Graph Theory and Cayley's Formula

Chad Casarotto

August 10, 2006

Contents

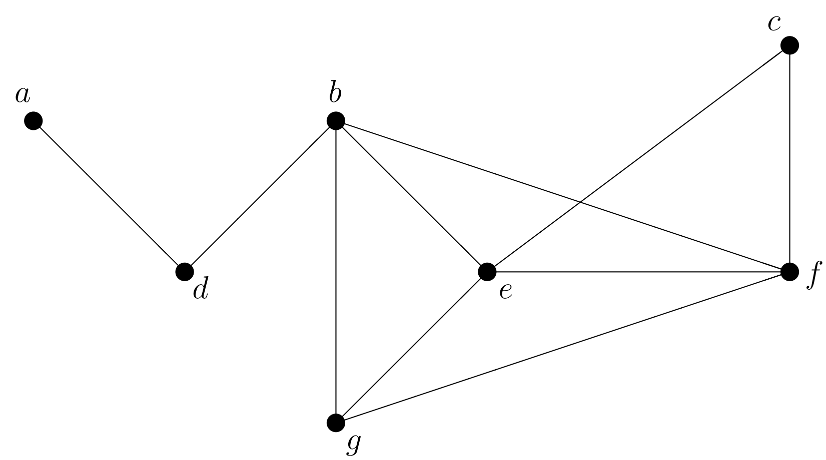
1 Introduction

In this paper, I will outline the basics of graph theory in an attempt to explore Cayley's Formula. Cayley's Formula is one of the most simple and elegant results in graph theory, and as a result, it lends itself to many beautiful proofs. I will examine a couple of these proofs and show how they exemplify different methods that are often used for all types of mathematics problems.

2 Basics and Definitions

Not surprisingly, graph theory is the study of things called graphs. So what is a graph? A *graph* can be thought of as a finite set of vertices $\{v_1, v_2, \ldots, v_n\}$ and a set of relations between each pair of vertices. Each of the $\binom{n}{2}$ pairs of vertices is either adjacent or not adjacent. If a pair of vertices is adjacent, we say there is an edge connecting the two vertices, and they are called neighbors.

The above definition lends itself to a convenient visualization. The n vertices can be considered as n distinct points in a plane, such as $\mathbb{R}^2$, and edges can be considered as lines between adjacent vertices. An example of a graph can be seen below.



A graph with 7 vertices and 10 edges

It is this visualized version of a graph that will be used from here on.
(*Note:* what is being defined here is formally called a simple graph; other types of graphs may have edges with arrows, multiple edges between vertices, or loops from a vertex to itself.)

Let's define some features of graphs:

- A *walk* of length k is a sequence of vertices $v_0, v_1, \ldots v_k$ such that v_i is adjacent to v_{i+1} for each i (for this we shall write: $v_i \sim v_{i+1} \; \forall \, i$).

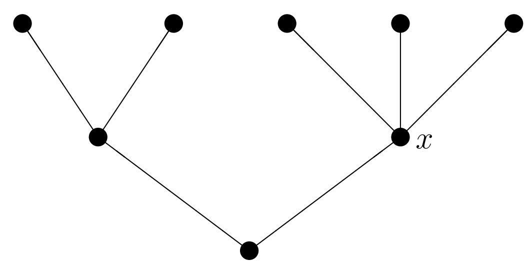
- A *closed walk* is a walk of length k such that $v_0 = v_k$.

- A *cycle* is a closed walk where none of the vertices repeat except for the first and the last (i.e. $i \neq j \Rightarrow v_i \neq v_j$ except when $(i, j) = (0, k)$).
  Example: in the above graph, the sequence b, f, g, b forms a cycle of length 3, denoted C_3.

- A *complete graph* on n vertices is a graph such that $v_i \sim v_j \; \forall \, i \neq j$. In other words, every vertex is adjacent to every other vertex.
  Example: in the above graph, the vertices b, e, f, g and the edges between them form the complete graph on 4 vertices, denoted K_4.

- A graph is said to be *connected* if for all pairs of vertices (v_i, v_j) there exists a walk that begins at v_i and ends at v_j. (Note: the above graph is connected.)

- A *tree* on n vertices is a connected graph that contains no cycles. Below is an example of a tree with 8 vertices.



- The *degree* of a vertex is defined as the number of vertices it is adjacent to (i.e. the number edges attached to it).
  Example: in the above tree, the point x has degree 4, denoted $d(x) = 4$.

Proposition: Every finite tree has at least two vertices of degree 1.

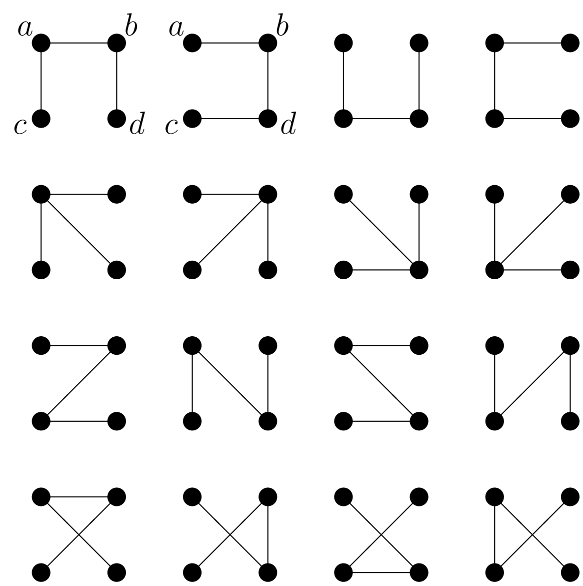
Proof. Notice that any tree must have at least one vertex with degree 1 because if every vertex had degree of at least 2, then one would always be able to continue any walk until a cycle is formed. Now let us assume that we have a tree with exactly one vertex of degree 1. If one removes this vertex of degree 1, the resulting graph must also be a tree since a cycle cannot be added by removing a vertex. In the resulting tree, either the first vertex's neighbor is now of degree 1 or the resulting graph is not a tree and we have a contradiction. If the latter, our proof is done. If the former, we must be able to remove vertices until a contradiction occurs, and there is no vertex of degree 1. We must be able to do this because if we could continue to remove vertices in this way, our graph must be something equivalent to a straight path, and a straight path has two vertices of degree 1, namely the endpoints.

We have shown that a tree must have at least one vertex of degree 1 and that it cannot have exactly one, so it must have at least two. □

3 Cayley's Formula

Using the above definitions, we can now begin to discuss Cayley's formula and its proofs. Cayley's Formula tells us how many different trees we can construct on n vertices. We can think about this process as beginning with n vertices and then placing edges to make a tree. Another way to think about it involves beginning with the complete graph on n vertices, K_n, and then removing edges in order to make a tree. Cayley's formula tells us how many different ways we can do this. These are called *spanning trees* on n vertices, and we will denote the set of these spanning trees by T_n.

The following is a diagram of all of elements of T_4:



Notice that the figures in each row are just rotations of the first one. Each of these graphs is distinct because each has a different set of adjacencies. For example, $a \sim c$ in the first graph above, but $a \nsim c$ in the second graph. Again, these graphs can be obtained by adding edges to 4 vertices or from taking edges away from K_4.

In its simplest form, Cayley's Formula says:

$$|T_n| = n^{n-2} \tag{1}$$

From our above example, we can see that $|T_4| = 16 = 4^2$. It is trivial that there is only one tree on 2 vertices (so $|T_2| = 1 = 2^0$). Also, the only possible tree type on 3 vertices is a 'V' and the 2 other trees are just rotations of that (so $|T_3| = 3 = 3^1$). We can see that Cayley's Formula holds for small n, but how can we prove that it is true for all n? We shall see how we can do this in different ways in the following sections.

4 Prüfer Encoding

The most straight forward method of showing that a set has a certain number of elements is to find a bijection between that set and some other set with a known number of elements. In this case, we are going to find a bijection between the set of Prüfer sequences and the set of spanning trees.

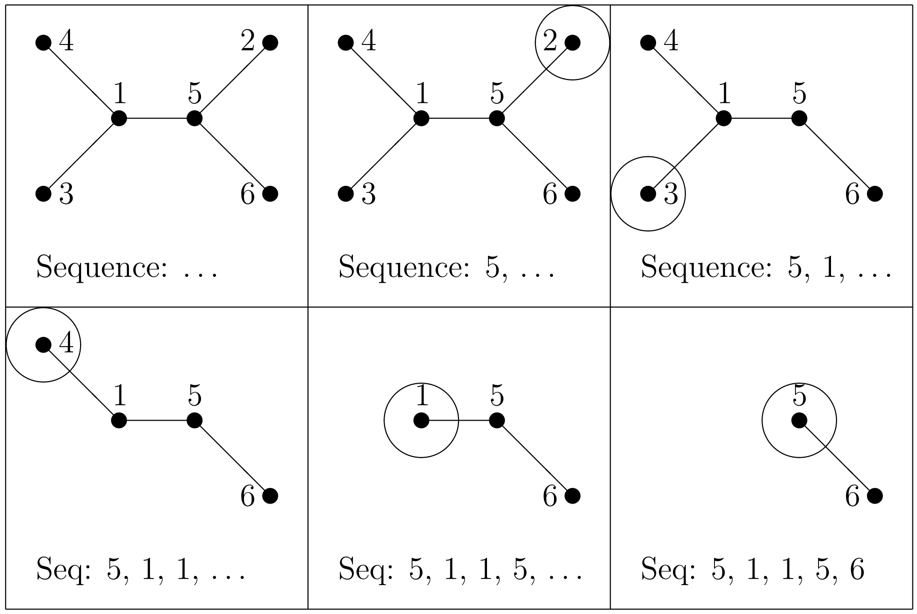
A Prüfer sequence is a sequence of $n-2$ numbers, each being one of the numbers 1 through n. We should initially note that indeed there are n^{n-2} Prüfer sequences for any given n. The following is an algorithm that can be used to encode any tree into a Prüfer sequence:

1. Take any tree, $T \in T_n$, whose vertices are labeled from 1 to n in any manner.

2. Take the vertex with the smallest label whose degree is equal to 1, delete it from the tree and write down the value of its only neighbor. (Note: above we showed that any tree must have at least two vertices of degree 1.)

3. Repeat this process with the new, smaller tree. Continue until only one vertex remains.

This algorithm will give us a sequence of $n-1$ terms, but we know that the last term will always be the number n because even if initially $d(n) = 1$, there will always be another vertex of degree 1 with a smaller label. Since we already know the number of vertices on our graph by the length of our sequence, we can drop the last term as it is redundant. So now we have a sequence of $n-2$ elements encoded from our tree. Below is an example of encoding a tree on 6 vertices:



After encoding our tree, we end up with the sequence: 5, 1, 1, 5, 6; then we can drop the ending 6 and end with our Prüfer Sequence and denote it by P. $P = 5, 1, 1, 5$. So what should make us think that this is the only tree that gives us this sequence? First, we must notice that all of the vertices of degree 1 do not occur in P. With a little thought we can see that this is true for any tree, as the vertices of degree 1 will never be written down as the neighbors of other degree 1 vertices (except when vertex n is of degree 1, but this will never end up in our sequence). In fact, it follows from this that every vertex has degree equal to $1 + a$, where a is the number of times that vertex appears in our sequence.

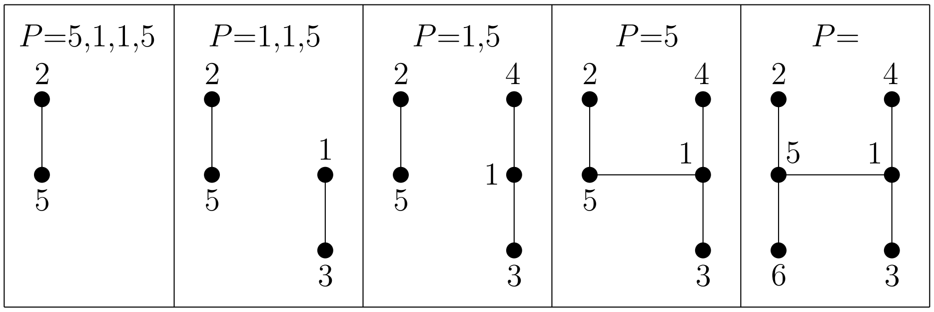
This way of analyzing a Prüfer Sequence provides us with a way of reconstructing an encoded tree. The algorithm goes as follows:

1. Find the smallest number from 1 to n that is not in the sequence P and attach the vertex with that number to the vertex with the first number in P. (We know that $n = 2 +$ number of elements in P.)

2. Remove the first number of P from the sequence. Repeat this process considering only the numbers whose vertices have not yet attained their correct degree.

3. Do this until there are no numbers left in P. Remember to attach the last number in P to vertex n.

Let's reconstruct our original tree from our sequence, $P = 5, 1, 1, 5$:



Following the above steps, we have now reconstructed our original tree on 6 vertices. It may be oriented differently, but all of the vertices are adjacent to their correct neighbors, and so we have the correct tree back. Since there were no ambiguities on how to encode the tree or decode the sequence, we can see that for every tree there is exactly one corresponding Prüfer Sequence, and for each Prüfer Sequence there is exactly one corresponding tree. More formally, the encoding function can be thought of as taking a member of the set of spanning trees on n vertices, T_n, to the set of Prüfer Sequences with $n-2$ terms, P_n. Decoding would then be the inverse of the encoding function, and we have seen that composing these two functions results in the identity map. If we let f be the encoding function, then the above statements can be summarized as follows:

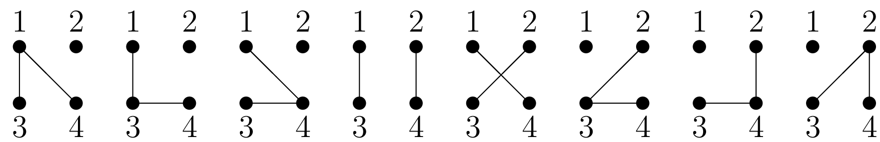
$$f : T_n \longrightarrow P_n, \quad f^{-1} : P_n \longrightarrow T_n, \quad \text{and} \quad f^{-1} \circ f = Id.$$

Since we have found a bijective function between T_n and P_n, we know that they must have the same number of elements. We know that $|P_n| = n^{n-2}$, and so $|T_n| = n^{n-2}$.

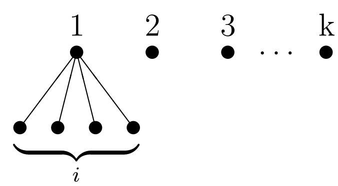
5 A Forest of Trees

Another common way of proving something in mathematics is to prove something more general of which what you want to prove is a specific case. We can use this method to prove Cayley's formula as well. First, we must define what a forest is. A *forest* on n vertices is a graph that contains no cycles, but does not need to be connected like a tree. In fact, a forest can be thought of as a group of smaller trees, hence the name forest.

Now, we are going to define a specific family of forests that we will denote $T_{n,k}$. Let A be an arbitrary set of k vertices chosen from the vertices $1, 2, ..., n$. We are going to define $T_{n,k}$ to be the set of all forests on n vertices that have k trees such that each element of A is in a different tree. The following is the set $T_{4,2}$ where $A = \{1, 2\}$.



There are a couple things that we should note before we move on. First, we should notice that a point by itself is a tree on 1 vertex. Second, we should notice that if we are counting the number of elements in $T_{n,k}$, it does not matter what elements are in A; only the number of elements in A matters, which we have defined as k. Let's take a look at $T_{n,k}$, specifically vertex 1:



In this picture, we are just looking at vertex 1; each vertex, 1 through k, is part of its own tree. Vertex 1 can be adjacent to i of the remaining $n - k$ vertices, and i can range from 0 to $n - k$. If we delete vertex 1 from our graph, we get $(k-1)+(i)$ vertices that must be in separate trees. (Note that none of the i vertices can be adjacent to each other because that would form a cycle.) If we now sum up the number of trees we get from deleting vertex 1 across all possible values of i, we can obtain a value for $T_{n,k}$. Formally we have:

$$T_{n,k} = \sum_{i=0}^{n-k} \binom{n-k}{i} T_{n-1,(k-1)+i} \tag{2}$$

We obtain the binomial coefficient because the i vertices can be chosen in that many ways from the $n - k$ vertices. In certain cases where the above summation may not be defined, we define those values. We set $T_{0,0} = 1$ and $T_{n,0} = 0$ for $n > 0$. Note that $T_{0,0} = 1$ is necessary so that $T_{n,n} = 1$.

Proposition:

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

Proof. We want to prove our proposition using equation (2) and induction:

$$\begin{aligned}
T_{n,k} &= \sum_{i=0}^{n-k} \binom{n-k}{i} T_{n-1,(k-1)+i} && \textit{Equation (2)} \\
&= \sum_{i=0}^{n-k} \binom{n-k}{i} (k-1+i)(n-1)^{(n-1)-(k-1+i)-1} && \textit{Proposition}
\end{aligned}$$

Switch the order of the summation letting $i = (n-k) - i$

$$\begin{aligned}
T_{n,k} &= \sum_{i=0}^{n-k} \binom{n-k}{i} (n-1-i)(n-1)^{i-1} \\
&= \sum_{i=0}^{n-k} \binom{n-k}{i} (n-1)^{i} - \sum_{i=1}^{n-k} \binom{n-k}{i} i(n-1)^{i-1} \\
&= \sum_{i=0}^{n-k} \binom{n-k}{i} (n-1)^{i}(1)^{n-k-i} - \sum_{i=1}^{n-k} \frac{(n-k)! \cdot i}{(i)!(n-k-i)!} \cdot (n-1)^{i-1} \\
&= \sum_{i=0}^{n-k} \binom{n-k}{i} (n-1)^{i}(1)^{n-k-i} - (n-k) \sum_{i=1}^{n-k} \binom{n-k-1}{i-1} (n-1)^{i-1} \\
&= \sum_{i=0}^{n-k} \binom{n-k}{i} (n-1)^{i}(1)^{n-k-i} - (n-k) \sum_{i=0}^{n-k-1} \binom{n-k-1}{i} (n-1)^{i}(1)^{n-k-1-i} \\
&= (n-1+1)^{n-k} - (n-k)(n-1+1)^{n-k-1} && \textit{Binomial Theorem} \\
T_{n,k} &= n^{n-k} - n^{n-k} + kn^{n-k-1} = kn^{n-k-1} && \square
\end{aligned}$$

Now that we have proven the more general case, we can look at how it proves Cayley's Formula. Cayley's Formula describes cases where there is exactly 1 tree on n vertices, so $T_n = T_{n,1}$. Plugging n and 1 into the formula we proved gives us $T_n = T_{n,1} = n^{n-2}$ proving Cayley's Formula.

References:

[1] Babai, László. Lectures and *Discrete Mathematics*.

[2] Aigner, Martin and Günter Ziegler. *Proofs from THE BOOK*. ©2001.