Automated Counting of Spanning Trees for Several Infinite families of Graphs

Pablo Blanco Doron Zeilberger

Abstract

Using the theoretical basis developed by Yao and Zeilberger, we consider certain graph families whose structure results in a rational generating function for sequences related to spanning tree enumeration. Said families are Powers of Cycles and Powers of Path; later, we briefly discuss Torus graphs and Grid graphs. In each case we know, a priori, that the set of spanning trees of the family of graphs can be described in terms of a finite-state-machine, and hence there is a finite transfer-matrix that guarantees the generating function is rational. Finding this "grammar", and hence the transfer-matrix is very tedious, so a much more efficient approach is to use experimental mathematics. Since computing numerical determinants is so fast, one can use the matrix tree theorem to generate sufficiently many terms, then fit the data to a rational function. The whole procedure can be done rigorously *a posteriori*.

We also construct generating functions for the quantity "total number of leaves" over all spanning trees, and automatically derive the asymptotics for both number of spanning trees and the sum of the number of leaves, enabling our computer to get the asymptotic for the average number of leaves per vertex in a random spanning tree in each family that always tends to a certain number, which we compute explicitly. This number, that we christen the B-Z constant for the infinite family is always some (often complicated) algebraic number, in contrast to the family of complete graphs where this constant is $1/e$ (a transcendental number).

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 contain several spanning trees. When we count the number of spanning trees of a graph in this paper, we 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 trees 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 obtained by setting 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 is 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 state the starting index of (the underlying graph in) our sequence to disambiguate $p(x)$ in the sequence's rational generating function.

Due to the recursive nature of the graph families we consider, we presume that there is a finite Transfer Matrix which describes the corresponding sequence (see [4] 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 is done in a Maple procedure, described in the same paper [4], called `GuessRec`.

To generate the terms of whichever sequence we consider, we apply Kirchhoff's Matrix Tree Theorem, which allows us to compute the number of spanning trees of a(n n-vertex) 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 describe similar methods that we used to count the total number of leaves (across all spanning trees) for members in the same families (in Section 3). In addition, we consider an asymptotic constant relating total leaves to total spanning trees in any families (for which the constant is well-defined) in Section 4. We state our main experimental results in Section 5 and Section 6, then briefly consider an additional pair of graph families in Section 7. In Section 8 and Section 9, we include experimental verification of our results' correctness and then a description of the accompanying Maple package that we used. Additional outputs and code are available at [1]. Finally, we conclude with some conjectures in Section 10

2 Preliminaries

For ease of notation, we assume all graphs are simple. We write $[n] := \{1, \ldots, n\}$.

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 the distance between them is infinite and write $|u, v| = \infty$.

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 (defined as $d := \max_{u,v \in V(G)} |u, v|$), then G^d is the complete graph on $|V(G)|$ vertices. In some cases, it's natural to exclude complete graphs and hence consider only G^k when $k < d$. Additionally, one may note that if there are two shortest paths (in G) between two vertices u, v of length $m > 1$, then G^m may be better viewed as a multigraph where there are two edges between u, v.

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. For those cases, we can once again recall Cayley's Formula: $\tau(K_n) = n^{n-2}$. Since the diameter of a cycle (resp.

path) increases with the number of vertices, the condition $r < \operatorname{diam} C_n = \lfloor n/2 \rfloor$ 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 determine the numerator of its rational generating function. The minimum values of n above will mark the beginning of that family's sequence.

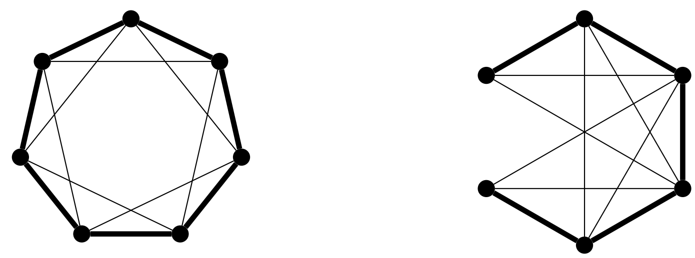

Figure 1: On the left, ${C_7}^2$. On the right, ${P_6}^3$. The thicker edges represent the edges from the corresponding original graph.

Since paths and cycles are ubiquitous in graph theory, they are the focus of this paper. We provide a quick application below.

Let G be a connected graph. Suppose that we wish to estimate the 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 vertex sets $H_1, \ldots, H_N$ 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,

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

3 Counting Total Number of Leaves

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 T. We write $\mathcal{L}(T)$ for the set of leaves of T and $\mathcal{T}(G)$ for the set of (labeled) spanning trees of G. When G is clear from context, we simply write $\mathcal{T}$.

The next proposition plays a key role in our implementation for computing a parameter for graph families that we call the B-Z constant. The idea is to count the total number of leaves (across all spanning trees) in a graph by counting how many times each vertex appears as a leaf. For each vertex in G, remove the vertex from G and count the number of spanning trees in 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

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

□

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).

4 The B-Z Constant: Average Number of Leaves per Vertex

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

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

whenever the limit exists. This is the limit of the average number of leaves in a random spanning tree (and normalized by number of vertices) in the discussed family. For the infinite family of complete graphs K_n, the limit is famously equal to $1/e$ (see below). 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. The rest of the section provides a few examples of B-Z constants and our approach for computing them.

The B-Z constants for both the family of Cycles and the family of Paths is equal to 0, since every tree (a path) has a constant number of leaves (two). A star graph is a connected graph where all edges share the same vertex. When $\mathcal{G}$ is the family of star graphs, its B-Z constant is 1 because the unique tree of a star graph (itself) has $n - 1$ leaves.

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 Formula:

Proposition 4.0.1. *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 $n \to \infty$. □

Earlier, we noted that star graphs have B-Z constant equal to 1. The next proposition observes that we can find a graph family with B-Z constant equal to any rational number (in the interval $[0, 1]$) by slightly modifying star graphs.

Proposition 4.0.2. *For any positive rational number* $\frac{p}{q} < 1$*, there is an indexed graph family* $\mathcal{G}$ *with* $BZ(\mathcal{G}) = p/q$*.*

Proof. First, we introduce a graph operation called subdivision. Let $e = uv$ be an edge of G, with $u, v \in V(G)$. Let $w \notin V(G)$. The result of subdividing e in G is the graph G' where $V(G') := V(G) \cup \{w\}$ and $E(G') := (E(G) \setminus \{e\}) \cup \{uw, wv\}$.

To construct each member $G_k \in \mathcal{G}$ (assuming $k \geq 1$): begin with a star graph S_{pk} with pk leaves; then, subdivide $(q-p)k$ edges in S_{pk} and call this G_k. Note that $|V(G_k)| = qk + 1$ and that G_k has pk leaves. Furthermore, G_k is a tree so it has exactly one spanning tree. It follows that the B-Z constant for $\mathcal{G}$ is $\lim_{k\to\infty} \frac{pk}{qk+1} = \frac{p}{q}$.

□

In upcoming sections, we state the B-Z constants of several graph families. It's possible, as we did in later sections, to compute the B-Z constant of a graph family from its asymptotic behavior in the total number of leaves and number of spanning trees (which can be obtained from their respective generating functions). We used standard residue calculations, implemented in procedure `BZc` from our Maple package (Section 9).

5 Powers of a Cycle: 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 for the family $\mathcal{G}_r$, with $2 \leq r \leq 5$.

Theorem 5.1.1. *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 5.1.2. *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 5.1.3. *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 5.1.4. *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*

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

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* $\mathcal{G}_2$ *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 5.2.2. *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 refer the reader to this website [1] for detailed results when $4 \le r \le 5$.

Theorem 5.2.3. *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*

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

5.3 B-Z Constants

See Section 4 to read about our method for computing these constants. Recall that $BZ(\mathcal{G})$ denotes the B-Z constant of a graph family $\mathcal{G}$.

Theorem 5.3.1. *The B-Z constant for* $\mathcal{G}_2$ *is*

$$-6+\frac{14\sqrt{5}}{5}.$$

Theorem 5.3.2. *The B-Z constant for $\mathcal{G}_3$ is*

$$\frac{3}{7}\left(\frac{45}{2}+9\sqrt{7}-\frac{1}{2}\sqrt{3857+1684\sqrt{7}}\right).$$

Theorem 5.3.3. *Let α be the smallest real root of $z^8-4z^7-17z^6+8z^5+49z^4+8z^3-17z^2-4z+1$. Then, the B-Z constant for $\mathcal{G}_4$ is*

$$\frac{1}{2025}\left(-216\alpha^7-2144\alpha^6+16344\alpha^5+41056\alpha^4-17064\alpha^3-87936\alpha^2-25304\alpha+6944\right).$$

Corollary 5.3.4.

$$BZ(\mathcal{G}_3)<\frac{117451}{355635}<BZ(\mathcal{G}_4)<\frac{117452}{355635}$$

In the previous theorem, $\alpha \approx 0.158778$.

6 Powers of a Path: Experimental Results

6.1 Generating functions for the Number of Spanning Trees

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

Theorem 6.1.1. *The generating function $f(t)$ for the number of spanning trees in $\mathcal{H}_2$ is*

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

Theorem 6.1.2. *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 6.1.3. *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

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

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 6.1.4. *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

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

6.2 Generating functions for the Total Number of Leaves

Theorem 6.2.1. *The generating function for the total number of leaves (across all spanning trees of a member) in $\mathcal{H}_2$ is*

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

Theorem 6.2.2. *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 6.2.3. *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.

6.3 B-Z Constants

See Section 4 to read about our method for computing these constants. Compare the results in this section to Section 5.3.

Theorem 6.3.1. *The B-Z constant for $\mathcal{H}_2$ is*

$$-6 + \frac{14\sqrt{5}}{5}.$$

Theorem 6.3.2. *The B-Z constant for $\mathcal{H}_3$ is*

$$\frac{3}{7}\left(\frac{45}{2} + 9\sqrt{7} - \frac{1}{2}\sqrt{3857 + 1684\sqrt{7}}\right).$$

Theorem 6.3.3. $BZ(\mathcal{H}_4) = BZ(\mathcal{G}_4)$.

We also verified that $BZ(\mathcal{H}_5) = BZ(\mathcal{G}_5)$. See [1].

7 B-Z Constants for Grid and Torus Graphs

In previous sections, we discussed powers of cycles and powers of paths. The members of these two graph families were related by a subgraph relation and we found their B-Z constants to be the same up to $r = 5$. Since a power of a cycle is vertex-transitive, computing the total number of leaves was faster (Corollary 3.0.2). In this section, we discuss another pair of graph families – Grid Graphs and Torus Graphs – which are also related by a subgraph relation. Similarly to the previous pair of families, Torus graphs are vertex-transitive. However, the B-Z constants for grid graphs and torus graphs differ.

Definition. The $a \times b$ grid graph $P_a \times P_b$ has vertex set $V(P_a \times P_b) := \{(i,j) \ : \ i \in [a], j \in [b]\}$ and two (distinct) vertices (u_1, u_2) and (v_1, v_2) are adjacent whenever $|u_1 - v_1| = 1$ or $|u_2 - v_2| = 1$, but not both.

Definition. The $a \times b$ torus graph $C_a \times C_b$ has vertex set $V(C_a \times C_b) := \{(i,j) \ : \ i \in [a], j \in [b]\}$ and two (distinct) vertices (u_1, u_2) and (v_1, v_2) are adjacent whenever $|u_1 - v_1| \in \{1, a-1\}$ or $|u_2 - v_2| \in \{1, b-1\}$, but not both.

Figure 2: On the left, the 3×3 Grid Graph. On the right, the 3×3 Torus Graph.

For the family of torus graphs, we assume $\min(a, b) \geq 3$ to avoid considering multigraphs. A torus graph (resp. grid graph) with dimensions $a \times b$ is isomorphic to another torus graph (resp. grid graph) with dimensions $b \times a$; hence, we may treat the two interchangeably.

Theorem 7.0.1. *The B-Z constant for the family of $2 \times n$ grid graphs is*

$$\sqrt{3} - \frac{3}{2}.$$

In the $3 \times n$ case, the B-Z constants of the two corresponding families differ.

Theorem 7.0.2. *The B-Z constant for the family of $3 \times n$ torus graphs is*

$$\frac{24\sqrt{21} - 104}{21}.$$

Theorem 7.0.3. *The B-Z constant for the family of $3 \times n$ grid graphs is*

$$\frac{-10500 + 2200\sqrt{21} + 1197\sqrt{230 - 50\sqrt{21}} + 149\sqrt{210(23 - 5\sqrt{21})}}{5040}.$$

Similarly, in the case of $4 \times n$ grid and torus graph, their B-Z constants differ again [1]. We conjecture that they differ for all $r \times n$ when $r \geq 3$.

The algebraic numbers of the families above increase in complexity as the fixed dimension of the corresponding family increases. In this website [1], we recorded the outputs of these results − symbolically and numerically. We summarize some of those results here, with decimal approximations (and at least six digits):

The B-Z constant for: $4 \times n$ torus graphs is $0.2917148\ldots$; $5 \times n$ torus graphs is $0.29342497\ldots$; and $6 \times n$ torus graphs is $0.294011\ldots$. The B-Z constant for $4 \times n$ grid graphs is $.2746417\ldots$.

We computed B-Z constants for fewer grid families (than torus families) because grid graphs are not vertex-transitive; hence, we were unable to apply Corollary 3.0.2 to speed up computation of the "total number of leaves".

8 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 [3].

9 Accompanying Maple package

In the following procedures, graphs are represented as an `exprseq` type in Maple. Particularly, a graph is `n,E`, where `n` is a positive integer and `E` is a set of edges on $[n]$. We assume all graphs are connected.

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, ArgList, a,b)` given the name of a graph-generating procedure `F` and a list of arguments `ArgList`, outputs a list of the number of spanning trees for the graphs `F(a, op(ArgList))`,...,`F(b, op(ArgList))`.

`NumLeaves(n,E)` given a graph `n,E`, returns the total number of leaves in such a graph across all its spanning trees.

`NumLeavesSeq(F, ArgList, a,b)` given the name of a graph-generating procedure `F` and a list of arguments `ArgList`, outputs a list of the total number of leaves for the graphs `F(a, op(ArgList))`,...,`F(b, op(ArgList))`.

`VtxTransNumLeavesSeq(F, ArgList, a,b)` similar to `NumLeavesSeq`. However, assumes that the graphs generated by `F` are vertex-transitive. Uses Corollary 3.0.2 to compute the output faster.

`Hnr(n,r)` constructs the r-th power of a path on n vertices. Returns n, E.

`Gnr(n,r)` constructs the r-th power of a cycle on n vertices. Returns n, E.

`BZc(F,Arglist,a,b,k)` Using a graph-generating procedure `F` with fixed arguments `ArgList` and varying the first argument from `a` to `b`, compute the B-Z constant. `k` adjusts the B-Z computation so that if the index of a graph (in the family) is n, then the graph has $k \cdot n$ vertices.

The following procedures generate the outputs for the (main) theorems in this paper, up to `R` ≥ 2, with `K` terms in the sequence, and in variable `x`:

`HnrS(R,x,K)`

`GnrS(R,x,K)`

From Doron Zeilberger's `Cfinite.txt` package:

`RGF(S,t)` given a C-finite sequence `S`, outputs its rational generating function in the variable `t`.

10 Conjectures and Further Discussion

Some observations from our results may indicate deeper structural insights:

1. For each r we computed, the denominator of a rational function from Section 6.1 divides the (corresponding) denominator of a rational function from Section 5.1. Similarly for Section 6.2 and Section 5.2.

2. In the denominators for the generating functions given in Section 5.1, Section 5.2, Section 6.1, Section 6.2, each polynomial factor has coefficients which are symmetric. Is this also true for all other recursive graph families?

While in this paper we mainly provided results for powers of cycles and powers of paths, we also computed the corresponding rational generating functions for Grid Graphs and Torus (Grid) Graphs (see [1]). Those two classes of families, when compared to each other, did not satisfy property 1.; however, they did satisfy property 2 (as listed above). Since a grid graph is a subgraph of a torus graph (of the same dimensions), we know that property 1. is not satisfied from just a subgraph relation. We conclude with a few conjectures.

Conjecture 10.0.1. *The B-Z constants of $\mathcal{G}_r$ and $\mathcal{H}_r$ are equal for all $r \geq 1$.*

Assuming that the previous conjecture holds, a natural question arises: given a graph family, is there a threshold for how many vertices you can "modify" (by edge deletion) while keeping the same B-Z constant? It turns out that Grid graphs and Torus Graphs have different B-Z constants, which may suggest that we can only modify a finite number of vertices. Originally, we considered the following conjecture. Note that the conditions in the following conjecture are satisfied when $\mathcal{G} = \mathcal{G}_r$ and $\mathcal{H} = \mathcal{H}_r$ (as defined in Section 2).

Conjecture 10.0.2. (False) *Let $\mathcal{G} := \{G_n\}_n$ and $\mathcal{H} := \{H_n\}_n$ be (indexed) families of connected graphs, such that $BZ(\mathcal{G})$ and $BZ(\mathcal{H})$ both exist. If the following hold:*

- *$H_n \subseteq G_n$ for all n and*
- *there is some c_1 for which $|E(G_n \setminus H_n)| \leq c_1$ for all n,*

then $\mathcal{G}$ and $\mathcal{H}$ have the same B-Z constant.

For two graphs G, H such that $V(G) = V(H)$, we define $G \setminus H$ as the graph $V(G \setminus H) := V(G)$ and $E(G \setminus H) := E(G) \setminus E(H)$. Unfortunately, the conjecture above fails for families of sparse graphs, as proven by considering the following counterexample.

Counterexample. Let $\mathcal{H}$ be the family of graphs constructed in the proof of Proposition 4.0.2 with B-Z constant equal to $\frac{1}{2}$. In other words, $\mathcal{H}$ is the family of graphs resulting from subdividing each edge in a star graph once. From each $H_n \in \mathcal{H}$, construct G_n by adding an edge between two non-leaf vertices. We see that $\tau(G_n) = 3 \cdot \tau(H_n)$, but $\mathcal{L}(G_n) = \mathcal{L}(H_n)$. Hence, $BZ(\mathcal{G}) = \frac{1}{3} \cdot BZ(\mathcal{H}) = \frac{1}{6} \neq \frac{1}{2} = BZ(\mathcal{H})$.

Observation 10.0.3. *Let $n \geq 2r+1$. If $G_n \in \mathcal{G}_r$ and $H_n \in \mathcal{H}_r$, then $G_n \setminus H_n$ contains a (copy of the) Half Graph on $2r$ vertices and $n - 2r$ isolated vertices.*

The *Half Graph* on $2n$ vertices is a bipartite graph with vertex set $\{u_1, \ldots, u_n, v_1, \ldots, v_n\}$ and edges $\{u_i v_j \ : \ i \leq j\}$. Since r is fixed, the number of differing edges between a Power of a Cycle and a Power of a Path is finite.

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] 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.

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