Orienting undirected phylogenetic networks[*]

Katharina T. Huber[†]   Leo van Iersel[‡ §]   Remie Janssen[§]   Mark Jones[§]

Vincent Moulton[†]   Yukihiro Murakami[§]   Charles Semple[¶]

October 5, 2023

Abstract

This paper studies the relationship between undirected (unrooted) and directed (rooted) phylogenetic networks. We describe a polynomial-time algorithm for deciding whether an undirected nonbinary phylogenetic network, given the locations of the root and reticulation vertices, can be oriented as a directed nonbinary phylogenetic network. Moreover, we characterize when this is possible and show that, in such instances, the resulting directed nonbinary phylogenetic network is unique. In addition, without being given the location of the root and the reticulation vertices, we describe an algorithm for deciding whether an undirected binary phylogenetic network N can be oriented as a directed binary phylogenetic network of a certain class. The algorithm is fixed-parameter tractable (FPT) when the parameter is the level of N and is applicable to classes of directed phylogenetic networks that satisfy certain conditions. As an example, we show that the well-studied class of binary tree-child networks satisfies these conditions.

1 Introduction

Phylogenetic networks are graphs which are used to describe, for example, the evolutionary relationships of extant species [HRS10]. Such networks generalize the more widely-known concept of phylogenetic trees. The leaves of such a phylogenetic network represent extant species, while the interior vertices represent hypothetical ancestors.

Phylogenetic networks are usually rooted acyclic directed graphs, where the vertices and arcs combine to represent evolutionary events (e.g., hybridization or horizontal gene transfer). However, unrooted undirected graphs have also been studied which still aim to describe an explicit evolutionary history, but do not include directions on the edges [Mor05]. Reasons for not including directions can be uncertainty about the location of the root and uncertainty about the order in which reticulate events occurred, that is, events where species or lineages merge. Moreover, it can be unclear which vertices represent reticulate events and which vertices represent speciation events or "vertical" descent. See Figure 1 for an example of an undirected and a directed phylogenetic network which illustrates these differences in perspective. Note that unrooted networks are also used as a tool to display patterns within data (e.g., split networks [BM04]) but, as these networks do not aim to explicitly represent the evolution of the underlying species, we do not focus on them here.

In addition to directed and undirected phylogenetic networks, a third option is partly-directed phylogenetic networks, that is, phylogenetic networks in which only some of the edges are oriented. Such networks make sense in light of the discussion above and, indeed, several published phylogenetic networks in the biological literature are partly-directed, e.g., of grape cultivars [MBO+11] and of the evolutionary history of Europeans [Laz18], or
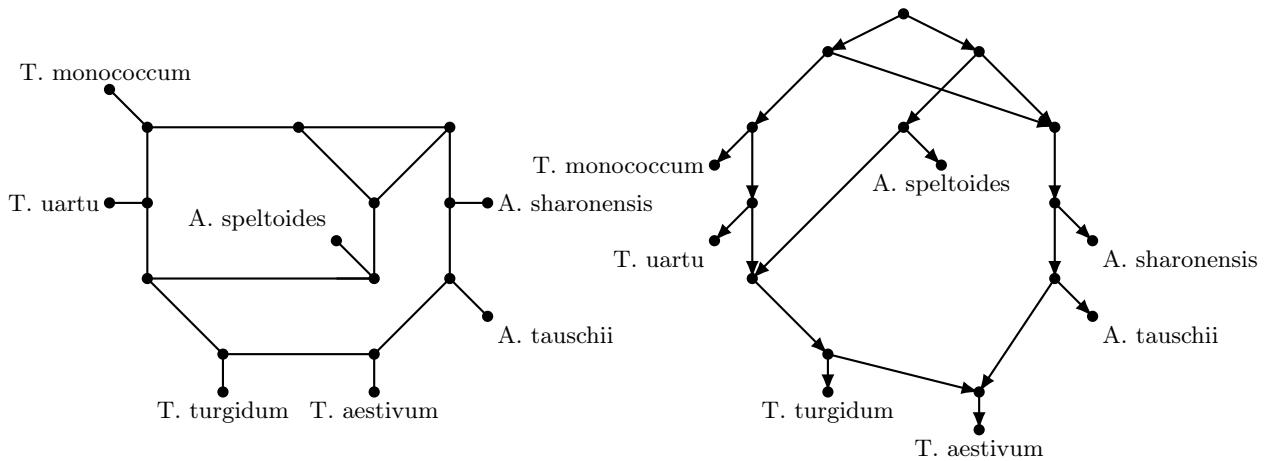

Figure 1: An undirected phylogenetic network (left) and a directed phylogenetic network (right) based on [Mor15, MSH⁺14]. Note that the directed phylogenetic network can be obtained from the undirected phylogenetic network by adding a root vertex and orienting the edges.

contain bi-directed arcs, e.g., of bears [KLB⁺17]. Also, the popular software tool SNAQ produces partly-directed phylogenetic networks [SLA16]. Despite these publications, partly-directed phylogenetic networks have yet to be studied from a mathematical perspective, even though this was suggested by David Morrison in 2013 [Mor13]: "Perhaps the possibility of partly directed phylogenetic networks needs more consideration."

In this paper, we study two fundamental questions regarding the relationship between undirected and directed phylogenetic networks. In the first part of the paper, we investigate the following. Suppose we are given the underlying undirected phylogenetic network of some directed (nonbinary) phylogenetic network N as well as the location of the root of N and the desired in-degrees of the reticulation vertices (the vertices where lineages merge) of N. Does this give us enough information to uniquely reconstruct N? We show that this is indeed the case. Moreover, given the locations of the root and the desired in-degrees of the reticulation vertices, we characterize when an undirected phylogenetic network N' can be oriented as a directed phylogenetic network (see Theorem 1). For an example of an undirected binary phylogenetic network where this is *not* possible, see Figure 2. Following this, we give a linear-time algorithm in the number of edges of N' to find such an orientation. We also show how to apply the algorithm to partly-directed networks. In particular, we show how one can decide in quadratic time in the number of edges whether a given partly-directed network is a semi-directed network, i.e., whether it can be obtained from some directed phylogenetic network by suppressing the root and removing all directions from non-reticulation edges (see Corollary 3).
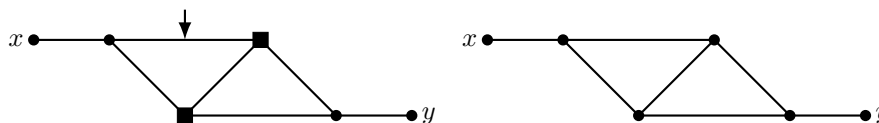

Figure 2: Left, an undirected binary phylogenetic network with specified reticulation vertices (indicated by squares) and root location (indicated with an arrow) that has no orientation as a directed phylogenetic network. Right, the same undirected binary phylogenetic network but with no information about the root and reticulation vertices. This latter undirected binary phylogenetic network can be oriented as a binary stack-free network, but not as a binary tree-child network.

In the second part of the paper, we study the following question. Given an undirected binary phylogenetic network N, can N be oriented to become a directed binary phylogenetic network of a given class (with no information about the location of the root or the reticulation vertices). Again see Figure 2 for an example. We give an algorithm for this task that is fixed-parameter tractable (FPT), where the *level* of N is the parameter (see Algorithm 4). The level of N is a measure of its tree-likeness. (A formal definition is given in the next section.) The algorithm can be applied to a wide range of classes of directed binary phylogenetic networks, including the well-studied classes of tree-child, tree-based, reticulation-visible, and stack-free networks, as well as the recently-introduced classes of valid networks [MvIJ⁺19] and orchard networks [ESS19, JM21]. We include the proof for the class tree-child as an example (see Section 5) since this is one of the most well-studied classes of phylogenetic networks. The proofs for the other classes, following a similar approach, can be found in [HvIJ⁺19, Appendix A]. To obtain this algorithm, we first describe an FPT algorithm where the number of reticulation

vertices is the parameter (see Algorithm 3). The final FPT algorithm (Algorithm 4, which relies on Algorithm 3) may scale better because it has the level as the parameter, which is always smaller or equal to the reticulation number. All of the algorithms in the paper have been implemented and are publicly available [Jan20].

To the best of our knowledge, the questions investigated in this paper have not been studied previously. To date, most publications consider either directed or undirected phylogenetic networks, but do not study how they are related. Exceptions are a paper studying how to optimally root unrooted trees as to minimize their hybridization number [IKS+18] and papers about orienting split networks [ABR19, HKLS05, SM00]. Also see [GHS17] which looks into the relationship between undirected phylogenetic networks and Buneman graphs. There is also a large body of literature on orienting graphs (see, e.g., [AJMO16, AT92]), but such papers are not applicable to our situation because, for example, they do not require the orientation to be acyclic (one exception being [Ata83] which is discussed later) or they do not have our degree restrictions. Lastly, there are two papers that provide results on the orientability of genealogical phylogenetic networks. However, these only provide such results as sidenotes to their main purpose: rearranging networks [JJE+18], and characterizing undirected (unrooted) tree-based networks [FHM18].

2   Preliminaries

Throughout the paper, X denotes a non-empty finite set. Biologically speaking, X can be viewed as a set of extant taxa. An *undirected phylogenetic network* N on X is an undirected connected (simple) graph, in which no vertex has degree 2, and the set of vertices of degree 1 (the *leaves*) is X. We say N is *binary* if each non-leaf vertex has degree 3. An undirected phylogenetic network with no cycles is an *undirected phylogenetic tree*. The *reticulation number* of an undirected phylogenetic network is the number of edges that need to be removed to obtain, after suppressing degree-2 vertices, an undirected phylogenetic tree.

A *directed phylogenetic network* N' on X is a directed acyclic graph with no parallel arcs in which exactly one vertex has in-degree 0 and this vertex has out-degree 2 (the *root*), no vertices have in-degree 1 and out-degree 1, and the set of vertices of out-degree 0 is X and all such vertices have in-degree 1. The vertices of out-degree 0 are the *leaves* of N'. We say N' is *binary* if all non-root non-leaf vertices either have in-degree 1 and out-degree 2, or have in-degree 2 and out-degree 1. Vertices with in-degree at least 2 are *reticulations*, while vertices with in-degree 1 are *tree vertices*. Arcs directed into a reticulation are called *reticulation arcs*. Furthermore, an arc of N' is *pendant* if it is incident to a leaf. If (u, v) is an arc of N', then u is a *parent* of v, and v is a *child* of u. A directed (binary) phylogenetic network with no reticulations is a *directed (binary) phylogenetic tree*.

To avoid ambiguity, when the need arises we will say a "nonbinary phylogenetic network" to mean a phylogenetic network that is not necessarily binary. Furthermore, we note that in the phylogenetics literature the terms *rooted* and *unrooted* phylogenetic network are often used. However, since the location of the root does not necessarily imply the direction of all the arcs, we will use *directed* and *undirected* instead of *rooted* and *unrooted*, respectively.

Two undirected phylogenetic networks N and M on X are *isomorphic* if there exists a bijection f from the vertex set of N to the vertex set of M such that $f(x) = x$ for all $x \in X$, and such that $\{u, v\}$ is an edge of N if and only if $\{f(u), f(v)\}$ is an edge of M. Given an undirected phylogenetic network N on X and a directed phylogenetic network N' on X, we say that N is the *underlying network* of N' and that N' is an *orientation* of N if the undirected phylogenetic network obtained from N' by replacing all directed arcs with undirected edges and suppressing its degree-2 root is isomorphic to N. We say that N is *orientable* if it has at least one orientation.

A *biconnected component* of a directed or undirected phylogenetic network is a maximal subgraph that cannot be disconnected by deleting a single vertex. A biconnected component is called a *blob* if it contains at least three vertices. An undirected phylogenetic network is *level-k* if, by deleting at most k edges from each blob, the resulting graph is a tree, that is, has no cycles. A directed phylogenetic network is *level-k* if its underlying network is level-k. Hence, a directed binary phylogenetic network is level-k if and only if each blob contains at most k reticulations.

A graph is *mixed* if it contains both undirected and directed edges. A *partly-directed* phylogenetic network is a mixed graph that is obtained from an undirected phylogenetic network by orienting a subset of its edges. An *orientation* of a partly-directed phylogenetic network N on X is a directed phylogenetic network on X
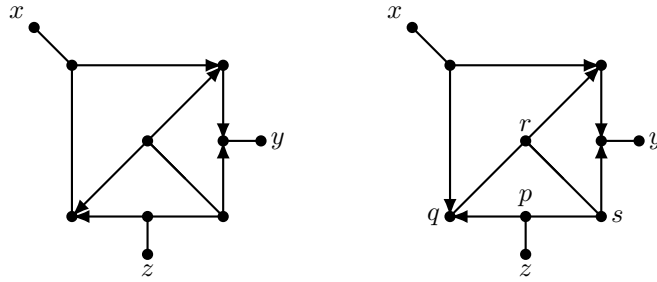
Figure 3: Left, a partly-directed phylogenetic network that is semi-directed (it can be rooted along the pendant edge incident with z). Right, a partly-directed phylogenetic network that is not semi-directed. If it were semi-directed, then a directed phylogenetic network from which it is obtained would have to be rooted along either the pendant edge incident with x or one of the arcs incident to the neighbour of x; otherwise, there is no directed path from the root to x. This makes (p, z) an arc, which implies that p has the incoming arc (s, p). For similar reasons, the orientation must include (r, s) and (q, r). But then, together with (p, q), these arcs form a directed cycle, a contradiction.

that is obtained from N by inserting the root along a directed or undirected edge, and orienting all undirected edges. A *semi-directed* phylogenetic network is a mixed graph obtained from a directed phylogenetic network by unorienting all non-reticulation arcs and suppressing the root. If the root, ρ say, is incident with the arcs (ρ, u) and (ρ, v), where u is a tree vertex and v is a reticulation, then this process replaces (ρ, u) and (ρ, v) with the arc (u, v). Note that, as the root has out-degree 2, it is not the parent of two reticulations. Such networks are of interest because they are used in practical software [SLA16]. A semi-directed phylogenetic network is a partly-directed phylogenetic network but the converse is not true in general, see Figure 3.

We emphasize that we do not allow parallel edges or parallel arcs in (undirected and directed) phylogenetic networks. However, replacing directed arcs of a directed phylogenetic network by undirected edges and suppressing the root may create parallel edges. We do not consider this case explicitly because it can be dealt with easily. In particular, if an undirected phylogenetic network has more than one pair of parallel edges, it cannot be oriented; since the oriented phylogenetic network would contain either a pair of parallel arcs or a directed cycle of length 2. If there is exactly one pair of parallel edges, then, for the same reason, one of these edges needs to be subdivided with the root to obtain an orientation.

Lastly, for an (undirected) graph $G = (V, E)$, let E' and V' be subsets of E and V, respectively. The graph obtained from G by deleting each of the edges in E' is denoted by $G \backslash E'$. Similarly, the graph obtained from G by deleting each of the vertices in V' is denoted by $G \backslash V'$. On the other hand, if A and B are sets, the set obtained from B by deleting each of the elements in $A \cap B$ is denoted by $B - A$.

3 Orienting an undirected phylogenetic network given the root and the desired in-degrees

Suppose that N is an undirected binary phylogenetic network, with a designated edge e_ρ, and R is a subset of the vertices of N. Does there exist an orientation N^r of N whose set of reticulations is R and whose root subdivides e_ρ? In this section, we characterize precisely when there exists such an orientation. Furthermore, we prove that if an orientation exists, then it is unique, and we present a linear-time algorithm that finds N^r.

We start by discussing nonbinary phylogenetic networks, which then allows us to treat binary phylogenetic networks as a special case. In directed nonbinary phylogenetic networks, vertices may have both their in-degree and out-degree greater than 1, in which case knowing the locations of the root and the reticulations may not guarantee a unique orientation of the network (see Figure 4). Therefore, in addition to knowing which vertices are reticulations, we also need to know their desired in-degrees. See Section 6 for a discussion on nonbinary networks in which reticulations are required to have out-degree 1.

In what follows, let $N = (V, E, X)$ denote an undirected nonbinary phylogenetic network on X with vertex set V and edge set E. In addition, let e_ρ denote a designated edge of N where we want to insert the root and, for all $v \in V$, let $d_N^-(v)$ and $d_N(v)$ denote the *desired in-degree* and the *total degree* of v, where $1 \leq d_N^-(v) \leq d_N(v)$, respectively. We say that (N, e_ρ, d_N^-) is *orientable* and that N^r is a *orientation* of (N, e_ρ, d_N^-) if there exists an

Figure 4: Two non-isomorphic directed phylogenetic networks that are both orientations of the same undirected phylogenetic network with the same root location and the same set of reticulations.

orientation N^r of N such that its root subdivides e_ρ and each $v \in V$ has in-degree $d_N^-(v)$ in N^r. Observe that (N, e_ρ, d_N^-) is not orientable if $d_N^-(v) = d_N(v)$ for some non-leaf vertex v of N, or if $d_N^-(l) \neq 1$ for some leaf l of N. This leads to the following decision problem.

CONSTRAINED ORIENTATION
Input: An undirected nonbinary phylogenetic network $N = (V, E, X)$, a distinguished edge $e_\rho \in E$, and a map $d_N^- : V \to \mathbb{N}$ assigning a desired in-degree to each vertex of N.
Output: An orientation of (N, e_ρ, d_N^-) if it exists, and NO otherwise.

3.1 Characterizing the orientability of undirected nonbinary phylogenetic networks

We start by introducing the notion of a degree cut, which will be the key ingredient for characterizing orientability.

Definition 1. Let $N = (V, E, X)$ be an undirected nonbinary phylogenetic network with $e_\rho \in E$ a distinguished edge, and let $N_\rho = (V_\rho, E_\rho, X)$ be the graph obtained from N by subdividing e_ρ by a new vertex ρ. Given the desired in-degree $d_N^-(v)$ of each vertex $v \in V$, a *degree cut* for (N, e_ρ, d_N^-) is a pair (V', E') with $V' \subseteq V$ and $E' \subseteq E_\rho$ such that the following hold in N_ρ:

- E' is an edge cut of N_ρ;

- ρ is not in the same connected component of $N_\rho \backslash E'$ as any $v \in V'$;

- each edge in E' is incident to exactly one element of V'; and

- each vertex $v \in V'$ is incident to at least one and at most $d_N^-(v) - 1$ edges in E'.

The notion of a degree cut is illustrated in Figure 5. Observe that if the desired in-degree of each vertex in V is at most one, then (N, e_ρ, d_N^-) has no degree cut. We say that a degree cut (V', E') for (N, e_ρ, d_N^-) is *minimal* if for any edge $e \in E'$, we have that $(V', E' - \{e\})$ is not a degree cut for (N, e_ρ, d_N^-).
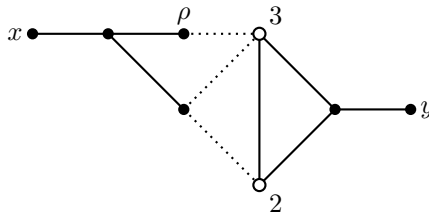
Figure 5: Illustration of a degree cut. Shown is the graph N_ρ obtained from an undirected phylogenetic network N by subdividing an edge e_ρ by a new vertex ρ. Each vertex v with $d_N^-(v) > 1$, represented by an unfilled vertex, is labelled by $d_N^-(v)$. A degree cut (V', E') for (N, e_ρ, d_N^-) is indicated by taking V' to be the set of unfilled vertices and E' to be the set of dashed edges.

We will show in Theorem 1 that the non-existence of a degree cut for (N, e_ρ, d_N^-) together with a condition on the desired in-degrees is equivalent to (N, e_ρ, d_N^-) being orientable. One direction of this theorem is established in Proposition 1(i) and (ii).
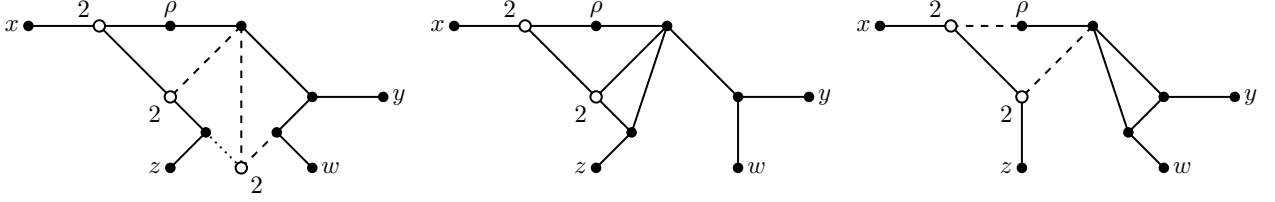
Figure 6: An illustration of Lemma 1. Left, the graph N_ρ obtained from an undirected nonbinary phylogenetic network N by subdividing the edge e_ρ by ρ. Each vertex with $d_N^-(v) > 1$, represented by an unfilled vertex, is labelled by $d_N^-(v)$. The triple (N, e_ρ, d_N^-) has no degree cut. If V denotes the vertex set of N and R denotes the set of unfilled vertices, the dashed edges are those edges with end-vertices in $V - R$ and R that can be reached from e_ρ without traversing an unfilled vertex ('$\{t, r\}$ edges' in the setting of Lemma 1). The dotted edge is an edge with end-vertices in $V - R$ and R that cannot be reached from e_ρ without traversing an unfilled vertex. Middle, the graph N_ρ' obtained by deleting the dashed edge that is incident to the neighbour of w from N_ρ and suppressing the resulting degree-two vertices. Observe that $(N', e_\rho, d_{N'}^-)$, as defined in Lemma 1, has no degree cut. Right, the graph N'' obtained by deleting the dotted edge from N_ρ and suppressing the resulting degree-two vertices. Here, $(N'', e_\rho, d_{N''}^-)$ has a degree cut (the unfilled vertices together with the dashed edges).

Proposition 1. *Let $N = (V, E, X)$ be an undirected nonbinary phylogenetic network, $e_\rho \in E$ be a distinguished edge, and $d_N^-(v)$ be the desired in-degree of each vertex $v \in V$, with $d_N^-(v) = 1$ if v is a leaf and $1 \leq d_N^-(v) < d_N(v)$ otherwise. If (N, e_ρ, d_N^-) is orientable, then each of the following holds:*

    (i) *(N, e_ρ, d_N^-) has no degree cut;*

    (ii) *$\sum_{v \in V} d_N^-(v) = |E| + 1$;*

    (iii) *$N \backslash R$ is a forest, where R is the set of vertices in V with desired in-degree at least two.*

Proof. To prove (i), suppose, for a contradiction, that (N, e_ρ, d_N^-) has a degree cut (V', E'). Consider an orientation N^r of (N, e_ρ, d_N^-). In this orientation, each vertex $v \in V'$ is incident to at most $d_N^-(v) - 1$ arcs corresponding to edges in E'. Hence, each vertex in V' is incident to at least one incoming arc that does not correspond to an edge in E'. Let $v \in V'$ be an arbitrarily chosen vertex, and let e be an incoming arc of v that does not correspond to an edge in E'. Since there is a directed path from ρ to v via e in N^r, and since (V', E') is a degree cut of (N, e_ρ, d_N^-), it must be the case that, prior to e, this path traverses an arc that corresponds to an edge in E'. In particular, this means that there is a directed path from some other vertex in V' to v. Observe that this property holds for all vertices in V', that is, for each $v' \in V'$, there is a directed path from some other vertex in V' to v'. Since V' is finite, this implies that N^r contains a cycle, a contradiction.

For (ii), the total in-degree in an orientation is $\sum_{v \in V} d_N^-(v)$. Since an orientation has $|E| + 1$ edges as edge e_ρ of N is subdivided by ρ, it follows that $\sum_{v \in V} d_N^-(v) = |E| + 1$.

To prove (iii), suppose $N \backslash R$ contains a cycle $C = (v_1, v_2, \ldots, v_1)$. Then, in an orientation N^r of (N, e_ρ, d_N^-), each vertex of C has one incoming and at least two outgoing arcs. Without loss of generality, suppose that $\{v_1, v_2\}$ is oriented from v_1 to v_2 in N^r. Then all other edges incident to v_2 are oriented away from v_2 in N^r, so $\{v_2, v_3\}$ is oriented from v_2 to v_3. By repeating this argument, it follows that N^r has a directed cycle $(v_1, v_2, \ldots, v_1)$. This contradiction completes the proof of (iii) and the proposition. $\square$

We will show later in Corollary 1 that (iii) in Proposition 1 is implied by (i) and (ii). We next prove a lemma which will be used in several proofs. See Figure 6 for an example.

Lemma 1. *Let $N = (V, E, X)$ be an undirected nonbinary phylogenetic network, $e_\rho \in E$ be a distinguished edge, and $d_N^-(v)$ be the desired in-degree of each vertex $v \in V$, with $d_N^-(v) = 1$ if v is a leaf and $1 \leq d_N^-(v) < d_N(v)$ otherwise. Let $R = \{v \in V : d_N^-(v) \geq 2\}$ denote the set of all vertices of N with desired in-degree at least two. Suppose that (N, e_ρ, d_N^-) has no degree cut and $R \neq \emptyset$. Then the following hold:*

    (i) *There exists an edge $\{t, r\} \neq e_\rho$ in N, where $t \in V - R$ and $r \in R$, such that there is a path from an endpoint of e_ρ to t not traversing any vertex in R.*

    (ii) *For any such edge $\{t, r\}$ in (i), $(N', e_\rho', d_{N'}^-)$ has no degree cut, where*

(I) N' is the undirected nonbinary phylogenetic network obtained from N by deleting $\{t, r\}$ and suppressing any resulting degree-two vertices,

(II) $e'_\rho = e_\rho$ unless $e_\rho = \{p, q\}$ and, p say, is suppressed (and so q is not suppressed as $\{t, r\} \neq e_\rho$), in which case, $e'_\rho = \{q, s\}$, where s is the neighbour of p that is not in $\{q, r, t\}$, and

(III) $d^-_{N'}$ is the desired in-degrees of the vertices of N' with

$$d^-_{N'}(v) = \begin{cases} d^-_N(v) - 1, & \text{if } v = r; \\ d^-_N(v), & \text{otherwise,} \end{cases}$$

for all vertices v in N'.

Proof. Let N_ρ be the graph obtained from N by subdividing e_ρ with a vertex ρ. If, in N_ρ, both vertices adjacent to ρ are in R, then these vertices together with the two edges incident with ρ form a degree cut for (N, e_ρ, d^-_N), a contradiction. It follows that at least one vertex adjacent to ρ is not in R. Avoiding ρ, take a path from such a vertex to a vertex $r \in R$, such that no other vertices of the path except r are in R. To show that such a path exists, assume it does not. This is only possible when exactly one neighbour of ρ is in R and there is no path avoiding ρ between the neighbours of ρ, i.e., the edges incident to ρ are cut-edges. In this case, the neighbour of ρ that is in R together with the edge between this vertex and ρ form a degree cut for (N, e_ρ, d^-_N), a contradiction. Hence, there exists a path from a neighbour of ρ that is not in R to a vertex $r \in R$, such that this path does not contain ρ and does not contain any vertices from R except r. Then the last edge on this path is an edge $\{t, r\}$ with $t \in V - R$ and $r \in R$ for which there is a path from ρ to t not using any vertex from R. Note that $t \neq \rho$ since we started the path at a neighbour of ρ. Also note that $r \neq \rho$ since $r \in R$. Hence, the edge $\{t, r\}$ is not incident to ρ and so it is an edge of N. Since it is also an edge of N_ρ, it is not equal to e_ρ. This establishes (i).

To prove (ii), consider any such edge $\{t, r\}$, and let P be a path in N_ρ from ρ to t avoiding vertices in R. Suppose $(N', e'_\rho, d^-_{N'})$ has a degree cut (V', E'). Let N'_ρ be the graph obtained from N' by subdividing e'_ρ with a vertex ρ. Observe that N'_ρ can be obtained from N_ρ by deleting $\{t, r\}$ and suppressing any resulting degree-2 vertices (except ρ). Also note that we can obtain N' from N'_ρ by suppressing ρ and that e'_ρ is the edge created by suppressing ρ. In this proof, we will work with N_ρ and N'_ρ (rather than with N and N') because degree cuts may contain edges incident to ρ.

If t is suppressed when obtaining N'_ρ from N_ρ, let $e_t = \{u, v\}$ denote the resulting edge in N'_ρ, where $u \in P$ (possibly $u = \rho$). Similarly, if r is suppressed when obtaining N'_ρ from N_ρ, let $e_r = \{u', v'\}$ denote the resulting edge in N'_ρ (possibly, $\rho \in \{u', v'\}$).

Let S be the subgraph of $N'_\rho \backslash E'$ consisting of all connected components containing at least one element of V', and let S_ρ be the subgraph of $N'_\rho \backslash E'$ consisting of the remaining connected components of $N'_\rho \backslash E'$. Since (V', E') is a degree cut of $(N', e'_\rho, d^-_{N'})$, the subgraph S_ρ contains ρ. Furthermore, as P contains no vertices in R (so for all $p \in P$, we have $d^-_N(p) \leq 1$), it follows by the fourth property of a degree cut that no $p \in P$ is an element of V'. Hence, the path P cannot contain any edges of E', so any node on P is in the component S_ρ and, in particular, either t or, if t is suppressed, u is contained in S_ρ.

We now derive a contradiction by distinguishing three cases depending on r.

For the first case, assume that either r or, if r is suppressed, e_r is contained in S_ρ. If either t is not suppressed or t is suppressed and $e_t \notin E'$, then (V', E') is a degree cut of (N, e_ρ, d^-_N), a contradiction. Now suppose that t is suppressed and $e_t = \{u, v\} \in E'$. Since $u \in S_\rho$ we have $v \in S$. Then, $(V', (E' - \{e_t\}) \cup \{\{t, v\}\})$ is a degree cut of (N, e_ρ, d^-_N), a contradiction.

For the second case, assume that either r or, if r is suppressed, e_r is contained in S. If either t is not suppressed or t is suppressed and $e_t \notin E'$, then $(V' \cup \{r\}, E' \cup \{\{t, r\}\})$ is a degree cut of (N, e_ρ, d^-_N), a contradiction. Furthermore, if t is suppressed and $e_t \in E'$, then $(V' \cup \{r\}, (E' - \{e_t\}) \cup \{\{t, v\}, \{t, r\}\})$ is a degree cut of (N, e_ρ, d^-_N), a contradiction.

For the last case, assume that r is suppressed and $e_r = \{u', v'\} \in E'$. Without loss of generality, say $v' \in V'$. If either t is not suppressed or t is suppressed and $e_t \notin E'$, then $(V', (E' - \{e_r\}) \cup \{\{r, v'\}\})$ is a degree cut of (N, e_ρ, d^-_N), a contradiction. If t is suppressed and $e_t = \{u, v\} \in E'$, then we have, as before, that $u \in S_\rho$ and $v \in S$. In this case, $(V', (E' - \{e_t, e_r\}) \cup \{\{t, v\}, \{r, v'\}\})$ is a degree cut of (N, e_ρ, d^-_N). This last contradiction completes the proof of the lemma. □

We are now ready to prove the above-mentioned characterization for when an undirected nonbinary phylogenetic network has an orientation respecting a given location for the root and in-degree of every vertex.

Theorem 1. *Let $N = (V, E, X)$ be an undirected nonbinary phylogenetic network, $e_\rho \in E$ be a distinguished edge, and $d_N^-(v)$ be the desired in-degree of each vertex $v \in V$, with $d_N^-(v) = 1$ if v is a leaf and $1 \leq d_N^-(v) < d_N(v)$ otherwise. Then (N, e_ρ, d_N^-) is orientable if and only if (N, e_ρ, d_N^-) has no degree cut and $\sum_{v \in V} d_N^-(v) = |E| + 1$.*

Proof. If (N, e_ρ, d_N^-) is orientable, then, by Proposition 1(i) and (ii), it has no degree cut and $\sum_{v \in V} d_N^-(v) = |E| + 1$. The proof of the converse is by induction on $\sum_{v \in V} d_N^-(v) - |V|$. Note that $\sum_{v \in V} d_N^-(v) - |V| \geq 0$ as $d_N^-(v) \geq 1$ for all $v \in V$. If $\sum_{v \in V} d_N^-(v) - |V| = 0$, then every vertex in V has desired in-degree 1. By assumption, $|V| = \sum_{v \in V} d_N^-(v) = |E| + 1$, and so N is an undirected phylogenetic tree, in which case, (N, e_ρ, d_N^-) is trivially orientable.

Now suppose that $\sum_{v \in V} d_N^-(v) - |V| \geq 1$, and the converse holds for any undirected nonbinary phylogenetic network in which the sum of the given in-degree of each vertex minus the size of its vertex set is at most $(\sum_{v \in V} d_N^-(v) - |V|) - 1$. Let R denote the set of all vertices in V with desired in-degree at least 2. Since $\sum_{v \in V} d_N^-(v) - |V| \geq 1$, it follows that R is nonempty. Let N_ρ be the graph obtained from N by subdividing e_ρ by ρ. By Lemma 1, there exists an edge $\{t, r\}$ in N_ρ with $t \in V - R$ and $r \in R$ for which there is a path from ρ to t not using any vertex from R. In this case, t and r are both vertices of total degree at least 2, and so they cannot be leaves (i.e., t and r must have required outdegree at least 1). Set $(N', e_\rho', d_{N'}^-)$ to be the same as its namesake in the statement of Lemma 1 and let E' be the edge set of N'. Recalling that $d_N(v)$ denotes the degree of a vertex $v \in V$ and $\sum_{v \in V} d_N^-(v) = |E| + 1$, there are four possibilities to consider depending on the degree of t and the degree of r in N:

- If $d_N(t) = 3$ and $d_N(r) = 3$, then both t and r are suppressed in obtaining N', and so

$$\sum_{v \in V'} d_{N'}^-(v) = \sum_{v \in V} d_N^-(v) - 3 = (|E| + 1) - 3 = (|E'| + 3) - 2 = |E'| + 1.$$

- If $d_N(t) = 3$ and $d_N(r) > 3$, then only t is suppressed in obtaining N', and so

$$\sum_{v \in V'} d_{N'}^-(v) = \sum_{v \in V} d_N^-(v) - 2 = (|E| + 1) - 2 = (|E'| + 2) - 1 = |E'| + 1.$$

- If $d_N(t) > 3$ and $d_N(r) = 3$, then only r is suppressed in obtaining N', and so

$$\sum_{v \in V'} d_{N'}^-(v) = \sum_{v \in V} d_N^-(v) - 2 = (|E| + 1) - 2 = (|E'| + 2) - 1 = |E'| + 1.$$

- If $d_N(t) > 3$ and $d_N(r) > 3$, then neither t nor r is suppressed in obtaining N', and so

$$\sum_{v \in V'} d_{N'}^-(v) = \sum_{v \in V} d_N^-(v) - 1 = (|E| + 1) - 1 = |E'| + 1.$$

In all four possibilities, $\sum_{v \in V'} d_{N'}^-(v) = |E'| + 1$. Furthermore, a routine check using the above calculations shows that, for all four possibilities, $\sum_{v \in V'} d_{N'}^-(v) - |V'| < \sum_{v \in V} d_N^-(v) - |V|$. By Lemma 1, $(N', e_\rho', d_{N'}^-)$ has no degree cut; we also have $d_{N'}^-(v) = 1$ if v is a leaf and $1 \leq d_{N'}^-(v) < d_{N'}(v)$ otherwise. It follows by the induction assumption that $(N', e_\rho', d_{N'}^-)$ is orientable. Now consider such an orientation, $(N')^r$ say, and impose the same arc directions on N_ρ except for the edge $\{t, r\}$. If t is suppressed in obtaining N', then $d_{N_\rho}(t) = 3$, in which case, the two edges incident with t that not $\{t, r\}$ are oriented to respect the orientation of the corresponding edge in $(N')^r$. Analogously, the edges incident with r that are not $\{t, r\}$ are orientated in a similar way if $d_{N_\rho}(r) = 3$. Now orient $\{t, r\}$ from t to r, and let N^r denote the resulting orientation of N_ρ. It follows by construction that each vertex in N^r has the correct in-degrees.

It remains to show that N^r is an orientation of (N, e_ρ, d_N^-) by showing that N^r has no directed cycle. If there exists such a cycle, then this directed cycle uses the oriented edge (t, r) as $(N')^r$ has no directed cycle. Hence N^r has a directed path P from r to t. On the other hand, by the choice of t, the directed graph N^r has a directed path Q from ρ to t not using any vertex from R. Since both P and Q end in t, they must meet. Let v be the first vertex on Q meeting P. Then $v \neq \rho$ as P starts at $r \neq \rho$ and both arcs incident with ρ are directed away from ρ. Therefore v has in-degree at least 2. But Q does not contain any vertices in R. This contradiction completes the proof of the theorem. □

A consequence of Theorem 1 is that Proposition 1(iii) is implied by Proposition 1(i) and (ii).

Corollary 1. *Let $N = (V, E, X)$ be an undirected nonbinary phylogenetic network, $e_\rho \in E$ be a distinguished edge, and $d_N^-(v)$ be the desired in-degree of each vertex $v \in V$, with $d_N^-(v) = 1$ if v is a leaf and $1 \leq d_N^-(v) < d_N(v)$ otherwise. Let R denote the set of all vertices in V with desired in-degree at least 2. If (N, e_ρ, d_N^-) has no degree cut and $\sum_{v \in V} d_N^-(v) = |E| + 1$, then $N \backslash R$ is a forest.*

Proof. If (N, e_ρ, d_N^-) has no degree cut and $\sum_{v \in V} d_N^-(v) = |E| + 1$ then, by Theorem 1, (N, e_ρ, d_N^-) is orientable. It now follows by Proposition 1 that $N \backslash R$ is a forest. □

3.2 Orientation algorithm

In this section, we present a polynomial-time algorithm for deciding if, given an undirected nonbinary phylogenetic network N, there is an orientation of N respecting a given location of the root and desired in-degree of each vertex, in which case, the algorithm returns such an orientation. The algorithm is different from the proof of Theorem 1. The main idea of the algorithm is as follows. First we insert the root and orient the edges incident to the root away from it. Then we iteratively look for a vertex that already has the desired number of incoming arcs and at least one incident edge that is not oriented, and orient all incident edges that are not oriented as outgoing arcs. We continue like this until there is no such vertex. We will show (in Theorem 2) that when there is no such vertex, we have either correctly oriented the whole network, or there does not exist an orientation. The pseudo code is as follows.

> **Input:** An undirected nonbinary phylogenetic network $N = (V, E, X)$, an edge $e_\rho \in E$, and the desired in-degree $d_N^-(v)$ for each $v \in V$, with $d_N^-(v) = 1$ if v is a leaf and $1 \leq d_N^-(v) < d_N(v)$ otherwise.
> **Output:** An orientation of (N, e_ρ, d_N^-) if it exists and NO otherwise.

1 **if** $\sum_{v \in V} d_N^-(v) \neq |E| + 1$ **then**
2   **return** NO
3 Subdivide e_ρ by a new vertex ρ and orient the two edges incident to ρ away from ρ;
4 **while** *there exist an unoriented edge* **do**
5   **if** *there is a vertex $v \in V$ with $d_N^-(v)$ incoming oriented edges and at least one incident unoriented edge* **then**
6   orient all unoriented edges incident to v away from v
7   **else**
8   **return** NO
9 **end**
10 **return** the resulting orientation

Algorithm 1: ORIENTATION ALGORITHM(N, e, d_N^-)

Theorem 2. *Let $N = (V, E, X)$ be an undirected nonbinary phylogenetic network, $e_\rho \in E$ be a distinguished edge, and $d_N^-(v)$ be the desired in-degree of each vertex $v \in V$, with $d_N^-(v) = 1$ if v is a leaf and $1 \leq d_N^-(v) < d_N(v)$ otherwise. Then Algorithm 1 decides whether (N, e_ρ, d_N^-) is orientable, in which case, it finds an orientation in time $O(|E|)$. Moreover, this orientation is the unique orientation of (N, e_ρ, d_N^-).*

Proof. By Proposition 1(ii), we may assume that $\sum_{v \in V} d_N^-(v) = |E| + 1$. Let N_ρ denote the graph obtained from N by subdividing e_ρ with ρ. We say that a vertex of N_ρ is *processed* by Algorithm 1 when the algorithm orients its outgoing edges. Note that Algorithm 1 only processes a vertex when it already has at least one incoming oriented edge, and when a vertex is processed all its remaining unoriented edges are oriented outwards.

First suppose that there exists an orientation N^r of (N, e_ρ, d_N^-). We will prove that Algorithm 1 returns N^r. To see this, we first show that if a vertex of N_ρ is processed by Algorithm 1, then every edge incident to this vertex obtains the same orientation as in N^r. Assume, for a contradiction, that this is not the case, and let v be the first vertex processed by Algorithm 1 for which at least one of its incident edges is not oriented as in N^r. Immediately before v is processed, it has $d_N^-(v)$ incoming oriented edges and at least one incident unoriented edge. By the choice of v, the incoming oriented edges of v are oriented the same way as in N^r because the other end-vertices of these edges have already been processed. Algorithm 1 orients all other edges incident to v away from v. These edges are also oriented away from v in N^r, since N^r is an orientation and v is required to have in-degree $d_N^-(v)$. This contradicts the assumption that at least one edge incident to v does not have the same orientation as in N^r. It follows that if there exists an orientation N^r of (N, e_ρ, d_N^-), then every vertex processed

by Algorithm 1 has all its incident edges assigned the same orientation as in N^r. To prove that Algorithm 1 returns N^r, it remains to show that every non-leaf vertex is processed by the algorithm.

Assume that Algorithm 1 stops without having processed all non-leaf vertices. Let P be the set of vertices of N_ρ that have been processed at this point. Let E' be the set of all edges of N_ρ with exactly one end-vertex in P, and let V' be the set of all vertices of N_ρ not in P that are incident to an edge in E'. Every edge $e \in E'$ is incident to one processed vertex $u \in P$ and one unprocessed vertex in V'. By construction, e is oriented away from u and, by the previous argument, e has the same orientation in N^r.

If $v \in V'$, then, as every oriented edge is oriented in the same direction as in N^r, we have that v is incident to at most $d_N^-(v)$ incoming oriented edges. Also, every edge in E' incident to v is oriented towards v. If v is incident to exactly $d_N^-(v)$ edges in E', then v is processed by Algorithm 1, a contradiction. So v is incident to fewer than $d_N^-(v)$ edges in E'. Since E' is an edge cut of N_ρ such that ρ is not in the same connected component of $N_\rho \backslash E'$ as any vertex in V', and each edge in E' is incident to exactly one element of V', it follows that (V', E') is a degree cut for (N, e_ρ, d_N^-), contradicting Proposition 1(i). This last contradiction implies that all non-leaf vertices of N_ρ are processed. Hence, if there exists an orientation N^r of (N, e_ρ, d_N^-), Algorithm 1 will return N^r, and N^r is the unique orientation of (N, e_ρ, d_N^-).

Now suppose that Algorithm 1 returns an orientation N^r of N_ρ. We will prove that N^r is an orientation of (N, e_ρ, d_N^-). It suffices to show that all vertices of N^r have the correct in-degree and out-degree, and N^r has no directed cycle.

Assume that there exists some vertex u in N^r that does not have the correct in-degree and out-degree. Each vertex that is processed (as well as each leaf) always obtains the correct in-degree and out-degree. Hence u has not been processed. Since all edges have been oriented and edges are oriented away from a vertex only if that vertex is processed, it follows that u has in-degree $d_N(v)$, and so u is not a leaf. Thus, $d_N^-(u) < d_N(u)$ and so u has in-degree at least $d_N^-(u) + 1$. By a similar reasoning, all vertices $v \in V$ have in-degree at least $d_N^-(v)$. Hence, as $\sum_{v \in V} d_N^-(v) = |E| + 1$, the total in-degree of N_ρ is at least $\sum_{v \in V} d_N^-(v) + 1 = |E| + 2$. But this implies that the total number of edges in N_ρ is at least $|E| + 2$, a contradiction as N_ρ has $|E| + 1$ edges. Thus, every vertex of N^r has the correct in-degree and out-degree.

Now assume that N^r has a directed cycle. Since every vertex of N^r has the correct in-degree and out-degree, every non-leaf vertex has been processed. Consider the vertex v of the cycle that is processed first. Let u be the neighbour of v on the cycle such that there is an oriented edge e from u to v. As any oriented edge incident to a vertex is oriented away from that vertex when it is processed, e must have been oriented before v was processed. But this implies that u was processed before v, contradicting our choice of v. Thus N^r has no directed cycles and it follows that, if Algorithm 1 returns an orientation of N, it is an orientation of (N, e_ρ, d_N^-).

To complete the proof of the theorem, it remains to show that Algorithm 1 runs in $O(|E|)$ time. A naive implementation takes $O(|V|^2)$ time, as there are $O(|V|)$ vertices to process and it may take $O(|V|)$ time to find the next vertex that can be processed and process it. However, this running time can be improved by observing that any vertex v (apart from the root ρ) only becomes suitable for processing after it has $d_N^-(v)$ incoming oriented edges. Thus it is enough to maintain a set S of such vertices and check, whenever an edge is oriented, whether an unprocessed end-vertex of this edge should be added to S. Then, instead of searching for a new vertex to process each time, we can simply take any vertex from the set S. As each edge is oriented exactly once, the total time spent maintaining S and orienting all edges is $O(|E|)$. $\square$

Partly-directed and semi-directed phylogenetic networks. We end this subsection with two consequences of Theorem 1 and Algorithm 1 concerning partly-directed and semi-directed phylogenetic networks. Recall that a partly-directed phylogenetic network is a mixed graph obtained from an undirected phylogenetic network by orienting some of its edges. Let $N = (V, E, A, X)$ be a partly-directed phylogenetic network on X with vertex set V, undirected edge set E, and directed edge set A. Let $e_\rho \in E$ and, for each $v \in V$, let $d_N^-(v)$ denote the desired in-degree of v. We say that (N, e_ρ, d_N^-) is *orientable* if there is an orientation of N in which the root subdivides e_ρ and, for each $v \in V$, the in-degree of v is $d_N^-(v)$. To decide if (N, e_ρ, d_N^-) is orientable, replace each arc of N by an undirected edge and apply Algorithm 1 to determine whether there exists an orientation. If it exists, it is unique by Theorem 2. Hence, we only need to check whether each arc in A is oriented the same way in the obtained orientation. Thus we have the following corollary of Theorem 2.

Corollary 2. *Let $N = (V, E, A, X)$ be a partly-directed nonbinary phylogenetic network, $e_\rho \in E$ and $d_N^-(v)$ the desired in-degree of each $v \in V$. Then there exists a linear-time algorithm that decides whether (N, e_ρ, d_N^-) is orientable and finds the unique orientation if it exists.*

We now consider semi-directed phylogenetic networks. Recall that a semi-directed phylogenetic network is a mixed graph that is obtained from a directed phylogenetic network by unorienting all non-reticulation arcs and suppressing the root. We noted in Section 2 that a partly-directed phylogenetic network is not necessarily a semi-directed phylogenetic network. Thus a natural question is whether it is easy to decide if a given partly-directed phylogenetic network is semi-directed. Corollary 2 allows us to answer this question positively.

Let $N = (V, E, A, X)$ be a partly-directed nonbinary phylogenetic network on X. If there is a vertex of N with exactly one incoming arc, then N is not semi-directed, so we may assume that there are no such vertices. Let R denote the subset of vertices of N with at least two incoming arcs. For each vertex $v \in V$, define the desired in-degree $d_N^-(v)$ of v as the number of arcs directed into v if $v \in R$; otherwise, set $d_N^-(v) = 1$ if $v \notin R$. For each choice of $e_\rho \in E$, we apply Corollary 2. Then N is semi-directed if and only if (N, e_ρ, d_N^-) is orientable for at least one choice of e_ρ. The running time is $O(|E|^2)$, since there are $|E|$ choices for e_ρ and Algorithm 1 runs in $O(|E|)$ time. Hence we have the following corollary.

Corollary 3. *Let $N = (V, E, A, X)$ be a partly-directed nonbinary phylogenetic network. Then we can decide in $O(|E|^2)$ time whether N is a semi-directed nonbinary phylogenetic network.*

3.3 Characterizing the orientability of undirected binary phylogenetic networks

We now consider the special case of the decision problem CONSTRAINED ORIENTATION for undirected binary phylogenetic networks. Here, rather than being given the desired in-degree of each vertex, we are simply given the set of desired reticulations as all such vertices have in-degree exactly two and all remaining vertices (except the root) have in-degree one.

Definition 2. Let $N = (V, E, X)$ be an undirected binary phylogenetic network with $e_\rho \in E$ a distinguished edge, and let $N_\rho = (V_\rho, E_\rho, X)$ be the graph obtained from N by subdividing e_ρ by a new vertex ρ. Given the set of desired reticulations $R \subseteq V$, a *reticulation cut* for (N, e_ρ, R) is a pair (R', E') with $R' \subseteq R$ and $E' \subseteq E_\rho$ such that the following hold in N_ρ:

- E' is an edge cut of N_ρ;
- ρ is not in the same connected component of $N_\rho \backslash E'$ as any $r \in R'$;
- each edge in E' is incident to exactly one element of R'; and
- $|R'| = |E'|$.

Observe that, if, in the definition of a degree cut, N is binary, then, because of the fourth property of a degree cut, V' is a subset of the set of vertices whose desired in-degree is two. Hence, the definition of a reticulation cut coincides with that of a degree cut when N is binary. We say (N, e_ρ, R) is *orientable* if (N, e_ρ, d_N^-) is orientable, where $d_N^-(r) = 2$ for all $r \in R$ and $d_N^-(v) = 1$ for all $v \in V - R$. An example of a reticulation cut of the triple (N, e_ρ, R) in Figure 2 is illustrated in Figure 7.
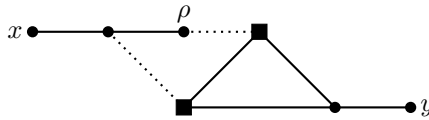

Figure 7: The graph N_ρ obtained from the undirected binary phylogenetic network in Figure 2 by subdividing e_ρ (the edge indicated with an arrow) by a new vertex ρ. The set E' consisting of the dotted edges and the set R' consisting of the two square vertices form the reticulation cut (R', E').

The next proposition is a consequence of Proposition 1.

Proposition 2. *Let $N = (V, E, X)$ be an undirected binary phylogenetic network, $e_\rho \in E$ and $R \subseteq V$. If (N, e_ρ, R) is orientable, then each of the following holds:*

(i) *(N, e_ρ, R) has no reticulation cut;*

(ii) *$|R| = |E| - |V| + 1$;*

(iii) $N\backslash R$ *is a forest.*

To illustrate Proposition 2, the example in Figure 2 satisfies (ii) and (iii) but, as shown in Figure 7, it does not satisfy (i), and hence it is not orientable.

The next theorem is the special case of Theorem 1 when restricted to undirected binary phylogenetic networks. It characterizes when an undirected binary phylogenetic network with given locations for the root and reticulations has an orientation. The correctness of this characterization follows from Theorem 1.

Theorem 3. *Let $N = (V, E, X)$ be an undirected binary phylogenetic network, $e_\rho \in E$ and $R \subseteq V$. Then (N, e_ρ, R) is orientable if and only if (N, e_ρ, R) has no reticulation cut and $|R| = |E| - |V| + 1$.*

4 Orientations within a specific subclass of directed binary phylogenetic networks

We now turn our attention to deciding whether a given undirected binary phylogenetic network has a C-orientation for a given class C of directed binary phylogenetic networks. Unlike CONSTRAINED ORIENTATION we are given no information about the location of the root or the reticulation vertices. Formally, given a class C of directed binary phylogenetic networks, the problem of interest is as follows:

C-ORIENTATION
Input: An undirected binary phylogenetic network N.
Output: A C-orientation of N if it exists, and NO otherwise.

In this section, we present algorithms for solving C-ORIENTATION for classes C of directed binary phylogenetic networks satisfying certain properties. In the next section, Section 5, we will show that the class tree-child (i.e., the class of directed binary phylogenetic networks in which each non-leaf vertex has a child that is a tree vertex) satisfies these properties. In [HvIJ+19, Appendix A] we use a similar approach to show the same holds for the classes stack-free, tree-based, valid, orchard, and reticulation-visible.

This section is organised as follows. First, in Section 4.1, we give an example of the final algorithm for the class tree-child, to get a high-level idea of the approach. Then, in Section 4.2, we give a detailed description of an FPT algorithm for C-ORIENTATION that is parameterized by the reticulation number of N. Subsequently, in Section 4.3, we extend this to an FPT algorithm for C-ORIENTATION but with the level of N as the parameter. These algorithms essentially guess the locations of the root and the reticulations, compute the unique corresponding orientation as in Section 3, and determine whether it is within the required class. To get an FPT running time, N needs to be reduced to a size which is dependent only on the reticulation number (or level) first. We will give such a reduction for any class C of directed binary phylogenetic networks whose members satisfy three certain properties. Intuitively, these properties are as follows. First, membership of C can be checked by considering each blob separately. Second, if N' is a directed binary phylogenetic network in C and new leaves are attached to N', then the resulting directed binary phylogenetic network is also in C. Lastly, the third property is based on reducing "chains" (sequences of leaves whose neighbours form a path). The third property implies that if N' is a directed binary phylogenetic network in C and all chains of N' are reduced to a certain constant length, then the resulting directed binary phylogenetic network N'' is also in C. Additionally, a particular relationship holds between the C-rooted edges of N' and N''. These three properties are formally defined in Definitions 3, 4, and 5, respectively.

4.1 Example

A directed phylogenetic network is said to be *tree-child* if each non-leaf vertex has a child that is a tree vertex. Before describing the algorithms in detail, we first give an example of the FPT algorithm, with the level of N as the parameter, for the case that C is the class of directed binary tree-child phylogenetic networks. The purpose of the example is to give a high-level overview of the approach.

Consider the undirected input phylogenetic network N indicated in Figure 8 (where N is the network obtained from N_C^o by ignoring the direction of the three arcs). We first consider each of the blobs B_1, B_2, B_3 separately by considering the induced networks N_{B_1} (which in this case is equal to N_{B_3}) in Figure 9 and N_{B_2} in Figure 10.
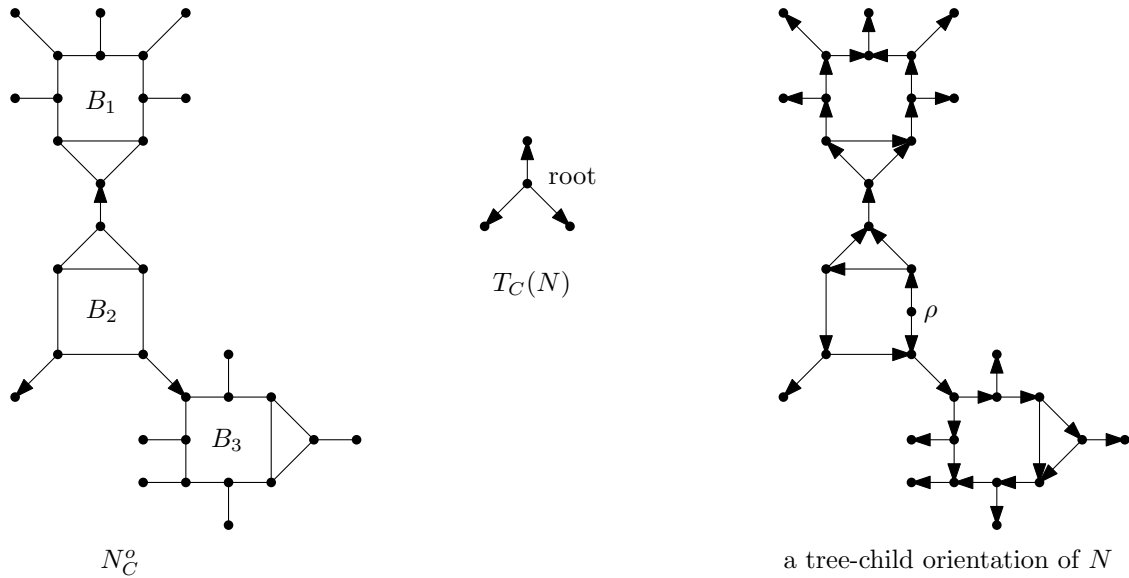
Figure 8: Left: The partly directed network N_C^o. The undirected input network N is obtained by ignoring the direction of the three arcs. The three arcs of N_C^o indicate where the root cannot be. Middle: The tree $T_C(N)$ obtained by contracting all undirected edges in N_C^o. The root of $T_C(N)$ corresponds to blob B_2, from which we learn that, in a tree-child orientation of N, the root location must be in B_2. Right: One possible tree-child orientation of N returned by the algorithm.

In order to obtain an FPT running time, we shorten long chains. We will show in Section 5 that for the class of binary tree-child networks, it is safe to reduce chains to length $\ell = 3$. Hence, in the network N_{B_1} in Figure 9, we reduce the long chain $(c_1, \dots, c_5)$ to a chain (c_1', c_2', c_3') of length 3, this gives the network $N_{B_1}^\ell$. Then we find all possible root locations in $N_{B_1}^\ell$ (indicated with thick edges) using Algorithm 2. For instance, if we choose edge e for the root location, the obtained tree-child orientation is given below $N_{B_1}^\ell$. From this, we can obtain a tree-child orientation of N_{B_1} by adding (and relabelling) leaves. In this example, we chose edge f as the root location in N_{B_1}, leading to the tree-child orientation shown below it.

Now we turn to the network N_{B_2} shown in Figure 10. Since there are no long chains, we immediately use Algorithm 2 to find all possible root locations (indicated with thick edges) and corresponding tree-child orientations. In particular, we see that N_{B_2} can only be rooted at internal (non-pendant) edges.

We now return to Figure 8 to see how the algorithm determines where the root location can be in N. Since N_{B_2} cannot be tree-child rooted at any pendant edge, we orient all cut-edges incident to B_2 away from B_2 (indicating that the root cannot be placed on those arcs or in the parts of the network they point at). Since $N_{B_1} = N_{B_3}$ can be tree-child rooted anywhere, we do not orient any other cut-edges yet. Contracting all undirected edges now gives $T_C(N)$. We will show that when $T_C(N)$ is a rooted tree (as is the case in this example), its root corresponds to possible root locations in N. In this case, the root of $T_C(N)$ corresponds to the blob B_2. Hence, we orient blob B_2 based on a tree-child orientation of N_{B_2}, where the root location is an internal edge. Blob B_1 is oriented based on a tree-child orientation of N_{B_1}, where the root location is the pendant edge corresponding to the cut-edge of N closest to B_2. Blob B_3 is oriented similarly. All cut-edges are oriented away from B_2. This gives the tree-child orientation of N shown in Figure 8 to the right.

For other classes C, the algorithm is exactly the same except for (possibly) the length ℓ that chains are reduced to and how it checks whether a produced oriented network is in the class.

4.2 FPT algorithm parameterized by the reticulation number

For a class C of directed binary phylogenetic networks, we begin by describing a simple exponential-time algorithm, namely, Algorithm 2, that finds all edges of a given undirected binary phylogenetic network where the root can be inserted in order to obtain a C-orientation and, for all such edges, one C-orientation. The FPT algorithm described later in this subsection uses Algorithm 2 as a subroutine. Let N be an undirected binary phylogenetic network, and let e be an edge of N. We say that N can be C-rooted at e if there is a C-orientation of N whose root subdivides e. If this is the case, we also say that e is a C-rooted edge of N. If e is incident to a
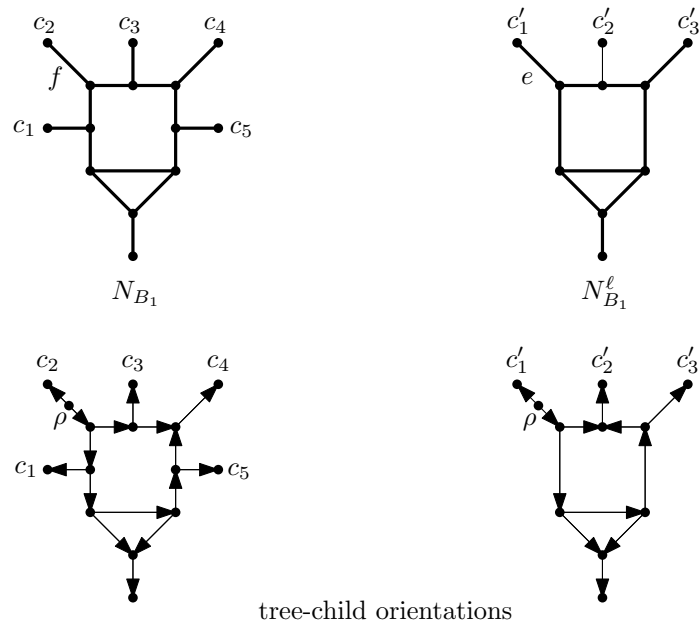
tree-child orientations

Figure 9: Top left: The network $N_{B_1}(= N_{B_3})$ induced by the blob B_1 (or B_3) from Figure 8. Top right: The network $N_{B_1}^\ell$ obtained from N_{B_1} by reducing chains to length 3. Bottom right: A tree-child orientation of $N_{B_1}^\ell$. Bottom left: A tree-child orientation of N_{B_1}. In the undirected networks, thick edges indicate possible root locations.

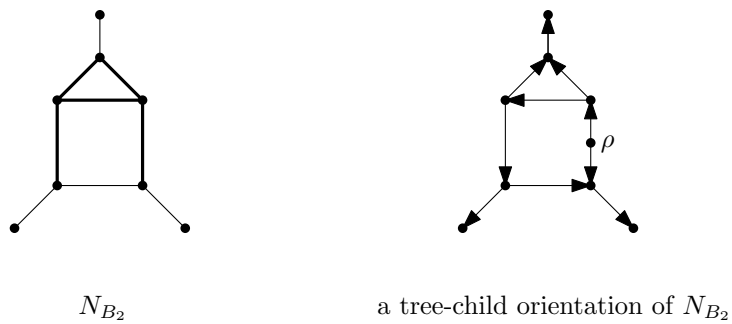N_{B_2} a tree-child orientation of N_{B_2}

Figure 10: Left: The network N_{B_2} induced by the blob B_2 from Figure 8. Thick edges indicate possible root locations. Right: A tree-child orientation of N_{B_2}.

leaf l and N can be C-rooted at e, we say that N can be C-*rooted at* l. For a set X and a non-negative integer n, we let $\binom{X}{n} = \{Y \subseteq X : |Y| = n\}$ denote the set of size n subsets of X.

Input: An undirected binary phylogenetic network $N = (V, E, X)$ with reticulation number k.
Output: The set of C-rooted edges of N and a corresponding C-orientation for each such edge.

1 Set $L := \emptyset$ for the root locations and orientations;
2 **for** *each edge e of N* **do**
3   **for** *each guess $R \in \binom{V}{k}$ of the set of k reticulation vertices* **do**
4   Set $d_N^-(v) = 2$ for each $v \in R$ and $d_N^-(v) = 1$ for each $v \in V \setminus R$;
5   Compute $N(e, R) = \text{ORIENTATION ALGORITHM}(N, e, d_N^-)$ (using Algorithm 1);
6   **if** *$N(e, R)$ is a C-orientation* **then**
7 $L := L \cup \{(e, N(e, R))\}$;
8   Quit the inner for-loop
9   **end**
10   **end**
11 **end**
12 **return** L

Algorithm 2: A simple exponential-time C-orientation algorithm for a class C of directed binary phylogenetic networks.

Note that Algorithm 2 does not necessarily return all C-orientations of N. Indeed, for each edge of N, the inner loop quits (Line 7) after one such orientation is found. To find the complete set of orientations, simply remove this line. The correctness of Algorithm 2 and its running time is established in the next lemma.

Lemma 2. *Let $N = (V, E, X)$ be an undirected binary phylogenetic network with reticulation number k. Then Algorithm 2 applied to N is correct and runs in $O(n^{k+1}(n + f_C(n, k)))$ time, where $n = |V|$ and $f_C(n, k)$ is the time-complexity of checking whether a directed binary phylogenetic network with n vertices and k reticulations is in the class C of directed binary phylogenetic networks.*

Proof. Let e be an edge of N. If N can be C-rooted at e, then there is a set R of k reticulations such that, by Theorem 2, ORIENTATION ALGORITHM(N, e, R) returns a C-orientation of N rooted along e. Since Algorithm 2 checks all possible locations for the k reticulations, it will find such a C-orientation. The correctness of Algorithm 2 now follows.

For the running-time, the outer loop runs $O(n)$ times, as the degree of every vertex of N is at most three and so $|E| \leq \frac{3}{2}n$. The inner loop runs at most $\binom{n}{k}$ times. Inside the inner loop, there are exactly two parts that run in non-constant time. First, by Theorem 2, ORIENTATION ALGORITHM runs in $O(n)$ time and, second, by definition, checking whether a directed binary phylogenetic network with n vertices and k reticulations is in C takes $O(f_C(n, k))$ time. These combine to give a total running time of $O\left(\binom{n}{k}n(n + f_C(n, k))\right)$, that is $O(n^{k+1}(n + f_C(n, k)))$. $\qquad\square$

To obtain an FPT algorithm for C-ORIENTATION, we need to pose some restrictions on the class C. The first of these restrictions is described in Definition 3. For a blob B of a directed binary phylogenetic network, the directed binary phylogenetic network *induced* by B is obtained from B by adjoining, to each vertex v of either in-degree 1 and out-degree 1, or in-degree 2 and out-degree 0, a new leaf x and a new arc (v, x).

Definition 3. A class C of directed binary phylogenetic networks is *blob-determined* if the following property holds: A directed binary phylogenetic network N is a member of C precisely if every network induced by a blob of N is a member of C.

Let N be an undirected (resp. directed) binary phylogenetic network on X, and suppose that e is a cut-edge (resp. cut-arc) of N. A connected component of $N \backslash e$ that is an undirected (resp. directed) phylogenetic tree on X', where $X' \subseteq X$, is called a *pendant phylogenetic subtree* of N. A pendant phylogenetic subtree is *trivial* if it consists of a single leaf; otherwise, it is *non-trivial*. If a class C of directed binary phylogenetic networks is blob-determined, then, in deciding whether an undirected binary phylogenetic network N has a C-orientation, we may assume that N has no non-trivial pendant phylogenetic subtrees. To see this, observe that if N' is an undirected binary phylogenetic network obtained from N by replacing a pendant phylogenetic subtree with a single leaf, say l, then, as C is blob determined and, thus, the existence of a C-orientation depends only on the biconnected components of N, it follows that N' has a C-orientation if and only if N has a C-orientation (see Figure 11). Moreover, if e is the pendant edge of N' incident with l, then N' can be C-rooted at e if and only
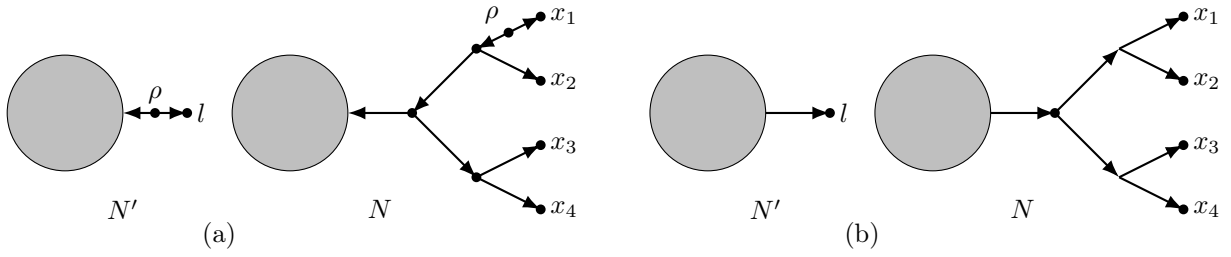
Figure 11: Orientating pendant phylogenetic subtrees. Two examples showing how orientations of an undirected binary phylogenetic network N and the undirected binary phylogenetic network N' obtained from N by replacing a pendant phylogenetic subtree with a single leaf can be derived from each other. In the first example (a), the root is placed along an edge in the pendant phylogenetic subtree of N and, in the second example (b), the root is placed elsewhere. Note that, for each example, the orientation of the edges within the grey circle are the same. Thus, for a blob-determined class C of directed binary phylogenetic networks, N has a C-orientation if and only if N' has a C-orientation.

if N can be C-rooted at each edge of the pendant phylogenetic subtree replaced by l (again, see Figure 11). Hence, we will assume throughout the remainder of Section 4, as well as Section 5, that if N is an undirected binary phylogenetic network, then N has no non-trivial pendant phylogenetic subtrees.

In addition, note that if N is an undirected binary phylogenetic network with reticulation number at most 1, then we can decide whether N can be C-rooted at an edge e by running Algorithm 2, with the running time being a polynomial in the number of vertices and the time needed to check membership of the class C (see Lemma 2). Therefore, we also assume throughout the remainder of Section 4, as well as Section 5, that each undirected binary phylogenetic network has reticulation number at least 2.

To describe the remaining two restrictions, we need some additional definitions. Let N be an undirected (resp. directed) phylogenetic network. *Adding* a leaf to N means that an edge, say $\{u, v\}$ (resp. arc (u, v)), of N is replaced by edges $\{u, w\}, \{w, v\}, \{w, x\}$ (resp. arcs $(u, w), (w, v), (w, x)$), where w is a new vertex and x is a new leaf. The second restriction is described in Definition 4.

Definition 4. A class C of directed binary phylogenetic networks is *leaf-addable* if the following property holds: If N is a member of C and N' is obtained from N by adding leaves, then N' is a member of C.

The *generator* $G(N)$ of an undirected (resp. directed) binary phylogenetic network N is the undirected (resp. directed) multi-graph obtained from N by deleting all (trivial and non-trivial) pendant phylogenetic subtrees together with the edges (resp. arcs) joining the pendant phylogenetic subtrees to the rest of N, and suppressing each of the resulting vertices of degree 2 (resp. in-degree 1 and out-degree 1). Note that if N is undirected, then, for the definition of $G(N)$, we additionally require the reticulation number of N to be at least 2 (which we assume already). Furthermore, $G(N)$ may have parallel edges (resp. arcs), as well as undirected (resp. directed) loops. The edges (resp. arcs) of $G(N)$ are called *sides*.

Let N be an undirected binary phylogenetic network N and let $s = \{u, v\}$ be a side of $G(N)$. Let P_s denote the undirected path in N starting at u and ending at v from which s is obtained in the construction of $G(N)$ by suppressing degree-2 vertices. A leaf x of N is said to be *on* s, and s is said to *contain* x, if x is adjacent to an internal vertex of P_s. Let n_s denote the number of leaves that are on side s. An edge of N is *on* s if it is an edge of P_s. If P_s is the undirected path $u = u_0, e_0, u_1, e_1, \ldots, u_{n_s}, e_{n_s}, u_{n_s+1} = v$ and c_i is the leaf adjacent to u_i for all $i \in \{1, 2, \ldots, n_s\}$, then, relative to P_s, we say that the leaves $c_1, c_2, \ldots, c_{n_s}$ and the edges $e_0, e_1, \ldots, e_{n_s}$ of N on s are *ordered from u to v*. In addition, if e_ρ is a distinguished edge in which we want to insert the root, then s is said to *contain the root* if e_ρ is incident to an internal vertex of P_s, that is either e_ρ is on s or e_ρ is a pendant edge incident to an internal vertex of P_s.

Similarly, if N is a directed binary phylogenetic network and s is a side of $G(N)$, then P_s is the directed path in N from which s is obtained in the construction of $G(N)$ by suppressing vertices of in-degree 1 and out-degree 1. A leaf x of N is said to be *on* s, and s is said to *contain* x, if x is adjacent to an internal vertex of P_s. Let n_s denote the number of leaves that are on side s. An arc of N is *on* s if it is an arc of P_s. If $s = (u, v)$ is a side of $G(N)$, and P_s is the directed path $u = u_0, e_0, u_1, e_1, \ldots, u_{n_s}, e_{n_s}, u_{n_s+1} = v$ and c_i is the leaf adjacent to u_i for all $i \in \{1, 2, \ldots, n_s\}$, then we say that the leaves $c_1, c_2, \ldots, c_{n_s}$ and the arcs $e_0, e_1, \ldots, e_{n_s}$ of N on s are *ordered from u to v*.
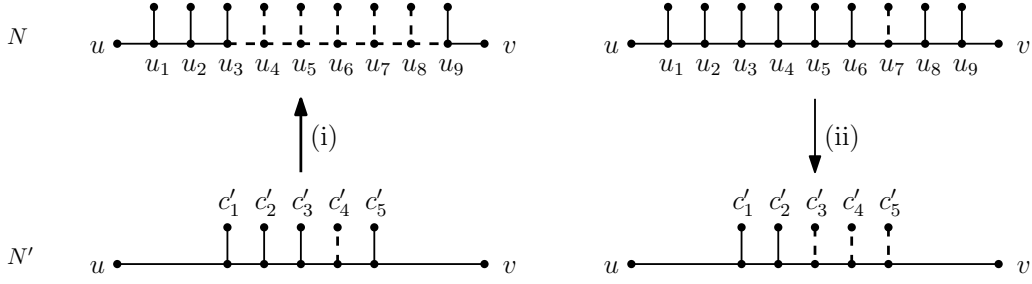
Figure 12: An illustration of an ℓ-chain reduction, and (i) and (ii) of Definition 5 applied to an undirected binary phylogenetic network N. The undirected binary phylogenetic network N' has been obtained from N by an ℓ-chain reduction, where $\ell = 5$. Here the side $s = \{u, v\}$ of $G(N)$ is reduced from $n_s = 9$ to $\ell = 5$. Now suppose that N is 5-chain reducible. To illustrate (i) of Definition 5, if N' can be C-rooted at leaf $\{u'_4, c'_4\}$ (that is, $i = 4$), then N can be C-rooted at all edges incident with u_j for all $j \in \{4, 5, \ldots, 8\}$ (dashed edges). Furthermore, to illustrate (ii) of Definition 5, if N can be C-rooted at the pendant edge incident with u_7, then N' can be C-rooted at $\{u'_i, c'_i\}$ for some $i \in \{1, 2, \ldots, 5\}$ satisfying $7 \in \{i, i+1, \ldots, i+4\}$, that is, for some $i \in \{3, 4, 5\}$ (dashed edges).

Let N be an undirected binary phylogenetic network. Let ℓ be a non-negative integer, and let s be a side of $G(N)$ that contains $n_s \geq \ell$ leaves of N. Then the undirected binary phylogenetic network obtained from N by deleting $n_s - \ell$ leaves that are on s and suppressing any resulting degree-2 vertices is said to be obtained from N by an ℓ-chain reduction on s. More generally, an ℓ-chain reduction on N consists of performing an ℓ-chain reduction on each side of $G(N)$ containing at least ℓ leaves.

The third restriction is described in Definition 5.

Definition 5. Let C be a class of directed binary phylogenetic networks, and let N be an undirected binary phylogenetic network that is C-orientable. Let N' be an undirected binary phylogenetic network obtained from N by an ℓ-chain reduction on N. Suppose that $s = \{u, v\}$ is a side of $G(N)$ that contains at least ℓ leaves of N, and let P_s be the undirected path $u = u_0, u_1, \ldots, u_{n_s}, u_{n_s+1} = v$ of N corresponding to s ordered from u to v. Viewing s as a side of $G(N')$, let $c'_1, c'_2, \ldots, c'_\ell$ denote the leaves of N' on s ordered from u to v and, for all $i \in \{1, 2, \ldots, \ell\}$, let u'_i denote the unique vertex of N' adjacent to c'_i. We say that N is ℓ-chain reducible along s if the following two properties hold:

(i) If N' can be C-rooted at $\{u'_i, c'_i\}$ with $i \in \{1, 2, \ldots, \ell\}$, then N can be C-rooted at all edges incident to u_j for all $j \in \{i, i+1, \ldots, n_s - (\ell - i)\}$.

(ii) If N can be C-rooted at an edge e incident with u_j with $j \in \{1, 2, \ldots, n_s\}$, then N' can be C-rooted at $\{u'_i, c'_i\}$ for some $i \in \{1, 2, \ldots, \ell\}$ satisfying $j \in \{i, i+1, \ldots, n_s - (\ell - i)\}$.

More generally, N is ℓ-chain reducible if N is ℓ-chain reducible along every side of $G(N)$ containing at least ℓ leaves and the following property holds:

(iii) If N can be C-rooted at an edge e that is neither on a side s containing at least ℓ leaves nor incident with a leaf on a side s containing at least ℓ leaves, then N' can also be C-rooted at e.

A class C of directed binary phylogenetic networks is ℓ-chain reducible if every C-orientable undirected binary phylogenetic network is ℓ-chain reducible. This concludes Definition 5.

Properties (i) and (ii) in Definition 5 are illustrated in Figure 12. Figure 13 shows an example where Property (iii) is necessary. Note that we can perform an ℓ-chain reduction on any undirected binary phylogenetic network, but not every such network is ℓ-chain reducible.

Let C be an ℓ-chain reducible, leaf-addable, blob-determined class of directed binary phylogenetic networks. We next describe an FPT algorithm, namely, Algorithm 3, for C-ORIENTATION with the reticulation number of N as the parameter. In the description of Algorithm 3, recall that *adding a leaf x* to a directed network means that an arc (u, v) is subdivided with a new vertex, w say, to create the two arcs (u, w) and (w, v), and that leaf x is added with an arc (w, x) (so, in particular, the orientation of the added pendant arcs is determined).
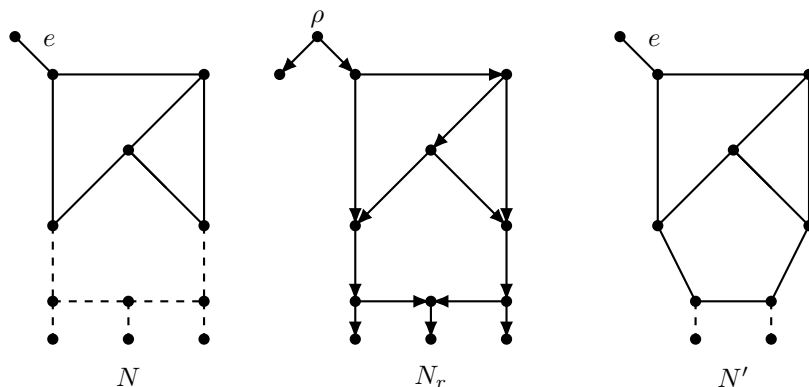
17

Figure 13: An example where (iii) of Definition 5 is not satisfied. Suppose that C is the class of stack-free networks (i.e., networks where no reticulation has a reticulation as a child) and $\ell = 2$. Then the undirected binary phylogenetic network N can be C-rooted at e since the directed network N_r is a C-orientation of N. However, a routine check shows that the undirected binary phylogenetic network N' obtained by performing an ℓ-chain reduction on N cannot be C-rooted at e. Furthermore, both (i) and (ii) of Definition 5 vacuously hold since neither N nor N' can be C-rooted at any of the dashed edges.

In particular, when we add back several leaves to form a chain, we repeat this operation sequentially for each leaf whilst respecting the ordering of the added leaves. See Figure 9 for an example.

As with Algorithm 2, Algorithm 3 finds all of the C-rooted edges of a given undirected binary phylogenetic network, say N, and, for all such edges, it also finds a C-orientation. Loosely speaking, Algorithm 3 starts by performing an ℓ-chain reduction on N to produce an undirected binary phylogenetic network N^ℓ, and then, using Algorithm 2, finds all the C-rooted edges of N^ℓ as well as a C-orientation of N^ℓ for each such edge (Lines 1–2). For each C-rooted edge e of N^ℓ, the algorithm then iteratively finds several C-rooted edges of N "linked" to e via Definition 5. It essentially does this by re-attaching the leaves that were removed in the ℓ-chain reduction (after optionally first removing the leaf edge where the root is located and relocating the root to the resulting degree-2 node). It thus also provides a corresponding C-orientation.

Noting that $G(N) = G(N^\ell)$, let s be the side of $G(N^\ell)$ that contains either e if e is not pendant or the leaf incident to e if e is pendant, and let n_s be the number of leaves of N on s. How this iterative process proceeds depends on whether (i) $n_s < \ell$ (Lines 7–10; uses Definition 5(iii)), (ii) $n_s \geq \ell$ and e is a pendant edge of N^ℓ (Lines 11–30; uses Definition 5(i)), or (iii) $n_s \geq \ell$ and e is not a pendant edge of N^ℓ (Lines 31–33; we argue that we do not need to consider this case explicitly). Most of the work is in (ii) where Algorithm 3 initially handles pendant edges of N linked to e (Lines 16–22) and then handles non-pendant edges of N linked to e (Lines 23–29). The fact that this process finds all C-rooted edges of N as well as a corresponding C-orientation of N for each such edge is established in Lemma 3.

Lemma 3. *Let $N = (V, E, X)$ be an undirected binary phylogenetic network with reticulation number k, where $k \geq 2$. Then Algorithm 3 applied to N is correct and runs in time*

$$O((8\ell(k-1))^{k+1}(\ell(k-1) + f_C(8\ell(k-1), k)) + \ell(k-1)n^2) = O(g(k, \ell) + \ell(k-1)n^2),$$

where $n = |V|$, $f_C(8\ell(k-1), k)$ is the time complexity of checking whether a directed binary phylogenetic network with $8\ell(k-1)$ vertices and k reticulations is in the ℓ-chain reducible, leaf-addable, blob-determined class C of directed binary phylogenetic networks, and g is a function of k and ℓ independent of n.

Proof. To establish the lemma, we use the same notation as in Algorithm 3. To prove correctness, we first show that the algorithm correctly infers C-rooted edges of N from the C-rooted edges of N^ℓ. Let e be a C-rooted edge of N^ℓ, and let s be the side of $G(N^\ell)$ containing either the leaf incident to e if e is pendant, or e if e is not pendant. If $n_s < \ell$, then e is an edge of N and, as C is leaf-addable, it follows that the algorithm correctly concludes that N can be C-rooted at e. On the other hand, if $n_s \geq \ell$, then, as C is ℓ-chain reducible, it follows by Property (i) of Definition 5 that each of the edges of N inferred by Algorithm 3 is a C-rooted edge of N on side s.

We now show that Algorithm 3 finds all C-rooted edges of N. Suppose that N can be C-rooted at edge e_ρ, and let s_ρ be the side of $G(N)$ that contains either the leaf incident to e_ρ if e_ρ is pendant, or e_ρ if e_ρ is not pendant. First suppose that s_ρ contains fewer than ℓ leaves of N. Then e_ρ is an edge of N^ℓ and, by

Input: An undirected binary phylogenetic network N with reticulation number $k \geq 2$ and no non-trivial pendant phylogenetic subtrees.

Output: The set of C-rooted edges of N and a corresponding C-orientation of N for each such edge.

1 Construct an undirected binary phylogenetic network N^ℓ by performing an ℓ-chain reduction on N;

2 Find the set of C-rooted edges of N^ℓ and a corresponding C-orientation N_e^ℓ for each such edge e using Algorithm 2;

3 Set $L := \emptyset$ for the root locations and orientations;

4 **for** *each C-rooted edge e of N^ℓ* **do**

5 $\quad$ Let $s = \{u, v\}$ be the side of $G(N^\ell)$ that contains either the leaf incident to e if e is pendant, or e itself if e is not pendant;

6 $\quad$ Let n_s be the number of leaves of N on s;

7 $\quad$ **if** $n_s < \ell$ **then**

8 $\quad\quad$ Extend N_e^ℓ to a C-orientation N_e of N by adding back the leaves deleted in the reduction (in Line 1) at their original location;

9 $\quad\quad$ Set $L := L \cup \{(e, N_e)\}$;

10 $\quad$ **end**

11 $\quad$ **if** $n_s \geq \ell$ *and e is a pendant edge, say $\{u_i', c_i'\}$, of N^ℓ* **then**

12 $\quad\quad$ Let $c_1', c_2', \ldots, c_\ell'$ be the leaves of N^ℓ on s ordered from u to v;

13 $\quad\quad$ Let $e_0', e_1', \ldots, e_\ell'$ be the edges of N^ℓ on s ordered from u to v;

14 $\quad\quad$ Let $c_1, c_2, \ldots, c_{n_s}$ be the leaves of N on s ordered from u to v;

15 $\quad\quad$ Let $e_0, e_1, \ldots, e_{n_s}$ be the edges of N on s ordered from u to v;

16 $\quad\quad$ **for** *each $j \in \{i, i+1, \ldots, n_s - (\ell - i)\}$* **do**

17 $\quad\quad\quad$ Let f be the pendant edge of N incident to c_j;

18 $\quad\quad\quad$ Modify N_e^ℓ to a C-orientation N_f of N as follows. First, add back $(j-1) - (i-1)$ leaves to an arbitrary arc on the path (along side s) between u_i' and u and add back $(n_s - j) - (\ell - i)$ leaves to an arbitrary arc on the path (along side s) between u_i' and v. Then, (re)label the leaves ordered from u_i' to u as $c_{j-1}, c_{j-2}, \ldots, c_1$, (re)label the leaves ordered from u_i' to v as $c_{j+1}, c_{j+2}, \ldots, c_{n_s}$ and relabel the leaf adjacent to u_i' as c_j. Now extend the resulting orientation by adding back the remaining leaves deleted in the reduction (in Line 1) at their original location;

19 $\quad\quad\quad$ **if** *L does not contain a pair with f as the first element yet* **then**

20 $\quad\quad\quad\quad$ Set $L = L \cup \{(f, N_f)\}$;

21 $\quad\quad\quad$ **end**

22 $\quad\quad$ **end**

23 $\quad\quad$ **for** *each $j \in \{i-1, i, \ldots, n_s - (\ell - i)\}$* **do**

24 $\quad\quad\quad$ Let $f = e_j$;

25 $\quad\quad\quad$ Modify N_e^ℓ to a C-orientation N_f of N as follows. First delete c_i' and the root, relocating the root to u_i'. Second, add back $(j-1) - (i-1)$ leaves to an arbitrary arc on the path (along side s) between u_i' (the new root) and u and add back $(n_s - (j-1)) - (\ell - i))$ leaves to an arbitrary arc on the path (along side s) between u_i' and v, (re)labelling the leaves ordered from u_i' to u and from u_i' to v as $c_{j-1}, c_{j-2}, \ldots, c_1$ and $c_j, c_{j+1}, \ldots, c_{n_s}$, respectively. Now extend the resulting orientation by adding back the remaining leaves deleted in the reduction (in Line 1) at their original location;

26 $\quad\quad\quad$ **if** *L does not contain a pair with f as the first element yet* **then**

27 $\quad\quad\quad\quad$ Set $L = L \cup \{(f, N_f)\}$;

28 $\quad\quad\quad$ **end**

29 $\quad\quad$ **end**

30 $\quad$ **end**

31 $\quad$ **if** $n_s \geq \ell$ *and e is not a pendant edge of N^ℓ* **then**

32 $\quad\quad$ Do nothing as e is incident with a pendant C-rooted edge of N^ℓ, and any corresponding C-orientation of N is constructed in Lines 23–29;

33 $\quad$ **end**

34 **end**

35 **return** L;

Algorithm 3: An FPT algorithm for C-ORIENTATION with the reticulation number of N as the parameter, where C is an ℓ-chain reducible, leaf-addable, and blob-determined class of directed binary phylogenetic networks.

Property (iii) of Definition 5, e_ρ is a C-rooted edge of N^ℓ. Thus, as Algorithm 2 finds all C-rooted edges of N^ℓ with corresponding orientations, the algorithm correctly finds e_ρ and, because C is leaf-addable, a corresponding C-orientation of N.

Now suppose that s_ρ contains at least ℓ leaves of N. We consider two cases depending on whether or not e_ρ is a pendant edge of N. If e_ρ is pendant, then e_ρ is incident to a leaf, say c_j, of N. By Property (ii) of Definition 5, N^ℓ can be C-rooted at c_i' for some i satisfying $j \in \{i, i+1 \ldots, n_s - (\ell - i)\}$. Since Algorithm 2 finds all C-rooted edges of N^ℓ with corresponding orientations, the algorithm will establish that N^ℓ can be C-rooted at c_i' and also find a corresponding C-orientation of N^ℓ. It follows that Algorithm 3 correctly finds that e_ρ is a C-rooted edge of N and, it is easily checked, as C is leaf-addable, a corresponding C-orientation of N in Line 18. If e_ρ is not a pendant edge of N, then e_ρ is incident to a vertex, say u_j which is adjacent to c_j, of N. By Property (ii) of Definition 5, N^ℓ can be C-rooted at c_i' for some i satisfying $j \in \{i, i+1, \ldots, n_s - (\ell - i)\}$. Thus, as Algorithm 2 finds all C-rooted edges of N^ℓ with corresponding orientations, the algorithm establishes that c_i' is a C-rooted edge of N^ℓ and also finds a corresponding C-orientation. Therefore, by the argument in the first paragraph of the proof, Algorithm 3 correctly finds that e_p is a C-rooted edge of N and, it is easily checked, as C is blob-determined and leaf-addable, it finds a corresponding C-orientation of N in Line 25. Hence Algorithm 3 correctly finds all C-rooted edges of N as well as a corresponding C-orientation of N.

Note that all C-rooted edges are indeed found in Lines 7–30, so the case of Line 31 can indeed be ignored in the algorithm.

For the running time, note that Algorithm 3 consists of three separate parts: the ℓ-chain reduction on N to get N^ℓ by deleting leaves (Line 1); the application of Algorithm 2 to find the C-rooted edges of N^ℓ and a corresponding C orientation for each such edge (Line 2); and the inference of the C-rooted edges of N as well as the corresponding C-orientations (Lines 3–34). It is clear that the reduction in Line 1 can be executed in $O(n^2)$ time.

Next we turn to the running time of applying Algorithm 2 to N^ℓ. As each side of the generator of N^ℓ contains at most ℓ leaves, the number of vertices and edges of N^ℓ are bounded by a function of k and ℓ. This makes the running time of Algorithm 2 a function of ℓ and k. To be more concrete, first observe that, since $G(N)$ is cubic, $3|V(G(N))| = 2|E(G(N))|$. Combining this with $k = |E(G(N))| - |V(G(N))| + 1$ (which follows from the definition of the reticulation number) gives $|E(G(N))| = 3(k-1)$ and $|V(G(N))| = 2(k-1)$. Hence, $|V(N^\ell)| \leq 2(k-1) + 6\ell(k-1) \leq 8\ell(k-1)$ and so, by Lemma 2, the running time of the second part is

$$O((8\ell(k-1))^{k+1}(\ell(k-1) + f_C(8\ell(k-1), k))).$$

For the last part, $G(N)$ can be found in $O(n)$ time by deleting all leaves and suppressing their neighbours. As $G(N)$ and $G(N^\ell)$ are isomorphic, each side of $G(N^\ell)$ has at most $\ell + 1$ edges, and so N^ℓ has at most $3(k-1)(2\ell+1)$ edges. For each C-rooted edge of N^ℓ, we modify a C-orientation of N^ℓ at most $2n$ times, each time taking $O(n)$ time. Hence, the running time of this part is $O(\ell(k-1)n^2)$. Taken altogether, the total running time of Algorithm 3 is

$$O((8\ell(k-1))^{k+1}(\ell(k-1) + f_C(8\ell(k-1), k)) + \ell(k-1)n^2).$$

$\square$

The next theorem is an immediate consequence of Lemma 3.

Theorem 4. *Let C be an ℓ-chain reducible, leaf-addable, blob-determined class of directed binary phylogenetic networks. If $f_C(8\ell(k-1), k)$ (described in Lemma 3) is a computable function, then C-ORIENTATION is FPT with the reticulation number of the undirected binary phylogenetic network as the parameter.*

In Section 4.3, we extend Algorithm 3 to an FPT algorithm for C-orientation, where the level of N is the parameter. Before doing this, we conclude this subsection with the following sufficient condition for a network to be C-orientable.

Proposition 3. *Let N be an undirected binary phylogenetic network with at least ℓ leaves on each side of $G(N)$, and let C be an ℓ-chain reducible, leaf-addable class of directed binary phylogenetic networks. If, by adding leaves, there is an undirected binary phylogenetic network that is C-orientable, then N is C-orientable.*

Proof. Let N be an arbitrary undirected binary phylogenetic network with at least ℓ leaves on each side of its generator. Suppose we can add leaves to N to obtain an undirected binary phylogenetic network N' that is

C-orientable. Since C is ℓ-chain reducible, it follows by Properties (ii) and (iii) of Definition 5 that applying an ℓ-chain reduction to N' gives a directed binary phylogenetic network N^ℓ that is C-orientable. Since N can be obtained from N^ℓ by adding leaves and C is leaf-addable, N is C-orientable. □

4.3 FPT algorithm parameterized by the level

Using Algorithms 2 and 3, in this section we establish an FPT algorithm for C-ORIENTATION, where the level of the undirected binary phylogenetic network N is the parameter. The main idea is to orient each blob of N and to combine these orientations into an orientation of N (see Figure 8 for an example). For this second step, we first need the following definitions.

Let N be an undirected binary phylogenetic network and let B be a blob of N. The undirected binary phylogenetic network *induced* by B is obtained from B by adjoining to each degree-2 vertex u a new leaf x and a new edge $\{u, x\}$. Furthermore, for a blob-determined class C of directed binary phylogenetic networks, if N is a member of C, we say that B can be C-*rooted* at a cut-edge $e = \{u, v\}$ of N with $u \in B$ and $v \notin B$ if the undirected binary phylogenetic network induced by B can be C-rooted at the pendant edge incident to u.

Allowing for bi-directed edges, let N_C^o be the mixed graph obtained from N by directing each cut-edge e of N incident to a blob B away from B if B cannot be C-rooted at e. Note that if a cut-edge e joins two blobs of N and neither blob can be C-rooted at e, then this cut-edge becomes bi-directed. Define $T_C(N)$ to be obtained from N_C^o by contracting every undirected edge of N_C^o. Note that (the underlying graph of) $T_C(N)$ is a tree as all edges in the blobs of N are undirected in N_C^o and therefore contracted (and a graph without blobs is a tree). Also note that $T_C(N)$ is not a phylogenetic tree, but a tree in the usual graph-theoretic sense and that all its edges are directed or bidirected.

Let C be an ℓ-chain reducible, leaf-addable, blob-determined class of directed binary phylogenetic networks. The FPT algorithm for C-ORIENTATION with the level of N as the parameter is described as Algorithm 4. The main idea behind the algorithm is captured by the following proposition. A *rooted tree* is a directed tree with a single vertex of in-degree 0, called the *root*, in which all arcs are directed away from the root. Note that a rooted tree may consist of a single vertex.

Proposition 4. *Let C be a blob-determined class of directed binary phylogenetic networks, and let N be an undirected binary phylogenetic network. Then N has a C-orientation if and only if, for each blob B of N, the undirected binary phylogenetic network induced by B has a C-orientation, and $T_C(N)$ is a rooted tree.*

Proof. First assume that N has a C-orientation N', and let B be a blob of N. Since C is blob-determined, N' induces a C-orientation of the undirected binary phylogenetic network N_B induced by B. Now let $\{u, v\}$ be a cut-edge of N, where u is a vertex of B. If $\{u, v\}$ is directed away from B in N_C^o, then B cannot be C-rooted at $\{u, v\}$, and so, as N' is a C-orientation of N, it follows that $\{u, v\}$ is not directed towards u in N'. Thus $\{u, v\}$ is directed away from u in N', that is, $\{u, v\}$ is directed away from B in N'. Therefore if an edge is orientated in N_C^o, then the orientation of that edge is in agreement with its orientation in N'. (In particular, it follows that no edge is bi-directed in N_C^o.) Therefore, by contracting the arcs of N' for which the corresponding edges of N_C^o have no orientation, we obtain $T_C(N)$. Since $T_C(N)$ is obtained from a directed binary phylogenetic network by contracting arcs, and $T_C(N)$ is a tree, it follows that $T_C(N)$ is a rooted tree.

To prove the converse, assume that the undirected binary phylogenetic network induced by each blob of N has a C-orientation and that $T_C(N)$ forms a rooted tree. Let K be the subgraph of N that contracts to the root of $T_C(N)$. Then either (i) K consists of a single blob B of N, or (ii) K contains at least one cut-edge of N. Depending on whether (i) or (ii) holds, we next show that there exists a C-orientation of N where the root is located either on an edge of B, or on a cut edge e of K.

If (i) holds, then, as $T_C(N)$ is a rooted tree (obtained from N_C^o by contracting all undirected edges), all of the cut-edges of N incident to a vertex of B are oriented away from B in N_C^o. Therefore, as the undirected binary phylogenetic network N_B induced by B is C-orientable, there exists an edge e_ρ of B at which N_B can be C-rooted. We now find a C-orientation of N as follows. Subdivide e_ρ by inserting the root, and orient the edges in B the same way as they are orientated in N_B. Orienting all cut-edges of N away from the root, each blob $B' \neq B$ of N now has exactly one incoming cut-arc, say (u, v). Since $T_C(N)$ is a rooted tree, the undirected binary phylogenetic network induced by B' can be C-rooted at the cut-edge incident to v. Orienting the edges of B' (and all other such blobs of N) accordingly, gives a C-orientation of N. For (ii), we subdivide the cut-edge

Input: An undirected binary phylogenetic network N with no non-trivial pendant phylogenetic subtrees.

Output: A C-orientation of N if it exists, and NO otherwise.

1 Find the set of blobs of N;

2 **for** *each blob B of N* **do**

3   Apply Algorithm 3 to the undirected binary phylogenetic network N_B induced by B and let L_B be the returned set of pairs (e, B_e) consisting of C-rooted edge e and corresponding orientation B_e of N_B;

4   **if** $L_B = \emptyset$ **then**

5   **return** NO;

6   **end**

7 **end**

8 Construct N_C^o from N by orienting each cut-edge e of N incident to a vertex of a blob B away from B if there is no pair in L_B with e as first element (possibly orienting edges in two directions);

9 Construct $T_C(N)$ from N_C^o by contracting all non-oriented edges in N_C^o;

10 **if** $T_C(N)$ *is a rooted tree* **then**

11   Determine the subgraph K of N that is contracted, in Line 9, to the root of $T_C(N)$;

12   **if** K *consists of a single blob B of N* **then**

13   Pick an arbitrary element $(e, B_e) \in L_B$ and orient B in N according to B_e, calling the root vertex ρ;

14   **end**

15   **if** K *contains a cut-edge* **then**

16   Subdivide an arbitrary cut-edge e by the root ρ;

17   **end**

18   Orient all cut-edges of N away from ρ;

19   **for** *each unoriented blob B' of N* **do**

20   Find the cut-arc (u, v) entering B';

21   Let $\{v, x\}$ be the cut-edge incident to v in the network induced by B';

22   Find a pair $(\{v, x\}, B'_{\{v,x\}}) \in L_{B'}$;

23   Orient the edges of B' in N as in $B'_{\{v,x\}}$;

24   **end**

25   **return** the oriented network N;

26 **else**

27   **return** NO;

28 **end**

Algorithm 4: An FPT algorithm for C-ORIENTATION with the level of N as the parameter, where C is an ℓ-chain reducible, leaf-addable, and blob-determined class of directed binary phylogenetic networks.

---

e of K by the root and proceed in the same way as for (i), starting by orienting all cut-arcs away from the root. $\qquad\square$

The correctness of Algorithm 4 and its running time is established in the next lemma.

Lemma 4. *Let $N = (V, E, X)$ be an undirected binary phylogenetic network. Then Algorithm 4 applied to N is correct and runs in time $O(g(L, \ell)n + \ell(L-1)n^3))$ if C is an ℓ-chain reducible, leaf-addable, and blob-determined class of directed binary phylogenetic networks, where $n = |V|$, L is the level of N, and g is a function of L and ℓ independent of n.*

Proof. The correctness of Algorithm 4 is essentially given in the proof of Proposition 4, and so it is omitted. For the running time, first note that all blobs can be found in $O(n^3)$ time by checking for each edge whether it is a cut-edge. The rest of the algorithm consists of two parts. The first part consists of finding all C-rooted edges of the undirected binary phylogenetic networks induced by the blobs of N and a corresponding C-orientation for each such edge (Lines 1–8), while the second part consists of constructing N_C^o and $T_C(N)$ and, provided $T_C(N)$ is a rooted tree, finding an orientation of the cut-edges and blob edges of N (Lines 9–29).

By Lemma 3, for a blob B with n_B vertices and k_B reticulations, running Algorithm 3 on the undirected binary phylogenetic network induced by B takes $O(g(k_B, \ell) + \ell(k_B - 1)n_B^2)$ time. Since N has at most n blobs and $k_B \leq L$ by the definition of level, the first part of Algorithm 4 runs in time

$$O(g(L, \ell)n + \ell(L - 1)n^3).$$

For the second part of Algorithm 4, we initially construct N_C^o and $T_C(N)$. Orientating the cut-edges of N incident to blob vertices to obtain N_C^o and then contracting the unorientated edges of N_C^o to obtain $T_C(N)$ takes $O(n^2)$ time. Once this is completed, the second part of the algorithm requires only one pass through N to orient its edges, as we may independently pick an orientation for each blob B from the set of orientations L_B with the correct root-edge (finding such orientation in the set may take $O(n)$ time). Hence, the second part of the algorithm only takes $O(n^2)$ time. This completes the proof of the lemma. □

Remark. If C is not necessarily ℓ-chain reducible and leaf-addable, but is a blob-determined class of directed binary phylogenetic networks, then we can adapt Algorithm 4 by replacing Line 3 with the following to obtain an algorithm for deciding if an undirected binary phylogenetic network N has a C-orientation:
Let L_B be the output of Algorithm 2 applied to the undirected binary phylogenetic network induced by B.
Taking the same approach as the proof of Lemma 4, the running time of this adaption is $O(\binom{n}{L}n^2(n+f_C(n,L)))$, where n is the number of vertices of N, and L is the level of N.

The next theorem is an immediate consequence of Lemma 4.

Theorem 5. *Let C be an ℓ-chain reducible, leaf-addable, blob-determined class of directed binary phylogenetic networks, for any fixed ℓ. If $g(L, \ell)$ is a computable function, then Algorithm 4 is an FPT algorithm for C-ORIENTATION, where the level L of the inputted undirected binary phylogenetic network is the parameter.*

5   A specific class

Recall that a directed phylogenetic network is said to be *tree-child* if each non-leaf vertex has a child that is a tree vertex. The main result of this section is Theorem 6, which establishes that C-ORIENTATION is FPT when C is the class of directed binary tree-child phylogenetic networks.

The proof of Theorem 6 relies on combining Theorem 5 with proofs showing that the class of directed binary tree-child phylogenetic networks is ℓ-chain reducible (with $\ell = 3$), leaf-addable, and blob-determined. To establish the ℓ-chain reducible property, we will show that this class satisfies a variant of this property, called rooted ℓ-chain reducible, which is described in Section 5.1. We then show that if a directed binary phylogenetic class C is rooted ℓ-chain reducible, leaf-addable, and blob-determined, then C is ℓ-chain reducible.

The same technique can be applied to many other known classes, see [HvIJ+19, Appendix A].

Theorem 6. *Algorithm 4 is an FPT algorithm for deciding whether an undirected binary phylogenetic network N has a tree-child orientation, where the level of N is the parameter.*

5.1   Rooted ℓ-chain reduction

We begin by defining the operation of rooted ℓ-chain reduction. Note that this operation is defined on undirected binary phylogenetic networks, but with a specified pendant edge e_ρ which will be used as the root location. We also remark that we will use the term "rooted ℓ-chain reduction" to refer to a network obtained by this operation, as well as to refer to the operation itself. Recall that we assume throughout Sections 4 and 5 that networks have no nontrivial pendant phylogenetic subtrees and that they have reticulation number at least 2.

Definition 6. Let N be an undirected binary phylogenetic network, let ℓ be a non-negative integer, and let e_ρ be a pendant edge of N. Furthermore, let $s_\rho = \{u, v\}$ be the side of $G(N)$ containing e_ρ, and let P_{s_ρ} denote the undirected path of N corresponding to s_ρ between u and v. We call an undirected binary phylogenetic network obtained from N by applying the following three operations a *rooted ℓ-chain reduction (from u, with respect to e_ρ)* on N:

(i) for each side s of $G(N)$ other than s_ρ that contains at least ℓ leaves, delete $n_s - \ell$ leaves on s, where n_s is the number of leaves on s;

(ii) delete all leaves on s_ρ that are adjacent to an internal vertex of P_{s_ρ} between u and the end vertex of e_ρ on P_{s_ρ}; and

(iii) if there are at least ℓ leaves adjacent to an internal vertex of P_{s_ρ} between the end vertex of e_ρ on P_{s_ρ} and v, then delete all but $\ell - 1$ of these leaves; otherwise, if there are at most $\ell - 1$ such leaves, do nothing.
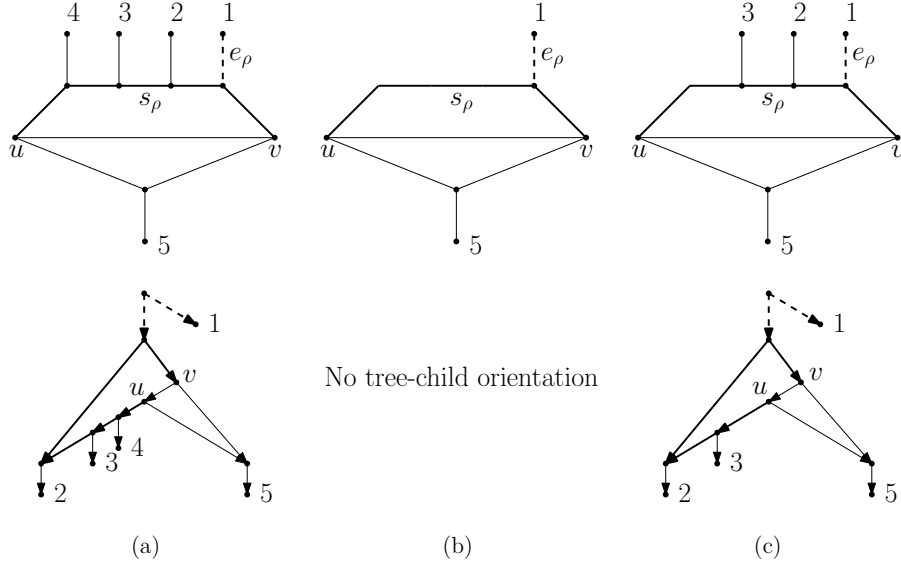
Figure 14: An example of rooted 3-chain reductions. Subfigure (a) shows an undirected binary phylogenetic network N that has a tree-child orientation rooted at edge e_ρ as shown below it. The side of the generator $G(N)$ that contains the root is denoted $s_\rho = \{u, v\}$. As the class of tree-child networks is rooted 3-chain reducible, with respect to e_ρ, a rooted 3-chain reduction on N from at least one of u and v results in an undirected binary phylogenetic network that can be tree-child rooted at e_ρ. Subfigure (b) shows a rooted 3-chain reduction on N from u, but, as indicated in (b), it cannot be tree-child rooted at e_ρ. However, as shown in Subfigure (c), a rooted 3-chain reduction from v results in an undirected binary phylogenetic network that can be tree-child rooted at e_ρ.

Definition 7. A class C of directed binary phylogenetic networks is *rooted ℓ-chain reducible* if the following property holds: Let N be an undirected binary phylogenetic network and let e_ρ be a pendant edge of N. Let $s_\rho = \{u, v\}$ be the side of $G(N)$ containing e_ρ and let N_u and N_v be rooted ℓ-chain reductions from u and v, respectively, with respect to e_ρ. If N can be C-rooted at e_ρ, then at least one of N_u and N_v can be C-rooted at e_ρ.

An example illustrating these definitions is given in Figure 14, where C is the class of binary tree-child networks. Note that, we will eventually show that the class of binary tree-child networks is rooted 3-chain reducible (Lemma 8).

The next two lemmas will be used to show that if a class of directed binary phylogenetic networks is rooted ℓ-chain reducible, leaf-addable, and blob-determined, then it is ℓ-chain reducible.

Lemma 5. *Let C be a rooted ℓ-chain reducible, leaf-addable class of directed binary phylogenetic networks, and let N be an undirected binary phylogenetic network that is C-orientable. Let N' be the undirected binary phylogenetic network obtained from N by an ℓ-chain reduction on N. Suppose that $s = \{u, v\}$ is a side of $G(N)$ that contains at least ℓ leaves. Let $c_1, c_2, \ldots, c_{n_s}$ denote the leaves of N on s ordered from u to v, and let $c'_1, c'_2, \ldots, c'_\ell$ denote the leaves of N' on s ordered from u to v. Then each of the following hold:*

(i) *If $i \in \{1, 2, \ldots, \ell\}$ and N' can be C-rooted at c'_i, then N can be C-rooted at c_j for all $j \in \{i, i+1, \ldots, n_s - (\ell - i)\}$.*

(ii) *If $j \in \{1, 2, \ldots, n_s\}$ and N can be C-rooted at c_j, then N' can be C-rooted at c'_i for some i satisfying $j \in \{i, i+1, \ldots, n_s - (\ell - i)\}$.*

Proof. For (i), suppose that N' can be C-rooted at c'_i, where $i \in \{1, 2, \ldots, \ell\}$, and let N_i^ℓ be a C-orientation of N' rooted at c'_i. Let $j \in \{i, i+1, \ldots, n_s - (\ell - i)\}$. Now construct an orientation N_j of N from N_i^ℓ as follows. First, add back $j - i$ leaves on the directed path from the neighbour u'_i of c'_i to u and add back $(n_s - j) - (\ell - i)$ leaves on the directed path from u'_i to v relabelling the leaves ordered from u'_i to u and u'_i to v as $c_{j-1}, c_{j-2}, \ldots, c_1$ and $c_{j+1}, c_{j+2}, \ldots, c_{n_s}$, respectively, and relabelling the leaf adjacent to u'_i as c_j. Note that, as $j \le n_s - (\ell - i)$, it follows that $(n_s - j) - (\ell - i) \ge 0$. Now extend the resulting orientation by adding back the remaining leaves deleted in the reduction at their original location. This gives N_j, an orientation of

N rooted at c_j. Since C is leaf-addable and N_i^ℓ is a C-orientation, it follows that N_j is a C-orientation of N rooted at c_j. This establishes (i).

To prove (ii), suppose that N can be C-rooted at c_j, where $j \in \{1, 2, \ldots, n_s\}$. Since C is rooted ℓ-chain reducible, there is a rooted ℓ-chain reduction N'' on N with respect to the edge incident with c_j that can be rooted at c_j. Without loss of generality, we may assume that in this reduction we deleted all the leaves on s between the neighbour u_j of c_j and u, and if there are at least ℓ leaves on s between u_j and v, we deleted all but $\ell - 1$ of these leaves.

First assume that $j \geq n_s - (\ell - 1)$, and let $j = n_s - t$, where $t \leq \ell - 1$. In this case, no leaves of N on s are deleted between u_j and v to obtain N''. Thus N'' has exactly $t + 1$ leaves on s and (by definition of c_j and of rooted ℓ-chain reduction) N'' can be C-rooted at c_j the first leaf on s ordered from u to v. Let N_j^ℓ denote a C-orientation of N'' rooted at c_j. We next construct an orientation N_i^ℓ of N' from N_j^ℓ as follows. Add back $\ell - (t + 1)$ leaves on the directed path from u_j to u, so that we have exactly ℓ leaves on s, and relabel the leaves ordered from u_j to u and from u_j to v as $c'_{i-1}, c'_{i-2}, \ldots, c'_1$ and $c'_{i+1}, c'_{i+2}, \ldots, c'_\ell$, respectively, and relabel c_j as c'_i. This gives N_i^ℓ. Since $G(N_i^\ell)$ is isomorphic to $G(N')$ and each side s of $G(N_i^\ell)$ and $G(N')$ contains the same number of leaves, it follows that, up to relabelling the leaves on each side, N_i^ℓ is an orientation of N'. Thus, as C is leaf-addable, N' has a C-orientation rooted at c'_i, where $i = \ell - t = \ell - (n_s - j)$. Since $n_s - (\ell - i) = j$, we have $j \in \{i, i + 1, \ldots, n_s - (\ell - i)\}$ as required.

Now assume that $j < n_s - (\ell - 1)$. Then, by applying the rooted ℓ-chain reduction, we delete all leaves of N on s between u and u_j while keeping the network C-rootable at the leaf-edge incident to u_j. Hence, we have that N'' can be C-rooted at the first leaf on s ordered from u to v. Moreover, as $j < n_s - (\ell - 1)$, side s of N'' contains exactly ℓ leaves. Therefore, as $G(N')$ is isomorphic to $G(N'')$, up to relabelling the leaves on each side, N' is isomorphic to N''. Hence, N' can be C-rooted at c'_1. Since $j \geq i$ and $n_s - (\ell - i) \geq n_s - (\ell - 1) > j$ as $i \geq 1$, we have $j \in \{i, i + 1, \ldots, n_s - (\ell - i)\}$, again, as required. This completes the proof of (ii) and the lemma. □

The next lemma is the non-pendant edge analogue of Lemma 5.

Lemma 6. *Let C be a rooted ℓ-chain reducible, leaf-addable, blob-determined class of directed binary phylogenetic networks, and let N be an undirected binary phylogenetic network that is C-orientable. Let N' be the undirected binary phylogenetic network obtained from N by an ℓ-chain reduction on N. Suppose that $s = \{u, v\}$ is a side of $G(N)$ that contains at least ℓ leaves. Let $e_0, e_1, \ldots, e_{n_s}$ denote the edges of N on s ordered from u to v, and let $c'_1, c'_2, \ldots, c'_l$ denote the leaves of N' on s ordered from u to v. Then each of the following hold:*

(i) *If $i \in \{1, 2, \ldots, \ell\}$ and N' can be C-rooted at c'_i, then N can be C-rooted at e_j for all $j \in \{i - 1, i, \ldots, n_s - (\ell - i)\}$.*

(ii) *If $j \in \{0, 1, \ldots, n_s\}$ and N can be C-rooted at e_j, then N' can be C-rooted at c'_i for some i satisfying $j \in \{i - 1, i, \ldots, n_s - (\ell - i)\}$.*

Proof. Let e_j be an edge of N on s, and let N_j be the undirected binary phylogenetic network obtained from N by subdividing e_j with a new vertex and adjoining a new leaf to this vertex via a new edge. Let $c_1, c_2, \ldots, c_{n_s+1}$ denote the leaves of N_j on s ordered from u to v. Thus c_{j+1} is the new leaf. Note that $G(N) = G(N_j)$. Since C is blob-determined, N_j can be C-rooted at c_{j+1} if and only if N can be C-rooted at e_j. Let N'_j be the undirected binary phylogenetic network obtained from N_j by an ℓ-chain reduction on N_j.

For the proof of (i), assume that N' can be C-rooted at c'_i, where $i \in \{1, 2, \ldots, \ell\}$. Let $j \in \{i - 1, i, \ldots, n_s - (\ell - i)\}$. Up to relabelling leaves, N'_j is isomorphic to N', and so N'_j can be C-rooted at the i-th leaf on s ordered from u to v. Therefore, as N_j has $n_s + 1$ leaves on side s, it follows by Lemma 5(i) applied to N_j that N_j can be C-rooted at $c_{j'}$ for all $j' \in \{i, i + 1, \ldots, (n_s - (\ell - i)) + 1\}$. In particular, as $j + 1 \in \{i, i + 1, \ldots, (n_s - (\ell - i)) + 1\}$, we have that N_j can be C-rooted at c_{j+1}. Thus N can be C-rooted at e_j.

To prove (ii), assume that N can be C-rooted at e_j, where $j \in \{0, 1, \ldots, n_s\}$. Then N_j can be C-rooted at c_{j+1}. To see this, take a C-orientation of N rooted at e_j, and orient the edges of N_j, except the pendant edge incident with c_{j+1}, in the same direction as the corresponding edges of the C-orientation of N. Now subdivide the edge incident with c_{j+1} by a vertex w and orient the two edges incident with w away from it. The resulting directed binary phylogenetic network is a C-orientation of N_j rooted at c_{j+1}.

25

By Lemma 5(ii), N'_j can be C-rooted at c'_i for some i satisfying $j + 1 \in \{i, i+1, \ldots, (n_s - (\ell - i)) + 1\}$. Since N'_j is isomorphic to N' up to relabelling leaves, N' can be C-rooted at c'_i for some i satisfying $j \in \{i - 1, i, \ldots, n_s - (\ell - i)\}$. This completes the proof of (ii) and the lemma. $\qquad\square$

A consequence of the last two lemmas is the following proposition.

Proposition 5. *Let C be a rooted ℓ-chain reducible, leaf-addable, blob-determined class of directed binary phylogenetic networks. Then C is ℓ-chain reducible.*

Proof. To see that C is ℓ-chain reducible, observe that properties (i) and (ii) of ℓ-chain reducibility follow directly from Lemmas 5 and 6, while property (iii) of ℓ-chain reducibility is a consequence of C being leaf-addable. $\qquad\square$

5.2 Tree-child networks

In this section, we establish Theorem 6. Recall that a directed binary phylogenetic network N is tree-child if every non-leaf vertex has a child that is a tree vertex. Equivalently, N is tree-child if and only if N has no *stack reticulations*, two reticulations one of which is the parent of the other, and no *sibling reticulations*, two reticulations sharing a common parent (see [Sem16]). This equivalence will be used throughout this subsection.

Let N be a directed binary phylogenetic network. Since adding leaves to N cannot create any stack or sibling reticulations, it follows that the class of binary tree-child networks is leaf-addable. The next two lemmas show that this class is also blob-determined and rooted 3-chain reducible, and thus, by Proposition 5, 3-chain reducible.

Lemma 7. *The class of binary tree-child networks is blob-determined.*

Proof. Let N be a directed binary phylogenetic network. If N is tree-child, then N has no stack and no sibling reticulations, and so every directed binary phylogenetic network induced by a blob of N is tree-child.

Now suppose that all directed binary phylogenetic networks induced by a blob of N are tree-child. Then each such network has no stack and no sibling reticulations. Observe that every reticulation of N is contained in a blob of N and that both parents of a reticulation are in the same blob (because there are paths from the root to each parent). If N contains sibling reticulations, then (for similar reasons) their common parent must be in the same blob as each of the reticulations, and so the network induced by this blob would also contain sibling reticulations, a contradiction. Similarly, N cannot contain stacks. Thus N is tree-child. This completes the proof of the lemma. $\qquad\square$

By Lemma 7, the class of binary tree-child networks is blob-determined. Therefore, as explained in Section 4.2, in the process of deciding if an undirected binary phylogenetic network N has a tree-child orientation, we may assume that N has no non-trivial pendant phylogenetic subtrees. The analogous assumption holds for other classes.

Lemma 8. *The class of binary tree-child networks is rooted 3-chain reducible.*

Proof. Let N be an undirected binary phylogenetic network that can be tree-child rooted at a pendant edge $e_\rho = \{v_\rho, x_\rho\}$, where x_ρ is a leaf, and let N_d be a tree-child orientation of N rooted at e_ρ. Note that v_ρ is a tree vertex in N_d. Recall that we assume that N, and therefore N_d, has reticulation number at least 2, so $G(N_d)$ is well defined and has two sides s_1 and s_2 leaving v_ρ. We next construct a directed binary phylogenetic network N'_d from N_d as follows. First, for each side s of $G(N_d)$ that is neither s_1 nor s_2 and contains at least two leaves, delete all except one of the leaves of N_d on s and suppress the resulting vertices of in-degree one and out-degree one. At this stage of the construction, it is easily seen that the resulting directed binary phylogenetic network remains tree-child, as no stack and no sibling reticulations have been created. Continuing the construction, delete all leaves of N_d that are on either s_1 or s_2 of $G(N_d)$, and suppress the resulting vertices of in-degree one and out-degree one. This gives N'_d. Like the first part, the second part of the construction also preserves the property of being tree-child. To see this, observe that at most one of s_1 and s_2 has a reticulation as an end-vertex; otherwise, N has a reticulation-cut, contradicting Proposition 2. Hence, N'_d has no sibling reticulations. Moreover, as the root of N_d is not a reticulation, it follows that N'_d has no stack reticulations. Hence N'_d is tree-child.
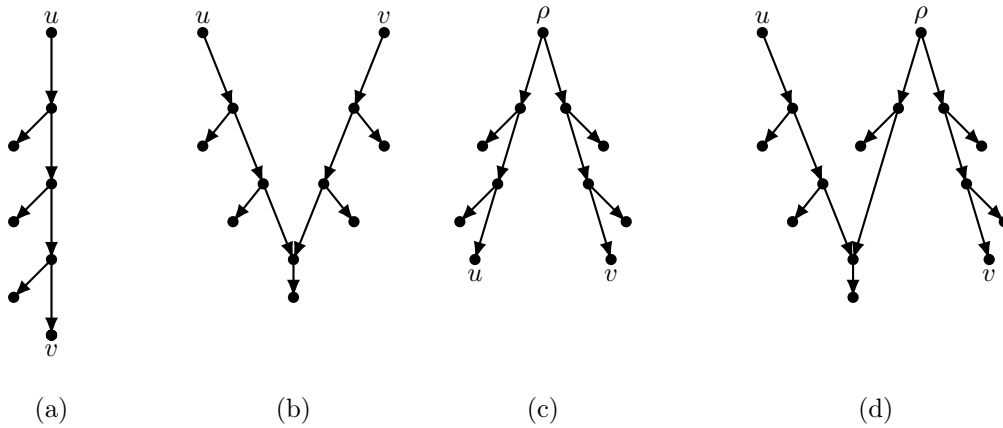
Figure 15: The (generic) correspondence of the undirected sides of the generator of an undirected binary phylogenetic network N to the directed sides of the generator of an orientation N' of N. If a side $\{u, v\}$ of $G(N)$ does not contain the root edge, this side corresponds to either (a) one side of $G(N')$ or (b) two sides of $G(N')$ separated by a reticulation with a leaf child. If the side $\{u, v\}$ of $G(N)$ does contain the root ρ, this side corresponds to either (c) the two sides of $G(N')$ incident with ρ, or to three sides of $G(N')$ as shown in (d). In this figure, all degree-one vertices are leaves, except the ones labeled u or v.

Let N' denote the underlying undirected binary phylogenetic network of N'_d, and let s_ρ be the side of $G(N')$ containing x_ρ. We next show that each side $s \neq s_\rho$ of $G(N')$ contains at most three leaves of N'. To see this, s corresponds to at most two sides of $G(N'_d)$; if s corresponds to exactly two sides, then these sides meet at a reticulation of N'_d with a leaf as a child (see Figure 15). Thus s contains at most three leaves of N'. On the other hand, the side s_ρ of $G(N')$ corresponds to at most three sides of $G(N'_d)$. Namely, s_1, s_2, and a third side s_3 if an internal vertex of s_ρ corresponds to a reticulation r of N'_d (see Figure 15). If a third side s_3 exists, then r is the parent of a leaf of N'_d. Sides s_1 and s_2 of $G(N'_d)$ contain no leaves of N'_d and, if it exists, s_3 contains at most one leaf of N'_d. In addition, the leaf x_ρ is on the side s_ρ of $G(N')$ and, if s_3 exists and r has a child that is a leaf, then this leaf is also on the side s_ρ of $G(N')$. Hence the side s_ρ also contains at most three leaves of N' in total, where x_ρ is either the first or the last leaf when the leaves of N' on s_ρ are ordered.

Let N^r be the undirected binary phylogenetic network obtained from N by applying a rooted 3-chain reduction with respect to e_ρ. Since $G(N^r) = G(N')$, it follows that N^r can be obtained from N' by adding leaves and, if necessary, relabelling leaves. Therefore, as N' can be tree-child rooted at e_ρ and the class of tree-child networks is leaf-addable, N^r can be tree-child rooted at e_ρ. Hence N is 3-chain reducible with respect to e_ρ. It now follows that the class of binary tree-child networks is 3-chain reducible. $\qquad\square$

By Lemmas 7 and 8, the algorithms of Section 4 are applicable to the class of binary tree-child networks. Thus, by Theorem 4, Theorem 6 holds provided $g(L, \ell)$ is a computable function. We end this subsection with the following lemma, which shows that this is indeed the case.

Lemma 9. *Let C be the class of binary tree-child networks, and let N be a directed binary phylogenetic network. Then deciding if N is in C takes $O(n)$ time, where n is the number of vertices in N.*

Proof. To check whether N is tree-child, we simply need to check that no reticulation is in a stack or in a pair of sibling reticulations. Since this only requires checking the (local) neighbourhood of each vertex, which is of size at most three as N is binary, this check can be executed in linear time. $\qquad\square$

6 Discussion

We have answered several foundational questions regarding the orientation of undirected phylogenetic networks. We have also shown that some of our results apply to partly-directed phylogenetic networks. Nevertheless, many interesting questions remain open.

Our results do not apply directly to some of the phylogenetic networks published in the biological literature. The reason for this is that these phylogenetic networks fall outside the framework of our definition. It would

be interesting to consider modifications of the definition given here that allow for the study of such networks from a mathematical point of view. For example, the phylogenetic network of grape cultivars in [MBO$^+$11, Fig. 3] contains several interesting complications. Firstly, it can be directly observed that any orientation of this phylogenetic network needs to have multiple roots (this can, for example, be concluded from the part of the network containing Muscat of Alexandria, Muscat Hamburg, and Trollinger). Secondly, as well as undirected and directed edges, the phylogenetic network contains dotted edges joining pairs of cultivars which are siblings or equivalent. Other examples include the phylogenetic network of bears in [KLB$^+$17, Fig. 4] and the phylogenetic network of the evolutionary history of Europeans in [Laz18, Fig. 1]. The first of these phylogenetic networks contains bidirected arcs, which we have not taken into account in this paper, while the second has dotted edges indicating that the direction is either unclear (corresponding to our undirected edges) or bidirectional.

More explicit (computational) questions are the following. Given an undirected binary phylogenetic network N, the problem of deciding if N has a tree-based orientation is NP-complete [HvIJ$^+$19, Appendix A]. Although we have shown that the analogous decision problems for the classes of binary tree-child and binary stack-free networks are fixed-parameter tractable with respect to the level of N, it remains open whether these problems are polynomial-time solvable. We expect both decision problems to be NP-complete, but have not found a proof. A related question concerns undirected nonbinary phylogenetic networks. It is common in the literature for directed nonbinary phylogenetic networks to have the restriction that each reticulation has exactly one outgoing arc. Calling such phylogenetic networks with this restriction *funneled*, an open question is whether one can decide in polynomial time if a given undirected nonbinary phylogenetic network has a funneled orientation. This is not always the case, as can be seen (with some effort) from the example shown in Figure 16.
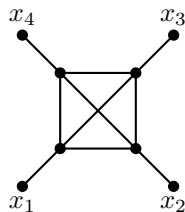

Figure 16: An undirected nonbinary phylogenetic network that has no funneled orientation.

Another question is whether our results generalize to directed phylogenetic networks with a root of out-degree 1 or out-degree greater than 2. This only makes sense when we are allowed to root an undirected phylogenetic network at an existing vertex, instead of at an edge as we have assumed in this paper. Note that, for directed binary phylogenetic networks that are blob determined, rooting at an edge is equivalent to adding a leaf to that edge, and rooting along the resulting pendant edge. Similarly, rooting at a vertex is equivalent to attaching a leaf to the vertex via a new edge, and rooting along this new edge. For these reasons, we expect our results to generalize. Finally, it would be interesting to find out whether the results in Section 5 generalize to partly-directed phylogenetic networks.

Acknowledgments

References

[ABR19]   Elizabeth S Allman, Hector Baños, and John A Rhodes. NANUQ: a method for inferring species networks from gene trees under the coalescent model. *Algorithms for Molecular Biology*, 14(1):1–25, 2019.

[AJMO16] Yuichi Asahiro, Jesper Jansson, Eiji Miyano, and Hirotaka Ono. Degree-constrained graph orientation: Maximum satisfaction and minimum violation. *Theory of Computing Systems*, 58(1):60–93, Jan 2016.

[AT92]   Noga Alon and Michael Tarsi. Colorings and orientations of graphs. *Combinatorica*, 12(2):125–134, 1992.

[Ata83]   Mikhail J Atallah. A graph orientation problem. *Purdue e-Pubs, Department of Computer Science Technical Reports*, 83-457, 1983.

[BM04]   David Bryant and Vincent Moulton. Neighbor-net: an agglomerative method for the construction of phylogenetic networks. *Molecular Biology and Evolution*, 21(2):255–265, 2004.

[ESS19]   Peter L Erdős, Charles Semple, and Mike Steel. A class of phylogenetic networks reconstructable from ancestral profiles. *Mathematical Biosciences*, 313:33–40, 2019.

[FHM18]   Andrew R Francis, Katharina T Huber, and Vincent Moulton. Tree-based unrooted phylogenetic networks. *Bulletin of Mathematical Biology*, 80(2):404–416, 2018.

[GHS17]   Philippe Gambette, Katharina T Huber, and Guillaume E Scholz. Uprooted phylogenetic networks. *Bulletin of Mathematical Biology*, 79(9):2022–2048, 2017.

[HKLS05]   Daniel H Huson, Tobias Klöpper, Pete J Lockhart, and Mike A Steel. Reconstruction of reticulate networks from gene trees. In *Annual International Conference on Research in Computational Molecular Biology*, pages 233–249. Springer, 2005.

[HRS10]   Daniel H Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic networks: concepts, algorithms and applications*. Cambridge University Press, 2010.

[HvIJ+19]   Katharina T Huber, Leo van Iersel, Remie Janssen, Mark Jones, Vincent Moulton, Yukihiro Murakami, and Charles Semple. Orienting undirected phylogenetic networks. *arXiv:1906.07430*, 2019.

[IKS+18]   Leo van Iersel, Steven Kelk, Georgios Stamoulis, Leen Stougie, and Olivier Boes. On unrooted and root-uncertain variants of several well-known phylogenetic network problems. *Algorithmica*, 80(11):2993–3022, 2018.

[Jan20]   Remie Janssen. Rooting networks, 2020. https://github.com/RemieJanssen/RootingNetworks.

[JJE+18]   Remie Janssen, Mark Jones, Péter L Erdős, Leo van Iersel, and Celine Scornavacca. Exploring the tiers of rooted phylogenetic network space using tail moves. *Bulletin of Mathematical Biology*, 80(8):2177–2208, 2018.

[JM21]   Remie Janssen and Yukihiro Murakami. On cherry-picking and network containment. *Theoretical Computer Science*, 856:121–150, 2021.

[KLB+17]   Vikas Kumar, Fritjof Lammers, Tobias Bidon, Markus Pfenninger, Lydia Kolter, Maria A. Nilsson, and Axel Janke. The evolutionary history of bears is characterized by gene flow across species. *Scientific Reports*, 7:46487, 2017.

[Laz18]   Iosif Lazaridis. The evolutionary history of human populations in europe. *Current Opinion in Genetics & Development*, 53:21–27, 2018.

[MBO+11]   Sean Myles, Adam R Boyko, Christopher L Owens, Patrick J Brown, Fabrizio Grassi, Mallikarjuna K Aradhya, Bernard Prins, Andy Reynolds, Jer-Ming Chia, Doreen Ware, et al. Genetic structure and domestication history of the grape. *Proceedings of the National Academy of Sciences*, 108(9):3530–3535, 2011.

[Mor05]   David A Morrison. Networks in phylogenetic analysis: new tools for population biology. *International journal for parasitology*, 35(5):567–582, 2005.

[Mor13]   David Morrison, March 20, 2013. http://phylonetworks.blogspot.com/2013/03/first-degree-relationships-and-partly.html.

[Mor15]   David Morrison, January 7, 2015. http://phylonetworks.blogspot.com/2015/01/complex-hybridizations-in-wheat.html.

[MSH+14]   Thomas Marcussen, Simen R Sandve, Lise Heier, Manuel Spannagl, Matthias Pfeifer, Kjetill S Jakobsen, Brande BH Wulff, Burkhard Steuernagel, Klaus FX Mayer, Odd-Arne Olsen, et al. Ancient hybridizations among the ancestral genomes of bread wheat. *Science*, 345(6194):1250092, 2014.

[MvIJ+19]   Yukihiro Murakami, Leo van Iersel, Remie Janssen, Mark Jones, and Vincent Moulton. Reconstructing tree-child networks from reticulate-edge-deleted subnetworks. *Bulletin of mathematical biology*, 81(10):3823–3863, 2019.

[Sem16]   Charles Semple. Phylogenetic networks with every embedded phylogenetic tree a base tree. *Bulletin of Mathematical Biology*, 78:132–137, 2016.

[SLA16]   Claudia Solís-Lemus and Cécile Ané. Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. *PLoS Genetics*, 12(3):e1005896, 2016.

[SM00]   Korbinian Strimmer and Vincent Moulton. Likelihood analysis of phylogenetic networks using directed graphical models. *Molecular Biology and Evolution*, 17(6):875–881, 2000.