Resolving Unresolved Resolved and Unresolved Triplets Consistency Problems

Daniel J. Harvey[1], Jesper Jansson[1], Mikołaj Marciniak[1(✉)], and Yukihiro Murakami[2]

[1] Graduate School of Informatics, Kyoto University, Kyoto, Japan
Daniel.Harvey87@gmail.com, jj@i.kyoto-u.ac.jp, marciniak@int.pl
[2] Delft University of Technology, Delft, The Netherlands
y.murakami@tudelft.nl

Abstract. The $\mathcal{R}^{+-}\mathcal{F}^{+-}$ Consistency problem is a basic problem related to the construction of phylogenetic trees. Its input is two sets R^+ and R^- of resolved triplets and two sets F^+ and F^- of unresolved triplets (also known as *fan triplets*). The objective of the problem is to determine if there exists a phylogenetic tree that includes all elements in $R^+ \cup F^+$ and excludes all elements in $R^- \cup F^-$ as embedded subtrees, and to construct such a tree if one exists. Jansson *et al.* [Journal of Computational Biology, 2018] cataloged the computational complexity of the problem under various restrictions, with four notable exceptions in which the output tree is required to be ternary, i.e., has degree at most three. Here, we resolve these four remaining cases by proving that for ternary trees: (i) $\mathcal{F}^+$ Consistency as well as $\mathcal{R}^+\mathcal{F}^+$ Consistency are solvable in polynomial time; and (ii) $\mathcal{F}^{+-}$ Consistency and $\mathcal{R}^+\mathcal{F}^{+-}$ Consistency are NP-hard. To obtain (i), we develop a novel way of expressing the triplets Consistency problem for ternary trees as a system of equations whose nontrivial solutions can be used to partition the leaf labels into subsets that label subtrees of the output tree. Result (ii) is obtained after observing some new equivalences between resolved triplets and fan triplets consistent with a given phylogenetic tree.

Keywords: phylogenetic tree · rooted triplets consistency · tree algorithm · computational complexity

1 Introduction

Phylogenetic trees are a key tool in evolutionary biology, used to describe evolutionary history and relationships between different species. They are also used in other fields; for example, in linguistics to model relationships between languages. An important challenge associated with phylogenetic trees is their reconstruction from different types of data [6,16]. Finding a phylogenetic tree that best describes the given data can be a difficult and, for large datasets, time-consuming

A. A. Rescigno and U. Vaccaro (Eds.): IWOCA 2024, LNCS 14764, pp. 193–205, 2024.
https://doi.org/10.1007/978-3-031-63021-7_15

task. In many cases, however, a compromise between computational efficiency and accuracy may be achieved by the *supertrees* technique [2,3]: First, using a computationally intensive method such as maximum likelihood [4,6], create a set of phylogenetic trees for small (e.g., 3-element), overlapping subsets of the leaf labels that each have a high probability of being correct. Next, using a combinatorial algorithm, merge all the small trees into a single tree.

A rooted phylogenetic tree with exactly three leaves can be either binary, in which case it is called a *resolved triplet*, or not, in which case it is called an *unresolved triplet* or a *fan triplet*. (This article will use the term fan triplet.) In the context of merging trees into a supertree, the following problem is fundamental: Determine if there exists a tree corresponding to a given collection of required and forbidden sets of resolved and fan triplets, and if so, construct such a tree. This problem has been investigated by many researchers [5,7–9,11–15,17] under different assumptions. In particular, a classic algorithm by Aho *et al.* [1] named BUILD that solves the problem for the case of required resolved triplets in polynomial time has been well studied and extended in various ways.

Jansson *et al.* [10] surveyed and cataloged the computational complexity of the many variants of the problem and obtained several new results in the process, but left four borderline variants open. The main goal of the present paper is to resolve these remaining variants, thereby completely characterizing the computational complexity of the problem when the output tree is required to have degree at most D for any integer $D \geq 2$, for all possible 15 nonempty combinations of the different types of inputs.

2 Preliminaries

Recall that a tree is a simple connected graph without cycles. A tree is *rooted* if one of its vertices has been designated as the root. The edges of a rooted tree can be assigned a natural orientation forming directed paths from the root to each leaf. If there exists a path from a vertex u to a vertex v, we say that u is an *ancestor* of v and that v is a *descendant* of u. When such a path is of length one (that is, there exists an edge from u to v) then we also say that u is the *parent* of v and that v is the *child* of u.

A *phylogenetic tree* is a rooted tree in which each leaf is given a unique label and each internal vertex has at least two children. Additionally, we will also treat the degenerate cases of a tree with a single vertex and the empty tree as phylogenetic trees. For simplicity, we will refer to phylogenetic trees as trees and identify each leaf with its label. For any (phylogenetic) tree T, we will denote the set of its leaves (labels) by L_T. For any two leaves $u, v \in L_T$, we denote by $\mathrm{lca}^T(u, v)$ their lowest common ancestor, that is, the vertex w that is an ancestor of both u and v such that no child of w is also an ancestor to both u and v.

Here we let the *degree of a vertex* be the number of its children, and the *degree of a tree* be the maximum degree of a vertex taken over all vertices of the tree. A *binary tree* is a tree in which the degree of each vertex is at most 2, and a *ternary tree* is a tree in which the degree of each vertex is at most 3.

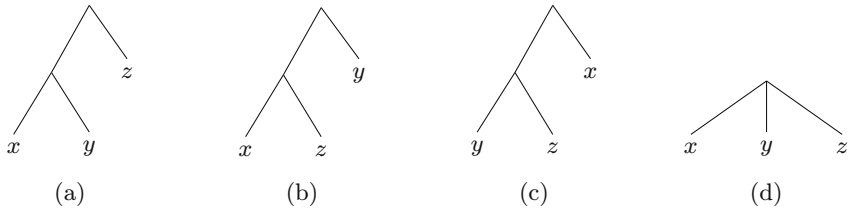

Fig. 1. All possible rooted triplets with the leaf set $\{x, y, z\}$ – resolved triplets (a) $xy|z$, (b) $xz|y$, and (c) $yz|x$ and fan triplet (d) $x|y|z$.

A *rooted triplet*, or *triplet* for short, is a tree with exactly three leaves. Let t be a rooted triplet and suppose that $L_t = \{x, y, z\}$. If t is a binary tree, then t is called a *resolved triplet* and we write $xy|z$, where $\mathrm{lca}^t(x, y)$ is a proper descendant of $\mathrm{lca}^t(x, z) = \mathrm{lca}^t(y, z)$. Otherwise, the triplet t is called a *fan triplet* and we write $x|y|z$. Note that there exist only four triplets (see Fig. 1) with a fixed set of leaves $\{x, y, z\}$, namely $xy|z, xz|y, yz|x$, and $x|y|z$.

Given a tree T and three distinct leaves $\{x, y, z\} \subseteq L_T$, the resolved triplet $xy|z$ is *consistent with* T if and only if $\mathrm{lca}^T(x, y)$ is a proper descendant of $\mathrm{lca}^T(x, z) = \mathrm{lca}^T(y, z)$. Likewise, the fan triplet $x|y|z$ is *consistent with* T if and only if $\mathrm{lca}^T(x, y) = \mathrm{lca}^T(x, z) = \mathrm{lca}^T(y, z)$. (Equivalently, we may say in each case that the tree is consistent with the triplet.) In other words, the rooted triplet consistent with T describes the relative position of its three leaves in T.

We will say that a tree T is *over* a set of leaves L if each leaf of T belongs to L. Let T be a tree. We define the *restriction* of the tree T into the set $L' \subseteq L_T$, denoted by $T|_{L'}$, to be the tree T' with $L_{T'} = L'$ such that each triplet t consistent with the tree T' is also consistent with the tree T. Less formally, $T|_{L'}$ can be obtained by "removing" all leaves $x \notin L'$ not belonging to the set L'. In the special case where the set L' has only 3 elements, a tree $t = T|_{L'}$ is a rooted triplet over L' consistent with the tree T. Let t_T denote the set of all resolved and fan triplets consistent with T, i.e., $t_T = \left\{T|_{\{x,y,z\}} : \{x, y, z\} \subseteq L_T\right\}$. Using the new notation, we can note that $T' = T|_{L'}$ is the restriction of tree T into the set of leaves $L' \subseteq L_T$ if and only if $t_{T'} \subseteq t_T$.

To illustrate, let

$$F = \{1|2|3, 1|4|5, 1|6|7, 2|4|7, 2|5|6, 3|4|6\}$$

be a set of fan triplets over the leaf set $\{1, 2, \ldots, 7\}$. There is no ternary tree T with $F \subseteq t_T$ (and we *strongly recommend* the reader to try to prove it), but if the last fan triplet $3|4|6 \in F$ is replaced by $3|4|\mathbf{7}$ then the situation changes; more precisely, the resulting set is consistent with a ternary tree with three maximal proper subtrees leaf-labeled by $\{4, 6\}$, $\{1, 2, 3\}$, and $\{5, 7\}$, respectively. It may seem to the reader that the only way to distinguish between such cases is through an optimized brute-force check of all possibilities. However, we will demonstrate how to perform this task in polynomial time. We hope that after reading this paper, the reader will appreciate the usefulness of the newly developed theory.

The general problem considered in this paper is as follows.

The $\mathcal{R}^{+-}\mathcal{F}^{+-}$ Consistency Problem. *Given two sets R^+ and R^- of resolved triplets and two sets F^+ and F^- of fan triplets over a set of leaves L, either return a tree with $L_T = L$ such that $R^+ \bigcup F^+ \subseteq t_T$ and $(R^- \bigcup F^-) \bigcap t_T = \emptyset$ if such a tree exists, or return the answer null if no such tree exists.*

In other words, the output tree must contain all triplets from the sets R^+ and F^+, and must not contain any triplets from the sets R^- or F^-. Different variants of the problem arise when some of the given sets are forced to be empty. In such cases, we name the problem by omitting the symbols from $\mathcal{F}, \mathcal{R}, +, -$ corresponding to the empty sets. For example, the $\mathcal{R}^+$ Consistency problem is the variant of the $\mathcal{R}^{+-}\mathcal{F}^{+-}$ Consistency problem where $R^- = F^+ = F^- = \emptyset$.

Tables 1 to 4 below (all entries can be found in [10]), show the computational complexity of the 15 variants of the $\mathcal{R}^{+-}\mathcal{F}^{+-}$ Consistency problem when the output tree has no degree limitations and when the output tree is required to have outdegree at most D for $D = 2$, $D = 3$, and any fixed integer $D \geq 4$, respectively. In the tables, P means that the corresponding problem is polynomial-time solvable, while NP-h means that the corresponding problem is NP-hard. For example, the $\mathcal{R}^-\mathcal{F}^-$ Consistency problem is NP-hard. Furthermore, a question mark indicates that the computational complexity was unknown.

As can be seen in Tables 1 to 4, the problem was almost fully solved except four variants in the borderline case of $D = 3$. For the remaining four variants, i.e., $\mathcal{F}^+$, $\mathcal{R}^+\mathcal{F}^+$, $\mathcal{F}^{+-}$, and $\mathcal{R}^+\mathcal{F}^{+-}$, the computational complexity remained open. In the rest of the paper, we resolve the last four remaining variants by either finding a polynomial-time solution or an NP-hardness proof for each one, thereby completely characterizing the computational complexity of the problem when the output tree is required to have degree at most D for any integer $D \geq 2$, for all possible 15 nonempty combinations of the different types of inputs.

Table 1. Unbounded degree case

	$\emptyset$	$\mathcal{F}^+$	$\mathcal{F}^-$	$\mathcal{F}^{+-}$
$\emptyset$	–	P	P	NP-h
$\mathcal{R}^+$	P	P	P	NP-h
$\mathcal{R}^-$	P	P	NP-h	NP-h
$\mathcal{R}^{+-}$	P	P	NP-h	NP-h

Table 2. Binary case

	$\emptyset$	$\mathcal{F}^+$	$\mathcal{F}^-$	$\mathcal{F}^{+-}$
$\emptyset$	–	P	P	P
$\mathcal{R}^+$	P	P	P	P
$\mathcal{R}^-$	NP-h	NP-h	NP-h	NP-h
$\mathcal{R}^{+-}$	NP-h	NP-h	NP-h	NP-h

Table 3. Ternary case

	$\emptyset$	$\mathcal{F}^+$	$\mathcal{F}^-$	$\mathcal{F}^{+-}$
$\emptyset$	–	?	P	?
$\mathcal{R}^+$	P	?	P	?
$\mathcal{R}^-$	NP-h	NP-h	NP-h	NP-h
$\mathcal{R}^{+-}$	NP-h	NP-h	NP-h	NP-h

Table 4. D-bounded degree case, $D \geq 4$

	$\emptyset$	$\mathcal{F}^+$	$\mathcal{F}^-$	$\mathcal{F}^{+-}$
$\emptyset$	–	NP-h	P	NP-h
$\mathcal{R}^+$	P	NP-h	P	NP-h
$\mathcal{R}^-$	NP-h	NP-h	NP-h	NP-h
$\mathcal{R}^{+-}$	NP-h	NP-h	NP-h	NP-h

3 The $\mathcal{F}^+$ Consistency Problem for a Ternary Tree

This section focuses on the following variant of the Consistency problem.

The Ternary $\mathcal{F}^+$ Consistency Problem. *For a given set of fan triplets $F = \{a_1|b_1|c_1, \ldots, a_n|b_n|c_n\}$ over a set of leaves L, return a ternary tree T with $L_T = L$ such that $F \subseteq t_T$ if such a tree exists, or return the answer null if it does not.*

Below, we will prove the following theorem.

Theorem 1. *The ternary $\mathcal{F}^+$ Consistency problem is solvable in polynomial time.*

For each fan triplet $a|b|c \in F$, we identify it with an integer domain modulo 3 equation $w_a + w_b + w_c \equiv^3 0$. The set F can thus be identified with the following system of integer domain equations:

$$\begin{cases} w_{a_1} + w_{b_1} + w_{c_1} \equiv^3 0, \\ \qquad\qquad\qquad\quad \vdots \\ w_{a_n} + w_{b_n} + w_{c_n} \equiv^3 0. \end{cases}$$

The above system of equations can be represented as a matrix equation $M(F, L)\overrightarrow{\mathrm{w}} \equiv^3 \overrightarrow{0}$, where $\overrightarrow{\mathrm{w}} = (w_j)_{j \in L}$ and the matrix $M(F, L)$ has $|F| = n$ rows and $|L|$ columns, with entries from the set $\{0, 1, 2\}$ and initially equal to 0 or 1. Specifically, $M(F, L)_{i,j} = 1$ if the leaf j appears in the i-th equation, that is, if $j \in \{a_i, b_i, c_i\}$; otherwise, the matrix element is equal to 0.

The solution to a system of equations will be called *trivial* if all the variables are equal. Otherwise, if at least two variables are different, a solution will be called *nontrivial*. Consider the following simple yet crucial observation.

Observation 1. *The equality $w_a + w_b + w_c \equiv^3 0$ holds if and only if either all variables are distinct $\{w_a, w_b, w_c\} = \{0, 1, 2\}$ or all are equal $w_a = w_b = w_c$.*

Proof. The three variables w_a, w_b, and w_c can take one of three values: $0, 1$, or 2. If the three variables are not all different, then at least two of them are equal; without loss of generality, let $w_a = w_b$. Then

$$w_c \equiv^3 -w_a - w_b \equiv^3 -2w_a \equiv^3 w_a.$$

Hence if the three variables are not all different, then they are all equal. □

Let T be a ternary tree consistent with a set of fan triplets F. The root vertex has at most 3 children and each child forms a separate subtree. Each triplet $a|b|c \in F$ can have all leaves in different subtrees or all in the same subtree. This property corresponds to the solution of the equation $w_a + w_b + w_c \equiv^3 0$. Intuitively, the value of the variable w_j corresponds to the index ($0, 1$, or 2) of the subtree in which the leaf j is located. More generally, we can formulate the following lemma.

Lemma 1. *Let $F \neq \emptyset$ be a set of fan triplets over a set of leaves L. There exists a ternary tree T with the leaf set $L_T = L$ such that T is consistent with F if and only if the equation $M(F,L)\overrightarrow{\mathrm{w}} \equiv^3 \overrightarrow{0}$ has a nontrivial solution $\overrightarrow{\mathrm{w}} = (w_j)_{j \in L}$ and there exist ternary trees T_0, T_1, and T_2 such that for each $i \in \{0,1,2\}$, the tree T_i with the set of leaves $L_{T_i} = L_i$ is consistent with F_i, where*

$$L_i = \{j : w_j = i\},$$
$$F_i = \{a|b|c \in F : a, b, c \in L_i\}.$$

Proof. Let $F \neq \emptyset$ be a set of fan triplets over L, and let T be a ternary tree with the leaf set $L_T = L$ such that T is consistent with F. Let T_0, T_1, and T_2 denote the subtrees of the root vertex in the tree T. For each leaf d located in the i-th subtree, we put $w_d = i$. Note that each triplet $a|b|c \in F$ has all leaves located either in different subtrees, in which case $\{w_a, w_b, w_c\} = \{0,1,2\}$, or all located in the same subtree, in which case $w_a = w_b = w_c$. In both cases, by Observation 1, we have the equality $w_a + w_b + w_c \equiv^3 0$. We have an identical equality for all triplets in F, and it follows that $\overrightarrow{\mathrm{w}}$ is a solution to the equation $M(F,L)\overrightarrow{\mathrm{w}} \equiv^3 \overrightarrow{0}$. Since $F \neq \emptyset$, the root vertex of T must have at least two children. Therefore, at least two of the trees T_0, T_1, and T_2 are non-empty, and thus the obtained solution is nontrivial. Since T is consistent with F, then trees T_0, T_1, T_2 are consistent consecutively with the sets F_0, F_1, F_2 by their definition.

Let $F \neq \emptyset$ be a set of fan triplets. Suppose we have given a nontrivial solution to the matrix equation $M(F,L)\overrightarrow{\mathrm{w}} \equiv^3 \overrightarrow{0}$ and let T_0, T_1, T_2 be ternary trees such that each T_i is consistent with F_i and $L_{T_i} = L_i$, where L_i, F_i for $i \in \{0,1,2\}$ are defined as in the lemma statement. Since we have a nontrivial solution, at least two of the sets L_i are non-empty, meaning that at least two of the trees T_i are non-empty. We create a tree T by taking a root vertex and adding the trees T_0, T_1, T_2 as subtrees of the root. Note that for each triplet $a|b|c \in F$ the equality $w_a + w_b + w_c \equiv^3 0$ holds and by Observation 1 we obtain that $\{w_a, w_b, w_c\} = \{0,1,2\}$ or $w_a = w_b = w_c$. In the first case, the leaves a, b, and c are located in distinct subtrees of T. Then $\mathrm{lca}^T(a,b) = \mathrm{lca}^T(b,c) = \mathrm{lca}^T(a,c)$ is the root vertex of the tree T, and thus the triplet $a|b|c$ is consistent with T. In the second case, the triplet $a|b|c$ is consistent with some tree T_i and therefore also with T. Therefore, all triplets in F are consistent with T. □

On the implementation side, we will represent each inner vertex of a ternary tree as a list of (two or three) pointers to its children, and a tree as a pointer to the root vertex. We now introduce the following auxiliary functions.

- SOLVE(F, L) – finds a nontrivial solution to a system of equations identified with a set of triplets F in variables from the set L and divides the variables into three sets according to their values. In the first step, we create the matrix $M(F,L)$. The equation we wish to solve is $M(F,L)\overrightarrow{\mathrm{w}} \equiv^3 \overrightarrow{0}$. Using Gauss–Jordan row elimination, we reduce the matrix to reduced row echelon form. In this way, the variables were divided into two subsets, that is, the dependent variables corresponding to the columns with leading ones and the independent variables corresponding to the remaining columns. Any solution

can be obtained by fixing any values of the independent variables and calculating from them the values of the dependent variables. When there are at least two independent variables, we find a nontrivial solution by fixing at least two different values to the independent variables. Otherwise, when there is at most one independent variable, we find all solutions by considering all possible values for that potential variable. When there exists a nontrivial solution $\overrightarrow{w} = (w_j)_j \in L$, the function finds and returns a triple of sets (L_0, L_1, L_2), where $L_i = \{j : w_j = i\}$. Otherwise, the function returns null.

- DIVIDE(F, L) – returns all triplets from the set F where all leaves of the triplet belong to the set L.
- NEWVERTEX() – returns a pointer to the newly created vertex.
- NEWEDGE(T, V) – creates at vertex T an edge to vertex V.
- ADDMISSINGVERTICES(T, L) – adds the missing vertices from the set L to the tree T in an arbitrary way such that the modified tree still remains a ternary tree. (Later, we will provide an example of such an addition.)

The first function is dominated by Gauss–Jordan row elimination. Since $2 \equiv^3 -1$, we will never have to use division, and thus SOLVE(F, L) can be implemented in $O(|L|^2|F|)$ time using a brute-force approach. All other auxiliary functions can be implemented in linear time.

Now, using Lemma 1 and the divide-and-conquer method we present an algorithm for the ternary $\mathcal{F}^+$ CONSISTENCY problem. When $|F| > 0$, we apply Lemma 1 and divide the original problem into two or three smaller instances. Otherwise, in the base case, when $|F| = 0$, we return any tree as the answer (for example, by creating an empty tree and adding missing leaves to it). The pseudocode of our algorithm is summarized in FANTERNARYBUILD.

On a high level, our new FANTERNARYBUILD algorithm, just like Aho *et al.*'s BUILD algorithm from [1], uses the natural divide-and-conquer strategy on the set of leaves. Both algorithms construct a tree by partitioning the set of leaves into blocks containing leaves belonging to the same subtree. They then recursively find a solution for each block and finally attach the recursively found subtrees as children to a new root vertex. If, at any point during execution, the current set of leaves contains more than one leaf but cannot be divided into more than one block, the algorithms stop and return null.

The crucial aspect is the method of partitioning the leaf set. The FANTERNARYBUILD algorithm creates a system of modulo 3 equations defined by the input triplets, solves it, and then assigns all leaves whose variables have the same value to one block of the partition. In comparison, BUILD creates an undirected graph whose vertices represent the leaves of the constructed tree, with edges defined by the input triplets. It then computes the connected components of the graph and assigns the leaves in each connected component to one block of the partition.

Algorithm FanTernaryBuild

Input: a set of leaves L, and a set of fan triplets F over L.
Output: a tree T consistent with the set F or null if no such tree exists.

```
1: function FanTernaryBuild(F, L)
2:     T = ∅
3:     if |F| > 0 then
4:         (L0, L1, L2) = solve(F, L)
5:         if (L0, L1, L2) = null then return null
6:         T=newVertex()
7:         for i = 0, 1, 2 do
8:             Fi = divide(F, Li)
9:             Ti = FanTernaryBuild(Fi, Li)
10:            if Ti =null then return null
11:            if |Li| > 0 then newEdge(T, Ti)
12:    addMissingVertices(T, L)
13:    return T
```

The computational complexity of FanTernaryBuild is $O(|L|^3|F|)$, since the output tree has at most $|L|-1$ inner vertices, and $O(|L|^2|F|)$ operations are performed at each inner vertex. Thus the ternary $\mathcal{F}^+$ Consistency problem has a polynomial-time solution, proving Theorem 1.

Equipped with the FanTernaryBuild algorithm, the reader can directly show that the set F in the example in Sect. 2 has no solution and also easily find a ternary tree that is consistent with the modified F.

4 The Remaining Ternary Consistency Problems

To determine the exact boundary between NP-hard versions of the ternary Consistency problem and those with a polynomial time solution, we now study the remaining variants. In what follows, we will often need to modify trees by adding or removing leaves. We will start with the simple observation that adding and removing leaves can be done in constant time and does not affect the other triplets unrelated to the added or removed leaves.

Observation 2. *Let T be a ternary tree over a set of leaves L, and let t be a triplet over $\{a, b, x\}$, where $a, b \in L_T$ are leaves and $x \notin L_T$ is not a leaf of T. We can add the leaf x to the tree T in constant time, thus obtaining a new ternary tree T' such that $t \in t_{T'}$.*

Similarly, for each leaf x in $L_{T'}$ of a given ternary tree T', we can remove it in constant time, thus obtaining a new ternary tree T over $L_{T'} \setminus \{x\}$.

Moreover, in both addition and removal, the triplets not containing leaf x will remain unchanged, that is, $t_T \subseteq t_{T'}$.

Proof. We will first consider adding a new leaf x to a tree T. Let $u = \mathrm{lca}^T(a, b)$ be the lowest common ancestor of the vertices $a, b \in L_T$. Consider the first case

when $t = ab|x$. If u is the root of the tree T, then we create a new vertex w as the new root of the tree T and then add x and the previous root as a children of w. Otherwise, we subdivide the edge from u to its parent to create a new vertex w, and add x as a child of w. Now consider the second case when $t = ax|b$. Let c be the child of u that is an ancestor of a. Subdivide the edge from u to c to create a new vertex w, and again add x as a child of w. The case $t = bx|a$ can be realized by symmetry. Finally consider the last case when $t = a|b|x$. If u has degree 2, then add x as a child of u. Otherwise, let c be the child of the vertex u which is neither an ancestor of a nor b. Subdivide the edge from u to c to create a new vertex w, and add x as a child of w.

Now consider the removal of a leaf. Let w be the parent of a leaf x. We remove x and its incident edge. If after removing x, the vertex w has degree 1, we need to remove the vertex w. If w is the root, then we remove w and its incident edge, setting a unique child of w as the new root. Otherwise, we remove the vertex w by replacing the two-edge path through w with a single edge.

In both adding and removing a leaf x, the relative positions of the other leaves remain unchanged. Thus, all triplets not containing x will remain unchanged. □

Remark. The most well-known algorithmic definition of tree restriction uses the above method for leaf removal. However, since each tree is uniquely defined by the set of its all triplets, the definitions are equivalent.

We will formulate two lemmas. The first lemma shows an equivalence between consistency with a resolved triplet and a related fan triplet.

Lemma 2. *Let T be a ternary tree such that $a, b, c, x \in L_T$ and T is consistent with fan triplet $a|c|x$. The tree T is consistent with the resolved triplet $p = ab|c$ if and only if the fan triplet $q = b|c|x$ is consistent with T.*

Proof. First, we will prove the theorem in the base case when $L_T = \{a, b, c, x\}$. Let L_0, L_1, and L_2 be the sets of leaves of the three connected components of T minus the root and its edges. As we already know from Sect. 3, the sets L_0, L_1, and L_2 correspond to a nontrivial solution of the system of equations corresponding to the set of fan triplets. Then, the variables w_a, w_b and w_c satisfy $w_a + w_c + w_x \equiv^3 0$, since $a|c|x$ is consistent with T.

If T is consistent with $b|c|x$, then $w_b \equiv^3 -w_c - w_x \equiv^3 w_a$ by Observation 1.

If T is consistent with $ab|c$, then also $w_a \equiv^3 w_b$.

In both cases, since the solution is nontrivial, by Observation 1 we obtain $\{w_a = w_b, w_c, w_x\} = \{0, 1, 2\}$. Thus T is consistent with both $ab|c$ and $b|c|x$.

Now we will prove the theorem for any tree. Let T be a tree consistent with the triplet $a|c|x$. Let $T' = T|_{\{x,a,b,c\}}$ be a restriction of T. It follows that $t_{T'} \subseteq t_T$. If the triplet p is consistent with T, then p is also consistent with T' because $t_{T'}$ consists of a triplet over the set of leaves $\{a, b, c\}$. Using the result for the base case, we obtain $q \in t_{T'} \subseteq t_T$. Symmetrically, if $q \in t_T$, then $q \in t_{T'}$, and thus $p \in t_{T'} \subseteq t_T$. □

The following lemma is a generalization of the preceding one and allows for the conversion of any triplet into another chosen unrelated triplet.

Lemma 3. *Let T be a ternary tree such that $a, b, c, x, y, z \in L_T$ and T is consistent with the set of fan triplets $F = \{a|c|x, a|b|y, a|x|z\}$. Define the triplets*

$$p_1 = ab|c, \quad p_2 = ac|b, \quad p_3 = bc|a, \quad p_4 = a|b|c;$$
$$q_1 = x|y|z, \quad q_2 = xz|y, \quad q_3 = xy|z, \quad q_4 = yz|x.$$

Then, for any index j, the tree T is consistent with the triplet p_j if and only if the triplet q_j is consistent with the tree T.

Proof. First, we will present a proof in the base case when $L_T = \{x, y, z, a, b, c\}$. Let L_0, L_1, and L_2 be the sets of leaves of the three connected components of T minus the root and its edges. As we already know from Sect. 3, the sets L_0, L_1, and L_2 correspond to a nontrivial solution to the equation $M(F, L)\overrightarrow{w} \equiv^3 \overrightarrow{0}$. By subtracting the row corresponding to $a|c|x$ from the row corresponding to $a|x|z$, we obtain the system of equations

$$\begin{cases} w_x \equiv^3 2w_a + 2w_c, \\ w_y \equiv^3 2w_a + 2w_b, \\ w_z \equiv^3 w_c. \end{cases}$$

Each solution can be determined by fixing the values of the independent variables w_a, w_b, w_c and calculating values the dependent variables w_x, w_y, w_z. Thus, there exist $3^3 = 27$ solutions. Consider the following four nontrivial solutions of the above equation system:

1. $L_0 = \{x\}$, $L_1 = \{a, b, y\}$, $L_2 = \{c, z\}$;
2. $L_0 = \{y\}$, $L_1 = \{a, c, x, z\}$, $L_2 = \{b\}$;
3. $L_0 = \{x, y\}$, $L_1 = \{a\}$, $L_2 = \{b, c, z\}$;
4. $L_0 = \{c, y, z\}$, $L_1 = \{a\}$, $L_2 = \{b, x\}$.

It can be easily checked that the j-th solution provides consistency with the triplets p_j and q_j. For each of these solutions we can construct $3! - 1 = 5$ additional solutions of the system by permuting the indices. Additionally, by setting $w_a = w_b = w_c$ we obtain all 3 trivial solutions. This gives a total of $4 * 6 + 3 = 27$ solutions, and hence there are no other solutions. Thus the lemma holds in this base case.

Now we will prove the theorem for any tree. Let T be a tree consistent with F. Let $T' = T|_{\{x,y,z,a,b,c\}}$ be a restriction of T. It follows that $t_{T'} \subseteq t_T$. Let $p_j \in t_T$ be a triplet consistent with T. Thus p_j is also consistent with T' because $t_{T'}$ consists of a triplet over the set of leaves $\{a, b, c\}$. Using the result for the base case, we obtain $q_j \in t_{T'} \subseteq t_T$. Symmetrically, if $q_j \in t_T$, then $q_j \in t_{T'}$, and thus $p_j \in t_{T'} \subseteq t_T$. □

Using Lemmas 2 and 3, we prove two theorems concerning the computational complexity of two ternary Consistency problems.

Theorem 2. *The ternary $\mathcal{R}^+\mathcal{F}^+$ Consistency problem is solvable in polynomial time.*

Proof. Let F^+ be the set of fan triplets over L and R^+ be the set of resolved triplets over L. For each triplet $ab|c \in R^+$, we create a new additional leaf label $x = x(a, b, c) \notin L$. We convert the sets of resolved triplets in R^+ and fan triplets in F^+ into a new set of fan triplets

$$F = F^+ \cup \{a|c|x, b|c|x : ab|c \in R^+\}$$

over an extended set of leaves $L_F = L \cup \{x : ab|c \in R^+\}$.

We will prove that the ternary $\mathcal{R}^+\mathcal{F}^+$ Consistency problem for sets of triplets R^+ and F^+ over L is equivalent to the ternary $\mathcal{F}^+$ Consistency problem for a set of triplets F over L_F.

Let T be a tree over L that is consistent with F^+ and R^+. For each triplet $ab|c \in R^+$, we define a new leaf $x = x(a, b, c)$, and using Observation 2, we add the new leaf x to the tree T so that the resulting tree T' is consistent with $a|c|x$. By Lemma 2 the tree T' is consistent with the triplet $b|c|x$. We repeat this for all other elements of R^+ to obtain a tree that is consistent with F.

Conversely, let T' be a tree over L_F that is consistent with F. Obviously, the tree T' is consistent with F^+ and by Lemma 2, the tree T' is also consistent with R^+. Remove all leaves $x \in L_F \backslash L$ to obtain a tree T over L. By Observation 2, T remains consistent with R^+ and F^+.

By Theorem 1, the ternary $\mathcal{F}^+$ Consistency problem has a polynomial time solution. Since we can convert every instance of the ternary $\mathcal{R}^+\mathcal{F}^+$ Consistency problem into an equivalent instance of the ternary $\mathcal{F}^+$ Consistency in polynomial time, the ternary $\mathcal{R}^+\mathcal{F}^+$ Consistency problem is also in P. □

Theorem 3. *The ternary $\mathcal{F}^{+-}$ Consistency problem is NP-hard.*

Proof. We give a polynomial-time reduction from the ternary $\mathcal{R}^-$ Consistency problem, which is known to be NP-hard. Let R^- be a set of resolved triplets over L. For each triplet $ab|c \in R^-$, we create three new additional leaf labels $x = x(a, b, c)$, $y = y(a, b, c)$, and $z = z(a, b, c)$. We convert the set of resolved triplets R^- into two new sets of fan triplets

$$F^+ = \{a|c|x, a|b|y, a|x|z : ab|c \in R^-\}$$
$$F^- = \{x|y|z : ab|c \in R^-\}$$

over an extended set of leaves $L_F = L \cup \{x, y, z : ab|c \in R^-\}$.

We will prove that the ternary $\mathcal{R}^-$ Consistency problem for a set of triplets R^- over L is equivalent to the ternary $\mathcal{F}^{+-}$ Consistency problem for sets of fan triplets F^+ and F^- over L_F.

Let T be a tree over L that is not consistent with R^-. For each triplet $ab|c \in R^-$, we invoke Observation 2 to successively add leaves x, y, and z to T so that the resulting tree T' is consistent with $a|c|x$, $a|b|y$, and $a|x|z$. By Lemma 3, since T' is not consistent with $ab|c$, it is also not consistent with $x|y|z$. We

repeat this for all other elements of R^- and obtain a tree that is consistent with F^+ and not consistent with F^-.

Conversely, let T' be a tree over L_F that is consistent with F^+ and not consistent with F^-. By Lemma 3, T' is not consistent with R^-. Remove all leaves in $L_F \backslash L$ to obtain a tree T. By Observation 2, T is not consistent with the set R^-, since $T = T'|_L$ and $t_T \subseteq t_{T'}$.

Thus, the ternary $\mathcal{F}^{+-}$ Consistency problem is NP-hard, because the ternary $\mathcal{R}^-$ Consistency problem is NP-hard. □

Since the ternary $\mathcal{R}^+\mathcal{F}^{+-}$ Consistency problem is a generalization of the ternary $\mathcal{F}^{+-}$ Consistency problem, the following result is immediate.

Corollary 1. *The ternary $\mathcal{R}^+\mathcal{F}^{+-}$ Consistency problem is NP-hard.*

5 Conclusion

Finally, we summarize our new findings and complete the classification.

Proposition 1. *The computational complexity of all 15 variants of the ternary $\mathcal{R}^{+-}\mathcal{F}^{+-}$ Consistency problem is as presented in Table 5.*

Proof. The variants $\mathcal{F}^+$ and $\mathcal{R}^+\mathcal{F}^+$ have polynomial-time solutions by Theorem 1 and its generalization Theorem 2. By [10, Corollary 1], the variant $\mathcal{R}^+\mathcal{F}^-$, and thereby also $\mathcal{R}^+$ and $\mathcal{F}^-$, have polynomial-time solutions, too. In contrast, using Theorem 3 and its consequent Corollary 1, we obtain that the variant $\mathcal{F}^{+-}$, and hence $\mathcal{R}^+\mathcal{F}^{+-}$, are NP-hard. All other variants are NP-hard since the variant $\mathcal{R}^-$ is NP-hard according to [10, Theorem 3]. □

Table 5. Completed version of Table 3, presenting the computational complexity of all variants of the ternary Consistency problem. As before, P means that the corresponding problem is polynomial-time solvable, while NP-h means that the problem is NP-hard.

	$\emptyset$	$\mathcal{F}^+$	$\mathcal{F}^-$	$\mathcal{F}^{+-}$
$\emptyset$	–	P	P	NP-h
$\mathcal{R}^+$	P	P	P	NP-h
$\mathcal{R}^-$	NP-h	NP-h	NP-h	NP-h
$\mathcal{R}^{+-}$	NP-h	NP-h	NP-h	NP-h

Acknowledgments. This work was partially funded by JSPS KAKENHI grant 22H03550. Mikołaj Marciniak was additionally supported by Narodowe Centrum Nauki, Grant Number 2017/26/A/ST1/00189 and Narodowe Centrum Badań i Rozwoju, Grant Number POWR.03.05.00-00-Z302/17-00.

References

1. Aho, A.V., Sagiv, Y., Szymanski, T.G., Ullman, J.D.: Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. SIAM J. Comput. **10**(3), 405–421 (1981)
2. Bininda-Emonds, O.R.P., et al.: The delayed rise of present-day mammals. Nature **456**, 274 (2008)
3. Bininda-Emonds, O.R.P.: The evolution of supertrees. Trends Ecol. Evol. **19**(6), 315–322 (2004)
4. Chor, B., Hendy, M., Penny, D.: Analytic solutions for three taxon ML trees with variable rates across sites. Discrete Appl. Math. **155**(6), 750–758 (2007). Computational Molecular Biology Series, Issue V
5. Constantinescu, M., Sankoff, D.: An efficient algorithm for supertrees. J. Classif. **12**, 101–112 (1995)
6. Felsenstein, J.: Inferring Phylogenies. Sinauer (2003)
7. Guillemot, S., Jansson, J., Sung, W.-K.: Computing a smallest multi-labeled phylogenetic tree from rooted triplets. IEEE/ACM Trans. Comput. Biol. Bioinform. **8**, 1141–1147 (2009)
8. He, Y.-J., Huynh, T.N.D., Jansson, J., Sung, W.-K.: Inferring phylogenetic relationships avoiding forbidden rooted triplets. J. Bioinform. Comput. Biol. **4**, 59–74 (2006)
9. Huber, K., van Iersel, L., Moulton, V., Scornavacca, C., Wu, T.: Reconstructing phylogenetic level-1 networks from nondense binet and trinet sets. Algorithmica **77**, 173–200 (2014)
10. Jansson, J., Lingas, A., Rajaby, R., Sung, W.-K.: Determining the consistency of resolved triplets and fan triplets. J. Comput. Biol. **25**(7), 740–754 (2018). PMID: 29451395
11. Jansson, J., Nguyen, N.B., Sung, W.-K.: Algorithms for combining rooted triplets into a galled phylogenetic network. SIAM J. Comput. **35**, 1098–1121 (2006)
12. Ng, M.P., Wormald, N.C.: Reconstruction of rooted trees from subtrees. Discrete Appl. Math. **69**(1–2), 19–31 (1996)
13. Semple, C.: Reconstructing minimal rooted trees. Discrete Appl. Math. **127**(3), 489–503 (2003)
14. Semple, C., Daniel, P., Hordijk, W., Page, R., Steel, M.: Supertree algorithms for ancestral divergence dates and nested taxa. Bioinformatics **20**, 2355–2360 (2004)
15. Snir, S., Rao, S.: Using max cut to enhance rooted trees consistency. IEEE/ACM Trans. Comput. Biol. Bioinf. **3**, 323–333 (2006)
16. Sung, W.-K.: Algorithms in Bioinformatics: A Practical Introduction. Chapman & Hall/CRC (2009)
17. Willson, S.: Constructing rooted supertrees using distances. Bull. Math. Biol. **66**, 1755–1783 (2004)