Pohl-Warnsdorf – Revisited

Ira Pohl Larry Stockmeyer

Computer Science Department
University of California, Santa Cruz

June 24, 2004

Abstract. Two new series of graphs are introduced. Properties of these graphs make them suitable as test beds for Hamiltonian path heuristics. The graphs are planar and regular of degree 3, and each series has a simple recursive definition. These graphs are "conceptual adversaries" to methods that try to exploit local characteristics. For each series, we give an analysis showing that the number of Hamiltonian paths in these graphs grows polynomially in the number of nodes of the graph. For both series, the analysis gives the degree of the polynomial. For one of the series, the number of Hamiltonian paths is determined exactly. A companion paper (LaFall and Pohl 2004) gives experimental results from applying the Pohl-Warnsdorf heuristic to these graphs.

1 Introduction

The Pohl-Warnsdorf rule is a greedy heuristic that tries to find a longest path in a graph. It was based on the work of the 19th century mathematician Warnsdorf, who proposed it as a rule for finding Knights tours. A Knights tour is a Hamiltonian path where the graph is the 64 chessboard squares and the edge connectivity is that of the Knights move from square to square. This was long considered a difficult task for even human chessmasters, for example, the Belgium International Master George Koltanowski used to be famous for giving it as part of his teaching lectures. Warnsdorfs rule is "go to a next square which has the fewest ways out." On the chessboard this means favoring corner squares over all others and edge squares in preference to central squares.

In (Pohl 1967), we pointed out that the Warnsdorf rule could be generalized to finding an arbitrary Hamiltonian path in a graph by making the rule be: "go to the next node of least degree." We tested this on the chessboard and found that it rarely failed. We modified the rule to be recursive in tie-breaking situations, the so-called Pohl-Warnsdorf rule, and demonstrated its effectiveness on the Knights tour problem. We also tested it on an important graph of Tutte's (Tutte 1946) of 46 nodes; see Figure 1. This is a regular planar graph of degree 3, and its significance in graph theory was that it disproved Taits conjecture (Wikipedia-Tait) for the 4 color problem.

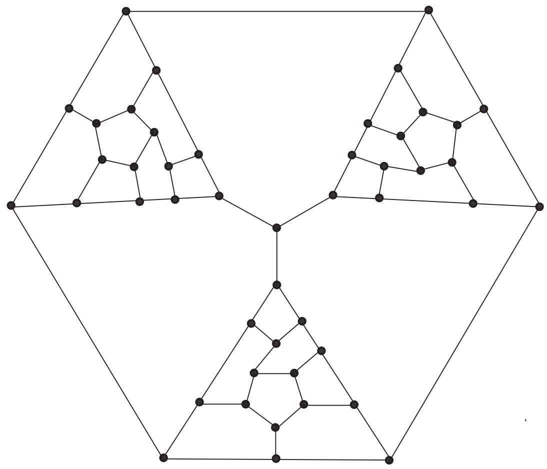


Figure 1: Tutte's graph

2 The Pohl-Warnsdorf Algorithm

The algorithm was originally coded in Algol60 (Pohl 1967) and has been recoded in C# (Pohl 2002), (LaFall and Pohl 2004). The details can be found in those papers but the essential algorithm is given in the pseudocode shown in Figure 2.

There were several reasons at that time to use Tutte's graph as a test case. It was a well known graph in the graph theory literature on Hamiltonian paths; it was a largish problem (46 nodes), and it was a conceptual adversary for the idea imbedded in the Pohl-Warnsdorf rule. By a conceptual adversary we mean that the regular character of the graph gives the heuristic no initial means to discriminate; in effect seemingly making it no better than randomly going to a next available node.

Todays desktop machines are at a minimum 1000 times more powerful than the machines on which this work was first carried out, namely the Burroughs 5500. Since not much has been done with this heuristic in the interim, it was felt that much larger problems could be attempted. Also more experimental results would give further insight into the efficacy and limitations of this scheme. For example Don Knuth in a private communication felt the algorithm would work on the generalization of Knights tour graphs: namely on Leaper graphs (Knuth 1994) on much larger boards.

While looking to test Pohl-Warnsdorf on larger graphs, Tutte's graph was re-examined and two characteristics were observed that had not originally been noticed. First Tutte's graph has three isomorphic subgraphs each having 15 nodes. Second when these are replaced as super-nodes the K_4 (complete graph of 4 nodes) is generated. This led to looking at what recursive schemes would generate Tutte style graphs.

Input: graph
Output: path that does not revisit a node (if complete, a Hamiltonian path)
For each node i (the Start Node):
    Graph is the input graph;
    Current Node is i;
    Repeat until all nodes are reached or a dead-end:
        Current Node is placed on the longest path;
        From the Current Node select an Adjacent Node that has least degree;
        If unique this becomes the new Current Node;
        Else if there is more than one such node:
            For each such node j, let tiebreak(j) be the least degree of an adjacent node of j
              other than the Current Node;
            Choose a j having smallest tiebreak(j) as the new Current Node
              (if not unique then break tie arbitrarily);
        Delete the old Current Node from the graph;
Output a longest path found

Figure 2: Pseudocode for the Pohl-Warnsdorf heuristic Longest Path Algorithm

3 New Series of Graphs

3.1 Recursive Tutte Style Graphs

The first such sequence that was found are the RT_k graphs (Recursive Tutte style graphs). The idea was to take K_4 and look at it as three corner nodes and a center node; see Figure 3 (left). Informally, we wish to replace the corner nodes by K_3 and reconnect them with a center node. However doing this naively would make some of the nodes degree 4 nodes. [This needs clarification.] Instead we view each isomorphic graph as a 3 node cluster and each of these are then hooked to the center node. The isomorphic graphs are K_3 triangles, where we consider each as having a left, right, and bottom node. The bottom node connects to the center and the left node connects to the adjacent isomorphic graph's right node. So RT_2 looks like Figure 3 (right)

The key point is that this process is recurrent. Take a given graph RT_k and remove its center node and the links to its center node. This creates three degree 2 nodes which we call left, right, and bottom. Now take the resulting RT'_k graph as one of three isomorphic components of RT_{k+1}. Then create a center node and hook the three bottom nodes to it. For adjacent components hook their left nodes to the adjacent component's right node. The resulting graph is a planar regular graph of degree 3. These graphs are $4, 10, 28, 82, \ldots, 3^k + 1$ nodes in size. For example, Figure 4 shows RT_3 (left) and RT'_3 (right).

The number of Hamiltonian paths in K_4 is 24. The graph is complete. We at first wrote an enumerative computer program to count the Hamiltonian paths in these graphs; see (LaFall and Pohl 2004). This produced 204 paths for RT_2 and 2688 paths for RT_3. Later these results were confirmed by the analysis given in Section 8. The same program enumerated 67,212 paths for Tutte's 46 node graph.

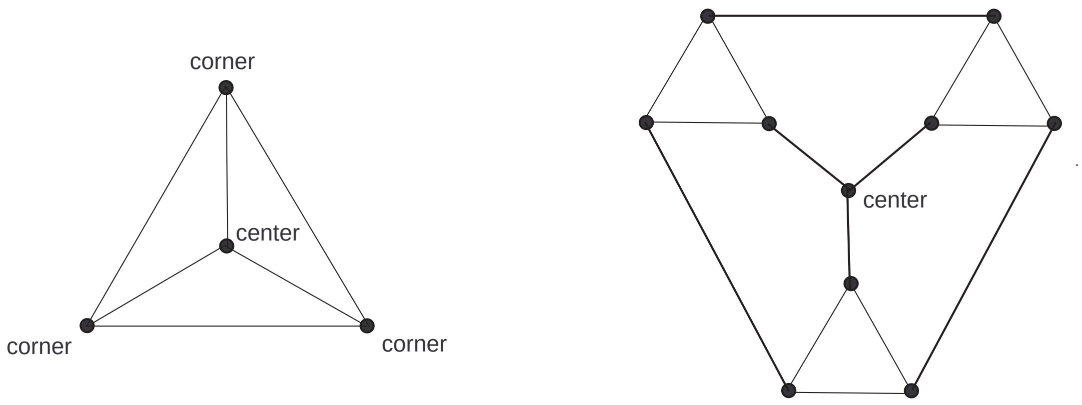


Figure 3: K_4 labeled with corners and center (left). The RT_2 graph (right) obtained by replacing each corner of K_4 by a triangle.

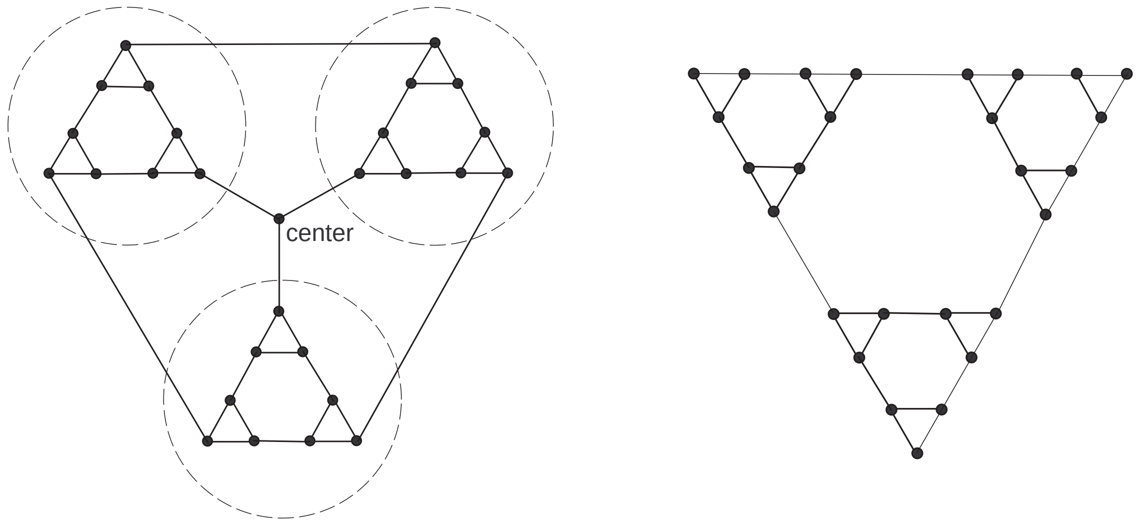


Figure 4: The RT_3 graph (left). The component within each dotted circle is a copy of RT_2', where RT_2' is RT_2 with its center removed. The RT_3' graph (right) is obtained by removing the center from RT_3. Each component RT_2' in RT_3' has been flipped to maintain planarity at the next step, RT_4.

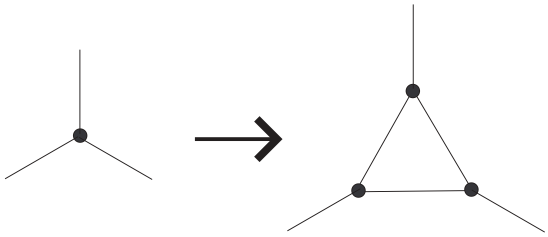


Figure 5: A fractal-like transformation for growing 3-regular graphs.

Looking at RT_3' in Figure 4 (right), one sees that the graph has a self-similar, or fractal, nature. In fact, the RT' series appears similar to one of the classic examples of fractal geometry, the Sierpinski triangle (Wikipedia-Sierpinski). (However, the sequence of iterates leading to the Sierpinski triangle has a somewhat different iterative definition, and when viewed as a graph all nodes have degree 4, except for three nodes of degree 2.) There is an alternate "fractal-like" way to generate the RT series. Starting from $\mathrm{RT}_1 = K_4$, the graph RT_{k+1} is obtained from RT_k by applying the transformation shown in Figure 5 to every node of RT_k except the center node. Similarly, starting from RT_2', the graph RT_{k+1}' is obtained from RT_k' by applying the transformation to every node of degree 3.

The recursive character of these graphs and their relative sparseness makes them a useful test bed for finding longest paths. It is very easy for an algorithm to dead end in a local isomorphic sub-component. Unless a path leads to one of the three (bottom, left, right) exit nodes you are locked in a subcomponent. They of course as mentioned earlier are a conceptual adversary to methods that try to exploit local characteristics.

3.2 Binary Tree Graphs

After discovering the above construction, we looked for other means to build interesting test sets. For example Tutte's graph is not in the RT series, although it could be used as a base case for its own recursive series. However an interesting second set of graphs was discovered by looking at an alternate way of constructing RT_2. The base building block here is the graph K_3. This graph is a simple binary tree of depth 1 whose leaf nodes are connected. This suggests the following:

Pseudocode for $\mathbf{BT}_k$ series
Construct BT_{k-1}' as follows :
    Construct a complete binary tree of depth $k-1$.
        //All internal non-root nodes are degree 3.
    Connect all leaf nodes from left-to-right.
        //All leaf nodes except the leftmost and rightmost are degree 3.
Create a center node and connect it to the roots of three copies of BT_{k-1}'.
    //Center and root nodes are degree 3.
Finally connect leftmost leaf node to rightmost leaf node of adjacent BT_{k-1}' components.

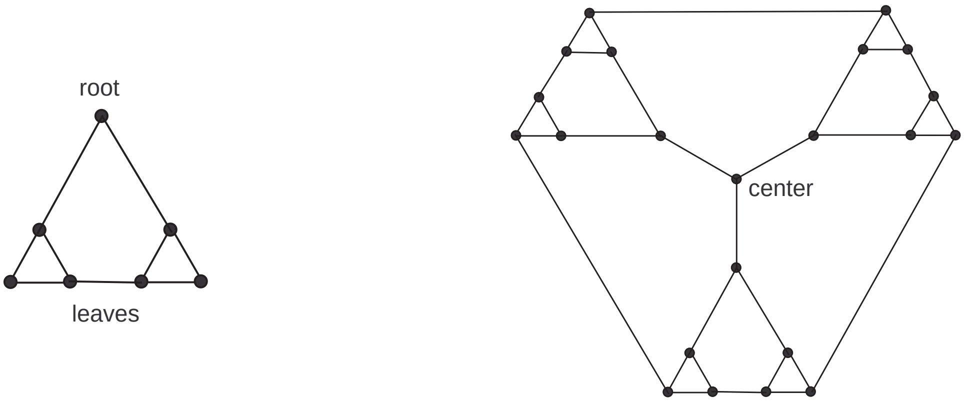


Figure 6: The BT_2' graph (left). This component of 7 nodes becomes part of the BT_3 graph of 22 nodes (right).

For example, BT_2' and BT_3 are shown in Figure 6.

The resulting graph is a planar regular graph of degree 3. The BT graphs are $4, 10, 22, 46, \ldots, 3 \cdot 2^k - 2$ nodes in size.

Our enumerative computer program to count the Hamiltonian paths in these graphs produced 204 paths for BT_2, 1524 paths for BT_3, and 10,740 paths for BT_4. Later these results were confirmed by the analysis given in Section 9. These graphs are also relatively sparse making them a useful test bed for finding longest paths. They are also intuitively satisfying to the computer scientist because of their use of the complete binary tree as part of their architecture. We suspect they may be useful in other domains, such as network architectures.

4 Analytical Results for the Number of Hamiltonian Paths

We consider Hamiltonian paths to be oriented from one end node to the other. Let $\mathit{Ham}(G)$ be the number of Hamiltonian paths in G. For example, $\mathit{Ham}(K_n) = n!$. If the orientation of paths is ignored, the number of unoriented Hamiltonian paths in G is $\mathit{Ham}(G)/2$.

In Sections 8 and 9 we give recurrence equations for $\mathit{Ham}(\mathrm{RT}_k)$ and $\mathit{Ham}(\mathrm{BT}_k)$, respectively. From these equations we give an exact closed form expression for $\mathit{Ham}(\mathrm{RT}_k)$, and we determine $\mathit{Ham}(\mathrm{BT}_k)$ to within a constant factor.

Theorem 1 $\mathit{Ham}(RT_k) = \frac{8}{13}16^k + 2 \cdot 4^k + \frac{18}{13}3^k + 2.$

Letting n be the number of nodes of RT_k and $\lambda = \log_3 16 = 2.52371\ldots$,

$$\mathit{Ham}(RT_k) = \frac{8}{13}(n-1)^\lambda + 2(n-1)^{\log_3 4} + \frac{18}{13}n + \frac{8}{13}.$$

The second part of Theorem 1 follows immediately from the first part by setting $k = \log_3(n-1)$ because $n = 3^k + 1$ for RT graphs. The second part says that the number of Hamiltonian paths in RT_k grows polynomially in n where the degree of the polynomial is $\lambda = \log_3 16 = 2.523\ldots$.

Theorem 2 *Let α be the (unique) real number satisfying $\alpha^3 - 2\alpha^2 - 3\alpha + 4 = 0$ and $2.5 \leq \alpha \leq 2.6$ ($\alpha = 2.56155\ldots$), and let $\beta = \alpha^2$ ($\beta = 6.56155\ldots$).*

$$1.46\beta^k \leq Ham(BT_k) \leq 21.7\beta^k + 12 \cdot 5.2^k + 14.4 \cdot 2.6^k.$$

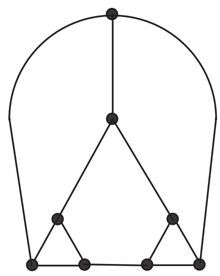
Letting n be the number of nodes of BT_k and $\theta = \log_2 \beta = 2.71403\ldots$,

$$0.074(n+2)^\theta \leq Ham(BT_k) \leq 1.11(n+2)^\theta + 0.88(n+2)^{2.38} + 3.17(n+2)^{1.38}.$$

The second part of Theorem 2 is immediate from the first part by setting $k = \log_2((n+2)/3)$ because $n = 3 \cdot 2^k - 2$ for BT graphs. The second part says that the number of Hamiltonian paths in BT_k grows polynomially in n where the degree of the polynomial is $\theta = \log_2 \beta = 2.714\ldots$. Because $\lambda < \theta$, the two theorems say that the BT graphs are (slightly) richer in Hamiltonian paths than the RT graphs, asymptotically for large enough n.

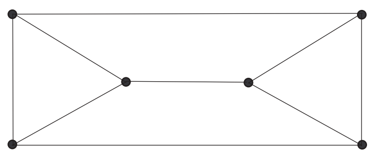
5 Hybrid Constructions

Returning to our original insight in building these graphs, we see that various of these components can be mixed and matched. Consider the 7 node leaf-connected complete binary tree as found in Figure 6 (left). It can be turned into a regular 8 node degree 3 graph by adding one node above the root. This node is then connected to the three corners of the binary tree "triangle".



In the same manner, any leaf-connected complete binary tree of $2^k - 1$ nodes can be turned into a regular 2^k node degree 3 graph.

We can also create other constructions starting with K_3 (the complete graph of 3 nodes). One 2-triangle construction is a 6 node graph.



This can be looked at as taking K_4 and replacing one corner node with a triangle. This pattern can be iterated as shown in Figure 7.

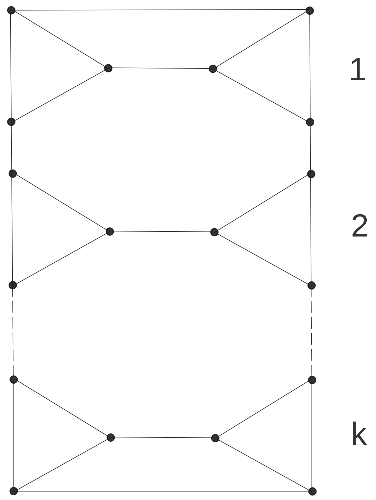


Figure 7: A graph of $6k$ nodes

The transformation in Figure 5 can be applied to produce various series of graphs by starting from various 3-regular graphs. For example, starting from K_4 would give a series similar to but not identical to the RT series.

An open problem is what constructions of 3-regular graphs yield the maximum density for Hamiltonian paths. Some bounds on the maximum number of Hamiltonain cycles in 3-regular graphs have recently been obtained by Eppstein (2003). A lower bound of $2^{n/3}$ on this maximum number follows from the construction of a $n = 6k$ node 3-regular graph having 4^k Hamiltonian cycles. Each Hamiltonian cycle in this graph corresponds to n Hamiltonian paths, by removing any one of the n edges from the cycle. He also shows that every n-node 3-regular graph has at most $2^{3n/8}$ Hamiltonian cycles. This does not directly imply anything about the maximum number of Hamiltonian paths, because a graph can have an exponential number of Hamiltonian paths but no Hamiltonian cycle.

6 Related Work

There has been interest recently in polynomial-time algorithms that approximate the longest path in a graph. Let us say that an *approximation algorithm for the longest path problem with guarantee* $g(L)$ is a polynomial-time algorithm which, when given any graph having a (simple) path of length L, returns a path of length at least $g(L)$. The currently best known result of this type, having $g(L) = c{\cdot}\log^2 L/\log\log L$ for some constant $c > 0$, is obtained by combining results of Björklund and Husfeldt (2003) and Gabow and Nie (2004). In particular, when given a Hamiltonian graph with n nodes, the algorithm always returns a path of length at least $c \cdot \log^2 n/\log\log n$. Vishwanathan (2000) had earlier obtained this result for graphs of bounded degree.

In addition, results about the nonapproximability of longest path are known. Karger, Motwani, and Ramakumar (1997) show that if P $\neq$ NP, then for no $\varepsilon > 0$ is there an approximation algorithm for longest path with guarantee εL. Under stronger assumptions about the complexity of NP, stronger nonapproximability results are proved. Because the L in these results can have a particular value much less than n, they allow for the possibility of guarantee εn on Hamiltonian graphs, for some $\varepsilon > 0$. A nonapproximability result for Hamiltonian graphs, also due to Karger, Motwani, and Ramakumar (1997), is that guarantee $n - n^{\varepsilon}$ is impossible if $\varepsilon < 1$ and $\mathrm{P} \neq \mathrm{NP}$. (If $\mathrm{P} = \mathrm{NP}$ then the longest path problem can be solved exactly (guarantee L) in polynomial time.) All of these nonapproximability results hold for graphs of bounded degree.

7 Further Work

Finally we hope to ascertain the efficacy of Pohl-Warnsdorf versus depth first search in finding longest paths in these graphs. Preliminary experiments (LaFall and Pohl 2004) suggest that these results strongly favor Pohl-Warnsdorf. Still Pohl-Warnsdorf is usually unsuccessful in producing a Hamiltonian path.

We hope to incorporate Pohl-Warnsdorf into HPA using a form of adaptive weighting to further bias the search to be very selective. We then intend to reapply it to these various benchmarks and test the efficacy of these searches. We expect the broadening of this search using HPA techniques will lead to significant improvement in finding longest paths. Computationally we hope to port these method in C# to a cluster using the .NET threading model.

8 Counting Hamiltonian Paths in RT_k

We first recall the recursive definition of the RT_k graphs. They are defined in terms of the related RT'_k graphs, and the RT'_k graphs have a recursive definition. Three nodes of RT'_k have degree 2; these nodes are called *ports*. The remaining nodes all have degree 3. It will follow from the recursive definition that the three ports of RT'_k are indistinguishable (formally, for each permutation of the ports, there is an automorphism of RT'_k that maps each port to its permuted image). First, RT'_1 is K_3, and the ports are the three nodes. The recursive construction of RT'_k for $k \geq 2$ is illustrated in Figure 8. The graph RT'_k is formed from three copies of RT'_{k-1}, labeled D, E, F, and three additional edges. One edge connects a port of D to a port of E, another edge connects a port of E to a port of F, and another edge connects a port of F to a port of D as shown. The three "unconnected" ports, t, w, and z, become the ports of RT_k.

Now the graph RT_k is formed from RT'_k by adding a center node c and three edges connecting c to the three ports of RT'_k; this is illustrated in Figure 9.

Because the graphs RT'_k are defined recursively, one might think that the number of Hamiltonian paths in RT'_k could be computed recursively. This is not correct because the number of Hamiltonian paths in RT'_{k-1} is not sufficient to calculate the number of Hamiltonian paths in RT'_k. Looking at Figure 8 we see that a Hamiltonian path in RT'_k might begin in one copy D of RT'_{k-1}, then leave through port r, proceed through E and F, re-enter D through port s, and finally end at port t. So for RT'_{k-1} we must know the number of ways that two paths can cover the nodes of RT'_{k-1} in a "Hamiltonian-like" way (i.e., each node appears in exactly one of the paths) where one path connects an arbitrary node to a port and the other path connects the remaining two ports. It turns out that it is sufficient to calculate four quantities for each k, as described next.

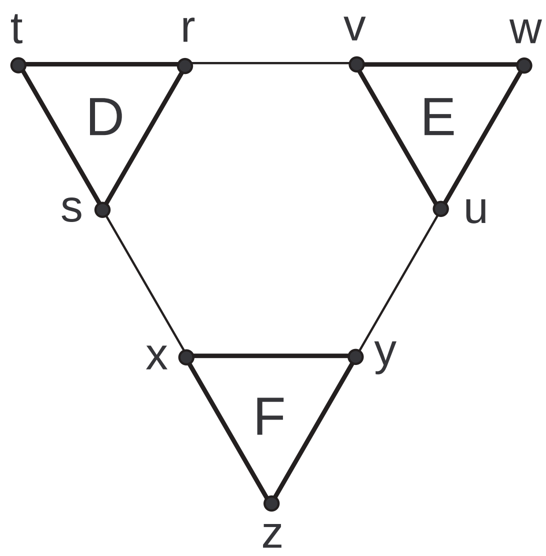


Figure 8: The recursive construction of RT'_k. Shown are three copies of RT'_{k-1}: one labeled D having ports r, s, t, one labeled E having ports u, v, w, and one labeled F having ports x, y, z. Three additional edges connect the ports as shown. The ports of RT'_k are t, w, z.

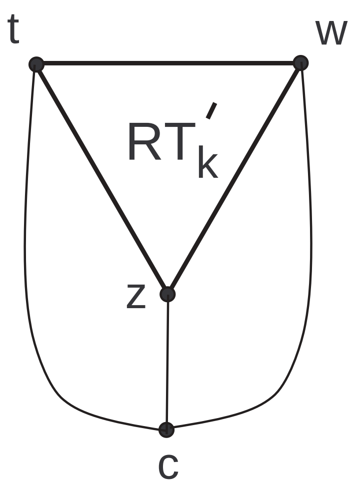


Figure 9: The graph RT_k.

Given a graph G and a collection of paths in G, we say that the paths *H-cover* G if each node of G belongs to exactly one of the paths. For example, a single path H-covers G iff it is a Hamiltonian path in G. Let t, w, z denote the three ports of RT'_k. Definitions of the four quantities are given next. Each of the quantities is the number of certain paths or pairs of paths. The paths here are *unoriented*, that is, a path and its reversal count as one (unoriented) path. A path may consist of a single node and no edges.

1. *"port-port" paths.* Let $[\text{P–P}]_k$ be the number of paths p that H-cover RT'_k and connect t and w. (Because the ports are indistinguishable, $[\text{P–P}]_k$ is also the number of H-covering paths that connect w and z, and the number that connect z and x. Similar comments, that the port names t, w, z can be arbitrarily permuted, apply to the remaining three quantities.)

2. *"port-any" paths.* Let $[\text{P–A}]_k$ be the number of paths p that H-cover RT'_k and connect t and a where a is an arbitrary node of RT'_k. (Necessarily $a \neq t$ because if $t = a$ then p would contain the same node twice. However, a could be one of the other ports, w or z.)

3. *"port-port, port-any" paths.* Let $[\text{P–P, P–A}]_k$ be the number of pairs (p_1, p_2) of paths that H-cover RT'_k such that p_1 connects t and w, and p_2 connects z and a, where a is an arbitrary node. (Necessarily a cannot be either t or w, but we may have $a = z$. In this case, p_2 is the path consisting of the single node z and no edges.)

4. *"port-any, port-any" paths.* Let $[\text{P–A, P–A}]_k$ be the number of pairs (p_1, p_2) of paths that H-cover RT'_k such that p_1 connects t and a_1, and p_2 connects w and a_2, where a_1 and a_2 are arbitrary nodes. (Necessarily $a_1 \neq a_2$, $a_1 \neq w$, and $a_2 \neq t$, although we could have $a_1 = t$ or $a_2 = w$.)

We first verify that these quantities suffice to compute the number of Hamiltonian paths in RT_k. Consider an arbitrary Hamiltonian path in RT_k; see Figure 9. There are two cases depending on whether the center node c is one end of the path or not. If c is an end, then the Hamiltonian path must proceed from c to one of the ports $q \in \{t, w, z\}$ of RT'_k and then proceed by an H-covering path to some node a of RT'_k. Because there are three choices for the port q and because we are counting the number of *oriented* Hamiltonian paths, the number of Hamiltonian paths belonging to the first case is $6[\text{P–A}]_k$. If c is not an end, then the Hamiltonian path must start at some node a_1 of RT'_k, leave through some port q_1, visit c, re-enter through some other port q_2, and proceed to some node a_2, where the path connecting q_1 and a_1 and the path connecting q_2 and a_2 must H-cover RT'_k. There are six ways to choose the ordered pair (q_1, q_2). Therefore, the number of Hamiltonian paths belonging to the second case is $6[\text{P–A, P–A}]_k$. This gives

$$Ham(\mathrm{RT}_k) = 6([\text{P–A}]_k + [\text{P–A, P–A}]_k). \tag{1}$$

To begin the recursive computation of the four quantities, first consider $k = 1$. Because RT'_1 is K_3, it can be seen directly that

$$[\text{P–P}]_1 = 1 \quad [\text{P–A}]_1 = 2 \quad [\text{P–P, P–A}]_1 = 1 \quad [\text{P–A, P–A}]_1 = 2. \tag{2}$$

Below we show that $[\text{P–P}]_k = 1$ for all k, and we derive the following recurrence equations for the other three quantities: for all $k \geq 2$

$$[\text{P–A}]_k = 2([\text{P–A}]_{k-1} + [\text{P–P, P–A}]_{k-1}) \quad (3)$$
$$[\text{P–P, P–A}]_k = [\text{P–A}]_{k-1} + 3\,[\text{P–P, P–A}]_{k-1} \quad (4)$$
$$[\text{P–A, P–A}]_k = 3\,[\text{P–A, P–A}]_{k-1} + 6\,[\text{P–P, P–A}]_{k-1}^2 + 2\,[\text{P–A}]_{k-1}^2 + 4\,[\text{P–A}]_{k-1}\,[\text{P–P, P–A}]_{k-1} \quad (5)$$

Paths in RT'_k will be described using the port labels and RT'_{k-1}-copy labels in Figure 8. The following notation will be used when describing paths in RT'_k. A path from one port to another port in a copy G of RT'_{k-1} will be denoted [P–P in G]; this notation will always be immediately preceded and followed by a port of G. For example, part of a path could be $\ldots r, [\text{P–P in } D], s \ldots$. Similarly, a path from a port of G to an arbitrary node in G will be denoted [P–A in G], and part of a path could be $\ldots, s, [\text{P–A in } D], a$, where a is any node of D that is not already on the path. The path denoted by [P–P in G] or [P–A in G] does not necessarily H-cover G, because there could be another path or another part of the same path that also visits G. However, the collection of all paths that visit G must together H-cover G. For example, one path might be $\ldots, t, [\text{P–P in } D], s, \ldots$ and another might be $\ldots, r, [\text{P–A in } D], a$. The number of pairs of paths ([P–P in D], [P–A in D]) that H-cover D is $[\text{P–P, P–A}]_{k-1}$ by definition.

$[\text{P–P}]_k$: The number of paths p from t to w.

Every path p from t to w has the form $p = (t, [\text{P–P in } D], s, x, [\text{P–P in } F], y, u, [\text{P–P in } E], w)$. (Here and below we let the reader verify that we have listed only and all possibilities for p or (p_1, p_2).) Therefore,

$$[\text{P–P}]_k = [\text{P–P}]_{k-1}^3.$$

Because $[\text{P–P}]_1 = 1$, this implies that $[\text{P–P}]_k = 1$ for all k.

$[\text{P–A}]_k$: The number of paths p from t to a (an arbitrary node) with $a \neq t$.

There are three cases: $a \in D$, $a \in E$, and $a \in F$. Because $t \in D$, the last two cases are symmetric, so we consider only $a \in D$ and $a \in E$.

1. $a \in D$.

$$p = (t, [\text{P–P in } D], r, v, [\text{P–P in } E], u, y, [\text{P–P in } F], x, s, [\text{P–A in } D], a), \text{ or}$$
$$p = (t, [\text{P–P in } D], s, x, [\text{P–P in } F], y, u, [\text{P–P in } E], v, r, [\text{P–A in } D], a).$$

The number of such paths p is $2\,[\text{P–P}]_{k-1}^2\,[\text{P–P, P–A}]_{k-1} = 2\,[\text{P–P, P–A}]_{k-1}$.

2. $a \in E$

$$p = (t, [\text{P–P in } D], s, x, [\text{P–P in } F], y, u, [\text{P–A in } E], a).$$

The number of such p is $[\text{P–P}]_{k-1}^2\,[\text{P–A}]_{k-1} = [\text{P–A}]_{k-1}$, and this should be doubled to $2\,[\text{P–A}]_{k-1}$ to cover also the case $a \in F$.

Therefore,

$$[\text{P–A}]_k = 2([\text{P–A}]_{k-1} + [\text{P–P, P–A}]_{k-1}).$$

$[\text{P–P, P–A}]_k$: The number of pairs (p_1, p_2) where p_1 is from t to w, and p_2 is from z to a where $a \neq t$ and $a \neq w$.

There are three cases: $a \in D$, $a \in E$, or $a \in F$. Because $t \in D$, $w \in E$, and $z \in F$, the first two cases are symmetric. So we consider only $a \in D$ and $a \in F$.

1. $a \in D$.

$$p_1 = (t, [\text{P–P in } D], r, v, [\text{P–P in } E], w) \text{ and } p_2 = (z, [\text{P–P in } F], x, s, [\text{P–A in } D], a).$$

   The number of such (p_1, p_2) is $[\text{P–P}]_{k-1}^2 \, [\text{P–P, P–A}]_{k-1} = [\text{P–P, P–A}]_{k-1}$, and this should be doubled to $2\,[\text{P–P, P–A}]_{k-1}$ to cover also the case $a \in E$.

2. $a \in F$.

$$p_1 = (t, [\text{P–P in } D], r, v, [\text{P–P in } E], w) \text{ and } p_2 = (z, [\text{P–A in } F], a), \text{ or}$$
$$p_1 = (t, [\text{P–P in } D], s, x, [\text{P–P in } F], y, u, [\text{P–P in } E], w) \text{ and } p_2 = (z, [\text{P–A in } F], a).$$

   The number of such (p_1, p_2) is

$$[\text{P–P}]_{k-1}^2 \, [\text{P–A}]_{k-1} + [\text{P–P}]_{k-1}^2 \, [\text{P–P, P–A}]_{k-1} = [\text{P–A}]_{k-1} + [\text{P–P, P–A}]_{k-1}.$$

Therefore,

$$[\text{P–P, P–A}]_k = [\text{P–A}]_{k-1} + 3\,[\text{P–P, P–A}]_{k-1}.$$

$[\text{P–A, P–A}]_k$: The number of pairs (p_1, p_2) where p_1 is from t to a_1, and p_2 is from w to a_2, where $a_1 \neq a_2$, $a_1 \neq w$ and $a_2 \neq t$.

There are $3^2 = 9$ cases depending on which of D, E or F contains a_1 and which contains a_2. Symmetries reduce this to 6 cases.

1. $a_1, a_2 \in F$.

$$p_1 = (t, [\text{P–P in } D], s, x, [\text{P–A in } F], a_1) \text{ and } p_2 = (w, [\text{P–P in } E] u, y, [\text{P–A in } F], a_2).$$

   The number of (p_1, p_2) is $[\text{P–P}]_{k-1}^2 \, [\text{P–A, P–A}]_{k-1} = [\text{P–A, P–A}]_{k-1}$.

2. $a_1, a_2 \in D$ (and symmetrically, $a_1, a_2 \in E$).

$$p_1 = (t, [\text{P–A in } D], a_1) \text{ and } p_2 = (w, [\text{P–P in } E], u, y, [\text{P–P in } F], x, s, [\text{P–A in } D], a_2).$$

   The number of (p_1, p_2) is $[\text{P–P}]_{k-1}^2 \, [\text{P–A, P–A}]_{k-1} = [\text{P–A, P–A}]_{k-1}$, and this should be doubled to $2\,[\text{P–A, P–A}]_{k-1}$ for the symmetric case.

3. $a_1 \in D$ and $a_2 \in E$.

$$\begin{aligned}
&p_1 = (t, [\text{P–A in } D], a_1) \text{ and}\\
&\quad p_2 = (w, [\text{P–P in } E], u, y, [\text{P–P in } F], x, s, [\text{P–P in } D], r, v, [\text{P–A in } E], a_2), \text{ or}\\
&p_1 = (t, [\text{P–A in } D], a_1) \text{ and}\\
&\quad p_2 = (w, [\text{P–P in } E], v, r, [\text{P–P in } D], s, x, [\text{P–P in } F], y, u, [\text{P–A in } E], a_2), \text{ or}\\
&p_1 = (t, [\text{P–P in } D], s, x, [\text{P–P in } F], y, u, [\text{P–P in } E], v, r, [\text{P–A in } D], a_1) \text{ and}\\
&\quad p_2 = (w, [\text{P–A in } E], a_2), \text{ or}\\
&p_1 = (t, [\text{P–P in } D], r, v, [\text{P–P in } E], u, y, [\text{P–P in } F], x, s, [\text{P–A in } D], a_1) \text{ and}\\
&\quad p_2 = (w, [\text{P–A in } E], a_2).
\end{aligned}$$

The number of (p_1, p_2) is $4\,[\text{P–P}]_{k-1}\,[\text{P–P, P–A}]_{k-1}^2 = 4\,[\text{P–P, P–A}]_{k-1}^2$.

4. $a_1 \in E$ and $a_2 \in D$.

$$\begin{aligned}
&p_1 = (t, [\text{P–P in } D], r, v, [\text{P–A in } E], a_1) \text{ and}\\
&\quad p_2 = (w, [\text{P–P in } E], u, y, [\text{P–P in } F], x, s, [\text{P–A in } D], a_2), \text{ or}\\
&p_1 = (t, [\text{P–P in } D], s, x, [\text{P–P in } F], y, u, [\text{P–A in } E], a_1) \text{ and}\\
&\quad p_2 = (w, [\text{P–P in } E], v, r, [\text{P–A in } D], a_2).
\end{aligned}$$

The number of (p_1, p_2) is $2\,[\text{P–P}]_{k-1}\,[\text{P–P, P–A}]_{k-1}^2 = 2\,[\text{P–P, P–A}]_{k-1}^2$.

5. $a_1 \in F$ and $a_2 \in D$ (and symmetically, $a_1 \in E$ and $a_2 \in F$).

$$p_1 = (t, [\text{P–P in } D], s, x, [\text{P–A in } F], a_1) \text{ and } p_2 = (w, [\text{P–P in } E], v, r, [\text{P–A in } D], a_2).$$

The number of (p_1, p_2) is $[\text{P–P}]_{k-1}\,[\text{P–A}]_{k-1}\,[\text{P–P, P–A}]_{k-1} = [\text{P–A}]_{k-1}\,[\text{P–P, P–A}]_{k-1}$, and this should be doubled to $2\,[\text{P–A}]_{k-1}\,[\text{P–P, P–A}]_{k-1}$ for the symmetric case.

6. $a_1 \in F$ and $a_2 \in E$ (and symmetrically, $a_1 \in D$ and $a_2 \in F$).

$$\begin{aligned}
&p_1 = (t, [\text{P–P in } D], s, x, [\text{P–A in } F], a_1) \text{ and } p_2 = (w, [\text{P–A in } E], a_2), \text{ or}\\
&p_1 = (t, [\text{P–P in } D], r, v, [\text{P–P in } E], u, y, [\text{P–A in } F], a_1) \text{ and } p_2 = (w, [\text{P–A in } E], a_2).
\end{aligned}$$

The number of (p_1, p_2) is

$$[\text{P–P}]_{k-1}\,[\text{P–A}]_{k-1}^2 + [\text{P–P}]_{k-1}\,[\text{P–A}]_{k-1}\,[\text{P–P, P–A}]_{k-1} = [\text{P–A}]_{k-1}^2 + [\text{P–A}]_{k-1}\,[\text{P–P, P–A}]_{k-1},$$

and this should be doubled to $2([\text{P–A}]_{k-1}^2 + [\text{P–A}]_{k-1}\,[\text{P–P, P–A}]_{k-1})$ for the symmetric case.

Totaling all 6 cases gives

$$[\text{P–A, P–A}]_k = 3\,[\text{P–A, P–A}]_{k-1} + 6\,[\text{P–P, P–A}]_{k-1}^2 + 2\,[\text{P–A}]_{k-1}^2 + 4\,[\text{P–A}]_{k-1}\,[\text{P–P, P–A}]_{k-1}.$$

Now $Ham(\mathrm{RT}_k)$ can be computed from the expression (1) for $Ham(\mathrm{RT}_k)$, the equations (2) for $k = 1$, and the recurrence equations (3), (4), (5). For example, we get $Ham(\mathrm{RT}_2) = 204$ and $Ham(\mathrm{RT}_3) = 2688$, which agrees with the results of computer enumeration (LaFall and Pohl 2004).

Although exact solution of recurrence equations is sometimes difficult, these equations are tractable. Subtracting (4) from (3) gives $[\text{P–A}]_k - [\text{P–P, P–A}]_k = [\text{P–A}]_{k-1} - [\text{P–P, P–A}]_{k-1}$ for all $k \geq 2$. Because $[\text{P–A}]_1 - [\text{P–P, P–A}]_1 = 2 - 1 = 1$, this implies that $[\text{P–A}]_k - [\text{P–P, P–A}]_k = 1$ for all $k \geq 1$. Substituting $[\text{P–A}]_{k-1} - 1$ for $[\text{P–P, P–A}]_{k-1}$ in (3) gives the recurrence $[\text{P–A}]_k = 4\,[\text{P–A}]_{k-1} - 2$. This solves as $[\text{P–A}]_k = (4^k + 2)/3$. The values obtained for $[\text{P–A}]_{k-1}$ and $[\text{P–P, P–A}]_{k-1}$ can be substituted into (5), giving $[\text{P–A, P–A}]_k = 3\,[\text{P–A, P–A}]_{k-1} + (4^{2k-1} + 2)/3$. This recurrence for $[\text{P–A, P–A}]_k$ solves as

$$[\text{P–A, P–A}]_k = \tfrac{4}{39}\,16^k + \tfrac{9}{13}\,3^{k-1} - \tfrac{1}{3}.$$

Substituting the values for $[\text{P–A}]_k$ and $[\text{P–A, P–A}]_k$ into (1) proves Theorem 1.

9 Counting Hamiltonian Paths in BT_k

The *binary tree graphs*, BT_k for $k \geq 1$ are defined in terms of the related graphs BT'_k. The graph BT'_k is a complete binary tree of depth k with the leaves connected from left to right in a line. The root is called the *root port*, and the two leaves of degree 2 are called the *leaf ports*. The two leaf ports are indistinguishable, although the root port is distiguishable from a leaf port if $k \geq 2$ (BT'_1 is K_3). Now BT_1 is K_4. The graph BT_k for $k \geq 2$ is formed from three copies of BT'_{k-1}, one additional center node, and six additional edges as shown in Figure 10. An edge connects the center to the root port of each copy of BT'_{k-1}. The remaining three new edges each connect a leaf port of a copy to a leaf port of another copy so that each leaf port is connected to exactly one other.

To count the Hamiltonian paths in BT_k, it will be useful to have a recursive definition of the graphs BT'_k. This is shown in Figure 11 where BT'_k is constructed from two copies of BT'_{k-1} labeled D and E, a new node q, and three new edges. Two of the new edges connect the root port of D and E to q. The other new edge connects a leaf port of D to a leaf port of E. The leaf ports of D and E that are not connected become the leaf ports of BT'_k, and q is the root port of BT'_k.

The method used to count Hamiltonian paths of BT_k is similar to the method used for RT_k. A complication is that the BT'_k graphs have two types of distinguishable ports, root ports and leaf ports. Thus in the quantities $[\text{P–P, P–A}]_k$, etc., we can replace "P" (port) by either "R" (for root port) or "L" (for leaf port) subject to the restriction that a quantity can have at most one "R" and at most two "L"s. As before, "A" stands for "any". This gives eight quantities: $[\text{R–L}]_k$, $[\text{L–L}]_k$, $[\text{R–A}]_k$, $[\text{L–A}]_k$, $[\text{R–L, L–A}]_k$, $[\text{L–L, R–A}]_k$, $[\text{R–A, L–A}]_k$, and $[\text{L–A, L–A}]_k$. For example, $[\text{R–L, L–A}]_k$ is the number of H-covering pairs (p_1, p_2) of paths such that p_1 is a path from the root port of BT'_k to a leaf port, and p_2 is a path from the other leaf port to an arbitrary node.

Because BT'_1 is K_3, it can be seen directly that

$$\begin{aligned} &[\text{R–L}]_1 = [\text{L–L}]_1 = [\text{R–L, L–A}]_1 = [\text{L–L, R–A}]_1 = 1 \\ &[\text{R–A}]_1 = [\text{L–A}]_1 = [\text{R–A, L–A}]_1 = [\text{L–A, L–A}]_1 = 2. \end{aligned} \tag{6}$$

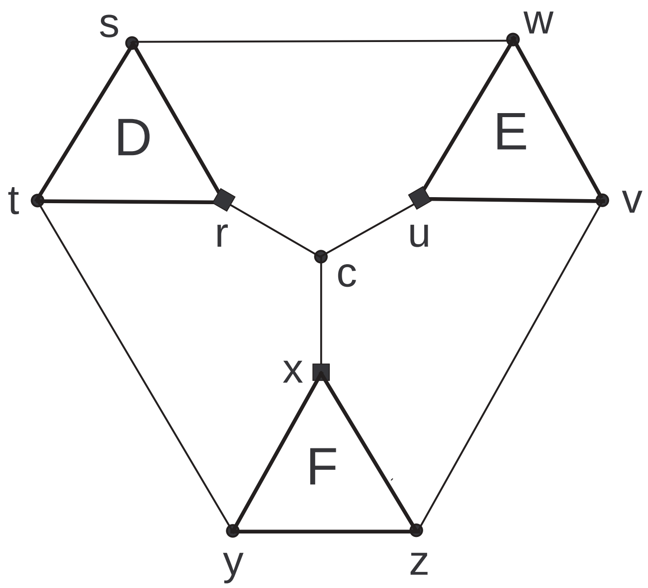


Figure 10: The graph BT_k consisting of three copies of BT'_{k-1} labeled D, E, F, a center node c, and six additional edges. The root ports of D, E, and F are drawn as squares.

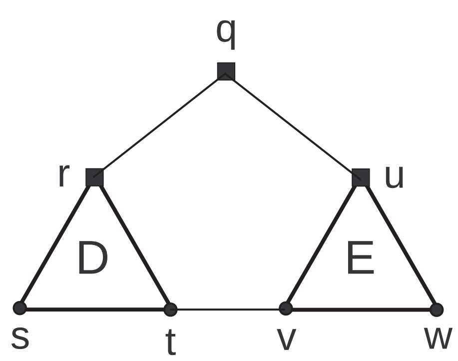


Figure 11: The recursive construction of BT'_k. Shown are two copies of BT'_{k-1}: one labeled D having root port r and leaf ports s, t, and one labeled E having root port u and leaf ports v, w. Three new edges interconnect D, E, and a new node q as shown. The root port of BT'_k is q, and the leaf ports are s and w. Root ports are drawn as squares.

We first note that $[\text{R–L}]_k = [\text{L–L}]_k = 1$ for all k. This can be seen from Figure 11. The only H-covering paths from the leaf port s to the leaf port w have the form $s, [\text{R–L in } D], r, q, u, [\text{R–L in } E], w$, so $[\text{L–L}]_k = [\text{R–L}]_{k-1}^2$. The only H-covering paths from the root port q to the leaf port s have the form $q, u, [\text{R–L in } E], v, t, [\text{L–L in } D], s$, so $[\text{R–L}]_k = [\text{R–L}]_{k-1}\,[\text{L–L}]_{k-1}$. It then follows by induction on k that $[\text{L–L}]_k = [\text{R–L}]_k = 1$ for all k. This will be used to simplify certain expressions below.

We now express $Ham(\text{BT}_k)$ in terms of these quantities, referring to Figure 10. We count the number of unoriented Hamiltonian paths in BT_k and then double to obtain the number of oriented ones. Let a_1 and a_2 be the two ends of an unoriented Hamiltonian path in BT_k. We consider three cases.

1. a_1 and a_2 belong to the same copy of BT'_{k-1}.

   Say that $a_1, a_2 \in D$, the other two cases being symmetric. The possible Hamiltonian paths are

$$a_1, [\text{R–A in } D], r, c, x, [\text{R–L in } F], z, v, [\text{L–L in } E], w, s, [\text{L–A in } D], a_2$$
$$a_1, [\text{R–A in } D], r, c, u, [\text{R–L in } E], v, z, [\text{L–L in } F], y, t, [\text{L–A in } D], a_2$$
$$a_1, [\text{L–A in } D], t, y, [\text{R–L in } F], x, c, u, [\text{R–L in } E], w, s, [\text{L–A in } D], a_2.$$

   The number of such paths, tripled to cover also the two symmetric cases, is

$$3(2[\text{R–A, L–A}]_{k-1} + [\text{L–A, L–A}]_{k-1}). \tag{7}$$

2. a_1 and a_2 belong to different copies of BT'_{k-1}.

   Say that $a_1 \in D$ and $a_2 \in E$, the other two cases being symmetric. The path starting at a_1 must exit D through either a root or a leaf port, and the path starting at a_2 must exit E through either a root or a leaf port. Therefore this case splits into four subcases. It is obvious that both paths cannot exit through a root port. If both exit through a leaf port then the possible Hamiltonian paths are

$$a_1, [\text{L–A in } D], t, y, [\text{R–L in } F], x, c, r, [\text{R–L in } D], s, w, [\text{L–A in } E], a_2$$
$$a_1, [\text{L–A in } D], t, y, [\text{L–L in } F], z, v, [\text{R–L in } E], u, c, r, [\text{R–L in } D], s, w, [\text{L–A in } E], a_2$$
$$a_1, [\text{L–A in } D], s, w, [\text{R–L in } E], u, c, x, [\text{R–L in } F], z, v, [\text{L–A in } E], a_2$$
$$a_1, [\text{L–A in } D], s, w, [\text{R–L in } E], u, c, r, [\text{R–L in } D], t, y, [\text{L–L in } F], z, v, [\text{L–A in } E], a_2.$$

   If one exits through a leaf port and the other exits through a root port, there are two symmetric cases. Say that the path from a_1 exits through a leaf port of D and the path from a_2 exits through the root port u of E. The possible Hamiltonian paths are

$$a_1, [\text{L–A in } D], s, w, [\text{L–L in } E], v, z, [\text{R–L in } F], x, c, u, [\text{R–A in } E], a_2$$
$$a_1, [\text{L–A in } D], s, w, [\text{L–L in } E], v, z, [\text{L–L in } F], y, t, [\text{R–L in } D], r, c, u, [\text{R–A in } E], a_2$$
$$a_1, [\text{L–A in } D], t, y, [\text{R–L in } F], x, c, u, [\text{R–A in } E], a_2$$
$$a_1, [\text{L–A in } D], t, y, [\text{L–L in } F], z, v, [\text{L–L in } E], w, s, [\text{R–L in } D], r, c, u, [\text{R–A in } E], a_2.$$

The number of Hamiltonian paths in case 2 is

$$6([\text{L–A}]_{k-1}([\text{R–A}]_{k-1} + [\text{L–L, R–A}]_{k-1}) \\ + [\text{R–L, L–A}]_{k-1}([\text{R–L, L–A}]_{k-1} + 2[\text{L–L, R–A}]_{k-1} + [\text{L–A}]_{k-1})). \tag{8}$$

3. $a_1 = c$ and a_2 belongs to a copy of BT'_{k-1}.

   Say that $a_2 \in D$, the other two cases being symmetric. The possible Hamiltonian paths are

$$c, r, [\text{R–L in } D], s, w, [\text{L–L in } E], v, z, [\text{L–L in } F], y, t, [\text{L–A in } D], a_2$$
$$c, r, [\text{R–L in } D], t, y, [\text{L–L in } F], z, v, [\text{L–L in } E], w, s, [\text{L–A in } D], a_2$$
$$c, u, [\text{R–L in } E], v, z, [\text{L–L in } F], y, t, [\text{L–A in } D], a_2$$
$$c, x, [\text{R–L in } F], z, v, [\text{L–L in } E], w, s, [\text{L–A in } D], a_2.$$

The number of Hamiltonian paths in case 3 is

$$6([\text{R–L, L–A}]_{k-1} + [\text{L–A}]_{k-1}). \tag{9}$$

Doubling the total of cases 1, 2, and 3 to count oriented Hamiltonian paths,

$$\mathit{Ham}(\text{BT}_k) = 2 \cdot (\,(7) + (8) + (9)\,). \tag{10}$$

The following recurrence equations for the quantities in (7), (8), and (9) are derived below: for all $k \geq 2$

$$[\text{R–A}]_k = 2\,[\text{L–A}]_{k-1} \tag{11}$$
$$[\text{L–A}]_k = [\text{R–L, L–A}]_{k-1} + [\text{R–A}]_{k-1} + [\text{L–L, R–A}]_{k-1} + 1 \tag{12}$$
$$[\text{R–L, L–A}]_k = 2\,[\text{R–L, L–A}]_{k-1} + [\text{L–A}]_{k-1} \tag{13}$$
$$[\text{L–L, R–A}]_k = 2\,[\text{L–L, R–A}]_{k-1} + 1 \tag{14}$$
$$[\text{R–A, L–A}]_k = [\text{R–A, L–A}]_{k-1} + [\text{L–A, L–A}]_{k-1} \\ + [\text{L–A}]_{k-1}([\text{R–L, L–A}]_{k-1} + [\text{R–A}]_{k-1} + [\text{L–L, R–A}]_{k-1} + 1) \tag{15}$$
$$[\text{L–A, L–A}]_k = 2([\text{R–A, L–A}]_{k-1} + [\text{L–A}]_{k-1} \\ + [\text{R–L, L–A}]_{k-1}([\text{R–L, L–A}]_{k-1} + 2\,[\text{L–L, R–A}]_{k-1} + 2)). \tag{16}$$

From equations (6) and (10)–(16) we calculate $\mathit{Ham}(\text{BT}_2) = 204$, $\mathit{Ham}(\text{BT}_3) = 1524$, and $\mathit{Ham}(\text{BT}_4) = 10740$, which agrees with the results of computer enumeration (LaFall and Pohl 2004).

We now prove the equations (11)–(16), referring to Figure 11. Recall that $[\text{R–L}]_k = [\text{L–L}]_k = 1$ for all k.

$[\text{R–A}]_k$: The number of paths p from q to a where, $a \neq q$.

There are two symmetric cases, $a \in D$ and $a \in E$. Say that $a \in D$. The paths are:

$$p = (q, u, [\text{R–L in } E], v, t, [\text{L–A in } D], a).$$

The number, doubled for the symmetric case, is $2\,[\text{L–A}]_{k-1}$.

$[\text{L–A}]_k$: The number of paths p from s to a, where $a \neq s$.

1. $a \in D$: The paths are

$$\begin{aligned} p &= (s, [\text{R–L in } D], r, q, u, [\text{R–L in } E], v, t, [\text{L–A in } D], a), \text{ or} \\ p &= (s, [\text{L–L in } D], t, v, [\text{R–L in } E], u, q, r, [\text{R–A in } D], a) \end{aligned}$$

   The number is $[\text{R–L, L–A}]_{k-1} + [\text{L–L, R–A}]_{k-1}$.

2. $a \in E$: The paths are $p = (s, [\text{R–L in } D], r, q, u, [\text{R–A in } E], a)$ and the number is $[\text{R–A}]_{k-1}$.

3. $a = q$: The paths are $p = (s, [\text{L–L in } D], t, v, [\text{R–L in } E], q)$ and there is one such path.

Summing the numbers in cases 1, 2, and 3 gives (12).

$[\text{R–L, L–A}]_k$: The number of pairs (p_1, p_2) where p_1 is from q to w and p_2 is from s to a, where a is different than q, s, and w.

1. $a \in D$: The paths are

$$\begin{gathered} p_1 = (q, u, [\text{R–L in } E], w) \text{ and } p_2 = (s, [\text{L–A in } D], a), \text{ or} \\ p_1 = (q, r, [\text{R–L in } D], t, v, [\text{L–L in } E], w) \text{ and } p_2 = (s, [\text{L–A in } D], a). \end{gathered}$$

   The number is $[\text{L–A}]_{k-1} + [\text{R–L, L–A}]_{k-1}$.

2. $a \in E$: The paths are $p_1 = (q, u, [\text{R–L in } E], w)$ and $p_2 = (s, [\text{L–L in } D], t, v, [\text{R–A in } E], a)$. The number is $[\text{R–L, L–A}]_{k-1}$.

$[\text{L–L, R–A}]_k$: The number of pairs (p_1, p_2) where p_1 is from s to w and p_2 is from q to a, where $a \neq s$ and $a \neq w$:

1. $a \in D$ (and symmetrically, $a \in E$): The paths are

$$p_1 = (s, [\text{L–L in } D], t, v, [\text{L–L in } E], w) \text{ and } p_2 = (q, r, [\text{R–A in } D], a).$$

   The number, doubled for the symmetric case, is $2\,[\text{L–L, R–A}]_{k-1}$.

2. $a = q$: The paths are $p_1 = (s, [\text{L–L in } D], t, v, [\text{L–L in } E], w)$ and $p_2 = (a)$, and there is one such path.

$[\text{R–A, L–A}]_k$: The number of pairs (p_1, p_2) where p_1 is from q to a_1 and p_2 is from s to a_2, where $a_1 \neq s$, $a_2 \neq q$, and $a_1 \neq a_2$.

1. $a_1, a_2 \in D$: Paths are $p_1 = (q, u, [\text{R–L in } E], v, t, [\text{L–A in } D], a_1)$ and $p_2 = (s, [\text{L–A in } D], a_2)$. The number is $[\text{L–A, L–A}]_{k-1}$.

2. $a_1, a_2 \in E$: Paths are $p_1 = (q, u, [\text{R–A in } E], a_1)$ and $p_2 = (s, [\text{L–L in } D], t, v, [\text{L–A in } E], a_2)$. The number is $[\text{R–A, L–A}]_{k-1}$.

3. $a_1 \in D$ and $a_2 \in E$: Paths are

$$p_1 = (q, r, [\text{R–A in } D], a_1) \text{ and } p_2 = (s, [\text{L–L in } D], t, v, [\text{L–A in } E], a_2).$$

   The number is $[\text{L–L, R–A}]_{k-1}[\text{L–A}]_{k-1}$.

4. $a_1 \in E$ and $a_2 \in D$: Paths are

$$p_1 = (q, u, [\text{R–A in } E], a_1) \text{ and } p_2 = (s, [\text{L–A in } D], a_2), \text{ or}$$
$$p_1 = (q, r, [\text{R–L in } D], t, v, [\text{L–A in } E], a_1) \text{ and } p_2 = (s, [\text{L–A in } D], a_2).$$

   The number is $[\text{R–A}]_{k-1}[\text{L–A}]_{k-1} + [\text{R–L, L–A}]_{k-1}[\text{L–A}]_{k-1}$.

5. $a_1 = q$ and $a_2 \in D$: There are no paths. The path from q to a_1 must be the single node q. Any path from s to a_2 must leave the component D to cover E and then return to D without going through q. This is impossible.

6. $a_1 = q$ and $a_2 \in E$: Paths are $p_1 = (q)$ and $p_2 = (s, [\text{L–L in } D], t, v, [\text{L–A in } E], a_2)$.
   The number is $[\text{L–A}]_{k-1}$.

Summing the numbers from the six cases gives (15).

$[\text{L–A, L–A}]_k$: The number of pairs (p_1, p_2) where p_1 is from s to a_1 and p_2 is from w to a_2, where $a_1 \neq w$, $a_2 \neq s$ and $a_1 \neq a_2$.

1. $a_1, a_2 \in D$ (and symetrically, $a_1, a_2 \in E$): Paths are

$$p_1 = (s, [\text{L–A in } D], a_1) \text{ and } p_2 = (w, [\text{R–L in } E], u, q, r, [\text{R–A in } D], a_2).$$

   The number (doubled) is $2\,[\text{R–A, L–A}]_{k-1}$.

2. $a_1 \in D$ and $a_2 \in E$: There are two symmetric cases depending on which one of p_1 or p_2 goes through q. Say that p_1 does. Paths are

$$p_1 = (s, [\text{R–L in } D], r, q, u, [\text{R–L in } E], v, t, [\text{L–A in } D], a_1) \text{ and } p_2 = (w, [\text{L–A in } E], a_2).$$

   The number (doubled) is $2\,[\text{R–L, L–A}]_{k-1}^2$.

3. $a_1 \in E$ and $a_2 \in D$. Again there are two symmetric cases depending on which one of p_1 or p_2 goes through q. Say that p_1 does. Path are

$$p_1 = (s, [\text{R–L in } D], r, q, u, [\text{R–A in } E], a_1) \text{ and } p_2 = (w, [\text{L–L in } E], v, t, [\text{L–A in } D], a_2).$$

   The number (doubled) is $2\,[\text{L–L, R–A}]_{k-1}[\text{R–L, L–A}]_{k-1}$.

4. $a_1 = q$:

(a) $a_2 \in D$: Paths are $p_1 = (s, [\text{R–L in } D], r, q)$ and $p_2 = (w, [\text{L–L in } E], v, t, [\text{L–A in } D], a_2)$. The number is $[\text{R–L, L–A}]_{k-1}$.

(b) $a_2 \in E$: Paths are

$$p_1 = (s, [\text{R–L in } D], r, q) \text{ and } p_2 = (w, [\text{L–A in } E], a_2), \text{ or}$$
$$p_1 = (s, [\text{L–L in } D], t, v, [\text{R–L in } E], u, q) \text{ and } p_2 = (w, [\text{L–A in } E], a_2).$$

The number is $[\text{R–L, L–A}]_{k-1} + [\text{L–A}]_{k-1}$.

The total for case 4 is $2\,[\text{R–L, L–A}]_{k-1} + [\text{L–A}]_{k-1}$.

5. $a_2 = q$: This case is symmetric to case 4, so the number is $2\,[\text{R–L, L–A}]_{k-1} + [\text{L–A}]_{k-1}$.

Summing the numbers for the five cases gives (16).

We have not found a closed form solution to the recurrence equations (11)–(16). However we can use the equations to prove bounds that determine $Ham(\text{BT}_k)$ to within a constant factor.

Proof of Theorem 2.

We will use the following lemma which determines $[\text{L–L, R–A}]_k$ exactly and determines $[\text{L–A}]_k$ to within a factor of 2. Recall that α satisfies (i) $\alpha^3 - 2\alpha^2 - 3\alpha + 4 = 0$, (ii) $\alpha \geq 2.5$, and (iii) $\alpha \leq 2.6$. For use in the proof, these imply

$$\text{(a)}\ \ 3 + \frac{2}{\alpha - 2} = \alpha^2 \qquad \text{(b)}\ \ 6 - \frac{\alpha}{\alpha - 2} \geq 0 \qquad \text{(c)}\ \ 3 - \frac{\alpha}{\alpha - 2} \leq 0 \tag{17}$$

where (a) follows from (i) and $\alpha \neq 2$, (b) follows from (ii), and (c) follows from (ii) and (iii).

Lemma 1 *For all $k \geq 1$,*

$$[\text{L–L, R–A}]_k = 2^k - 1 \quad \textit{and} \quad \alpha^k/2 \leq [\text{L–A}]_k \leq \alpha^k.$$

Proof. It is straightforward to check that $[\text{L–L, R–A}]_k = 2^k - 1$ satisfies the initial condition (6) and the recurrence (14).

To prove the bounds on $[\text{L–A}]_k$, first note that they hold for $k = 1, 2$: $2.6/2 < [\text{L–A}]_1 = 2 < 2.5$, and $2.6^2/2 < [\text{L–A}]_2 = 5 < 2.5^2$. To prove the bounds for $k \geq 3$ by induction on k note that (6) and (13) imply for all j that

$$[\text{R–L, L–A}]_j = 2^{j-1} + 2^{j-1} \sum_{i=1}^{j-1} 2^{-i}\,[\text{L–A}]_i. \tag{18}$$

Make the following substitutions into (12): substitute the expression (18) with $j = k - 1$ for $[\text{R–L, L–A}]_{k-1}$; from (11), substitute $2\,[\text{L–A}]_{k-2}$ for $[\text{R–A}]_{k-1}$; from the first part of the lemma, substitute 2^{k-1} for $[\text{L–L, R–A}]_{k-1} + 1$. This gives:

$$\begin{aligned} [\text{L–A}]_k &= 2^{k-1} + 2^{k-2} + 2\,[\text{L–A}]_{k-2} + 2^{k-2} \sum_{i=1}^{k-2} 2^{-i}\,[\text{L–A}]_i \\ &= 2^{k-1} + 2^{k-2} + 3\,[\text{L–A}]_{k-2} + 2^{k-2} \sum_{i=1}^{k-3} 2^{-i}\,[\text{L–A}]_i. \end{aligned} \tag{19}$$

Note that (19) is a recurrence involving only [L–A]. Using this we prove bounds on $[\text{L–A}]_k$ by induction on k. First we prove $\alpha^k/2 \le [\text{L–A}]_k$. Assume that $[\text{L–A}]_j \ge \alpha^j/2$ for $1 \le j < k$. By (19) and the induction hypothesis,

$$\begin{aligned}
[\text{L–A}]_k &\ge 2^{k-1} + 2^{k-2} + \frac{3\alpha^{k-2}}{2} + 2^{k-3}\sum_{i=1}^{k-3}\left(\frac{\alpha}{2}\right)^i \\
&= 2^{k-1} + 2^{k-2} + \frac{3\alpha^{k-2}}{2} + 2^{k-3}\frac{2}{\alpha-2}\left(\left(\frac{\alpha}{2}\right)^{k-2} - \frac{\alpha}{2}\right) \\
&= \left(4 + 2 - \frac{\alpha}{\alpha-2}\right)2^{k-3} + \frac{1}{2}\left(3 + \frac{2}{\alpha-2}\right)\alpha^{k-2} \\
&\ge \frac{1}{2}\left(3 + \frac{2}{\alpha-2}\right)\alpha^{k-2} \quad \text{using (17)(b)} \\
&= \alpha^k/2 \quad \text{using (17)(a).}
\end{aligned}$$

The inductive proof of the upper bound $[\text{L–A}]_k \le \alpha^k$ is similar. Assume that $[\text{L–A}]_j \le \alpha^j$ for $1 \le j < k$.

$$\begin{aligned}
[\text{L–A}]_k &\le 2^{k-1} + 2^{k-2} + 3\alpha^{k-2} + 2^{k-2}\sum_{i=1}^{k-3}\left(\frac{\alpha}{2}\right)^i \\
&= 2^{k-1} + 2^{k-2} + 3\alpha^{k-2} + 2^{k-2}\frac{2}{\alpha-2}\left(\left(\frac{\alpha}{2}\right)^{k-2} - \frac{\alpha}{2}\right) \\
&= \left(2 + 1 - \frac{\alpha}{\alpha-2}\right)2^{k-2} + \left(3 + \frac{2}{\alpha-2}\right)\alpha^{k-2} \\
&\le \left(3 + \frac{2}{\alpha-2}\right)\alpha^{k-2} \quad \text{using (17)(c)} \\
&= \alpha^k \quad \text{using (17)(a).}
\end{aligned}$$

This completes the proof of the lemma. ∎

Below we will use the following Lemma.

Lemma 2 $[\text{R–L, L–A}]_k \ge [\text{L–A}]_k$ *for all* $k \ge 3$.

Proof. The proof is by induction on k. It can be checked directly that $[\text{R–L, L–A}]_3 = 13$, $[\text{L–A}]_3 = 12$, $[\text{R–L, L–A}]_4 = 38$, and $[\text{L–A}]_4 = 31$. For the induction step, fix some $k \ge 5$ and assume that $[\text{R–L, L–A}]_j \ge [\text{L–A}]_j$ for $3 \le j < k$. To prove the induction step, subtract (12) from (13) and set $[\text{L–L, R–A}]_{k-1} + 1 = 2^{k-1}$ and $[\text{R–A}]_{k-1} = 2\,[\text{L–A}]_{k-2}$ to obtain

$$\begin{aligned}
[\text{R–L, L–A}]_k - [\text{L–A}]_k &= [\text{R–L, L–A}]_{k-1} - [\text{L–A}]_{k-1} + (2\,[\text{L–A}]_{k-1} - 2^{k-1} - 2\,[\text{L–A}]_{k-2}) \\
&\ge [\text{R–L, L–A}]_{k-1} - [\text{L–A}]_{k-1} + (2\,[\text{L–A}]_{k-1} - 2^{k-1} - 2\,[\text{R–L, L–A}]_{k-2}) \\
&\ge [\text{R–L, L–A}]_{k-1} - [\text{L–A}]_{k-1}.
\end{aligned}$$

For the last inequality we use (12) that $[\text{L–A}]_{k-1} \ge [\text{R–L, L–A}]_{k-2} + 2^{k-2}$. This completes the proof of the lemma. ∎

We first prove the lower bound $Ham(\mathrm{BT}_k) \geq 1.46(\alpha^2)^k$. From (10),

$$Ham(\mathrm{BT}_k) \geq 12([\text{R–A, L–A}]_{k-1} + \tfrac{1}{2}[\text{L–A, L–A}]_{k-1} + [\text{L–A}]_{k-1}[\text{R–A}]_{k-1} + [\text{R–L, L–A}]_{k-1}([\text{R–L, L–A}]_{k-1} + [\text{L–A}]_{k-1}). \tag{20}$$

We now bound each term on the right hand side of (20) in terms of [L–A], assuming that $k \geq 5$ (so $k-2 \geq 3$): Using (12), the equation (15) can be written

$$[\text{R–A, L–A}]_k = [\text{R–A, L–A}]_{k-1} + [\text{L–A, L–A}]_{k-1} + [\text{L–A}]_{k-1}[\text{L–A}]_k, \text{ so}$$

$[\text{R–A, L–A}]_{k-1} \geq [\text{L–A}]_{k-1}[\text{L–A}]_{k-2}$;
$[\text{L–A, L–A}]_{k-1} \geq 2\,[\text{R–L, L–A}]_{k-2}^2 \geq 2\,[\text{L–A}]_{k-2}^2$ using (16) and Lemma 2;
$[\text{L–A}]_{k-1}[\text{R–A}]_{k-1} = 2\,[\text{L–A}]_{k-1}[\text{L–A}]_{k-2}$ using (11);
$[\text{R–L, L–A}]_{k-1}([\text{R–L, L–A}]_{k-1} + [\text{L–A}]_{k-1}) \geq 2\,[\text{L–A}]_{k-1}^2$ using Lemma 2.

Substituting all these bounds into (20),

$$\begin{aligned} Ham(\mathrm{BT}_k) &\geq 12(2\,[\text{L–A}]_{k-1}^2 + 3\,[\text{L–A}]_{k-1}[\text{L–A}]_{k-2} + [\text{L–A}]_{k-2}^2) \\ &\geq 12(\tfrac{1}{2}\alpha^{2k-2} + \tfrac{3}{4}\alpha^{2k-3} + \tfrac{1}{4}\alpha^{2k-4}) \text{ because } [\text{L–A}]_k \geq \alpha^k/2 \\ &\geq 1.46(\alpha^2)^k \text{ because } \alpha \leq 2.6. \end{aligned}$$

This proves the lower bound for all $k \geq 5$. Using the calculated values of $Ham(\mathrm{BT}_k)$ for $k = 2, 3, 4$, the bound holds for these k as well.

To prove the upper bound of Theorem 2 we use the following bounds.

Lemma 3 *For all $k \geq 1$:*

$$\begin{aligned} [\text{R–A}]_k &\leq 2\alpha^{k-1} \\ [\text{R–L, L–A}]_k &\leq 2\alpha^k \\ \max\{[\text{L–A, L–A}]_k, [\text{R–A, L–A}]_k\} &\leq 3\alpha^{2k} \end{aligned}$$

Proof. The first inequality is immediate from (6), (11), and Lemma 1.

By (18) and Lemma 1

$$\begin{aligned} [\text{R–L, L–A}]_k &\leq 2^{k-1} + 2^{k-1}\sum_{i=1}^{k-1}\left(\frac{\alpha}{2}\right)^i \\ &= 2^{k-1} + 2^{k-1}\frac{2}{\alpha-2}\left(\left(\frac{\alpha}{2}\right)^k - \frac{\alpha}{2}\right) \\ &= \left(1 - \frac{\alpha}{\alpha-2}\right)2^{k-1} + \left(\frac{1}{\alpha-2}\right)\alpha^k \\ &\leq 2\alpha^k \text{ because } \alpha \geq 2.5 \end{aligned}$$

For the final inequality we prove that $[\text{L–A, L–A}]_k \leq 3\alpha^{2k}$ and $[\text{R–A, L–A}]_k \leq 3\alpha^{2k}$ simultaneously by induction on k. These hold for $k = 1, 2, 3$ because $[\text{R–A, L–A}]_1 = [\text{L–A, L–A}]_1 = 28 < 3\alpha^2$ and $[\text{R–A, L–A}]_2 < [\text{L–A, L–A}]_2 = 18 < 3\alpha^4$ and $[\text{R–A, L–A}]_3 < [\text{L–A, L–A}]_3 = 134 < 3\alpha^6$. For $k \geq 4$ we give the induction step for $[\text{L–A, L–A}]_k$ by using the following inequalities:

$[\text{R–A, L–A}]_{k-1} \leq 3\alpha^{2k-2}$ by induction;
$[\text{L–A}]_{k-1} \leq \alpha^{k-1}$ by Lemma 1;
$[\text{R–L, L–A}]_{k-1} \leq 2\alpha^{k-1}$ by the second part of this lemma;
$2\,[\text{L–L, R–A}]_{k-1} + 2 = 2^k$ by Lemma 1;
$2\alpha^{k-1}2^k + \alpha^{k-1} \leq 0.9\alpha^{2k-1}$ for all $k \geq 4$, because $\alpha \geq 2.5$; and
$3/\alpha^2 + 4/\alpha^2 + 0.9/\alpha \leq 1.5$ because $\alpha \geq 2.5$.

Using these inequalities in (16):

$$\begin{aligned}
[\text{L–A, L–A}]_k &\leq 2(\,3\alpha^{2k-2} + \alpha^{k-1} + 2\alpha^{k-1}(2\alpha^{k-1} + 2^k)\,) \\
&\leq 2\left(\frac{3\alpha^{2k}}{\alpha^2} + \frac{4\alpha^{2k}}{\alpha^2} + \frac{0.9\alpha^{2k}}{\alpha}\right) \\
&\leq 3\alpha^{2k}.
\end{aligned} \tag{21}$$

Using (15) instead of (16), the inequality (21) holds also for $[\text{R–A, L–A}]_k$. ∎

Using the bounds of Lemmas 1 and 3 in (10) proves the upper bound in Theorem 2.

References

Berge, C. *The Theory of Graphs and Its Applications*, Wiley, New York, 1962, pp. 107–118.

Björklund A., and Husfeldt, T. Finding a path of superlogarithmic length, *SIAM J. Comput.* 32 (2003), pp. 1395–1402.

D. Eppstein, The traveling salesman problem for cubic graphs, *Proc. 8th Annual Workshop on Algorithms and Data Structures*, Ottawa, Canada, 2003, 307–318.

Gabow, H. N., and Nie, S. Finding a long directed cycle, *Proc. 15th Annual ACM-SIAM Symp. on Discrete Algorithms*, New Orleans, 2004, pp. 49–58.

Karger, D., Motwani, R., and Ramakumar, G. D. S. On approximating the longest path in a graph, *Algorithmica* 18 (1997), pp. 82–98.

Knuth D. E. Leaper graphs, *Mathematical Gazette* 78 (1994), pp. 274–297.

LaFall, J., and Pohl, I., UCSC tech report, 2004.

Pohl, I. A method for finding Hamilton paths and knight's tours, *CACM* 10 (1967), pp. 446–449.

Pohl, I. *C# by Dissection*, Addison-Wesley, 2002.

Saaty, T. L., and Kainen, P. C. *The Four-Color Problem: Assaults and Conquest*, Dover, New York, p. 112, 1986.

Tutte, W. T. On Hamiltonian circuits, *J. London Math. Soc.* 21 (1946), pp. 98–101.

Vishwanathan, S. An approximation algorithm for finding a long path in Hamiltonian graphs, *Proc. 11th Annual ACM-SIAM Symp. on Discrete Algorithms*, San Francisco, 2000, pp. 680–685.

Wikipedia-Sierpinski, `en.wikipedia.org/wiki/Sierpinski_triangle`.

Wikipedia-Tait, `en.wikipedia.org/wiki/Tait%27s_conjecture`. "Tait's conjecture states that 'Every polyhedron has a Hamiltonian cycle (along the edges) through all its vertices'. It was proposed in 1886 by P. G. Tait and disproved in 1946, when W. T. Tutte constructed a counterexample with 25 faces, 69 edges and 46 vertices. The conjecture could have been significant, because if true, it would have implied the four color theorem."