Powers of Cycles and Paths: The Generating Functions for Enumerating Their Spanning Trees

Pablo Blanco Doron Zeilberger

Abstract

Using the theoretical basis developed by Yao and Zeilberger in [5] and some of their accompanying packages, we consider certain graph families whose structure results in rational generating function for sequences related to spanning tree enumeration. Said families are Powers of Cycles, Powers of Paths, and (later) Torus graphs and Grid graphs. As in the Yao and Zeilberger paper [5], the sequences we will consider are C-finite in nature and hence their (rational) generating function can be computed by finding a large number of terms in the sequence. For each graph family, we consider: the sequence of the number of spanning trees and the sequence of the total number of leaves across all spanning trees.

1 Introduction and Background

A subgraph T of a graph G such that T is a tree with $V(T) = V(G)$ is called a *spanning tree* of G. If G is connected, simple, and not a tree, then it will contain several spanning trees. When we count the number of spanning trees of a graph in this paper, we will consider two isomorphic trees with different vertex labels to be different trees; for example, the complete graph on three vertices K_3 has three spanning trees, not one. For a connected graph G, we denote its number of spanning trees of G by $\tau(G)$. Cayley's formula counts the number of spanning of a complete graph on n vertices to be $n^{n-2} = \tau(K_n)$.

The *generating function* $f(x)$ for a sequence $(a_0, a_1, \ldots)$ is the formal power series given by taking the sequence terms as its coefficients:

$$f(x) := \sum_{n=0}^{\infty} a_n x^n.$$

Whenever a sequence has a recurrence of finite length, with constant coefficients, it is called *C-finite*. In other words, $(a_0, a_1, \ldots)$ is C-finite (of order r) if there is a fixed r and coefficients $c_0, \ldots, c_{r-1}$ with $c_0 \neq 0$ such that

$$a_{n+r} = c_{r-1}a_{n+r-1} + \ldots + c_0 a_n$$

for all $n \geq 0$.

The following will be a useful property of C-finite sequences:

Theorem 1.0.1. [2][Kauers-Paule, Thm. 4.3] *A sequence $(a_0, a_1, \ldots)$ is C-finite (of order r) with recurrence*

$$a_{n+r} = c_{r-1}a_{n+r-1} + \ldots + c_0 a_n$$

if and only if

$$\sum a_n x^n = \frac{p(x)}{1 - c_{r-1}x + \ldots - c_1 x^{r-1} - c_0 x^r}$$

for some polynomial $p(x)$ of degree at most $r-1$.

In fact, the polynomial $p(x)$ in the previous Theorem is determined by the initial values $(a_0, \ldots, a_{r-1})$ of the sequence. Later, we will state the value a_0 of our sequence to disambiguate $p(x)$ in the sequence's rational generating function.

For the graph families we will be considering, we can presume that there is a finite Transfer Matrix which describes the corresponding sequence (see [5] for more details). Thanks to this, we can find the sequence's rational generating function by computing a large number of terms in the sequence. The guessing will be done in a Maple procedure, described in the same paper [5], called `GuessRec`.

To generate the terms of whichever sequence we consider, we will apply Kirchhoff's Matrix Tree Theorem, which allows us to compute the number of spanning trees of a(n n-vertex graph) graph by looking at its Laplacian matrix, taking an $n-1$ by $n-1$ (matrix) minor, and computing its determinant. Below, we define the Laplacian matrix of a graph for the reader's convenience.

If G is a (simple) graph with vertices $v_1, \ldots, v_n$, its *Laplacian matrix* is a symmetric matrix with its entries defined by

$$a_{i,j} := \begin{cases} -1 & \text{if } v_i \sim v_j \text{ and } i \neq j \\ \deg(v_i) & \text{if } i = j \end{cases}.$$

Afterwards, we will describe similar methods that we used to counting total number of leaves (across all spanning trees) for members in the same families. In addition, we consider an asymptotic constant relating total leaves to total spanning trees in any families (for which the constant is well-defined). Finally, we include experimental verification of our results' correctness and a description of the accompanying Maple package that we used. Additional outputs and code will be available at [1].

2 Preliminaries and Graph Families

For ease of notation, we assume all graphs are simple.

Definition. For a graph G, the *distance* $|u, v|$ between two vertices u, v is the length of the shortest path between them. If u and v are in different components of G, we say $|u, v| = \infty$ and say the distance between them is infinite.

Definition. Let G be a graph. For an integer $k \geq 1$, the k-th *power* of G, denoted G^k, is the graph obtained from G such that $V(G^k) := V(G)$ and $E(G^k) := \{uv : u, v \in V(G),\ 1 \leq |u, v| \leq k\}$. See Figure 1.

From the definition, we see that if d is the diameter of a connected graph G, that is $d := \max_{u,v \in V(G)} |u, v|$. Then, G^d is the complete graph on $|V(G)|$ vertices. Later in the paper, we will only consider G^k where $k < d$ for this reason.

Notation. We denote the path graph on n vertices by P_n and the cycle graph on n vertices by C_n.

In subsequent sections, we state the generating functions for the number of spanning trees in the graph families

$$\mathcal{G}_r := \{{C_n}^r \ : \ r < \text{diam } C_n\}$$
$$\mathcal{H}_r := \{{P_n}^r \ : \ r < \text{diam } P_n\}.$$

The condition in each construction ensures that our graph families do not contain any "unnatural" complete graphs. The number of spanning trees in a complete graph on n vertices is given by Cayley's formula to be n^{n-2}. Since the diameter of a cycle (resp. path) increases with the number of vertices, the condition $r < \text{diam } C_n = \lfloor n/2 \rfloor$ is implicitly a lower bound on the number of vertices. Explicitly,

$$\mathcal{G}_r = \{{C_n}^r \ : \ n \geq 2r+1\}$$
$$\mathcal{H}_r = \{{P_n}^r \ : \ n \geq r+2\}.$$

As discussed in the introduction, the initial values of a C-finite sequence determines the numerator in its rational generating function. Hence, for the generating functions below in the rest of the paper, we will begin every sequence at $n = 2r+1$ for $\mathcal{G}_r$ and at $n = r+2$ for $\mathcal{H}_r$.

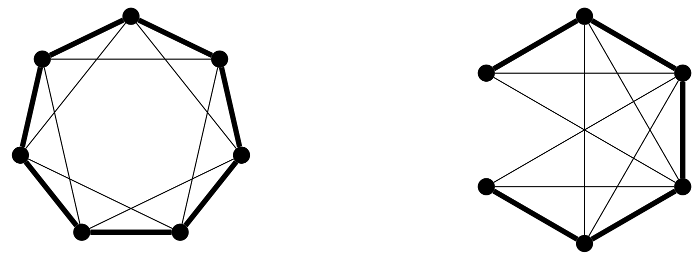

Figure 1: On the left, ${C_7}^2$. On the right, ${P_6}^3$. The thicker edges represent the edges from the corresponding original graph.

Since paths and cycles are ubiquitous objects in graph theory, ou. Let G be a connected graph. Suppose that we wish to estimate the complexity (number of spanning trees) of its k-th power graph, $\tau(G^k)$. One way to estimate $\tau(G^k)$ is to fix a vertex $v \in V(G)$, let $H_1, \ldots, H_N$ be the components of $G - v$ and $m_i(v) := \max_{u \in H_i} |u, v|$. Then, deduce the bound:

$$\tau(G^k) \geq \prod_{i=1}^{N} \tau({P_{m_i(v)}}^k)$$

for all v. In particular,

$$\tau(G^k) \geq \max_{v \in V(G)} \prod_{i=1}^{N} \tau({P_{m_i(v)}}^k).$$

There are a few questions that arise from this type of estimate: Given a graph G, what choice of v maximizes the bound? Which graphs have a "good" choice of v? [To Z: Could we try answering this later?]

3 Counting Total Number of Leaves Across All Spanning Trees

Let G be a graph and T be a spanning tree of G. A *leaf* of a tree T is a vertex with degree exactly 1 in the tree. Denote by $\mathcal{L}(T)$, the set of leaves of T. For ease of notation, we will write $\mathcal{T}(G)$ for the set of (labeled) spanning trees of G and will shorten this as $\mathcal{T}$ whenever G is clear from context. Cite [3] (where?).

The next proposition plays a key role in our implementation for computing a parameter of a graph we later call the B-Z constant. The idea is to count the total number of leaves (of all spanning trees) in a graph by removing a vertex and finding a spanning tree of the resulting graph.

Proposition 3.0.1. *If G is a labeled connected simple graph and for $v \in V(G)$, then*

$$\sum_{T \in \mathcal{T}} |\mathcal{L}(T)| = \sum_{v \in V(G)} \deg_G(v) \cdot |\mathcal{T}_v|$$

where $\mathcal{T} := \mathcal{T}(G)$ and $\mathcal{T}_v := \mathcal{T}(G - v)$.

Proof. Fix $v \in V(G)$. Write $E_v := \{e \in E(G) \ : \ v \in e\}$. There is a bijection between $E_v \times \mathcal{T}_v$ and the spanning trees of T which contain v as a leaf. Hence, we use indicators to obtain

$$\begin{aligned}\sum_{v \in V(G)} \deg_G(v) \cdot |\mathcal{T}_v| &= \sum_{v \in V(G)} |\{T \in \mathcal{T} \ : v \in \mathcal{L}(T)| \\ &= \sum_{v \in V(G)} \sum_{T \in \mathcal{T}} \mathbf{1}_{v \in \mathcal{L}(T)} = \sum_{T \in \mathcal{T}} |\mathcal{L}(T)|.\end{aligned}$$

□

We say G is *vertex-transitive* if for any $u, v \in V(G)$ there is an automorphism φ of G such that $\varphi(u) = v$ and $\varphi(v) = u$.

Corollary 3.0.2. *If G is vertex-transitive, then*

$$\sum_{T\in\mathcal{T}} |\mathcal{L}(T)| = n \cdot \deg_G(v) \cdot |\mathcal{T}_v|$$

for any $v \in V(G)$.

Next, we introduce a parameter for graph families whose member graphs are indexed by number of vertices. For such a graph family, our parameter represents the number of times (on average) that a vertex appears as a leaf in a spanning tree, averaged across all spanning trees, (asymptotically).

For a graph family indexed by number of vertices, $\mathcal{G}$, where G_n represents the n-th graph in the family, we call the following the *B-Z constant for $\mathcal{G}$*:

$$\lim_{n\to\infty} \frac{\sum_{T\in\mathcal{T}(G_n)} |\mathcal{L}(T)|}{n \cdot \tau(G_n)}$$

whenever the limit exists. Since the bound $\frac{\sum|\mathcal{L}(T)|}{n\tau(G)} \leq 1$ holds for all G, we see that the B-Z constant only fails to exist for those graph families with members whose underlying spanning trees are radically different.

Thanks to Corollary 3.0.2, computation of the B-Z constant for vertex-transitive graphs is made easier. Our implementation uses this optimization when appropriate. Incidentally, it's easy to compute the B-Z constant for complete graphs by using Corollary 3.0.2 in conjunction with Cayley's Theorem:

Proposition 3.0.3. *The B-Z constant for the graph family $\{K_n : n \geq 3\}$ is $\frac{1}{e}$.*

Proof. Recall Cayley's formula, which states $\tau(K_n) = n^{n-2}$ for $n \geq 2$. With $\mathcal{T} := \mathcal{T}(K_n)$, Corollary 3.0.2 tells us that $\sum|\mathcal{L}(T)| = n \cdot (n-1)^{n-2}$. Hence,

$$\frac{\sum|\mathcal{L}(T)|}{n \cdot \tau(K_n)} = \left(\frac{n-1}{n}\right)^{n-2} = \left(1 - \frac{1}{n}\right)^{n-2}$$

which approaches e^{-1} as $n \to \infty$. □

4 Powers of a Cycle: Experimental Results

Note that the denominator of RGF for $\mathcal{H}_i$ is always a factor of the denom of RGF for $\mathcal{F}_i$. Likely due to the former being a subgraph of the latter.

4.1 Generating Functions for the Number of Spanning Trees

In this section, we include our results for the Number of Spanning Trees of the class $\mathcal{G}_r$, with $2 \leq r \leq 5$.

Theorem 4.1.1. *The generating function $f(t)$ for the number of spanning trees in $\mathcal{G}_2$ is*

$$\frac{-36t^5 + 132t^4 + 46t^3 - 353t^2 - 116t + 125}{(t+1)^2(t^2-3t+1)^2}$$

Theorem 4.1.2. *The generating function $f(t)$ for the number of spanning trees in $\mathcal{G}_3$ is*

$$\frac{N_3}{(t-1)^2(t^4+3t^3+6t^2+3t+1)^2(t^4-4t^3-t^2-4t+1)^2}$$

where

$$N_3 := -3072t^{17} + 11683t^{16} + 26868t^{15} + 60636t^{14} - 356682t^{13} - 844329t^{12} - 1651344t^{11} - 104646t^{10} + 813834t^9 + 3128248t^8 + 1452330t^7 + 512250t^6 - 1392528t^5 - 1049445t^4 - 579514t^3 - 54068t^2 + 15716t + 16807.$$

Theorem 4.1.3. *The generating function $f(t)$ for the number of spanning trees in $\mathcal{G}_4$ is*

$$\frac{N_4}{D_4}$$

where N_4 is a polynomial of degree 53 and

$$D_4 = (t+1)^2(t^6 - 3t^5 + 6t^4 - 10t^3 + 6t^2 - 3t + 1)^2(t^8 - 4t^7 - 17t^6 + 8t^5 + 49t^4 + 8t^3 - 17t^2 - 4t + 1)^2 \\ (t^{12} + 3t^{11} + 12t^{10} + 28t^9 - 27t^8 + 36t^7 - 81t^6 + 36t^5 - 27t^4 + 28t^3 + 12t^2 + 3t + 1)^2.$$

Theorem 4.1.4. *The generating function $f(t)$ for the number of spanning trees in $\mathcal{G}_5$ is*

$$\frac{N_5}{D_5}$$

where N_5 is a polynomial of degree 161 and

$$D_5 = (t-1)^2(t^8 + 3t^7 + 6t^6 + 10t^5 + 15t^4 + 10t^3 + 6t^2 + 3t + 1)^2(t^8 + 3t^7 + 6t^6 - t^5 + 15t^4 - t^3 + 6t^2 + 3t + 1)^2 \\ (t^{16} - 5t^{15} + 10t^{14} - 10t^{13} - 28t^{12} + 10t^{11} + 110t^{10} + 110t^9 + 88t^8 + 110t^7 + 110t^6 + 10t^5 - 28t^4 - 10t^3 + 10t^2 \\ - 5t + 1)^2(t^{16} - 5t^{15} - 23t^{14} - 10t^{13} - 94t^{12} - 485t^{11} + 242t^{10} + 110t^9 + 649t^8 + 110t^7 + 242t^6 - 485t^5 - 94t^4 \\ - 10t^3 - 23t^2 - 5t + 1)^2(t^{32} + t^{31} + 12t^{30} + 45t^{29} + 45t^{28} - 1561t^{27} + 3917t^{26} - 3222t^{25} - 3981t^{24} + 7745t^{23} \\ + 26379t^{22} - 88937t^{21} + 84093t^{20} + 63864t^{19} - 153881t^{18} - 202281t^{17} + 550163t^{16} - 202281t^{15} - 153881t^{14} \\ + 63864t^{13} + 84093t^{12} - 88937t^{11} + 26379t^{10} + 7745t^9 - 3981t^8 - 3222t^7 + 3917t^6 - 1561t^5 + 45t^4 + 45t^3 \\ + 12t^2 + t + 1)^2$$

4.2 Generating functions for the Total Number of Leaves

Theorem 4.2.1. *The generating function for the total number of leaves (across all spanning trees of a member) in $\mathcal{G}_2$ is*

$$\frac{-8(10t^7 - 67t^6 + 109t^5 + 99t^4 - 282t^3 - 30t^2 + 145t - 40)}{(t+1)^2(t^2 - 3t + 1)^3}.$$

Theorem 4.2.2. *The generating function for the total number of leaves (across all spanning trees of a member) in $\mathcal{G}_3$ is*

$$\frac{A_3}{(t-1)^3(t^4 + 3t^3 + 6t^2 + 3t + 1)^3(t^4 - 4t^3 - t^2 - 4t + 1)^3}$$

where

$$A_3 := -8820t^{26} + 51390t^{25} + 61812t^{24} + 2088t^{23} - 2539950t^{22} - 2981160t^{21} + 2492784t^{20} + 45845688t^{19} \\ + 83018808t^{18} + 107694630t^{17} - 44892840t^{16} - 166389300t^{15} - 333210654t^{14} - 121438506t^{13} + 42702660t^{12} \\ + 312824052t^{11} + 213402930t^{10} + 100784592t^9 - 77616756t^8 - 90041700t^7 - 62209728t^6 - 13836186t^5 \\ + 276924t^4 + 2761596t^3 + 501534t^2 + 32592t - 54432.$$

As the reader might have noticed, the numerators for these generating functions become too cumbersome to write (and more quickly than in the previous section). Henceforth, we omit the numerator and will eventually refer the reader to this website [1] for detailed results when $4 \leq r \leq 5$.

Theorem 4.2.3. *The generating function for the total number of leaves (across all spanning trees of a member) in $\mathcal{G}_4$ is*

$$\frac{A_4}{B_4}$$

where A_4 *is a polynomial of degree* 80 *and*

$$B_4 = (t+1)^3(t^6-3t^5+6t^4-10t^3+6t^2-3t+1)^3(t^8-4t^7-17t^6+8t^5+49t^4+8t^3-17t^2-4t+1)^3 \\ (t^{12}+3t^{11}+12t^{10}+28t^9-27t^8+36t^7-81t^6+36t^5-27t^4+28t^3+12t^2+3t+1)^3$$

4.3 B-Z Constants

5 Powers of a Path: Experimental Results

5.1 Generating functions for the Number of Spanning Trees

In this section, we include our results for the Number of Spanning Trees of the class $\mathcal{H}_r$, with $2 \le r \le 6$.

Theorem 5.1.1. *The generating function* $f(t)$ *for the number of spanning trees in* $\mathcal{H}_2$ *is*

$$\frac{-3t+8}{t^2-3t+1}.$$

Theorem 5.1.2. *The generating function* $f(t)$ *for the number of spanning trees in* $\mathcal{H}_3$ *is*

$$\frac{-16t^4+77t^3-33t^2+39t-75}{(t-1)(t^4-4t^3-t^2-4t+1)}.$$

Theorem 5.1.3. *The generating function* $f(t)$ *for the number of spanning trees in* $\mathcal{H}_4$ *is*

$$\frac{M_4}{(t^6-3t^5+6t^4-10t^3+6t^2-3t+1)(t^8-4t^7-17t^6+8t^5+49t^4+8t^3-17t^2-4t+1)}.$$

where

$$M_4 = -125t^{13}+859t^{12}-13t^{11}-3141t^{10}+3475t^9-5968t^8-11312t^7 \\ +36080t^6-5597t^5-7893t^4+2435t^3-2741t^2-413t+864$$

We know the generating functions for larger r values, but there are many terms with longer coefficients. So, we do not state them in the paper. They are stated in this website [1].

Theorem 5.1.4. *The generating function* $f(t)$ *for the number of spanning trees in* $\mathcal{H}_5$ *is*

$$\frac{M_5}{E_5}.$$

where M_5 *is a degree* 40 *polynomial in* t *and*

$$E_5 = (t-1)(t^8+3t^7+6t^6-t^5+15t^4-t^3+6t^2+3t+1)(t^{16}-5t^{15}+10t^{14}-10t^{13}-28t^{12} \\ +10t^{11}+110t^{10}+110t^9+88t^8+110t^7+110t^6+10t^5-28t^4-10t^3+10t^2-5t+1)(t^{16} \\ -5t^{15}-23t^{14}-10t^{13}-94t^{12}-485t^{11}+242t^{10}+110t^9+649t^8+110t^7+242t^6-485t^5 \\ -94t^4-10t^3-23t^2-5t+1)$$

Seq. of span. trees for $\mathcal{H}_3, \mathcal{H}_4$ not found in OEIS.

5.2 Generating functions for the Total Number of Leaves

Theorem 5.2.1. *The generating function for the total number of leaves (across all spanning trees of a member) in $\mathcal{H}_2$ is*

$$\frac{-2(2t^3 - 15t^2 + 27t - 9)}{(t^2 - 3t + 1)^2}$$

Note: the numerator and denominator in the following have the same degree.

Theorem 5.2.2. *The generating function for the total number of leaves in $\mathcal{H}_3$ is*

$$\frac{2(16t^{10} - 154t^9 + 403t^8 - 340t^7 + 963t^6 - 768t^5 + 1109t^4 - 788t^3 + 509t^2 - 470t + 96)}{(t-1)^2(t^4 - 4t^3 - t^2 - 4t + 1)^2}$$

Theorem 5.2.3. *The generating function for the total number of leaves in $\mathcal{H}_4$ is*

$$\frac{C_4}{(t^6 - 3t^5 + 6t^4 - 10t^3 + 6t^2 - 3t + 1)^2(t^8 - 4t^7 - 17t^6 + 8t^5 + 49t^4 + 8t^3 - 17t^2 - 4t + 1)^2}$$

where C_4 is a degree 29 polynomial. Note that the degree of the numerator is 28.

5.3 B-Z Constants

6 Verification of Results by Random Sampling

In our implementation, we wrote a procedure to compute the total number of leaves of a graph (`NumLeaves` or `VtxTransNumLeaves`) and another to compute the number of spanning trees (`NumSpanTree`). The ratio between the outputs of these procedures is the average number of leaves of a graph.

To verify the accuracy of these procedures, we selected a large member from each of our graph families (at least 100 vertices) and then sampled a large number of spanning trees from that graph (at least 100 and usually more than 200). After doing so, we computed the average number of leaves of the graph in the sample and compared it to our procedures' exact computation. In all cases, we found that the estimate and the true values were similar. For more detailed results on this verification, we direct the reader to [1].

The algorithm we used to sample uniformly random spanning trees is Wilson's Algorithm, as described in [4].

7 Accompanying Maple package

Our Maple package broadly depends on the `LinearAlgebra` library. The `RandomTree` procedure also makes use of the `GraphTheory` library. Below we list some key procedures along with their descriptions:

`NumSpanTree(n,E)` given a positive integer `n` and a set of edges `E` on the set $\{1, \ldots, n\}$, the procedure returns the number of spanning trees of the corresponding graph n, E.

`NumSpanTreeSeq(F, arguments, a,b)`

`NumLeaves`

`NumLeavesSeq`

`VtxTransNumLeavesSeq`

`Hnr`

`Gnr`

8 Questions

References

[1] P. Blanco and D. Zeilberger. Automatic Generation of Generating Functions for Enumerating Spanning Trees. `https://sites.math.rutgers.edu/~zeilberg/mamarim/mamarimhtml/spt.html`.

[2] M. Kauers and P. Paule. *The Concrete Tetrahedron: Symbolic Sums, Recurrence Equations, Generating Functions, Asymptotic Estimates*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[3] J. W. Moon. *Counting Labelled Trees*. Canadian mathematical monographs ; no. 1. Canadian Mathematical Congress, Montreal, 1970.

[4] D. B. Wilson. Generating Random Spanning Trees More Quickly than the Cover Time. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, page 296–303. Association for Computing Machinery, 1996.

[5] Y. Yao and D. Zeilberger. *Untying the Gordian Knot via Experimental Mathematics*, pages 387–410. Springer International Publishing, Cham, 2020.