The Intersection Graphs of Subtrees in Trees Are Exactly the Chordal Graphs

Fănică Gavril

Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel

Communicated by W. T. Tutte

Received March 19, 1973

The intersection graph of a family of subtrees in an undirected tree is called a subtree graph. A graph is called chordal if every simple circuit with more than three vertices has an edge connecting two non-consecutive vertices. In this paper, we prove that, for a graph G, the following conditions are equivalent:
(i) G is a chordal graph; (ii) G is a subtree graph; (iii) G is a proper subtree graph.

Consider a chordal graph G. We give an efficient algorithm for constructing a representation of G by a family of subtrees in a tree.

1. Introduction

In this paper, we consider only finite, undirected graphs G(V) with no parallel edges and no self-loops, where V is the set of the graph vertices. For a subset V₁ of V, the subgraph G(V₁) of G(V) is the graph whose set of vertices is V₁, two vertices being connected by an edge if and only if they are connected in G(V). A completely connected set of G is a set of vertices which has every pair of vertices connected by an edge. A clique of the graph is a maximal, completely connected set of vertices. For a vertex v of G, we will denote the set of vertices connected to v by Tv. For two sets A and B, we denote the set of elements of A which are not in B by A - B. A tree is a connected graph without simple circuits. Consider a tree T(V), when V is the set of its vertices. If U ⊆ V, then T(U) is the subgraph defined by U.

Consider a finite family of non-empty sets. The intersection graph of this family is obtained by representing each set by a vertex, two vertices being connected by an edge if and only if the corresponding sets intersect. If a graph G is the intersection graph of a family of sets F, we say that F is a representation of G. The problem of characterizing the intersection graph
of a family of sets having a defined topologic pattern is of great interest in
different domains. The intersection graph of a family of intervals on a
linearly ordered set is called an interval graph. Hajós [1] first put the
problem of characterizing the interval graphs. Complete characterizations
were given by Gilmore and Hoffman [2], Lekkerkerker and Boland [3], and
Fulkerson and Gross [4]. The latter paper [4] also contains the most
efficient algorithm for constructing a representation by intervals of a given
graph if one exists. The interval graphs have applications in genetics
(Benzer [7]), psychophysics (Roberts [5]), archeology (Kendall [6]), and
ecology (J. Cohen [8]).

The intersection graph of a family of arcs on a circularly ordered set is
called a circular-arc graph. The problem of characterizing these graphs
appears in [9] and [10]. Tucker [11] found a characterization of the cir-
cular-arc graphs by means of the adjacency matrix. He also gave an
efficient algorithm for identifying the proper circular-arc graphs. But an
efficient recognition algorithm for all the circular-arc graphs is not known.
Such an algorithm could be quite useful in genetics, as pointed out by
Stahl [12].

More generally, for any topologic graph, it is of interest to characterize
the intersection graphs of finite families of connected subgraphs with a
given pattern. For example, Renz [13] considered the problem of charac-
terizing the intersection graphs of families of arcs in trees. In this paper,
we continue the above ideas as follows: Let us consider an undirected tree
as a topologic pattern (such as a line or a circle). A connected portion of
the tree will be called a subtree. Hence, a subtree, defined in this way, can
have its end-points on an edge, between the vertices of the tree (see Fig. 1b).
The intersection graph of a family of subtrees in an undirected tree will be
called a subtree graph. For example, the graph G of Figure 1a is the
subtree graph of the family \( F = \{ \overline{v_1}, \ldots, \overline{v_7} \} \) on the tree \( T \) of Figure 1b. Let
us assume that this graph has a representation by a family of arcs in a tree
(discussed in [13]). Every vertex \( v_i, 1 \leq i \leq 7 \), must be represented by an
arc \( \overline{v_i} \). For every \( 1 \leq i \leq 6 \), \( v_i \) is connected to \( v_7 \), hence \( \overline{v_i} \cap \overline{v_7} \neq \emptyset \).
Since \( T \) is a tree, \( \overline{v_i} \cap \overline{v_j} \neq \emptyset \) if and only if \( \overline{v_i} \cap \overline{v_j} \cap \overline{v_7} \neq \emptyset \). Therefore,
the family \( \{ \overline{v_i} \cap \overline{v_j} \}_{i,j=1}^6 \) must be a representation by intervals on \( \overline{v_7} \) of
\( G(\{v_1, \ldots, v_6\}) \). But \( G(\{v_1, \ldots, v_6\}) \) is not an interval graph, as proved in [4].
Thus, the graph of Figure 1a has no representations by a family of arcs in
a tree. Therefore, the family of subtree graphs contains the family of graphs
discussed in [13], but is different from it.

A graph is called chordal if every simple circuit with more than three
vertices has an edge connecting two non-consecutive vertices. Dirac [14]
and Rose [15] have proved that every chordal graph \( G(V) \) has a vertex \( v \),
so that \( Fv \) is completely connected; \( v \) is called a simplicial vertex. Based on
this fact, Rose [15] gives an efficient algorithm for recognizing the chordal graphs. It is known also (see [16]) that the number of the cliques of a chordal graph is at most as the number of its vertices.

The purpose of this paper is to prove that a graph is a subtree graph if and only if it is a chordal graph. We will also describe an efficient algorithm which will construct a representation by a family of subtrees, to a given chordal graph.

We can assume without loss of generality that the families we deal with contain only subtrees which are closed subsets of the tree—this because we can move in a suitable way the end-points of the subtrees without changing the intersection relations. Throughout the paper we will consider only undirected trees. As is known, in an undirected tree, for every two points there exists a unique simple path connecting them. If the family $F$ is a representation of a graph $G$, we will denote the subtree corresponding to a vertex $v$ by $\bar{v}$. 

![Diagram](image-url)
2. **Subtree Graphs**

Consider a subtree graph $G$, and its representation $F$ on a tree $T$.

**Lemma 1.** For every completely connected set $A$ of $G$, $\bigcap_{i \in A} \bar{v} \neq \emptyset$.

**Proof.** Consider three points $a_1, a_2, a_3$ on $T$. Let $F$ be a subset of $F$ so that every element of $F$ contains at least two of the points $a_1, a_2, a_3$. Let us denote:

$$F_1 = \{t \mid t \in F \text{ and } a_1, a_2 \in t\},$$

$$F_2 = \{t \mid t \in F \text{ and } a_1, a_3 \in t\},$$

$$F_3 = \{t \mid t \in F \text{ and } a_2, a_3 \in t\}.$$

Let $p_1, p_2, p_3$ be the simple paths of $T$ connecting $a_1$ with $a_2$, $a_1$ with $a_3$, and $a_2$ with $a_3$, correspondingly. Since $T$ is a tree, it follows that $p_1 \cap p_2 \cap p_3 \neq \emptyset$. For every $1 \leq j \leq 3$, if $t \in F_j$, then $p_j \subseteq t$. Hence $p_i \subseteq \bigcap_{i \in F_j} t$, and thus

$$\left( \bigcap_{i \in F_1} t \right) \cap \left( \bigcap_{i \in F_2} t \right) \cap \left( \bigcap_{i \in F_3} t \right) \supseteq p_1 \cap p_2 \cap p_3 \neq \emptyset.$$

Therefore $\bigcap_{i \in F} t \neq \emptyset$.

Let us now prove the lemma by induction on the number of vertices $i$, in a completely connected set. If $i = 1$ or $i = 2$, the lemma is clearly true. Assume it is true for every $i < k$. Consider a completely connected set $A = \{v_1, ..., v_k\}$. By the induction hypothesis;

$$\bigcap_{i=1}^{k-1} \bar{v}_i \neq \emptyset, \quad \bigcap_{i=2}^{k} \bar{v}_i \neq \emptyset, \quad \bar{v}_1 \cap \bar{v}_k \neq \emptyset.$$

Let

$$a_1 \in \bigcap_{i=1}^{k-1} \bar{v}_i, \quad a_2 \in \bigcap_{i=2}^{k} \bar{v}_i, \quad a_3 \in \bar{v}_1 \cap \bar{v}_k.$$

Every element of $F = \{\bar{v}_1, ..., \bar{v}_k\}$ contains at least two of the points $a_1, a_2, a_3$, Therefore, by the previous remark, $\bigcap_{i=1}^{k} \bar{v}_i \neq \emptyset$. Q.E.D.

Clearly, for two different cliques $A_1$ and $A_2$ we have

$$\left( \bigcap_{v \in A_1} \bar{v} \right) \cap \left( \bigcap_{v \in A_2} \bar{v} \right) = \emptyset.$$

For a graph $G$, we will denote the set of its cliques by $\mu(G)$, and the set of the cliques containing a vertex $v$, by $\mu_v(G)$. 
**Theorem 2.** A graph $G(V)$ is a subtree graph if and only if there exists a tree $T$ whose set of vertices is $\mu(G)$, so that, for every $v \in V$, $T(\mu_v(G))$ is connected.

**Proof.** Assume that $G(V)$ is a subtree graph, and let $F$ be its representing family on a tree $T_1$. Let $\mu(G) = \{A_1, ..., A_k\}$. By Lemma I, for every clique $A_i$ we have $s_i = \bigcap_{v \in A_i} \overline{v} \neq \emptyset$, and for $i \neq j$, $s_i \cap s_j = \emptyset$. For every $A_i$, take a point in $s_i$ and denote it $A_i$. With the points $A_1, ..., A_k$ on $T_1$ we construct a graph $T$ in the following way: we connect in $T$ the points $A_i$ and $A_j$ by an edge if and only if the simple path connecting $A_i$ with $A_j$ on $T_1$ does not contain other points of $A_1, ..., A_k$. In other words, we connect $A_i$ with $A_j$ if they are neighbors on $T_1$. Clearly, $T$ is a tree, and, for every $v \in V$, $T(\mu_v(G))$ is connected.

Conversely, let $T$ be a tree whose set of vertices is $\mu(G)$, so that, for every $v \in V$, $T(\mu_v(G))$ is connected. Let

$$F = \{T(\mu_v(G)) | v \text{ is a vertex of } G\}.$$ If $T(\mu_v(G)) \cap T(\mu_u(G)) \neq \emptyset$ then there exists a vertex $A_i$ of $T$ so that $A_i \in T(\mu_v(G)) \cap T(\mu_u(G))$. Hence $A_i \in \mu_v(G)$, $A_i \in \mu_u(G)$, thus $v, u \in A_i$ and therefore $v$ is connected to $u$ in $G$. On the other side, if $v$ is connected to $u$ in $G$, then there exists a clique $A_i$ so that $u, v \in A_i$, and hence $A_i \in T(\mu_v(G)) \cap T(\mu_u(G))$. Thus, $G$ is the intersection graph of $F$, and therefore it is a subtree graph. Q.E.D.

Now we will prove the main theorem of this paper:

**Theorem 3.** A graph $G$ is a subtree graph if and only if it is a chordal graph.

**Proof.** Consider a subtree graph $G$, and $F$ its representing family on a tree $T$. Let us assume that $G$ contains a simple circuit $v_1, v_2, ..., v_k, v_1, k > 3$, so that every two non-consecutive vertices are not connected by an edge. For every $1 \leq i \leq k - 1$ let $a_i$ be a point of $T$ in $\overline{v_i} \cap \overline{v_{i+1}}$, and let $a_k$ be a point in $\overline{v_k} \cap \overline{v_1}$. Let $p_i$, $1 \leq i \leq k - 1$, be the simple path of $T$ which connects $a_i$ with $a_{i+1}$, and let $p_k$ be the simple path connecting $a_k$ with $a_1$. Hence, for every $1 \leq i \leq k - 1$, $p_i \subseteq \overline{v_{i+1}}$ and $p_k \subseteq \overline{v_1}$. Therefore, $p_i \cap p_j \neq \emptyset$ only if $j = i + 1$, or $i = k$ and $j = 1$. Also, $p_i \cap p_{i+1} = \{a_{i+1}\}$ and $p_k \cap p_1 = \{a_1\}$. Thus $\bigcup_{i=1}^k p_i$ is a simple circuit in $T$, contradicting the fact that $T$ is a tree. Therefore, $G$ is chordal.

Conversely, we will prove by induction that for a chordal graph $G(V)$ there exists a tree $T(\mu(G))$ so that, for every $v \in V$, $T(\mu_v(G))$ is connected. This is clearly true for graphs with only one vertex. Assume this is true for
every chordal graph with less than \( n \) vertices. Consider a chordal graph \( G(V) \) with \( n \) vertices. If \( G \) is completely connected, then \( \mu(G) = \{G\} \), and \( T \) is a tree with one vertex, denoted \( G \). If \( G \) is disconnected, let \( G_1 \) and \( G_2 \) be two subgraphs so that there are no edges from a vertex of \( G_1 \) to a vertex of \( G_2 \). By the induction hypothesis, there exist the trees \( T_1(\mu(G_1)) \), \( T_2(\mu(G_2)) \) corresponding to \( G_1 \) and \( G_2 \). The tree \( T(\mu(G)) \) corresponding to \( G \) is obtained by connecting a vertex of \( T_1 \) with a vertex of \( T_2 \). Let us assume that \( G \) is connected and that it is not completely connected. Since \( G \) is chordal, by a theorem mentioned in the introduction, \( G \) has a simplicial vertex \( \bar{v} \). Thus \( T\bar{v} \) is completely connected and \( \bar{A} = \{\bar{v}\} \cup \bar{V} \) is a clique of \( G \). Let

\[ S = \{u \mid u \in A \quad \text{and} \quad Tu \not\subseteq A\}. \]

Hence, \( u \in S \) if and only if \( u \in A \) and there exists a vertex \( u' \in V - A \) adjacent to \( u \). Also, \( w \in A - S \) if and only if \( w \in A \) and there are no vertices of \( V - A \) adjacent to it. Clearly, \( A - S \neq \emptyset \) since \( \bar{v} \in A - S \). Also \( V - A \neq \emptyset \) and \( S \neq \emptyset \) since \( G \) is connected and it is not completely connected. It follows that there are no edges connecting a vertex of \( A - S \) with a vertex of \( V - A \). The subgraph \( G_1 = G((V - A) \cup S) \) is chordal and has less vertices than \( G \). Thus, by the induction hypothesis there exists a tree \( T_1 \) whose set of vertices is \( \mu(G_1) \) so that, for every vertex \( u \) of \( G_1 \), \( T_1(\mu(G_1)) \) is connected. There are two cases:

(a) \( S \) is a clique of \( G_1 \). Consider the tree \( T \) obtained from \( T_1 \) by renaming the vertex \( S \) by \( A \). The set of vertices of \( T \) is \( \mu(G) \). Consider a vertex \( v \) of \( G \). If \( v \in V - A \), then \( \mu_v(G) = \mu_v(G_1) \) hence \( T(\mu_v(G)) = T_1(\mu_v(G_1)) \), and thus \( T(\mu_v(G)) \) is connected. If \( v \in A - S \), then \( \mu_v(G) = \{A\} \), and hence \( T(\mu_v(G)) \) is connected. If \( v \in S \), then \( T(\mu_v(G)) \) is obtained from \( T_1(\mu_v(G_1)) \) by renaming the vertex \( S \) by \( A \), and hence it is connected. Thus, for every vertex \( v \) of \( G \), \( T(\mu_v(G)) \) is connected.

(b) \( S \) is not a clique of \( G_1 \). Let \( B \) be a clique of \( G_1 \), so that \( S \subset B \). Consider the tree \( T \) obtained by adding to \( T_1 \) the vertex \( A \) and an edge connecting \( A \) with \( B \). Consider a vertex \( v \) of \( G \). If \( v \in V - A \), then \( T(\mu_v(G)) = T_1(\mu_v(G_1)) \) and hence it is clearly connected. If \( v \in S \), then \( \mu_v(G) = \mu_v(G_1) \cup \{A\} \). Hence \( T(\mu_v(G)) \) is obtained from \( T_1(\mu_v(G_1)) \) by adding the vertex \( A \) and the edge connecting \( A \) and \( B \). Thus \( T(\mu_v(G)) \) is connected. If \( v \in A - S \), then \( \mu_v(G) = \{A\} \), and thus \( T(\mu_v(G)) \) is connected.

Thus, we have proved that, for a chordal graph \( G(V) \), there exists a tree \( T(\mu(G)) \) so that, for every \( v \in V \), \( T(\mu_v(G)) \) is connected. Therefore, by Theorem 2, \( G \) is a subtree graph. Q.E.D.

Based on the proof of Theorem 3, we will describe an efficient algorithm
which constructs the tree $T(\mu(G))$ for a chordal graph $G(V)$. The algorithm works as follows: If $G$ is completely connected, then $T$ has only one vertex, denoted $\hat{v}$. If $G$ is disconnected, let $G_1, G_2$ be two subgraphs so that there are no edges from one to the other. By induction, there exist the trees $T_1(\mu(G_1))$ and $T_2(\mu(G_2))$. Then $T$ is obtained by connecting a vertex of $T_1$ with a vertex of $T_2$. Let us assume that $G$ is connected and that it is not completely connected. Since $G$ is chordal, there exists a vertex $v \in V$, so that $A = \{v\} \cup T\hat{v}$ is a clique of $G$. Let

$$S = \{u \mid u \in A \quad \text{and} \quad Tu \notin A\}.$$  

Hence, there are no edges of $G$ connecting a vertex of $A - S$ with a vertex of $V - A$. Let $G_1 = G((V - A) \cup S)$. $G_1$ has less vertices than $G$, and thus by the induction hypothesis we can construct for $G_1$ the tree $T_1(\mu(G_1))$ so that, for every vertex $u$ of $G_1$, $T_1(\mu_u(G_1))$ is connected. There are two cases:

(a) $S$ is a clique of $G$. Then $T$ is obtained from $T_1$ by renaming the vertex $S$ by $A$.

(b) $S$ is not a clique of $G$. Let $B$ be a clique of $G$ so that $S \subset B$. Then $T$ is obtained by adding to $T_1$ the vertex $A$ and an edge connecting $A$ with $B$.

Clearly, in every inductive stage we can memorize only the connections between the vertices of $T$, and, at the end, we can draw the tree $T$ at once. Let $n$ be the number of $G$ vertices. Since $G$ is chordal, it has at most $n$ cliques. Thus, $T$ has at most $n$ vertices. By the proof of Theorem 2, $G$ is the intersection graph of the family of subtrees

$$F = \{T(\mu_v(G)) \mid v \in V\}.$$  

In this way we have constructed a representation by a family of subtrees for a given chordal graph. The above algorithm takes no more than $n^4$ steps. This bound is given by the work needed for finding at every stage a simplicial vertex. We must find at every stage a simplicial vertex also for testing chordality (see [16]). Therefore we can perform our algorithm simultaneously with the algorithm which tests chordality.

Let us apply the algorithm to the graph $G$ of Figure 1a. A simplicial vertex of $G$ is $v_1$. Then

$$A_1 = \{v_1\} \cup TV_1 = \{v_1, v_4, v_6, v_7\}; S_1 = \{v_4, v_6, v_7\}.$$  

The remaining subgraph is $G_1 = G(\{v_2, ..., v_7\})$, and $S_1$ is contained in the clique $\{v_4, v_5, v_6, v_7\}$. 
In the same way, we have:

\[ A_2 = \{v_2\} \cup \Gamma v_2 = \{v_2, v_4, v_5, v_7\}; S_2 = \{v_4, v_5, v_7\}, \]

\[ G_3 = G(\{v_3, \ldots, v_7\}), S_2 \subseteq \{v_4, v_5, v_6, v_7\}, \]

\[ A_3 = \{v_3\} \cup \Gamma v_3 = \{v_3, v_5, v_6, v_7\}; S_3 = \{v_5, v_6, v_7\}, \]

\[ G_4 = G(\{v_4, v_5, v_6, v_7\}). \]

\( G_4 \) is completely connected. Let us denote the set of vertices of \( G_4 \) by \( A_4 \).

Thus \( S_1, S_2, S_3 \subseteq A_4 \). It follows that every one of \( A_1, A_2, A_3 \) is connected in \( T \) to \( A_4 \). Therefore the graph \( G \) is chordal. The tree \( T \) and the family \( F \) corresponding to \( G \) are those of Figure 2.

\[ \text{FIGURE 2} \]

3. PROPER SUBTREE GRAPHS

A graph is called a **proper interval graph** if it is the intersection graph of a family of intervals on a line so that no one of the intervals is contained in another. Roberts [5] characterized these graphs and showed that they form a proper family of the family of interval graphs. We will see that this is not true for the proper subtree graphs.

A graph is called a **proper subtree graph** if it is the intersection graph of a family of subtrees of a tree so that no one of the subtrees is contained in another.

Let \( G \) be a subtree graph, and \( F \) its representing family on a tree \( T \). Let \( \bar{v}_1, \ldots, \bar{v}_k \in F \) be all the subtrees which are contained in other subtrees of \( F \). For every \( 1 \leq i \leq k \), consider a point \( a_i \) in \( \bar{v}_i \). Let \( T_i \) be the tree obtained by adding to \( T \), for every \( i \), a vertex \( b_i \) and an edge connecting \( a_i \) with \( b_i \).

It is not necessary that the points \( a_1, \ldots, a_k \) be different, but \( b_1, \ldots, b_k \) must be different. For every \( 1 \leq i \leq k \), let \( \overline{a_i} \) be the subtree obtained by adding to \( \overline{v_i} \), the edge connecting \( a_i \) with \( b_i \). Consider the family \( F_1 \) obtained from
THE SUBTREE GRAPHS

$F$ by changing every $\bar{v}_i$ by $\bar{v}_i$. $G$ is also the intersection graph of $F_1$. But $F_1$ has no elements contained in others. Hence $G$ is a proper subtree graph. Therefore the following three conditions are equivalent:

(i) $G$ is a subtree graph,
(ii) $G$ is a proper subtree graph,
(iii) $G$ is a chordal graph.

To obtain a representation of a chordal graph by a family of proper subtrees, we first construct a representation by subtrees as in Part 2, and we transform it into a representation by proper subtrees as above. In Figure 3 we see the family of proper subtrees obtained by this method from the family given in Figure 2.

**Figure 3**

ACKNOWLEDGMENT

The author wishes to express his thanks to Professor Shimon Even for his assistance.

REFERENCES