

RECONSTRUCTING MINIMAL ROOTED TREES

CHARLES SEMPLE

ABSTRACT. For a set \mathcal{T} of rooted binary leaf-labelled trees, we present an algorithm that finds all of the minor-minimal trees that are compatible with \mathcal{T} . The running time of this algorithm is polynomial up to the number of trees with this property. This type of problem arises in several areas of classification, particularly evolutionary biology.

1. INTRODUCTION

Let T be a leaf-labelled tree. If T has exactly one distinguished non-pendant vertex (called the *root*), while the remaining non-pendant vertices each have degree at least three, then T is called a *rooted phylogenetic tree*. A fundamental task in evolutionary biology, and in other areas of classification, is to combine a collection of rooted phylogenetic trees (the input trees) into a single rooted phylogenetic tree (the output tree) whose leaf set consists of the union of the leaf sets of the input trees. If it is possible, we would like the output tree to “display” each of the input trees. In general, however, the set of input trees will carry conflicting information, in which case, no output tree can possibly display each of the input trees.

In this paper, we consider the case where the set of input trees carry no conflicting information. Let T be a rooted phylogenetic tree, and let A be a subset of the leaf set of T . Consider the minimal subtree $T(A)$ of T containing A . Let $T|A$ denote the rooted phylogenetic tree obtained from $T(A)$ by distinguishing the vertex of $T(A)$ closest to the root of T , and by suppressing all vertices with degree two except for the distinguished vertex. We call $T|A$ the *subtree of T induced by A* . For example, Figure 1(i) shows a rooted phylogenetic tree T with leaf set $\{a, b, c, d, e, f\}$, and Figure 1(ii) shows the subtree of T induced by $\{b, c, e\}$. For a set \mathcal{T} of rooted phylogenetic trees, a rooted phylogenetic tree T is *compatible* with \mathcal{T} if every element of \mathcal{T} is an induced subtree of T , in which case \mathcal{T} is *consistent*. If no such tree exists, then we say that \mathcal{T} is *inconsistent*.

A rooted tree is *binary* if every non-pendant vertex has degree three except for the root which has degree two. A *rooted triple* is a rooted binary phylogenetic tree on three leaves. The rooted triple with leaves x , y , and z is denoted by $xy|z$ if the

Date: June 24, 1999.

1991 *Mathematics Subject Classification.* 05C05, 92D15.

Key words and phrases. Rooted phylogenetic tree, tree compatibility.

This work was supported by the grant UOC-MIS-003 to Mike Steel from the New Zealand Marsden Fund.

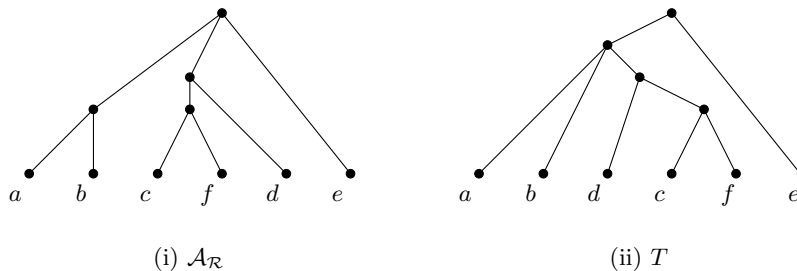


FIGURE 2. Two minimal rooted phylogenetic trees compatible with $\mathcal{R} = \{ab|e, cd|e, cd|a, cf|d\}$.

This paper has two purposes. The first is to describe an algorithm that finds, for a consistent set \mathcal{R} of rooted triples, the set $\mathcal{T}_{\mathcal{R}}^{\min}$ of all minimal rooted phylogenetic trees compatible with \mathcal{R} . Up to the cardinality of $\mathcal{T}_{\mathcal{R}}^{\min}$, this algorithm runs in polynomial time, that is, each tree in $\mathcal{T}_{\mathcal{R}}^{\min}$ is constructed in polynomial time. Since a rooted binary phylogenetic tree can be defined by a set of rooted triples in polynomial time, this result extends to a consistent set of rooted binary phylogenetic trees. The motivation for determining the set $\mathcal{T}_{\mathcal{R}}^{\min}$ is that all of the information provided by the set $\mathcal{T}_{\mathcal{R}}$ of all rooted phylogenetic trees compatible with \mathcal{R} is contained in $\mathcal{T}_{\mathcal{R}}^{\min}$ since $\mathcal{T}_{\mathcal{R}}$ can be (easily) deduced from $\mathcal{T}_{\mathcal{R}}^{\min}$ by “partially resolving” each of the internal vertices of every tree of $\mathcal{T}_{\mathcal{R}}^{\min}$ in all possible ways.

The second purpose of this paper is the following. Let \mathcal{R} be a consistent set of rooted triples, and let $\mathcal{T}_{\mathcal{R}}$ denote the set of all rooted phylogenetic trees compatible with \mathcal{R} . Let $\mathcal{A}_{\mathcal{R}}$ denote the unique tree outputted by the algorithm described by Aho *et al.* [1] whose input is \mathcal{R} . Note that $\mathcal{A}_{\mathcal{R}}$ is the “particular tree” mentioned above and is compatible with \mathcal{R} . The algorithm that constructs $\mathcal{A}_{\mathcal{R}}$ is fundamental in determining $\mathcal{T}_{\mathcal{R}}$ (see [4, 6]). In this paper, we give a characterization of $\mathcal{A}_{\mathcal{R}}$ with respect to the other members of $\mathcal{T}_{\mathcal{R}}$ based on the clusters of $\mathcal{A}_{\mathcal{R}}$. This characterization is stated as Theorem 3.1.

The paper is organized as follows. In the next section, we present some additional preliminaries. Section 3 contains the statement of Theorem 3.1 and its proof. In Section 4, we present our method, called ALLMINTREES, for finding all minimal rooted phylogenetic trees compatible with a consistent set of rooted triples, and show that each tree is outputted from ALLMINTREES in polynomial time.

2. PRELIMINARIES

In this section, we state some further terminology and notation that will be needed throughout the paper. If $G = (V, E)$ is a graph, then we denote the subgraph of G induced by a subset V' of V by $G[V']$.

Clusters

Let T be a rooted phylogenetic tree having leaf set X . An element of X is a *descendant* of a vertex v of T if the path from this element to the root of T

passes through v . A *cluster* of T is a subset of X in which all the elements are the descendants of some particular vertex of T . If this vertex is v , then we denote this cluster by C_v . The set X is always a cluster of T ; all other clusters of T are said to be *proper*. We denote the collection of all clusters of T by $\mathcal{C}(T)$. If a and b are elements of X , and u is a vertex of T such that C_u is the (unique) minimal cluster of T containing a and b , then u is the *least common ancestor* of a and b .

Rooted triples, a set \mathcal{R} of rooted triples, and the tree $\mathcal{A}_{\mathcal{R}}$

Let T be a rooted phylogenetic tree. We let $\mathcal{L}(T)$ denote the leaf set of T . Furthermore, we let $r(T)$ denote the set of rooted triples of T , that is, the set whose members are precisely the binary subtrees of T induced by a 3-element subset of $\mathcal{L}(T)$.

Let $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$ be a set of rooted triples. The set $\bigcup_{i=1}^k \mathcal{L}(R_i)$ is said to be the *leaf set* of \mathcal{R} . We shall denote by $\mathcal{T}_{\mathcal{R}}$ the set of all rooted phylogenetic trees compatible with \mathcal{R} and denote by $\mathcal{T}_{\mathcal{R}}^{\min}$ the set of all minimal rooted phylogenetic trees compatible with \mathcal{R} . Note that $\mathcal{T}_{\mathcal{R}}$, and therefore $\mathcal{T}_{\mathcal{R}}^{\min}$, is non-empty precisely when \mathcal{R} is consistent.

Let \mathcal{R} be a set of rooted triples, and let S be a subset of the leaf set of \mathcal{R} . We denote by $\mathcal{R}(S)$ the subset of rooted triples of \mathcal{R} whose leaves are completely labelled by elements of S . We denote by $S_{\mathcal{R}}$ the graph that has vertex set S and an edge joining two vertices a and b precisely if there is a rooted triple in $\mathcal{R}(S)$ of the form $ab|c$. This graph will be fundamental throughout this paper.

Let \mathcal{R} be a set of rooted triples having leaf set S . The importance of $S_{\mathcal{R}}$ is that, provided \mathcal{R} is consistent, the tree $\mathcal{A}_{\mathcal{R}}$ described by Aho *et al.* [1] can be recursively constructed from subgraphs of $S_{\mathcal{R}}$ [2]. For completeness, we briefly outline how this can be done. Firstly, construct the graph $S_{\mathcal{R}}$. If $|S| > 2$ and this graph has exactly one component, then \mathcal{R} is not consistent and a statement to this effect is returned. Otherwise, the vertex sets S_1, S_2, \dots, S_k of the components of $S_{\mathcal{R}}$ are the maximal clusters of $\mathcal{A}_{\mathcal{R}}$ (provided it turns out that \mathcal{R} is consistent). Now, for each $i \in \{1, 2, \dots, k\}$, we construct the graph $(S_i)_{\mathcal{R}}$. If, for some i , $|S_i| > 2$ and $(S_i)_{\mathcal{R}}$ consists of a single component, then \mathcal{R} is inconsistent and a statement to this effect is returned. Otherwise, the vertex sets of the components of $(S_i)_{\mathcal{R}}$ are the maximal clusters of $\mathcal{A}_{\mathcal{R}}|S_i$ (provided it turns out that \mathcal{R} is consistent). This process continues in the obvious way eventually returning either the statement that \mathcal{R} is inconsistent, or $\mathcal{A}_{\mathcal{R}}$. As an example, suppose that $\mathcal{R} = \{ab|e, cd|e, cd|a, cf|d\}$. Letting $S = \{a, b, c, d, e, f\}$, the graph $S_{\mathcal{R}}$ is shown in Figure 3(i) and, noting that \mathcal{R} is consistent, the tree $\mathcal{A}_{\mathcal{R}}$ is shown in Figure 2(i).

3. A CHARACTERIZATION OF $\mathcal{A}_{\mathcal{R}}$

Let \mathcal{R} be a consistent set of rooted triples. In this section, we present Theorem 3.1, a characterization of $\mathcal{A}_{\mathcal{R}}$ with respect to the other trees in $\mathcal{T}_{\mathcal{R}}$ based on the clusters of $\mathcal{A}_{\mathcal{R}}$. Another characterization of $\mathcal{A}_{\mathcal{R}}$ is given by [3, Theorem 6.2] which says that $\mathcal{A}_{\mathcal{R}}$ is the ‘‘Adams Consensus’’ tree for $\mathcal{T}_{\mathcal{R}}$.

Theorem 3.1. *Let \mathcal{R} be a consistent set of rooted triples, and let T be a member of $\mathcal{T}_{\mathcal{R}}$. Then T is isomorphic to $\mathcal{A}_{\mathcal{R}}$ if and only if, for every member T' of $\mathcal{T}_{\mathcal{R}}$, the mapping $\phi : \mathcal{C}(T) \rightarrow \mathcal{C}(T')$, defined by $\phi(C) = C'$, where C' is the (unique) minimal cluster of T' containing C , satisfies the following property:*

$$(1) \quad \text{if } C_1, C_2 \in \mathcal{C}(T) \text{ and } C_1 \subset C_2, \text{ then } \phi(C_1) \subset \phi(C_2).$$

Proof. Since the labelling of the internal vertices, other than the root, of a rooted phylogenetic tree are unimportant, we shall treat a tree that is isomorphic to $\mathcal{A}_{\mathcal{R}}$ as being equal to $\mathcal{A}_{\mathcal{R}}$ in the proof of Theorem 3.1. Suppose that T is isomorphic to $\mathcal{A}_{\mathcal{R}}$. Treating T as being equal to $\mathcal{A}_{\mathcal{R}}$, suppose that C_1 and C_2 are members of $\mathcal{C}(\mathcal{A}_{\mathcal{R}})$ with the property that $C_1 \subset C_2$. To prove the “only if” part of the theorem, it is easily seen that we may also suppose C_1 and C_2 have the additional property that there is no cluster C_3 in $\mathcal{C}(\mathcal{A}_{\mathcal{R}})$ such that $C_1 \subset C_3 \subset C_2$.

Let T' be a member of $\mathcal{T}_{\mathcal{R}}$, and let ϕ be the map $\phi : \mathcal{C}(\mathcal{A}_{\mathcal{R}}) \rightarrow \mathcal{C}(T')$ as defined in the statement of Theorem 3.1. Suppose, to the contrary, that $\phi(C_1) = \phi(C_2)$. Then there is at least two distinct maximal clusters of $T'|\phi(C_2)$ such that, for each of these clusters, their intersection with C_1 is non-empty. Consider the graph $(C_2)_{\mathcal{R}}$. Since C_1 is a maximal cluster of $\mathcal{A}_{\mathcal{R}}|C_2$, the set C_1 is the vertex set of exactly one component of $(C_2)_{\mathcal{R}}$. Thus there must be elements, a and b say, of C_1 such that a and b are in distinct maximal clusters of $T'|\phi(C_2)$ and $ab|c$ is a rooted triple of \mathcal{R} , where c is an element of C_2 . But then, as $\phi(C_1) = \phi(C_2)$, it follows that $ab|c$ cannot be a rooted triple of T' ; a contradiction. Hence $\phi(C_1) \subset \phi(C_2)$, thus completing the proof of the necessary part of the theorem.

Now suppose that T satisfies the “if” condition in the statement of Theorem 3.1. Then there is map $\phi_2 : \mathcal{C}(T) \rightarrow \mathcal{C}(\mathcal{A}_{\mathcal{R}})$ defined by $\phi_2(C) = D$, where D is the minimal cluster of $\mathcal{A}_{\mathcal{R}}$ containing C , that satisfies property (1) with “ ϕ ” replaced by “ ϕ_2 ”. Let ϕ_1 denote the map $\phi_1 : \mathcal{C}(\mathcal{A}_{\mathcal{R}}) \rightarrow \mathcal{C}(T)$ that is defined in the canonical way to which ϕ is defined in the statement of the theorem. From above, ϕ_1 satisfies property (1) with “ T ” and “ ϕ ” replaced by “ $\mathcal{A}_{\mathcal{R}}$ ” and “ ϕ_1 ”, respectively. We prove the converse of the theorem, by showing that $\mathcal{C}(T) = \mathcal{C}(\mathcal{A}_{\mathcal{R}})$.

We first show that $\mathcal{C}(\mathcal{A}_{\mathcal{R}})$ is a subset of $\mathcal{C}(T)$. Let $\mathcal{C}' = \{C_1, C_2, \dots, C_j\}$ be a maximal subset of $\mathcal{C}(\mathcal{A}_{\mathcal{R}})$ with the property that, for all $i \in \{2, 3, \dots, j\}$, C_{i-1} is a proper subset of C_i . Then C_j is equal to the leaf set of \mathcal{R} , and therefore $\phi_1(C_j) = C_j$ and $\phi_2(\phi_1(C_j)) = C_j$. We now show that if $\phi_1(C_i) = C_i$ and $\phi_2(\phi_1(C_i)) = C_i$ for some i , then $\phi_1(C_{i-1}) = C_{i-1}$ and $\phi_2(\phi_1(C_{i-1})) = C_{i-1}$. Thus showing by induction that every member of \mathcal{C}' is a member of $\mathcal{C}(T)$, which in turn implies that $\mathcal{C}(\mathcal{A}_{\mathcal{R}}) \subseteq \mathcal{C}(T)$. Now $\phi_1(C_{i-1}) \subset \phi_1(C_i)$, and so

$$(2) \quad \phi_2(\phi_1(C_{i-1})) \subset \phi_2(\phi_1(C_i)).$$

By combining (2) with the induction assumption, we deduce that

$$(3) \quad C_{i-1} \subseteq \phi_1(C_{i-1}) \subseteq \phi_2(\phi_1(C_{i-1})) \subset \phi_2(\phi_1(C_i)) = C_i.$$

Since \mathcal{C}' is maximal, it follows from (3) that $\phi_2(\phi_1(C_{i-1})) = C_{i-1}$, which in turn implies from (3) that $\phi_1(C_{i-1}) = C_{i-1}$ as required.

By interchanging the roles of $\mathcal{A}_{\mathcal{R}}$ and T , and applying the argument of the previous paragraph, we deduce that $\mathcal{C}(T) \subseteq \mathcal{C}(\mathcal{A}_{\mathcal{R}})$. Hence $\mathcal{C}(T) = \mathcal{C}(\mathcal{A}_{\mathcal{R}})$ as required. \square

An attractive property of $\mathcal{A}_{\mathcal{R}}$ for a consistent set \mathcal{R} of rooted triples is stated as Corollary 3.2. For a consistent set \mathcal{R} of rooted triples, let $\text{cl}(\mathcal{R})$ denote the set $\bigcap_{T \in \mathcal{T}_{\mathcal{R}}} r(T)$. The set $\text{cl}(\mathcal{R})$ is called the *closure* of \mathcal{R} . The notion of a closed set of rooted triples is introduced in [2], where a number of properties of such sets are established.

Corollary 3.2. *Let \mathcal{R} be a consistent set of rooted triples. Then $\mathcal{A}_{\mathcal{R}} \cong \mathcal{A}_{\text{cl}(\mathcal{R})}$.*

Proof. Clearly, $\mathcal{T}_{\text{cl}(\mathcal{R})} \subseteq \mathcal{T}_{\mathcal{R}}$. Moreover, if T is a member of $\mathcal{T}_{\mathcal{R}}$, then, by the definition of $\text{cl}(\mathcal{R})$, T is a member of $\mathcal{T}_{\text{cl}(\mathcal{R})}$, and so $\mathcal{T}_{\mathcal{R}} \subseteq \mathcal{T}_{\text{cl}(\mathcal{R})}$. Hence $\mathcal{T}_{\mathcal{R}} = \mathcal{T}_{\text{cl}(\mathcal{R})}$. It is now straightforward to deduce using Theorem 3.1 that $\mathcal{A}_{\mathcal{R}} \cong \mathcal{A}_{\text{cl}(\mathcal{R})}$ as required. \square

The next corollary will be useful in describing the strategy of ALLMINTREES.

Corollary 3.3. *Let \mathcal{R} be a consistent set of rooted triples. Let T be a tree compatible with \mathcal{R} . If C is a maximal cluster of $\mathcal{A}_{\mathcal{R}}$, then C is contained in a maximal cluster of T .*

Proof. Suppose, to the contrary, that C is not contained in a maximal cluster of T . This implies that there must be two maximal clusters of T that have a non-empty intersection with C . Let S be the leaf set of \mathcal{R} . Then the minimal cluster of T containing C is S . It now follows that we have a contradiction to Theorem 3.1. \square

4. THE ALLMINTREES ALGORITHM

In this section, we present the algorithm ALLMINTREES which outputs the set of all minimal rooted phylogenetic trees compatible with a consistent set \mathcal{R} of rooted triples when applied to \mathcal{R} , and show that each tree that is outputted from ALLMINTREES is constructed in polynomial time. Since determining the consistency of a set of rooted triples can be done in polynomial time and since every subset of a consistent set of rooted triples is consistent, no generality is lost in restricting the input to ALLMINTREES to a set of rooted triples that are consistent. Throughout this section, the reader may find it useful to refer to Figure 2 and Example 4.3 which illustrate many of the concepts introduced.

We begin with some preliminaries.

Definition. Let \mathcal{R} be a consistent set of rooted triples, and let S be a cluster of a rooted phylogenetic tree compatible with \mathcal{R} . We call the non-empty union of the vertex sets of some (possibly one), but not all, of the components of $S_{\mathcal{R}}$ a *merging* of $S_{\mathcal{R}}$. A partition $\mathcal{P}_{S_{\mathcal{R}}}$ of S is said to be a *partition of $S_{\mathcal{R}}$* if each part of $\mathcal{P}_{S_{\mathcal{R}}}$ is a merging of $S_{\mathcal{R}}$ and $\mathcal{P}_{S_{\mathcal{R}}}$ contains at least two parts. The reason for these definitions will become apparent by the end of the next two paragraphs.

We now outline a method for outputting all rooted phylogenetic trees compatible with a consistent set \mathcal{R} of rooted triples having leaf set S . The reason for doing this is that ALLMINTREES is based on this method. Recall the construction of $\mathcal{A}_{\mathcal{R}}$ described in Section 2. The set $\mathcal{T}_{\mathcal{R}}$ of all rooted phylogenetic trees compatible with \mathcal{R} can be obtained in the following recursive way. Firstly, construct the graph $S_{\mathcal{R}}$. Since \mathcal{R} is consistent, the number of components of $S_{\mathcal{R}}$ is at least two provided $|S| \geq 2$. Let $\mathcal{P}_{S_{\mathcal{R}}}$ be a partition of $S_{\mathcal{R}}$. Then, from the way in which $S_{\mathcal{R}}$ has been constructed, it is easily seen that $\mathcal{P}_{S_{\mathcal{R}}}$ is the set of maximal clusters of at least one tree in $\mathcal{T}_{\mathcal{R}}$. Furthermore, it follows by Corollary 3.3 that the set of all maximal clusters of a tree in $\mathcal{T}_{\mathcal{R}}$ can be obtained in this way. Note that in constructing $\mathcal{A}_{\mathcal{R}}$ the partition of $S_{\mathcal{R}}$ considered at this stage is the one in which each part consists of the vertex set of exactly one component of $S_{\mathcal{R}}$.

For each part of each partition of $S_{\mathcal{R}}$, we repeat this process. For example, let S' be a part of $\mathcal{P}_{S_{\mathcal{R}}}$. Construct $S'_{\mathcal{R}}$. Since \mathcal{R} is consistent, the number of components of $S'_{\mathcal{R}}$ is at least two provided $|S'| \geq 2$. By considering $S'_{\mathcal{R}}$, it is straightforward to deduce that, for each member T of $\mathcal{T}_{\mathcal{R}}$ whose maximal clusters are the parts of $\mathcal{P}_{S_{\mathcal{R}}}$, the maximal clusters of the subtree of T induced by S' are exactly the parts of some partition of $S'_{\mathcal{R}}$. Moreover, by considering the subset $\mathcal{R}(S')$ of rooted triples of \mathcal{R} that are labelled by elements of S' and applying Corollary 3.3 to $\mathcal{R}(S')$, the converse also holds. This process continues until all associated graphs consist of a single vertex. The fact that this method determines all members of $\mathcal{T}_{\mathcal{R}}$ is also observed in [6].

Remark. The method just described determines all trees that are compatible with a set of rooted triples. This method is the basis for the algorithm “ALLTREES” in [6], which outputs all trees compatible with a set containing not only rooted triples but also fans, and is polynomial time in the size of the output. The objective in their case is to recognize exactly which further mergings are necessary at each iteration so that the output trees are compatible with each of the fans, while maintaining the property that each such desired merging is recognized in polynomial time. We have an analogous, but quite different, objective in constructing ALLMINTREES, which we describe next.

Definition. Let \mathcal{R} be a consistent set of rooted triples, and let S be a cluster of a minimal rooted phylogenetic tree compatible with \mathcal{R} . A partition of $S_{\mathcal{R}}$ is *good* if it is the set of maximal clusters of a minimal rooted phylogenetic tree in $\mathcal{T}_{\mathcal{R}}^{\min}$ restricted to S .

From this last definition, it now follows that the above recursive description for outputting all rooted phylogenetic trees compatible with \mathcal{R} can be used to find all minimal rooted phylogenetic trees compatible with \mathcal{R} with one modification: at each iteration, instead of determining all partitions of $S_{\mathcal{R}}$, one needs to determine all good partitions of $S_{\mathcal{R}}$ as not all partitions of $S_{\mathcal{R}}$ realize at least one member of $\mathcal{T}_{\mathcal{R}}^{\min}$. This is the approach used by ALLMINTREES. Furthermore, as we desire an algorithm that outputs each tree in $\mathcal{T}_{\mathcal{R}}^{\min}$ in polynomial time, we need to be able to determine in advance the good partitions of $S_{\mathcal{R}}$. Proposition 4.1 is needed for the proof of Theorem 4.2, the latter providing us with a useful graph-theoretic way of recognizing all good partitions of $S_{\mathcal{R}}$.

Proposition 4.1. *Let \mathcal{R} be a consistent set of rooted triples. Then $\mathcal{A}_{\mathcal{R}}$ is a minimal rooted phylogenetic tree compatible with \mathcal{R} .*

Proof. Suppose, to the contrary, that $\mathcal{A}_{\mathcal{R}}$ is not a minimal rooted phylogenetic tree compatible with \mathcal{R} . Then there exists an internal edge e of $\mathcal{A}_{\mathcal{R}}$ such that $\mathcal{A}_{\mathcal{R}}/e$ is a member of $\mathcal{T}_{\mathcal{R}}$. Let u and v be the end-vertices of e such that v is in the path from u to the root of $\mathcal{A}_{\mathcal{R}}$. Since $C_u \subset C_v$, it follows by Theorem 3.1 that $\phi(C_u) \subset \phi(C_v)$, where $\phi(C_u)$ and $\phi(C_v)$ are the minimal clusters of $\mathcal{A}_{\mathcal{R}}/e$ containing C_u and C_v , respectively. But $\phi(C_u) = \phi(C_v)$; a contradiction. Hence $\mathcal{A}_{\mathcal{R}}$ is a minimal rooted phylogenetic tree compatible with \mathcal{R} . \square

Proposition 4.1 shows that, given a consistent set \mathcal{R} of rooted triples, $\mathcal{A}_{\mathcal{R}}$ is an element of $\mathcal{T}_{\mathcal{R}}^{\min}$. However, in general, $\mathcal{A}_{\mathcal{R}}$ is not the only tree in $\mathcal{T}_{\mathcal{R}}^{\min}$ as Figure 2 illustrates.

Theorem 4.2. *Let \mathcal{R} be a consistent set of rooted triples, and let S be a cluster of a minimal rooted phylogenetic tree compatible with \mathcal{R} . A partition $\mathcal{P}_{S_{\mathcal{R}}}$ of $S_{\mathcal{R}}$ is good if and only if each part M of $\mathcal{P}_{S_{\mathcal{R}}}$ satisfies one of the following conditions:*

- (i) M is the vertex set of exactly one component of $S_{\mathcal{R}}$; or
- (ii) M is the union of the vertex sets of at least two components of $S_{\mathcal{R}}$ and the number of components of $M_{\mathcal{R}}$ is strictly greater than the number of components of $S_{\mathcal{R}}[M]$.

Proof. Suppose that $\mathcal{P}_{S_{\mathcal{R}}}$ is a good partition of $S_{\mathcal{R}}$, and that M is not the vertex set of exactly one component of $S_{\mathcal{R}}$. Then M is the union of the vertex sets of at least two components of $S_{\mathcal{R}}$. Let T' be a minimal rooted phylogenetic tree compatible with \mathcal{R} such that S is a cluster of T' and M is a maximal cluster of $T'|S$. Assume, to the contrary, that the number of components of $M_{\mathcal{R}}$ is equal to the number of components of $S_{\mathcal{R}}[M]$. Then there is no pair of elements, a and b say, of M such that a and b are in separate components of $M_{\mathcal{R}}$, and $ab|c$ is an element of \mathcal{R} , where $c \in S - M$. It follows that the rooted phylogenetic tree T'/e is compatible with \mathcal{R} , where e is the edge of T' incident with the vertices associated with the clusters S and M of T' , contradicting the minimality of T' .

To prove the converse, suppose that each part of $\mathcal{P}_{S_{\mathcal{R}}}$ satisfies either (i) or (ii). Let T be an element of $\mathcal{T}_{\mathcal{R}}^{\min}$ such that S is a cluster of T . By the hypothesis of Theorem 4.2, there exists such a tree. Let M_1, M_2, \dots, M_k denote the parts of $\mathcal{P}_{S_{\mathcal{R}}}$. Now let T' be the rooted phylogenetic tree obtained from T by replacing $T|S$ with the rooted phylogenetic tree whose maximal rooted phylogenetic subtrees are $\mathcal{A}_{\mathcal{R}(M_i)}$, where $i \in \{1, 2, \dots, k\}$. By Proposition 4.1, $\mathcal{A}_{\mathcal{R}(M_i)}$ is a minimal rooted phylogenetic tree compatible with $\mathcal{R}(M_i)$ for all i . Let v denote the vertex of T' for which $C_v = S$, and let u_1, u_2, \dots, u_k denote the vertices of T' that are adjacent to v such that, for all i , v lies on the path from u_i to the root of T' . The converse of Theorem 4.2 now follows by showing that, for all i , the tree $T'/\{u_i, v\}$ is not compatible with \mathcal{R} .

Suppose, to the contrary, that $T/\{u_i, v\}$ is compatible with \mathcal{R} for some i . Assume first that M_i satisfies (ii). Then there are two components of $(M_i)_{\mathcal{R}}$ whose vertex sets, S_1 and S_2 say, are subsets of a vertex set of a component of $S_{\mathcal{R}}[M_i]$. Thus there must be a rooted triple $ab|c$ in \mathcal{R} such that $a \in S_1$, $b \in S_2$, and c is an element of $S - M_i$. But then, as $T'|M_i$ is isomorphic to $\mathcal{A}_{\mathcal{R}(M_i)}$, $ab|c$ is not a rooted triple of $T'/\{u_i, v\}$. Hence, in this case, $T'/\{u_i, v\}$ is not compatible with \mathcal{R} .

Now assume that M_i satisfies (i). If $|M_i| = 1$, then $T'/\{u_i, v\}$ is not a rooted phylogenetic tree. Therefore we may assume that $|M_i| \geq 2$. Then, as \mathcal{R} is consistent, the number of components of $(M_i)_{\mathcal{R}}$ is at least two. Thus M_i satisfies (ii), and so, once again, $T'/\{u_i, v\}$ is not compatible with \mathcal{R} . This completes the proof of Theorem 4.2. \square

In view of Theorem 4.2, we make the following definition.

Definition. Let \mathcal{R} be a consistent set of rooted triples, and let S be a cluster of rooted phylogenetic tree compatible with \mathcal{R} . We call a merging of $S_{\mathcal{R}}$ *good* if it satisfies either (i) or (ii) of Theorem 4.2.

At last we present ALLMINTREES. This algorithm has a subroutine called GOODMERGINGS that is repeatedly used at each iteration to construct an appropriate set of good mergings. The details of this subroutine, including its description, follows ALLMINTREES. Let \mathcal{R} be a consistent set of rooted triples, and let S be a subset of the leaf set of \mathcal{R} . Consider the graph $S_{\mathcal{R}}$. Let S_1, S_2, \dots, S_k denote the vertex sets of the components of $S_{\mathcal{R}}$. Since \mathcal{R} is consistent, $k \geq 2$ provided $|S| \geq 2$. Let $S_{\mathcal{R}}^*$ denote the graph obtained from $S_{\mathcal{R}}$ by removing, for all $i \in \{1, 2, \dots, k\}$, those edges in S_i whose end-vertices, a_i and b_i say, have the property that if $a_i b_i | c$ is a rooted triple in \mathcal{R} with $c \in S$, then c is an element of $S - S_i$. For all i , we denote the vertex sets of $S_{\mathcal{R}}^*[S_i]$, the subgraph of $S_{\mathcal{R}}^*$ induced by S_i , by $S_{i1}, S_{i2}, \dots, S_{in_i}$ (see Example 4.3). The graph $S_{\mathcal{R}}^*$ is needed for GOODMERGINGS.

Algorithm: ALLMINTREES($\mathcal{R}, \mathcal{T}_{\mathcal{R}}^{\min}$).

Input: A consistent set \mathcal{R} of rooted triples.

Output: The set $\mathcal{T}_{\mathcal{R}}^{\min}$ of all minimal rooted phylogenetic trees compatible with \mathcal{R} .

- (1) Set S to be the leaf set of \mathcal{R} .
- (2) If $|S| \leq 2$, then return the rooted tree with the elements of S as leaves.
- (3) Otherwise, if $|S| > 2$, then construct $S_{\mathcal{R}}$ and $S_{\mathcal{R}}^*$. Let S_1, S_2, \dots, S_k ($k \geq 2$) denote the vertex sets of the components of $S_{\mathcal{R}}$.
- (4) Initially set $\mathcal{M}_{S_{\mathcal{R}}}$ to be the empty set.
- (5) Set $r_1 = r_2 = \dots = r_k = 1$.
- (6) Set i to be the least number for which $r_i = 1$, and set $r_i = 0$.
- (7) Set $\mathcal{P}_i = \{S_i, S_{i+1}, \dots, S_k\}$.
- (8) GOODMERGINGS($\mathcal{P}_i, \mathcal{M}_i$).
- (9) Replace $\mathcal{M}_{S_{\mathcal{R}}}$ with $\mathcal{M}_{S_{\mathcal{R}}} \cup \mathcal{M}_i$.
- (10) If $i \neq k$, then go to Step 6.
- (11) For each element M of $\mathcal{M}_{S_{\mathcal{R}}}$, ALLMINTREES($\mathcal{R}(M), \mathcal{T}_{\mathcal{R}(M)}^{\min}$).

- (12) For every distinct partition $\{M_1, M_2, \dots, M_p\}$ of $S_{\mathcal{R}}$ in which each part is an element of $\mathcal{M}_{S_{\mathcal{R}}}$ do the following: for each $q \in \{1, 2, \dots, p\}$, choose a tree in $\mathcal{T}_{\mathcal{R}(M_q)}^{\min}$ and make a new tree by attaching the roots of each of these chosen trees to a new vertex, add the resulting tree to $\mathcal{T}_{\mathcal{R}}^{\min}$; repeat this process for all possible combinations of trees chosen in this way.
- (13) Output $\mathcal{T}_{\mathcal{R}}^{\min}$.

Remark. We show in Lemma 4.5 that the set “ $\mathcal{M}_{S_{\mathcal{R}}}$ ” in Step 11 of ALLMINTREES is the set of all good mergings of $S_{\mathcal{R}}$. Furthermore, we note here that the set “ \mathcal{M}_i ” in Step 9 of ALLMINTREES is the set of all good mergings of $S_{\mathcal{R}}$ that contain S_i , but none of the sets S_1, S_2, \dots, S_{i-1} .

Before presenting the subroutine GOODMERGINGS, we need to define a matrix that is used in GOODMERGINGS which enables us to output each tree in $\mathcal{T}_{\mathcal{R}}^{\min}$ in polynomial time. Example 4.3 illustrates this definition. Suppose that \mathcal{R} is a consistent set of rooted triples. Let S be a subset of the leaf set of \mathcal{R} , and consider the graphs $S_{\mathcal{R}}$ and $S_{\mathcal{R}}^*$. Let S_1, S_2, \dots, S_k denote the vertex sets of the components of $S_{\mathcal{R}}$. Let S' be a subset of S such that S' is the union of the vertex sets of some components of $S_{\mathcal{R}}$. Let \mathcal{P} be a partition of S' such that each part of \mathcal{P} is a merging of $S_{\mathcal{R}}$. Note that, in general, \mathcal{P} is not a partition of $S_{\mathcal{R}}$. We now define a $0 - 1$ matrix $A_{\mathcal{P}}$ associated with \mathcal{P} , and the components of $S_{\mathcal{R}}$ and $S_{\mathcal{R}}^*$. Let \mathcal{P}^* denote the partition of S' in which each part is an element of

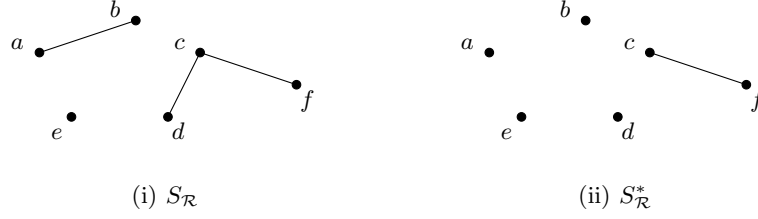
$$\{S_{11}, \dots, S_{1n_1}, S_{21}, \dots, S_{kn_k}\}.$$

The rows of $A_{\mathcal{P}}$ are labelled with the parts of \mathcal{P} and, respecting the ordering of the rows, the columns of $A_{\mathcal{P}}$ are labelled with the parts of \mathcal{P}^* . This labelling canonically partitions the matrix $A_{\mathcal{P}}$ into *blocks* each consisting of a single row and whose columns are labelled by the parts of \mathcal{P}^* whose union is a part of \mathcal{P} . An entry of $A_{\mathcal{P}}$ with row label M and column label S_{in_j} , for some $i, j \in \{1, 2, \dots, k\}$, is given a value of 1 precisely if one of the following holds:

- (a) there exists a rooted triple $ab|c$ in $\mathcal{R}(S')$ such that a is an element of S_{in_j} , b is not an element of S_{in_j} , and c is an element of M ; or
- (b) S_{in_j} labels a row of $A_{\mathcal{P}}$ and $M \neq S_{in_j}$.

Otherwise, the entry is given the value of 0. Observe that if there is such a rooted triple that satisfies (a), then b labels a vertex of $S_{\mathcal{R}}$ that is in the same component as the vertex labelled by a , and c labels a vertex of $S_{\mathcal{R}}$ that is in a different component from the vertex labelled by a . Furthermore, if (b) applies, then $|S_{in_j}| = 1$ as \mathcal{R} is a consistent set of rooted triples.

Example 4.3. Consider the consistent set $\mathcal{R} = \{ab|e, cd|e, cd|a, cf|d\}$ of rooted triples (see Figure 2). Let $S = \{a, b, c, d, e, f\}$ and $\mathcal{P} = \{\{a, b\}, \{c, d, f\}, \{e\}\}$. The graphs $S_{\mathcal{R}}$ and $S_{\mathcal{R}}^*$ are shown in Figure 3, and the matrix $A_{\mathcal{P}}$ is shown in Figure 4. Furthermore, the good mergings of $S_{\mathcal{R}}$ are $\{a, b\}$, $\{c, d, f\}$, $\{e\}$, and $\{a, b, c, d, f\}$, and so, by Theorem 4.2, the good partitions of $S_{\mathcal{R}}$ are $\{\{a, b\}, \{c, d, f\}, \{e\}\}$ and $\{\{a, b, c, d, f\}, \{e\}\}$.


 FIGURE 3. (i) $S_{\mathcal{R}}$. (ii) $S_{\mathcal{R}}^*$.

$$\begin{array}{c}
 \{a\} \quad \{b\} \quad \{c, f\} \quad \{d\} \quad \{e\} \\
 \begin{array}{l}
 \{a, b\} \\
 \{c, d, f\} \\
 \{e\}
 \end{array}
 \begin{bmatrix}
 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 1 \\
 1 & 1 & 1 & 1 & 0
 \end{bmatrix}
 \end{array}$$

 FIGURE 4. The matrix $A_{\mathcal{P}}$.

$$\begin{array}{c}
 \{a\} \quad \{b\} \quad \{c, f\} \quad \{d\} \\
 \begin{array}{l}
 \{a, b\} \\
 \{c, d, f\}
 \end{array}
 \begin{bmatrix}
 0 & 0 & 1 & 1 \\
 0 & 0 & 0 & 0
 \end{bmatrix}
 \end{array}
 \qquad
 \begin{array}{c}
 \{a\} \quad \{b\} \quad \{e\} \\
 \begin{array}{l}
 \{a, b\} \\
 \{e\}
 \end{array}
 \begin{bmatrix}
 0 & 0 & 1 \\
 1 & 1 & 0
 \end{bmatrix}
 \end{array}$$

 FIGURE 5. Two submatrices of $A_{\mathcal{P}}$.

GOODMERGINGS repeatedly constructs matrices of the type defined above for the following reason. Let \mathcal{R} be a consistent set of rooted triples, and let S be a cluster of a minimal rooted phylogenetic tree compatible with \mathcal{R} . Let \mathcal{P} denote the partition of $S_{\mathcal{R}}$ in which each part is the vertex set of exactly one component of $S_{\mathcal{R}}$. By Theorem 4.2, the good partitions of $S_{\mathcal{R}}$ are precisely the partitions of $S_{\mathcal{R}}$ in which each part is a good merging of $S_{\mathcal{R}}$. In turn, the good mergings of $S_{\mathcal{R}}$ can be recognized by the following observation. Let M be a merging of $S_{\mathcal{R}}$ and, viewing the matrix $A_{\mathcal{P}}$ as being partitioned into the blocks described above, let $A'_{\mathcal{P}}$ be the submatrix of $A_{\mathcal{P}}$ that (i) consists of blocks, (ii) the number of blocks in each row is equal to the number of blocks in each column, and (iii) the union of the row labels of $A'_{\mathcal{P}}$ is M . Then M is a good merging of $S_{\mathcal{R}}$ if and only if $A'_{\mathcal{P}}$ contains a column of zeros. For example, in Example 4.3, $\{a, b, c, d, f\}$ is a good merging of $S_{\mathcal{R}}$ as the submatrix of $A_{\mathcal{P}}$ shown on the left in Figure 5 contains a column of zeros. On the other hand, $\{a, b, e\}$ is not a good merging of $S_{\mathcal{R}}$ as the submatrix of $A_{\mathcal{P}}$ shown on the right in Figure 5 does not contain a column of zeros.

In combination with the last observation, GOODMERGINGS recursively finds all of the good mergings of $S_{\mathcal{R}}$ in which a particular component's vertex set is a subset. Intuitively, GOODMERGINGS does this by starting with a vertex set of one component of $S_{\mathcal{R}}$, and then increases the size of this set by adding the elements of another component's vertex set (one at a time) while maintaining the property of being a good merging of $S_{\mathcal{R}}$, until a maximal good merging of $S_{\mathcal{R}}$ is obtained. The fact that this works is shown after the description of GOODMERGINGS, in particular, see Lemma 4.4.

We now present GOODMERGINGS. Note that the order in which the parts appear in the input partition \mathcal{P} will be the order in which we label the rows and columns of $A_{\mathcal{P}}$.

Algorithm: GOODMERGINGS($\mathcal{P}, \mathcal{M}_{\mathcal{P}}$).

Input: A partition \mathcal{P} of a set S .

Output: A collection $\mathcal{M}_{\mathcal{P}}$ of subsets of S .

- (1) Initially set $\mathcal{M}_{\mathcal{P}}$ to be the empty set.
- (2) Let $P_1, P_2, \dots, P_{k'}$ denote (in order) the parts of \mathcal{P} .
- (3) Add P_1 to $\mathcal{M}_{\mathcal{P}}$.
- (4) If P_1 is the only part in \mathcal{P} , then go to Step 14.
- (5) Construct $A_{\mathcal{P}}$.
- (6) Let j be an element of $\{2, 3, \dots, k'\}$. Referring to $A_{\mathcal{P}}$, consider the following possible events:
 - (a) There is a 1 in each column of the block b_{j1} for which there is a 0 in the corresponding column of the block b_{11} .
 - (b) Each of the entries in the block b_{1j} is a 1.
 - (c) For every entry of the j -th row which is 0, the corresponding entry in the first row is 1.

For each j , do the following: delete the j -th row and j -th column of blocks of $A_{\mathcal{P}}$ if all of (a), (b), and (c) hold; move the j -th row and j -th column to the last row and last column, respectively, if exactly (a) and (b) hold; otherwise do nothing. Relabel the rows of the resulting matrix $P_1, P_2, \dots, P_n, P_{n+1}, \dots, P_{k''}$, where P_1 is the first row label of $A_{\mathcal{P}}$, and P_2, P_3, \dots, P_n label those rows of $A_{\mathcal{P}}$ for which at most one of (a) and (b) hold.
- (7) Set $w_2 = w_3 = \dots = w_n = 1$.
- (8) Set i' to be the least number for which $w_{i'} = 1$, and set $w_{i'} = 0$.
- (9) Set $S_{i'}$ to be the union of the sets $P_1, P_{i'}, P_{i'+1}, P_{i'+2}, \dots, P_{k''}$.
- (10) Set $\mathcal{P}_{i'}$ to be the partition $\{P_1 \cup P_{i'}, P_{i'+1}, P_{i'+2}, \dots, P_{k''}\}$ of $S_{i'}$.
- (11) GOODMERGINGS($\mathcal{P}_{i'}, \mathcal{M}_{\mathcal{P}_{i'}}$).
- (12) Replace $\mathcal{M}_{\mathcal{P}}$ with $\mathcal{M}_{\mathcal{P}} \cup \mathcal{M}_{\mathcal{P}_{i'}}$.
- (13) If $i' \neq n$, then go to Step 8.
- (14) Output $\mathcal{M}_{\mathcal{P}}$.

Remark. In conjunction with ALLMINTREES, the objective of Step 6 of GOODMERGINGS is to determine, for each $j \in \{2, 3, \dots, k'\}$, if $P_1 \cup P_j$ is a good merging of $S_{\mathcal{R}}$ and, if not, whether there is a good merging of $S_{\mathcal{R}}$ in which $P_1 \cup P_j$ is a subset. Step 6 of GOODMERGINGS recognizes these possibilities, and manipulates the matrix constructed in the previous step accordingly. If all of (a), (b), and (c) are satisfied for j , then there is no good merging of $S_{\mathcal{R}}$ in which $P_1 \cup P_j$ is a subset. If both (a) and (b) are satisfied for j , but not (c), then $P_1 \cup P_j$ is not a good merging of $S_{\mathcal{R}}$. However, in this case, it is easily checked that there is a good merging of $S_{\mathcal{R}}$ in which $P_1 \cup P_j$ is a proper subset. If neither of these two options are satisfied for j , then $P_1 \cup P_j$ is a good merging of $S_{\mathcal{R}}$.

The next two lemmas are needed for the proof of Theorem 4.6, in which we show that ALLMINTREES applied to a consistent set \mathcal{R} of rooted triples returns the set of all minimal rooted phylogenetic trees compatible with \mathcal{R} .

Lemma 4.4. *Let \mathcal{R} be a consistent set of rooted triples, and let S be a cluster of a minimal rooted phylogenetic tree compatible with \mathcal{R} . Let M_1 and M_2 be good mergings of $S_{\mathcal{R}}$ such that M_1 is a proper subset of M_2 . Then there is a subset M of $M_2 - M_1$ such that $M_1 \cup M$ is a good merging of $S_{\mathcal{R}}$ and M is the vertex set of a component of $S_{\mathcal{R}}$.*

Proof. Let N_1, N_2, \dots, N_k denote the vertex sets of the components of $S_{\mathcal{R}}$ such that, for all $i \in \{1, 2, \dots, k\}$, N_i is a subset of $M_2 - M_1$. Suppose, to the contrary, that $M_1 \cup N_i$ is not a good merging of $S_{\mathcal{R}}$ for all i . Then the number of components of $S_{\mathcal{R}}[M_1 \cup N_i]$ is equal to the number of components of $(M_1 \cup N_i)_{\mathcal{R}}$ for all i . But this implies that the number of components of $S_{\mathcal{R}}[M_2]$ is equal to the number of components of $(M_2)_{\mathcal{R}}$, contradicting that M_2 is a good merging of $S_{\mathcal{R}}$. \square

Lemma 4.5. *Let \mathcal{R} be a consistent set of rooted triples, and let S be the leaf set of \mathcal{R} . Then the set $\mathcal{M}_{S_{\mathcal{R}}}$ at Step 11 of ALLMINTREES, when applied to \mathcal{R} , is the set of all good mergings of $S_{\mathcal{R}}$.*

Proof. From the sequence of remarks preceding the description of GOODMERGINGS and the way in which $\mathcal{M}_{S_{\mathcal{R}}}$ is constructed, it is clear that every member of $\mathcal{M}_{S_{\mathcal{R}}}$ is a good merging of $S_{\mathcal{R}}$. Thus it suffices to show that if M is a good merging of $S_{\mathcal{R}}$, then M is an element of $\mathcal{M}_{S_{\mathcal{R}}}$. We now show that this is indeed the case.

Evidently, the vertex sets of each of the components of $S_{\mathcal{R}}$ are in $\mathcal{M}_{S_{\mathcal{R}}}$. Therefore we may assume that M is the union of the vertex sets of at least two components of $S_{\mathcal{R}}$. Suppose, to the contrary, that M is not an element of $\mathcal{M}_{S_{\mathcal{R}}}$ at Step 11 of ALLMINTREES. Then it follows by Lemma 4.4 that at some iteration of GOODMERGINGS there is a matrix, $A_{\mathcal{P}}$ say, constructed at Step 5 in which a subset M' of M labels the first row and each of the vertex sets of the components of $S_{\mathcal{R}}$ whose union is $M - M'$ label one of the other rows. Furthermore, one of these vertex sets labels a row of $A_{\mathcal{P}}$ that is deleted in Step 6 of GOODMERGINGS. But from the remark following the description of GOODMERGINGS this implies that M is not a good merging of $S_{\mathcal{R}}$. This contradiction completes the proof of Lemma 4.5. \square

By combining Theorem 4.2, Lemma 4.5, and the fact that the set of all rooted phylogenetic trees compatible with a consistent set of rooted triples can be obtained in the way described at the start of this section, we get Theorem 4.6.

Theorem 4.6. *Let \mathcal{R} be a consistent set of rooted triples. Then ALLMINTREES applied to \mathcal{R} returns the set of all minimal rooted phylogenetic trees compatible with \mathcal{R} .*

We now address the issue of the running time of ALLMINTREES applied to a consistent set \mathcal{R} of rooted triples. Since it is more than likely that there is a faster method for finding each tree in $\mathcal{T}_{\mathcal{R}}^{\min}$, a detailed analysis of the running time of ALLMINTREES has been omitted. The point is to show that there does exist

a method for finding each tree in $\mathcal{T}_{\mathcal{R}}^{\min}$ in polynomial time. It is easily checked that, up to the number of trees in $\mathcal{T}_{\mathcal{R}}^{\min}$, each step of ALLMINTREES can be done in polynomial time. Furthermore, no set “added” at Step 3 of GOODMERGINGS appears twice at this step during the running of ALLMINTREES and, moreover, such a set is obtained in polynomial time. It now follows from Theorem 4.2 that each tree outputted by ALLMINTREES is constructed in polynomial time.

Although each tree in $\mathcal{T}_{\mathcal{R}}^{\min}$ is outputted in polynomial time, the following example shows that the total running time of ALLMINTREES may not be polynomial.

Example 4.7. Let $\mathcal{R}_1 = \{ab|c, ac|d\}$. A routine check shows that \mathcal{R}_1 defines a binary tree T with three internal vertices, in which one of the internal vertices is adjacent to both a and b . Let u_1 and u_2 denote the other two internal vertices of T . Let $\mathcal{R}_2 = \{ab|g_1, ab|g_2, \dots, ab|g_n\}$, where n is some positive integer and g_i is not in the leaf set of T for all $i \in \{1, 2, \dots, n\}$, and consider $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$. Evidently, \mathcal{R} is a consistent set of rooted triples. Moreover, it is straightforward to deduce that the cardinality of $\mathcal{T}_{\mathcal{R}}^{\min}$ is equal to the number of ways of attaching all of the elements g_1, g_2, \dots, g_n to u_1 and u_2 via pendant edges. It follows that there are 2^n minimal rooted phylogenetic trees compatible with \mathcal{R} . Thus the cardinality of $\mathcal{T}_{\mathcal{R}}^{\min}$ is exponential in the size of \mathcal{R} , and therefore the total running time of ALLMINTREES applied to \mathcal{R} is not polynomial.

ACKNOWLEDGEMENTS

I thank Mike Steel for originally suggesting the problem of characterizing the tree outputted by the algorithm in [1] and for providing comments on an earlier draft of this paper. I thank the referees for their suggestions which has led to a significant improvement in the exposition.

REFERENCES

- [1] A. V. Aho, S. Yehoshua, T. G. Szymanski, and J. D. Ullman, Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions, *SIAM J. Comput.* **10(3)** (1981), 405–421.
- [2] D. Bryant and M. Steel, Extension operations on sets of leaf-labelled trees, *Adv. Appl. Math.* **16** (1995), 425–453.
- [3] D. Bryant, *Building trees, hunting for trees, and comparing trees: theory and methods in phylogenetic analysis*, Ph.D. Thesis, University of Canterbury, 1997.
- [4] M. Constantinescu and D. Sankoff, An efficient algorithm for supertrees, *J. Classif.* **12** (1995), 101–112.
- [5] M. R. Henzinger, V. King, and T. Warnow, Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology, *Algorithmica* **24** (1999), 1–13.
- [6] M. P. Ng and N. C. Wormald, Reconstruction of rooted trees from subtrees, *Discrete Appl. Math.* **69** (1996), 19–31.
- [7] M. Steel, The complexity of reconstructing trees from qualitative characters and subtrees, *J. Classif.* **9** (1992), 91–116.

BIOMATHEMATICS RESEARCH CENTRE, DEPARTMENT OF MATHEMATICS AND STATISTICS, UNIVERSITY OF CANTERBURY, PRIVATE BAG 4800, CHRISTCHURCH, NEW ZEALAND

E-mail address: c.semple@math.canterbury.ac.nz