Growth in Catacondensed Benzenoid Graphs†

Primož Lukšič*
IMFM, Jadranska 19, Ljubljana 1000, Slovenia
Tomaž Pisanski
IMFM, Jadranska 19, University of Ljubljana, Ljubljana 1000, Slovenia, and University of Primorska, Koper, Slovenia

Received November 13, 2006

The generating function of the sequence counting the number of graph vertices at a given distance from the root is called the spherical growth function of the rooted graph. The vertices farthest from the root form an induced subgraph called the distance-residual graph. These mathematical notions are applied to benzenoid graphs which are used in graph theory to represent benzenoid hydrocarbons. An algorithm for calculating the growth in catacondensed benzenoids is presented, followed by some examples.

INTRODUCTION

A *benzenoid graph* (sometimes called a *benzenoid system*) is a graph-theoretical representation of the carbon skeleton of a benzenoid hydrocarbon.[1] It is represented as a finite, connected, planar graph, embedded into a hexagonal lattice with all hexagons being mutually congruent and is always drawn in such a way that some of its edges are vertical. If all vertices of a benzenoid graph G lie on its perimeter (boundary), then G is said to be *catacondensed*, otherwise it is *pericondensed*.[2] In each case we consider only simple graphs which represent benzenoid molecules without holes, therefore excluding *coronoids*.[3]

Catacondensed benzenoid graphs have two types of vertices: the ones of degree 2 which belong to only one hexagon, and the ones of degree 3 which lie on exactly two adjacent hexagons. Since all vertices of catacondensed benzenoid graphs lie on their perimeters, these graphs belong to a set of *outerplanar graphs*[4] which have an embedding in a plane such that all vertices lie on a circle and all edges lie in the interior without intersecting.

We recall the definition of *growth* in rooted graphs.[5] Let G be a connected, finite or locally finite graph (infinite but with finite vertex degrees). By selecting a vertex r from G we can define the (*spherical) growth sequence* $\{\delta(G,r,n)|n = 1,2,...\}$, where $\delta(G,r,n)$ denotes the number of vertices at a distance n from r. By taking the generating function for $\delta(G,r,n)$,[6] we get the *growth function* of graph G rooted at r

$$\Delta(G,r;x) = \sum_{n=0}^{\infty} \delta(G,r,n)x^n$$

The vertex r, upon which the growth function depends, is called the *root*. We can extend the definition by allowing the root to be any induced subgraph R of graph G with the distance between some vertex u from G and the root R defined as the shortest path between u and any vertex from the root graph R. In particular, the expression $\delta(G,R,0)$ equals the number of vertices in R. Two root types are of a special importance. If the root consists of a single vertex, it is known as a *vertex root*, and if it consists of two adjacent vertices, it is called an *edge root*. The growth functions of all vertex roots can be used calculate the Wiener index of a graph.[7]

In the next section we show how the growth function can be calculated recursively for all catacondensed benzenoids. Furthermore, the same process can be used for calculating the growth function of nonplanar catacondensed benzenoid graphs (see *helicenes*[8]). First we present the calculation of the growth function in case of a vertex root, followed by the growth function of an edge root. Since adjacent hexagons share a common edge, the above calculations can be reused when the root is an arbitrary connected subgraph, the whole hexagon for example. We conclude section 2 with an example of graphs in which the growth depends only on the number of hexagons they consist of and calculate the growth of an infinite linear hexagonal chain.

The properties of vertices of graph G that are the farthest from the root R are examined in section 3. If we include the edges of G that have both endpoints in those vertices, we get an induced subgraph of graph G called a *distance-residual graph* and denoted $Res(G, R)$.[9] In section 3 we present the distance-residual graphs of some catacondensed benzenoid graphs without calculating the growth functions. There are two main results from this section:

Proposition 1. Let G be a catacondensed benzenoid graph with a vertex root R. Then the distance-residual graph $Res(G, R)$ is a union of nonadjacent vertices.

Proposition 2. Let G be a catacondensed benzenoid graph with an edge root R. Then the distance-residual graph $Res(G, R)$ is induced on one of these alternatives: (1) a union of nonadjacent vertices, (2) a pair of adjacent vertices, (3) a union of 1. and 2., and (4) two pairs of adjacent vertices.

We conclude the paper with some remarks about the possibility of applying the algorithm for growth calculation to the class of pericondensed benzenoids or to coronoids.

Some standard notation from graph theory is used in this paper;[10] K_n denotes a complete graph on n vertices, and C_6 denotes a hexagon or, as known in graph theory, a cycle on six vertices. The distance between the vertices u and v, i.e., the shortest path between them, is denoted by $d(u,v)$.

† Dedicated to Professor Nenad Trinajstić on the occasion of his 70th birthday.

* Corresponding author e-mail: primoz.luksic@fmf.uni-lj.si.

10.1021/ci600508s CCC: $37.00
Published on Web 05/10/2007

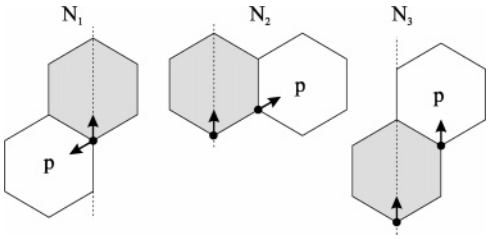


Figure 1. Different possibilities for selecting the vertex root in naphthalene. The starting hexagons are gray, and the arrows represent the direction of the growth in the adjacent (white) hexagons. Dotted lines are used to show the symmetries.

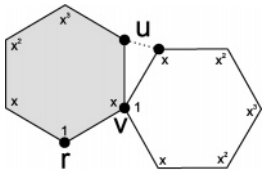


Figure 2. Calculating the growth function of graph N_2 from Figure 1 depicting naphthalene with the root r. The hexagons are separated to show how to determine the growth function recursively. The starting hexagon is gray. The growth of the adjacent hexagon (in white) with the root v gets multiplied by x since $d(r,v) = 1$.

THE GROWTH FUNCTION

We start with a simple hexagon graph C_6 depicting benzene. It is a symmetric graph, meaning that if the root is a single vertex, the growth function is the same no matter which vertex we choose. More precisely, there exists an automorphism of C_6 that maps any vertex of C_6 to any other vertex. Thus we can write $\Delta(C_6,K_1;x) = 1 + 2x + 2x^2 + x^3$, where the use of K_1 explicitly shows an independence from the root. The same is true for the edge root. In that case the growth function is $\Delta(C_6,K_2;x) = 2 + 2x + 2x^2$.

For the next simplest graph depicting naphthalene, the above observations no longer hold. Due to the symmetry, there are still some choices of a vertex root that give the same growth function, but in general we must consider different cases depending on the selection of the vertex. We look at this problem from another perspective by selecting an arbitrary vertex from the first hexagon, called the *starting hexagon* and shown in gray in all figures. Next, we find all possible positions for the adjacent hexagon thereby getting all possible growth functions. There are six possible placements, but because of the symmetry only three give different growth functions as shown in Figure 1. Flipping the graphs over the dotted lines gives us the other three possibilities that generate the same growth as the original graphs.

To efficiently calculate the growth functions of graphs in Figure 1, we follow a recursive approach of determining the growth of the adjacent hexagon and using it to calculate the growth of the whole graph. Special care must be taken to avoid the possibility of counting the same vertex twice, in particular if the vertices u and v lie on the common edge of two adjacent hexagons. This can be avoided by selecting the vertex which is closer to the root r, for example v in Figure 2, and using the growth function of the adjacent hexagon as the contribution of v to $\delta(G,r,n)$, where $d(v,r) = n$. As a result, the vertex u is already included in the growth function of the adjacent hexagon and must not be separately counted in the starting hexagon (see Figure 2).

Let us consider the graph N_1 in Figure 1. We use the above conclusions to write the growth function as $\Delta(N_1,r;x) = p + x + 2x^2 + x^3$, where p is the growth function of the adjacent hexagon. In order to calculate p, we need to determine the root of the adjacent hexagon, i.e., the vertex closest to the root of N_1. The arrows in Figure 1 help us disambiguate the root vertices of each hexagon and show how the growth "spreads'' through the graph. We see that $p = \Delta(C_6,K_1;x)$, therefore

$$\Delta(N_1,r;x) = 1 + 3x + 4x^2 + 2x^3 \quad (1)$$

Similarly, we obtain

$$\Delta(N_2,r;x) = 1 + (1 + p)x + x^2 + x^3$$

$$= 1 + 2x + 3x^2 + 3x^3 + x^4 \quad (2)$$

$$\Delta(N_3,r;x) = 1 + 2x + (1 + p)x^2$$

$$= 1 + 2x + 2x^2 + 2x^3 + 2x^4 + x^5 \quad (3)$$

To calculate the growth of the naphthalene when the root is in a different position in the starting hexagon, we just rotate the whole graph until we get one of the possibilities, shown in Figure 1, and then use the respective formula 1, 2, or 3.

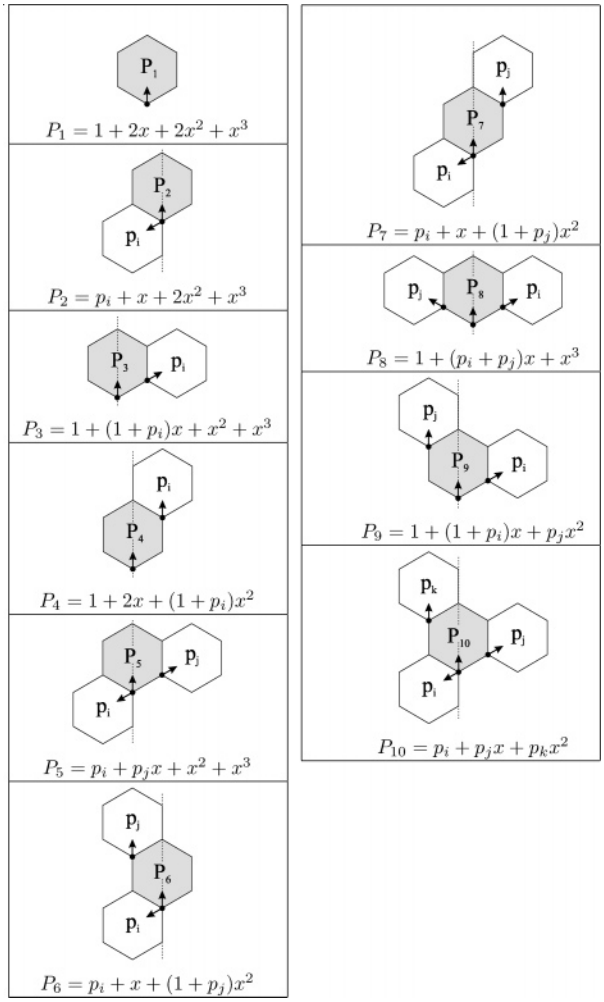
If an edge root is taken in the starting hexagon, then there are six possible placements of the adjacent hexagon to form a naphthalene, but only four of them have different growth functions (see Table 2). As an example, the growth for the one shown in Figure 3 is calculated. The root R consists of adjacent vertices r_1 and r_2. By following the before mentioned procedure of recursive calculation, we arrive at the problem of selecting the vertex of the adjacent hexagon that is closest to the root. Moreover, both hexagons have the same root. The solution is in selecting one of the root vertices (in our case r_1) and proceeding as before. This can be seen in Figure 3. As a result, the growth function is $\Delta(N_4,R;x) = \Delta(C_6,K_2;x) + 2x + 2x^2 = 2 + 4x + 4x^2$.

When generalizing these results to catacondensed benzenoids of arbitrary size, we incorporate the recursive rule of calculating the growth in the following algorithm. Its inputs are the catacondensed benzenoid graph G with a vertex or an edge root R and the output is the growth function $\Delta(G,R;x)$.

```
GROWTH(G, R)
  INPUT: graph G, subgraph R
  OUTPUT: growth function Δ(G, R; x)
{
  let S be the hexagon containing the root R;
  label hexagons adjacent to S in succession → S_i;
  if no such hexagon exist
    return Δ(G, R; x);
  for each i {
    let the vertices of S_i closest to R define a new root → R_i;
    by removing the vertices of S \ S_i let G_i be the
    connected component of G that includes S_i;
    Δ(G_i, R_i; x) := GROWTH(G_i, R_i);
  }
  calculate Δ(G, R; x) using only the vertices of S \ ⋃_i S_i
  and add Δ(G_i, R_i; x) · x^{d(R_i,R)} for each i;
  return Δ(G, R; x);
}
```

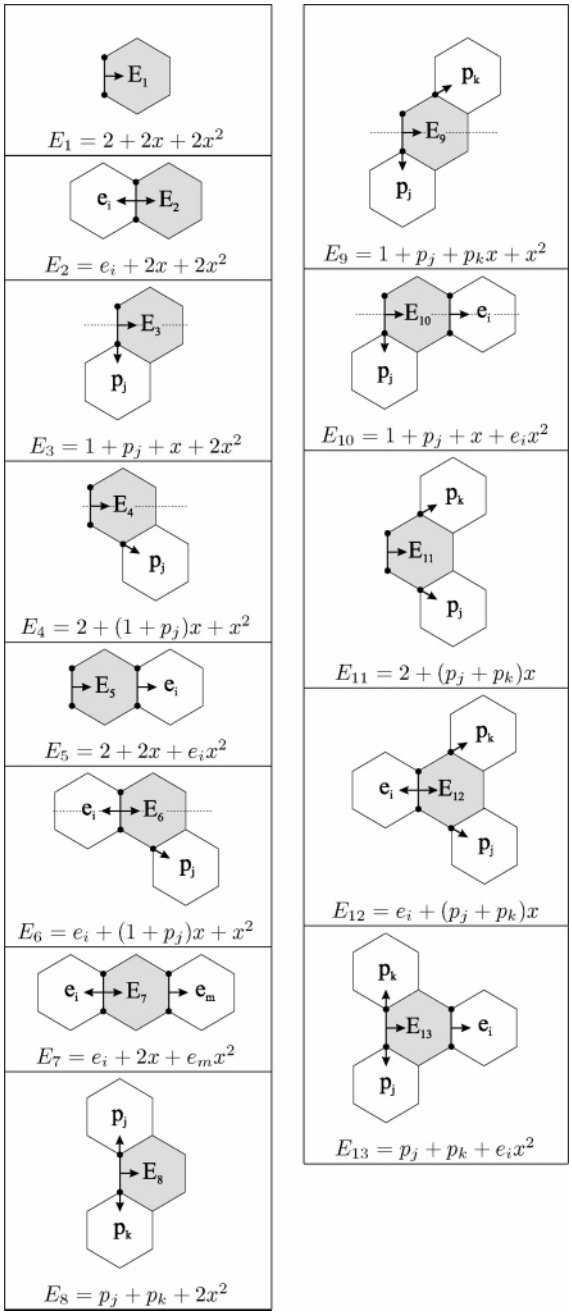
Let us explain the algorithm in more detail. It starts by finding the hexagon in which the root is located, i.e., the starting hexagon. Next, all hexagons adjacent to the starting

Table 1. Possible Starting Hexagon Types with a Vertex Root (i, j, $k \in \{1, 2, 3, 4, 5\}$)[a]



[a] Dotted lines show the possible symmetries.

Table 2. Possible Starting Hexagon Types with an Edge Root (i, $m \in \{1, 2, 3, 4\}$; j, $k \in \{1, 2, 3, 4, 5\}$)



[a] Dotted lines show the possible symmetries.

hexagon are determined (S_i). If there are none, then the recursion is finished and the growth of the starting hexagon is returned. Depending on the root this can either be $\Delta(C_6,K_1;x)$ or $\Delta(C_6,K_2;x)$. If adjacent hexagons exist, a vertex v having the shortest distance to the root is determined in each of them. The vertex v then becomes the root of the respective hexagon (R_i). If there are two vertices that have the shortest distance to the root, then they both become a new root (more precisely, the subgraph induced on them).

In order for the recursion to work, we must define the subgraphs G_i in the direction of each of the adjacent hexagons S_i. They are the connected components of the graph G that arise when deleting the vertices of $S / \cup_i S_i$ with G_i including the hexagon S_i for each i. The algorithm is then applied on the subgraphs G_i with the new roots R_i. Furthermore, the hexagons S_i are the starting hexagons in the new calls of the algorithm, so to distinguish them from the starting hexagon S, all of the starting hexagons acquired during recursion are named *derived hexagons*. Similarly, the roots are denoted *derived roots*.

The last step requires constructing the growth function from the growth functions of subgraphs G_i. First, we determine the growth of the starting hexagon without the vertices shared with adjacent hexagons since their growth is included in the growth functions of subgraphs G_i. Then we add the growths of the subgraphs in the right positions within the generating function. Here, the distance between the root of the subgraph and the root of the whole graph, denoted as

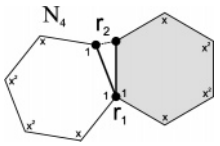
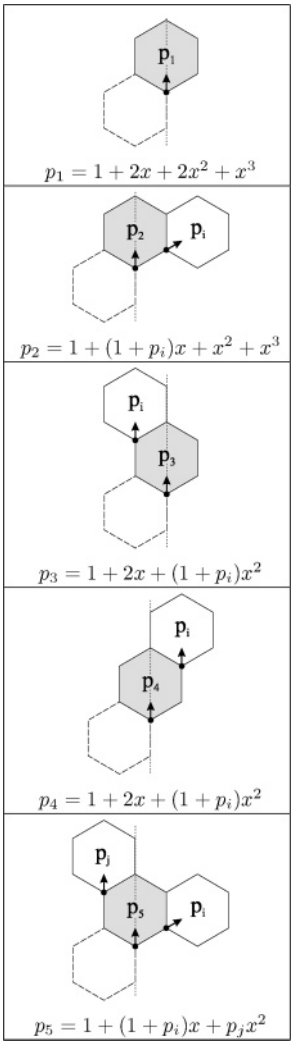


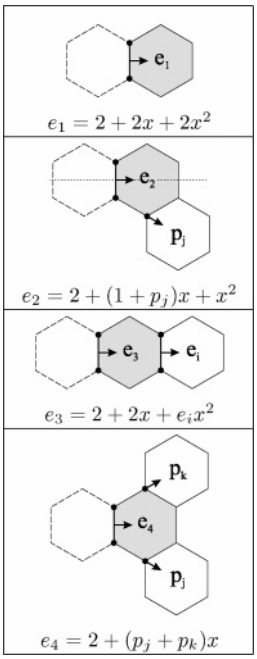
Figure 3. Calculating the growth function of a graph N_4 depicting naphthalene with an edge root r_1r_2. The hexagons are separated to show how to determine the growth function recursively. The starting hexagon is gray. The growth of the adjacent hexagon (in white) gets multiplied by 1 since the roots of both hexagons are the same.

Table 3. Possible Derived Hexagon Types When the Derived Root Is a Vertex ($i, j \in \{1, 2, 3, 4, 5\}$)



[a] Dotted lines show the possible symmetries.

Table 4. Possible Derived Hexagon Types When the Derived Root Is an Edge ($i \in \{1, 2, 3, 4\}$; $j, k \in \{1, 2, 3, 4, 5\}$)



[a] Dotted lines show the possible symmetries.

$d(R_i,R)$, is important since the growth function of the subgraph G_i is added to $\delta(G,R,n)$ if $d(R_i,R) = n$.

The algorithm uses the well-known breadth-first search (BFS) method of examining a graph, which can be used generally to calculate the growth of any connected graph $G = (V,E)$. Instead of using the BFS on the whole benzenoid graph, we focus on its *inner dual* $\mathscr{G} = (\mathscr{V}, \mathscr{E})$,[1] where each vertex represents a whole hexagon. This approach maintains the time complexity of the classical BFS, i.e., $O(|V|+|E|)$, since $|\mathscr{V}| = (|V|-2)/4$ and $|\mathscr{E}| = (|E|-6)/5$. However, as shown further on, the inner dual representation allows the growth to be presented using a system of equations, which is especially useful in the case of infinite graphs and graphs having repetitive substructures.

The algorithm must calculate the distances between roots in each step of the recursion. We can avoid this process by finding all possible types of starting and derived hexagons with different growth functions and using them as an input instead of the graph G. As can be seen in Tables 1 and 2 there are 10 types of starting hexagons when the root is a vertex and 13 types when the root is an edge. Their growth functions are labeled P_i, $i = 1,2,...,10$, and E_i, $i = 1,2,...,13$, respectively. The other possible types can be obtained by mirroring the existing ones over the dotted lines (if they exist) and have the same growth functions as the originals. Tables 3 and 4 show the derived hexagons which can be obtained only in the process of recursion. Their growth functions are $p_1,...,p_5$ with the vertex as a derived root and $e_1,...,e_4$ if the derived root is an edge.

To shorten the writing, the hexagon types and their growths use the same notation where possible.

Some interesting conclusions follow. First, if the root is a vertex, the derived roots can also only be vertices. If the root is an edge, we can obtain an edge or a vertex as a derived root. But, if a vertex is obtained, it is not possible to get an edge as a derived root in the next steps of recursion. These facts are further used in section 3 when determining the distance-residual graphs.

The algorithm can be generalized by taking an arbitrary connected subgraph of some graph G for the root. The idea here is to calculate the growth of the hexagons which include the root in the same way as with the starting hexagons in Tables 1 and 2. The only difference is that we can have a group of starting hexagons instead of just one. The other steps of the algorithm are the same since the derived hexagons can only be the ones from Tables 3 and 4. This is due to two adjacent hexagons having the same edge, so the derived root can either be a vertex or an edge.

Predefined hexagon types enable us to easily calculate the growth functions of graphs that include repetitive structures, i.e., hexagons of the same type. But even if a graph consists of different types, it helps if the types have the same growth. One such example are the derived hexagon types p_3 and p_4 (see Table 3). By using these two types we can construct a number of different graphs with the growth functions depending only on the number of hexagons the graphs have. Graphs in Figure 4 have the starting hexagon P_4, use the

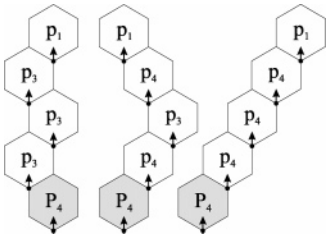


Figure 4. Three benzenoid graphs having the same growth function of $(x^{11}-1)(x+1/x-1)$ calculated by formula 5.

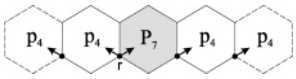


Figure 5. The two-way infinite hexagonal linear chain H with the root r.

same number of types p_3 and p_4 in total, and end with the type p_1. It follows that they all have the same growth function. We can calculate it in general when the number of hexagons is n:

$$P_4 = 1 + 2x + (1 + p_3)x^2 = 1 + 2x + x^2 + (1 + 2x + x^2 + p_3x^2)x^2 =$$

$$= 1 + 2x + 2x^2 + 2x^3 + x^4 + (1 + 2x + x^2 + p_3x^2)x^4 = \cdots =$$

$$= 1 + 2x + 2x^2 + \cdots + 2x^{2n-3} + x^{2n-2} + (1 + 2x + 2x^2 + x^3)x^{2n-2} =$$

$$= 2\sum_{i=0}^{2n} x^i + x^{2n+1} - 1 = 2\frac{x^{2n+1} - 1}{x - 1} + x^{2n+1} - 1 =$$

$$= (x^{2n+1} - 1)\frac{x + 1}{x - 1} \qquad (4)$$

Another use of preprocessing the possible hexagon types is in calculating the growth of infinite graphs consisting of the same hexagon types, the two-way infinite hexagonal linear chain H for example (see Figure 5). If we take a trivalent vertex r for the root, we get a system of equations for the growth

$$P_7 = p_4 + x + (1 + p_4)x^2$$

$$p_4 = 1 + 2x + (1 + p_4)x^2$$

from which it follows $\Delta(H,r;x) = P_7 = (1 + x)^2/(1 - x) = 1 + 3x + 4x^2 + 4x^3 + \ldots$

Note that not all types of derived hexagons from Tables 3 and 4 are feasible in order to create a benzenoid structure. The problem is that overlapping of hexagons could occur which is possible for helicenes but not with the catacondensed benzenoids which are planar by definition.

DISTANCE-RESIDUAL GRAPHS

In this section we present the distance-residual graphs of catacondensed benzenoids, i.e., the induced subgraphs on vertices farthest away from the root. We use some observations about the hexagon types defined in section 2 to prove the propositions mentioned in the Introduction. The construction of the distance-residual graphs, proved possible in the propositions, is not hard and therefore left to the reader.

Let G be a catacondensed benzenoid graph. If its starting hexagon has a vertex root R, then it belongs to one of the types in Table 1. Each of the derived hexagons also has a vertex root (see Table 3). Hence, the distance-residual graph $Res(G, R)$ is the same as the distance-residual graph $Res(C_6,T)$ of a hexagon with a vertex root T, which is K_1. In some cases $Res(G, R)$ can include more vertices, the simplest being if we take a starting hexagon of type P_2 and its derived hexagon of type p_1. But no two vertices are adjacent since each lies in a different hexagon. Proposition 1 is thus proven.

Next, we look at the graph G with an edge root R. From the possible derived hexagon types in Tables 3 and 4 it can be seen that a derived hexagon either has a vertex root or an edge root. Thus the distance-residual graph of such a hexagon can either be K_1 or K_2. As in the case above, the distance-residual graph can include more disjunct vertices. It can also have a pair of adjacent vertices, alone or together with the nonadjacent ones. The latter is possible only if the starting hexagon is of type E_6, E_7, E_{10}, E_{12}, or E_{13}. The $Res(G, R)$ can also be made of two pairs of adjacent vertices. This can be achieved in two ways. Either by using the type E_2 as starting and type e_1 as the derived hexagon or type E_7 as starting and types e_3 and e_1 as the derived hexagons.

To complete the proof of proposition 2, we show there are no other possibilities for the $Res(G, R)$. There are some facts that follow from looking at the possible starting and derived hexagons. The first being that once a derived hexagon has a vertex root, all the hexagons derived from it have a vertex root also (see Table 3). The second observation is that a hexagon having an edge root can only have one derived hexagon with also an edge root (see Table 4). Since any starting hexagon can have at most two derived hexagons with an edge root (see Table 2), there cannot be more than two pairs of adjacent vertices in $Res(G, R)$.

CONCLUSION

The algorithm for calculating the growth of catacondensed benzenoids can be used with any connected subgraph as the root. We have also proven it can be used if the graph is not planar, i.e., in the case of helicenes.[7] When trying to adapt it for use on pericondensed benzenoids, some problems occur. Since a vertex can be included in three hexagons, we would need to be careful not to count it more than once in the calculation of growth function. But the procedure for determining such vertices is not as simple as the one presented in our algorithm, which is based on the distance between roots. A similar problem occurs when looking at coronoids or graphs with the root that is not connected. The same derived hexagon can be taken twice, thereby counting its growth function twice. Therefore, the existing algorithm cannot be used on these types of graphs.

ACKNOWLEDGMENT

Research was supported by grant no. 1000-05-310004 from the Research Agency of Slovenia and by grant no. P1-0294 from the Ministry of Higher Education, Science and Technology of Slovenia.

REFERENCES AND NOTES

(1) Trinajstić, N. Chemical graphs. In *Chemical graph theory*, 2nd ed.; CRC Press: Boca Raton, FL, 1992; pp 30−31.
(2) Gutman, I.; Cyvin, S. J. *Introduction to the Theory of Benzenoid Hydrocarbons*, 1st ed.; Springer-Verlag: Berlin, Germany, 1989.
(3) Brunvoll, J.; Cyvin, B. N.; Cyvin, S. J. Enumeration and Classification of Coronoid Hydrocarbons. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 14−21.
(4) Fleischner, H. J.; Geller, D. P.; Harary, F. Outerplanar graphs and weak duals. *J. Indian Math. Soc. (N.S.)* **1974**, *38*, 215−219.
(5) Pisanski, T.; Tucker, T. W. Growth in products of graphs. *Australas. J. Combin.* **2002**, *26*, 155−169.
(6) Tucker, A. Generating functions. In *Applied combinatorics*, 4th ed.; John Wiley & Sons: New York, 2001; pp 241−272.
(7) Lukšič, P.; Pisanski, T. Growth in [*n*]helicenes. *Croat. Chem. Acta* **2007**, *80*, to appear.
(8) Randić, M.; Nikolić, S.; Trinajstić, N. Enumeration of Kekulé Structures for Helicenic Systems. *Croat. Chem. Acta* **1988**, *61*, 821−831.
(9) Lukšič, P.; Pisanski, T. Distance-residual graphs. 2006, arXiv:math/0609810. arXiv.org ePrint archive. http://arxiv.org/abs/math.CO/0609810 (accessed Nov 30, 2006).
(10) Gross, J. L.; Yellen, J. Fundamentals of graph theory. In *Handbook of graph theory*, 1st ed.; CRC Press: Boca Raton, FL, 2004; pp 2−20.

CI600508S