Displaying trees across two phylogenetic networks

Janosch Döcker[a], Simone Linz[b], Charles Semple[c]

[a] *Department of Computer Science, University of Tübingen, Germany*

[b] *School of Computer Science, University of Auckland, New Zealand*

[c] *School of Mathematics and Statistics, University of Canterbury, New Zealand*

Abstract

Phylogenetic networks are a generalization of phylogenetic trees to leaf-labeled directed acyclic graphs that represent ancestral relationships between species whose past includes non-tree-like events such as hybridization and horizontal gene transfer. Indeed, each phylogenetic network embeds a collection of phylogenetic trees. Referring to the collection of trees that a given phylogenetic network $\mathcal{N}$ embeds as the display set of $\mathcal{N}$, several questions in the context of the display set of $\mathcal{N}$ have recently been analyzed. For example, the widely studied TREE-CONTAINMENT problem asks if a given phylogenetic tree is contained in the display set of a given network. The focus of this paper are two questions that naturally arise in comparing the display sets of two phylogenetic networks. First, we analyze the problem of deciding if the display sets of two phylogenetic networks have a tree in common. Surprisingly, this problem turns out to be NP-complete even for two temporal normal networks. Second, we investigate the question of whether or not the display sets of two phylogenetic networks are equal. While we recently showed that this problem is polynomial-time solvable for a normal and a tree-child network, it is computationally hard in the general case. In establishing hardness, we show that the problem is contained in the second level of the polynomial-time hierarchy. Specifically, it is Π_2^P-complete. Along the way, we show that two other problems are also Π_2^P-complete, one of which being a generalization of TREE-CONTAINMENT.

Keywords: Phylogenetic network, TREE-CONTAINMENT, polynomial-time hierarchy, display set, temporal network, normal network

1. Introduction

In trying to disentangle the evolutionary history of species, phylogenetic networks, which are leaf-labeled directed acyclic graphs, are becoming increasingly important. From a biological as well as from a mathematical viewpoint, phylogenetic networks are often regarded as a tool to summarize a collection of conflicting phylogenetic trees. Due to processes such as hybridization and lateral gene transfer, the evolution at the species-level is not necessarily tree-like. Nevertheless, individual genes or parts thereof are usually assumed to evolve in a tree-like way. It is consequently of interest to construct phylogenetic networks that embed a collection of phylogenetic trees or, conversely, summarize the phylogenetic trees that are embedded in a given phylogenetic network. These and related types of problems have recently attracted considerable attention from the mathematical community as they lead to a number of challenging questions. One of the most studied questions in this context is called TREE-CONTAINMENT. Given a phylogenetic

Email addresses: `janosch.doecker@uni-tuebingen.de` (Janosch Döcker), `s.linz@auckland.ac.nz` (Simone Linz), `charles.semple@canterbury.ac.nz` (Charles Semple)

network $\mathcal{N}$ and a phylogenetic tree $\mathcal{T}$, this problem asks whether or not $\mathcal{N}$ embeds $\mathcal{T}$. While Tree-Containment is NP-complete in general [7], it has been shown to be polynomial-time solvable for several popular classes of phylogenetic networks, e.g. so-called tree-child and reticulation-visible networks [1, 6, 14]. Formal definitions of these classes are given in the next section. Currently, the fastest algorithm that solves Tree-Containment for these latter types of networks has a running time that is linear in the size of $\mathcal{N}$ [15]. Since the number of vertices in a tree-child and a reticulation-visible network is linear in the number of leaves [1, 2], it follows that the running time is in fact linear in the number of leaves of $\mathcal{N}$.

Pushing Tree-Containment into a novel direction, Gunawan et al. [6] have recently posed the question of how one can check if two reticulation-visible networks embed the same set of phylogenetic trees. Since the number of trees that a phylogenetic network $\mathcal{N}$ embeds grows exponentially with the number k of vertices in $\mathcal{N}$ whose in-degree is at least two, there is no immediate check that can be performed in polynomial time. In particular, the number of phylogenetic trees that $\mathcal{N}$ embeds is bounded above by 2^k, and it was shown independently in [14, Theorem 1] and [17, Corollary 3.4] that this upper bound is sharp for the class of normal networks.

Referring to the collection of phylogenetic trees that a given phylogenetic network embeds as its *display set* (formally defined in Section 2), we investigate two questions that naturally arise in comparing the display sets of two phylogenetic networks. The first question asks if the display sets of two phylogenetic networks have a common element. We call this problem Common-Tree-Containment and show in Section 3 that it is NP-complete even when the two input networks are both temporal and normal. The class of temporal and normal networks is a strict subclass of the class of tree-child and, hence, reticulation-visible networks for which Tree-Containment is polynomial-time solvable. The second problem, which we refer to as Display-Set-Equivalence, is the problem of Gunawan et al. [6] mentioned above that asks, without restricting to a particular class of phylogenetic networks, if the display sets of two networks are equal. While we recently showed that this problem has a polynomial-time algorithm for when the input consists of a normal and a tree-child network [3], we show in Section 4 that the problem is computationally hard for two arbitrary phylogenetic networks. Specifically, we show that Display-Set-Equivalence is Π_2^P-complete or, in other words, complete for the second level of the polynomial-time hierarchy [13]. In particular, unless there is a collapse in the polynomial hierarchy, such a problem has no polynomial-time reduction from itself to any NP-complete or co-NP-complete problem. From a practical viewpoint, this means that the frequently-taken approach of applying SAT and ILP solvers to find solutions to NP-complete problems is going to be of limited use when applied to a Π_2^P-complete problem. In establishing that Display-Set-Equivalence is Π_2^P-complete, we also show that deciding if the display set of one phylogenetic network is contained in the display set of another network is Π_2^P-complete.

The paper is organized as follows. The next section contains preliminaries that are used throughout the paper, formal statements of the decision problems that are mentioned in the previous paragraph, and some relevant details about the polynomial-time hierarchy. Section 3 establishes NP-completeness of Common-Tree-Containment and Section 4 establishes Π_2^P-completeness of Display-Set-Equivalence. Lastly, Section 5 contains some concluding remarks and highlights two corollaries that follow from the results in Sections 3.

2. Preliminaries

This section provides notation and terminology that is used in the remaining sections. Throughout this paper, X denotes a non-empty finite set. Let G be a directed acyclic graph. For two distinct vertices u and v in G, we say that u is an *ancestor* of v and v is a *descendant* of u, if there is a directed path from u to v in G. If (u, v) is an edge in G, then u is a *parent* of v and v is a *child* of u. Moreover, a vertex of G with in-degree one and out-degree zero is a *leaf* of G.

Phylogenetic networks and trees. A *rooted binary phylogenetic network $\mathcal{N}$ on X* is a (simple) rooted acyclic digraph that satisfies the following properties:

(i) the (unique) root has in-degree zero and out-degree two,

(ii) the set X is the set of vertices of out-degree zero, each of which has in-degree one, and

(iii) all other vertices have either in-degree one and out-degree two, or in-degree two and out-degree one.

The set X is the *leaf set* of $\mathcal{N}$. Furthermore, the vertices of in-degree one and out-degree two are *tree vertices*, while the vertices of in-degree two and out-degree one are *reticulations*. An edge directed into a reticulation is called a *reticulation edge* while each non-reticulation edge is called a *tree edge*.

Let $\mathcal{N}$ be a rooted binary phylogenetic network on X. If $\mathcal{N}$ has no reticulations, then $\mathcal{N}$ is said to be a *rooted binary phylogenetic X-tree*. To ease reading and since all phylogenetic networks considered in this paper are rooted and binary, we refer to a rooted binary phylogenetic network (resp. a rooted binary phylogenetic tree) simply as a *phylogenetic network* (resp. *a phylogenetic tree*).

Now let $\mathcal{T}$ be a phylogenetic X-tree. If $Y = \{y_1, y_2, \ldots, y_m\}$ is a subset of X, then $\mathcal{T}[-y_1, y_2, \ldots, y_m]$ and, equivalently, $\mathcal{T}|(X - Y)$ denote the phylogenetic tree with leaf set $X - Y$ that is obtained from the minimal rooted subtree of $\mathcal{T}$ that connects all leaves in $X - Y$ by suppressing all vertices of in-degree one and out-degree one.

Remark. Throughout the paper, we frequently detail constructions of phylogenetic networks. To this end, we sometimes need labels of internal vertices. Their only purpose is to make references. Indeed, they should not be regarded as genuine labels as those used for the leaves of a phylogenetic network.

Classes of phylogenetic networks. Let $\mathcal{N}$ be a phylogenetic network on X with vertex set V. An edge $e = (u, v)$ is a *shortcut* if there is a directed path from u to v whose set of edges does not contain e. A vertex v of $\mathcal{N}$ is called *visible* if there exists a leaf $\ell \in X$ such that each directed path from the root of $\mathcal{N}$ to ℓ passes through v. Now $\mathcal{N}$ is *reticulation-visible* if each reticulation in $\mathcal{N}$ is visible, and $\mathcal{N}$ is *tree-child* if each non-leaf vertex in $\mathcal{N}$ has a child that is a leaf or a tree vertex. Lastly, $\mathcal{N}$ is *normal* if it is tree-child and does not contain any shortcuts. Clearly, by definition, each normal network is also tree-child. Furthermore, it follows from the next well-known equivalence result [2] that each tree-child network is also reticulation-visible.

Lemma 2.1. *Let $\mathcal{N}$ be a phylogenetic network. Then $\mathcal{N}$ is tree-child if and only if each vertex of $\mathcal{N}$ is visible.*

Thus, the class of normal networks is a subclass of tree-child networks. Furthermore, if there exists a map $t : V \rightarrow \mathbb{R}^+$ that assigns a time stamp to each vertex of $\mathcal{N}$ and satisfies the following two properties:

(i) $t(u) = t(v)$ whenever (u, v) is a reticulation edge and

(ii) $t(u) < t(v)$ whenever (u, v) is a tree edge,

then we say that $\mathcal{N}$ is *temporal*, in which case we call t a *temporal labeling* of $\mathcal{N}$. Note that, although normal networks have no shortcuts, a normal network need not be temporal. Tree-child, normal, and temporal networks were first introduced by Cardona et al. [2], Willson [16], and Moret et al. [10], respectively.

Caterpillars. Let C be a phylogenetic tree with leaf set $\{\ell_1, \ell_2, \ldots, \ell_n\}$. Furthermore, for each $i \in \{1, 2, \ldots, n\}$ let p_i denote the parent of ℓ_i. Then C is called a *caterpillar* if $n \geq 2$ and the elements in the leaf set of C can be ordered, say $\ell_1, \ell_2, \ldots, \ell_n$, so that $p_1 = p_2$ and, for all $i \in \{3, 4, \ldots, n\}$, we have (p_i, p_{i-1}) as an edge in C. In this case, we denote C by $(\ell_1, \ell_2, \ldots, \ell_n)$. Additionally, we say that a phylogenetic X-tree $\mathcal{T}$ *contains a caterpillar* $C = (\ell_1, \ell_2, \ldots, \ell_n)$ if $\mathcal{T}$ has a subtree that is a subdivision of C.

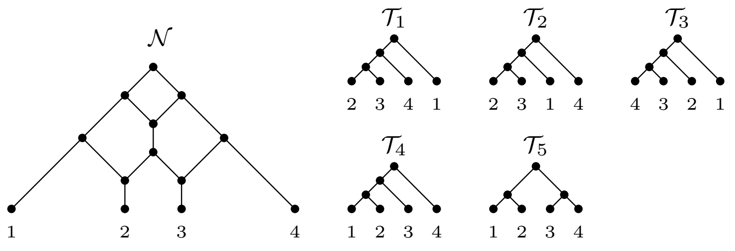


Figure 1: A phylogenetic network $\mathcal{N}$ and the display set of $\mathcal{N}$ that consists of the five trees shown on the right-hand side.

Displaying. Let $\mathcal{N}$ be a phylogenetic network on X and let $\mathcal{T}$ be a phylogenetic Y-tree such that $Y \subseteq X$. Then $\mathcal{N}$ *displays* $\mathcal{T}$ if, up to suppressing vertices of in-degree one and out-degree one, $\mathcal{T}$ can be obtained from $\mathcal{N}$ by deleting edges and vertices, in which case, the edge set, denoted by $E_{\mathcal{T}}$, of the resulting acyclic directed graph is called an *embedding* of $\mathcal{T}$ in $\mathcal{N}$. Note that, if $\mathcal{N}$ displays $\mathcal{T}$, then the root of an embedding of $\mathcal{T}$ in $\mathcal{N}$ need not coincide with the root of $\mathcal{N}$. Moreover, the *display set* of $\mathcal{N}$, denoted by $T(\mathcal{N})$, consists of all phylogenetic X-trees that are displayed by $\mathcal{N}$. As mentioned in the introduction, the size of $T(\mathcal{N})$ is bounded above by 2^k, where k is the number of reticulations in $\mathcal{N}$. To illustrate, Figure 1 shows a phylogenetic network $\mathcal{N}$ with $T(\mathcal{N}) = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_5\}$, where the five trees in $T(\mathcal{N})$ are shown on the right-hand side of the same figure. In this as well as in all other figures throughout the paper, edges are directed downwards.

Again, let $\mathcal{N}$ be a phylogenetic network on X, and let S be a subset of the edges of $\mathcal{N}$. Then S is a *switching* of $\mathcal{N}$ if, for each reticulation v of $\mathcal{N}$, S contains precisely one of the two reticulation edges that are directed into v. Now, let S be a switching of $\mathcal{N}$. If we delete each reticulation edge in $\mathcal{N}$ that is not in S and, repeatedly, suppress each resulting vertex with in-degree one and out-degree one, delete each vertex with in-degree one and out-degree zero that is not in X, and delete each vertex with in-degree zero and out-degree one, we obtain a phylogenetic X-tree $\mathcal{T}$, in which case, we say that S *yields* $\mathcal{T}$. Note that $\mathcal{T}$ is displayed by $\mathcal{N}$. Conversely, observe that, if $\mathcal{T}$ is a phylogenetic X-tree that is displayed by $\mathcal{N}$, then there exists a switching of $\mathcal{N}$ that yields $\mathcal{T}$. We summarize this in the following observation.

Observation 2.2. *A phylogenetic network $\mathcal{N}$ on X displays a phylogenetic X-tree $\mathcal{T}$ if and only if there exists a switching of $\mathcal{N}$ that yields $\mathcal{T}$.*

Problem statements. Tree-Containment is a well known problem in the study of phylogenetic networks and its computational complexity has extensively been analyzed for various network classes. In the language of this paper, it can be stated as follows.

Tree-Containment
Input. A phylogenetic X-tree $\mathcal{T}$ and phylogenetic network $\mathcal{N}$ on X.
Question. Is $\mathcal{T} \in T(\mathcal{N})$?

While Tree-Containment is concerned with a single display set, it is natural to compare display sets across phylogenetic networks, e.g. in the context of comparing networks. To make a first step in this direction, the focus of this paper are the following three decision problems that compare the display sets of two phylogenetic networks.

Common-Tree-Containment
Input. Two phylogenetic networks $\mathcal{N}$ and $\mathcal{N}'$ on X.

Question. Is $T(\mathcal{N}) \cap T(\mathcal{N}') \neq \emptyset$?

Display-Set-Containment
Input. Two phylogenetic networks $\mathcal{N}$ and $\mathcal{N}'$ on X.
Question. Is $T(\mathcal{N}) \subseteq T(\mathcal{N}')$?

Display-Set-Equivalence
Input. Two phylogenetic networks $\mathcal{N}$ and $\mathcal{N}'$ on X.
Question. Is $T(\mathcal{N}) = T(\mathcal{N}')$?

We note that Tree-Containment is a special case of both Display-Set-Containment and Common-Tree-Containment. Hence, NP-hardness of the two latter problems follows immediately for when $\mathcal{N}$ and $\mathcal{N}'$ are two arbitrary phylogenetic networks. Nevertheless, as we will see in Sections 3 and 4, we pinpoint the complexity of Common-Tree-Containment and Display-Set-Containment exactly. In particular, we will show that (i) Common-Tree-Containment is NP-complete even for when $\mathcal{N}$ and $\mathcal{N}'$ are both temporal and normal and (ii) Display-Set-Containment is complete for the second level of the polynomial-time hierarchy. This last result turns out to be a key ingredient in showing that Display-Set-Equivalence is also complete for the second level of the polynomial-time hierarchy.

The polynomial hierarchy. An *oracle* for a complexity class A is a black box that, in constant time, outputs the answer to any given instance of a decision problem contained in A. The *polynomial-time hierarchy* (or short, *polynomial hierarchy*) [5, 13] consists of a system of nested complexity classes that are defined recursively and generalize the classes P, NP, and co-NP. In particular, for any integer $k \geq 0$, we have

$$\Sigma_0^P = \Pi_0^P = \text{P},$$
$$\Sigma_{k+1}^P = \text{NP}^{\Sigma_k^P} \quad \text{and} \quad \Pi_{k+1}^P = \text{co-NP}^{\Sigma_k^P},$$

where a problem is in $\text{NP}^{\Sigma_k^P}$ (resp. $\text{co-NP}^{\Sigma_k^P}$) if we can verify an appropriate certificate of a yes-instance (resp. no-instance) in polynomial-time when given access to an oracle for Σ_k^P. By definition, $\Sigma_1^P = \text{NP}$ and $\Pi_1^P = \text{co-NP}$, and $\Pi_2^P = \text{co-NP}^{\text{NP}}$.

For all $k \geq 0$, we say that the classes Σ_k^P and Π_k^P are on the *k-th level* of the polynomial hierarchy. Although, Σ_{k+1}^P (resp. Π_{k+1}^P) generalizes Σ_k^P (resp. Π_k^P), it is an open problem whether $\Sigma_k^P = \Sigma_{k+1}^P$ or $\Pi_k^P = \Pi_{k+1}^P$ for any $k \geq 0$. Specifically, for $k = 0$, this is the fundamental P versus NP problem. If $\Sigma_k^P = \Sigma_{k+1}^P$ or $\Pi_k^P = \Pi_{k+1}^P$ for some $k \geq 0$, then this would result in a collapse of the polynomial hierarchy to the k-th level.

In Section 4, we show that Display-Set-Containment and Display-Set-Equivalence are both Π_2^P-complete. Intuitively, problems that are complete for the second level of the polynomial hierarchy are more difficult than problems that are complete for the first level. Recall that a decision problem is in co-NP if a no-instance can be verified in polynomial time given an appropriate certificate. Now, similar to showing that a problem is co-NP-complete, a proof that establishes Π_2^P-completeness consists of two steps: (i) show that a problem is in Π_2^P, and (ii) establish a polynomial-time reduction from a problem that is known to be Π_2^P-complete to the problem at hand. With regards to (i), a decision problem is *in* Π_2^P if a no-instance can be verified in polynomial time when one is given an appropriate certificate and has access to an NP-*oracle*, that is, an oracle that can solve NP-complete problems in constant time.

3. Hardness of Common-Tree-Containment

As noted in the introduction, Tree-Containment is NP-complete in general, but polynomial-time solvable for several popular classes of phylogenetic networks such as tree-child and reticulation-visible networks. In this section, we

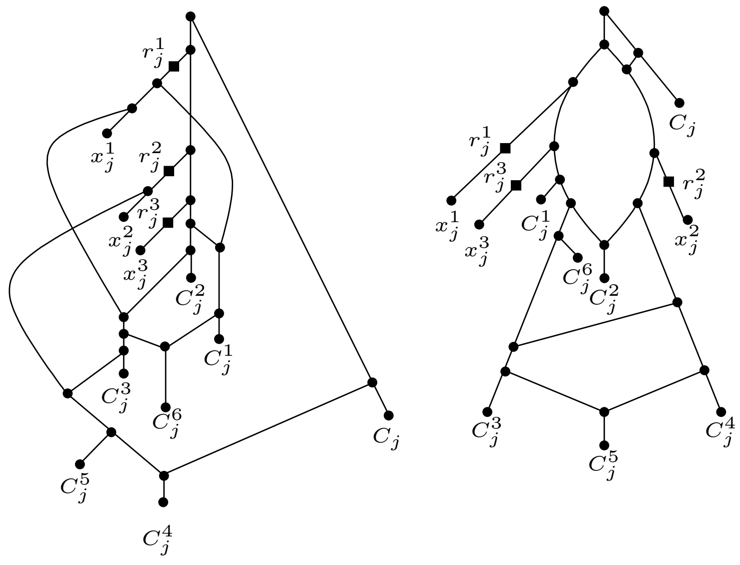


Figure 2: For a clause $C_j = (x_j^1 \vee x_j^2 \vee x_j^3)$, the clause gadget G_j^A (left) and the clause gadget G_j^B (right). Leaves are bijectively labeled with the elements in $\{C_j, C_j^1, C_j^2, \ldots, C_j^6, x_j^1, x_j^2, x_j^3\}$. Furthermore, each gadget has three vertices of in-degree one and out-degree one indicated by small squares labeled r_j^1, r_j^2, and r_j^3.

show that no such dichotomy holds for Common-Tree-Containment. In particular, we will show that this problem is NP-complete even if the input consists of two temporal normal networks. To establish the result, we use a reduction from the classical computational problem 3-SAT.

3-SAT
Input. A set $V = \{v_1, v_2, \ldots, v_n\}$ of variables, and a set $\{C_1, C_2, \ldots, C_m\}$ of clauses such that each clause is a disjunction of exactly three literals and each literal is an element in $\{v_i, \bar{v}_i : i \in \{1, 2, \ldots, n\}\}$.
Question. Does there exist a truth assignment for V that satisfies each clause C_j with $j \in \{1, 2, \ldots, m\}$?

Let I be an instance of 3-SAT, and let $C_j = (x_j^1 \vee x_j^2 \vee x_j^3)$ be a clause of I for $j \in \{1, 2, \ldots, m\}$. Then, for some indices k, k', and k'' in $\{1, 2, \ldots, n\}$, we have $x_j^1 \in \{v_k, \bar{v}_k\}$, $x_j^2 \in \{v_{k'}, \bar{v}_{k'}\}$, and $x_j^3 \in \{v_{k''}, \bar{v}_{k''}\}$. Without loss of generality, we impose the following two restrictions on I:

(R1) for each $v_i \in V$ with $i \in \{1, 2, \ldots, n\}$, at most one element in $\{v_i, \bar{v}_i\}$ is a literal of C_j and

(R2) $k < k' < k''$.

Now, for each clause C_j, we construct the two clause gadgets G_j^A and G_j^B that are shown in Figure 2. We next establish a simple lemma.

Lemma 3.1. *Let G_j^A and G_j^B be the two clause gadgets that are shown in Figure 2. Obtain two phylogenetic networks $\mathcal{G}_j^A$ and $\mathcal{G}_j^B$ from G_j^A and G_j^B, respectively, by suppressing the three vertices r_j^1, r_j^2, and r_j^3 of in-degree one and out-degree one. Then $T(\mathcal{G}_j^A) \cap T(\mathcal{G}_j^B) = \emptyset$.*

Proof. To see that $T(\mathcal{G}_j^A) \cap T(\mathcal{G}_j^B) = \emptyset$, observe that each tree in $T(\mathcal{G}_j^A)$ contains the caterpillar (x_j^2, x_j^3, x_j^1), whereas each tree in $T(\mathcal{G}_j^B)$ contains the caterpillar (x_j^1, x_j^3, x_j^2). □

Following on from Lemma 3.1, let $L = \{x_j^1, x_j^2, x_j^3\}$. Although $\mathcal{G}_j^A$ and $\mathcal{G}_j^B$ display no common phylogenetic tree with leaf set $\{C_j, C_j^1, C_j^2, \ldots, C_j^6\} \cup L$, they do display a common phylogenetic with leaf set $\{C_j, C_j^1, C_j^2, \ldots, C_j^6\} \cup L'$ for each proper subset L' of L. In the proof of Theorem 3.2, for some truth assignment β, the latter corresponds to at least one literal in C_j being satisfied by β, while the former corresponds to no literal in C_j being satisfied by β.

Let $S = (s_1, s_2, \ldots, s_n)$ be an arbitrary tuple, and let r be an element that is not contained in S. We write $(r)\|S$ to denote the tuple $(r, s_1, s_2, \ldots, s_n)$ obtained by concatenating r and S. With this definition in hand, we are now in a position to establish the main result of this section.

Theorem 3.2. Common-Tree-Containment *is* NP-*complete when the input consists of two temporal normal networks.*

Proof. For two normal networks, van Iersel et al. [14] showed that the running time of Tree-Containment is polynomial in the size of this leaf set. Hence, it follows that Common-Tree-Containment is in NP for two normal networks.

Let I be an instance of 3-SAT with n variables and m clauses. Using the same notation as in the formal statement of 3-SAT, we construct two phylogenetic networks $\mathcal{N}$ and $\mathcal{N}'$ on

$$X = \{C_j, C_j^1, C_j^2, \ldots, C_j^6, x_j^1, x_j^2, x_j^3 : j \in \{1, 2, \ldots, m\}\} \bigcup \{v_i : i \in \{1, 2, \ldots, n\}\}$$

as follows. Let $\mathcal{T}$ be the phylogenetic tree obtained by creating a vertex ρ, adding an edge that joins ρ with the root of the caterpillar $(v_1, v_2, \ldots, v_n)$, and adding an edge that joins ρ with the root of the caterpillar $(c_1, c_2, \ldots, c_m)$. Now, setting $\mathcal{M} = \mathcal{M}' = \mathcal{T}$, let $\mathcal{N}$ and $\mathcal{N}'$ be the two phylogenetic networks obtained from $\mathcal{M}$ and $\mathcal{M}'$, respectively, by applying the following four-step process.

1. For all $j \in \{1, 2, \ldots, m\}$, replace c_j with G_j^A in $\mathcal{M}$ and replace c_j with G_j^B in $\mathcal{M}'$.
2. For all $i \in \{1, 2, \ldots, n\}$, subdivide the edge directed into v_i with a new vertex d_i in $\mathcal{M}$ and $\mathcal{M}'$.
3. For each $j \in \{1, 2, \ldots, m\}$ in increasing order, consider $C_j = (x_j^1 \vee x_j^2 \vee x_j^3)$. Let v_{k_ℓ} be the unique element in V such that $x_j^\ell \in \{v_{k_\ell}, \bar{v}_{k_\ell}\}$ for each $\ell \in \{1, 2, 3\}$. If $x_j^\ell = v_{k_\ell}$, subdivide the edge directed into v_{k_ℓ} with a new vertex u_j^ℓ in $\mathcal{M}$ and subdivide the edge directed into d_{k_ℓ} with a new vertex u_j^ℓ in $\mathcal{M}'$. Otherwise, subdivide the edge directed into d_{k_ℓ} with a new vertex u_j^ℓ in $\mathcal{M}$ and subdivide the edge directed into v_{k_ℓ} with a new vertex u_j^ℓ in $\mathcal{M}'$. Add a new edge (u_j^ℓ, r_j^ℓ) in $\mathcal{M}$ and $\mathcal{M}'$.
4. For each $i \in \{1, 2, \ldots, n\}$, suppress the vertex d_i of in-degree one and out-degree one in $\mathcal{M}$ and $\mathcal{M}'$.

To illustrate, Figure 3 gives a high-level overview of the construction of $\mathcal{N}$ and $\mathcal{N}'$. Observe that, for each $j \in \{1, 2, \ldots, m\}$, the three vertices r_j^1, r_j^2, and r_j^3 in $\mathcal{N}$ and $\mathcal{N}'$ are reticulations.

We next show that $\mathcal{N}$ and $\mathcal{N}'$ are both temporal and normal.

3.2.1. *Both $\mathcal{N}$ and $\mathcal{N}'$ are temporal and normal.*

Proof. We first show that $\mathcal{N}$ is temporal and normal. Let

$$V_r = \{r_j^\ell : j \in \{1, 2, \ldots, m\} \text{ and } \ell \in \{1, 2, 3\}\}.$$

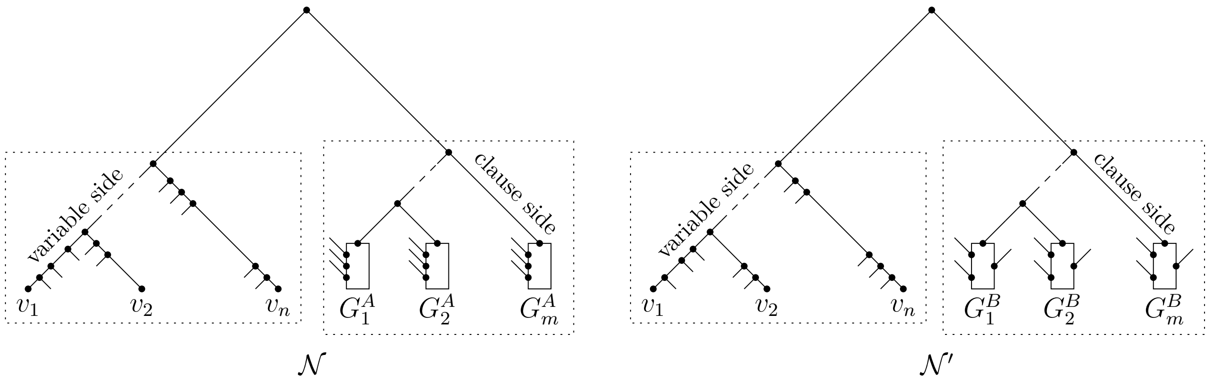


Figure 3: Overview of the construction of the two temporal normal networks $\mathcal{N}$ and $\mathcal{N}'$ in the proof of Theorem 3.2. Dangling edges on the clause and variable side of $\mathcal{N}$ and $\mathcal{N}'$, respectively, are paired up depending on I. For details, see Step (3) of the construction.

Furthermore, for each $i \in \{1, 2, \ldots, n\}$, let V_i consist of all vertices that lie on the unique directed path from the root of $\mathcal{N}$ to v_i, and let

$$V_v = \bigcup_{i=1}^{n} V_i.$$

We begin by assigning a positive real-valued labeling t to each vertex in $V_v \cup V_r$ as follows. First, under t, each vertex in V_v is assigned a labeling such that the following two properties are satisfied.

(i) If $u, v \in V_v$ and u is an ancestor of v, then $t(u) < t(v)$.

(ii) For all $i \in \{1, 2, \ldots, n-1\}$, the temporal labeling of each vertex in V_i that is not contained in V_{i+1} is smaller than the minimum temporal labeling over all vertices that are contained in V_{i+1} and not in V_i .

By construction of $\mathcal{N}$, note that such a labeling always exists. Second, under t, each vertex in V_r is assigned the same labeling as its unique parent that is contained in V_v. Because of restrictions (R1) and (R2) that we have imposed on I and the way we have assigned temporal labelings to the vertices in V_v, we have

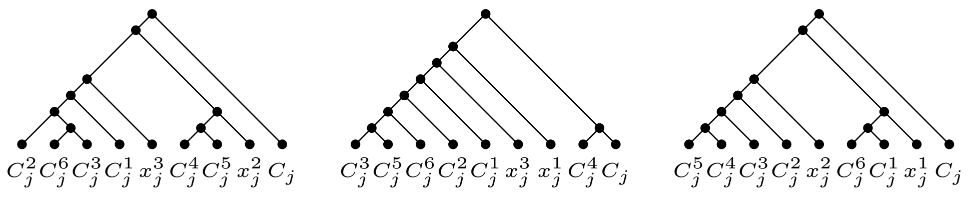
$$t(r_j^1) < t(r_j^2) < t(r_j^3)$$

for each $j \in \{1, 2, \ldots, m\}$. A routine check now shows that t can be extended to a temporal labeling of $\mathcal{N}$ and, thus, $\mathcal{N}$ is temporal.

Now, since $\mathcal{N}$ is temporal, it follows that $\mathcal{N}$ has no shortcuts. Hence, to show that $\mathcal{N}$ is normal, it suffices to show that $\mathcal{N}$ is tree-child. It is straightforward to check that $\mathcal{N}$ has no edge (u, v) such that u and v are both reticulations. Hence, each reticulation in $\mathcal{N}$ has a child that is a tree vertex or a leaf. Furthermore, by construction, each tree vertex of $\mathcal{N}$ that is a vertex of some G_j^A with $j \in \{1, 2, \ldots, m\}$ has a child that is a tree vertex or a leaf. Lastly, for each non-leaf vertex v of $\mathcal{N}$ that is neither a reticulation nor a vertex of some G_j^A, consider a directed path P from v to an element in $\{v_1, v_2, \ldots, v_n, C_1, C_2, \ldots, C_m\}$. By construction, P exists. It is now easily seen that the second vertex of P is a child of v that is either a tree vertex or a leaf. This establishes that $\mathcal{N}$ is normal. An analogous argument that uses G_j^B instead of G_j^A can be used to show that $\mathcal{N}'$ is temporal and normal, thereby completing the proof of (3.2.1). □

Since the number of vertices of a normal network is polynomial in the size of X [9] and $|X| = 10m + n$, it follows that $\mathcal{N}$ and $\mathcal{N}'$ can be constructed in time polynomial in the size of X.

3.2.2. *The instance I is a yes-instance if and only if $T(\mathcal{N}) \cap T(\mathcal{N}') \neq \emptyset$.*


$C_j^2\, C_j^6\, C_j^3\, C_j^1\, x_j^3\, C_j^4\, C_j^5\, x_j^2\, C_j$
$C_j^3\, C_j^5\, C_j^6\, C_j^2\, C_j^1\, x_j^3\, x_j^1\, C_j^4\, C_j$
$C_j^5\, C_j^4\, C_j^3\, C_j^2\, x_j^2\, C_j^6\, C_j^1\, x_j^1\, C_j$

(1) $\mathcal{T}_j^{TFF}$ (2) $\mathcal{T}_j^{FTF}$ (3) $\mathcal{T}_j^{FFT}$

(4) $\mathcal{T}_j^{FTT} = \mathcal{T}_j^{FTF}[-x_j^3]$ (5) $\mathcal{T}_j^{TFT} = \mathcal{T}_j^{FFT}[-x_j^1]$

(6) $\mathcal{T}_j^{TTF} = \mathcal{T}_j^{FTF}[-x_j^1]$ (7) $\mathcal{T}_j^{TTT} = \mathcal{T}_j^{FTF}[-x_j^1, x_j^3]$

Figure 4: The seven trees that are used in the proof of Theorem 3.2.

Proof. First, suppose that I is a yes-instance. We construct a *variable tree* $\mathcal{T}_v$ and a *clause tree* $\mathcal{T}_c$ that, joined together, result in a phylogenetic X-tree that is displayed by $\mathcal{N}$ and $\mathcal{N}'$. Let $\beta : V \rightarrow \{F, T\}$ be a truth assignment that satisfies each clause, and let

$$Y = \{x_j^\ell : j \in \{1, 2, \ldots, m\} \text{ and } \ell \in \{1, 2, 3\}\}.$$

Furthermore, for each $i \in \{1, 2, \ldots, n\}$, let Y_i (resp. $\bar{Y}_i$) be the tuple consisting of the elements in Y that equal v_i (resp. $\bar{v}_i$) such that, for any two elements x_j^ℓ and $x_{j'}^{\ell'}$ in Y_i (resp. $\bar{Y}_i$), x_j^ℓ precedes $x_{j'}^{\ell'}$ precisely if $j > j'$. By construction, note that the two caterpillars $(v_i)||Y_i$ and $(v_i)||\bar{Y}_i$ are displayed by $\mathcal{N}$ and $\mathcal{N}'$. Now, obtain $\mathcal{T}_v$ from the caterpillar $(v_1, v_2, \ldots, v_n)$ by doing the following for each $i \in \{1, 2, \ldots, n\}$. If $\beta(v_i) = T$, replace v_i with the caterpillar $(v_i)||Y_i$; otherwise, replace v_i with the caterpillar $(v_i)||\bar{Y}_i$. Again, by construction, it is easily checked that $\mathcal{T}_v$ is displayed by $\mathcal{N}$ and $\mathcal{N}'$. We next construct $\mathcal{T}_c$. Consider a clause $C_j = (x_j^1 \vee x_j^2 \vee x_j^3)$. For each $\ell \in \{1, 2, 3\}$, set $z_\ell = T$ if x_j^ℓ is satisfied by β and, otherwise, set $z_\ell = F$. Depending on which elements in $\{z_1, z_2, z_3\}$ equal F and T, respectively, and noting that there exists some ℓ for which $z_\ell = T$, we define the *clause tree* $\mathcal{T}_j^{z_1z_2z_3}$ relative to C_j to be one of the seven trees that are listed in Figure 4. Intuitively, x_j^ℓ is a leaf in $\mathcal{T}_j^{z_1z_2z_3}$ precisely if $z_\ell = F$. Now, obtain $\mathcal{T}_c$ from the caterpillar $(c_1, c_2, \ldots, c_m)$ by replacing, for each $j \in \{1, 2, \ldots, m\}$, the leaf c_j with the clause tree relative to C_j. As $\mathcal{T}_j^{z_1z_2z_3}$ is displayed by the two phylogenetic networks obtained from G_j^A and G_j^B by suppressing the three vertices r_j^1, r_j^2, and r_j^3 of in-degree one and out-degree one, it follows that $\mathcal{T}_j^{z_1z_2z_3}$ is also displayed by $\mathcal{N}$ and $\mathcal{N}'$. In turn, this implies that, by construction, $\mathcal{T}_c$ is displayed by $\mathcal{N}$ and $\mathcal{N}'$. Lastly, we construct a phylogenetic tree $\mathcal{T}$ on X by creating a vertex ρ, adding a new edge that joins ρ with the root of $\mathcal{T}_v$, and a new edge that joins ρ with the root of $\mathcal{T}_c$. As $\mathcal{T}_v$ and $\mathcal{T}_c$ are displayed by $\mathcal{N}$ and $\mathcal{N}'$, it is easily checked that $\mathcal{T}$ is displayed by $\mathcal{N}$ and $\mathcal{N}'$, and so $T(\mathcal{N}) \cap T(\mathcal{N}') \neq \emptyset$.

Second, suppose that $T(\mathcal{N}) \cap T(\mathcal{N}') \neq \emptyset$. Let $\mathcal{T}$ be a phylogenetic X-tree that is displayed by $\mathcal{N}$ and $\mathcal{N}'$. Furthermore, let $j, j' \in \{1, 2, \ldots, m\}$, and let $\ell, \ell' \in \{1, 2, 3\}$. For each reticulation r_j^ℓ in $\mathcal{N}$ (resp. $\mathcal{N}'$), we say that $\mathcal{T}$ *picks* x_j^ℓ *from the clause side* of $\mathcal{N}$ (resp. $\mathcal{N}'$) if $\mathcal{T}$ has a vertex whose set of descendants contains x_j^ℓ and C_j but does not contain any element in V; otherwise, we say that $\mathcal{T}$ *picks* x_j^ℓ *from the variable side* of $\mathcal{N}$ (resp. $\mathcal{N}'$). Intuitively, x_j^ℓ is picked from the clause side of $\mathcal{N}$ (resp. $\mathcal{N}'$) precisely if the embedding of $\mathcal{T}$ in $\mathcal{N}$ (resp. $\mathcal{N}'$) contains the reticulation edge directed into r_j^ℓ whose two end vertices are vertices of G_j^A (resp. G_j^B). Note that, as $\mathcal{T}$ is displayed by $\mathcal{N}$ and $\mathcal{N}'$, we have that $\mathcal{T}$ picks x_j^ℓ from the variable side of $\mathcal{N}$ if and only if $\mathcal{T}$ picks x_j^ℓ from the variable side of $\mathcal{N}'$. We next make two observations:

(O1) For each clause $C_j = (x_j^1 \vee x_j^2 \vee x_j^3)$, it follows from Lemma 3.1 that $\mathcal{T}$ picks at most two of x_j^1, x_j^2, and x_j^3 from the clause side of $\mathcal{N}$ and $\mathcal{N}'$.

(O2) It follows from Step (3) in the construction of $\mathcal{N}$ and $\mathcal{N}'$, and the fact that $\mathcal{T}$ is displayed by $\mathcal{N}$ and $\mathcal{N}'$ that, if $\mathcal{T}$ picks x_j^ℓ from the variable side of $\mathcal{N}$ and $\mathcal{N}'$, and $x_j^\ell = v_i$ for some $i \in \{1, 2, \ldots, n\}$, then each $x_{j'}^{\ell'}$ with $x_{j'}^{\ell'} = \bar{v}_i$

is picked from the clause side of $\mathcal{N}$ and $\mathcal{N}'$. Similarly, if $\mathcal{T}$ picks x_j^ℓ from the variable side of $\mathcal{N}$ and $\mathcal{N}'$, and $x_j^\ell = \bar{v}_i$ for some $i \in \{1, 2, \ldots, n\}$, then each $x_{j'}^{\ell'}$ with $x_{j'}^{\ell'} = v_i$ is picked from the clause side of $\mathcal{N}$ and $\mathcal{N}'$.

Now, let β be the truth assignment that is defined as follows. For each $i \in \{1, 2, \ldots, n\}$, we set $v_i = T$ if there exists an element x_j^ℓ with $x_j^\ell = v_i$ that is picked from the variable side of $\mathcal{N}$ and $\mathcal{N}'$. On the other hand, we set $v_i = F$ if either there exists an element x_j^ℓ with $x_j^\ell = \bar{v}_i$ that is picked from the variable side of $\mathcal{N}$ and $\mathcal{N}'$ or there is no x_j^ℓ with $x_j^\ell \in \{v_i, \bar{v}_i\}$ that is picked from the variable side of $\mathcal{N}$ and $\mathcal{N}'$. Because of (O2), β is well defined. Moreover, by (O1) it follows that β satisfies at least one literal of each clause and, hence, I is a yes-instance. □

This completes the proof of Theorem 3.2. □

The next corollary is an immediate consequence of Theorem 3.2.

Corollary 3.3. *Let $\mathcal{N}$ and $\mathcal{N}'$ be two temporal normal networks on X. It is* co-NP-*complete to decide if $T(\mathcal{N}) \cap T(\mathcal{N}') = \emptyset$.*

4. Hardness of Display-Set-Equivalence

In this section, we show that Display-Set-Equivalence is Π_2^P-complete, that is, the problem is complete for the second level of the polynomial hierarchy. To establish this result, we use a chain of three polynomial-time reductions that are described in Subsections 4.1, 4.2, and 4.3. Before detailing the reductions, we introduce two more decision problems that play an important role in this section.

Recall the (ordinary) 3-SAT problem as introduced in Section 3. The input to an instance of 3-SAT consists of a boolean formula over a set of variables. Importantly, each variable is existentially quantified since we are asking whether or not there *exists* a truth assignment to each variable that satisfies each clause of the formula. In contrast, the following quantified version of 3-SAT has two different types of variables, i.e. each variable is either existentially or universally quantified.

∀∃ 3-SAT
Input. A quantified boolean formula

$$\Psi = \forall v_1 \forall v_2 \cdots \forall v_p \exists v_{p+1} \exists v_{p+2} \cdots \exists v_n \bigwedge_{j=1}^{m} C_j$$

over a set of variables $V = \{v_1, v_2, \ldots, v_n\}$ such that each clause C_j is a disjunction of exactly three literals and each literal is an element in $\{v_i, \bar{v}_i : i \in \{1, 2, \ldots, n\}\}$.
Question. For each truth assignment $\beta^\forall : \{v_1, v_2, \ldots, v_p\} \to \{F, T\}$, does there exist a truth assignment $\beta^\exists : \{v_{p+1}, v_{p+2}, \ldots, v_p\} \to \{F, T\}$ such that, collectively, $\beta^\forall$ and $\beta^\exists$ satisfy each clause in Ψ?

It was shown in [13] that ∀∃ 3-SAT is Π_2^P-complete. Let I be an instance of ∀∃ 3-SAT. Note that each clause of I has at least one literal that is an element in $\{x_i, \bar{x}_i : i \in \{p + 1, p + 2, \ldots, n\}\}$ since, otherwise, I is a no-instance. Furthermore, if all variables are existentially quantified, then I is an instance of the (ordinary) 3-SAT problem. Hence, we may assume throughout this section that $1 \leq p < n$.

We next formally state a quantified version of the well-known NP-complete decision problem Directed-Disjoint-Connecting-Paths [5, 11]. Let G be a directed graph with vertex set V, and let $\{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$ be a collection of disjoint pairs of vertices in V. In what follows, we write π_i to denote a directed path in G from s_i to t_i with $i \in \{1, 2, \ldots, k\}$.

$\forall\exists$ Directed-Disjoint-Connecting-Paths
Input. A directed graph G and two collections

$$P^\forall = \{(s_1, t_1), (s_2, t_2), \ldots, (s_p, t_p)\},$$
$$P^\exists = \{(s_{p+1}, t_{p+1}), (s_{p+2}, t_{p+2}), \ldots, (s_k, t_k)\}$$

of disjoint pairs of vertices in G such that $1 \leq p < k$ and, for each $(s_i, t_i) \in P^\forall$, there exists a directed path from s_i to t_i in G.
Question. For each set $\Pi^\forall = \{\pi_1, \pi_2, \ldots, \pi_p\}$ of directed paths, does there exist a set $\Pi^\forall \cup \{\pi_{p+1}, \pi_{p+2}, \ldots, \pi_k\}$ of mutually vertex-disjoint directed paths in G?

4.1. $\forall\exists$ Directed-Disjoint-Connecting-Paths *is* Π_2^P*-complete*

To show that $\forall\exists$ Directed-Disjoint-Connecting-Paths is complete for the second level of the polynomial hierarchy, we use a polynomial-time reduction from $\forall\exists$ 3-SAT. This reduction constructs a special instance of $\forall\exists$ Directed-Disjoint-Connecting-Paths for which the input graph is a particular type of phylogenetic network.

Let $\mathcal{N}$ be a phylogenetic network on X, let $S = \{s_1, s_2, \ldots, s_k\}$ and $T = \{t_1, t_2, \ldots, t_k\}$ be two disjoint subsets of the vertices of $\mathcal{N}$ such that $T = X$, and let $p \in \{1, 2, \ldots, k\}$. We call $\mathcal{N}$ a *caterpillar-inducing* network with respect to S if the network obtained from $\mathcal{N}$ by deleting each vertex that lies on a directed path from a child of a vertex in S to a leaf of $\mathcal{N}$ is a caterpillar up to deleting all leaf labels. Moreover, we say that $\mathcal{N}$ has the *two-path property relative to p* if, for each $i \in \{1, 2, \ldots, p\}$, there are two directed paths, say π_i and π'_i, from s_i to t_i such that the following three properties are satisfied:

(i) π_i and π'_i are the only directed paths from s_i to t_i in $\mathcal{N}$,

(ii) π_i and π'_i only have the three vertices s_i, t_i, and the (unique) parent of t_i as well as the edge directed into t_i in common, and

(iii) no path in $\{\pi_i, \pi'_i : i \in \{1, 2, \ldots, p\}\}$ intersects with any path in $\{\pi_j, \pi'_j : j \in \{1, 2, \ldots, p\} - \{i\}\}$.

Using the same notation as in the statement of $\forall\exists$ Directed-Disjoint-Connecting-Paths, we now introduce a similar problem whose input graph is a phylogenetic network.

$\forall\exists$ Phylo-Directed-Disjoint-Connecting-Paths
Input. A phylogenetic network $\mathcal{N}$ on X, two disjoint sets $S = \{s_1, s_2, \ldots, s_k\}$ and $T = X = \{t_1, t_2, \ldots, t_k\}$ of vertices of $\mathcal{N}$, and an integer p with $1 \leq p < k$ such that $\mathcal{N}$ is caterpillar-inducing with respect to S and has the two-path property relative to p. Furthermore, the two collections

$$P^\forall = \{(s_1, t_1), (s_2, t_2), \ldots, (s_p, t_p)\},$$
$$P^\exists = \{(s_{p+1}, t_{p+1}), (s_{p+2}, t_{p+2}), \ldots, (s_k, t_k)\}$$

of pairs of elements in S and T.
Question. For each set $\Pi^\forall = \{\pi_1, \pi_2, \ldots, \pi_p\}$ of directed paths, does there exist a set $\Pi^\forall \cup \{\pi_{p+1}, \pi_{p+2}, \ldots, \pi_k\}$ of mutually vertex-disjoint directed paths in $\mathcal{N}$?

The next theorem establishes the Π_2^P-completeness of $\forall\exists$ Phylo-Directed-Disjoint-Connecting-Paths. The reduction that we use for the proof has a flavor that is similar to that in [8, page 86].

Theorem 4.1. *The decision problem $\forall\exists$ Phylo-Directed-Disjoint-Connecting-Paths is Π_2^P-complete.*

Proof. We first show that $\forall\exists$ Phylo-Directed-Disjoint-Connecting-Paths is in Π_2^P. Using the same notation as in the formal statement of this problem, let $\Pi^\forall = \{\pi_1, \pi_2, \ldots, \pi_p\}$ be a set of directed paths in $\mathcal{N}$. Since $\mathcal{N}$ has the two-path property relative to p, the paths in $\Pi^\forall$ are mutually vertex disjoint. Next obtain the directed graph G from $\mathcal{N}$ by deleting all vertices that lie on a path in $\Pi^\forall$. Lastly, use an NP-oracle for the unquantified version of Directed-Disjoint-Connecting-Paths to decide if there exists a set $\Pi^\exists = \{\pi_{p+1}, \pi_{p+2}, \ldots, \pi_k\}$ of mutually vertex-disjoint directed paths in G. Since a given instance of $\forall\exists$ Phylo-Directed-Disjoint-Connecting-Paths is a no-instance precisely if there exists some set $\Pi^\forall$ for which no choice of $\Pi^\exists$ results in a set $\Pi^\forall \cup \Pi^\exists$ of mutually vertex-disjoint directed paths in $\mathcal{N}$, it follows that this problem is in co-NP$^{\text{NP}} = \Pi_2^P$.

We now establish a polynomial-time reduction from the quantified 3-SAT problem. Let I be an instance of $\forall\exists$ 3-SAT with boolean formula

$$\Psi = \forall v_1 \forall v_2 \cdots \forall v_p \exists v_{p+1} \exists v_{p+2} \cdots \exists v_n \bigwedge_{j=1}^{m} C_j$$

over a set $V = \{v_1, v_2, \ldots, v_n\}$ of variables. Throughout the proof, we use $C_j = (x_{3j-2} \vee x_{3j-1} \vee x_{3j})$ to refer to the three literals in C_j for each $j \in \{1, 2, \ldots, m\}$. Now, for each $i \in \{1, 2, \ldots, n\}$, let $\mathcal{J}_i^+$ be the set that consists of the indices of the literals that are equal to v_i and, similarly, let $\mathcal{J}_i^-$ be the set that consists of the indices of the literals that are equal to $\bar{v}_i$. Without loss of generality, we may assume that $\mathcal{J}_i^+ \neq \emptyset$ or $\mathcal{J}_i^- \neq \emptyset$ since, otherwise, v_i can be deleted from V.

For each variable v_i, we construct a variable gadget G_i^v as follows:

1. Create three vertices s_i^v, t_i^v, and y_i.
2. Create the (possibly empty) set of vertices $\bigcup_{l \in \mathcal{J}_i^+} \{p_l^{\text{in}}, p_l^{\text{out}}\}$ and construct the directed path
$$\pi_i^+ = (s_i^v,\, p_{l_1}^{\text{in}},\, p_{l_1}^{\text{out}},\, p_{l_2}^{\text{in}},\, p_{l_2}^{\text{out}},\, \ldots,\, p_{l_q}^{\text{in}},\, p_{l_q}^{\text{out}},\, y_i, t_i^v)$$
with $\{l_1, l_2, \ldots, l_q\} = \mathcal{J}_i^+$.
3. Create the (possibly empty) set of vertices $\bigcup_{k \in \mathcal{J}_i^-} \{n_k^{\text{in}}, n_k^{\text{out}}\}$ and construct the directed path
$$\pi_i^- = (s_i^v,\, n_{k_1}^{\text{in}},\, n_{k_1}^{\text{out}},\, n_{k_2}^{\text{in}},\, n_{k_2}^{\text{out}}, \ldots,\, n_{k_r}^{\text{in}},\, n_{k_r}^{\text{out}},\, y_i, t_i^v)$$
with $\{k_1, k_2, \ldots, k_r\} = \mathcal{J}_i^-$.

Note that, since we do not allow for parallel edges, the last edge (y_i, t_i^v) of π_i^+ and π_i^- only appears once in G_i^v. Intuitively, the two paths π_i^+ and π_i^- correspond to the two possible truth assignments for the variable v_i. To illustrate, a generic variable gadget for v_i is shown on the left-hand side of Figure 5. The additional edges in this figure that are directed into vertices of the variable gadget and directed out of vertices of this gadget will be defined as part of the clause gadget construction which we describe next.

For a clause $C_j = (x_{3j-2} \vee x_{3j-1} \vee x_{3j})$, let i_j, i'_j, and i''_j be the elements in $\{1, 2, \ldots, n\}$ such that $x_{3j-2} \in \{v_{i_j}, \bar{v}_{i_j}\}$, $x_{3j-1} \in \{v_{i'_j}, \bar{v}_{i'_j}\}$, and $x_{3j} \in \{v_{i''_j}, \bar{v}_{i''_j}\}$. Now, for each $j \in \{1, 2, \ldots, m\}$, add the following vertices and edges to the variable gadgets.

1. Create the vertices $\{s_j^c, t_j^c, u_j, w_j, w'_j\}$.

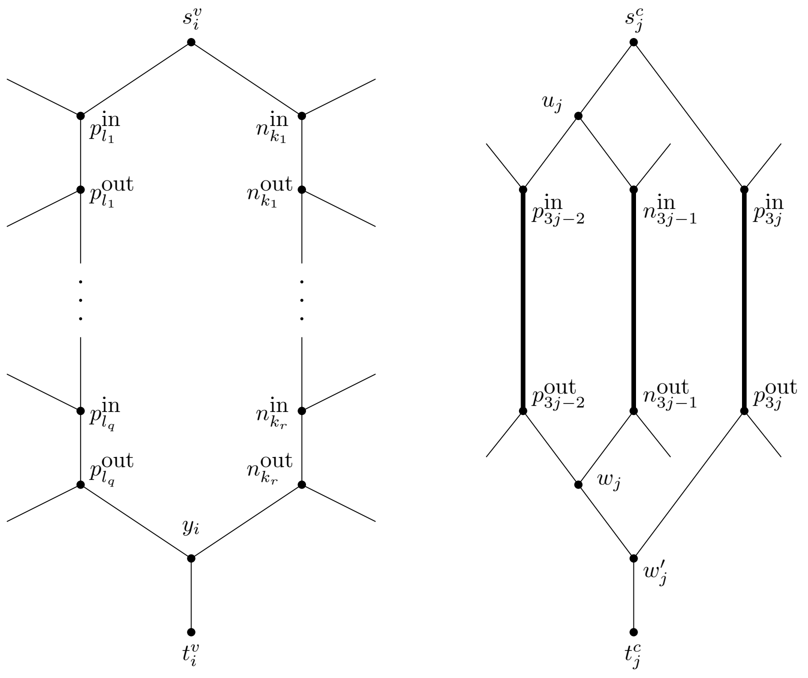


Figure 5: Left: Variable gadget for a variable v_i. Right: Clause gadget for a clause $C_j = (x_{3j-2} \vee x_{3j-1} \vee x_{3j})$, where the second literal equals a negated variable and each of the other two literals equals an unnegated variable. The three thick edges are edges of a variable gadget. The complete construction is detailed in the proof of Theorem 4.1.

2. Add the edges in $\{(s_j^c, u_j), (w_j, w_j'), (w_j', t_j^c)\}$.

3. If $x_{3j-2} = v_{i_j}$, add the edges $(u_j, p_{3j-2}^{\text{in}})$ and $(p_{3j-2}^{\text{out}}, w_j)$. Otherwise, add the edges $(u_j, n_{3j-2}^{\text{in}})$ and $(n_{3j-2}^{\text{out}}, w_j)$.

4. If $x_{3j-1} = v_{i_j'}$, add the edges $(u_j, p_{3j-1}^{\text{in}})$ and $(p_{3j-1}^{\text{out}}, w_j)$. Otherwise, add the edges $(u_j, n_{3j-1}^{\text{in}})$ and $(n_{3j-1}^{\text{out}}, w_j)$.

5. If $x_{3j} = v_{i_j''}$, add the edges $(s_j^c, p_{3j}^{\text{in}})$ and $(p_{3j}^{\text{out}}, w_j')$. Otherwise, add the edges $(s_j^c, n_{3j}^{\text{in}})$ and $(n_{3j}^{\text{out}}, w_j')$.

In what follows, we refer to the edges and vertices that get added in the aforementioned five-step construction relative to a given C_j as the clause gadget for C_j. For each clause $C_j = (x_{3j-2} \vee x_{3j-1} \vee x_{3j})$, there are three directed paths from s_j^c to t_j^c each of which corresponds to one of the three literals in C_j. For example, for the first literal x_{3j-2}, there is a directed path from s_j^c to t_j^c that intersects with the edge $(p_{3j-2}^{\text{in}}, p_{3j-2}^{\text{out}})$ on $\pi_{i_j}^+$ if $x_{3j-2} = v_{i_j}$ and that intersects with the edge $(n_{3j-2}^{\text{in}}, n_{3j-2}^{\text{out}})$ on $\pi_{i_j}^-$ if $x_{3j-2} = \bar{v}_{i_j}$. To illustrate, assume that $x_{3j-2} = v_{i_j}$, $x_{3j-1} = \bar{v}_{i_j'}$, and $x_{3j} = v_{i_j''}$. For this specific case, the clause gadget for C_j is shown on the right-hand side of Figure 5.

Now, let G be the directed graph that results from the construction of all variable and all clause gadgets. Observe that G is acyclic. We next set up an instance I' of ∀∃ Phylo-Directed-Disjoint-Connecting-Paths. Let $\mathcal{T}$ be the caterpillar $(\ell_1^v, \ell_2^v, \ldots, \ell_n^v, \ell_1^c, \ell_2^c, \ldots, \ell_m^c)$. We obtain a directed acyclic graph $\mathcal{N}$ from $\mathcal{T}$ and G by identifying ℓ_i^v with s_i^v for each $i \in \{1, 2, \ldots, n\}$ and identifying ℓ_j^c with s_j^c for each $j \in \{1, 2, \ldots, m\}$. Clearly, $\mathcal{N}$ is connected and has no parallel edges. Moreover, except for the root, since each vertex of G has in-degree one and out-degree two, in-degree two and out-degree one, or in-degree one and out-degree zero, it follows that $\mathcal{N}$ is a phylogenetic network on $T = \{t_1^v, t_2^v, \ldots, t_n^v, t_1^c, t_2^c, \ldots, t_m^c\}$. Let $S = \{s_1^v, s_2^v, \ldots, s_n^v, s_1^c, s_2^c, \ldots, s_m^c\}$. Since every vertex of G that is not contained in S lies on a directed path from a child of a vertex in S to a leaf in $\mathcal{N}$, it follows that $\mathcal{N}$ is caterpillar-inducing with respect to S. Moreover, for each $i \in \{1, 2, \ldots, n\}$, there are exactly two directed paths from s_i^v to t_i^v in G_i^v and, hence, in $\mathcal{N}$ that only intersect in the vertices s_i^v, t_i^v, and y_i, and the edge (y_i, t_i^v). Recalling that $1 \leq p < n$, it follows from the construction that $\mathcal{N}$ has the two-path property relative to p, and that both $P^{\forall}$ and $P^{\exists}$ are non-empty. We now set

$$P^{\forall} = \{(s_1^v, t_1^v), (s_2^v, t_2^v), \ldots, (s_p^v, t_p^v)\} \text{ and}$$
$$P^{\exists} = \{(s_{p+1}^v, t_{p+1}^v), (s_{p+2}^v, t_{p+2}^v), \ldots, (s_n^v, t_n^v)\} \cup \{(s_1^c, t_1^c), (s_2^c, t_2^c), \ldots, (s_m^c, t_m^c)\}.$$

This completes the description of I'.

Since the number of vertices of G is $3n + 11m$, the number of vertices of $\mathcal{T}$ is $2(n + m) - 1$, and G and $\mathcal{T}$ have $n + m$ vertices in common, it follows that $\mathcal{N}$ has size $O(n + m)$ and can be constructed in polynomial time.

We complete the proof by establishing the following sublemma.

4.1.1. *The instance I is a yes-instance if and only if the instance I' is a yes-instance.*

Proof. First, suppose that I is a yes-instance. Let $\Pi^{\forall} = \{\pi_1^v, \pi_2^v, \ldots, \pi_p^v\}$ be a set of directed paths in $\mathcal{N}$ such that each π_i^v begins at s_i^v and ends at t_i^v. As $p < n$, we have $\pi_i^v \in \{\pi_i^+, \pi_i^-\}$. Moreover, since $\mathcal{N}$ has the two-path property relative to p, the paths in $\Pi^{\forall}$ are mutually vertex disjoint in $\mathcal{N}$. Now, let $\beta : V \to \{F, T\}$ be a truth assignment that satisfies each clause of Ψ such that, if $\pi_i^v = \pi_i^+$, then $v_i = F$ and, otherwise, $v_i = T$ for each $i \in \{1, 2, \ldots, p\}$. Since I is a yes-instance, β exists. We next construct a directed path for each pair of vertices in $P^{\exists}$ such that, collectively, these paths together with the elements in $\Pi^{\forall}$ form a solution to I'. For each $i \in \{p + 1, p + 2, \ldots, n\}$, set $\pi_i^v = \pi_i^+$ if $v_i = F$ and set $\pi_i^v = \pi_i^-$ if $v_i = T$. Furthermore, for each $j \in \{1, 2, \ldots, m\}$, let $x_{j'}$, with $j' \in \{3j - 2, 3j - 1, 3j\}$, be a literal in C_j that is satisfied by β, and let i be the element in $\{1, 2, \ldots, n\}$ such that $x_{j'} \in \{v_i, \bar{v}_i\}$. By construction of the clause gadget, there is a directed path, say π_j^c, from s_j^c to t_j^c in $\mathcal{N}$ such that one of the following properties applies.

(i) If $x_{j'} = v_i$, then π_j^c contains the edge $(p_{j'}^{\text{in}}, p_{j'}^{\text{out}})$.

(ii) If $x_{j'} = \bar{v}_i$, then π_j^c contains the edge $(n_{j'}^{\text{in}}, n_{j'}^{\text{out}})$.

In Case (i), as $v_i = T$, we have $\pi_i^v = \pi_i^-$, and it follows that π_j^c does not intersect π_i^v. Similar in Case (ii), as $v_i = F$, we have $\pi_i^v = \pi_i^+$, and it again follows that π_j^c does not intersect π_i^v. By construction of $\mathcal{N}$, it is now straightforward to check that

$$\Pi^{\forall} \cup \{\pi_{p+1}^v, \pi_{p+2}^v, \ldots, \pi_n^v, \pi_1^c, \pi_2^c, \ldots, \pi_m^c\}$$

is a collection of mutually vertex-disjoint directed-paths in $\mathcal{N}$ that connect each pair of vertices in $P^{\forall} \cup P^{\exists}$. In particular, since the argument presented in this paragraph applies to all choices of directed paths in $\Pi^{\forall}$, we conclude that I' is a yes-instance.

Second, suppose that I' is a yes-instance. Let $\beta^{\forall} : \{v_1, v_2, \ldots, v_p\} \to \{F, T\}$ be a truth assignment. Furthermore, let

$$\Pi = \{\pi_1^v, \pi_2^v, \ldots, \pi_p^v\} \cup \{\pi_{p+1}^v, \pi_{p+2}^v, \ldots, \pi_n^v, \pi_1^c, \pi_2^c, \ldots, \pi_m^c\}$$

be a collection of mutually vertex-disjoint directed paths in $\mathcal{N}$ such that $\pi_i^v = \pi_i^-$ if $v_i = T$ and $\pi_i^v = \pi_i^+$ if $v_i = F$ for each $i \in \{1, 2, \ldots, p\}$. Since I' is a yes-instance, Π exists. Now, let $\beta : V \to \{F, T\}$ such that

(i) for each $i \in \{1, 2, \ldots, p\}$, we have $\beta(v_i) = \beta^{\forall}(v_i)$ and,

(ii) for each $i \in \{p + 1, p + 2, \ldots, n\}$, we have $\beta(v_i) = F$ if $\pi_i^v = \pi_i^+$ and, $\beta(v_i) = T$ if $\pi_i^v = \pi_i^-$.

We next show that β satisfies each clause of Ψ. Let $C_j = (x_{3j-2} \vee x_{3j-1} \vee x_{3j})$ be a clause of Ψ with $j \in \{1, 2, \ldots, m\}$. Consider the directed path $\pi_j^c \in \Pi$ from s_j^c to t_j^c in $\mathcal{N}$. Let j' be the unique element in $\{3j - 2, 3j - 1, 3j\}$ such that π_j^c

contains either the edge $(p^{\text{in}}_{j'}, p^{\text{out}}_{j'})$ or the edge $(n^{\text{in}}_{j'}, n^{\text{out}}_{j'})$, and let i be the element in $\{1, 2, \ldots, n\}$ such that $x_{j'} \in \{v_i, \bar{v}_i\}$. First, assume that π^c_j contains $(p^{\text{in}}_{j'}, p^{\text{out}}_{j'})$. Then, as $x_{j'} = v_i$ and the paths in Π are mutually vertex disjoint in $\mathcal{N}$, it follows that $\pi^v_i = \pi^-_i$. Hence $\beta(v_i) = T$. Second, assume that π^c_j contains $(n^{\text{in}}_{j'}, n^{\text{out}}_{j'})$. Then, as $x_{j'} = \bar{v}_i$ and the paths in Π are mutually vertex disjoint, it follows that $\pi^v_i = \pi^+_i$. Hence $\beta(v_i) = F$. Under both assumptions, β satisfies C_j because $\beta(x_{j'}) = T$. It now follows that β satisfies Ψ and, as the argument applies to all choices of truth assignments for the elements in $\{v_1, v_2, \ldots, v_p\}$, we conclude that I is a yes-instance. □

This completes the proof of Theorem 4.1. □

While the next corollary is not needed for the remainder of the paper, it may be of independent interest in the theoretical computer science community.

Corollary 4.2. *The decision problem* $\forall\exists$ Directed-Disjoint-Connecting-Paths *is* Π^P_2*-complete.*

Proof. Since every instance of $\forall\exists$ Phylo-Directed-Disjoint-Connecting-Paths is also an instance of $\forall\exists$ Directed-Disjoint-Connecting-Paths, it follows from Theorem 4.1 that the latter problem is Π^P_2-hard. To establish that $\forall\exists$ Directed-Disjoint-Connecting-Paths is in Π^P_2, we use the same argument as in the first paragraph of the proof of Theorem 4.1 and, additionally, check in polynomial time if the paths in $\Pi^\forall$ are vertex disjoint. □

4.2. Display-Set-Containment *is* Π^P_2*-complete*

In this section, we show that Display-Set-Containment is complete for the second level of the polynomial hierarchy. This problem is a generalization of the well-known NP-complete Tree-Containment problem [7].

Theorem 4.3. Display-Set-Containment *is* Π^P_2*-complete.*

Proof. We first show that Display-Set-Containment is in Π^P_2. Let $\mathcal{N}$ and $\mathcal{N}'$ be two phylogenetic networks on X. To decide if $T(\mathcal{N}) \subseteq T(\mathcal{N}')$, let S be a switching of $\mathcal{N}$, and let $\mathcal{T}$ be the phylogenetic X-tree yielded by S. Then use an NP-oracle for Tree-Containment to decide if $\mathcal{T}$ is displayed by $\mathcal{N}'$. Since $\mathcal{N}$ and $\mathcal{N}'$ form a no-instance precisely if there exists some switching for $\mathcal{N}$ that yields a phylogenetic tree that is not displayed by $\mathcal{N}'$, it follows that Display-Set-Containment is in co-NP$^{\text{NP}}$ = Π^P_2.

To complete the proof, we establish a reduction from $\forall\exists$ Phylo-Directed-Disjoint-Connecting-Paths. Using the same notation as in the formal statement of $\forall\exists$ Phylo-Directed-Disjoint-Connecting-Paths, let I be the following instance of this problem. Let $\mathcal{N}$ be a phylogenetic network on X, let $S = \{s_1, s_2, \ldots, s_k\}$ and $T = X = \{t_1, t_2, \ldots, t_k\}$ be two disjoint sets of vertices of $\mathcal{N}$, and let p be an integer with $1 \le p < k$ such that $\mathcal{N}$ is caterpillar-inducing with respect to S and has the two-path property relative to p. Furthermore, let

$$P^\forall = \{(s_1, t_1), (s_2, t_2), \ldots, (s_p, t_p)\},$$
$$P^\exists = \{(s_{p+1}, t_{p+1}), (s_{p+2}, t_{p+2}), \ldots, (s_k, t_k)\}$$

be two collections of pairs of elements in S and T. This completes the description of I.

Now, let $\mathcal{N}_1$ be the phylogenetic network obtained from the caterpillar $(t_0, s_1, s_2, \ldots, s_p, t_{p+1}, t_{p+2}, \ldots, t_k)$ by adding the following edges and vertices for each $i \in \{1, 2, \ldots, p\}$. Create three vertices u^1_i, u^2_i, and u^3_i and add the set

$$\{(s_i, u^1_i), (s_i, u^2_i), (u^1_i, u^3_i), (u^2_i, u^3_i), (u^3_i, t_i), (u^1_i, t'_i), (u^2_i, t''_i)\}$$

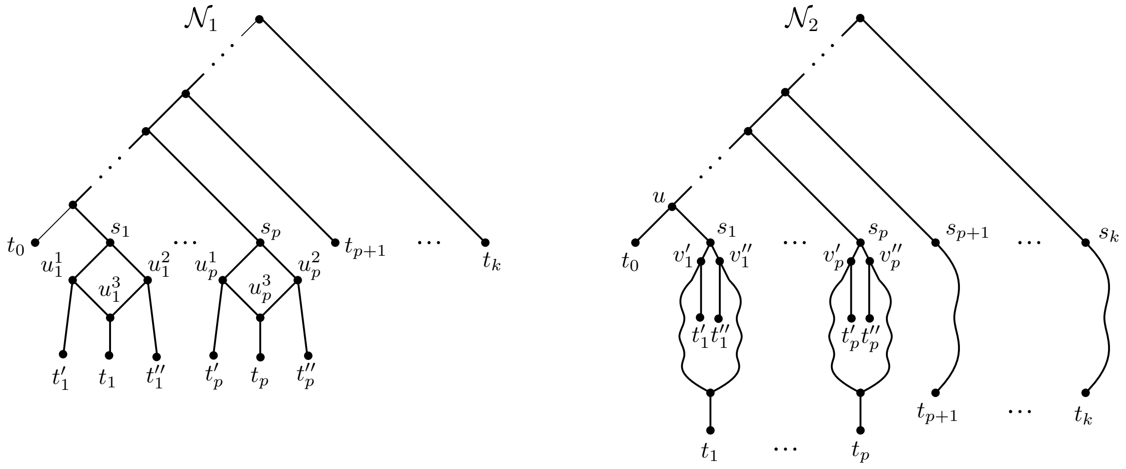


Figure 6: The two phylogenetic networks $\mathcal{N}_1$ and $\mathcal{N}_2$ that are constructed in the proof of Theorem 4.3. For reasons of simplicity, not all edges of $\mathcal{N}_2$ are shown. In particular, each squiggly line is a directed path and, depending on the given instance of $\forall\exists$ Phylo-Directed-Disjoint-Connecting-Paths, squiggly paths may intersect with each other and may be further interconnected by paths that are not shown.

of edges. Observe that the leaf set of $\mathcal{N}_1$ is

$$X' = \{t_0, t_1, t_2, \ldots, t_k\} \cup \{t_i', t_i'' : i \in \{1, 2, \ldots, p\}\}.$$

The construction of $\mathcal{N}_1$ is shown on the left-hand side of Figure 6. We complete the reduction to an instance of Display-Set-Containment by describing a second phylogenetic network $\mathcal{N}_2$. For each $i \in \{1, 2, \ldots, p\}$, let w_i' and w_i'' be the two children of s_i in $\mathcal{N}$. As $\mathcal{N}$ has the two-path property relative to p, recall that there are exactly two directed paths from s_i to t_i in $\mathcal{N}$, and these two paths only have s_i, t_i, and the parent of t_i in common. In the remainder of the proof, we denote the directed path from s_i to t_i that contains w_i' with π_i' and, similarly, we denote the directed path from s_i to t_i that contains w_i'' with π_i''. Lastly, we denote the parent of s_1 with p_1. Now, obtain $\mathcal{N}_2$ from $\mathcal{N}$ in the following way.

(i) Subdivide the edge (p_1, s_1) with a new vertex u and add the edge (u, t_0).

(ii) For each $i \in \{1, 2, \ldots, p\}$, subdivide (s_i, w_i') with a new vertex v_i', subdivide (s_i, w_i'') with a new vertex v_i'', and add the two edges (v_i', t_i') and (v_i'', t_i'').

Clearly, the leaf set of $\mathcal{N}_2$ is X'. To illustrate, $\mathcal{N}_2$ is shown on the right-hand side in Figure 6.

As the size of X' is polynomial in the size of X, it follows that the size of $\mathcal{N}_1$ and $\mathcal{N}_2$ is polynomial in the size of $\mathcal{N}$. Furthermore, the construction of $\mathcal{N}_1$ and $\mathcal{N}_2$ takes polynomial time.

4.3.1. *The instance I is a yes-instance if and only if $T(\mathcal{N}_1) \subseteq T(\mathcal{N}_2)$.*

Proof. First, suppose that I is a yes-instance. Let $\mathcal{T}'$ be a phylogenetic X'-tree that is displayed by $\mathcal{N}_1$. For each $i \in \{1, 2, \ldots, p\}$, note that $\mathcal{T}'$ contains one of the two caterpillars (t_i, t_i', t_i'') or (t_i, t_i'', t_i'). Let $\mathcal{J}'$ be the set that consists of each element $i \in \{1, 2, \ldots, p\}$ for which $\mathcal{T}'$ contains (t_i, t_i', t_i'') and, similarly, let $\mathcal{J}''$ be the set that consists of each element $i \in \{1, 2, \ldots, p\}$ for which $\mathcal{T}'$ contains (t_i, t_i'', t_i'). Furthermore, let $\Pi^\forall = \{\pi_1, \pi_2, \ldots, \pi_p\}$ be the set of directed paths in $\mathcal{N}$ such that $\pi_i = \pi_i'$ if $i \in \mathcal{J}'$ and $\pi_i = \pi_i''$ if $i \in \mathcal{J}''$. Since I is a yes-instance, there exists a set $\Pi = \Pi^\forall \cup \{\pi_{p+1}, \pi_{p+2}, \ldots, \pi_k\}$ of mutually vertex-disjoint directed paths in $\mathcal{N}$, where π_j is a directed path from s_j to t_j for each $j \in \{p+1, p+2, \ldots, k\}$. Moreover, as $\mathcal{N}$ is caterpillar-inducing with respect to S, it is straightforward to check that there exists a phylogenetic X-tree $\mathcal{T}$ such that the following three properties are satisfied:

(i) $\mathcal{T}$ is displayed by $\mathcal{N}$,

(ii) $\mathcal{T} = \mathcal{T}'|X$, and

(iii) there exists an embedding of $\mathcal{T}$ in $\mathcal{N}$ that contains all edges of paths in Π.

Let $E_{\mathcal{T}}$ be an embedding of $\mathcal{T}$ in $\mathcal{N}$ that satisfies (iii). By construction of $\mathcal{N}_2$ from $\mathcal{N}$, there exists an embedding of $\mathcal{T}$ in $\mathcal{N}_2$ whose set of edges is

$$\begin{aligned} E'_{\mathcal{T}} &= (E_{\mathcal{T}} - (\{(p_1, s_1)\} \cup \{(s_i, w'_i) : i \in \mathcal{J}'\} \cup \{(s_i, w''_i) : i \in \mathcal{J}''\})) \cup \\ &\quad \{(p_1, u), (u, s_1)\} \cup \{(s_i, v'_i), (v'_i, w'_i) : i \in \mathcal{J}'\} \cup \\ &\quad \{(s_i, v''_i), (v''_i, w''_i) : i \in \mathcal{J}''\}. \end{aligned}$$

For each $i \in \{1, 2, \ldots, p\}$, let E'_i be the subset $\{(v'_i, t'_i), (v''_i, t''_i), (s_i, v''_i)\}$ of edges in $\mathcal{N}_2$ if $i \in \mathcal{J}'$, and the subset $\{(v''_i, t''_i), (v'_i, t'_i), (s_i, v'_i)\}$ of edges in $\mathcal{N}_2$ if $i \in \mathcal{J}''$. Since $E'_{\mathcal{T}}$ is an embedding of $\mathcal{T}$ in $\mathcal{N}_2$, it now follows that

$$E'_{\mathcal{T}} \cup E'_1 \cup E'_2 \cup \cdots \cup E'_p \cup \{(u, t_0)\}$$

is an embedding of $\mathcal{T}'$ in $\mathcal{N}_2$. Hence, $T(\mathcal{N}_1) \subseteq T(\mathcal{N}_2)$.

Second, suppose that I is a no-instance. Throughout this part of the proof, we use π_i to denote a directed path from s_i to t_i in $\mathcal{N}$ for each $i \in \{1, 2, \ldots, k\}$. Then, as $\mathcal{N}$ has the two-path property relative to p, there is a set $\Pi^{\forall} = \{\pi_1, \pi_2, \ldots, \pi_p\}$ of mutually vertex-disjoint directed paths in $\mathcal{N}$ for which every set $\Pi = \Pi^{\forall} \cup \{\pi_{p+1}, \pi_{p+2}, \ldots, \pi_k\}$ of directed paths in $\mathcal{N}$ contains two elements that are not vertex disjoint. For each $i \in \{1, 2, \ldots, k\}$, let E_i be the set of edges of π_i in $\mathcal{N}$. Furthermore, for each $i \in \{1, 2, \ldots, p\}$, let E'_i be the subset

$$(E_i - \{(s_i, w'_i)\}) \cup \{(s_i, v'_i), (v'_i, w'_i), (v'_i, t'_i), (s_i, v''_i), (v''_i, t''_i)\}$$

of edges in $\mathcal{N}_2$ if $\pi_i = \pi'_i$, and the subset

$$(E_i - \{(s_i, w''_i)\}) \cup \{(s_i, v''_i), (v''_i, w''_i), (v''_i, t''_i), (s_i, v'_i), (v'_i, t'_i)\}$$

of edges in $\mathcal{N}_2$ if $\pi_i = \pi''_i$, where π'_i or π''_i are as described in the construction of $\mathcal{N}_2$ from $\mathcal{N}$. Clearly, there is a phylogenetic tree $\mathcal{T}_p$ with leaf set $\{t_i, t'_i, t''_i : i \in \{1, 2, \ldots, p\}\}$ for which there exists an embedding in $\mathcal{N}_2$ that contains all edges in $E'_1 \cup E'_2 \cup \cdots \cup E'_p$. Observe that $\mathcal{T}_p$ can be obtained from the caterpillar $(\ell_1, \ell_2, \ldots, \ell_p)$ by replacing each $\ell_i \in \{\ell_1, \ell_2, \ldots, \ell_p\}$ with the caterpillar (t_i, t'_i, t''_i) if $\pi_i = \pi'_i$ and with the caterpillar (t_i, t''_i, t'_i) if $\pi_i = \pi''_i$. By construction, it now follows that $\mathcal{N}_1$ displays $\mathcal{T}_p$. Let $\mathcal{T}$ be the unique phylogenetic X'-tree that is displayed by $\mathcal{N}_1$ such that $\mathcal{T}|\{t_i, t'_i, t''_i : i \in \{1, 2, \ldots, p\}\} = \mathcal{T}_p$. We complete the argument by showing that $\mathcal{T}$ is not displayed by $\mathcal{N}_2$. Towards a contradiction, assume that $\mathcal{T}$ is displayed by $\mathcal{N}_2$. Let $E'_{\mathcal{T}}$ be an embedding of $\mathcal{T}$ in $\mathcal{N}_2$. Then, since $\mathcal{T}$ contains (t_i, t'_i, t''_i) or (t_i, t''_i, t'_i) for each $i \in \{1, 2, \ldots, p\}$ and $\mathcal{N}$ satisfies the two-path property relative to p, it follows from the construction of $\mathcal{N}_2$ that $E'_{\mathcal{T}}$ contains all edges in $E'_1 \cup E'_2 \cup \cdots \cup E'_p$. Furthermore, observe that there is a unique directed path from the root, say ρ, of $\mathcal{N}_2$ to t_0, and so the edges on this path are elements of $E'_{\mathcal{T}}$. For each pair i and i' of distinct elements in $\{1, 2, \ldots, k\}$, it therefore follows that the directed path from ρ to t_i in $E'_{\mathcal{T}}$ and the directed path from ρ to $t_{i'}$ in $E'_{\mathcal{T}}$ only intersect in vertices that are ancestors of t_0 in $\mathcal{N}_2$. Hence, as $\mathcal{N}_2$ is caterpillar-inducing with respect to S, there exist directed paths $\pi^*_1, \pi^*_2, \ldots, \pi^*_p, \pi^*_{p+1}, \ldots, \pi^*_k$ in $E'_{\mathcal{T}}$ such that the following three properties are fulfilled.

(i) For each $i \in \{1, 2, \ldots, p\}$, π^*_i is the unique directed path from s_i to t_i in $\mathcal{N}_2$ that contains v'_i if $\pi_i = \pi'_i$ and that contains v''_i if $\pi_i = \pi''_i$.

(ii) For each $i \in \{p + 1, p + 2, \ldots, k\}$, π^*_i is a directed path from s_i to t_i in $\mathcal{N}_2$.

(iii) The elements in $\Pi^* = \{\pi^*_1, \pi^*_2, \ldots, \pi^*_k, \}$ are mutually vertex disjoint.

Now, by construction, observe that π_i^* is also a directed path from s_i to t_i in $\mathcal{N}$ for each $i \in \{p+1, p+2, \ldots, k\}$. As Π^* is a set of mutually vertex-disjoint directed paths in $\mathcal{N}_2$, it now follows that, $\Pi^\forall \cup \{\pi_{p+1}^*, \pi_{p+2}^*, \ldots, \pi_k^*\}$ is a set of mutually vertex-disjoint directed paths in $\mathcal{N}$. In turn, this implies that I is a yes-instance; a contradiction. Hence, $\mathcal{T} \notin T(\mathcal{N}_2)$, and so $T(\mathcal{N}_1) \nsubseteq T(\mathcal{N}_2)$. □

This establishes Theorem 4.3. □

We end this section with a brief discussion of the structural properties of the phylogenetic network $\mathcal{N}_1$ that is constructed in the proof of Theorem 4.3. These properties will play an important role in the next section when we establish Π_2^P-completeness of Display-Set-Equivalence. Let $\mathcal{N}$ be a phylogenetic network on X. We say that $\mathcal{N}$ is a *caterpillar network* if it can be obtained from a caterpillar $(\ell_1, \ell_2, \ldots, \ell_k)$ with $2 \leq k \leq |X|$ by replacing each ℓ_i with a phylogenetic network $\mathcal{N}_i$ on X_i such that the elements in $\{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_k\}$ are pairwise vertex disjoint and

$$\bigcup_{i=1}^{k} X_i = X.$$

By construction, $\mathcal{N}_1$ is a caterpillar network. Moreover, it is easily seen that $\mathcal{N}_1$ is temporal and tree-child.

The next corollary now immediately follows from Theorem 4.3.

Corollary 4.4. *Let $\mathcal{N}_1$ be a temporal tree-child caterpillar network on X, and let $\mathcal{N}_2$ be a phylogenetic network on X. Then deciding whether $T(\mathcal{N}_1) \subseteq T(\mathcal{N}_2)$ is Π_2^P-complete.*

4.3. Display-Set-Equivalence *is Π_2^P-complete*

With the result of Corollary 4.4 in hand, we are now in a position to establish the main result of Section 4 which is the following theorem.

Theorem 4.5. Display-Set-Equivalence *is Π_2^P-complete.*

Proof. Let $\mathcal{N}$ and $\mathcal{N}'$ be two phylogenetic networks on X. By Theorem 4.3, the problem of deciding whether or not $T(\mathcal{N}) \subseteq T(\mathcal{N}')$ is in Π_2^P. Similarly, the problem of deciding whether or not $T(\mathcal{N}') \subseteq T(\mathcal{N})$ is in Π_2^P. Hence, Display-Set-Equivalence is in Π_2^P.

We next establish a polynomial-time reduction from Display-Set-Containment to Display-Set-Equivalence. Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be two phylogenetic networks on $X = \{\ell_1, \ell_2, \ldots, \ell_n\}$ that form the input to an instance of Display-Set-Containment that asks if $T(\mathcal{N}_1) \subseteq T(\mathcal{N}_2)$. By Corollary 4.4, we may assume that $\mathcal{N}_1$ is a caterpillar network. Then there exist two vertex-disjoint phylogenetic networks $\mathcal{M}_1$ and $\mathcal{M}_{1'}$ with leaf sets W_1 and $W_{1'}$, respectively, such that $W_1 \cup W_{1'} = X$, and $\mathcal{N}_1$ can be obtained from the caterpillar $\{x_1, x_2\}$ by replacing x_1 with $\mathcal{M}_1$ and x_2 with $\mathcal{M}_{1'}$. To ease reading, let $\mathcal{N}_1'$ and $\mathcal{N}_2'$ be the two phylogenetic networks on $X' = \{\ell_1', \ell_2', \ldots, \ell_n'\}$ that are obtained from $\mathcal{N}_1$ and $\mathcal{N}_2$, respectively, by replacing ℓ_i with ℓ_i' in both networks for each $i \in \{1, 2, \ldots, n\}$. Similarly, let $\mathcal{M}_1'$ and $\mathcal{M}_{1'}'$ be the two phylogenetic networks obtained from $\mathcal{M}_1$ and $\mathcal{M}_{1'}$, respectively, by replacing ℓ_i with ℓ_i' in exactly one of $\mathcal{M}_1$ and $\mathcal{M}_{1'}$ for each $i \in \{1, 2, \ldots, n\}$. If W_1' (resp. $W_{1'}'$) denotes the leaf set of $\mathcal{M}_1'$ (resp. $\mathcal{M}_{1'}'$), then $W_1' \cup W_{1'}' = X'$.

Set $\mathcal{T}$ as well as $\mathcal{T}'$ to be the caterpillar $(w_1, w_2, \ldots, w_{2n+3})$. Furthermore, let $u_{2n+3}, u_{2n+2}, \ldots, u_2$ be the directed path in $\mathcal{T}$ (and $\mathcal{T}'$) such that, for all $j \in \{2, 3, \ldots, 2n+3\}$, u_j is the parent of w_j. Now, let G_1^* and G_2^* be the two directed acyclic graphs that are obtained from $\mathcal{T}$ and $\mathcal{T}'$, respectively, by applying the following six-step process.

1. For all $j \in \{1, 2, \ldots, n\}$, replace w_j with $\mathcal{N}_2$ in $\mathcal{T}$ and $\mathcal{T}'$ by identifying w_j with the root of $\mathcal{N}_2$.

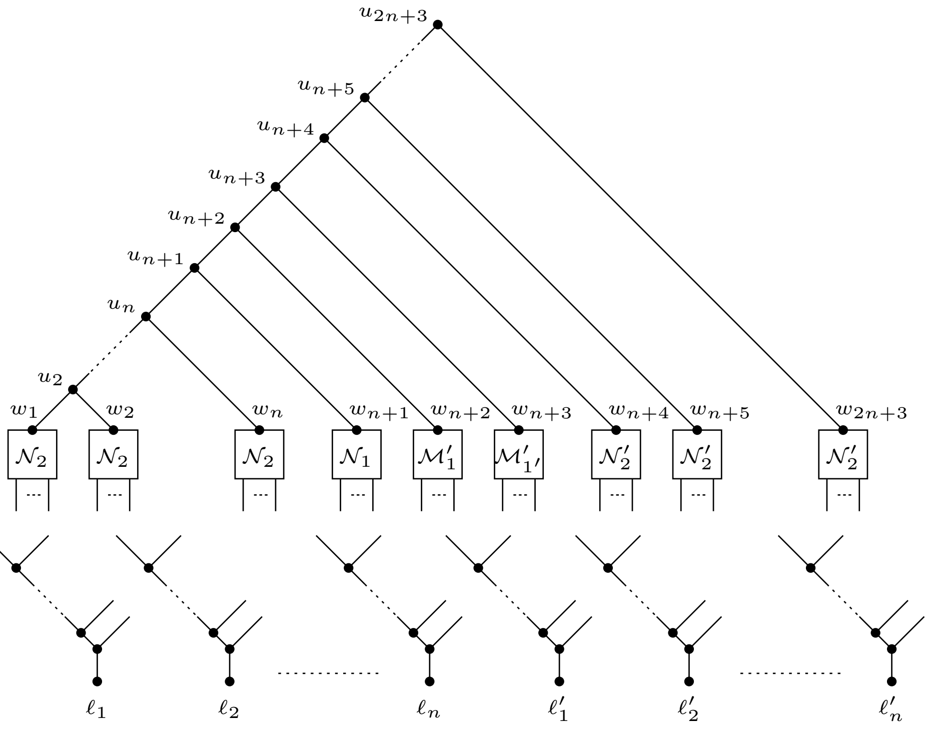


Figure 7: The phylogenetic network $\mathcal{N}_1^*$ on $X \cup X'$ as constructed in the proof of Theorem 4.5. Each of the squares labeled $\mathcal{N}_2$, $\mathcal{N}_1$, $\mathcal{M}'_1$, $\mathcal{M}'_{1'}$, and $\mathcal{N}'_2$ refers to the network obtained from its namesake phylogenetic network by deleting all leaf labels. The dangling edges of each square are paired up with the edges shown in the bottom part of the figure as described in Step (6) of the construction in the proof of Theorem 4.5.

2. Replace w_{n+1} with the root of $\mathcal{N}_1$ in $\mathcal{T}$ by identifying w_{n+1} with the root of $\mathcal{N}_1$, and replace w_{n+1} with the root of $\mathcal{M}_1$ in $\mathcal{T}'$ by identifying w_{n+1} with the root of $\mathcal{M}_1$

3. Replace w_{n+2} with $\mathcal{M}'_1$ in $\mathcal{T}$ by identifying w_{n+2} with the root of $\mathcal{M}'_1$, and replace w_{n+2} with $\mathcal{M}_{1'}$ in $\mathcal{T}'$ by identifying w_{n+2} with the root of $\mathcal{M}_{1'}$

4. Replace w_{n+3} with $\mathcal{M}'_{1'}$ in $\mathcal{T}$ by identifying w_{n+3} with the root of $\mathcal{M}'_{1'}$, and replace w_{n+3} with $\mathcal{N}'_1$ in $\mathcal{T}'$ by identifying w_{n+3} with the root of $\mathcal{N}'_1$.

5. For all $j \in \{n+4, n+5, \ldots, 2n+3\}$, replace w_j with $\mathcal{N}'_2$ in $\mathcal{T}$ and $\mathcal{T}'$ by identifying w_j with the root of $\mathcal{N}'_2$.

6. For each $i \in \{1, 2, \ldots, n\}$, identify all leaves labeled ℓ_i (resp. ℓ'_i) in $\mathcal{T}$ with a new vertex v_i (resp. v'_i), add a new edge (v_i, ℓ_i) (resp. (v'_i, ℓ'_i)). Do the same for all leaves labeled ℓ_i (resp. ℓ'_i) in $\mathcal{T}'$.

To complete the construction, let $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$ be two phylogenetic networks such that G_1^* and G_2^* can be obtained from $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$, respectively, by contracting edges. Clearly, the leaf set of $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$ is $X \cup X'$. Moreover, the directed path $u_{2n+3}, u_{2n+2}, \ldots, u_2$ of $\mathcal{T}$ and $\mathcal{T}'$ is also a directed path of $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$. We refer to this path as the *backbone* of $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$. The phylogenetic networks $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$ are shown in Figures 7 and 8, respectively. Lastly, observe that the size of both $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$ is $O(n(|E_1| + |E_2|))$, where E_1 and E_2 is the edge set of $\mathcal{N}_1$ and $\mathcal{N}_2$, respectively. Hence, the construction of $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$ takes polynomial time.

4.5.1. $T(\mathcal{N}_1) \subseteq T(\mathcal{N}_2)$ *if and only if* $T(\mathcal{N}_1^*) = T(\mathcal{N}_2^*)$.

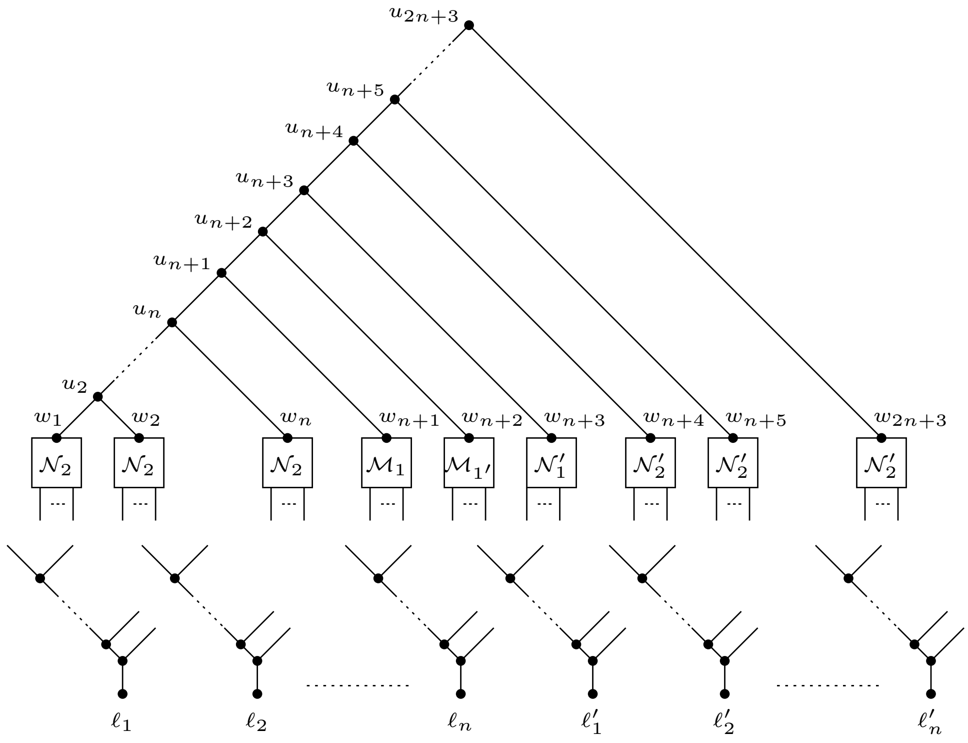


Figure 8: The phylogenetic network $\mathcal{N}_2^*$ on $X \cup X'$ as constructed in the proof of Theorem 4.5. Each of the squares labeled $\mathcal{N}_2$, $\mathcal{M}_1$, $\mathcal{M}_{1'}$, $\mathcal{N}_1'$ and $\mathcal{N}_2'$ refers to the network obtained from its namesake phylogenetic network by deleting all leaf labels. The dangling edges of each square are paired up with the edges shown in the bottom part of the figure as described in Step (6) of the construction in the proof of Theorem 4.5.

Proof. Throughout this proof, let $U = \{u_2, u_3, \ldots, u_{2n+3}\}$ be the vertex set of the backbone of $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$, and let

$$E_U = \{(u_2, w_1), (u_2, w_2), (u_3, w_3), \ldots, (u_{2n+3}, w_{2n+3})\}$$

be the set of edges in $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$ that are directed from a vertex in U to a vertex not in U. Furthermore, for a vertex v and an embedding E, we say that v is *in* E if there exists an edge in E that is incident with v. If v is in E, then we denote this by $v \in E$.

First, suppose that $T(\mathcal{N}_1) \nsubseteq T(\mathcal{N}_2)$. Let $\mathcal{T}_1$ be a phylogenetic X-tree such that $\mathcal{T}_1 \in T(\mathcal{N}_1)$ and $\mathcal{T}_1 \notin T(\mathcal{N}_2)$. Let $\mathcal{T}_1'$ be the phylogenetic X'-tree obtained from $\mathcal{T}_1$ by replacing ℓ_i with ℓ_i' for each $i \in \{1, 2, \ldots, n\}$. Furthermore, let $\mathcal{T}$ be the phylogenetic $(X \cup X')$-tree obtained from $\mathcal{T}_1$ and $\mathcal{T}_1'$ by creating a new vertex ρ, adding an edge that joins ρ with the root of $\mathcal{T}_1$, and adding an edge that joins ρ with the root of $\mathcal{T}_1'$. As $\mathcal{N}_1$ displays $\mathcal{T}_1$ and $\mathcal{N}_1'$ displays $\mathcal{T}_1'$, it is easy to check that an embedding of $\mathcal{T}$ in $\mathcal{N}_2^*$ can be obtained from adding edges of $\mathcal{N}_2^*$ to

$$\{(u_{n+3}, u_{n+2}), (u_{n+2}, u_{n+1}), (u_{n+1}, w_{n+1}), (u_{n+2}, w_{n+2}), (u_{n+3}, w_{n+3})\}$$

such that each element in X is a descendant of u_{n+2}, each element in X' is a descendant of w_{n+3}. Hence, $\mathcal{T}$ is displayed by $\mathcal{N}_2^*$.

We next show that $\mathcal{T}$ is not displayed by $\mathcal{N}_1^*$. Towards a contradiction, assume that $\mathcal{T}$ is displayed by $\mathcal{N}_1^*$. Let E_1 be an embedding of $\mathcal{T}$ in $\mathcal{N}_1^*$. Furthermore, let k be the maximum element in $\{1, 2, \ldots, 2n+3\}$ such that $w_k \in E_1$. By construction of $\mathcal{T}$, either each element in X is a descendant of w_k in E_1 or each element in X' is a descendant of w_k in E_1. Thus, as $\mathcal{N}_2$ does not display $\mathcal{T}_1$ and $\mathcal{N}_2'$ does not display $\mathcal{T}_1'$, we have $k = n + 1$. In particular, each element in X is a descendant of w_k in E_1. But no element in X' is a descendant of u_k in E_1; a contradiction. Hence, $\mathcal{T}$ is not displayed by $\mathcal{N}_1^*$, and so $T(\mathcal{N}_1^*) \neq T(\mathcal{N}_2^*)$.

Second, suppose that $T(\mathcal{N}_1) \subseteq T(\mathcal{N}_2)$. Let $\mathcal{T}$ be a phylogenetic $(X \cup X')$-tree that is displayed by $\mathcal{N}_1^*$, and let E_1 be an embedding of $\mathcal{T}$ in $\mathcal{N}_1^*$. For each $j \in \{1, 2, \ldots, 2n+3\}$ with $w_j \in E_1$, let Y_j be the set that consists of all leaves that are descendants of w_j in E_1, and let $\mathcal{T}_j$ be the phylogenetic tree obtained from the minimal rooted subtree of E_1 that connects all leaves in Y_j by suppressing all vertices with in-degree one and out-degree one. If $w_{n+1} \in E_1$, then, by the pigeonhole principle, there exists an element $j \in \{1, 2, \ldots, n\}$ such that $w_j \notin E_1$. Similarly, if $w_{n+3} \in E_1$, then there exists an element $j' \in \{n+4, n+5, \ldots, 2n+3\}$ such that $w_{j'} \notin E_1$. Without loss of generality, we may therefore assume by the construction of $\mathcal{N}_1^*$ that E_1 satisfies the following property.

(P) If $w_{n+1} \in E_1$, then $w_n \notin E_1$ and, if $w_{n+3} \in E_1$, then $w_{n+4} \notin E_1$.

Intuitively, property (P) allows enough 'play' so that any embedding of a phylogenetic $(X \cup X')$-tree in N_1^* can be replicated in N_2^*.

Since $T(\mathcal{N}_1) \subseteq T(\mathcal{N}_2)$, each tree in $T(\mathcal{N}_1)$ is displayed by $\mathcal{N}_2$, and so each tree in $T(\mathcal{M}'_1)$ is displayed by $\mathcal{N}'_1$, and each tree in $T(\mathcal{M}'_{1'})$ is displayed by $\mathcal{N}'_2$. Hence, there exists an embedding E_2 of a phylogenetic $(X \cup X')$-tree in $\mathcal{N}_2^*$ such that the following conditions are satisfied:

(i) For each $j \in \{1, 2, \ldots, n, n+4, n+5, \ldots, 2n+3\}$, if $w_j \in E_1$, then w_j is the root of a subtree in E_2 that is a subdivision of $\mathcal{T}_j$.

(ii) If $w_{n+1} \in E_1$ and so $w_n \notin E_1$, then w_n is the root of a subtree in E_2 that is a subdivision of $\mathcal{T}_{n+1}$.

(iii) If $w_{n+3} \in E_1$ and so $w_{n+4} \notin E_1$, then w_{n+4} is the root of a subtree in E_2 that is a subdivision of $\mathcal{T}_{n+3}$.

(iv) If $w_{n+2} \in E_1$, then w_{n+3} is the root of a subtree in E_2 that is a subdivision of $\mathcal{T}_{n+2}$.

Since E_1 satisfies (P), E_2 is well defined. By construction of $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$, it now follows that the edges in E_2 are an embedding of $\mathcal{T}$ in $\mathcal{N}_2^*$. Thus $T(\mathcal{N}_1^*) \subseteq T(\mathcal{N}_2^*)$.

Now, let $\mathcal{T}$ be a phylogenetic $(X \cup X')$-tree that is displayed by $\mathcal{N}_2^*$. To see that $\mathcal{T}$ is displayed by $\mathcal{N}_1^*$, we can effectively use the same argument as the one to show that $T(\mathcal{N}_1^*) \subseteq T(\mathcal{N}_2^*)$ even though the assumption that $T(\mathcal{N}_1) \subseteq T(\mathcal{N}_2)$ is not symmetric. In particular, let E_2 be an embedding of $\mathcal{T}$ in $\mathcal{N}_2^*$. For each $j \in \{1, 2, \ldots, 2n+3\}$ with $w_j \in E_2$, let Z_j be the set that consists of all leaves that are descendants of w_j in E_2, and let T_j be the phylogenetic tree obtained from the minimal rooted subtree of E_2 that connects all the leaves in Z_j by suppressing all vertices with in-degree one and out-degree one. If $w_{n+1} \in E_2$, then, by the pigeonhole principle, there exists an element $j \in \{1, 2, \ldots, n\}$ such that $w_j \notin E_2$. Similarly, if $w_{n+3} \in E_2$, then there exists an element $j' \in \{n+4, n+5, \ldots, 2n+3\}$ such that $w_{j'} \notin E_2$. Thus, by construction, we may assume without loss of generality that

(P)′ if $w_{n+1} \in E_2$, then $w_n \notin E_2$ and, if $w_{n+3} \in E_2$, then $w_{n+4} \notin E_2$.

As $T(\mathcal{N}_1) \subseteq T(\mathcal{N}_2)$, each tree in $T(\mathcal{M}_1)$ is displayed by $\mathcal{N}_2$, each tree in $T(\mathcal{M}_{1'})$ is displayed by $\mathcal{N}_1$, and each tree in $T(\mathcal{N}'_1)$ is displayed by $\mathcal{N}'_2$. Thus there exists an embedding E_1 of a phylogenetic $(X \cup X')$-tree in $\mathcal{N}_1^*$ satisfying the following conditions:

(i)′ For each $j \in \{1, 2, \ldots, n, n+4, n+5, \ldots, 2n+3\}$, if $w_j \in E_2$, then w_j is the root of a subtree in E_1 that is a subdivision of $\mathcal{T}_j$.

(ii)′ If $w_{n+1} \in E_2$ and so $w_n \notin E_2$, then w_n is the root of a subtree in E_1 that is a subdivision of $\mathcal{T}_{n+1}$.

(iii)′ If $w_{n+3} \in E_2$ and so $w_{n+4} \notin E_2$, then w_{n+4} is the root of a subtree of E_1 that is a subdivision of $\mathcal{T}_{n+3}$.

(iv)′ If $w_{n+2} \in E_2$, then w_{n+1} is the root of a subtree in E_1 that is a subdivision of $\mathcal{T}_{n+2}$.

It is now easily checked that E_1 is well defined and, by construction of $\mathcal{N}_1^*$ and $\mathcal{N}_2^*$, is an embedding of $\mathcal{T}$ in $\mathcal{N}_1^*$. So $T(\mathcal{N}_2^*) \subseteq T(\mathcal{N}_1^*)$. Combining both cases establishes that $T(\mathcal{N}_2^*) = T(\mathcal{N}_1^*)$. □

This completes the proof of Theorem 4.5. □

5. Conclusion

We end this paper, with two corollaries that are implied by the results presented in Section 3 and two open problems. In 2015, Francis and Steel [4] introduced tree-based networks. A phylogenetic network $\mathcal{N}$ on X is *tree-based* if, up to suppressing vertices of in-degree one and out-degree one, $\mathcal{N}$ displays a phylogenetic X-tree $\mathcal{T}$ that can be obtained by only deleting reticulation edges, in which case, $\mathcal{T}$ is a *base tree* of $\mathcal{N}$. If $\mathcal{N}$ is tree-based, it is well known that not every phylogenetic X-tree displayed by $\mathcal{N}$ is a base tree. However, noting that each tree-child network is also a tree-based network, it is shown in [12] that a phylogenetic tree $\mathcal{T}$ is displayed by a tree-child network $\mathcal{N}$ if and only if $\mathcal{T}$ is a base tree of $\mathcal{N}$. Hence, for two tree-child networks $\mathcal{N}$ and $\mathcal{N}'$, the problem of deciding whether or not $T(\mathcal{N}) \cap T(\mathcal{N}') \neq \emptyset$ is equivalent to deciding whether or not $\mathcal{N}$ and $\mathcal{N}'$ have a common base tree.

Corollary 5.1. *Let $\mathcal{N}$ and $\mathcal{N}'$ be two tree-based networks on X. Then deciding if $\mathcal{N}$ and $\mathcal{N}'$ have a common base tree is* NP*-complete.*

Proof. Let S be a switching of $\mathcal{N}$, and let $\mathcal{T}$ be a phylogenetic X-tree. We say that S is a *base-tree switching* if, for each non-leaf vertex u in $\mathcal{N}$ that is the parent of only reticulations, there exists an edge (u, v) in S. By the definition of a tree-based network it follows that $\mathcal{T}$ is a base tree of $\mathcal{N}$ if and only if there exists a base-tree switching S of $\mathcal{N}$ that yields $\mathcal{T}$. Now, let S be a switching of $\mathcal{N}$, and let S' be a switching of $\mathcal{N}'$. If S is a base-tree switching of $\mathcal{N}$ and S' is a base-tree switching of $\mathcal{N}'$, and S and S' yield the same tree, then $\mathcal{N}$ and $\mathcal{N}'$ have a common base tree. Since it can be checked in polynomial time if S (resp. S') is a base-tree switching of $\mathcal{N}$ (resp. $\mathcal{N}'$), and if S and S' yield the same tree, it follows that deciding whether or not $\mathcal{N}$ and $\mathcal{N}'$ have a common base tree is in NP. The corollary now follows from Theorem 3.2. □

Using (ordinary) switchings instead of base-tree switching, ideas analogous to the ones described in the proof of Corollary 5.1 can be used to show that Common-Tree-Containment is in NP for two arbitrary phylogenetic networks. The next corollary is now an immediate consequence of Theorem 3.2.

Corollary 5.2. Common-Tree-Containment *is* NP*-complete for two arbitrary phylogenetic networks.*

In possible contrast to the last two corollaries and, in particular, Theorem 3.2, we leave it as an open problem to decide on the complexity of Common-Tree-Containment for two level-one networks.

Lastly, let C be a class of phylogenetic networks for which Tree-Containment is solvable in polynomial time such as tree-child or, more generally, reticulation-visible networks [1, 6, 14]. Furthermore, let $\mathcal{N}$ and $\mathcal{N}'$ be two networks in C. Then deciding if $T(\mathcal{N}) = T(\mathcal{N}')$ is in co-NP because, given a tree $\mathcal{T}$ that is displayed by $\mathcal{N}$ or $\mathcal{N}'$, it can be checked in polynomial time, if $\mathcal{T}$ is also displayed by the other network. If this is not the case, then $\mathcal{N}$ and $\mathcal{N}'$ form a no-instance of Display-Set-Equivalence. Whether Display-Set-Equivalence for $\mathcal{N}$ and $\mathcal{N}'$ is co-NP-complete remains an open problem. Nevertheless, it is unlikely that Display-Set-Equivalence for $\mathcal{N}$ and $\mathcal{N}'$ is Π_2^P-complete since a problem that is Π_2^P-complete and in co-NP would imply that co-NP=Π_2^P which, in turn, would result in a collapse of the polynomial hierarchy to the first level.

Acknowledgements. We thank Britta Dorn for insightful discussions as well as the anonymous referees for their careful reading of the paper. The second and third authors thank the New Zealand Marsden Fund for their financial support.

[1] M. Bordewich and C. Semple, Reticulation-visible networks, Advances in Applied Mathematics, 76 (2016), pp. 114–141.
[2] G. Cardona, F. Rosselló, and G. Valiente, Comparison of tree-child phylogenetic networks, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 6 (2009), pp. 552–569.
[3] J. Döcker, S. Linz, and C. Semple, Display sets of normal and tree-child networks, submitted.
[4] A. Francis and M. Steel, Which phylogenetic networks are merely trees with additional arcs? Systematic Biology, 64 (2015), pp. 768–777.
[5] M. R. Garey and D. S. Johnson, Computers and intractability: a guide to the theory of NP-completeness, W. H. Freeman and Company, 1979.
[6] A. D. M. Gunawan, B. DasGupta, and L. Zhang, A decomposition theorem and two algorithms for reticulation-visible networks, Information and Computation, 252 (2017), pp. 161–175.
[7] I. A. Kanj, L. Nakhleh, C. Than, and G. Xia, Seeing the trees and their branches in the network is hard, Theoretical Computer Science, 401 (2008), pp. 153–164.
[8] S. Khuller, Design and analysis of algorithms: course notes, Available at `https://drum.lib.umd.edu/bitstream/handle/1903/592/CS-TR-3113.ps?sequence=1`, 1994
[9] C. McDiarmid, C. Semple, and D. Welsh, Counting phylogenetic networks, Annals of Combinatorics, 19 (2015), pp. 205–224.
[10] B. M. E. Moret, L. Nakhleh, T. Warnow, C. R. Linder, A. Tholse, A. Padolina, J. Sun, and R. Timme, Phylogenetic networks: modeling, reconstructibility, and accuracy, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1 (2004), pp. 13–23
[11] Y. Perl and Y. Shiloach, Finding two disjoint paths between two pairs of vertices in a graph, Journal of the Association for Computing Machinery, 25 (1978), pp. 1–9.
[12] C. Semple, Phylogenetic networks with every embedded phylogenetic tree a base tree, Bulletin of Mathematical Biology, 78 (2016), pp. 132–137.
[13] L. J. Stockmeyer, The polynomial-time hierarchy, Theoretical Computer Science, 3 (1976), pp. 1–22.
[14] L. van Iersel, C. Semple, and M. Steel, Locating a tree in a phylogenetic network, Information Processing Letters, 110 (2010), pp. 1037–1043.
[15] M. Weller, Linear-time tree containment in phylogenetic networks, in Blanchette, M., Ouangraoua, A. (eds), RECOMB-CG 2018, Lecture Notes in Computer Science, vol 11183. Springer, Cham.
[16] S. J. Willson, Properties of normal phylogenetic networks, Bulletin of Mathematical Biology, 72 (2010), pp. 340–358.
[17] S. J. Willson, Tree-average distances on certain phylogenetic networks have their weights uniquely determined, Algorithms for Molecular Biology, 7 (2012):13.