)!}$$

Squaring $f(x)$ gives:

$$[f(x)]^2 = \sum_{n=1}^{\infty}\sum_{m=1}^{n-1}\left[|T_m|\frac{x^m}{(m-1)!}\right]\left[|T_{n-m}|\frac{x^{n-m}}{(n-m-1)!}\right]$$

And re-indexing the second sum with $k = m-1$ gives:

$$[f(x)]^2 = \sum_{n=1}^{\infty}\sum_{k=0}^{n-2}\left[|T_{k+1}|\frac{x^{k+1}}{k!}\right]\left[|T_{n-k-1}|\frac{x^{n-k-1}}{(n-k-2)!}\right]$$

Note that $x^{k+1}x^{n-k-1} = x^{k+1+n-k-1} = x^n$ and the factorials can be written as:

$$\frac{1}{k!(n-k-2)!} = \frac{1}{k!(n-2-k)!} = \frac{(n-2)!}{k!(n-2-k)!(n-2)!} = \frac{1}{(n-2)!}\frac{(n-2)!}{k!(n-2-k)!}$$

$$= \frac{1}{(n-2)!}\binom{n-2}{k} = \frac{n-1}{(n-1)(n-2)!}\binom{n-2}{k} = (n-1)\frac{1}{(n-1)!}\binom{n-2}{k}$$

Therefore,

$$[f(x)]^2 = \sum_{n=1}^{\infty}\sum_{k=0}^{n-2}(n-1)\binom{n-2}{k}|T_{k+1}||T_{n-k-1}|\frac{x^n}{(n-1)!}$$

Since the inner sum doesn't depend on n, we can pull the $(n-1)$ and $\frac{x^n}{(n-1)!}$ factors outside the sum

$$[f(x)]^2 = \sum_{n=1}^{\infty}(n-1)\frac{x^n}{(n-1)!}\sum_{k=0}^{n-2}\binom{n-2}{k}|T_{k+1}||T_{n-k-1}|$$

Notice that the inner sum is equal to $\frac{2|T_n|}{n}$ from the previous step.

Therefore,

$$[f(x)]^2 = \sum_{n=1}^{\infty}(n-1)\frac{x^n}{(n-1)!}\frac{2|T_n|}{n} = \sum_{n=1}^{\infty}\frac{2(n-1)}{n}|T_n|\frac{x^n}{(n-1)!}$$

Differentiating this expression with respect to x gives:

$$2f(x)f'(x) = \sum_{n=1}^{\infty}\frac{2(n-1)}{n}|T_n|\frac{nx^{n-1}}{(n-1)!} = \sum_{n=1}^{\infty}2(n-1)|T_n|\frac{x^{n-1}}{(n-1)!}$$

Dividing by 2 gives:

$$f(x)f'(x) = \sum_{n=1}^{\infty}(n-1)|T_n|\frac{x^{n-1}}{(n-1)!}$$

Distributing the $n-1$ and separating the sums gives:

$$f(x)f'(x) = \sum_{n=1}^{\infty}n|T_n|\frac{x^{n-1}}{(n-1)!} - \sum_{n=1}^{\infty}|T_n|\frac{x^{n-1}}{(n-1)!}$$

By rearranging the notation of the first sum and multiplying by $\frac{x}{x}$ in the second sum, we get:

$$f(x)f'(x) = \sum_{n=1}^{\infty}|T_n|\frac{nx^{n-1}}{(n-1)!} - \sum_{n=1}^{\infty}|T_n|\frac{x^n}{(n-1)!x}$$

Therefore, by the original definition of the generating function:

$$f(x)f'(x) = f'(x) - \frac{f(x)}{x} = \frac{xf'(x)}{x} - \frac{f(x)}{x} = \frac{xf'(x)-f(x)}{x}$$

Therefore,

$$f'(x) = \frac{xf'(x)-f(x)}{xf(x)}$$

Note that for $G(x) = ln(\frac{f(x)}{x})$,

$$G'(x) = \frac{x}{f(x)} \frac{f'(x)x - f(x)}{x^2} = \frac{f'(x)x - f(x)}{xf(x)}$$

Therfore, by integrating the expression for $f'(x)$ we get,

$$f(x) = ln\left(\frac{f(x)}{x}\right) + c$$

Where c is an integration constant.

Because the generating function was defined to be $f(x) = \sum_{n=1}^{\infty} |T_n| \frac{x^n}{(n-1)!}$, then $f(0) = 0$ which implies that $c = 0$.

Therefore,

$$f(x) = ln\left(\frac{f(x)}{x}\right)$$

$$e^{f(x)} = \frac{f(x)}{x}$$

$$f(x) = xe^{f(x)}$$

Then use the Lagrange Inversion Theorem to write the power series of $f(x)$,

$$f(x) = \sum_{n=1}^{\infty} \frac{n^{n-1}}{n!} x^n = \sum_{n=1}^{\infty} \frac{n^{n-1} x^n}{n(n-1)!} = \sum_{n=1}^{\infty} n^{n-2} \frac{x^n}{(n-1)!}$$

5.2.3 Conclusion:

Therefore, by definition of the generating function,

$$|T_n| = n^{n-2}$$

Therefore, the number of trees on n labeled vertices is n^{n-2}.

6 Linear Algebra Proof

Linear Algebra is a thoroughly studied field and it is often useful to use its theorems to solve combinatorics problems.

The strategy of applying linear algebra to combinatorics is to use a matrix to describe some aspect of the problem and then prove a connection between the linear algebra we can do with that matrix and the problem we are trying to solve.

The following Linear Algebra proof of Cayley's Theorem is outlined in *Proofs from the Book* [2]

Summary

Kirchoff's Theorem states that the number of trees in a graph is equal the the determinant of the graph's Laplacian matrix. Therefore, to calculate the number of trees in a complete graph, we can simply find the Laplacian matrix of a complete graph and find its determinant.

To prove Kirchoff's Theorem, we will first need to cover some background information.

So, this section is organized as follows:

- **Determinants in Bipartite Graphs**

  Use the simple example of bipartite graphs to understand how the determinant of a square matrix relates to the disjoint path systems in the graph.

- **Gessel Viennot Lemma**

  Prove the Gessel Viennot Lemma to extend the relationship between determinants and disjoint path systems to include arbitrary graphs instead of only bipartite graphs.

- **Cauchy-Binet Theorem**

  Use the Gessel Viennot Lemma to prove the Cauchy-Binet Theorem.

- **Kirchoff's Theorem**

  Use the Cauchy-Binet Theorem to prove Kirchoff's Theorem.

- **Apply Kirchoff's Theorem to Complete Graphs**

  Prove Cayley's Theorem by applying Kirchoff's Theorem to complete graphs.

6.1 Determinants in Bipartite Graphs

Permutation Notation

A permutation σ on the set $\{1, 2, ..., n\}$ is a bijection that maps the set to itself.

An example of a permutation on $\{1, 2, ..., 8\}$ is

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 8 & 7 & 1 & 5 & 2 & 6 & 3 \end{pmatrix}$$

In this example, $\sigma(1) = 4$ and $\sigma(2) = 8$ and $\sigma(3) = 7$ and so on.

All permutations can be written as the product of transpositions and the identity permutation.

In the example above, there are three disjoint cycles as follows:

$$1 \to 4 \to 1$$

$$2 \to 8 \to 3 \to 7 \to 6 \to 2$$

$$5 \to 5$$

These disjoint cycles can be converted into transpositions so that

$$\sigma = (14)(28)(83)(37)(76) \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}$$

The product of transpositions can be calculated to confirm that they really do equal σ.

$$\sigma = (14)(28)(83)(37) \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 7 & 6 & 8 \end{pmatrix}$$

$$\sigma = (14)(28)(83) \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 7 & 4 & 5 & 3 & 6 & 8 \end{pmatrix}$$

$$\sigma = (14)(28) \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 7 & 4 & 5 & 8 & 6 & 3 \end{pmatrix}$$

$$\sigma = (14)\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 8 & 7 & 4 & 5 & 2 & 6 & 3 \end{pmatrix}$$

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 8 & 7 & 1 & 5 & 2 & 6 & 3 \end{pmatrix}$$

The sign of a permutation σ is -1 if the permutation has an odd number of transpositions and $+1$ if the permutation has an even number of transpositions.

In the example above, $sign(\sigma) = -1$ because there are 5 transpositions.

Determinant Definition

For an $n \times n$ matrix M with real entries m_{ij}, the determinant of M is defined to be:

$$det(M) = \sum_{\sigma} sign(\sigma) m_{1\sigma(1)} m_{2\sigma(2)} ... m_{n\sigma(n)}$$

Where the summation sums across all of the $n!$ possible permutations on $\{1, 2, ..., n\}$.

For example, let

$$M = \begin{pmatrix} 3 & -9 & 8 \\ 1 & -5 & 0 \\ 2 & 4 & 6 \end{pmatrix}$$

There are 6 possible permutations on $\{1, 2, 3\}$ so the determinant of M can be calculated as:

Permutation	sign(σ)	m$_{1\sigma(1)}m_{2\sigma(2)}m_{3\sigma(3)}$	sign(σ)$m_{1\sigma(1)}m_{2\sigma(2)}m_{3\sigma(3)}$
$\sigma_1 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} = (13) \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$	-1	(8)(-5)(2)=-80	80
$\sigma_2 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} = (23) \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$	-1	(3)(0)(4)=0	0
$\sigma_3 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} = (12) \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$	-1	(-9)(1)(6)=-54	54
$\sigma_4 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = (12)(23) \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$	1	(-9)(0)(2)=0	0
$\sigma_5 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} = (13)(32) \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$	1	(8)(1)(4)=32	32
$\sigma_6 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$	1	(3)(-5)(6)=-90	-90
			det(M)=80+54+32-90=76

Table 4

A quick computer calculation verifies that $det(M) = 76$.

Interpreting Determinants in Bipartite Graphs

Define vertices $A_1, A_2, ..., A_n$ and $B_1, B_2, ..., B_n$ where the directed edges $A_i \to B_j$ are weighted with the value of the m_{ij} entry in the matrix M.

Then for any permutation σ, the expression $m_{1\sigma(1)}m_{2\sigma(2)}...m_{n\sigma(n)}$ represents the weight of a vertex-disjoint path system P_σ where the weight of the path system is defined as the product of the weights of the individual paths.

In the example above, where

$$\sigma_1 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \text{ and } M = \begin{pmatrix} 3 & -9 & 8 \\ 1 & -5 & 0 \\ 2 & 4 & 6 \end{pmatrix}$$

Then the disjoint path system P_{σ_1} would look like this:

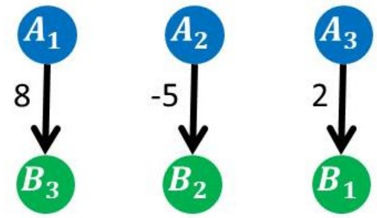


Figure 28

And the weight of P_{σ_1} would be:

$$w(P_{\sigma_1}) = w(A_1 \to B_3)w(A_2 \to B_2)w(A_3 \to B_1) = (8)(-5)(2) = -80$$

The determinant is interpreted as the sum of the signed weights of all the possible disjoint path systems.

6.2 Gessel Viennot Lemma

The Gessel Viennot Lemma extends the concept above from bipartite graphs to arbitrary graphs.

6.2.1 Definitions

Finite Acyclic Directed Graph

A finite acyclic directed graph G is a graph where there are finitely many directed paths from any vertex $A \in V(G)$ to a vertex $B \in V(G)$ and there are no directed cycles in the graph.

Weighted Paths

Let G be a finite acyclic directed graph with specified weights $w(e)$ for all $e \in E(G)$.

If $P : A \to B$ is a directed path from vertex A to vertex B such that $A \neq B$, then define the weight of the path P to be

$$w(P) = \prod_{e \in P} w(e)$$

If $A = B$, then $P : A \to B$ is the trivial path from A to itself, so we will define $w(P) = 1$.

Path Matrix

Let $A = \{A_1, A_2, ..., A_n\}$ and $B = \{B_1, B_2, ..., B_n\}$ be two sets of n vertices in G where the sets A and B are not necessarily disjoint.

Define the $n \times n$ path matrix M by the entries:

$$m_{ij} = \sum_{P:A_i \to B_j} w(P)$$

Path System

A path system $\mathscr{P}$ from A to B is a permutation σ with n paths $P_i : A_i \to B_{\sigma(i)}$ for $i = 1, 2, ..., n$.

$$\mathscr{P} = (\sigma, P_1, P_2, ..., P_n)$$

Define the sign and weight of $\mathscr{P}$ to be

$$sign(\mathscr{P}) = sign(\sigma)$$

$$w(\mathscr{P}) = \prod_{i=1}^{n} w(P_i)$$

A path system $\mathscr{P}$ is called vertex-disjoint if the paths P_i and P_j are vertex disjoint for all $1 \leq i \neq j \leq n$.

I will designate vertex-disjoint path systems with $\mathscr{P}^*$ and non-vertex-disjoint path systems with $\mathscr{P}^\times$.

6.2.2 Lemma Statement:

The Gessel Viennot Lemma states that if G is a finite weighted acyclic directed graph, $A = \{A_1, A_2, ..., A_n\}$ and $B = \{B_1, B_2, ..., B_n\}$ are two n-sets of vertices in G, and M is the path matrix from A to B, then

$$det(M) = \sum_{\mathscr{P}^*} sign(\mathscr{P}^*) w(\mathscr{P}^*)$$

6.2.3 Proof:

Writing out the Determinant of M

By the definition of determinant,

$$det(M) = \sum_{\sigma} sign(\sigma) m_{1\sigma(1)} m_{2\sigma(2)} ... m_{n\sigma(n)}$$

And by the way we defined $sign(\mathscr{P})$, $w(\mathscr{P})$ and M, this can be rewritten as:

$$det(M) = \sum_{\mathscr{P}} sign(\mathscr{P}) w(\mathscr{P})$$

Because every path system $\mathscr{P}$ is either vertex-disjoint or it is non-vertex-disjoint,

$$det(M) = \sum_{\mathscr{P}^*} sign(\mathscr{P}^*) w(\mathscr{P}^*) + \sum_{\mathscr{P}^\times} sign(\mathscr{P}^\times) w(\mathscr{P}^\times)$$

Therefore, to prove the lemma, we only need to show that

$$\sum_{\mathscr{P}^\times} sign(\mathscr{P}^\times) w(\mathscr{P}^\times) = 0$$

Summary of the Strategy

An involution is a function that is its own inverse.

If we can define an involution $\pi : \{\mathscr{P}^\times\} \to \{\mathscr{P}^\times\}$ such that for every $\mathscr{P}^\times$,

$$w(\pi \mathscr{P}^\times) = w(\mathscr{P}^\times)$$

$$sign(\pi \mathscr{P}^\times) = -sign(\mathscr{P}^\times)$$

Then this will imply that $\sum_{\mathscr{P}^\times} sign(\mathscr{P}^\times) w(\mathscr{P}^\times) = 0$.

This is because if $sign(\pi \mathscr{P}^\times) = -sign(\mathscr{P}^\times)$, then

$$\pi \mathscr{P}^\times \neq \mathscr{P}^\times$$

Therefore, the fact that every involution is a bijection implies that every $\mathscr{P}_i^\times$ is uniquely paired with a $\mathscr{P}_j^\times = \pi \mathscr{P}_i^\times$ such that $\mathscr{P}_j^\times \neq \mathscr{P}_i^\times$.

This means that the sum $\sum_{\mathscr{P}^{\times}} sign(\mathscr{P}^{\times})w(\mathscr{P}^{\times})$ can be arranged as the sum of i-j pairs.

$$sign(\mathscr{P}_i^{\times})w(\mathscr{P}_i^{\times}) + sign(\mathscr{P}_j^{\times})w(\mathscr{P}_j^{\times})$$

$$= sign(\mathscr{P}_i^{\times})w(\mathscr{P}_i^{\times}) + sign(\pi\mathscr{P}_i^{\times})w(\pi\mathscr{P}_i^{\times})$$

$$= sign(\mathscr{P}_i^{\times})w(\mathscr{P}_i^{\times}) - sign(\mathscr{P}_i^{\times})w(\mathscr{P}_i^{\times})$$

$$= 0$$

Therefore, if such an involution exists, then the following sum is a sum of pairs that sum to zero.

$$\sum_{\mathscr{P}^{\times}} sign(\mathscr{P}^{\times})w(\mathscr{P}^{\times}) = 0$$

Defining the Involution

Let $\mathscr{P} \in \{\mathscr{P}^{\times}\}$ be defined by:

$$\mathscr{P} = (\sigma, P_1, P_2, ..., P_n)$$

By definition, $\mathscr{P}$ is a non-vertex-disjoint path system.

Therefore, there are at least two paths in $P_i, P_j \in \mathscr{P}$ that share a vertex with another path.

Let $S = \{i : P_i \text{ shares a vertex with another path}\}$.

Define $s = min(S)$ and let X be the first vertex that the path P_s shares with another path.

Let $R = \{j : P_j \text{ shares a vertex with } P_s\}$ and define $r = min(R)$.

Define the function $\pi : \{\mathscr{P}^{\times}\} \to \{\mathscr{P}^{\times}\}$ by

$$\pi(\mathscr{P}) = (\sigma', P_1', P_2', ..., P_n')$$

Where $P_i' = P_i$ for all $i \neq s$ and $i \neq r$.

Then define the path P'_s as the path from $A_s \to X$ along P_s and then $X \to B_{\sigma(r)}$ along P_r.

Similarly, the path P'_r is the path from $A_r \to X$ along P_r and then from $X \to B_{\sigma(s)}$ along P_s.

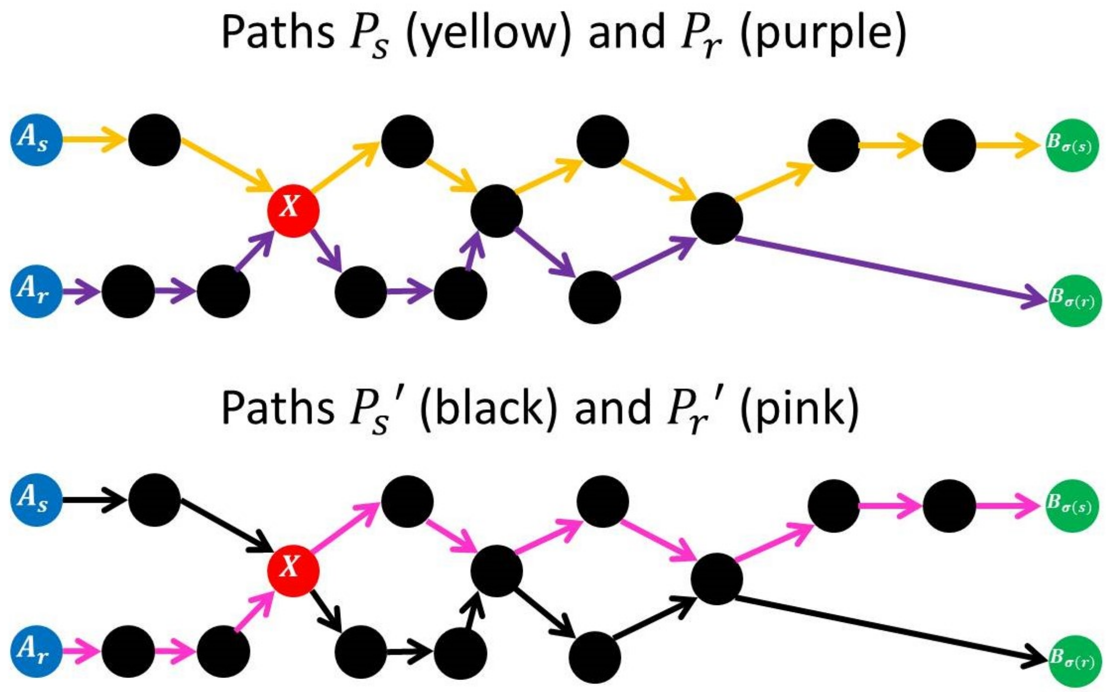


Figure 29

Notice that it doesn't matter how many times the paths P_s and P_r cross after the vertex X.

Because all the paths except for P'_s and P'_r were identical to the original paths and the new definition of P'_s and P'_r essentially switches the endpoints and the tail end of the paths P_s and P_r, then the permutation σ' has to be defined as

$$\sigma' = (sr)\sigma$$

Therefore,

$$sign(\pi(\mathscr{P})) = sign(\sigma') = -sign(\sigma) = -sign(\mathscr{P})$$

And because the edges traversed in P_s and P_r are the same edges in P'_s and P'_r, then

$$w(P_s)w(P_r) = \prod_{e \in P_s} w(e) \prod_{e \in P_r} w(e) = \prod_{e \in P'_s} w(e) \prod_{e \in P'_r} w(e) = w(P'_s)w(P'_r)$$

This fact and the fact that $P_i = P_i'$ for all $i \neq s, r$ implies that

$$w(\mathscr{P}) = \prod_{i=1}^{n} w(P_i) = \prod_{i=1}^{n} w(P_i') = w(\pi(\mathscr{P}))$$

Therefore, our function meets all the requirements listed in our strategy.

It is clear that our function is an involution because if we apply π to the path system $\pi(\mathscr{P})$ then the tail end of the paths P_s' and P_r' will switch back to the original paths P_s and P_r which means that

$$\pi(\pi(\mathscr{P})) = \mathscr{P}$$

Conclusion:

Therefore, by the reasoning outlined in our strategy,

$$\sum_{\mathscr{P}^{\times}} sign(\mathscr{P}^{\times}) w(\mathscr{P}^{\times}) = 0$$

And this implies that

$$det(M) = \sum_{\mathscr{P}^{*}} sign(\mathscr{P}^{*}) w(\mathscr{P}^{*})$$

6.3 Cauchy-Binet Theorem

Let $r \leq s \in \mathbb{N}$ and let X be an $r \times s$ matrix and Y be an $s \times r$ matrix.

6.3.1 Notation:

$\binom{[s]}{r}$ denotes the set of all possible sets of r indices chosen from the set of indices $\{1, 2, 3, ..., s\}$.

For any $Z \in \binom{[r]}{s}$, then

$X_{[r],Z}$ denotes the $r \times r$ matrix whose columns are the columns of X at the indices in Z.

$Y_{Z,[r]}$ denotes the $r \times r$ matrix whose rows are the rows of Y at the indices in Z.

Example to Illustrate Notation:

Suppose that

$$X = \begin{pmatrix} 4 & 7 & -1 \\ 8 & -3 & 5 \end{pmatrix} \qquad Y = \begin{pmatrix} 2 & -6 \\ 3 & 0 \\ 2 & 7 \end{pmatrix}$$

Because $r = 2$ and $s = 3$, then

$$\binom{[s]}{2} = \binom{[3]}{2} = \{\{1,2\},\{2,3\},\{1,3\}\}$$

Choose $Z \in \binom{[3]}{2}$ to be $Z = \{1,3\}$, then

$$X_{[2],Z} = \begin{pmatrix} 4 & -1 \\ 8 & 5 \end{pmatrix}$$

$$Y_{Z,[2]} = \begin{pmatrix} 2 & -6 \\ 2 & 7 \end{pmatrix}$$

6.3.2 Theorem Statement:

The Cauchy-Binet Theorem states that:

$$det(XY) = \sum_{Z \in \binom{[s]}{r}} det(X_{[r],Z}) det(Y_{Z,[r]})$$

Example to Illustrate Theorem Statement:

Using the same definitions of A and B as the example above,

$$det(XY) = \begin{vmatrix} 4 & 7 \\ 8 & -3 \end{vmatrix} \cdot \begin{vmatrix} 2 & -6 \\ 3 & 0 \end{vmatrix} + \begin{vmatrix} 4 & -1 \\ 8 & 5 \end{vmatrix} \cdot \begin{vmatrix} 2 & -6 \\ 2 & 7 \end{vmatrix} + \begin{vmatrix} 7 & -1 \\ -3 & 5 \end{vmatrix} \cdot \begin{vmatrix} 3 & 0 \\ 2 & 7 \end{vmatrix}$$

$$det(XY) = (-68)(18) + (28)(26) + (32)(21) = 176$$

This is easily verified because

$$det(XY) = \left| \begin{pmatrix} 4 & 7 & -1 \\ 8 & -3 & 5 \end{pmatrix} \begin{pmatrix} 2 & -6 \\ 3 & 0 \\ 2 & 7 \end{pmatrix} \right| = \begin{vmatrix} 27 & -31 \\ 17 & -13 \end{vmatrix} = 176$$

6.3.3 Proof:

Create a directed bipartite graph from the set of vertices $A = \{A_1, A_2, ..., A_r\}$ to $B = \{B_1, B_2, ..., B_s\}$ such that the edge from A_i to B_j is weighted with weight X_{ij}. Then extend the graph so there is a bipartite graph from $B = \{B_1, B_2, ..., B_s\}$ to $C = \{C_1, C_2, ..., C_r\}$ such that the edge from B_j to C_k is weighted with weight Y_{jk}.

Consider all the possible paths from A_i to C_k. Such a path must originate at A_i, pass through some B_j, and then end at C_k. So, this path includes two edges, namely the ones weighted X_{ij} and Y_{jk}. Therefore, the weight of the path from A_i to B_j to C_k is

$$w(P_{ijk}) = X_{ij}Y_{jk}$$

Because A and C both have r elements, the path matrix M of this graph is an $r \times r$ matrix.

Furthermore, the M_{ik} entries of the matrix can be calculated as follows because the vertex B_j that the path passes through can be any B_j such that $1 \leq j \leq s$, then the number of possible paths from $A_i \to C_k$ is

$$M_{ik} = \sum_{1 \leq j \leq n} X_{ij}Y_{jk}$$

Notice that the expression above is equal to $(XY)_{ij}$ by the definition of matrix multiplication.

Therefore, XY is the path matrix for the graph we generated from X and Y.

We know from the Gessel Viennot Lemma that when $\mathscr{P}^*$ is a vertex-disjoint path system from A to C, then

$$det(XY) = \sum_{\mathscr{P}^*} sign(\mathscr{P}^*)w(\mathscr{P}^*)$$

For any given $\mathscr{P}^*$, the vertex B_j must be unique for each $A_i \to B_j \to C_{\sigma(i)}$ because all the paths in $\mathscr{P}^*$ are vertex-disjoint. Therefore, every $\mathscr{P}^*$ has a corresponding subset of B:

$$\mathscr{B} = \{B_{j_1}, B_{j_2}, ..., B_{j_r} : A_i \to B_{j_i} \to C_{\sigma(i)} \in \mathscr{P}^*\}$$

Note that the indices $J = \{j_1, j_2, ..., j_r\} \in \binom{[s]}{r}$.

Furthermore, we can think of $\mathscr{P}^*$ as a "composition" of two path systems $\mathscr{P}_x$ and $\mathscr{P}_y$ where $\mathscr{P}_x$ is a path system from $A \to \mathscr{B}$ and $\mathscr{P}_y$ is a path system from $\mathscr{B} \to C$ defined by the concatenated path matrices $\mathscr{M}_{X_J}$ and $\mathscr{M}_{Y_J}$.

For any given $\mathscr{P}^*$,

$$w(\mathscr{P}^*) = \prod_{i=1}^{r} w(P_{ij_i\sigma(i)}) = \prod_{i=1}^{r} X_{ij_i} Y_{j_i\sigma(i)} = \prod_{i=1}^{r} X_{ij_i} \prod_{i=1}^{r} Y_{j_i\sigma(i)} = w(\mathscr{P}_x)w(\mathscr{P}_y)$$

And because $sign(\sigma \circ \tau) = sign(\sigma)sign(\tau)$ for any permutations σ, τ, then

$$sign(\mathscr{P}) = sign(\mathscr{P}_x)sign(\mathscr{P}_y)$$

Therefore,

$$det(XY) = \sum_{\mathscr{P}_x, \mathscr{P}_y} sign(\mathscr{P}_x)sign(\mathscr{P}_y)w(\mathscr{P}_x)w(\mathscr{P}_y)$$

$$det(XY) = \sum_{\mathscr{P}_x, \mathscr{P}_y} sign(\mathscr{P}_x)w(\mathscr{P}_x)sign(\mathscr{P}_y)w(\mathscr{P}_y)$$

And for a given subset $\mathscr{B} \subset B$ of size r and every $\mathscr{P}_x$ from $A \to \mathscr{B}$, there are multiple terms in the sum that contain $\mathscr{P}_x$ because there are multiple $\mathscr{P}_y$ that go from $\mathscr{B} \to C$. Therefore,

$$det(XY) = \sum_{\mathscr{B}} \left(\sum_{\mathscr{P}_x} sign(\mathscr{P}_x)w(\mathscr{P}_x) \sum_{\mathscr{P}_y} sign(\mathscr{P}_y)w(\mathscr{P}_y) \right)$$

Therefore, by the Gessel-Viennot Lemma,

$$det(XY) = \sum_{\mathscr{B}} det(X_{[r,\mathscr{B}]})det(Y_{[\mathscr{B},r]})$$

Because $\binom{[s]}{r}$ includes all of the possible subsets $\mathscr{B}$, then

$$det(XY) = \sum_{Z \in \binom{[s]}{r}} det(X_{[r],Z})det(Y_{Z,[r]})$$

6.4 Kirchoff's Theorem

6.4.1 Definitions

Let G be a simple, undirected graph.

Adjacency Matrix

If G has n vertices and m edges, let $A(G)$ be the $n \times m$ adjacency matrix of G where the rows of $A(G)$ correspond to the vertices of G, the columns of $A(G)$ correspond to the edges of G and the entries $A(G)_{ij}$ are 1 if the i^{th} vertex is in the j^{th} edge and 0 otherwise.

Directed Adjacency Matrix

It is obvious that each column of $A(G)$ will contain exactly two entries of 1 because every edge is incident to exactly two vertices. We can assign an arbitrary direction to all of the edges in G by changing one of the 1's in each column to a -1. Let $D(G)$ be the $n \times m$ arbitrary directed adjacency matrix of G.

Laplacian Matrix

Let $\mathscr{L}(G) = D(G)D(G)^T$ denote the $n \times n$ Laplacian matrix of G and let $\mathscr{L}_i(G)$ denote the $n-1 \times n-1$ submatrix of $\mathscr{L}(G)$ where the i^{th} row and column are deleted. This can also be thought of as $\mathscr{L}_i(G) = D_i(G)D_i(G)^T$ where $D_i(G)$ is the submatrix of $D(G)$ where the i^{th} row is deleted. In the graph, this is essentially equivalent to deleting the vertex i which leaves all of the edges incident to v_i without an endpoint.

Because of the way $D(G)$ is defined, the rows and columns of the $\mathscr{L}(G)$ correlated to the n vertices in G and if vertices $v_i \neq v_j$ are adjacent to each other, then $\mathscr{L}(G)_{ij} = -1$ and the diagonal entries $\mathscr{L}(G)_{ii}$ correspond to the degree of vertex v_i. This result shows that the Laplacian matrix does not depend on which arbitrary direction is assigned to each edge.

Example:

Let G be the graph below:

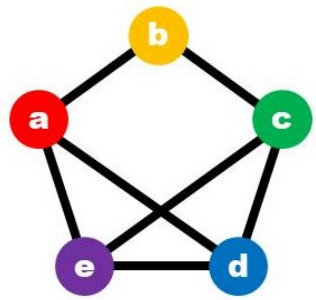


Figure 30

The adjacency matrix of G is:

$A(G) =$

	(ab)	(ad)	(ae)	(bc)	(cd)	(ce)	(de)
a	1	1	1	0	0	0	0
b	1	0	0	1	0	0	0
c	0	0	0	1	1	1	0
d	0	1	0	0	1	0	1
e	0	0	1	0	0	1	1

By arbitrarily assigning a $-$ to one of the 1's in each column, we get

$D(G) =$

	(ab)	(ad)	(ae)	(bc)	(cd)	(ce)	(de)
a	-1	1	-1	0	0	0	0
b	1	0	0	-1	0	0	0
c	0	0	0	1	1	1	0
d	0	-1	0	0	-1	0	-1
e	0	0	1	0	0	-1	1

Which corresponds to the following graph:

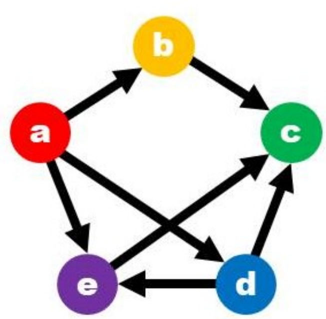


Figure 31

And the Laplacian matrix is calculated as:

$$\mathscr{L}(G) = \begin{pmatrix} -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & -1 & -1 & 3 \end{pmatrix}$$

6.4.2 Theorem Statement:

For any graph G and all $i = 1, 2, ..., n$, we have $|T(G)| = det\ \mathscr{L}_i(G)$.

6.4.3 Proof:

Let $D(G)$ be an arbitrary directed adjacency matrix of the graph G.

Let $1 \leq i \leq n$.

Because $\mathscr{L}_i(G) = D_i(G)d_i(G)^T$, then by the Cauchy-Binet Theorem,

$$det\ \mathscr{L}_i(G) = \sum_N det(N)det(N^T)$$

Where N runs through all the $n-1 \times n-1$ submatrices of $D_i(G)$.

Because $det(A) = det(A^T)$ for any matrix A, then

$$det\ \mathscr{L}_i(G) = \sum_N det(N)^2$$

Any $n-1 \times n-1$ submatrix N corresponds to a subgraph of G that has $n-1$ edges on n vertices.

This is because the $n-1$ rows correspond to the $n-1$ vertices that are not vertex v_i and the $n-1$ columns correspond to $n-1$ of the edges in G that are incident to those vertices.

So, the submatrix N will have $n-1$ vertices incident with $n-1$ edges and some of those edges could be missing an endpoint because the row corresponding to v_i was deleted from $D(G)$. By simply adding the vertex v_i as the missing endpoint to these edges, a subgraph of G that has $n-1$ edges on n vertices can be generated even though the v_i vertex is not explicitly represented in the matrix.

Therefore, $|\{N\}|$ represents the number of subgraphs of G that have $n-1$ edges. Some of those graphs will be trees and others will not.

To show that $|T(G)| = det\ \mathscr{L}_i(G) = \sum_N det(N)^2$, we will show that

$$det(N) = \begin{cases} \pm 1 & \text{if N represents a tree} \\ 0 & \text{otherwise} \end{cases}$$

Prove that the determinant of a matrix that does not represent a tree is zero

Suppose that for some matrix N_0, the $n-1$ edges represented by N_0 do not form a tree on n vertices.

If a graph on n vertices has $n-1$ edges and it is not a tree, then there are at least two connected components in the graph. Therefore, there exists a component that does not contain v_i.

If we add the rows in N_0 that correspond to the vertices that are in this component, they will sum to 0 because none of the edges in that component are missing an endpoint and all edges are directed with 1's and -1's.

Therefore, N_0 is linearly independent so $det(N_0) = 0$ when N_0 does not form a tree.

Prove that the determinant of a matrix that represent a tree is ± 1

Suppose that for some matrix N_1, the $n-1$ edges represented by N_1 form a tree on n vertices.

Then because every tree has at least two leaves, there exists a vertex $w_1 \neq v$ of degree 1.

Let the edge that attaches w_1 to the graph be relabeled e_1.

Delete w_1 and e_1 to get a tree on $n-1$ vertices with $n-2$ edges.

By repeating the reasoning above with the new tree we can relabel all the vertices except v_i as $\{w_1, ..., w_{n-1}\}$ and their corresponding edges $\{e_1, ..., e_{n-1}\}$.

These relabeled vertices and edges generate a new $n-1 \times n-1$ matrix N' that is lower triangular because $w_j \notin e_i$ for any $j < i$ by construction.

Furthermore, the diagonal entries are all 1 or -1 depending on the orientation of the edges.

Therefore $det(N_1') = 1$ or $det(N_1') = -1$.

And because N_1' and N_1 describe the same graph with different labels, N_1' can be found by permuting rows and columns of N_1 so $det(N_1) = det(N_1') = \pm 1$.

Count the trees in G

The fact that $det(N) = \pm 1$ for all subgraphs of G with $n-1$ edges that are trees and $det(N) = 0$ for all other subgraphs with $n-1$ edges implies that the following sum counts the number of trees in G,

$$T(G) = det\ \mathscr{L}_i(G) = \sum_N det(N)^2$$

6.5 Applying Kirchoff's Theorem to a Complete Graph

Let K_n represent the complete graph on n vertices.

Because every one of the n vertices shares an edge with all the other $n-1$ vertices, the degree of every vertex is $n-1$.

Therefore, the Laplacian matrix $\mathscr{L}(K_n)$ is an $n \times n$ matrix with -1 in all non-diagonal entries and $n-1$ in all the diagonal entries.

$$\mathscr{L}(K_n) = \overbrace{\begin{pmatrix} n-1 & -1 & \dots & -1 \\ -1 & n-1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & \dots & -1 & n-1 \end{pmatrix}}^{n} \Big\}n$$

Therefore, $\mathscr{L}_i(K_n)$ is an $n-1 \times n-1$ matrix with the same structure

$$\mathscr{L}_i(K_n) = \overbrace{\begin{pmatrix} n-1 & -1 & \dots & -1 \\ -1 & n-1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & \dots & -1 & n-1 \end{pmatrix}}^{n-1} \Big\}n-1$$

And we can calculate the number of trees in K_n as follows:

$$|T(K_n)| = det(\mathscr{L}_i(K_n))$$

The matrix $\mathscr{L}_i(K_n)$ can be brought to upper triangular form by first adding all the columns to the first column and then subtracting the first row from all the other rows.

$$det(\mathscr{L}_i(K_n)) = det\left(\begin{pmatrix} 1 & -1 & \dots & -1 \\ 1 & n-1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \dots & -1 & n-1 \end{pmatrix}\right) = det\left(\begin{pmatrix} 1 & -1 & \dots & -1 \\ 0 & n & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & n \end{pmatrix}\right)$$

Therefore, because there are $n-1$ entries in the diagonal where the first entry of the diagonal is a 1 and the remaining $n-2$ entries are n, then by Kirchoff's Theorem,

$$|T(K_n)| = det(\mathscr{L}_i(K_n)) = n^{n-2}$$

7 Probability Theory Proofs

On the simplest level, probability is the number of favorable outcomes divided by the number of possible outcomes. We can use probability theory to count objects by calculating the probability of constructing the desired objects at random and then interpreting the probability to count the number of desired objects.

7.1 Poisson Distribution

The following proof is based on the proof found in Hofstad's *Random Graphs and Complex Networks Volume 1* [6] In this proof we build a family tree of n vertices with a Poisson branching process, label the vertices of the tree and then calculate the probability of randomly constructing any given labeled tree by this process.

The set up for this proof is fairly intensive but once the terms and labeling process are defined, the proof itself is relatively simple.

7.1.1 Definitions

Family Trees

A family tree is an unlabeled tree with a specified root and a family-like structure between the vertices. Originally, family trees were used to represent the paternal ancestry of the European aristocracy so the family tree structure only accounts for one parent in each family.

The i^{th} generation of the family tree are all the vertices that have a path of length i to the root.

The vertices in the first generation are called the "children" of the root.

The vertices in the second generation are called the "grandchildren" of the root and they can also be called the "children" of the vertices in the first generation.

If vertex b is a child of vertex a, then a is the parent of b.

If vertices b and c are both children of vertex a, then b and c are siblings. It is important to note that the structure of a family tree preserves the sibling order of vertices.

The Ulam-Harris representation of trees encodes family trees as sets of words where each vertex is "labeled" with a word w that describes its location in the structure of the family tree. Two family trees are considered to be the same tree if and only if they are represented by the same set of words.

The notation d_w is used to denote the number of children that the vertex w has and it is clear that $d_w = degree(w) - 1$ because the vertex represented by the word w has 1 parent and the remaining vertices it is adjacent to are its children.

1. The root of the tree is represented with the empty word $\emptyset$.
2. The children of the root are represented with the words $(1), (2), ..., (d_\emptyset)$ where $d_\emptyset$ denotes the number of children that the root vertex has.
3. The grandchildren of the root are represented with the words $(11), (12), ..., (1d_1)$ and $(21), (22), ..., (2d_2)$ and $(d_\emptyset 1), (d_\emptyset 2), ..., (d_\emptyset d_{d_\emptyset})$ where the first letter in their word is the word of their parent and the last letter is their sibling order.
4. The great-grandchildren of the root whose parents are (11) and $(2d_2)$ would be represented with the words $(111), (112), ..., (11d_{11})$ and $(2d_2 1), (2d_2 2), ..., (2d_2 d_{2d_2})$ respectively.
5. Iterate this process for all the generations of the family tree such that each vertex in the i^{th} generation is labeled with a word that is i letters long and the first $i - 1$ letters in their word is the word of their parent and the last letter in their word represents their sibling order.

These "labels" are not true labels in the sense that the "labels" of two different vertices cannot be swapped and maintain the structure of the family tree because the "labels" are generated by the vertex's location in the structure.

As an example, consider the vertices (3) and (12) in the family tree described by the set of words $\{\emptyset, (1), (2), (3), (11), (12), (111), (112), (113), (114), (121), (1211), (1212), (1213)\}$.

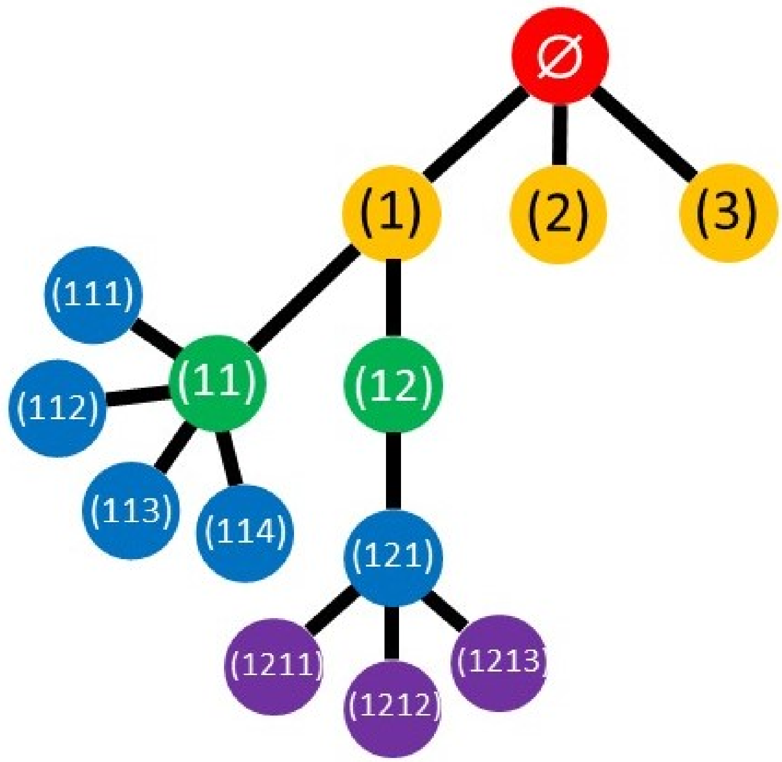


Figure 32

If you were to switch the "labels" of the vertices (3) and (12), then the "labels" would no longer accurately describe the structure of the family tree.

Poisson Branching Process

Let x be a Poisson random variable.

Poisson distributions describe the probability of x number of events happening when the events happen at a known constant mean rate λ and the events happen independently of each other.

The probability generating function for a Poisson distribution is:

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

The Poisson branching process can generate a family tree F where a collection of random variables x describes the number of children d_w that each vertex $w \in F$ has.

Progeny

If T is a family tree generated by a Poisson branching process, then the total progeny T^* of T is the total number of vertices in the family tree.

The probability of generating a family tree T such that the distribution of having x offspring is a Poisson distribution with mean rate λ and the total progeny of the tree is $T^* = n$ is

$$P(T^* = n) = \frac{(\lambda n)^{n-1}}{n!} e^{-\lambda n}$$

Labeling Family Trees

For any family tree T such that $T^* = n$, we can create a labeled tree L by the following process:

1. Label the root vertex with the label v_1.

2. Label the non-root vertices with unique labels from $\{v_2, v_3, ..., v_n\}$ uniformly and at random.

Size of a Labeled Tree

For a labeled tree L, we will use the notation $|L|$ to denote the number of vertices in L.

Isomorphic Family Trees and Labeled Trees

For a labeled tree L and a family tree T, we say that $T \sim L$ if L can be generated by labeling T.

It is obvious that if $T \sim L$, then $T^* = |L|$.

For any $T \sim L$, let the notation $N(T, L)$ denote the number of ways to label T such that the resulting labeled tree is isomorphic to L. By the justification provided in the proof of Lemma 1 below, this notation will be simplified to $N(L)$ for the majority of this paper.

At first glance, it may seem like $N(T, L)$ should always be 1. However, because the sibling order doesn't matter in a labeled tree, $N(T, L)$ is often greater than 1.

For example, let T and L be the following graphs:

Figure 33

When assigning labels to the children of the root vertex, the label v_6 must be assigned to (1) because (1) is the only child that has the correct number of grandchildren. But the labels v_4 and v_5 can be assigned to either (2) or (3) because their (non-existent) children are isomorphic to each other.

By similar reasoning, when assigning labels to the grandchildren of the root vertex, the labels v_2 and v_3 can be assigned to either (11) or (12).

Therefore, there are 4 ways to assign labels to the non-root vertices of T so $N(T, L) = 4$.

	Labeling A	Labeling B	Labeling C	Labeling D
($\emptyset$)	v_1	v_1	v_1	v_1
(1)	v_6	v_6	v_6	v_6
(2)	v_4	v_5	v_4	v_5
(3)	v_5	v_4	v_5	v_4
(11)	v_3	v_3	v_2	v_2
(12)	v_2	v_2	v_3	v_3

Table 5

7.1.2 Properties of Isomorphic Family Trees and Labeled Trees

Lemma 1:

If l is a labeled tree and t_1, t_2 are family trees such that $t_1 \sim l$ and $t_2 \sim l$, then $N(t_1, l) = N(t_2, l)$.

Proof:

The number $N(t, l)$ represents the number of unique ways that the family tree t can be labeled to generate the labeled tree l.

This can also be interpreted as the number of symmetries within the labeled tree l.

The number of symmetries in l doesn't depend on which isomorphic family tree is being labeled.

Therefore, it is clear that if $t_1 \sim l$ and $t_2 \sim l$, then $N(t_1, l) = N(t_2, l)$.

Example:

Let t_1, t_2, and l be the following trees:

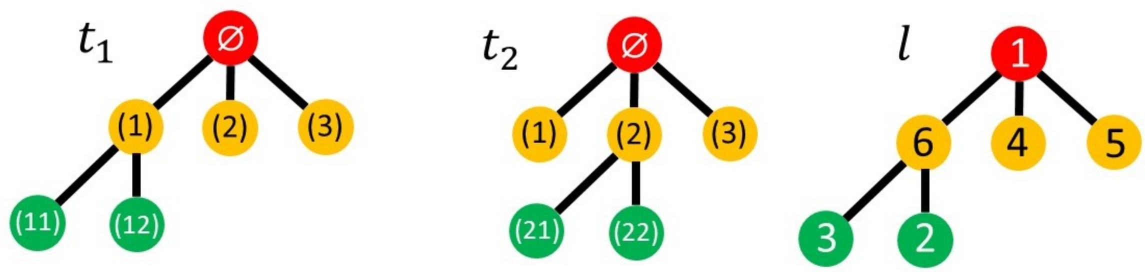


Figure 34

Even though the sibling order in t_1 is different than the sibling order in t_2, the number of possible labelings that generate l is 4 in both cases because the labeled tree does not preserve sibling order and therefore has 4 symmetries.

	A	B	C	D
(∅)	v_1	v_1	v_1	v_1
(1)	v_6	v_6	v_6	v_6
(2)	v_4	v_5	v_4	v_5
(3)	v_5	v_4	v_5	v_4
(11)	v_3	v_3	v_2	v_2
(12)	v_2	v_2	v_3	v_3

	A	B	C	D
(∅)	v_1	v_1	v_1	v_1
(1)	v_6	v_6	v_6	v_6
(2)	v_4	v_5	v_4	v_5
(3)	v_5	v_4	v_5	v_4
(21)	v_3	v_3	v_2	v_2
(22)	v_2	v_2	v_3	v_3

Table 6

Notation:

Because the value of $N(t, l)$ is constant for any given labeled tree l regardless of which family tree $t \sim l$ is chosen to be labeled, I will simplify the notation $N(t, l)$ and write it as $N(l)$ from now on.

Lemma 2:

For any labeled tree l, the number of family trees isomorphic to l is

$$|\{t : t \sim l\}| = \frac{1}{N(l)} \prod_{w \in l} d_w!$$

Proof:

Let l be a labeled tree and let $t_0 \in \{t : t \sim l\}$ be a family tree that is isomorphic to l.

The family tree t_0 can be converted into any other $t_i \in \{t : t \sim l\}$ by permuting the sibling orders in t_0.

For any vertex $w \in t_0$, the number of children that w has is denoted by d_w so there are $d_w!$ ways to permute the sibling order of w's children. This means that there are $\prod_{w \in t_0} d_w!$ ways to permute the sibling orders in the entire family tree.

And because there are $N(l)$ symmetries in the way that l can be labeled,

$$|\{t : t \sim l\}| = \frac{1}{N(l)} \prod_{w \in t_0} d_w!$$

For any vertex $w \in t_0$ there is a corresponding vertex in $v \in l$ that has the same degree and $d_w = degree(w) - 1$ so if you define $d_v = deg(v) - 1$ then the product can be indexed by vertices in l instead of t_0.

$$|\{t : t \sim l\}| = \frac{1}{N(l)} \prod_{w \in l} d_w!$$

By similar reasoning, you could also index the product by vertices in any of the other $t \sim l$.

$$|\{t : t \sim l\}| = \frac{1}{N(l)} \prod_{w \in t} d_w!$$

7.1.3 Calculate Probabilities

Let l be a labeled tree with vertices $\{v_1, v_2, ..., v_n\}$.

Let t be a family tree such that $t \sim l$. Because $t \sim l$, then t also has n vertices.

Let T be a family tree generated by a Poisson branching process with $\lambda = 1$.

Label the vertices of T to produce a labeled tree L.

Show that the probability that T is t is:

$$P(T = t) = \frac{e^{-n}}{\prod_{w \in t} d_w!}$$

In order for T to be equal to the tree t, then the collection of random variables x in the Poisson branching process must be equal to d_w for all $w \in t$.

Therefore, the probability that $T = t$ is:

$$P(T = t) = \prod_{w \in t} P(x = d_w)$$

The probability generating function for a Poisson distribution is:

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

And because T is generated with a Poisson branching process with $\lambda = 1$ then

$$P(T = t) = \prod_{w \in t} P(x = d_w) = \prod_{w \in t} \frac{e^{-1}}{d_w!}$$

The factor e^{-1} does not depend on $w \in t$ and there are n vertices in t.

Therefore, the product can be written as:

$$P(T = t) = \frac{e^{-n}}{\prod_{w \in t} d_w!}$$

Show that if t is labeled, the probability that t receives the labels l is:

$$P(t \text{ receives labels } l) = \frac{N(l)}{(n-1)!}$$

Because the root vertex $\emptyset$ in t is always given the label v_1 in the labeling process, there are $(n-1)!$ ways to randomly assign the labels $\{v_2, v_3, ..., v_n\}$ to the remaining vertices of t.

Therefore, the probability that an arbitrary labeling of t called m is the labeled tree l is:

$$P(m = l) = \frac{1}{(n-1)!}$$

There are $N(l)$ such labelings, therefore, the probability that t receives the labels l is:

$$P(t \text{ receives labels } l) = \frac{N(l)}{(n-1)!}$$

Show that the probability that L is l is:

$$P(L = l) = \frac{e^{-n}}{(n-1)!}$$

In order for L to be equal to l, there are two conditions that must be met.

1. Because L is defined as a labeling of T, then it is necessary that $T = t_l$ where $t_l \sim l$.

2. When that t_l is labeled, it must receive the labels l.

It is possible that there are multiple such T_l, so

$$P(L = l) = \sum_{t_l \sim l} P(T = t_l) P(t_l \text{ receives labels } l)$$

By substituting in the results found above,

$$P(L = l) = \sum_{t_l \sim l} \frac{e^{-n}}{\prod_{w \in t_l} d_w!} \frac{N(l)}{(n-1)!}$$

Because the terms in the sum are equal for all $t_l \in \{t : t \sim l\}$, then

$$P(L = l) = |\{t : t \sim l\}| \frac{e^{-n}}{\prod_{w \in t} d_w!} \frac{N(l)}{(n-1)!}$$

We know that $|\{t : t \sim l\}| = \frac{1}{N(l)} \prod_{w \in l} d_w!$ so

$$P(L = l) = \frac{1}{N(l)} \prod_{w \in t} d_w! \frac{e^{-n}}{\prod_{w \in t} d_w!} \frac{N(l)}{(n-1)!}$$

Simplifying the expression gives:

$$P(L = l) = \frac{e^{-n}}{(n-1)!}$$

Show that the probability that $T^* = n$ is:

$$P(T^* = n) = \frac{e^{-n} n^{n-2}}{(n-1)!}$$

By definition, the probability that the total progeny for a tree R generated by a Poisson branching process is n can be calculated as:

$$P(R^* = n) = \frac{(\lambda n)^{n-1}}{n!} e^{-\lambda n}$$

Since T was generated by a Poisson branching process with $\lambda = 1$, then

$$P(T^* = n) = \frac{n^{n-1}}{n!} e^{-n} = \frac{n^{n-2} e^{-n}}{(n-1)!}$$

Show that conditioning on $T^* = n$, the probability that $L = l$ is:

$$P(L = l \big| T^* = n) = \frac{1}{n^{n-2}}$$

By the definition of conditional probability,

$$P(L = l \big| T^* = n) = \frac{P(L = l)}{P(T^* = n)}$$

Therefore,

$$P(L = l \big| T^* = n) = \frac{e^{-n}}{(n-1)!} \frac{(n-1)!}{n^{n-2} e^{-n}} = \frac{1}{n^{n-2}}$$

7.1.4 Prove Cayley's Theorem

The probability found above is uniform over all labeled trees.

Therefore, the number of labeled trees with n vertices is

$$P(L = l \Big| |L| = n)^{-1} = n^{n-2}$$

7.2 Katz - Indecomposable Functional Digraphs

In 1955, Katz [8] published a paper that categorized functional digraphs as decomposable or indecomposable and then he further categorized indecomposable digraphs by their cycles and cycle length.

In his paper, he counted indecomposable digraphs by calculating the probability of building an indecomposable digraph by defining the function with a random mapping. His approach can be used to prove Cayley's Theorem because trees are a specific type of indecomposable digraph.

7.2.1 Definitions

Functional Digraph - A functional digraph on n vertices is a directed graph defined by a function $f : [n] \to [n]$ where $f(i) = j_i$ defines the directed edge from $i \to j_i$.

Components of a Functional Digraph - For any function $f : [n] \to [n]$, f decomposes the set $[n]$ into k minimal, disjoint, non-null, invariant subsets $\omega_1 \sqcup \omega_2 \sqcup ... \sqcup \omega_k = [n]$ such that $f(\omega_i) \subset \omega_i$ and $f^{-1}(\omega_i) \subset \omega_i$. These k subsets form the k components of the digraph.

Indecomposable Functional Digraph - A functional digraph is indecomposable if it contains only one component.

7.2.2 Katz-Frame Lemma:

The following lemma was proved in Katz's paper [8] and he credited part of the proof to J.S. Frame.

For $1 \leq m \leq n$, and partitions $P(p)$ of size p such that $n_1 + n_2 + ... + n_p = n - m$, then

$$\sum_{p} \sum_{P(p)} \frac{1}{m} \frac{m^{n_1} n_1^{n_2} ... n_{p-1}^{n_p}}{n_1! n_2! ... n_p!} = \frac{n^{n-m-1}}{(n-m)!}$$

Proof: Let $M_0 = n - m$

$$\frac{n^{n-m-1}}{(n-m)!} = \frac{(n-m+m)^{n-m-1}}{(n-m)!} = \frac{(M_0+m)^{M_0-1}}{M_0!}$$

By the binomial theorem,

$$\frac{n^{n-m-1}}{(n-m)!} = \frac{1}{M_0!}\sum_{n_1=1}^{M_0}\binom{M_0-1}{n_1-1}M_0^{M_0-n_1}m^{n_1-1}$$

Expand the binomial coefficient and simplify to get:

$$\frac{n^{n-m-1}}{(n-m)!} = \sum_{n_1=1}^{M_0}\frac{1}{M_0!}\frac{(M_0-1)!}{(n_1-1)!(M_0-1-n_1+1)!}M_0^{M_0-n_1}m^{n_1-1}$$

$$\frac{n^{n-m-1}}{(n-m)!} = \sum_{n_1=1}^{M_0}\frac{M_0^{M_0-n_1}}{M_0}\frac{1}{(n_1-1)!(M_0-n_1)!}m^{n_1-1}$$

$$\frac{n^{n-m-1}}{(n-m)!} = \sum_{n_1=1}^{M_0}\frac{m^{n_1-1}}{(n_1-1)!}\frac{M_0^{M_0-n_1-1}}{(M_0-n_1)!}$$

Then let $M_1 = M_0 - n_1$ to get

$$\frac{n^{n-m-1}}{(n-m)!} = \sum_{n_1=1}^{M_0}\frac{m^{n_1-1}}{(n_1-1)!}\frac{(M_1+n_1)^{M_1-1}}{M_1!}$$

By applying the binomial theorem, expanding, and simplifying again where $M_2 = M_1 - n_2$, we get

$$\frac{n^{n-m-1}}{(n-m)!} = \sum_{n_1=1}^{M_0}\frac{m^{n_1-1}}{(n_1-1)!}\sum_{n_2=1}^{M_1}\frac{n_1^{n_2-1}}{(n_2-1)!}\frac{(M_2+n_2)^{M_2-1}}{M_2!}$$

By repeating this process and letting $M_i = M_{i-1} - n_i$, we get

$$\frac{n^{n-m-1}}{(n-m)!} = \sum_{n_1=1}^{M_0}\frac{m^{n_1-1}}{(n_1-1)!}\sum_{n_2=1}^{M_1}\frac{n_1^{n_2-1}}{(n_2-1)!}\cdots\sum_{n_p=1}^{M_{p-1}}\frac{n_{p-1}^{n_p-1}}{(n_p-1)!}n_p^{-1}$$

Where p is arbitrary.

The structure of the summations is equivalent to the sum over all p and $P(p)$, therefore,

$$\frac{n^{n-m-1}}{(n-m)!} = \sum_{p}\sum_{P(p)}\frac{m^{n_1-1}}{(n_1-1)!}\frac{n_1^{n_2-1}}{(n_2-1)!}\cdots\frac{n_{p-1}^{n_p-1}}{(n_p-1)!}n_p^{-1}$$

Multiply the summation term by $\frac{mn_1n_2...n_{p-1}}{mn_1n_2...n_{p-1}}$ and we get:

$$\frac{n^{n-m-1}}{(n-m)!} = \sum_{p} \sum_{P(p)} \frac{1}{m} \frac{m^{n_1} n_1^{n_2} ... n_{p-1}^{n_p}}{n_1! n_2! ... n_p!}$$

7.2.3 Proof

Characterize Indecomposable Digraphs by Cycles

Because each of the ω_i sets are finite and invariant, then each of the k components of a digraph contains a cycle because for any $x \in \omega_i$, the path $x \to f(x) \to f^2(x) \to ...$ eventually has to circle back to one of the $f^r(x)$.

Furthermore, each component in a functional digraph has exactly one cycle because if there were two or more cycles in the same component then there would be at least one vertex i that mapped to at least 2 other vertices and that contradicts the definition of a function.

Categorize Indecomposable Digraphs by Generation Sizes

Let the 0^{th} generation be the set of points that are in the cycle of a component. Let the 1^{st} generation be the set of points that map to a vertex in the 0^{th} generation. Let the 2^{nd} generation be the set of points that map to a vertex in the 1^{st} generation and so on.

Let m denote the length of the cycle or the number of vertices in the 0^{th} generation. Let $n_1, n_2, ...n_p$ denote the number of vertices in the 1^{st}, 2^{nd},..., p^{th} generations respectively.

Notice that $n_1, n_2, ...n_p$ is a partition of $n - m$.

Calculate the Probability of Building a Specific Indecomposable Functional Digraph

If the function f is defined by a random mapping, then $i \to f(i)$ with independent probability

$$P(i \to f(i)) = \frac{1}{n}$$

The probability of building an indecomposable functional digraph of type $(m, n_1, n_2, ..., n_p)$ is

$$\binom{n}{m, n_1, n_2, ..., n_p} \frac{(m-1)!}{n^m} \left(\frac{m}{n}\right)^{n_1} \left(\frac{n_1}{n}\right)^{n_2} ... \left(\frac{n_{p-1}}{n}\right)^{n_p}$$

The first factor comes from the fact that there are $\binom{n}{m,n_1,n_2,...,n_p}$ ways to choose which vertices are in each of the generations such that there are m vertices in the cycle, n_1 vertices in the 1^{st} generation, n_2 vertices in the 2^{nd} generation, and so on.

Once we know which m vertices need to be in the cycle, for any i in the cycle, there is a $\frac{m-1}{n}$ chance that $i \to f(i) = j$ such that j is also one of the points that is supposed to be in the cycle. Similarly, $\frac{m-2}{n}$ chance that $j \to f(j) = k$ such that k is another of the points that is supposed to be in the cycle. So, the second factor is the probability that the cycle of size m is formed with the specific vertices chosen in the first factor.

For every vertex i in the 1^{st} generation, there is a $\frac{m}{n}$ chance that $i \to f(i) = j$ maps to some j in the cycle. Because there are n_1 vertices in the 1^{st} generation, the third factor is the probability that the vertices chosen to be in the 1^{st} generation are mapped in such a way that they really are in the in the 1^{st} generation.

By the same reasoning, the remaining factors denote the probability that the n_k vertices in the k^{th} generation are mapped to vertices in the $(k-1)^{th}$ generation.

Expanding the multinomial coefficient and simplifying gives:

$$\binom{n}{m, n_1, n_2, ..., n_p} \frac{(m-1)!}{n^m} \left(\frac{m}{n}\right)^{n_1} \left(\frac{n_1}{n}\right)^{n_2} ... \left(\frac{n_{p-1}}{n}\right)^{n_p}$$

$$= \frac{n!}{m!n_1!n_2!...n_p!} \frac{(m-1)!m^{n_1}n_1^{n_2}...n_{p-1}^{n_p}}{n^n}$$

$$= \frac{n!}{n^n} \frac{(m-1)!}{m!} \frac{m^{n_1}n_1^{n_2}...n_{p-1}^{n_p}}{n_1!n_2!...n_p!}$$

$$= \frac{n!}{n^n} \frac{1}{m} \frac{m^{n_1}n_1^{n_2}...n_{p-1}^{n_p}}{n_1!n_2!...n_p!}$$

Calculate the Probability of Building any Indecomposable Functional Digraph

Because an indecomposable functional digraph can have a cycle of any length $1 \leq m \leq n$ and the $n - m$ vertices that are not in the cycle can be partitioned into any $1 \leq p \leq n - m$ number of generations and there are multiple ways those generations can be partitioned such that $n_1 + n_2 + ... + n_p = n - m$,

$$P(indecomposable) = \sum_{m=1}^{n} \sum_{p} \sum_{P(p)} \frac{n!}{n^n} \frac{1}{m} \frac{m^{n_1} n_1^{n_2} ... n_{p-1}^{n_p}}{n_1! n_2! ... n_p!}$$

Where $P(p)$ is a partition of $n - m$ with p parts.

The sums do not depend on n so, this is equal to

$$P(indecomposable) = \frac{n!}{n^n} \sum_{m=1}^{n} \sum_{p} \sum_{P(p)} \frac{1}{m} \frac{m^{n_1} n_1^{n_2} ... n_{p-1}^{n_p}}{n_1! n_2! ... n_p!}$$

By the lemma proved above, $\sum_p \sum_{P(p)} \frac{1}{m} \frac{m^{n_1} n_1^{n_2} ... n_{p-1}^{n_p}}{n_1! n_2! ... n_p!} = \frac{n^{n-m-1}}{(n-m)!}$ so

$$P(indecomposable) = \frac{n!}{n^n} \sum_{m=1}^{n} \frac{n^{n-m-1}}{(n-m)!}$$

7.2.4 Prove Cayley's Theorem

An indecomposable functional digraph with cycle length $m = 1$ is essentially a labeled, rooted tree where the root is designated as the single vertex contained in the cycle.

Therefore, the probability of randomly choose a function f that generates a labeled rooted tree is:

$$P(rooted\ tree) = \frac{n!}{n^n} \frac{n^{n-1-1}}{(n-1)!} = \frac{n^{n-1}}{n^n}$$

The total number of unique functions f that can be randomly chosen is n^n so this probability tells us that there are n^{n-1} rooted trees on n vertices.

$$R_n = n^{n-1}$$

Because there are n ways to choose a root on a tree with n vertices,

$$nT_n = R_n$$

Therefore, the number of trees on n vertices is:

$$T_n = \frac{R_n}{n} = \frac{n^{n-1}}{n} = n^{n-2}$$

References

[1] Abu-Sbeih, M. (1990). *On the Number of Spanning Trees of K_n and $K_{m,n}$.* Discrete Mathematics, 84, 205-207.

[2] Aigner, M. & Ziegler G. (2009). *Proofs from the Book: 4th Edition.* Springer, 195-206.

[3] Ariannejad, M. & Emami, M. (2014). *A New Proof of Cayley's Formula for Labeled Spanning Trees.* Electronic Notes in Discrete Mathematics, 45, 99-102.

[4] Cayley, A. (1889). *A Theorem on Trees.* The Quarterly Journal of Mathematics, 23, 376-378.

[5] Clarke, L.E. (1958). *On Cayley's Formula for Counting Trees.* London Mathematical Society, 4, 471-474.

[6] Hofstad, R. (2016). *Random Graphs and Complex Networks (Cambridge Series in Statistical and Probabilistic Mathematics).* Cambridge University Press, 100-103.

[7] Joyal, A. (1981). *Une Théorie Combinatorie des Séries Formelles.* Advances in Mathematics, 42, 1-82.

[8] Katz, L. (1955). *Probability of Indecomposability of a Random Mapping Function.* The Annals of Mathematical Statistics, 26(3), 512–517.

[9] Moon, J.W. (1970). *Counting Labelled Trees.* Canadian Mathematical Monographs, 1, 4-6.

[10] Pitman, J. (1997). *Coalescent Random Forests.* Journal of Combinatorial Theory, 85 (A), 165-193.

[11] Pólya, G. (1937). *Kombinatorische Anzahlbestimmungen fúr Gruppen, Graphen und chemische Verbindungen.* Acta Mathematica, 68, 145-254.

[12] Prüfer, H. (1918). *Neuer Beweis eines Satzes über Permutationen.* Archiv der Mathematik Und Physik, 27(3), 142-144.

[13] Rényi, A. (1956). *Some Remarks on the Theory of Trees.* Magyar Tud. Akad. Mat. Kutató Int. Közl, 4, 73-85.

[14] Riordan, J. (1968). *Forests of Labeled Trees.* Journal of Combinatorial Theory, 5, 90-103.

[15] Shukla, A. (2018). *A Short Proof of Cayley's Tree Formula.* The American Mathematical Monthly, 125(1), 65-68.

[16] Stanley, R. (1997). *Enumerative Combinatorics: Volume 2*. Cambridge University Press, 1-54.

[17] West, D.B. (2001). *Introduction to Graph Theory*. Prentice Hall, 1-83.

[18] Wilf, H.S. (1990). *generatingfunctionology*. Academic Press.