Enumeration of Spanning Trees in Certain Vertex Corona Product Graph

Fouad Yakoubi

Department of Computer Sciences, Faculty of Sciences
Mohammed V-Agdal University, P.O. Box 1014, Rabat, Morocco

Mohamed EL Marraki

LRIT associated unit the CNRST(URAC29), Faculty of Sciences
Mohammed V-Agdal University, B.P. 1014 RP, Agdal, Rabat, Morocco

Copyright © 2014 Fouad Yakoubi and Mohamed EL Marraki. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

The corona product of two graphs $G_{n_1}$ and $G_{n_2}$ is one of the most important graph operations that allows us to build a complex graph by means of simple graphs. In this paper, firstly we will define the corona product and other necessary notions. Next, we will give the explicit major formula which counts the number of spanning trees in corona product graph of two planar graphs. Then, we will propose the corona product of a planar graph and certain families of outerplanar graphs such as the Fan graph, the Star and the complete binary tree, for which we will calculate the number of spanning trees.

Mathematics Subject Classification: 05C85, 05C30

Keywords: outerplanar graph, complexity, spanning tree, corona product graph, complete binary tree
1 Introduction

Let \( G(\mathcal{E}, \mathcal{V}) \) be a graph with \( \mathcal{V}(G) \) the set of vertices of \( G \) and \( \mathcal{E}(G) \) the set of edges. The graph \( G \) is called connected, if each two vertices of \( \mathcal{V}(G) \) \( v_i \) and \( v_j \) are connected by a path in \( G \). We call \( G \) a cycled graph, if it constitutes a sequence of vertices starting and ending by the same vertex in a closed walk with no repetitions of vertices or edges. However, this work is focused only on planar graphs, which don’t contain edges that intersect. So, the planar graph \( G \) divides the plan into a set of faces \( \mathcal{F}(G) \). If all of the vertices belong to the unbounded face of \( G \), then \( G \) is called an outerplanar graph (See The Figure 1). We denote by \( n, m \) and \( f \) respectively the sizes of \( \mathcal{V}(G), \mathcal{E}(G) \) and \( \mathcal{F}(G) \), i.e. \( (|\mathcal{V}(G)| = n, |\mathcal{F}(G)| = f, |\mathcal{E}(G)| = m) \). Furthermore, the numbers of vertices, edges and faces of the planar graph \( G \) are related by Euler’s equation [2]:

\[
 n + f - m = 2
\]  

Any connected graph without cycle is called a Tree, and a spanning tree of the graph \( G \) is a subgraph of \( G \) which is a tree that contains all vertices of \( G \).

![Figure 1: a planar graph and the corresponding spanning tree.](image)

Spanning trees have many practical applications as well as interesting theoretical properties. For example they are used in the context of meshed networking technology where the hosts are connected step by step with no central hierarchy. So, we need to calculate a subnet in such way that two hosts can always be connected in a unique manner. In order to avoid the presence of loops which is the origin of broadcast storms that paralyze the network, the algorithm used to generate the spanning trees is called STP (Spanning Tree Protocol) and defined in the IEEE 802.1 D standards.

The number of spanning trees in a network is very important in many problems that are associated with networks, such as Potts model [6, 7], dimer coverings [5] and the origin of fractality for fractal scale-free networks [8, 9]. The number of spanning trees of a graph \( G \), also called the complexity of \( G \), denoted by \( \tau(G) \).

The Graph operations have an important role in the study of graph decompositions into isomorphic subgraphs, and the corona product is one of the
best among them. There are two types of corona product, which are corona product vertex and corona product edge. In this work, we will focus only on the corona product vertex. Let $G_{n_1}$ and $G_{n_2}$ be two planar graphs with $|V(G_{n_1})| = n_1$, $|E(G_{n_1})| = m_1$ and $|V(G_{n_2})| = n_2$, $|E(G_{n_2})| = m_2$. The corona product vertex of $G_{n_1}$ and $G_{n_2}$ is the graph $G_n$ denoted by $G_{n_1} \odot G_{n_2}$ obtained by taking one copy of $G_{n_1}$ and $n_1$ copies of $G_{n_2}$ and joining the $i^{th}$ vertex of $G_{n_1}$ to each vertex in the $i^{th}$ copy of $G_{n_2}$ (See Figure 3). Therefore, $n = n_1 + n_1 n_2$ and $m = m_1 + n_1 m_2 + n_1 n_2$. So, it is clear from the definition that the corona product of two graphs isn’t commutative. Moreover, when we say “corona product” of two graphs in that follows, then we mean the corona product vertex.

Remark : In this work, when we want to calculate the corona product of two graphs $G_{n_1}$ and $G_{n_2}$. So, in order to avoid having a corona product graph which isn’t a planar, we consider $G_{n_1}$ is a planar graph and $G_{n_2}$ an outerplanar graph.

2 Preliminary Notes

In this section, we introduce all the theorems that will be used in this work. Let $G_n$ be an undirected graph and its complexity is denoted by $\tau(G_n)$.

The Matrix-Tree Theorem [2]:

Let $L(G_n)$ The Laplacian matrix of $G_n$ with $L = D - A$, where $A$ the adjacency matrix and $D$ the diagonal matrix of $G_n$. When we delete the $i^{th}$ row and $j^{th}$ column of $L(G_n)$, we get the matrix $L^*(G_n)$, Therefore:

$$\tau(G_n) = (-1)^{i+j} \det \left(L^*(G_n)\right)$$

(2)
If $G_n = G_1 \uparrow G_2$ a graph is composed of two planar graphs $G_{n_1}$ and $G_{n_2}$, which intersect in a path of $k$ edges as is shown in Figure 4, then the number of spanning tree in $G_n$ is given by the following equation [3]:

$$\tau(G_n) = \tau(G_{n_1}) \times \tau(G_{n_2}) - k^2 \times \tau(G_{n_1} - e) \times \tau(G_{n_2} - e)$$ \hspace{1cm} (3)

Let $G_n$ be a graph of type $G_{n_1} : G_{n_2}$ as illustrated in Figure 4, where $v_1$ and $v_2$ are two articulation vertices of $G_n$ which is formed by two planar graphs $G_{n_1}$ and $G_{n_2}$, in such way that they cross into $v_1$ and $v_2$, then the complexity of $G_n$ is given as follows [3]:

$$\tau(G_n) = \tau(G_{n_1}) \times \tau(G_{n_2}.v_1v_2) + \tau(G_{n_1}.v_1v_2) \times \tau(G_{n_2})$$ \hspace{1cm} (4)

If $G_n$ a graph is composed of two planar graphs $G_{n_1}$ and $G_{n_2}$, which are intersected in a single articulation vertex $v$ (See the Figure 4) i.e they are connected graphs which intersect exactly in one vertex, we denote $G_n$ by $G_{n_1} \circ G_{n_2}$. Therefore, the number of spanning trees in $G_n$ is given as follows [3]:

$$\tau(G_n) = \tau(G_{n_1}) \times \tau(G_{n_2})$$ \hspace{1cm} (5)

Let $G_n$ be a graph, $e$ an edge in $G_n$ (See The Figure 5), where $v_1$ and $v_2$ are the end vertices of the edge $e$, we denote by $G_n - e$ the graph got by removing $e$, and we denote by $G_n.e$ the graph got by removing $e$ and pasting the vertex $v_1$ with $v_2$. Then, the complexity of $G_n$ is given by [1]:

$$\tau(G_n) = \tau(G_n - e) + \tau(G_n.e).$$ \hspace{1cm} (6)
3 Main Results

In the following section, we will give the explicit formula to calculate the complexity of the corona product of a planar graph and an outerplanar graph. We denote by $G_n$, a planar graph which contains $n$ vertices and $P_1$ a path that has a single vertex (See The Figure 2).

**Theorem 3.1**

Let $G_{n_1}$ a planar graph and $G_{n_2}$ an outerplanar graph. Then, the number of spanning trees in the corona product graph $G_{n_1} \odot G_{n_2}$ is given by:

$$\tau(G_{n_1} \odot G_{n_2}) = \tau(G_{n_1}) \times \left(\tau(G_{n_2} \odot P_1)\right)^{n_1} \quad (7)$$

Proof:

Let $G_n = G_{n_1} \odot G_{n_2}$ the corona product graph of $G_{n_1}$ and $G_{n_2}$. So, $G_n$ is formed by connecting $G_{n_1}$ and each copy $G_i$ of $G_{n_2}$ in an articulation vertex $v_i$ of $G_{n_1}$. Consequently, we will have $n_1$ articulation vertices between $G_{n_1}$ and each copy of $G_{n_2}$. To calculate $\tau(G_{n_1} \odot G_{n_2})$, we use the equation 5. Therefore, we obtain:

$$\tau(G_n) = \tau(G_{n_1}) \times \prod_{i=1}^{n_1} \tau(G_{n_2} \odot P_1), \text{ hence}$$

$$\tau(G_n) = \tau(G_{n_1}) \times \left(\tau(G_{n_2} \odot P_1)\right)^{n_1}$$

According to the previous theorem, to calculate the number of spanning trees in corona product graph $(G_{n_1} \odot G_{n_2})$ of $G_{n_1}$ and $G_{n_2}$, we only need to have the number of spanning trees of $G_{n_2} \odot P_1$.

3.1 Complexity of a planar graph $G_{n_1}$ and Fan graph $F_{n_2}$

In this part, we will give the explicit formula that calculates the complexity of corona product graph of a planar graph $G_{n_1}$ and The Fan graph $F_{n_2}$.

Figure 6: The Corona product of $F_{n_2}$ and $P_1 : F_{n_2} \odot P_1$. 
Lemma 3.2:
Let $F_{n_2}$ be the Fan graph (See Figure 6) and $\xi_{n_2+1}$ the Maximal planar graph [4], the corona product graph of $F_{n_2}$ and $P_1$ ($\xi_{n_2+1} = F_{n_2} \circ P_1$). Then, the complexity of $\xi_{n_2+1}$ is given by the following formula:

$$\tau(\xi_{n_2+1}) = \frac{n_2 + 1}{2\sqrt{3}} \left((2 + \sqrt{3})^{n_2-1} - (2 + \sqrt{3})^{n_2-1}\right), \; n_2 \geq 2$$

(8)

Proof: see [4]

Corollary 3.3:
Let $G_{n_1}$ a planar graph, $F_{n_2}$ the Fan graph and $G_n$ the corona product graph of $G_{n_1}$ and $F_{n_2}$. The number of spanning trees in $G_n$ is given by the following equation:

$$\tau(G_n) = \tau(G_{n_1}) \times \left(\frac{n_2 + 1}{2\sqrt{3}} \left((2 + \sqrt{3})^{n_2-1} - (2 + \sqrt{3})^{n_2-1}\right)\right)^{n_1}, \; n_2 \geq 2$$

(9)

Proof:
Let $G_{n_1}$ be a planar graph and $F_{n_2}$ the fan graph. In order to calculate $\tau(G_{n_1} \circ F_{n_2})$, we simply apply Theorem 3.1 and Lemma 3.2. Then:

$$\tau(G_{n_1} \circ F_{n_2}) = \tau(G_{n_1}) \times (\tau(F_{n_2} \circ P_1))^{n_1} = \tau(G_{n_1}) \times \left(\tau(\xi_{n_2+1})\right)^{n_1}$$

Hence the result.

Let $C_{n_1}$ be a cycle graph which contains $n_1$ vertices, the number of spanning trees in this graph is equal to the length of its cycle ($\tau(C_{n_1}) = n_1$). By applying Corollary 3.3, we obtain the following corollary.

Corollary 3.4:
If $G_n$ is the corona product graph of $C_{n_1}$ and $F_{n_2}$, then the spanning trees number in $G_n$ is given as follows:

$$\tau(G_n) = n_1 \times \left(\frac{n_2 + 1}{2\sqrt{3}} \left((2 + \sqrt{3})^{n_2-1} - (2 + \sqrt{3})^{n_2-1}\right)\right)^{n_1}, \; n_2 \geq 2$$

(10)

3.2 The Corona product complexity of a planar graph $G_{n_1}$ and a tree $T_{n_2}$

In this part, we will treat the corona product of a planar graph $G_{n_1}$ and a tree $T_{n_2}$, focusing on a few cases of trees, such as the path $P_{n_2}$, the star $S_{n_2}$ and the complete binary tree $T_{n_2}$. 
3.2.1 Case of a path $P_{n_2}$

In this case, we will study the Corna product of a planar graph $G_{n_1}$ and a simple path $P_{n_2}$, calculating its complexity $\tau(G_{n_1} \odot P_{n_2})$. In order to count the number of spanning tree in $G_{n_1} \odot P_{n_2}$. Firstly, we have to calculate $\tau(P_{n_2} \odot P_1)$ which is equal to $\tau(F_{n_2+1})$. In addition, we will apply Theorem 3.1, hence

$$\tau(G_{n_1} \odot P_{n_2}) = \tau(G_{n_1}) \times \left( \tau(P_{n_2} \odot P_1) \right)^{n_1} = \tau(G_{n_1}) \times \left( \tau(F_{n_2+1}) \right)^{n_1}$$

**Lemma 3.5**:

Let $G_{n_1}$ a planar graph, $P_{n_2}$ a path and $F_{n_2+1}$ the corona product graph of $P_{n_2}$ and $P_1$. So the number of spanning trees in $P_{n_2} \odot P_1$ (i.e $F_{n_2+1}$) and $G_{n_1} \odot P_{n_2}$ are given by the following equations (with $n_1 \geq 1$ and $n_2 \geq 2$):

$$\tau(F_{n_2+1}) = \frac{1}{\sqrt{5}} \left( \left( \frac{3 + \sqrt{5}}{2} \right)^{n_2} - \left( \frac{3 - \sqrt{5}}{2} \right)^{n_2} \right)$$  \hspace{1cm} (11)

$$\tau(G_{n_1} \odot P_{n_2}) = \tau(G_{n_1}) \times \left( \frac{1}{\sqrt{5}} \left( \left( \frac{3 + \sqrt{5}}{2} \right)^{n_2} - \left( \frac{3 - \sqrt{5}}{2} \right)^{n_2} \right) \right)^{n_1}$$  \hspace{1cm} (12)

**Proof**:

The formula (11) is already demonstrated in [3]. Let $G_{n_1}$ be a planar graph. In order to calculate the complexity in corona product graph of $G_{n_1}$ and $P_{n_2}$ ($\tau(G_{n_1} \odot P_{n_2})$), we apply Theorem 3.1 and (11) the first equation of Lemma 3.5, hence the result.

**Corollary 3.6**:

Let $G_n$ be the corona product graph of $C_{n_1}$ and $P_{n_2}$. The complexity of $G_n$ is given as follows (with $n_1 \geq 2$ and $n_2 \geq 2$):

$$\tau(G_n) = \frac{n_1}{\sqrt{5}} \left( \left( \frac{3 + \sqrt{5}}{2} \right)^{n_2} - \left( \frac{3 - \sqrt{5}}{2} \right)^{n_2} \right)^{n_1}$$  \hspace{1cm} (13)

3.2.2 Case of a star $S_{n_2}$

Let $S_{n_2}$ a star tree which contains $n_2$ vertices (See the Figure 7), the corona product graph of $S_{n_2}$ and $P_1$ ($S_{n_2} \odot P_1$) is denoted by $\xi_{n_2+1}'$.

**Lemma 3.7**:

The number of spanning trees in $S_{n_2} \odot P_1$ and $G_{n_1} \odot S_{n_2}$ are calculated as follows, where $n_1 \geq 1$ and $n_2 \geq 3$:

$$\tau(S_{n_2} \odot P_1) = (n_2 + 1) \times 2^{n_2-2}$$  \hspace{1cm} (14)

$$\tau(G_{n_1} \odot S_{n_2}) = \tau(G_{n_1}) \times \left( (n_2 + 1) \times 2^{n_2-2} \right)^{n_1}$$  \hspace{1cm} (15)
Proof:
Let $S_{n_2}$ be the star tree as illustrated in Figure 7. In order to calculate $\tau(\xi'_{n_2+1})$ or $\tau(S_{n_2} \circ P_1)$, we choose $e$ the last edge of $\xi'_{n_2+1}$ and we apply the equation (6). Then: $\tau(\xi'_{n_2+1}) = \tau(\xi'_{n_2+1} - e) + \tau(\xi'_{n_2+1}.e)$. The graph $\xi'_{n_2+1}.e$ has a single articulation vertex (See the Figure 8). Using the equation (5), it is obvious that $\tau(\xi'_{n_2+1}.e) = 2^{n_2-1}$. In order to calculate $\tau(\xi'_{n_2+1} - e)$, we consider $\tau(\xi'_{n_2+1} - e) = x_{n_2+1}$ and we cut along the path $p$ which contains two edges (See the Figure 8).

Then, by applying the formula (3) we obtain the following recursion formula: $x_{n_2+1} = 4x_{n_2} - 2^2x_{n_2-1}$, where the associated characteristic equation is $r^2 - 4r + 4 = 0$. Therefore, the solution of this equation is: $r = 2$, hence $x_{n_2+1} = (a(n_2+1) + b)2^{n_2+1}$. Using the initial conditions $x_4 = 4$ and $x_5 = 12$, we get $x_{n_2+1} = \left(\frac{1}{3}(n_2 + 1) - \frac{1}{3}\right) 2^{n_2+1}$. Then, $\tau(\xi'_{n_2+1}) = 2^{n_2-1} + \left(\frac{n_2-1}{2}\right) 2^{n_2-1}$. Hence the result.

Let $C_{n_1}$ be a cycle graph with $n_1$ vertices. In order to calculate the complexity of corona product of $C_{n_1}$ and $S_{n_2}$, we apply Lemma 3.7.
Corollary 3.8:
The number of spanning trees in the corona product graph \((G_n)\) of \(C_{n_1}\) and \(S_{n_2}\) is given by:

\[
\tau(G_n) = n_1 \times ((n_2 + 1) \times 2^{n_2-2})^{n_1}; \quad n_1 \geq 2, n_2 \geq 3
\] (16)

3.2.3 Case of a complete binary tree

In this part, we will calculate the number of spanning trees in corona product graph of a planar graph \(G_{n_1}\) and a complete binary tree \(T_{n_2}\). This latter is formed by \(n_2\) vertices, with \(n_2 = 2^{k+1} - 1\) (i.e. \(n_2 + 1 = 2^{k+1}\)) where \(k\) represents the depth of \(T_{n_2}\). We denote by \(Y_{n_2+1}\) the corona product graph of \(T_{n_2}\) and \(P_1\) (see Figure 9). The edge \(e\) divides \(Y_{2^{k+1}}\) into two identical graphs denoted by \(X_{2^{k+1}}\).

![Diagram](image)

Figure 9: The corona product graph of \(T_{n_2}\) and \(P_1\)

Theorem 3.9:
Let \(Y_{n_2+1}\) be the corona product graph of \(T_{n_2}\) and \(P_1\). So the number of spanning trees of \(Y_{2^{k+1}}\) (with \(k \geq 2\)) is given by the following system of recurrence equations:

\[
\begin{align*}
\tau(Y_{2^{k+1}}) &= \left(\tau(X_{2^{k+1}})\right)^2 - \left(\tau(Y_{2^k})\right)^2 \\
\tau(X_{2^{k+1}}) &= \tau(Y_{2^k}) + 2 \times \tau(X_{2^{(k-1)+1}}) \left[\tau(X_{2^{(k-1)+1}}) - \tau(Y_{2^{k-1}})\right] \\
\tau(X_{2^{2+1}}) &= 20 \\
\tau(Y_{2^2}) &= 8
\end{align*}
\] (17)

According the first equation of the above system, In order to count the complexity \(\tau(Y_{2^{k+1}})\), we need to have \(\tau(X_{2^{k+1}})\) and \(\tau(Y_{2^k})\). Using the following Algorithm (See Figure10), we calculate \(\tau(Y_{2^{i+1}})\) for each iteration \(i\) (for \(i = 2, ..., k\)). So, this algorithm has a linear complexity \(\Theta(k)\) where \(n_2 + 1 = 2^{k+1}\), therefore \(\Theta(k) = \Theta(\log_2(n_2))\).

Proof of Theorem 3.9:
Firstly, if we cut the graph \(Y_{2^{k+1}}\) in the middle edge \(e\) (See Figure 11), we’ll get two subgraphs which are identical denoted by \(X_{2^{k+1}}\), both of them contain
5436

> τ := proc (k)
>    local x := 20, y := 8, a, b, i := 2;
>    while (i < k) do
>      a := y; b := x;
>      y := b^2 - a^2;
>      x := y + 2*b*(b-a); i := i + 1; od;
>    return y; end;

Figure 10: Maple implementation of Algorithm for calculating \( \tau(Y_{2k+1}) \)

2^k + 1 edges. When we delete the same edge \( e \) of \( X_{2^k+1} \), we get the subgraph \( X_{2^k+1}-e \), that has the same complexity of \( Y_{2^k} \).

By applying Equation 3, the number of spanning trees in \( Y_{2k+1} \) is given by:

\[
\tau(Y_{2k+1}) = \tau(X_{2^k+1}) \times \tau(X_{2^k+1}) - \tau(X_{2^k+1}) \times \tau(X_{2^k+1}) - e \times \tau(X_{2^k+1}) - e .
\]

Next, in order to calculate the complexity of \( X_{2^k+1} \), we choose again the same edge \( e \) in \( X_{2^k+1} \). So, we get the following formula by applying equation 6:

\[
\tau(X_{2^k+1}) = \tau(Y_{2^k}) + 2 \times \tau(X_{2^k} - 1) = \tau(Y_{2^k}) + \tau(X_{2^k+1}, e)
\]  

Then, we cut \( (X_{2^k+1}, e) \) from the middle as illustrated in Figure 12, with a view to count its complexity \( \tau(X_{2^k+1}, e) \), using Equation 3 we have:

\[
\tau(X_{2^k+1}, e) = \tau(X_{2^k} - 1) \times \tau(U_{2^k} - 1) - \tau(X_{2^k} - 1) \times \tau(Y_{2^k} - 1).\]

Where \( U_{2^k-1+1} \) represents the left part of the graph \( (X_{2^k+1}, e) \) after cutting. Finally, to calculate the complexity of \( U_{2^k-1+1} \), we cut it as shown in Figure 12. If we apply Equation 3, we get:

\[
\tau(U_{2^k-1+1}) = 2 \times \tau(X_{2^k-1+1}) - \tau(Y_{2^k-1})
\]

We replace \( \tau(U_{2^k-1+1}) \) and \( \tau(X_{2^k+1}, e) \) by their expressions in Equation 18, hence the result:

\[
\tau(X_{2^k+1}) = \tau(Y_{2^k}) + 2 \times \tau(X_{2^k} - 1) \tau(Y_{2^k-1+1}) - \tau(Y_{2^k-1})].
\]

The table 1 gives some values of the number of spanning trees in the graph \( Y_{2k+1} \) calculated by the above Algorithm (See Figure 10).

**Corollary 3.10:**

Let \( G_{n_1} \) a planar graph, and \( T_{n_2} \) a complete binary tree. So, the complexity
Figure 12: Graphs $X_{2^{k+1}}e$, $U_{2^{k-1}+1}$, $X_{2^{k-1}+1}$ and $Y_{2^{k-1}}$

<table>
<thead>
<tr>
<th>$k+1$</th>
<th>$	au(Y_{2^{k+1}})$</th>
<th>digits number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>$2^4 \times 3 \times 7$</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>$2^{12} \times 3^3 \times 5$</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>$2^{20} \times 3^4 \times 5^2 \times 17 \times 41$</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>$2^{43} \times 3^8 \times 7 \times 5^5 \times 17^2 \times 29$</td>
<td>26</td>
</tr>
<tr>
<td>7</td>
<td>$2^{54} \times 3^{18} \times 5^5 \times 17^4 \times 29^2 \times 11 \times 239$</td>
<td>51</td>
</tr>
<tr>
<td>8</td>
<td>$2^{173} \times 3^{47} \times 5^{16} \times 11^2 \times 13^2 \times 17^{29} \times 29^4$</td>
<td>103</td>
</tr>
<tr>
<td>9</td>
<td>$2^{340} \times 3^{72} \times 5^{32} \times 11^4 \times 13^4 \times 17^{16} \times 29^8 \times 7 \times 199 \times 577$</td>
<td>205</td>
</tr>
<tr>
<td>10</td>
<td>$2^{683} \times 3^{134} \times 5^{65} \times 11^8 \times 13^8 \times 17^{32} \times 29^{17} \times 19741 \times 577^2$</td>
<td>411</td>
</tr>
<tr>
<td>11</td>
<td>$19 \times 2^{1364} \times 3^{289} \times 5^{130} \times 11^{16} \times 13^{16} \times 17^{64} \times 23 \times 29^{32} \times 197^2 \times 59 \times 577^4 \times 353$</td>
<td>822</td>
</tr>
</tbody>
</table>

Table 1: some values of $\tau(Y_{2^{k+1}})$

of corona product graph of $G_{n_1}$ and $T_{n_2}$ is given as follows (with $n_1 \geq 1$ and $k \geq 2$):

$$\tau(G_{n_1} \circ T_{n_2}) = \tau(G_{n_1}) \times \left(\tau(Y_{2^{k+1}})\right)^{n_1}$$  \hspace{1cm} (19)

To calculate $\tau(G_{n_1} \circ T_{n_2})$, we apply Theorem 3.1 and Theorem 3.9, hence the formula of the above corollary.

## 4 Conclusion

In this paper, we managed to calculate the number of spanning trees in graphs that are complex through the corona product. If we can compose a complex graph with corona product from two small planar graphs, then we can easily calculate its complexity. As a matter of fact, using the Matrix-Tree Theorem (The Kirchhoff Theorem), we can count the complexity of any graph $G$, computing the determinant of the laplacian matrix $L^*(G)$. However, this calculation isn’t often obvious. Moreover, we have been interested in calculating
the number of spanning trees in corona product graph. For this reason, we have given the explicit formula for the general case to calculate the complexity of corona product graph. Then, we have deduced the formulas that give us the number of spanning trees in corona product graph of a planar and certain families of outerplanar graphs, such as the Fan graph, Star and the Complete Binary tree.

References


Received: July 5, 2014