Journal of Graph Algorithms and Applications

http://www.cs.brown.edu/publications/jgaa/

vol. 2, no. 8, pp. 1–21 (1998)

Rectangle-visibility Layouts of Unions and Products of Trees

Alice M. Dean

Department of Mathematics
and Computer Science
Skidmore College
http://www.skidmore.edu/~adean
adean@skidmore.edu

Joan P. Hutchinson

Department of Mathematics
and Computer Science
Macalester College
http://www.macalester.edu/~hutchinson/
hutchinson@macalester.edu

Abstract

The paper considers representations of unions and products of trees as *rectangle-visibility graphs* (abbreviated *RVGs*), i.e., graphs whose vertices are rectangles in the plane, with adjacency determined by horizontal and vertical visibility. Our main results are that the union of any tree (or forest) with a depth-1 tree is an RVG, and that the union of two depth-2 trees and the union of a depth-3 tree with a matching are subgraphs of RVGs. We also show that the cartesian product of two forests is an RVG.

Communicated by J. S. B. Mitchell: submitted January 1998; revised September 1998.

1 Introduction

In this paper we study aspects of the question of how to represent a graph in the plane as a *rectangle-visibility graph* (*RVG* for short). In such a representation the vertices are drawn as rectangles with horizontal and vertical sides, and two vertices are adjacent if and only if their rectangles can be connected by a horizontal or vertical band of positive width that intersects no other rectangle. We call such a representation an *RVG layout* of the graph. Shermer [11] has shown that it is NP-complete to determine if a graph is an RVG, and so it is of interest to determine classes of graphs that are and are not RVGs. There is now a considerable body of research on RVGs; for results and applications see [5, 6, 3, 4, 2] and others.

The focus of this paper is those graphs whose edges can be partitioned into particular types of trees (or forests); we say then that the graph is the union of these trees. An RVG is seen to be the union of two planar graphs (i.e., has thickness at most two) by considering the vertical and horizontal edges in its layout, and so the union of two trees is potentially an RVG. In addition to being planar, a tree has a natural representation as a *bar-visibility graph* (or *BVG*), in which each vertex is represented by a horizontal bar, and two bars are adjacent if they can be connected by a vertical band of positive width that intersects no other bar; BVGs are well understood and can be recognized in polynomial time [12, 13]. The horizontal and vertical edges of an RVG decompose it into a union of two BVGs . This leads to the question of when a union of two BVGs is an RVG; our question is when the union of two trees is an RVG. The union of two trees has at most $2n-2$ edges, when the union has n vertices, and this edge-bound is well below the bound of $6n-20$ edges for general RVGs [5, 6]; however, in the same papers it is shown that for each $n \geq 9$ and $m \geq 35$ there is a thickness-2 graph with n vertices and m edges that is not an RVG. The union of two trees has clique number at most 4, and it is possible to lay out K_8 as an RVG; however, again it is not hard to construct graphs with clique number 4 and at most $2n-2$ edges that are not RVGs.

Since a planar graph is an RVG [13] and has arboricity at most 3 [9] (i.e., it is the union of three or fewer forests), one might hope that a union of two trees might be "nearly planar," and thus be an RVG. In [10] it is shown that a union of three trees can have thickness greater than 2, and more recently Bjorling-Sachs and Shermer [1] have constructed an example showing that the union of two trees needn't be an RVG. In [2] certain types of unions of trees, as well as certain classes of k-trees, are shown to be RVGs. Here we look at classes of tree-unions that are described primarily by the *depth* of one or both of the component trees, i.e., by the maximum distance from vertex to root in each tree. For instance, the example of [10] shows that the union of three depth-2 trees need not have thickness 2. Our main results are that the union of any tree (or forest) with a depth-1 tree is an RVG, and that the union of two depth-2 trees and the union of a depth-3 tree with a matching are subgraphs of RVGs. We also show that the cartesian product of two forests is an RVG.

2 Background and definitions

Throughout the paper we use terminology established in [12, 13] for bar-visibility graphs, and in [3, 4, 2] and others for rectangle-visibility graphs. We call a graph G a *bar-visibility graph* (*BVG* for short) if its vertices can be represented by closed horizontal line segments in the plane, nonintersecting except possibly at endpoints, in such a way that two vertices are adjacent if and only if there is a vertical *visibility band* joining the corresponding segments. By a visibility band we mean a nondegenerate rectangle, with opposite sides subsets of the two segments, and intersecting no other segments. Such a collection of segments is called a *BVG layout* of G.

BVGs are easily seen to be planar graphs and have been characterized as those planar graphs having a planar embedding with all cut-points lying on a common face [12, 13]. It is also easy to see that any tree (or forest) is a BVG; in a standard tree layout, choose a root for the tree and represent it by a horizontal line segment, and represent each remaining vertex as a segment lying below the segment representing its parent; for forests put BVG layouts for the component trees side by side.

A graph G is called a *rectangle-visibility graph* (*RVG* for short) if its vertices can be represented by closed rectangles in the plane that have horizontal and vertical sides and are pairwise disjoint except possibly along their boundaries, in such a way that two vertices are adjacent if and only if there is a *vertical or horizontal* band of visibility joining the two rectangles. BVGs are easily seen to be a subclass of RVGs, by fattening the horizontal segments of a BVG layout into rectangles, and staggering them vertically to avoid any horizontal visibilities; Wismath [13] has shown that every planar graph is an RVG. By partitioning the edges of an RVG according to horizontal and vertical visibilities, it is clear that every RVG has thickness at most 2; in other words, its edges can be partitioned into two sets, each of which induces a planar graph. More generally, the *thickness* of a graph G, denoted $\theta(G)$, is the least number of sets into which its edges can be partitioned so that each set induces a planar subgraph of G. By results of [3, 4, 5, 6] it is known that not every thickness-2 graph is an RVG.

An RVG (resp., BVG) layout is called *noncollinear* if no two rectangles have collinear sides (resp., no two segments have endpoints with the same x-coordinate). A graph having such a layout is called a *noncollinear RVG* (resp., noncollinear BVG), or NCRVG (resp., NCBVG) for short. In [8] a characterization of NCBVGs is given that shows them to be a strict subclass of BVGs; in [3, 4] it is shown that NCRVGs form a strict subclass of RVGs.

A graph G is called a *weak* RVG (resp., BVG) if it is a subgraph of an RVG (resp., BVG); in other words, G has a layout in which every edge corresponds to a visibility band, but there may also be visibility bands that correspond to edges not in G. Thus RVGs (resp., BVGs) comprise a subclass of weak RVGs (resp., weak BVGs). In [13, 12] it is shown that the class containment is strict for BVGs, and in [3, 4] it is shown to be strict for RVGs.

It is easy to see that the standard BVG layout described above for trees can be constructed in a noncollinear manner, and that such a layout can be modified to give an NCRVG layout for any tree or forest. In what follows we will make use of a special form of such a layout, which we call a "standard diagonal arrangement." In laying out a tree T, we represent each vertex of T as a rectangle (or square) contained in a rectangular area; for contrast we call these rectangular areas "boxes" and will indicate how and where to place the (vertex-) rectangles in a suitable box or subbox. If a vertex v is represented as a rectangle R, we typically represent its children as squares lying in a (square) box S below or to the right of R (although there will be some exceptions to this scheme). We choose a positive integer d so that T is a subtree of a d-ary tree, and we subdivide S into d^2 subboxes. We represent the children of v as squares, each a strict subsquare of one of the d boxes along the main diagonal. Thus we retain noncollinearity and v can see between any two of its children. Such an arrangement is called a "*standard diagonal arrangement*" (abbreviated *SDA*) of the children of v. Figure 1 shows an SDA layout of a depth-2 tree, in which depth-1 vertices lie below the root, and depth-2 vertices lie to the right of their depth-1 parents.

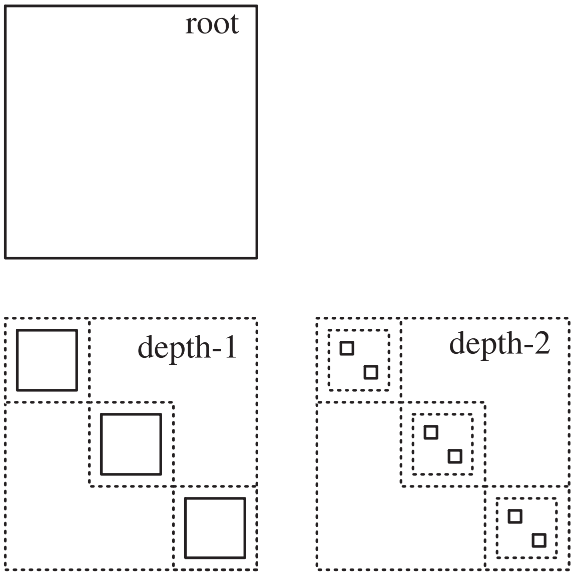

Figure 1: SDA layout of a depth-2 tree

We make several observations about the SDA. First, such an arrangement induces both a vertical and a horizontal ordering of the children of v. Second, if all descendants of the root are arranged in SDAs below or to the right of their parent, then all vertices in the layout retain both *east-* and *south-visibility*, meaning there are regions in which they see nothing to the east and nothing to the south. Third, note that if the children of a vertex v are in an SDA in a box S below the rectangle representing v, then a child can be moved vertically up or down in S without altering its visibility to v and without altering its horizontal ordering as a child of v (of course moving it may cause it to become horizontally visible to other vertices). An analogous observation holds if v sees its children in an SDA to its right. If the layout is such that each vertex sees its children in an SDA to its south (resp., east), then we call the layout a *standard vertical NCRVG tree layout* (resp., standard horizontal NCRVG tree layout). These standard layouts are extended to forests by laying out each component southeast of the previous one.

Throughout we use lower-case letters to refer to vertices and upper-case letters for the rectangles representing those vertices and for the surrounding boxes; for example, we use X to denote the rectangle representing the vertex x, placed within a box A. If A and B are disjoint boxes with the same dimensions, and S is a subbox of A, then by the "*box in B corresponding to S*" we mean the subbox T of B that has the same dimensions as S and the same relative position in B as S has in A. We sometimes use subscripts to indicate the depth of a vertex in a tree. For example, if T is a d-ary tree with root r, then we typically denote the children of r by x_i, and the children of x_i by $x_{i,j}$, where the indices i, j run from 1 to d.

The results in this paper are primarily concerned with graphs that are combinations of certain types of trees, particularly unions and products of trees. By the *union* of two graphs, $G_1 \cup G_2$, we mean the graph G with $V(G) = V(G_1) \cup V(G_2)$ and $E(G) = E(G_1) \cup E(G_2)$. The sets $V(G_1)$ and $V(G_2)$ (resp., $E(G_1)$ and $E(G_2)$) may or may not have elements in common. In contrast, we say that a graph G can be *decomposed* into the subgraphs G_1 and G_2 if $V(G) = V(G_1) = V(G_2)$ and $E(G)$ is the *disjoint* union of $E(G_1)$ and $E(G_2)$. Note that if G can be decomposed into G_1 and G_2 then $G = G_1 \cup G_2$, but the converse need not hold.

The *cartesian product* (or simply *product*) of the graphs G_1 and G_2, denoted $G_1 \times G_2$, is the graph with vertex set $V(G_1 \times G_2) = V(G_1) \times V(G_2)$ and with vertices (u_1, u_2) and (v_1, v_2) adjacent if either $u_1 = v_1$ and $(u_2, v_2) \in E(G_2)$ or $u_2 = v_2$ and $(u_1, v_1) \in E(G_1)$. By analogy with the product of a subset of the x-axis with a subset of the y-axis, a *horizontal* (resp., *vertical*) edge of $G_1 \times G_2$ is one that connects vertices with the same G_2-coordinate (resp., G_1-coordinate). The *n-dimensional hypercube*, denoted Q_n, is defined recursively as a cartesian product: $Q_1 = K_2$, the complete 2-vertex graph, and $Q_{n+1} = Q_n \times Q_1$, for $n \geq 1$.

A *caterpillar* is a tree containing a simple path P such that every vertex not on P is distance one from P. A *caterpillar forest* is a forest in which every component is a caterpillar. In [2] it is shown that if a graph G is a union of two caterpillar forests, then G is an NCRVG. The proof uses the fact that a BVG layout of a caterpillar forest can be projected onto an interval graph in which no more than two intervals have common intersection; the cartesian product of the two interval graphs gives an NCRVG layout of G.

Theorem 1 (The Caterpillar theorem [2]) *If G is the union of two caterpillar* *forests, then G is a noncollinear rectangle-visibility graph.*

3 Unions with depth-1 trees and forests

In this section we consider the union of two trees, one of which has depth 1. By the Caterpillar Theorem of [2], the union of two caterpillar forests is an NCRVG and thus so is the union of two depth-1 trees. Even more, the union of two forests, each forest with all components being depth-1 trees, is an NCRVG. We show now that not only does the union of any forest with a depth-1 tree have a weak RVG layout, but also the union of any forest with as many as three trees of depth 1 has a weak layout. In contrast we show that the union of five trees, four of which have depth 1, need not be weakly representable. The main result of this section is that any forest union a depth-1 tree is an NCRVG.

Proposition 1 *If a graph G is a union of a forest and one, two, or three depth-1 trees, then G is a weak RVG.*

Proof: Denote the forest by F, and denote the depth-1 trees by T_i, with roots r_i, $1 \leq i \leq 3$. We claim we may assume that F does not contain any of the r_i vertices, but does contain all other vertices of G: For each vertex x not in F and $x \neq r_i$ for any i, add x to F as an isolated vertex. Then, for each $i > 0$ such that $r_i \in F$, remove r_i and any incident edges from F, and add those edges to T_i. Then F and the trees T_i still have the properties given in the proposition.

Lay out F with the standard vertical NCRVG layout; in this layout each rectangle is east-, south-, and west-visible. Then place long rectangles R_i to represent $r_i, i > 0$, along the entire west, south, and east borders, respectively. Thus each R_i, for $i > 0$, sees every other rectangle in the layout, and hence all the vertices and edges of G are represented, giving a weak representation of G as claimed. □

Proposition 2 *A graph that is the union of a forest and four depth-1 trees need not be a weak RVG.*

Proof: In [DH1, DH2] it is shown that $K_{5,9}$, though of thickness-2, does not have a weak RVG representation. That graph is the union of five depth-1 trees. □

The layout of Prop. 1 can be improved to give a noncollinear layout when G is the union of a forest and a single depth-1 tree.

Theorem 2 *The union of a forest and a depth-one tree is an NCRVG.*

Proof: Suppose $G = F \cup T$, where F is a forest and T is a depth-1 tree. Let r be the root of one component of F and r_1 the root of T; as shown in the proof of Prop. 1 we may assume F contains all the vertices of G, except for r_1. Lay out F with the standard vertical NCRVG layout, with all vertices represented by squares and r represented at the top by a square R so that all squares in the representation lie below R or (for components not containing r) southeast of R. Next, make a duplicate, *horizontal* NCRVG layout of F, using the same square R to represent r, with all squares in the same component as r duplicated to the right of R, and with all other components of F represented by duplicate squares NE of R. Thus each vertex of G is currently represented twice, except for r, which is represented once, and r_1, which is not yet represented. In the end a rectangle R_1, representing r_1, will be placed along the left-hand (western) border of the whole array. The general plan is to place squares visible to R_1 below R and those invisible to R_1 to the right of other squares in F that will block them from R_1, as described below.

For each child x_i of r, choose either the representing square below R or to the right of R, according as x_i is or is not adjacent to r_1 in T. Then delete the unused square, to the right or below R, and delete the squares representing descendants of x_i to the right of or below R, respectively. That is, there are two potential positions for a square representing x_i. Choose one of these and erase the other and all its descendants from the layout. If X_i is below (respectively, to the right of) R, then place a horizontal (respectively, vertical) NCRVG layout of its descendants to its right (resp., beneath it); place the new layout to the right of (resp., below) the entire current configuration. Thus, for each i, the descendants of x_i are still represented twice in the full layout.

Then continue this process for each grandchild $x_{i,j}$ of r and then for each descendant of r. For each descendant d of r, there are two choices for its placement, one with west-visibility and one with visibility blocked to the west. And when a choice is made, two potential locations for each descendant of d are found, one with west-visibility and one with that visibility blocked. Once the component of F containing r is completed, the same procedure is carried out for all vertices in a second component of F and then for each component of F. At the end, a long rectangle R_1 is placed to the left of the entire configuration, extended to see R if r and r_1 are adjacent, and the noncollinear layout is complete. The layout process is illustrated in Figure 2, in which solid rectangles represent final positions, and dashed rectangles represent rejected potential positions. □

We do not know in general about the possibility of a noncollinear representation of a tree union two depth-1 trees or even of a weak representation of a tree union a depth-2 tree; in the next section we turn to the question of the union of two depth-2 trees.

4 Unions with depth-2 trees

In this section we show that the union of two depth-2 trees is always a weak RVG and, if the two trees have the same root, then it is an RVG. If the roots are different, the weak RVG result follows from a modification of the proof of the Caterpillar Theorem of [2], as shown in Prop. 3 below; to achieve the non-weak RVG result when the roots are the same, a different approach is needed, as shown in Thm. 3.

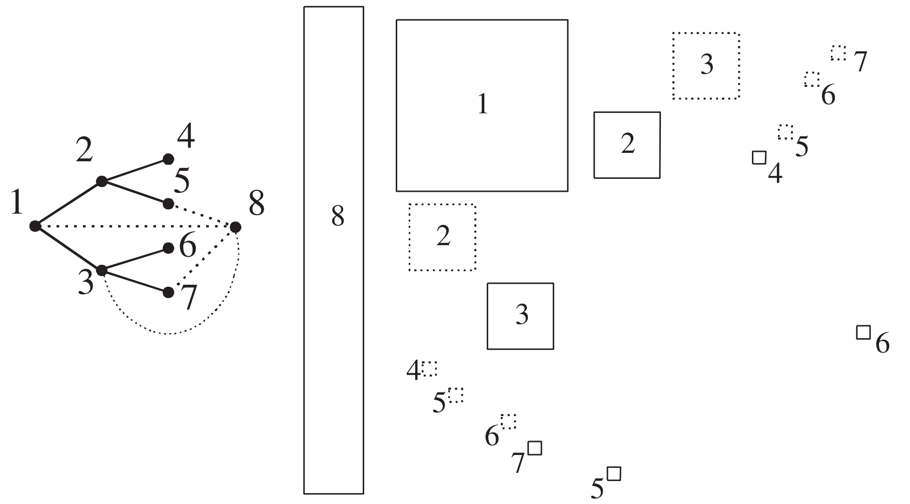

Figure 2: NCRVG layout for a union of a forest and a depth-one tree

Proposition 3 *If a graph G is the union of two depth-2 trees with different roots, then G is a weak RVG.*

Proof: Let $G = T_0 \cup T_1$, where, for $i = 0, 1$, T_i is a depth-2 tree with root r_i, and $r_0 \neq r_1$. Then for $i = 0, 1$ we define $F_i = T_i \setminus r_i$, so that F_i is a depth-1 forest. We add to F_i, as isolated vertices, any vertices of $G \setminus r_i$ not already in F_i, thus keeping F_i a depth-1 forest (note that now $r_{1-i} \in F_i$). As a final adjustment to the F_i graphs, we remove from F_1 any edges that already appear in F_0, so that F_0 and F_1 become edge-disjoint forests.

We begin by representing F_0 as an interval graph on the positive x-axis with each depth-1 vertex of F_0 represented by a proper subinterval of its parent's interval. In addition we arrange the interval graph on the axis so that the intervals representing the depth-1 tree containing r_1 are leftmost. We then delete the interval representing r_1 from the interval representation of F_0. Next we perform an analogous process, representing F_1 along the positive y-axis, with the intervals representing the depth-1 tree containing r_0 lowest in the layout; we then delete r_0 from the interval representation of F_1. We now take the cartesian product of the intervals for F_0 and F_1 to form rectangles in the first quadrant, non-intersecting since the forests are edge-disjoint. We draw a rectangle, below the x-axis (resp., left of the y-axis) and extending the full extent of the layout, label it R_0 (resp., R_1), and if r_0 is adjacent to r_1, extend R_0 leftward to see R_1. The adjacencies of the trees in F_0 (resp., F_1) are represented vertically (resp., horizontally). The projection of a T_0-level-2 rectangle on the x-axis is a proper subinterval of its parent's projection; hence R_0 is visible to all its T_0-children. In addition, R_0 sees all the vertices to which it is adjacent in T_1, since they are the closest to the x-axis. Similarly, all adjacencies with r_1 are represented, and so the configuration is a weak layout of G. The construction is illustrated in Figure 3. □

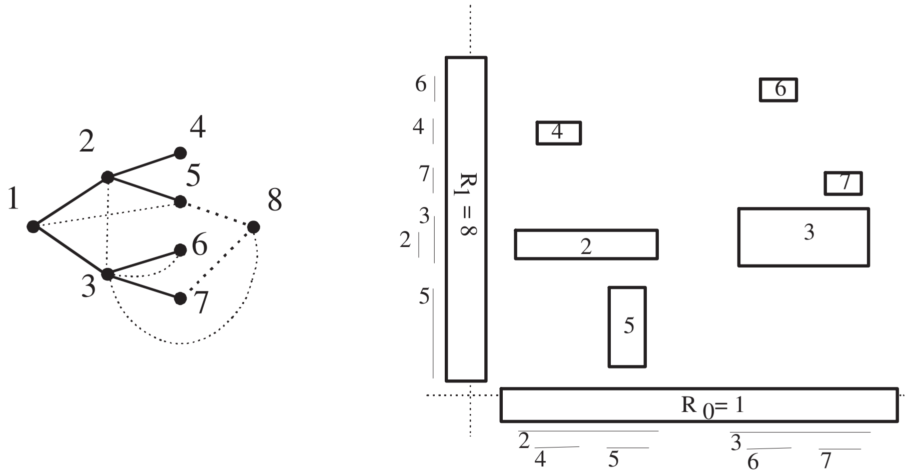

Figure 3: Weak RVG layout for a union of two depth-2 trees with different roots

If, using the notation of the proof of Prop. 3, $r_0 = r_1$, we can get an RVG layout G, but the proof is more involved.

Theorem 3 *If a graph G is the union of two depth-2 trees with a common root, then G is an RVG.*

Proof: Let $G = T_0 \cup T_1$, where T_0 and T_1 are depth-2 trees with common root r. We direct the edges of each tree from parent to child and color the edges of T_0 and T_1, red and blue, respectively (if $e \in T_0 \cap T_1$, color e red). Let D_1 be the induced graph on $V(D_1) = \{v \in G | v \text{ is depth-1 in } T_0 \text{ or } T_1 \text{ (or both)}\}$. Then D_1 is a 2-edge-colored digraph with the following properties. First, there can be no monochromatic directed path of length 2 or more in D_1; otherwise the second vertex on the path would have depth-1 and depth-2 in the same tree. Second, no vertex of D_1 can have more than one incoming edge: two incoming edges of the same color are forbidden since each T_i is a tree, and two incoming edges of different colors would mean the vertex is level-2 in both trees, hence not in D_1. Thus each (undirected) connected component of D_1 is either a tree (of arbitrary depth) or a collection of vertex-disjoint trees (also of arbitrary depth) whose roots are joined in a cycle (which must be *even*).

We begin the layout of G by specifying how to lay out each component of D_1 so that, for every pair of vertices in the component, there is an empty rectangular region that is west-visible from one vertex and south-visible from the other (we say the pair has *southwest visibility*).

For a component that is a tree T with root r_T, we lay out T (but not in the standard vertical SDA format). First we represent r_T as a wide rectangle R_T

and place the depth-1 vertices in an SDA below R_T; each pair of vertices thus laid out has southwest visibility. We repeat this process for the children of each vertex v having depth-1 in T, first shifting all rectangles positioned southeast of v southwards to create space for an SDA of v's children below v and northwest of all these shifted rectangles; southwest visibility of pairs is thus maintained. We continue this process for the children of the vertices at each level of T, until the entire tree is laid out, maintaining southwest visibility of pairs as we proceed. We call the resulting NCRVG layout a *southwest-visible* layout of T. Note that, in addition to each *pair* of vertices having southwest visibility, each *individual* vertex sees nothing to its south, west, or east.

Suppose next that a component C of D_1 consists of a cycle $v_1, v_2, ..., v_k, v_1$, together with vertex-disjoint trees, T_{v_i} rooted at v_i, for $i = 1, ..., k$. We lay out each T_{v_i} in a southwest-visible configuration as described above, with the total layout of T_{v_i} contained in a box B_i. We adjust the proportions of the B_i boxes so that they are all squares of the same size, and then we set them along the main diagonal of a $k \times k$ square, with B_1 at the upper left and B_k at the lower right, so that no rectangle in B_i sees any rectangle in B_j if $i \neq j$. We then modify this configuration to include the cycle; thus far we have an NCRVG layout, but we will give up noncollinearity in this step. Let the rectangle representing vertex v_i of the cycle be labeled V_i. First extend the right side of V_1 beyond the left side of V_k but not as far as the right side of V_k, so that it sees all the other V_i rectangles to its south. Then extend the right sides of $V_2, ..., V_{k-1}$ so that they are collinear and are half-way between the left side of V_k and the right side of V_1. This is a (collinear) southwest-visible layout of the component C, and as before, in addition to each pair of vertices having southwest visibility, each individual vertex sees nothing to its south, west, or east.

Once each component C_i (where $C_i = C$ or T as described above) is given a southwest-visible layout as described above, with component C_i's layout contained in a rectangular region E_i, we adjust the proportions of the boxes E_i so that they are all squares of the same size, and we set them in an SDA along the main diagonal of a larger square box E. The entire configuration is a southwest-visible RVG layout of D_1. The construction guarantees that, in addition to each pair of vertices having southwest-visibility, each individual vertex sees nothing to its south, west, or east.

It is now a simple matter to add rectangles representing the remaining vertices of G to the layout. With the exception of the common root r, any vertex not in D_1 is depth-2 in either T_0 or T_1, or in both, and hence a child of either one or two vertices in D_1. For each pair $v \neq w$, with $v, w \in D_1$, we represent their common children in an SDA in the left half of the rectangular region that is south-visible to one and west-visible to the other. To handle the vertices not in D_1 that have exactly one parent in D_1 we use a rectangular region F below the current layout contained in E and having the same size as E. For each $v \in D_1$, we locate the rectangle F_v in F that corresponds to v's rectangle in E, and we place those vertices, not in D_1, that are children only of v in an SDA in the right half of the rectangle F_v. Thus E contains all the vertices that are depth-1 in either tree, plus those depth-2 vertices that have two level-1 parents;

the latter are blocked to the east by one of their parents, and are seen to the south from the left half of the other parent. The box F below E contains no depth-1 vertices; each depth-2 vertex in F is seen to the south from the right half of its (unique) parent in E. Hence we can represent the final vertex, namely the common root r of T_0 and T_1, by a rectangle R that lies to the right of E and extends the entire length of E; R sees only the vertices of D_1, as it should. Figure 4 illustrates the layout process for a particular graph G, showing G, the subgraph D_1, and the final RVG layout for G.

All the above operations preserve an RVG layout at every stage, and so the proof is complete. □

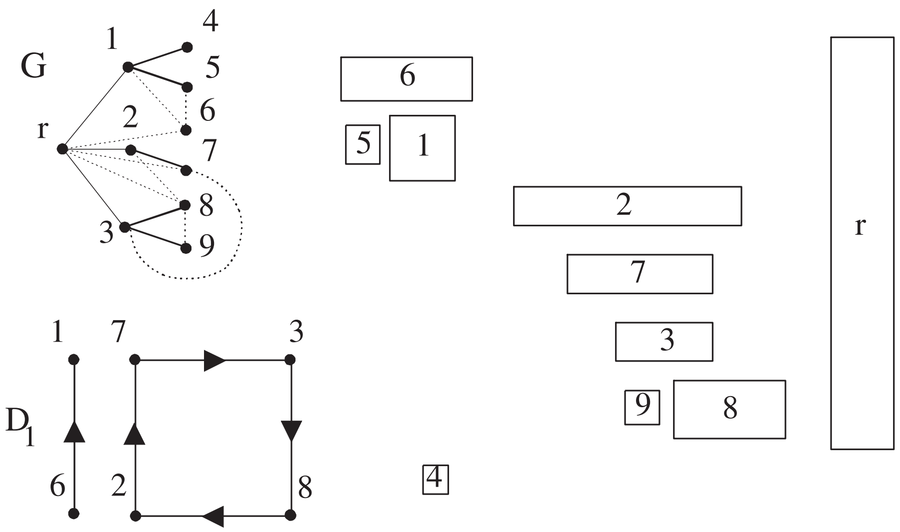

Figure 4: RVG layout for a union of two depth-2 trees with a common root

If, in the proof of Thm. 3, D_1 contains no cycles, then a noncollinear layout is obtained. Prop. 3 and Thm. 3 together imply the following.

Corollary 1 *If G is the union of two depth-2 trees, then G is a weak RVG.*

Note that the proof of Thm. 3 is easily modified to give an alternate proof of Prop. 3, by putting a second root along the west side of the layout. Furthermore, by putting a third root along the south side of the layout, we get a weak representation of the union of two depth-2 trees and a depth-1 tree (recall from Thm. 2 that the union of one depth-2 tree with a depth-1 tree is an NCRVG).

Corollary 2 *If $G = T_1 \cup T_2 \cup T_3$, where T_1 and T_2 are depth-2 trees and T_3 is a depth-1 tree, then G is a weak RVG.*

Aside from the above results, we do not know in general about the possibility of representing a tree union a depth-2 tree. In [1] it is shown that the union of two depth-3 trees need not be an RVG. In the next section we show that the union of a depth-3 tree with a matching is a weak RVG.

5 Unions of depth-3 trees and matchings

In this section our primary focus is on unions involving depth-3 trees. By Thm. 2 the union of a depth-3 tree with a depth-1 tree is an NCRVG. It is not known if the union of a depth-3 tree with a depth-2 tree is an RVG, but the result of [1] says that the union of two depth-3 trees need not be an RVG. Here we try to enlarge the class of RVGs that are unions with depth-3 trees by considering unions of trees with matchings, especially the union of a depth-3 tree with a matching.

By the Caterpillar Theorem, the union of a depth-1 forest and a matching is an NCRVG since both are caterpillar forests. By Thm. 3 the union of a depth-2 tree and a matching is an RVG since the matching can be extended to a depth-2 tree using the same root as the given tree; one corollary of the main result of this section, Theorem 4, is that this union is an NCRVG. We obtain some additional NCRVG results on special unions of trees with matchings; however, the main theorem is that the union of a depth-3 tree and a matching is a weak RVG.

To introduce some new techniques involving the use of SDAs, here is a first result.

Proposition 4 *Suppose G is the union of a tree T with root r and a matching M such that $(x, y) \in M$ implies $dist_T(x, r) = dist_T(y, r)$. Then G is an NCRVG.*

Proof: Suppose T is a subtree of a d-ary tree with root r and depth p. We use a standard NCBVG layout for T, then expanding it to be formed by rectangles as follows. Use a collection of $p+1$ disjoint, congruent square boxes, placed in the plane, one above the other. Label these $A_0, A_1, ..., A_p$ from top to bottom, and let the root r be represented by the square R that equals A_0. Then represent the children of r as an SDA in box A_1. If X is a square in A_1, representing vertex x, then represent its children as an SDA in the square corresponding to X in A_2, and do this for each child of the root. Repeat this process for each vertex in each level of the tree T.

Suppose vertices x and y, both at distance i from r in T, are joined in the matching M. These vertices are represented by squares X and Y in box A_i, lying in subboxes along the diagonal; suppose X lies above and to the left of Y. Then Y can be shifted vertically upwards until it sees X and no other vertex to its left. Y still sees its parent above and its children below, and sees no additional vertex.

□

Note that such vertical shifting can also allow matchings between a child of vertex x with y when the corresponding rectangle X lies above and to the left of Y, but in this case such a simple shift does not work to match a child of Y with

X. Some results, similar to Prop. 4, will follow from our main result, which we turn to now.

Theorem 4 *If G is a union of a depth-3 tree T and a matching M, then G is a weak RVG.*

Proof: We lay out G using a collection of seven boxes: a rectangle R to represent the root r of T, and six disjoint, congruent square boxes, labeled A through F, alphabetically in the order they are used. Choose a positive integer d so that T is a subtree of a d-ary tree. The vertices of T are labeled r, x_i, $x_{i,j}$, and $x_{i,j,k}$, $1 \leq i, j, k \leq d$, as described in Section 2.

The layout process of the proof is illustrated in Figure 5, in which T is a subtree of a ternary tree; for ease of viewing, the rectangles are not drawn to scale. Its root r is adjacent to three vertices x_1, x_2, and x_3, each themselves with three additional level-2 neighbors (children). Of these $x_{1,1}$, $x_{1,2}$, $x_{1,3}$, $x_{2,1}$, and $x_{2,2}$ are adjacent to two leaves each, $x_{3,1}$ to three, and the rest are leaves themselves. The matching M joins the following pairs of vertices of T: $\{x_1, x_{2,1}\}$, $\{x_2, x_{1,1}\}$, $\{x_{3,2}, x_{3,3}\}$, $\{x_3, x_{2,1,1}\}$, $\{x_{2,3}, x_{2,1,2}\}$, and $\{x_{3,1}, x_{1,2,1}\}$. The final position of each rectangle is shown in black outline; a shaded rectangle represents an initial position from which the rectangle is later moved.

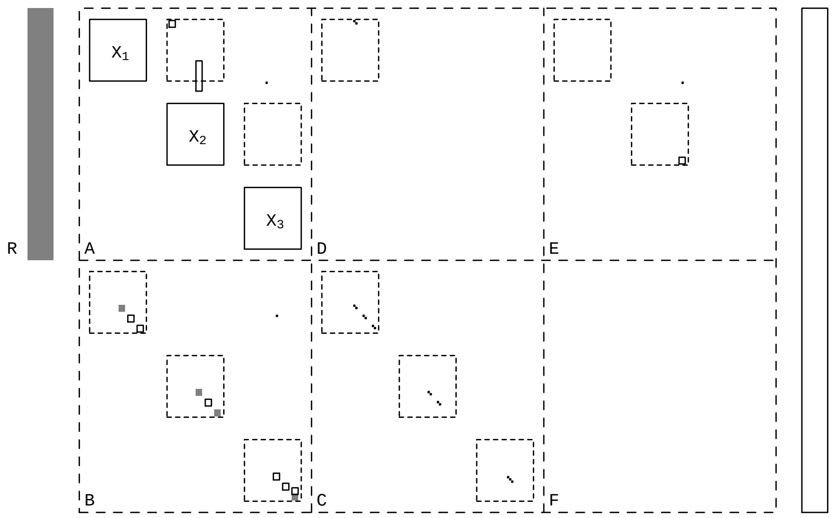

Figure 5: Weak RVG layout for a union of a depth-3 tree and a matching

In general we place the level-1 children of r to its right in an SDA in the box A. In particular, if we divide A into d^2 subboxes labeled $A_{i,j}$, $1 \leq i, j \leq d$, then X_i is a square occupying the middle third of $A_{i,i}$ (that is, the middle third in each dimension). The level-2 children of a level-1 vertex x_i will lie in box B. We begin by choosing the box B_i in B that corresponds to the square X_i in A.

We place the children of x_i in an SDA in the lower-right quadrant of B_i, each occupying the middle third of a square on the diagonal of this quadrant. Hence all edges from x_i to its children correspond to visibility lines running vertically down from the right half of the square X_i. The level-3 children of a level-2 vertex $x_{i,j}$ lie in box C, in an SDA in the subbox $C_{i,j}$ of C corresponding to the square $X_{i,j}$, and again each occupies the middle third of a square on the diagonal. Note that due to the one-third sizing, each square now lies in the middle third of a subbox with enough perimeter room so that another square of the same size or a whole SDA of another same-sized square can be fit into any side of the same subbox. Thus we have laid out the entire depth-3 tree; see Figure 5.

Next we modify (in seven steps) this layout to add in the visibilities corresponding to the edges of the matching. Note that originally and through step 4 we maintain a noncollinear RVG layout, but the layout may become weak in step 5.

Step 1: Matching two level-1 vertices. Suppose x_i is matched with x_j with $i < j$. First we shift the square X_j vertically upwards so that the top of X_j lies halfway up the height of X_i; thus X_j sees X_i to its left, protrudes below X_i, but fits within the original box $A_{i,j}$ so that X_j gains no additional visibilities. Both X_i and X_j maintain east-visibility.

Before this shift, for $k > j$, $A_{j,k}$ was a subbox of A with visibility to both X_j and X_k; we refer to "$A_{j,k}$ *translated*" or $A^t_{j,k}$ to mean the new subbox of A that after the vertical shift of X_j lies to right of X_j and above X_k. $A^t_{j,k}$ will be used in the next step.

Step 2: Matching a level-1 and level-2 vertex. First consider a matching between a child $x_{i,k}$ of x_i with a level-1 vertex x_j, where $i < j$. Let $A^*_{i,j}$ denote the middle third subbox of $A_{i,j}$ or of $A^t_{i,j}$ if X_i was translated in step 1 (in the latter case x_i was matched with some $x_{i'}$, where $i' < i$). We use the upper-left quadrant of the box $A^*_{i,j}$. Find the rectangle $X_{i,k}$ lying in the lower-right quadrant of B_i, and move it into the corresponding position in the upper-left quadrant of $A^*_{i,j}$. Then X_i sees $X_{i,k}$ to its east, and X_j sees $X_{i,k}$ vertically. We must also move the children of $x_{i,k}$ so that they retain visibility to their parent, and we move them into box D from their position in box C as follows.

Let $h_{i,k}$ be the horizontal strip east of (the newly relocated) $X_{i,k}$, intersected with D. The children of $x_{i,k}$ currently form an SDA in a subbox of C. Move this SDA up vertically until these rectangles all lie within $h_{i,k}$. $X_{i,k}$ now sees its children to the east, and we repeat the above process for each child of each x_i that is matched with a level-1 vertex x_j with $i < j$.

Suppose next that x_i is matched with a child $x_{j,k}$ of some x_j, where $i < j$. Then, as above, move the rectangle $X_{j,k}$ from the lower-right quadrant of B_j into the corresponding position in the same quadrant of $A^*_{i,j}$, and vertically move the box containing its children in C into D so that they are directly to the right of $X_{j,k}$.

In summary, each $A^*_{i,j}$, with $i < j$, contains at most two squares, a child of x_i matched with x_j in the upper-left quadrant, and either a child of x_j matched with x_i in the lower-right quadrant or the level-1 vertex x_j matched with x_i in the lower half. Note also that the rectangles in $C \cup D$ retain their original horizontal ordering from box C. Also, if x_i has more than one child matched with different level-1 vertices, then X_i sees them all either to the east in the same relative vertical order that they held before being moved or north in the same horizontal order as before.

Note that by initially laying out a tree in a zigzag pattern, these ideas can be used to prove the following.

Proposition 5 *Suppose G is the union of a tree T with root r and a matching M such that $(x, y) \in M$ implies that either $dist_T(x, r) = dist_T(y, r)$ or that $dist_T(x, r) = 2k - 1$ and $dist_T(y, r) = 2k$ for some integer k (and the values of k may vary). Then G is an NCRVG.*

In other words, edges of the matching join vertices at the same level or at alternate adjacent levels; a similar result holds when all matching edges are between levels 2k and 2k+1.

Step 3: Matching two level-2 vertices. Suppose $x_{i,j}$ is matched with $x_{k,m}$ with (i, j) lexicographically less than (k, m); note that both vertices are still in their original position in box B. First shrink the SDA containing $x_{i,j}$'s children in C by half in both dimensions and move this SDA into the top half of its previous position, clearing the bottom half of the box. Next move the square $X_{k,m}$ vertically upwards so that the top of $X_{k,m}$ lies halfway up the height of $X_{i,j}$ (as in step 1) so that $X_{k,m}$ sees $X_{i,j}$ to its left and protrudes below $X_{i,j}$, but sees nothing more to its left. Finally, move the SDA containing $X_{k,m}$'s children vertically upwards in C until it is directly to the right of $X_{k,m}$. These operations preserve an NCRVG layout.

The proof so far can be used to give the following.

Proposition 6 *The union of a depth-2 tree and a matching is an NCRVG.*

Step 4: Matching a level-1 vertex with a level-3 vertex. Suppose we must match x_i with $x_{j,k,m}$ (where possibly $i = j$). Suppose first that the rectangle $X_{j,k}$ has not been moved outside of box B. Then move $X_{j,k,m}$ horizontally until it is entirely below the left-hand half of X_i. Either this area was empty or, if $i = j$, contained rectangles only below the right-hand half of X_i. Then $X_{j,k,m}$ sees its parent and X_i, but it blocks no visibilities nor introduces any new ones.

Otherwise $X_{j,k}$ has been moved to the right of, say, $X_{i'}$. We consider two cases according as $x_{j,k}$ is or is not a child of $x_{i'}$. In the former case the square $X_{j,k}$ lies in (and is the only square in) the upper-left quadrant of a subbox $A^*_{i',j}$ of A, and the square $X_{j,k}$ can be expanded leftwards until it almost reaches the left boundary of the subbox $A^*_{i',j}$ ($= A_{i',j}$ or $A^t_{i',j}$). Thus this new rectangle enters a vertical band v, free of all other rectangles and wide enough to contain an SDA of d level-3 vertices. Place $X_{j,k,m}$ within the band v so that it sees $X_{j,k}$ vertically and X_i horizontally in the top third of X_i's horizontal band of visibility. In the latter case $X_{j,k}$ lies in the lower-right quadrant of $A^*_{i',j}$ and can be extended downward into a horizontal band h, similarly free of all other rectangles and high enough to contain an SDA of d level-3 vertices. Place $X_{j,k,m}$ within h so that it sees $X_{j,k}$ horizontally and X_i vertically in its leftmost third. Note that we may put several level-3 children of $x_{j,k}$, staggered in the bands h and v and positioned so that all desired visibilities and nonvisibilities are maintained.

Step 5: Matching a level-2 vertex with a level-3 vertex. If vertex $x_{i,j}$ is matched with vertex $x_{k,m,n}$, then $(i,j) \neq (k,m)$ (but possibly $i = k$) and $X_{i,j}$ is in its original position in box B. If $X_{k,m,n}$ is still in box C (when $X_{k,m}$ is unmatched or matched with another level-2 vertex), then we move $X_{k,m,n}$ horizontally until it is above or below the left-hand half of $X_{i,j}$; thus it is placed in a subbox of B in the same horizontal band as $X_{k,m}$ and in the same vertical band as $X_{i,j}$. This subbox was previously empty, but if $(k,m) < (i,j)$, the layout becomes weak since $X_{k,m,n}$ sees X_i vertically. ($X_{k,m}$ and its SDA of children may have been shifted or its SDA of children may have been contracted if $X_{k,m}$ was matched with another level-2 vertex, but this does not matter.)

If $X_{k,m,n}$ has been moved into box D because its parent $X_{k,m}$ has been matched with a level-1 vertex, then we will use box E, the square box congruent to and to the right of D, to contain $X_{i,j}$ (and all level-2 vertices that are matched with level-3 vertices in D). Let E_i be the subbox of E corresponding to X_i in A. Form d^2 subboxes along the diagonal of the upper-left quadrant of E_i and place $X_{i,j}$ in the middle third of the j'th subbox down the diagonal (this is the subbox corresponding to its subbox in the lower-right quadrant of B_i). X_i will now see $X_{i,j}$ to the east (but $X_{i,j}$ does not yet see $X_{k,m,n}$); X_i may also see other children in this direction in subboxes $A_{i,j}$ or $A^t_{i,j}$, $j > i$, and in E_i, but they will each be placed along the diagonal of their enclosing subbox so that X_i sees these children vertically in the same order in which it initially saw them horizontally below in B. X_i sees the rest of its children vertically in boxes A and B.

Next all the (matched and unmatched) children of $x_{i,j}$ must be moved. First put all children of $x_{i,j}$ back in their original SDA in C. Then move this SDA vertically upward into D until it lies within the horizontal band west of $X_{i,j}$ and east of X_i. Due to spacing between rectangles, this SDA does not block visibility between X_i and $X_{i,j}$, but the layout may have become weak, if it wasn't already. All unmatched children of $x_{i,j}$ are now in their proper positions as is a child that was matched with x_i. Let $v_{i,j}$ be the vertical band extending above and below $X_{i,j}$ in boxes E and F, the latter box below E and to the east of C. All other matched children of $X_{i,j}$, and also $X_{k,m,n}$, will be placed appropriately in $v_{i,j}$ as follows.

If a child $x_{i,j,p}$ of $x_{i,j}$ was matched with x_q, $q \neq i$, (and so previously had been moved horizontally into box B), then let D_i be the box in D corresponding to X_i (and now containing all of $X_{i,j}$'s children in its upper-left quadrant). We move $x_{i,j,p}$ from its position within the upper-left quadrant of D_i to the corresponding position in the upper-left quadrant of E_i, and then move it vertically (within $v_{i,j}$) until it is contained in the lower half of the (empty) horizontal band east of X_q. Then X_q sees $X_{i,j,p}$ to its east from its lower half, $X_{i,j}$ sees $X_{i,j,p}$ (and any other such children) in the vertical band $v_{i,j}$. If several children of $x_{i,j}$ are matched with different x_k's, they are staggered in $v_{i,j}$ so that visibility to $X_{i,j}$ is maintained.

If a child $x_{i,j,p}$ of $x_{i,j}$ had previously been matched with a level-2 vertex $x_{q,r}$ (by a prior application of this case), then $X_{i,j}$ was in B when this match occurred, and $X_{i,j,p}$ was in C. Thus $X_{q,r}$ was left in its original position, and $X_{i,j,p}$ was moved into B to see $X_{q,r}$ vertically. Move $X_{i,j,p}$ from its position in the upper-left quadrant of D_i to the corresponding position in the upper-left quadrant of E_i and then move it downwards (within $v_{i,j}$) until it is within the horizontal band east of the lower third of $X_{q,r}$ (so that visibility is not blocked by $X_{q,r}$'s children). Note that $X_{i,j,p}$ has been moved into the box F. Again if several children of $x_{i,j}$ are matched with different $x_{q,r}$'s, they are staggered in $v_{i,j}$ so that visibility to $X_{i,j}$ is maintained and no two of these children see each other.

Finally we are ready to move $X_{k,m,n}$ horizontally into the right-most part of the band $v_{i,j}$ so that it sees $X_{i,j}$ vertically and obstructs no visibility of $X_{i,j}$ with its children. $X_{k,m,n}$ still sees its parent horizontally in A or in E. Because of the "middle-third" construction of the vertices in the original tree layout, $X_{k,m,n}$ sees none of the children of $X_{i,j}$, although it might be collinear with the right-most child of $X_{i,j}$.

Step 6: Matching two level-3 vertices. Note that all unmatched level-3 vertices lie in C or D; there may also be matched level-3 vertices in these boxes. The vertical ordering of the level-3 vertices in $C \cup D$ may have been permuted, but their horizontal ordering is unchanged. Furthermore, they all see their parents horizontally. To match two of these level-3 vertices, take the leftmost one and move it right until it sees the other vertically.

Step 7: Matching the root with another vertex. Observe that all level-1 vertices and unmatched level-2 vertices are east-visible. All unmatched level-3 vertices are either in C or D, and they are east-visible except when their parent lies in E. We move the root r and represent it now by a rectangle to the right of the entire layout, extending from top to bottom of the layout. Then r sees all its children. If r is matched with a level-3 vertex $X_{i,j,k}$ in D whose parent $X_{i,j}$ lies in E, then we move $X_{i,j,k}$ horizontally into the upper-right quadrant of E_i. Note that moving r introduces many extraneous visibilities.

□

We do not know if the results of Theorem 4 are best possible, meaning that we do not know whether the union of a depth-4 tree and a matching or the union of a depth-3 tree and a depth-2 tree is an RVG, though by the results of [1] the union of two depth-3 trees is not necessarily an RVG. The example of [1] does have a weak RVG representation and so we also wonder if unions of depth-4 trees have a weak representation. Clearly there are other unions of trees to be studied; for example, the case of caterpillar trees suggests that a breadth-first analysis might be as fruitful as the depth-oriented one we have pursued here.

6 Cartesian Products

In this section we give results on thickness and rectangle-visibility for cartesian products of graphs. Our two main results are that the thickness of a product is bounded above by the sum of the thicknesses of the factors (and this is best possible), and that the product of two BVGs is an RVG. From the latter we conclude that the product of two forests is an RVG. We also generalize the second result to products of what we call "left-right-bar-visibility" graphs with certain paths and cycles. We apply these results to conclude that several classes of graphs are RVGs.

We first give several results on the thickness of product graphs. If G is the product of two planar graphs, then the horizontal and vertical edges partition G into two planar graphs. More generally, if G_1 and G_2 are two arbitrary graphs and $G = G_1 \times G_2$, then the horizontal edges of G can be partitioned into $\theta(G_1)$ planar graphs, and likewise the vertical edges of G can be partitioned into $\theta(G_2)$ planar graphs. We have thus proved the following:

Proposition 7 *If G is the product of two planar graphs, then $\theta(G) \leq 2$.*

Proposition 8 *If $G = G_1 \times G_2$, then $\theta(G) \leq \theta(G_1) + \theta(G_2)$.*

Corollary 3 *If $G = \prod(G_i)$, then $\theta(G) \leq \sum \theta(G_i)$.*

The graphs $P_n \times K_2$ and $C_n \times K_2$ (where P_n and C_n are, resp., the path and cycle on n vertices) show that equality need not hold in Prop. 7. On the other hand, if G is planar and contains K_4, equality is achieved by $G \times K_2$, since $G \times K_2$ contains a K_5 minor. The hypercube graphs provide a useful class of examples to demonstrate that the bounds in Prop. 8 can always be achieved, and that they also can always be made strict, for any two thicknesses θ_1 and θ_2. Kleinert [7] has shown that the thickness of the hypercubes is given by the formula $\theta(Q_n) = \lceil (n+1)/4 \rceil$. Given any two thicknesses, θ_1 and θ_2, we can achieve the upper bound in Prop. 8 by taking $G_i = Q_{4\theta_i - 1}$, for $i = 1, 2$. On the other hand, to get strict inequality we take $G_i = Q_{4\theta_i - 4}$, in which case $\theta(G_1 \times G_2) = \theta_1 + \theta_2 - 1$.

It follows from Proposition 7 that the product of two trees has thickness at most two. In the next proposition we show that the product of two BVGs is an RVG, and so the product of two trees is an RVG.

Proposition 9 *If the graphs G and H are both BVGs, then $G \times H$ is an RVG. Analogous statements hold if G and H are noncollinear or weak BVGs.*

Proof: Assume we have BVG layouts of G and H using horizontal bars with left and right endpoints in the interval $[0, 1]$; without loss of generality, we assume that in each layout the bars have distinct y-coordinates. Number the vertices of G and H in order of their bars in each case from bottom to top, $i = 1, ..., n_G$ and $j = 1, ..., n_H$; the bars themselves will be denoted $B_G(i)$ and $B_H(j)$. We give an RVG layout for $G \times H$ in the region $[0, 2n_G] \times [0, 2n_H]$, so that the rectangle

$R(i,j)$ lies in the region $[2i-1,2i]\times[2j-1,2j]$. The top, bottom, left and right coordinates (indicated by t, b, l, r, respectively) of $R(i,j)$, $i=1,...,n_G$, $j=1,...,n_H$, are as follows:

$$\begin{aligned} t[R(i,j)] &= 2j-1+r[B_G(i)] \\ b[R(i,j)] &= 2j-1+l[B_G(i)] \\ l[R(i,j)] &= 2i-1+l[B_H(j)] \\ r[R(i,j)] &= 2i-1+r[B_H(j)] \end{aligned}$$

Looking vertically in this layout we see $|V(G)|$ copies of H, each in a different column, and similarly $|V(H)|$ copies of G when looking horizontally. It is easy to see that $R(i,j)$ sees $R(i',j')$ if and only if either $i=i'$ and $B_H(j)$ sees $B_H(j')$ in the BVG layout of H, or $j=j'$ and $B_G(i)$ sees $B_G(i')$ in the BVG layout of G. The construction is illustrated in Figure 6. It is also clear that this construction preserves noncollinearity and weakness. □

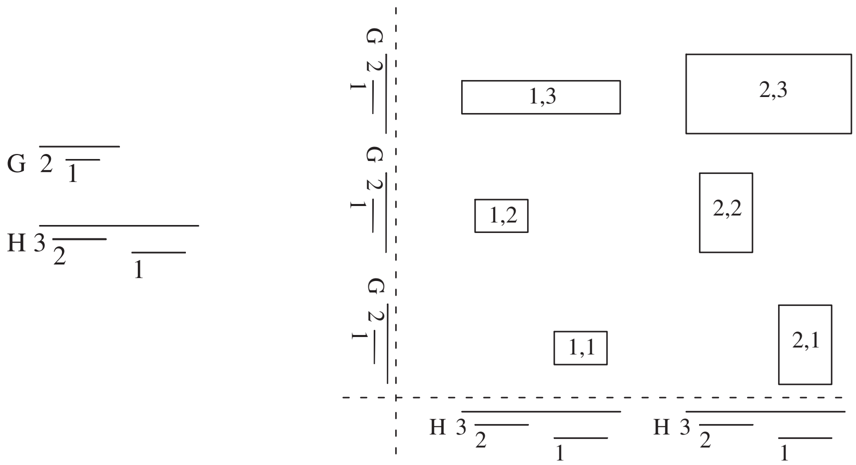

Figure 6: RVG layout of $G\times H$, for BVGs G and H

Corollary 4 *If G is the product of two trees or forests, then G is an NCRVG.*

Although the focus of this paper is combinations of trees, we note that since the p-dimensional hypercube Q_p is a BVG for $p=1,...,3$, we can apply Prop. 9 to get that Q_p is an RVG for $p=1,...,6$.

Corollary 5 *The hypercube graphs Q_n, for $n=1,...,6$, are RVGs.*

Q_3 is actually a BVG, since it is a 2-connected planar graph. It is known that the hypercubes of thickness 2 are $Q_4,...,Q_7$ [7]. A bipartite RVG has $e\le 4v-12$ [3, 4], and it follows from results in [8] that a bipartite NCRVG has $e\le 2n-2$; hence Q_7 cannot be an NCRVG but could be an RVG. It is not known if Q_7 is an RVG.

In the proof of Prop. 9 we can get a somewhat stronger result if the two BVG layouts have an additional property. A *left-right* bar-visibility layout of a graph G, abbreviated LRBVG layout, is a BVG layout of G using horizontal bars contained in a rectangular region $[a, b] \times [c, d]$, such that each bar has either left endpoint a or right endpoint b (these bars are called *left* and *right* bars, respectively). Examples of LRBVGs include all paths and cycles. It is easy to see that if G is an LRBVG, then G is *outerplanar*, i.e., it has a planar embedding in which all vertices lie on the outer face. The proofs of the following propositions concerning LRBVGs are fairly straightforward, and are omitted.

Proposition 10 *If G and H are BVGs, and at least one of G and H is an LRBVG, then $G \times H \times P_2$ is an RVG.*

Corollary 6 *If G and H are both LRBVGs, then $G \times H \times K$ is an RVG, for $K = P_2, P_3, P_4$, or C_4.*

If Q_3 were an LRBVG, then it would follow from Cor. 6 that Q_7 is an RVG, but this is not the case: It is well known that a graph G is outerplanar if and only if it does not contain a subgraph homeomorphic to K_4 or $K_{2,3}$. Since Q_3 contains a homeomorph of $K_{2,3}$, it is not outerplanar, hence not an LRBVG.

References

[1] I. Bjorling-Sachs and T. Shermer, *personal communication*, 1996.

[2] P. Bose, A. Dean, J. Hutchinson, and T. Shermer, On Rectangle Visibility Graphs, *Lecture Notes in Computer Science #1190* (S. North, ed.), Springer-Verlag, Berlin, 25–44, 1997.

[3] A. Dean and J. Hutchinson, Rectangle-visibility representations of bipartite graphs, *Discrete Applied Mathematics*, 75:9–25, 1997.

[4] A. Dean and J. Hutchinson, Rectangle-visibility representations of bipartite graphs (extended abstract), *Lecture Notes in Computer Science #894* (R. Tamassia and I.G. Tollis, eds.), Springer-Verlag, Berlin, 159–166, 1995.

[5] J. Hutchinson, T. Shermer, and A. Vince, On representations of some thickness-two graphs, *Computational Geometry: Theory and Applications*, to appear.

[6] J. Hutchinson, T. Shermer, and A. Vince, On representations of some thickness-two graphs (extended abstract), *Lecture Notes in Computer Science #1027*, (F. Brandenburg, ed.), Springer-Verlag, Berlin, 324–332, 1995.

[7] M. Kleinert, Die Dicke des n-dimensionalen Würfel-Graphen, *J. Comb. Theory*, 3:10–15, 1967.

[8] F. Luccio, S. Mazzone, and C. Wong, A note on visibility graphs, *Discrete Mathematics*, 64:209–219, 1987.

[9] C. Nash-Williams, Decomposition of finite graphs into forests, *J. London Math. Soc.*, 39:12, 1964.

[10] L. Pyber Problem C.23, *Contests in Higher Mathematics, Milkós Schweitzer Competitions 1962-1991*, (G.J. Székely, ed.), Springer, Berlin, 42, 1996.

[11] T. Shermer, On rectangle visibility graphs III. external visibility and complexity, *Proc. 8th Canad. Conf. Comput. Geom.*, Carleton University Press, Ottawa, 234–239, 1996.

[12] R. Tamassia and I.G. Tollis, A unified approach to visibility representations of planar graphs, *Discrete and Computational Geometry*, 1:321–341, 1986.

[13] S. Wismath, Characterizing bar line-of-sight graphs, *Proc. 1st Symp. Comp. Geom.*, ACM, 147–152, 1985.