Equivalence of the filament and overlap graphs of subtrees of limited trees

Jessica Enright[1]   Lorna Stewart[2]

[1] *University of Stirling, Stirling, Scotland, UK*
[2] *University of Alberta, Edmonton, Alberta, Canada*

The overlap graphs of subtrees of a tree are equivalent to subtree filament graphs, the overlap graphs of subtrees of a star are cocomparability graphs, and the overlap graphs of subtrees of a caterpillar are interval filament graphs. In this paper, we show the equivalence of many more classes of subtree overlap and subtree filament graphs, and equate them to classes of complements of cochordal-mixed graphs. Our results generalise the previously known results mentioned above.

Keywords: graph algorithms, intersection graphs, filament graphs

1   Introduction

The class of subtree overlap graphs is equivalent to the class of subtree filament graphs, which means that those graphs have both overlap and filament intersection representations on trees [6]. The class contains many graph classes that have extensive structural properties, algorithms, and complexity results, such as interval graphs, permutation graphs, cocomparability graphs, chordal graphs, circle graphs, circular-arc graphs, polygon-circle graphs, and interval filament graphs. Some of these graphs have been characterised in terms of subtree overlap representations on restricted host trees. In particular, cocomparability graphs are the overlap graphs of subtrees of a star (follows from [12]), circle graphs are the overlap graphs of subtrees of a path [7], and interval filament graphs are the overlap graphs of subtrees of a caterpillar (this fact was presented at a workshop but not published [3]). Thus, we have the equivalence of general subtree overlap graphs and subtree filament graphs, and we know that some subtree overlap graphs that admit representations on restricted host trees are equivalent to well-known graph classes, one of which is a natural class of subtree filament graphs.

In this paper, we identify new equivalences between subtree overlap and subtree filament graph classes based on host trees of their representations. We first introduce the notion of a covering subtree of a tree representation. We show that the host tree of any subtree overlap representation can be modified so that it consists of just a covering subtree plus some additional leaves, without altering the represented graph. In addition, we prove that a graph has a subtree overlap representation with a given covering subtree if and only if it is the complement of a restricted type of cochordal-mixed graph. Finally, we show that for a set $\mathcal{S}$ of trees that is closed under edge subdivision, a graph has a subtree filament representation with host tree in $\mathcal{S}$ if and only if it has a subtree overlap representation with covering subtree in $\mathcal{S}$.

Our first theorem generalises the characterisation of cocomparability graphs as the overlap graphs of subtrees of a star by equating the overlap graphs of subtrees of a star with subtree overlap graphs that have a representation with a single-vertex covering subtree. Theorem 1 generalises this correspondence for any given covering subtree.

Our second theorem bridges the gap between the previously known equivalence of general subtree overlap graphs and subtree filament graphs, and the characterisation of interval filament graphs as the overlap graphs of subtrees of a caterpillar. Both of those previously known results and new equivalences between subtree overlap and subtree filament graph classes are given in Theorem 2. The theorem suggests a division of subtree overlap graphs into subclasses, each of which contains all interval filament graphs, and the union of which is the class of subtree overlap graphs. This view of subtree overlap graphs may give insight into their structure and the algorithmic complexity of problems in that domain.

We consider finite, simple graphs. Let $G = (V, E)$ be a graph. The *neighbourhood* of a vertex $v \in V$ in G is $N_G(v) = \{u \mid uv \in E\}$. K_n denotes the complete graph on n vertices.

Two sets A and B *intersect* if $A \cap B \neq \emptyset$, and *overlap*, denoted $A \between B$, if $A \cap B \neq \emptyset$, $A \not\subseteq B$, and $B \not\subseteq A$. Sets A and B are *disjoint*, denoted $A|B$, if $A \cap B = \emptyset$. Let A, B, A', B' be four nonempty sets. We say that (A, B) and (A', B') are *similarly related*, denoted $(A, B) \sim (A', B')$ if $A|B$ if and only if $A'|B'$ and $A \between B$ if and only if $A' \between B'$.

Let $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ be a multiset of nonempty sets. We use the term multiset rather than set to allow for the possibility that $S_i = S_j$ for some $1 \leq i, j \leq n$ where $i \neq j$. The *intersection graph* (respectively, *overlap graph*, *disjointness graph*, *containment graph*) of $\mathcal{S}$ is the graph $G = (V, E)$ where $V = \{v_1, v_2, \ldots, v_n\}$ and, for all $1 \leq i, j \leq n$, $v_i v_j \in E$ if and only if S_i and S_j intersect (respectively, overlap, are disjoint, are contained one in the other). If G is the intersection, overlap, disjointness, or containment graph of $\mathcal{S}$ then $\mathcal{S}$ is called an intersection, overlap, disjointness, or containment representation of G. Every graph has both an intersection and a disjointness representation [15] as well as an overlap representation (obtained by adding a unique new element to each set of an intersection representation). Note that, for $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ and $\mathcal{S}' = \{S'_1, S'_2, \ldots, S'_n\}$ where $(S_i, S_j) \sim (S'_i, S'_j)$ for all $1 \leq i, j \leq n$, the intersection (respectively, overlap, disjointness, containment) graphs of $\mathcal{S}$ and $\mathcal{S}'$ are isomorphic.

In this paper, we are concerned with the intersection, overlap, disjointness, and containment graphs of subtrees of a tree. For a given tree T, we will assume that a collection of n subtrees of T is given as a multiset $\mathcal{T} = \{t_1, \ldots, t_n\}$ of subsets of the vertices of T, each of which induces a subtree of T.

Interval graphs are the intersection graphs of intervals on a line or, equivalently, the intersection graphs of subtrees of a path. *Cointerval graphs* are the complements of interval graphs, that is, the disjointness graphs of subtrees of a path. *Circle graphs* are the intersection graphs of chords in a circle or, equivalently, the overlap graphs of subtrees of a path. *Chordal graphs* are graphs in which every cycle of length greater than three has a chord or, equivalently, the intersection graphs of subtrees in a tree. *Cochordal graphs* are the complements of chordal graphs. *Comparability graphs* are graphs whose edges can be transitively oriented. Equivalently, comparability graphs are the containment graphs of subtrees of a tree, the containment graphs of subtrees of a star, and the set of all containment graphs [12]. *Cocomparability graphs* are the complements of comparability graphs. *Subtree overlap graphs* are the overlap graphs of subtrees in a tree. If a graph $G = (V, E)$ is the overlap (respectively, intersection, containment, or disjointness) graph of subtrees $\mathcal{T}$ of a tree T, then $\mathcal{T}$ is a *subtree overlap (respectively, intersection, containment, or disjointness) representation* of G. T is termed the *host tree* of the representation. For convenience, we will use the notation that vertex $v_i \in V$ corresponds to subtree $t_i \in \mathcal{T}$. A *caterpillar* is a

tree such that the removal of its leaves results in a path. All of the graph classes defined in this section are *hereditary*, that is, every induced subgraph of a graph in the class is also in the class. For more information about graph classes, see [1].

Interval filament graphs, subtree filament graphs, and $\mathcal{G}$-mixed graphs were introduced by Gavril [8]. We give the definitions of those graph classes and related concepts next.

Let $\mathcal{I} = \{I_1, \ldots, I_n\}$ be a multiset of (closed) intervals on a line L and let P be a plane containing L. We will refer to one of the half-planes into which L divides P as being above L. Then $\mathcal{F} = \{f_1, \ldots, f_n\}$ is a set of interval filaments on the intervals of $\mathcal{I}$ if each f_i, $1 \le i \le n$, is a curve in P, on and above L, connecting the endpoints of I_i, such that if two intervals are disjoint then their curves do not intersect. Thus, pairs of filaments corresponding to disjoint intervals do not intersect, pairs of filaments corresponding to overlapping intervals intersect, and pairs of filaments corresponding to intervals where one is contained in the other may or may not intersect. The intersection graph of $\mathcal{F}$ is called an *interval filament graph*.

Subtree filaments and subtree filament graphs are defined analogously. Let T be a tree and let $\mathcal{T} = \{t_1, \ldots, t_n\}$ be a multiset of subtrees of T. Suppose that T is embedded in a plane P and let S be a surface perpendicular to P whose intersection with P is T. (One can imagine forming the part of S that is above T by drawing T upwards from P.) Then $\mathcal{F} = \{f_1, \ldots, f_n\}$ is a set of subtree filaments on the subtrees of $\mathcal{T}$ if each f_i, $1 \le i \le n$, is a curve in S, in and above T, connecting the leaves of t_i, such that (i) if two subtrees are disjoint then their curves do not intersect, and (ii) if two subtrees overlap then their curves intersect. Thus, pairs of filaments corresponding to disjoint subtrees do not intersect, pairs of filaments corresponding to overlapping subtrees intersect, and pairs of filaments corresponding to intervals where one is contained in the other may or may not intersect. If a graph G is the intersection graph of a collection of filaments on subtrees of a tree T, then G is a *subtree filament graph* and the collection of filaments is a *subtree filament representation* of G. The tree T is called the *host tree* of the representation.

Let $\mathcal{G}$ a hereditary graph class. A graph $G = (V, E)$ is said to be $\mathcal{G}$-*mixed* if there is a partition of its edges into E_1 and E_2 such that:

- $G_1 = (V, E_1)$ is in $\mathcal{G}$ and

- there is a transitive orientation $(V, \overrightarrow{E_2})$ of the graph (V, E_2) such that for every three distinct vertices $u, v, w \in V$, if $(u \to v) \in \overrightarrow{E_2}$ and $vw \in E_1$, then $uw \in E_1$.

Such a partition is called a $\mathcal{G}$-*mixed partition* of the edges of G. Subtree filament graphs are exactly the complements of cochordal-mixed graphs, and interval filament graphs are exactly the complements of cointerval-mixed graphs [8].

Let $\mathcal{T}$ be a multiset of subtrees of a tree T. A subtree t of T is called a *covering subtree* of $\mathcal{T}$ if it intersects every member of $\mathcal{T}$. Note that the intersection of each element of $\mathcal{T}$ with a covering subtree t is a subtree of t. Let T be a tree and let t be a subtree of T. A vertex v of t is called *bushy* (with respect to t in T) if every neighbour of v that is not in t is a leaf of T; the entire subtree t is called *bushy* (in T) if every vertex of t is bushy (with respect to t in T). An example of a tree that contains both bushy and non-bushy vertices is given in Figure 1. Now we can define the graph classes that will be examined in Section 3. Let G be a graph and $\mathcal{S}$ be a set of trees.

- G is an $\mathcal{S}$-*covered subtree overlap graph* if it has a subtree overlap representation that has a covering subtree isomorphic to a tree in $\mathcal{S}$. Such a representation is an $\mathcal{S}$-*covered subtree overlap representation* of G.
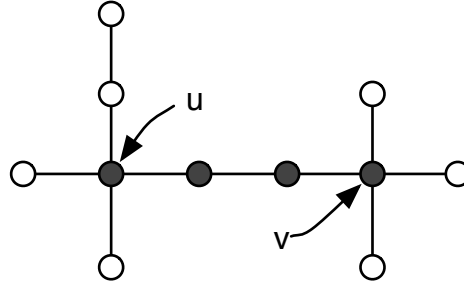
Fig. 1: A tree T and a subtree t, induced by the darker grey vertices. The vertex u is not bushy with respect to t in T because it has a neighbour (above it in the diagram) that is not in t and is not a leaf of T. Vertex v is bushy with respect to t in T because all of its neighbours that are not in t are leaves of T.

- G is a *bushy $\mathcal{S}$-covered subtree overlap graph* if it has a subtree overlap representation with a bushy covering subtree isomorphic to a tree in $\mathcal{S}$. Such a representation is a *bushy $\mathcal{S}$-covered subtree overlap representation* of G.

- G is an *$\mathcal{S}$-subtree-filament graph* if there is a subtree filament representation of G such that the host tree is isomorphic to a member of $\mathcal{S}$. Such a representation is an *$\mathcal{S}$-subtree-filament representation* of G.

- G is an *$\mathcal{S}$-cochordal graph* if it has a subtree disjointness representation such that the host tree is isomorphic to a member of $\mathcal{S}$. Such a representation is an *$\mathcal{S}$-cochordal representation* of G.

When $\mathcal{S}$ has just one element, say T, we sometimes write T instead of $\{T\}$ in the above notation.

2   Subtree representations

In this section, we give methods for transforming a given multiset of subtrees of a tree into another representation of the same type (i.e., intersection, overlap, disjointness, containment) for the same graph. Let $G = (V, E)$ be a graph and let $x \notin V$. For any given $uv \in E$, the *subdivision of edge uv (with vertex x)* is the operation of removing the edge uv from G and adding the vertex x and the edges ux and xv. The vertex x is called a *subdivision vertex*. A graph H is a *subdivision* of G if H can be obtained from G by zero or more edge subdivisions.

We first mention some simple alterations that can be made to a given tree T and multiset $\mathcal{T}$ of subtrees of T without changing the relationships among the elements of $\mathcal{T}$. First, if a new leaf is added to T but to no element of $\mathcal{T}$, then the subtrees are unchanged and therefore the relationships among them remain the same. Second, if an edge is subdivided in T and in every element of $\mathcal{T}$ that contains the edge, then the relationships among the subtrees remain unchanged. Therefore, if T and T' are trees such that T' can be obtained from T by a sequence of leaf additions and edge subdivisions, then any graph that is the intersection, overlap, containment, or disjointness graph of subtrees of T is also the intersection, overlap, containment, or disjointness (respectively) graph of subtrees of T'. Furthermore, the T-subtree-filament graphs form a subset of the T'-subtree filament graphs, and the T-covered subtree overlap graphs form a subset of the T'-covered subtree overlap graphs.

Recalling that we denote subtrees as subsets of vertices of a tree, each of which induces a subtree:

Lemma 1. *Let $T = (V_T, E_T)$ be a tree and $\mathcal{T} = \{t_1, \ldots, t_n\}$ be a multiset of subtrees of T. Let v be a vertex of T and let w be a neighbour of v in T. Let T' be the tree obtained from T by subdividing the edge $vw \in E_T$ with a vertex $x \notin V_T$. Let $\mathcal{S}$ be a (possibly empty) subset of $\mathcal{T}$, where each element of $\mathcal{S}$ contains v. For all $1 \leq i \leq n$, let*

$$t'_i = \begin{cases} t_i \cup \{x\} & \textit{if } v, w \in t_i, \textit{ or } t_i \in \mathcal{S}, \textit{ or there is an element } t_k \in \mathcal{S} \textit{ such that } t_k \subset t_i, \\ t_i & \textit{otherwise.} \end{cases}$$

Then $\{t'_1, \ldots, t'_n\}$ is a multiset of subtrees of T' and, for all $1 \leq i, j \leq n$, $(t_i, t_j) \sim (t'_i, t'_j)$.

Proof: The elements of $\{t'_1, \ldots, t'_n\}$ induce connected subgraphs of T' by the construction. Note that, for all $1 \leq i \leq n$, $t'_i = t_i \cup \{x\}$ only if $v \in t_i$, and $t'_i = t_i$ if and only if $x \notin t'_i$. Let $t_i, t_j \in \mathcal{T}$ where $1 \leq i, j \leq n$. We will prove that $(t_i, t_j) \sim (t'_i, t'_j)$ by showing that both of the following hold: $t_i \cap t_j = \emptyset$ if and only if $t'_i \cap t'_j = \emptyset$; ($t_i \setminus t_j = \emptyset$ or $t_j \setminus t_i = \emptyset$) if and only if ($t'_i \setminus t'_j = \emptyset$ or $t'_j \setminus t'_i = \emptyset$). There are three cases to be considered, based on whether x is in neither, one, or both of t'_i and t'_j.

If x is in neither t'_i nor t'_j then $t'_i = t_i$ and $t'_j = t_j$ and the result clearly follows.

If x is in both t'_i and t'_j then v is in both t_i and t_j, and therefore $t_i \cap t_j$ and $t'_i \cap t'_j$ are both nonempty. Furthermore, $t'_i \setminus t'_j = (t_i \cup \{x\}) \setminus (t_j \cup \{x\}) = t_i \setminus t_j$ and, similarly, $t'_j \setminus t'_i = t_j \setminus t_i$. So the result follows.

If x is in just one of t'_i and t'_j, then suppose without loss of generality that x is in t'_i and not in t'_j. Then $t'_i = t_i \cup \{x\}$, $t'_j = t_j$, and $v \in t_i$. So $t'_i \cap t'_j = (t_i \cup \{x\}) \cap t_j = t_i \cap t_j$. Furthermore, $t'_i \setminus t'_j = (t_i \cup \{x\}) \setminus t_j = (t_i \setminus t_j) \cup \{x\} \neq \emptyset$, and $t'_j \setminus t'_i = t_j \setminus (t_i \cup \{x\}) = t_j \setminus t_i$. If $t_j \setminus t_i = \emptyset$ then $t'_j \setminus t'_i = \emptyset$ and we have ($t_i \setminus t_j = \emptyset$ or $t_j \setminus t_i = \emptyset$) if and only if ($t'_i \setminus t'_j = \emptyset$ or $t'_j \setminus t'_i = \emptyset$), so the result follows. In the remainder of the proof, we handle the case where $t_j \setminus t_i \neq \emptyset$. Since $t'_j \setminus t'_i = t_j \setminus t_i$ we also have $t'_j \setminus t'_i \neq \emptyset$ and, from before, $t'_i \setminus t'_j \neq \emptyset$. Therefore, to complete the proof, we must show that $t_i \setminus t_j \neq \emptyset$. Suppose for contradiction that $t_i \setminus t_j = \emptyset$, that is, $t_i \subseteq t_j$, which implies that $t_i \subset t_j$ since $t_j \setminus t_i \neq \emptyset$. Since x is in t'_i, one of the following must hold: (i) $v, w \in t_i$, (ii) $t_i \in \mathcal{S}$, or (iii) there is an element $t_k \in \mathcal{S}$ such that $t_k \subset t_i$. But then, since $t_i \subset t_j$, one of the following must also hold (respectively): (i) $v, w \in t_j$, (ii) t_i is an element of $\mathcal{S}$ such that $t_i \subset t_j$, or (iii) t_k is an element of $\mathcal{S}$ such that $t_k \subset t_i \subset t_j$. In any case, this implies that $x \in t'_j$, a contradiction. Therefore $t_i \setminus t_j \neq \emptyset$ as required.

In each case, $(t_i, t_j) \sim (t'_i, t'_j)$ and therefore the proof is complete. $\qquad\square$

In order to define a subtree filament representation on subtrees $\mathcal{T}$ of a tree T, it is convenient to make some assumptions about the elements of $\mathcal{T}$. Since we are concerned with the host trees of representations, we need to consider the effect on a host tree of enforcing those assumptions. This is the subject of the next definition and lemma, which will be used in the proof of Theorem 2.

Property 1. *Subtrees $\mathcal{T}$ of tree T are said to satisfy the* nontrivial intersection distinct leaf property *if:*

- *each element of $\mathcal{T}$ is nontrivial,*

- *every pair of elements of $\mathcal{T}$ are either disjoint or share two or more vertices, and*

- *no vertex of T is a leaf of two distinct members of $\mathcal{T}$.*

Note that the third requirement in the above property guarantees that the elements of $\mathcal{T}$ are distinct.

The next lemma shows that every multiset of subtrees of a tree can be transformed into subtrees of a tree that satisfy Property 1, without altering the relationships among the subtrees. Because we are concerned with specific host trees, we consider the effect of the transformation on the host tree.

Lemma 2. *Let $T = (V_T, E_T)$ be a nontrivial tree and $\mathcal{T} = \{t_1, \ldots, t_n\}$ be a multiset of subtrees of T. There exists a tree T' and set $\mathcal{T}' = \{t'_1, \ldots, t'_n\}$ of subtrees of T' such that*

- *for all $1 \leq i, j \leq n$, $(t_i, t_j) \sim (t'_i, t'_j)$,*

- *T' is a subdivision of T, and*

- *$\mathcal{T}'$ satisfies Property 1.*

Proof: Let $T = (V_T, E_T)$ be a nontrivial tree and $\mathcal{T} = \{t_1, \ldots, t_n\}$ be a multiset of subtrees of T. We may assume that no leaf of T is contained in any element of $\mathcal{T}$. Otherwise, for each leaf ℓ of T that is contained in an element of $\mathcal{T}$, we could add a new leaf to T adjacent to ℓ. Then T would be isomorphic to a subdivision of the original tree and would satisfy the assumption. We show how to transform T and $\mathcal{T}$ into T' and $\mathcal{T}'$, respectively, such that the conditions of the lemma are satisfied.

First, we perform $2|E_T|$ applications of the transformation of Lemma 1, as follows. The initial application is performed on T and $\mathcal{T}$ with v and w being the endpoints of an edge of T and $\mathcal{S}$ being the set of all elements of $\mathcal{T}$ that contain v. By Lemma 1, this transformation results in a tree and a multiset of subtrees of the tree. Each subsequent application is performed on the tree and subtrees resulting from the previous step, and again produces a tree and a multiset of subtrees. In the following description, we refer to the current tree and subtrees as T_c and $\mathcal{T}_c$ respectively. Overall, for each edge pq of the initial tree T, we apply the transformation of Lemma 1 twice, once with $v = p$, $w = q$, and $\mathcal{S}$ being all elements of $\mathcal{T}_c$ that contain p; then with $v = q$, $w = x$ where x is the subdivision vertex from the previous step, and $\mathcal{S}$ being all elements of $\mathcal{T}_c$ that contain q. By Lemma 1, this process finally results in a tree T_2 and subtrees $\mathcal{T}_2 = \{t_1^2, \ldots, t_n^2\}$ of T_2, with $(t_i, t_j) \sim (t_i^2, t_j^2)$ for all $1 \leq i, j \leq n$. Clearly, T_2 is a subdivision of T.

Let t_i be a trivial element of $\mathcal{T}$ and let p be the single vertex of t_i. Since T is nontrivial, p is incident with an edge of T, and the new vertex introduced in the subdivision of that edge with $v = p$ is added to t_i in the construction of t_i^2. Therefore each element of $\mathcal{T}_2$ is nontrivial.

Suppose that distinct subtrees $t_i, t_j \in \mathcal{T}$ intersect in just one vertex, p. Since T is nontrivial, p is incident with an edge of T. The new vertex that subdivides that edge is in both t_i^2 and t_j^2. Therefore every pair of elements of $\mathcal{T}_2$ are either disjoint or share an edge.

Since no leaf of T is contained in an element of $\mathcal{T}$, the leaves of members of $\mathcal{T}_2$ are all vertices of T_2 that are not in V_T. Therefore, every vertex that is a leaf of a subtree of $\mathcal{T}_2$ has degree two in T_2. Furthermore, if two elements of $\mathcal{T}_2$ share a leaf, say p, then they both contain the neighbour of p that played the role of v during the subdivision when p was introduced, and not the other neighbour as that would contradict p being a leaf of both subtrees.

To complete the proof, we show how to reduce the number of vertices of T_2 that are leaves of two or more distinct elements of $\mathcal{T}_2$. Applied iteratively, this leads to a representation that satisfies all conditions of the lemma. Let p be a node in T_2 and let $\mathcal{T}_p$ be the elements of $\mathcal{T}_2$ that contain p as a leaf. Let $n_p = |\mathcal{T}_p|$ and suppose that $n_p \geq 2$. By the observation of the preceding paragraph, p has degree two in T_2 and every element of $\mathcal{T}_p$ also contains one of p's neighbours and not the other. Let q and r be the neighbours of p, such that every element of $\mathcal{T}_p$ contains q and not r. Let the elements of $\mathcal{T}_p$ be sorted by nondecreasing size so that each element of $\mathcal{T}_p$ has a position from 1 to n_p in the sorted list.

We now use n_p new vertices, $s_1, \ldots, s_{n_p}$, which are not vertices of T_2, as subdivision vertices in n_p applications of Lemma 1. The first application is performed on T_2 and $\mathcal{T}_2$, and produces a tree and subtrees of a tree. Each subsequent application is performed on the tree and subtrees resulting from the previous step. We will refer to the current tree and subtrees as T_c and $\mathcal{T}_c$, respectively, and the current subtrees corresponding to those of $\mathcal{T}_p$ as $\mathcal{T}_{pc}$. First, apply the transformation of Lemma 1 to T_2 and $\mathcal{T}_2$ with $v = p$, $w = r$, and $\mathcal{S}$ consisting of the single element of $\mathcal{T}_p$ having position n_p in the sorted list, using the vertex s_{n_p} as the subdivision vertex. Then, for each ℓ from $n_p - 1$ down to 1, apply the transformation of Lemma 1 to T_c and $\mathcal{T}_c$ with $v = p$ and $w = s_{\ell+1}$ (the subdivision vertex from the previous step), with $\mathcal{S}$ being the set of all elements of $\mathcal{T}_{pc}$ with corresponding elements in the sorted list of elements of $\mathcal{T}_p$ having positions greater than or equal to ℓ, and using subdivision vertex s_ℓ. By Lemma 1, this process terminates with a tree T_3 and subtrees $\mathcal{T}_3$ of T_3.

By repeated application of Lemma 1, $\mathcal{T}_3$ is a multiset of subtrees of T_3 and $(t_i^2, t_j^2) \sim (t_i^3, t_j^3)$ for all $1 \leq i, j \leq n$. In addition, T_3 is a subdivision of T_2 and therefore of T. Since $t_i^2 \subseteq t_i^3$ for all $1 \leq i \leq n$, each t_i^3 has at least two vertices and each pair of intersecting subtrees of $\mathcal{T}_3$ shares an edge. No vertex has had its degree increased, and only degree two vertices have been added; therefore, every vertex of T_3 that is a leaf of any subtree of $\mathcal{T}_3$ has degree two in T_3. By the construction of $\mathcal{T}_2$, each element t_i^2 of $\mathcal{T}_2 \backslash \mathcal{T}_p$ contains all or none of p, q, and r, and therefore t_i^3 either contains all of the s_ℓ's and r, or none of the s_ℓ's, and therefore does not contain any of the s_ℓ's as a leaf. Each $t_i^3 \in \mathcal{T}_3$ such that $t_i^2 \in \mathcal{T}_p$ has exactly one of the s_ℓ's as a leaf, namely, s_ℓ where ℓ is the position of t_i^2 in the sorted list of the elements of $\mathcal{T}_p$. Therefore, p is not a leaf of any element of $\mathcal{T}_3$ and each new vertex s_ℓ of T_3 is a leaf of just one element of $\mathcal{T}_3$, namely, the subtree corresponding to the element of $\mathcal{T}_p$ in position ℓ of the sorted list. Thus the number of vertices of T_3 that are leaves of two or more distinct elements of $\mathcal{T}_3$ is less than than the number of vertices of T_2 that are leaves of two or more distinct elements of $\mathcal{T}_2$. Applied iteratively, this method eventually produces subtrees $\mathcal{T}'$ of a tree T' that satisfy the lemma. □

3　Equivalence of $\mathcal{S}$-covered subtree overlap graphs and $\mathcal{S}$-subtree-filament graphs

In this section, we show that for any tree R, every R-covered subtree overlap graph has a representation in which the host tree is just R with some additional leaves, and that R-covered subtree overlap graphs are equivalent to the complements of R-cochordal-mixed graphs. This equivalence does not extend to R-subtree filament graphs since, for example, C_4, the cycle on four vertices, is a K_2-covered subtree overlap graph but not a K_2-subtree-filament graph. However, the equivalence does extend to subtree filament graphs when edge subdivision is allowed. In Theorem 2 we show the equivalence of $\mathcal{S}$-covered subtree overlap graphs, $\mathcal{S}$-subtree-filament graphs, and complements of $\mathcal{S}$-cochordal-mixed graphs, when $\mathcal{S}$ is a nontrivial set of trees that is closed under edge subdivision.

Theorem 1. *Let R be a tree and let G be a graph. The following statements are equivalent:*

1. *G is an R-covered subtree overlap graph.*

2. *G is the complement of an R-cochordal-mixed graph.*

3. *G is a bushy R-covered subtree overlap graph.*

Proof:

1 ⇒ 2: Let $\mathcal{T} = \{t_1, \ldots, t_n\}$ be a subtree overlap representation for G in tree T with covering subtree R, and suppose that the elements of $\mathcal{T}$ are indexed such that $i < j$ implies $|t_i| \leq |t_j|$. Then the sets $E_1 = \{v_i v_j \mid t_i | t_j\}$ and $\overrightarrow{E_2} = \{v_i \rightarrow v_j \mid (t_i \subseteq t_j) \text{ and } i < j\}$ define a cochordal-mixed partition of the edges of $\overline{G}$. Since R is a covering subtree of $\mathcal{T}$, for all $1 \leq i, j \leq n$, $t_i \cap R$ is a subtree of R and $t_i \cap t_j \cap R = \emptyset$ if and only if $t_i \cap t_j = \emptyset$. Therefore, subtrees $\{t_i \cap R \mid t_i \in \mathcal{T}\}$ of R form an R-cochordal representation of the graph (V, E_1).

2 ⇒ 3: This part of the proof combines elements of the proof that complements of cointerval-mixed graphs are interval filament graphs [8] and the proof that subtree filament graphs are subtree overlap graphs [6]. Let $G = (V, E)$ be the complement of an R-cochordal-mixed graph. Let E_1 and E_2 be a partition of the edges of $\overline{G}$ such that (V, E_1) is an R-cochordal graph and $(V, \overrightarrow{E_2})$ is a transitive orientation of (V, E_2) such that for all $1 \leq i, j, k \leq n$, if $(v_i \rightarrow v_j) \in \overrightarrow{E_2}$ and $v_j v_k \in E_1$ then $v_i v_k \in E_1$. Let subtrees $\mathcal{T} = \{t_1, \ldots, t_n\}$ of tree $R = (V_R, E_R)$ be an R-cochordal representation of (V, E_1), that is, for all $1 \leq i, j \leq n$, $v_i v_j \in E_1$ if and only if $t_i | t_j$.

Suppose that $(v_i \rightarrow v_j) \in \overrightarrow{E_2}$. Then $t_i \cap t_j \neq \emptyset$. Furthermore, if $t_i \not\subseteq t_j$, then replacing t_i with $t_i \cap t_j$ produces another R-cochordal representation of (V, E_1), as justified by the following argument from [8]. Suppose that $t_i \not\subseteq t_j$. By the definition of $(V, \overrightarrow{E_2})$, for all $1 \leq k \leq n$, if $v_i v_k \notin E_1$ then $v_j v_k \notin E_1$. Equivalently, every $t_k \in \mathcal{T}$ that intersects t_i also intersects t_j (as well as $t_i \cap t_j$ since t_i, t_j, and t_k are all subtrees of a tree). Therefore, replacing t_i with $t_i \cap t_j$ produces another R-cochordal representation of (V, E_1). Applied repeatedly, this transformation results in an R-cochordal representation of (V, E_1) such that for all $1 \leq i, j \leq n$, $(v_i \rightarrow v_j) \in \overrightarrow{E_2}$ implies $t_i \subseteq t_j$.

Let T' be the tree R with n additional nodes: $x_1, \ldots, x_n \notin V_R$ where, for each $1 \leq i \leq n$, x_i is adjacent in T' to exactly one arbitrary node of $t_i \cap R$. Then, for $1 \leq i \leq n$, let $t_i' = t_i \cup \{x_i\} \cup \{x_k \mid (v_k \rightarrow v_i) \in \overrightarrow{E_2}\}$ and let $\mathcal{T}' = \{t_1', \ldots, t_n'\}$. Each element of $\mathcal{T}'$ induces a connected subgraph and therefore a subtree of T'. Suppose there are two elements of $\mathcal{T}'$, t_i' and t_j' such that $i \neq j$ and $t_i' = t_j'$. Then x_i is in t_j' and x_j is in t_i' which implies that $(v_i \rightarrow v_j)$ and $(v_j \rightarrow v_i)$ are both in $\overrightarrow{E_2}$. But this contradicts transitivity since $\overline{G}$ is a simple graph. Therefore, the elements of $\mathcal{T}'$ are distinct.

For each $1 \leq i, j \leq n$, $i \neq j$, $v_i v_j$ is in exactly one of E_1, E_2, or E. Since $\mathcal{T}$ is a cochordal representation of (V, E_1), $v_i v_j \in E_1$ if and only if $t_i | t_j$. Of the nodes $x_1, \ldots, x_n$, only x_i (respectively x_j) and those corresponding to subtrees contained in or equal to t_i (respectively t_j) are in t_i' (respectively t_j'). Furthermore, $t_i \subset t_i'$ and $t_j \subset t_j'$. Therefore $t_i | t_j$ if and only if $t_i' | t_j'$ and so $v_i v_j \in E_1$ if and only if $t_i' | t_j'$. If $v_i v_j \in E_2$ then suppose without loss of generality that $(v_i \rightarrow v_j) \in \overrightarrow{E_2}$. Then, by our earlier argument, $t_i \subseteq t_j$. In addition, every vertex of $t_i' \backslash t_i$ is also in t_j' by transitivity of $\overrightarrow{E_2}$, and $x_j \in t_j' \backslash t_i'$. Therefore, if $(v_i \rightarrow v_j) \in \overrightarrow{E_2}$ then $t_i' \subset t_j'$, and if $v_i v_j \in E_2$ then $t_i' \subset t_j'$ or $t_j' \subset t_i'$. Finally, if $v_i v_j \in E$ then, since $\mathcal{T}$ is a cochordal representation of (V, E_1), $t_i \cap t_j \neq \emptyset$ and therefore $t_i' \cap t_j' \neq \emptyset$. In addition, $x_i \in t_i' \backslash t_j'$ and $x_j \in t_j' \backslash t_i'$. Thus, $v_i v_j \in E$ implies $t_i' \between t_j'$. We conclude that the subtrees $\mathcal{T}'$ of tree T' form a subtree overlap representation of G.

Furthermore, since for each $1 \leq i \leq n$, $t_i \subset t_i'$ and each x_i is adjacent to a vertex of R, subtrees $\mathcal{T}'$ of tree T' form an R-covered subtree overlap representation of G in which R is bushy.

3 ⇒ 1: Obvious. □

The classes of K_1-covered subtree overlap graphs and bushy K_1-covered subtree overlap graphs are

equivalent to the class of cocomparability graphs. This follows from [12] combined with the observation that subtrees of a tree that all have a vertex in common overlap if and only if neither is contained in the other. Thus, Theorem 1 generalises characterisations of cocomparability graphs as the overlap graphs of subtrees of a tree where all subtrees have a vertex in common, the complements of cochordal-mixed graphs where all edges are in the E_2 block of the partition, and the overlap graphs of subtrees of a star [2, 8, 12].

Theorem 2. *Let G be a graph and $\mathcal{S} \neq \{K_1\}$ be a nonempty set of trees that is closed under edge subdivision. The following statements are equivalent:*

1. *G is an $\mathcal{S}$-covered subtree overlap graph.*

2. *G is the complement of an $\mathcal{S}$-cochordal-mixed graph.*

3. *G is a bushy $\mathcal{S}$-covered subtree overlap graph.*

4. *G is an $\mathcal{S}$-subtree-filament graph.*

Proof:
1 $\Leftrightarrow$ 2 $\Leftrightarrow$ 3: by Theorem 1.

4 $\Leftrightarrow$ 2: By Theorem 4 of [8], a graph is a subtree filament graph if and only if it is the complement of a cochordal-mixed graph. In the proof of that theorem, a subtree-filament representation of a graph $G = (V, E)$ is transformed to a cochordal representation of the graph (V, E_1) on the same host tree, where E_1 and E_2 is a cochordal-mixed partition of the edges of $\overline{G}$. Thus 4 implies 2.

For the other direction, suppose that G is the complement of an R-cochordal-mixed graph where $R \in \mathcal{S}$, and let E_1 and E_2 be a cochordal-mixed partition of the edges of $\overline{G}$. By Lemma 2, there is a T-cochordal representation of (V, E_1) that satisfies Property 1, such that T is a subdivision of R. The proof of Theorem 4 of [8] transforms a T-cochordal representation of (V, E_1) and cochordal-mixed partition E_1 and E_2 of the edges of $\overline{G}$ to a T-subtree-filament representation of G, provided that the T-cochordal representation of (V, E_1) satisfies Property 1. Therefore, since $T \in \mathcal{S}$, we have 2 implies 4. $\qquad\square$

Theorem 2 does not hold for $\mathcal{S} = \{K_1\}$ since only complete graphs are K_1-subtree-filament graphs while, as previously noted, the classes of K_1-covered subtree overlap graphs, K_1-cochordal-mixed graphs, and bushy K_1-covered subtree overlap graphs are all equivalent to cocomparability graphs.

When $\mathcal{S}$ is the set of subdivisions of K_2, Theorem 2 becomes the following characterisation of interval filament graphs, which includes results of [3] and [8].

Corollary 1. *The following statements are equivalent for a graph G: G is a path-covered subtree overlap graph; G is the complement of a cointerval-mixed graph; G is the overlap graph of subtrees of a caterpillar; G is an interval filament graph.*

4   Conclusion

We have presented two main results:

1. The following graph classes are equivalent for any tree R: R-covered subtree overlap graphs, the complements of R-cochordal-mixed graphs, and bushy R-covered subtree overlap graphs.

   2. The following graph classes are equivalent for any nonempty set of trees $\mathcal{S} \neq \{K_1\}$ that is closed under edge subdivision: $\mathcal{S}$-covered subtree overlap graphs, the complements of $\mathcal{S}$-cochordal-mixed graphs, bushy $\mathcal{S}$-covered subtree overlap graphs, and $\mathcal{S}$-subtree-filament graphs.

The first result is a generalization of characterisations of cocomparability graphs, as can be seen in the simplest case of Theorem 1, when $R = K_1$. The second result generalises characterisations of interval filament graphs. The simplest case of Theorem 2, when $\mathcal{S}$ is the set of subdivisions of K_2, states that the following graph classes are equivalent: path-covered subtree overlap graphs, the complements of cointerval-mixed graphs, the overlap graphs of subtrees of caterpillars, and interval filament graphs. The second result suggests that the $\mathcal{S}$-covered subtree overlap graphs, for sets $\mathcal{S}$ of trees closed under edge subdivision, might be a useful way of breaking down the class of subtree overlap graphs. We propose three avenues based on that idea for future study.

While some subclasses of subtree overlap graphs can be recognised in polynomial time (including interval, permutation, cocomparability, chordal, circular arc, and circle graphs), for others the recognition problem is NP-complete (including interval filament graphs [16], overlap graphs of subtrees of a tree with a bounded number of leaves, the overlap graphs of subtrees of subdivisions of a fixed tree with at least three leaves, and the overlap graphs of paths in a tree with bounded maximum degree [5, 17]). The complexity of the recognition problem for subtree overlap graphs is open. An efficient recognition algorithm that could output subtree overlap representations for yes instances would have significant algorithmic implications since several optimisation problems that are NP-complete in general can be solved efficiently for subtree overlap graphs when a subtree overlap representation is given [2, 9, 10, 11, 13]. Does the recognition problem on $\mathcal{S}$-covered subtree overlap graphs give insight into the recognition problem on subtree overlap graphs as a whole?

Several optimisation problems remain NP-hard on subtree overlap graphs by virtue of hardness results on the subclasses. It would be interesting to explore the possible P vs. NP-complete boundaries for various optimisation problems within the $\mathcal{S}$-covered subtree overlap graphs over sets $\mathcal{S}$ of trees that are closed under edge subdivision.

Parameters of chordal graphs based on their subtree intersection representations include leafage, that is, the minimum number of leaves in the host tree of a representation [14], and vertex leafage, that is, the minimum maximum number of leaves of a subtree in a representation [4]. How do analogous and other parameters of subtree overlap graphs relate to the $\mathcal{S}$-covered subtree overlap graph classes of this paper?

5 Acknowledgements

References

[1] Andreas Brandstädt, Van Bang Le, Jeremy P. Spinrad, Graph Classes: A Survey, SIAM Monographs on Discrete Mathematics and Applications, Volume 3, SIAM, 1999.

[2] Eowyn Čenek, Lorna Stewart, Maximum independent set and maximum clique algorithms for overlap graphs, Discrete Applied Mathematics 131 (2003) 77-91.

[3] Jérémie Chalopin, Daniel Gonçalves, and Pascal Ochem, On graph classes defined by overlap and intersection models, Paper presented at Sixth Czech-Slovak International Symposium on Combinatorics, Graph Theory, Algorithms and Applications, Prague, July 10-15, 2006.

[4] Steven Chaplick, Juraj Stacho, The vertex leafage of chordal graphs, Discrete Applied Mathematics 168 (2014) 14-25.

[5] Jessica Enright, Results on Set Representations of Graphs, University of Alberta PhD Thesis, 2011.

[6] Jessica Enright, Lorna Stewart, Subtree filament graphs are subtree overlap graphs, Information Processing Letters 104 (2007) 228-232.

[7] Fanica Gavril, Algorithms for a maximum clique and a maximum independent set of a circle graph, Networks 3 (1973) 261-273.

[8] Fanica Gavril, Maximum weight independent sets and cliques in intersection graphs of filaments, Information Processing Letters 73 (2000) 181-188.

[9] Fanica Gavril, 3D-interval-filament graphs, Discrete Applied Mathematics 155(18) (2007) 2625-2636.

[10] Fanica Gavril, Algorithms on subtree filament graphs, in Marina Lipshteyn, Vadim Efimovich Levit, Ross M. McConnell (Eds.): Graph theory, computational intelligence and thought: essays dedicated to Martin Charles Golumbic on the occasion of his 60th Birthday, LNCS 5420 (2009) 27-35.

[11] Fanica Gavril, Algorithms for induced biclique optimization problems, Information Processing Letters 111(10) (2011) 469-473.

[12] Martin C. Golumbic, Edward R. Scheinerman, Containment graphs, posets, and related classes of graphs, in Combinatorial Mathematics: Proceedings of the Third International Conference, Annals of the New York Academy of Sciences 555 (1989) 192-204.

[13] J. Mark Keil, Lorna Stewart, Approximating the minimum clique cover and other hard problems in subtree filament graphs, Discrete Applied Mathematics 154 (2006) 1983-1995.

[14] In-Jen Lin, Terry A. McKee, Douglas B. West, The leafage of a chordal graph, Discussiones Mathematicae Graph Theory 18 (1998) 23-48.

[15] Edward Marczewski, Sur deux propriétés des classes d'ensembles, Fundamenta Mathematicae 33 (1945) 303-307.

[16] Martin Pergel, Recognition of polygon-circle graphs and graphs of interval filaments is NP-complete, in Andreas Brandstädt, Dieter Kratsch, and Haiko Müller (Eds.): WG 2007, LNCS 4769 (2007) 238-247.

[17] Martin Pergel, Special Graph Classes and Algorithms on Them, Charles University in Prague PhD Thesis, 2008.