Martin C. Golumbic
March 19, 1976

# Recognizing Comparability Graphs in SETL

## 1. Introduction

An undirected graph (V,E) is a comparability graph if one can assign an orientation to the edges which is transitive as a binary relation on the vertices. This transitive orientation will be a partial ordering of the vertices whose comparability relation is exactly E. Gilmore and Hoffman [1964] and Ghouila-Houri [1962] characterized comparability graphs, and an algorithm for testing transitive orientatability and producing such an orientation is given in Pnueli, Lempel and Even [1971]. An improved version of that TRO algorithm follows from the results of Golumbic [1975].

In this paper we show how SETL allows the mathematician to move from the strict algebraic setting into an algorithmic presentation in a very natural and effortless manner. This is in fact one of the main purpose of SETL's design. We use our algorithm for recognition of comparability graphs as the setting for this discussion. In a preliminary version of another paper by this author, we discuss the computational complexity of the algorithm. Efficient solutions to the clique problem and minimum coloring problem for comparability graphs are also discussed.

## 2. Definitions

A graph (V,E) consists of an anti-reflexive binary relation E over a finite set V of vertices. The members of E are called arcs or edges and can be thought of as ordered pairs of distinct vertices.

Thus we are assuming all graphs are loop-free and have no multiple edges. We define the relations

$$[ab \in E^{-1} \Longleftrightarrow ba \in E] \quad \text{and} \quad \bar{E} = E \cup E^{-1}$$

respectively. A graph is <u>undirected</u> if $E = E^{-1}$.

An undirected graph (V,E) is called a <u>comparability graph</u> if there exists a graph (V,E) such that

$$F \cap F^{-1} = \emptyset; \quad F \cup F^{-1} = E; \; F^2 \subseteq F$$

where $F^2 = \{ac \mid ab, bc \in F \text{ for some vertex } b\}$. The relation F is a partial ordering of V whose comparability relation is precisely E and F is called a <u>transitive orientation</u> of E.

Let (V,E) be an undirected graph. Define the binary

$$ab \; \Gamma \; a'b' \quad \text{iff} \quad \begin{array}{l} \text{either } a = a',\; bb' \notin E \\ \text{or } b = b',\; aa' \notin E \end{array}$$

The relation $\Gamma$ represents a type of local forcing. Since E is anti-reflexive, ab $\Gamma$ ab, <u>however</u>, ab $\not\Gamma$ ba. The reader should not continue until he is convinced of this fact. The reflexive, transitive closure $\Gamma^*$ of $\Gamma$ is an equivalence relation on E and hence partitions E into what we shall call the <u>implication classes</u> of E. Thus edges ab and cd are in the same implication class if and only if there exists a $\Gamma$-<u>chain</u> of edges

$$ab = a_0\, b_0 \;\Gamma\; a_1\, b_1 \;\Gamma\; \ldots \;\Gamma\; a_k\, b_k = cd, \quad (k \geq 0).$$

If one considers the graph $(E,\Gamma)$, then the implication classes of (V,E) correspond to the connected components of $(E,\Gamma)$.

Examples.

i. The graph in Figure 1 has implication classes:

$A_1 = \{ab, ac\}$  $A_2 = \{bd, cd\}$  $A_3 = \{ad\}$

$A_1^{-1} = \{ba, ca\}$  $A_2^{-1} = \{db, dc\}$  $A_3^{-1} = \{da\}$

ii. The graph in Figure 2 has only one implication class:

$$A = \{ab, cb, cd, ed\ \ ea\ \ ba, bc, dc, de, ae\} = \bar{A}$$

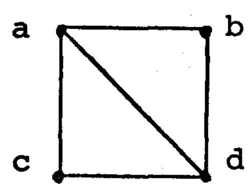


Figure 1

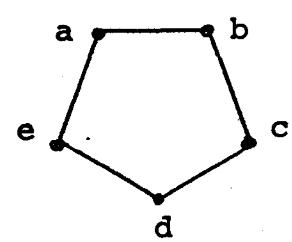


Figure 2.

## 3. Bridging the gap between mathematics and computing.

A complete investigation of the subject at hand must satisfy two criteria. First, mathematics demands that our characterization of comparability graphs be elegant. The definitions must be stated in such a way that the theorems and their proofs are a sweet song. Second, computing requires an algorithm, moreover, an efficient one. Too many works unsuccessfully use the language of computer science as a setting for mathematics. SETL is an exception. This is not to say that cumbersome programs are not to be found in the SETL literature, but the opportuntity for a transparent treatment is available. So, without further delay, let us avail ourselves of SETL.

Let (V,E) be an undirected graph. $E = \bar{B}_1 + \ldots + \bar{B}_k$ is called a <u>G-decomposition</u> of E if $B_i$ is an implication class of $\bar{B}_i + \ldots + \bar{B}_k$ for all $i = 1, \ldots, k$. Every undirected graph has a G-decomposition, perhaps many, and we can construct one in a simple way:

$$\left.\begin{array}{l}\text{Arbitrarily choose an edge } e_1\text{, enumerate its}\\ \text{implication class } B_1\text{, remove all edges in } \bar{B}_1\\ \text{and continuing inductively until no edge remain.}\end{array}\right\} \quad (1)$$

THEOREM (Golumbic) Let (V,E) be an undirected graph with G-decomposition $E = \bar{B}_1 + \ldots + \bar{B}_k$. The following statements are equivalent:

i. (V,E) is a comparability graph.
ii. $A \cap A^{-1} = \emptyset$ for all implication classes A of E.
iii. $B_i \cap B_i^{-1} = \emptyset$ for all $i = 1,\ldots,k$.

Furthermore, when these conditions hold, $B_1 + \ldots + B_k$ is a transitive orientation of E.

The interested reader is refered to Golumbic [1975] for a proof of this theorem. It is also shown there that the length of any G-decomposition is an invariant for a given graph which is due to an underlying matroid structure.

We now translate the method outlined in (1) directly into a SETL algorithm which uses the theorem to recognize and transitively orient a comparability graph.

<u>Algorithm 1</u>. Transitive orientation

Let (V,E) be an undirected graph. An arbitrary edge is chosen and its implication class B is enumerated via a depth first search of $\Gamma$. If $B \cap Binv \neq \emptyset$, then the graph has no transitive orientation by the theorem. Otherwise, B is stored as part of F, B + Binv is deleted from E and we continue until E is empty. The algorithm consists of a call to the function transitiveorientation.

```
Static B,Binv; /* declares variables to be global and not stacked */

definef transitiveorientation (V,E);
     /*  (V,E) is an undirected graph.  Transitiveorientation
     will return a set of ordered pairs F such that (V,F) is
     a transitive orientation of (V,E) provided that (V,E)
     is a comparability graph, and will return nℓ otherwise. */

initialize:  F = nℓ;
(while E ne nℓ)
     B = nℓ;  Binv = nℓ;
     <x,y> =  ∋ E;
          explore (x,y);
     if B * Binv ne nℓ then return nℓ;;
     F = F + B;
     E = E - (B + Binv);
end while;
return F;
end transitiveorientation;
```

```
define explore (x,y);

     <x,y> in B; <y,x> in Binv;
(∀ z ∈ V)
     if   <x,z> ∈ E and not (<z,y> ∈ E) and not(<x,z> ∈ B)
     then
               explore (x,z);;

     if   <z,y> ∈ E and not (<x,z> ∈ E) and not (<z,y> ∈ B)
     then
               explore (z,y);;
end ∀;
return;
end explore;
```

Computational Complexity: The algorithm as presented here EXPLOREs each edge or its reversal once, and it the proper data structures are chosen, then EXPLOR requires $O(N)$ steps. The entire process could thus be done in $O(|V|.\ |E|)$ steps. In Golumbic [1976] we discuss an implementation which assigns the transitive orientation in $O(\delta.|E|)$ steps where $\delta$ is the maximum degree of a vertex. Both versions use $O(|E|)$ space.

The author notes with regret that certain natural statements are not permitted in SETL. Replacing

```
          (while E ne nℓ)
               <x,y> = ∋ E;
```

with single statement

```
          (while ∃ <x,y> ∈ E)
```

would free the researcher from programming worries even further and add to the "layman readability".

## 4. Cliques and colorings of comparability graphs

Any acyclic orientation (V,F) of an undirected graph gives a natural partial ordering of V where x > y if there exists a path in F from x to y. A height function is then induced on the vertices: h(r) = 0 if v is a sink; otherwise, h(v) = 1 + MAX {h(w) | vw ∈ F}. The height function can be calculated in a straightforward manner.

Algorithm 2. Height of a poset

```
definef height (V,F);
    /* (V,F) is an acyclic orientation of an undirected
     graph. The routine height will return the height
     function h of the natural partial ordering on V. */

initialize: h = nℓ;  i = 0;

(while V ne nℓ)
    S = { y ∈ V | F{y} eq nℓ};
    T = {<x,y> ∈ F | y ∈ S};
    (∀ y ∈ S)  h(y) = i;;
    i = i + 1;
    V = V - S;
    F = F - T;
end while;
return h;
end height;
```

The height function is a valid vertex coloring (adjacent vertices have different colors), but it is <u>not</u> necessarily a minimum coloring. The situation is much better if F is also transitive.

It is well known that for <u>comparability graphs</u> the size of the largest clique and the chromatic number are equal, (see Berge [1973].) Furthermore, the height function of a transitive orientation F <u>is</u> a minimum vertex coloring, and the maximal paths of F correspond precisely to the maximal cliques.

The more general weighted clique problem, where each vertex is assigned a positive integer weight and a clique whose vertices have largest total weight is to be found, is similarly solved for comparability graphs. These problems are NP-complete for arbitrary graphs, but polynomial solutions can be found for many special classes of graphs by exploiting their particular structure as we have in algorithms 1 and 2.

## Acknowledgement

The author would like to express his gratitude to Dr. Joseph Cross for the many useful discussion we've had.

## Bibliography

Claude Berge, *Graphs and Hypergraphs*, Chapter 16, North-Holland, Amsterdam, (1973).

S. Even, A. Pnueli and A. Lempel, *Permutation Graphs and Transitive Graphs*, J. ACM, 19 (1972), pp. 400-410.

P.C. Gilmore and A.J. Hoffman, *A Characterization of Comparability Graphs and of Interval graphs*, Can. J. Math., 16 (1964), pp. 539-548.

Martin C. Golumbic, *Comparability Graphs and a new Matroid*, Ph.D. Thesis, Columbia University, (1975); to appear, J. Comb. Th., Series B.

Martin C. Golumbic, *The Complexity of Comparability Graph Recognition and Coloring*, Courant Institute Research Report, (1976).

A. Pnueli, A. Lempel and S. Even, *Transitive Orientation of Graphs and Identification of Permutation Graphs*. Can J. Math., 23 (1971), pp. 160-175.