Applied Mathematics Letters 24 (2011) 719–723

Contents lists available at ScienceDirect

Applied Mathematics Letters

journal homepage: www.elsevier.com/locate/aml

Graph triangulations and the compatibility of unrooted phylogenetic trees

Sudheer Vakati *, David Fernández-Baca

Department of Computer Science, Iowa State University, Ames, IA 50011, USA

ARTICLE INFO

Article history:
Received 23 April 2010
Received in revised form 16 December 2010
Accepted 17 December 2010

Keywords:
Compatibility
Chordal graphs
Graph triangulation
Phylogenetics
Supertrees
Tree decompositions

ABSTRACT

We characterize the compatibility of a collection of unrooted phylogenetic trees as a question of determining whether a graph derived from these trees — the display graph — has a specific kind of triangulation, which we call legal. Our result is a counterpart to the well-known triangulation-based characterization of the compatibility of undirected multi-state characters.

1. Introduction

A *phylogenetic tree* or *phylogeny* is an unrooted tree T whose leaves are in one-to-one correspondence with a set of *labels* (*taxa*) $\mathcal{L}(T)$. If $\mathcal{L}(T) = X$, we say that T is a *phylogenetic tree for* X, or a *phylogenetic* X*-tree* [1]. A phylogenetic tree represents the evolutionary history of a set of species, which are the labels of the tree.

Suppose T is a phylogenetic tree. Given a subset $Y \subseteq \mathcal{L}(T)$, the *subtree of* T *induced by* Y, denoted $T|Y$, is the tree obtained by forming the minimal subgraph of T connecting the leaves with labels in Y and then suppressing vertices of degree two. Let T' be some other phylogenetic tree such that $\mathcal{L}(T') \subseteq \mathcal{L}(T)$. We say that T *displays* T' if T' can be obtained by contracting edges in the subtree of T induced by $\mathcal{L}(T')$.

A *profile* is a tuple $\mathcal{P} = (T_1, T_2, \ldots, T_k)$, where each T_i is a phylogenetic tree for some set of labels $\mathcal{L}(T_i)$. The T_is are called *input trees*, and we may have $\mathcal{L}(T_i) \cap \mathcal{L}(T_j) \neq \emptyset$ for $i \neq j$. A *supertree* for $\mathcal{P}$ is a phylogeny T with $\mathcal{L}(T) = \bigcup_{i=1}^{k} \mathcal{L}(T_i)$. Profile $\mathcal{P}$ is *compatible* if there exists a supertree T for $\mathcal{P}$ that displays T_i, for each $i \in \{1, \ldots, k\}$. The *phylogenetic tree compatibility problem* asks, given a profile $\mathcal{P}$, whether or not $\mathcal{P}$ is compatible. This question arises when trying to assemble a collection of phylogenies for different sets of species into a single phylogeny (a supertree) for all the species [2]. The phylogenetic tree compatibility problem asks whether or not it is possible to do so via a supertree that displays each of the input trees.

Phylogenetic tree compatibility is NP-complete [3] (but the problem is polynomially solvable for rooted trees [4]). Nevertheless, Bryant and Lagergren have shown that the problem is fixed-parameter tractable for fixed k [5]. Their argument relies on a partial characterization of compatibility in terms of tree decompositions and tree-width of a structure that they call the "display graph" of a profile (this graph is defined in Section 3). Here we build on their argument to produce a complete characterization of compatibility in terms of the existence of a special kind of triangulation of the display graph. These *legal* triangulations (defined in Section 3) only allow certain kinds of edges to be added. Our result is a counterpart to the well-known characterization of character compatibility in terms of triangulations of a class of intersection graphs [6],

* Corresponding author.
E-mail addresses: svakati@iastate.edu (S. Vakati), fernande@cs.iastate.edu (D. Fernández-Baca).

0893-9659/$ – see front matter
doi:10.1016/j.aml.2010.12.015

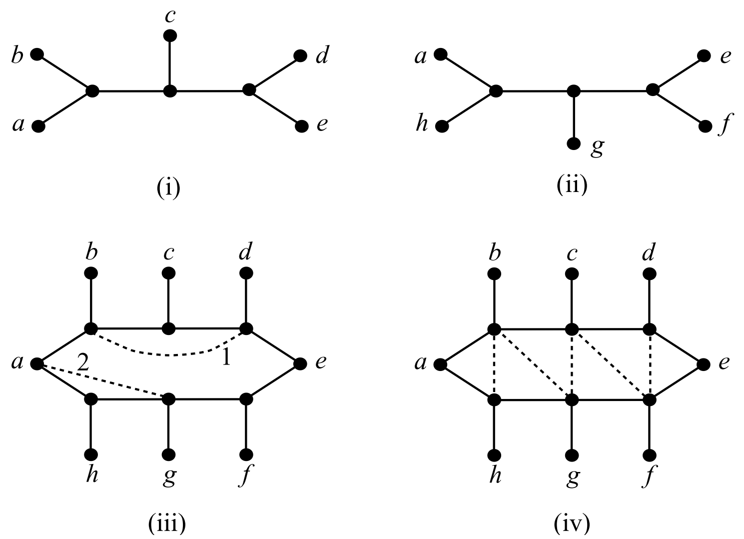

Fig. 1. (i) First input tree. (ii) Second input tree. (iii) The display graph of the input tress with two fill-in edges, indicated by dashed lines. Edge 1 cannot appear in a legal triangulation, since the result would violate (LT1). Edge 2 is not allowed, because it would result in a violation of (LT2). (iv) The display graph with a legal triangulation, indicated by dashed lines.

which has algorithmic consequences [7,8]. Our characterization of tree compatibility may have analogous implications. A different characterization of the phylogenetic tree compatibility problem in terms of a structure called the "quartet graph" is given in [9].

2. Preliminaries

Let G be a graph. We write $V(G)$ and $E(G)$ to denote the vertex set and edge set of G, respectively. Suppose C is a cycle in G. A *chord* in C is any edge of G whose endpoints are two nodes that are not adjacent in C. G is said to be *chordal* if and only if every cycle of length at least four has a chord. A graph G' is a *chordal fill-in* or *triangulation* of G if $V(G') = V(G)$, $E(G') \supseteq E(G)$, and G' is chordal. The set $E(G') \setminus E(G)$ is called a *fill-in* for G and the edges in it are called *fill-in edges*.

A *tree decomposition* for a graph G is a pair (T, B), where T is a tree and B is a mapping from $V(T)$ to subsets of $V(G)$ that satisfies the following three properties.

(TD1) (*Vertex Coverage*) For every $v \in V(G)$ there is an $x \in V(T)$ such that $v \in B(x)$.
(TD2) (*Edge Coverage*) For every edge $\{u, v\} \in E(G)$ there exists an $x \in V(T)$ such that $\{u, v\} \subseteq B(x)$.
(TD3) (*Coherence*) For every $u \in V(G)$ the set of vertices $\{x \in V(T) : u \in B(x)\}$ forms a subtree of T.

It is well known that if G is chordal, G has a tree decomposition (T, B) where (i) there is a one-to-one mapping C from the vertices of T to the maximal cliques of G and (ii) for each vertex x in T, $B(x)$ consists precisely of the vertices in the clique $C(x)$ [10]. This sort of tree decomposition is called a *clique tree* for G. Conversely, let (T, B) be a tree decomposition of a graph G and let F be the set of all $\{u, v\} \notin E(G)$ such that $\{u, v\} \subseteq B(x)$ for some $x \in V(T)$. Then, F is a chordal fill-in for G [10]. We shall refer to this set F as the *chordal fill-in of G associated with tree decomposition* (T, B) and to the graph G' obtained by adding the edges of F to G as the *triangulation of G associated with* (T, B).

3. Legal triangulations and compatibility

The *display graph* of a profile $\mathcal{P} = (T_1, \ldots, T_k)$ is the graph $G = G(\mathcal{P})$ formed from the disjoint graph union of $T_1, \ldots, T_k$ by identifying the leaves with common labels. An example of display graph is given in Fig. 1 (see also Fig. 1 of [5]). An edge e of G is *internal* if, in the input tree where it originated, both endpoints of e were internal vertices; otherwise, e is *non-internal*. A vertex v of G is called a *leaf* if it was obtained by identifying input tree leaf nodes with the same label ℓ. The label of v is ℓ. A non-leaf vertex of G is said to be *internal*.

A triangulation G' of the display graph G is *legal* if it satisfies the following conditions.

(LT1) Suppose a clique in G' contains an internal edge. Then, this clique can contain no other edge from G (internal or non-internal).
(LT2) Fill-in edges can only have internal vertices as their endpoints.

Note that the above conditions rule out a chord between vertices of the same tree. Also, in any legal triangulation of G, any clique that contains a non-internal edge cannot contain an internal edge from any tree. See Fig. 1.

The importance of legal triangulations derives from the next results, which are proved in the next section.

Lemma 1. *Suppose a profile $\mathcal{P} = (T_1, \dots, T_k)$ of unrooted phylogenetic trees is compatible. Then the display graph of $\mathcal{P}$ has a legal triangulation.*

Lemma 2. *Suppose the display graph of a profile $\mathcal{P} = (T_1, \dots, T_n)$ of unrooted trees has a legal triangulation. Then $\mathcal{P}$ is compatible.*

The preceding lemmas immediately imply our main result.

Theorem 1. *A profile $\mathcal{P} = (T_1, \dots, T_k)$ of unrooted trees is compatible if and only if the display graph of $\mathcal{P}$ has a legal triangulation.*

4. Proofs

The proofs of Lemmas 1 and 2 rely on a new concept. Suppose T_1 and T_2 are phylogenetic trees such that $\mathcal{L}(T_2) \subseteq \mathcal{L}(T_1)$. An *embedding function from T_1 to T_2* is a surjective map ϕ from a subgraph of T_1 to T_2 satisfying the following properties.

(EF1) For every $\ell \in \mathcal{L}(T_2)$, ϕ maps the leaf labeled ℓ in T_1 to the leaf labeled ℓ in T_2.
(EF2) For every vertex v of T_2 the set $\phi^{-1}(v)$ is a connected subgraph of T_1.
(EF3) For every edge $\{u, v\}$ of T_2 there is a unique edge $\{u', v'\}$ in T_1 such that $\phi(u') = u$ and $\phi(v') = v$.

The next result extends Lemma 1 of [5].

Lemma 3. *Let T_1 and T_2 be phylogenetic trees and $\mathcal{L}(T_2) \subseteq \mathcal{L}(T_1)$. Tree T_1 displays tree T_2 if and only if there exists an embedding function ϕ from T_1 to T_2.*

Proof. The "only if" part was already observed by Bryant and Lagergren (see Lemma 1 of [5]). We now prove the other direction.

To prove that T_1 displays T_2, we argue that T_2 can be obtained from $T_1|\mathcal{L}(T_2)$ by a series of edge contractions, which are determined by the embedding function ϕ from T_1 to T_2. Let T_1' be the graph obtained from $T_1|\mathcal{L}(T_2)$ by considering each vertex v of T_2 and identifying all vertices of $\phi^{-1}(v)$ in $T_1|\mathcal{L}(T_2)$ to obtain a single vertex u' with $\phi(u') = v$. Property (EF2) ensures that, each such operation is well defined and yields a tree. By properties (EF1)–(EF3), each vertex v of $T_1|\mathcal{L}(T_2)$ is in the domain of ϕ. Thus, function ϕ is now a bijection between T_2 and T_1' that satisfies (EF1)–(EF3). We now prove that T_1' is isomorphic to T_2. It then follows from property (EF1) that T_1 displays T_2.

We claim that for any two vertices $u, v \in V(T_2)$, there is an edge $\{u, v\} \in E(T_2)$ if and only if there is an edge $\{\phi^{-1}(u), \phi^{-1}(v)\} \in E(T_1')$. The "only if" part follows from property (EF3). For the other direction, assume by way of contradiction that $\{x, y\} \notin E(T_2)$, but that $\{\phi^{-1}(x), \phi^{-1}(y)\} \in E(T_1')$. Let P be the path between vertices x and y in T_2. By property (EF3), there is a path between nodes $\phi^{-1}(x), \phi^{-1}(y)$ in tree T_1' that does not include the edge $\{\phi^{-1}(x), \phi^{-1}(y)\}$. This path along with the edge $\{\phi^{-1}(x), \phi^{-1}(y)\}$ forms a cycle in T_1', which gives the desired contradiction. Thus, the bijection ϕ between T_2 and T_1' is an isomorphism between the two trees. □

The preceding lemma immediately implies the following characterization of compatibility.

Lemma 4. *Profile $\mathcal{P} = (T_1, \dots, T_k)$ is compatible if and only if there exist a supertree T for $\mathcal{P}$ and functions $\phi_1, \dots, \phi_k$, where, for $i = 1, \dots, k$, ϕ_i is an embedding function from T to T_i.*

Proof of Lemma 1. If $\mathcal{P}$ is compatible, there exists a supertree for $\mathcal{P}$ that displays T_i for $i = 1, \dots, k$. Let T be any such supertree. By Lemma 4, for $i = 1, \dots, k$, there exists an embedding function ϕ_i from T to T_i. We will use T and the ϕ_is to build a tree decomposition (T_G, B) corresponding to a legal triangulation G' of the display graph G of $\mathcal{P}$. The construction closely follows that given by Bryant and Lagergren in their proof of Theorem 1 of [5]; thus, we only summarize the main ideas.

Initially we set $T_G = T$ and, for every $v \in V(T)$, $B(v) = \{\phi_i(v) : v \text{ in the domain of } \phi_i;\ 1 \le i \le k\}$. Now, (T_G, B) satisfies the vertex coverage property and the coherence property, but not edge coverage [5]. To obtain a pair (T_G, B) that satisfies all three properties, subdivide the edges of T_G and extend B to the new vertices. Do the following for each edge $\{x, y\}$ of T_G. Let $F = \{\{u_1, v_1\}, \dots, \{u_m, v_m\}\}$ be the set of edges of G such that $u_i \in B(x)$ and $v_i \in B(y)$. Observe that F contains at most one edge from T_i, for $i = 1, \dots, k$ (thus, $m \le k$). Replace edge $\{x, y\}$ by a path $x, z_1, \dots, z_m, y$, where $z_1, \dots, z_m$ are new vertices. For $i = 1, 2, \dots, m$, let $B(z_i) = (B(x) \cap B(y)) \cup \{v_1, \dots, v_i, u_i, \dots, u_m\}$. The resulting pair (T_G, B) can be shown to be a tree decomposition of G of width k (see [5]).

The preceding construction guarantees that (T_G, B) satisfies two additional properties:

(i) For any $x \in V(T_G)$, if $B(x)$ contains both endpoints of an internal edge of T_i, for some i, then $B(x)$ cannot contain both endpoints of any other edge, internal or not.
(ii) Let $x \in V(T_G)$ be such that $B(x)$ contains a labeled vertex $v \in V(G)$. Then, for every $u \in B(x) \setminus \{v\}$, $\{v, u\} \in E(G)$.

Properties (i) and (ii) imply that the triangulation of G associated with (T_G, B) is legal. □

Next, we prove Lemma 2. For this, we need some definitions and auxiliary results. Assume that the display graph of profile $\mathcal{P}$ has a legal triangulation G'. Let (T', B) be a clique tree for G'. For each vertex $v \in V(G)$, let $C(v)$ denote the set of all nodes in the clique tree T' that contain v. Observe that the coherence property implies that $C(v)$ induces a subtree of T'.

Lemma 5. *Suppose vertex v is a leaf in tree T_i, for some $i \in \{1, \ldots, k\}$. Let $U(v) = \bigcup_{x \in C(v)} B(x)$. Then, for any $j \in \{1, \ldots, k\}$, at most one internal vertex u from input tree T_j is present in $U(v)$. Furthermore, for any such vertex u we must have that $\{u, v\} \in E(G)$.*

Proof. Follows from condition (LT2). □

Lemma 6. *Suppose $e = \{u, v\}$ is an internal edge from input tree T_i, for some $i \in \{1, \ldots, k\}$. Let $U(e) = \bigcup_{x \in C(u) \cap C(v)} B(x)$. Then,*

(i) *$U(e)$ contains at most one vertex of T_j, for any $j \in \{1, \ldots, k\}, j \neq i$, and*
(ii) *$V(T_i) \cap U(e) = \{u, v\}$.*

Proof. Part (ii) follows from condition (LT1). We now prove part (i).

Assume by way of contradiction that the claim is false. Then, there exists a $j \neq i$ and an edge $\{x, y\} \in T'$ such that $e \subseteq B(x)$, $e \subseteq B(y)$, and there are vertices $a, b \in V(T_j)$, $a \neq b$, such that $a \in B(x)$ and $b \in B(y)$.

Deletion of edge $\{x, y\}$ partitions $V(T')$ into two sets X and Y. Let $P = \{a \in V(T_j) : a \in B(z) \text{ for some } z \in X\}$ and $Q = \{b \in V(T_j) : b \in B(z) \text{ for some } z \in Y\}$. By the coherence property, (P, Q) is a partition of $V(T_j)$. There must be a vertex p in set P and a vertex q in set Q such that $\{p, q\} \in E(T_j)$. Since G' is a legal triangulation, there must be a node z in T' such that $p, q \in B(z)$. Irrespective of whether z is in set X or Y, the coherence property is violated, a contradiction. □

A legal triangulation of the display graph of a profile is *concise* if

(C1) each internal edge is contained in exactly one maximal clique in the triangulation and
(C2) every vertex that is a leaf in some tree is contained in exactly one maximal clique of the triangulation.

Lemma 7. *Let G be the display graph of a profile $\mathcal{P}$. If G has a legal triangulation, then G has a concise legal triangulation.*

Proof. Let G' be a legal triangulation of the display graph G of profile $\mathcal{P}$ that is not concise. Let (T', B) be a clique tree for G'. We will build a concise legal triangulation for G by repeatedly applying contraction operations on (T', B). The *contraction* of an edge $e = \{x, y\}$ in T' is the operation that consists of (i) replacing x and y by a single (new) node z, (ii) adding edges from node z to every neighbor of x and y, and (iii) making $B(z) = B(x) \cup B(y)$. Note that the resulting pair (T', B) is a tree decomposition for G (and G'); however, it is not guaranteed to be a clique tree for G'.

We proceed in two steps. First, for every leaf v of G such that $|C(v)| > 1$, contract each edge $e = \{x, y\}$ in T' such that $x, y \in C(v)$. In the second step, we consider each edge $e = \{u, v\}$ of G such that $|C(u) \cap C(v)| > 1$, contract each edge $\{x, y\}$ in T' such that $x, y \in C(u) \cap C(v)$. Lemma 5 (respectively, Lemma 6) ensures that each contraction done in the first (respectively, second) step leaves us with a new tree decomposition whose associated triangulation is legal. Furthermore, the triangulation associated with the final tree decomposition is concise. □

Proof of Lemma 2. We will show that, given a legal triangulation G' of the display graph G of profile $\mathcal{P}$, we can generate a supertree T for $\mathcal{P}$ along with an embedding function ϕ_i from T to T_i, for $i = 1, \ldots, k$. By Lemma 4, this immediately implies that $\mathcal{P}$ is compatible.

By Lemma 7, we can assume that G' is concise. Let (T', B) be a clique tree for G'. Initially, we make $T = T'$. Next, for each node x of T, we consider three possibilities.

Case 1: *$B(x)$ contains a labeled vertex v of G.* Then, v is a leaf in some input tree T_i; further, by conciseness, x is the unique node in T such that $v \in B(x)$, and, by the edge coverage property, if u is the neighbor of v in T_i, $u \in B(x)$. Now, do the following.
  (i) Add a new node x_v and a new edge $\{x, x_v\}$ to T.
  (ii) Label x_v with ℓ, where ℓ is the label of v.
  (iii) For each $i \in \{1, \ldots, k\}$ such that v is a leaf in T_i, make $\phi_i(x_v) = v$ and $\phi_i(x) = u$, where u is the neighbor of v in T_i.

Case 2: *$B(x)$ contains both endpoints of an internal edge $e = \{u, v\}$ of some input tree T_i.* By legality, $B(x)$ does not contain both endpoints of any other edge of any input tree, and, by conciseness, x is the only node of T that contains both endpoints of e. Now, do the following.
  (i) Replace node x with nodes x_u and x_v, and add edge $\{x_u, x_v\}$.
  (ii) Add an edge between node x_u and every node neighbor y of x such that $u \in B(y)$.
  (iii) Add an edge between node x_v and every neighbor y of x such that $v \in B(y)$.
  (iv) For each neighbor y of x such that $u \notin B(y)$ and $v \notin B(y)$, add an edge from y to node x_u or node x_v, but not to both (the choice of which edge to add is arbitrary).
  (v) For every $j \in \{1, \ldots, k\}$, $(j \neq i)$ such that $B(x) \cap V(T_j) \neq \emptyset$, make $\phi_j(x_u) = \phi_j(x_v) = z$ where, z is the vertex of T_j contained in $B(x)$. Also, make $\phi_i(x_u) = u$ and $\phi_i(x_v) = v$.

Case 3: *$B(x)$ contains at most one internal vertex from T_i for $i \in \{1, \ldots, k\}$.* Then, for every i such that $B(x) \cap V(T_i) \neq \emptyset$ make $\phi_i(x) = v$, where v is the vertex of T_i contained in $B(x)$.

By construction (Case 1) and the legality and conciseness of (T', B), for every $\ell \in \bigcup_{i=1}^{k} \mathcal{L}(T_i)$ there is exactly one leaf $x \in V(T)$ that is labeled ℓ. Thus, T is a supertree of profile $\mathcal{P}$. Property (TD1) also ensures that the function ϕ_i is a surjective map from a subgraph of T to T_i. Furthermore, the handling of Case 1 guarantees that ϕ_i satisfies (EF1). The coherence of (T', B) and the handling of all cases ensure that ϕ_i satisfies (EF2). The handling of Case 2 and conciseness ensure that ϕ_i satisfies (EF3). Thus, ϕ_i is an embedding function, and, by Lemma 4, profile $\mathcal{P}$ is compatible. □

Acknowledgement

This work was supported in part by the National Science Foundation under grants DEB-0334832, DEB-0829674, and CCF-106029.

References

[1] C. Semple, M. Steel, Phylogenetics, in: Oxford Lecture Series in Mathematics, Oxford University Press, Oxford, 2003.
[2] A.D. Gordon, Consensus supertrees: the synthesis of rooted trees containing overlapping sets of labelled leaves, Journal of Classification 9 (1986) 335–348.
[3] M.A. Steel, The complexity of reconstructing trees from qualitative characters and subtrees, Journal of Classification 9 (1992) 91–116.
[4] A.V. Aho, Y. Sagiv, T.G. Szymanski, J.D. Ullman, Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions, SIAM Journal on Computing 10 (3) (1981) 405–421.
[5] D. Bryant, J. Lagergren, Compatibility of unrooted phylogenetic trees is FPT, Theoretical Computer Science 351 (2006) 296–302.
[6] P. Buneman, A characterisation of rigid circuit graphs, Discrete Mathematics 9 (1974) 205–212.
[7] D. Gusfield, The multi-state perfect phylogeny problem with missing and removable data: solutions via integer-programming and chordal graph theory, in: S. Batzoglou (Ed.), RECOMB, in: Lecture Notes in Computer Science, vol. 5541, Springer, 2009, pp. 236–252.
[8] F.R. McMorris, T.J. Warnow, T. Wimer, Triangulating vertex colored graphs, SIAM Journal on Discrete Mathematics 7 (2) (1993) Preliminary version in 4th Annual Symposium on Discrete Algorithms, Austin, Texas.
[9] S. Grunewald, P.J. Humphries, C. Semple, Quartet compatibility and the quartet graph, The Electronic Journal of Combinatorics 15 (1) (2008) R103.
[10] P. Heggernes, Treewidth, partial *k*-trees, and chordal graphs, Partial curriculum in INF334—advanced algorithmical techniques, Department of Informatics, University of Bergen, Norway, 2005.