

A System for Drawing Graphs with Geometric Symmetry

Joseph Manning^{1*}, Mikhail Atallah^{2**},
Kingsley Cudjoe¹, Joe Lozito¹, Ron Pacheco³

¹ Vassar College, NY, USA

² Purdue University, IN, USA

³ Harding University, AR, USA

Abstract. This demonstration presents a computer system to generate symmetric drawings of certain classes of planar graphs.

Among the most important criteria for generating good drawings of graphs is the display of axial symmetry and/or rotational symmetry, collectively known as *geometric symmetry*. Its importance stems from the fact that, given a symmetric drawing, a conceptual understanding of the entire graph can be built up from that of a smaller subgraph, replicated a number of times. This demonstration presents an interactive graphical system which can detect geometric symmetries in certain classes of graphs, and construct drawings of the graphs which display these symmetries. The system also includes a number of convenient facilities for the manual construction, editing, and layout of graphs.

The basic problem of determining if a general abstract graph possesses any geometric symmetry, along with several variations, are all \mathcal{NP} -complete [7, 8]. Accordingly, the current research has focused on detecting symmetry in planar graphs, since these form an important subclass of general graphs and frequently admit efficient solutions to problems which are otherwise intractable. Optimal linear-time algorithms, outlined below, have been developed for detecting and displaying both axial and rotational symmetries in *trees*, *outerplanar graphs*, and *embedded planar graphs*. Note that the input to the algorithms consist simply of the adjacency lists in the case of trees and outerplanar graphs; in the case of embedded planar graphs, each adjacency list is ordered to reflect the clockwise arrangement of edges emanating from each vertex, and a list of the vertices on the outer face is also specified.

Test runs have shown that the optimal theoretical time complexities of the drawing algorithms do indeed translate into a fast practical system. Running on a mid-range workstation, graphs with up to one hundred vertices can be processed in a fraction of a second. The system is written in Modula-2, and runs under OpenWindows on a SUN SPARCstation.

The system also includes facilities for direct manipulation of abstract graphs and their drawings. Using mouse clicks and menu selections, it provides for the

* Contact Address: manning@cs.vassar.edu

** Research supported by the National Science Foundation under Grant CCR-9202807

creation and deletion of vertices and edges, repositioning of vertices, loading of both abstract graphs and drawings, and storing and printing of drawings, in addition to invocation of the symmetry routines.

A number of other systems, primarily those which are based on the concept of *spring embedder*[2], can also generate symmetric drawings, but without actually setting out to do so. By contrast, the present system explicitly computes the geometric symmetries of the given graph, and uses these to generate its drawing. One other system which adopts a similar approach is described in [4].

An overview of the algorithms for the three classes of graphs which the system can process, together with an example drawing from each class, is given below; more complete details of these algorithms may be found in [5, 6, 8].

Trees [5, 8]: A tree is a connected acyclic graph. All symmetries of a tree must keep its center fixed (a *center* of a tree is any vertex whose maximum distance from any leaf is minimized; every tree has either one center or two adjacent centers; the latter case may be reduced to the former by introducing a new vertex on the edge joining the two centers). Removing the center from a tree divides the remainder of the tree into a number of subtrees, and any symmetry of the overall tree must permute these among themselves, mapping subtrees to isomorphic subtrees. Using a variation of the linear-time tree-isomorphism test [1, p84], these subtrees are partitioned into isomorphism classes, from which the geometric symmetries are subsequently determined. A radial drawing of the tree, which displays these symmetries, is then constructed. Since it is impossible, in general, to display all of its axial symmetries in a single drawing of a given tree, the algorithm instead determines the maximum number of simultaneously-displayable axial symmetries, and constructs the corresponding "most symmetric" drawing.

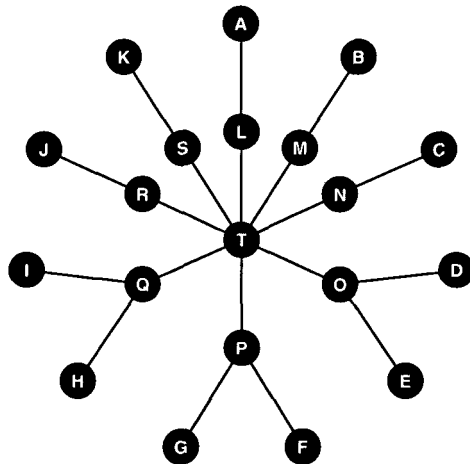


Fig. 1. Symmetric Drawing of a Tree

Outerplanar Graphs [6, 8]: An outerplanar graph is one which can be drawn in the plane with no edge crossings and with all vertices on the outer face. Every biconnected outerplanar graph has a unique Hamilton cycle, which may be found in linear time. By traversing its Hamilton cycle, the graph is transformed into a string, in such a way that geometric symmetries of the graph correspond to certain “symmetries” of the string. These, in turn, are found using an efficient pattern-matching algorithm [3], from which the symmetries of the graph are then recovered. By contrast with the situation for trees, all geometric symmetries of a biconnected outerplanar graph may be displayed in a single drawing, which the algorithm then constructs. (Linear-time algorithms for non-biconnected outerplanar graphs have also been developed, but have not yet been implemented in this system.)

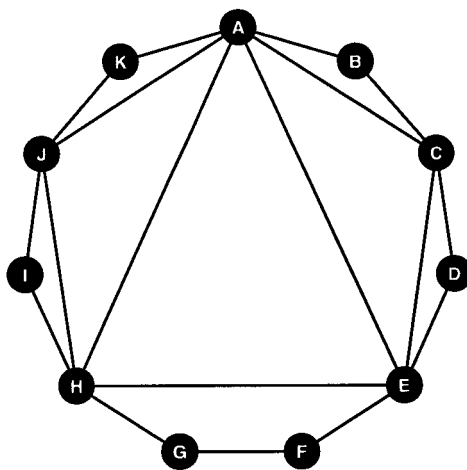


Fig. 2. Symmetric Drawing of an Outerplanar Graph

Embedded Planar Graphs [8]: A planar graph is one which can be drawn in the plane with no edge crossings, and a plane embedding merely lists the order of edges emanating from each vertex as well as the vertices on the outer face, without specifying either the coordinates of the vertices or the shapes of the edges. A planar graph, with such an embedding, is first transformed into a number of “concentric” biconnected outerplanar levels, and the geometric symmetries of these levels are then found using the outerplanar algorithm and “intersected” to give the symmetries of the original graph. This algorithm has particular relevance to drawing triconnected planar graphs, whose plane embeddings are unique up to the choice of outer face; enumerating each of its $\mathcal{O}(n)$ embeddings, and applying the present algorithm to each in turn, yields an $\mathcal{O}(n^2)$ algorithm for symmetry detection in any triconnected planar graph.

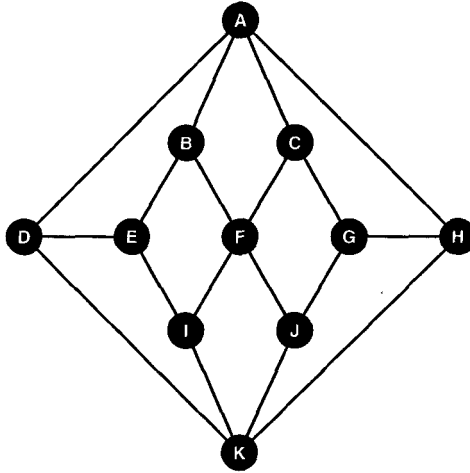


Fig. 3. Symmetric Drawing of an Embedded Planar Graph

References

1. A. Aho, J. Hopcroft, J. Ullman: *The Design and Analysis of Computer Algorithms*. Addison-Wesley (1974)
2. P. Eades: A Heuristic for Drawing Graphs. *Congressus Numerantium* 42 (1984) 149-160
3. D. Knuth, J. Morris, V. Pratt: Fast Pattern Matching in Strings. *SIAM Journal on Computing* 6 (1977) 323-350
4. R. Lipton, S. North, J. Sandberg: A Method for Drawing Graphs. *Proceedings of the ACM Symposium on Computational Geometry* 1 (1986) 153-160
5. J. Manning, M. Atallah: Fast Detection and Display of Symmetry in Trees. *Congressus Numerantium* 64 (1988) 159-169
6. J. Manning, M. Atallah: Fast Detection and Display of Symmetry in Outerplanar Graphs. *Discrete Applied Mathematics* 39 (1992) 13-35
7. J. Manning: Computational Complexity of Geometric Symmetry Detection in Graphs. *Lecture Notes in Computer Science* 507 (1991) 1-7
8. J. Manning: Geometric Symmetry in Graphs. Ph.D. Thesis, Purdue University (1990)