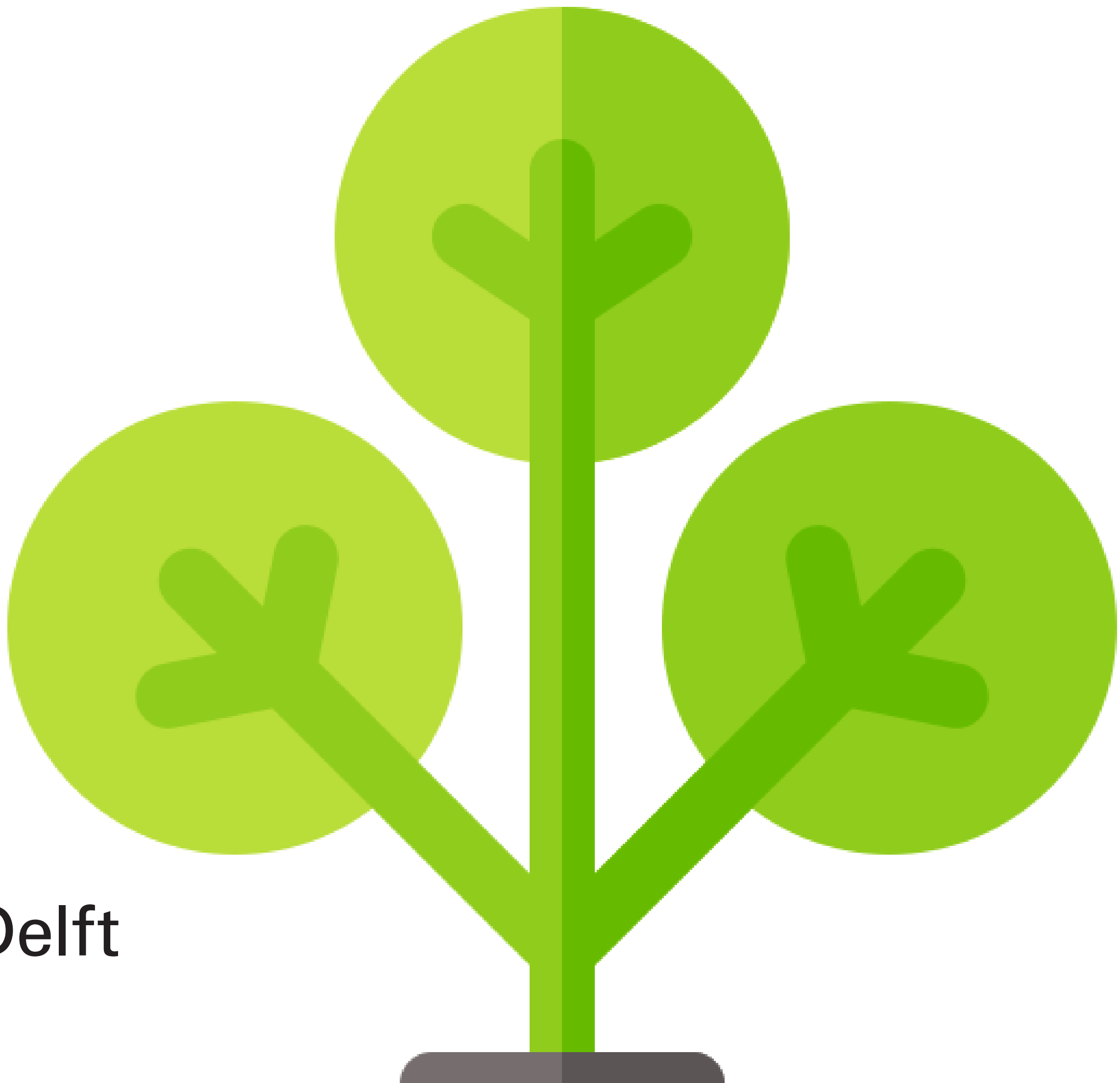
Encoding level-k phylogenetic networks

Frank Janisse



TU Delft

Encoding level-k phylogenetic networks

by

Frank Janisse

to obtain the degree of Master of Science in Applied Mathematics,
for the specialisation Optimization,
at the faculty of Electrical Engineering, Mathematics and Computer Science,
at the Delft University of Technology,
to be defended publicly on Tuesday August 17, 2021 at 10:00.

Student number:	4349571	
Project duration:	September 1, 2020 – August 17, 2021	
Thesis committee:	Dr. ir. L. J. J. van Iersel,	Delft University of Technology, chair
	Dr. M. E. L. Jones,	Delft University of Technology, daily supervisor
	Dr. ir. W. G. M. Groenevelt,	Delft University of Technology

TU Delft

Abstract

Phylogenetic networks are used to describe evolutionary histories and are a generalisation of evolutionary trees. They can contain so called *reticulations*, representing reticulate evolution, such as hybridization, lateral gene transfer and recombination. Methods are being developed to construct certain rooted phylogenetic networks from their subnetworks. A constructed network is *encoded* by their subnetworks if it is uniquely determined by that set. It has been shown that phylogenetic trees are encoded by their set of *triplets*, which are rooted trees on three species. However, triplets do not encode phylogenetic networks. Huber and Moulton introduced *trinets*, rooted networks on three species, which do encode *level-1* phylogenetic networks, which are networks containing at most one reticulation in each biconnected component. Van Iersel and Moulton proved that level-2 phylogenetic networks are encoded by their set of trinets and Nipius proved that level-3 phylogenetic networks are encoded by their set of quarnets, which are rooted networks on four species.

In this thesis we prove that for all $k \geq 2$, level-k networks without symmetry in their biconnected components are encoded by their set of $(k+1)$-nets, which are rooted networks on $k+1$ leaves. This result provides some evidence for the conjecture that *all* level-k phylogenetic networks are encoded by their set of $(k+1)$-nets. Thereafter, we generalise encoding results for level-2 and level-3 networks, where the underlying structure, called *generator*, and its *sides* play an important role. A *generator* is a directed acyclic biconnected multigraph, containing only vertices with indegree 2 and outdegree at most 1, indegree 1 and outdegree 2, and indegree 0 and outdegree 2. The *sides* of a generator are the arcs and outdegree-0 vertices of a generator. We have not been able to prove that level-k networks with symmetry in the generators of their biconnected components are in general encoded by $k+1$-nets. For the networks with symmetry, we prove encoding results for networks with leaves on at most p sides of the underlying generators of their biconnected components. We further prove that level-4 networks are encoded by 6-nets.

Although Nipius gave a counterexample showing that not all (level-3) phylogenetic networks are encoded by their set of trinets, it is useful to know which networks *are* encoded by their set of trinets or k-nets. In this thesis, we provide an algorithm which can serve as tool for proving that certain level-k networks are encoded by their set of k-nets. Our presented algorithm is a first step to generalise encoding results to level-k networks with $k \geq 4$ by using trinets. Furthermore, we are a step closer to proving the conjecture that all level-k networks are encoded by $(k+1)$-nets, including networks with symmetry in the generators of their biconnected components.

Preface

During my education, I have been fascinated by discrete mathematics and graph theory. In the research field of phylogenetics, many questions are unanswered. Last year, I spent my time on trying to solve a few of the open problems to complete my master Applied Mathematics at the Delft University of Technology. The project was carried out under the supervision of dr. ir. Leo van Iersel and dr. Mark Jones. I did theoretical research on constructing phylogenetic networks from smaller subnets. I proved encoding results for networks with a different number of reticulations, where I mainly focused on generalising existing results to a higher number of reticulations.

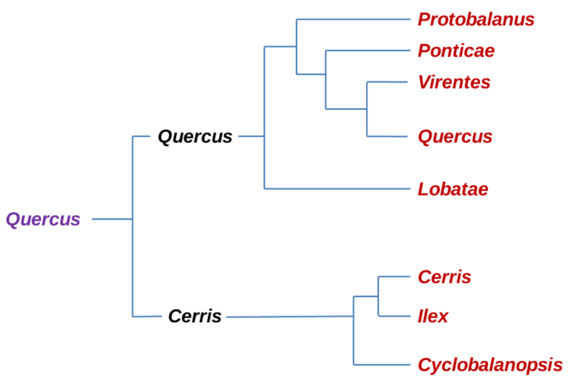
I would like to thank Leo van Iersel and Mark Jones for the supervision during the year. Their ideas, feedback and enthusiasm is very appreciated. I also want to thank Leonie Boortman for organising weekly meetings with other graduation students, in which we helped each other with studying during the pandemic. Lastly, I want to thank Wolter Groenevelt for being part of the thesis committee and my family and friends for the support.

Frank Janisse
Delft, August 2021

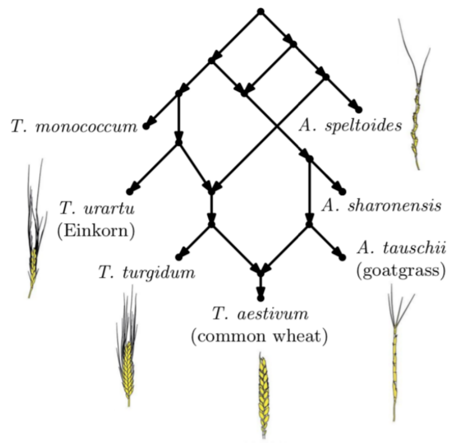
Contents

1 Introduction

Phylogenetics is the biological study of the evolutionary history of a set of species or taxa. A famous example is the Tree of Life from Darwin, 1859, a model where he describes evolution and the relationships between organisms. Phylogenetic *trees* are traditionally used, but they are limited in the sense that they are unable to represent complex evolutionary events such as hybridization, horizontal gene transfer and recombination (Mallet et al., 2016, Soucy et al., 2015), called reticulation events. Trees can represent speciation events, the formation of new, distinct species form one species, but phylogenetic *networks* are used to give a more complete representation of the history (Bapteste et al., 2013) because they can represent the formation of a new species from multiple parent species, too. In graph theoretic terms, a *rooted phylogenetic network* is a directed acyclic graph that has a single root, no indegree-1 outdegree-1 vertices, and has its leaves bijectively labelled by the elements of a set of species X (Huson et al., 2010). In contrary to a phylogenetic tree, a network can contain *reticulations*, vertices with indegree greater than one. In Figure 1.1a, an example of a phylogenetic tree is given; in Figure 1.1b, an example of a phylogenetic network is given.



(a) An example of a phylogenetic tree, representing the phylogeny of the major clades of oaks (Mcvay et al., 2017).



(b) A phylogenetic network for wheat species (Marcussen et al., 2014). This network contains three reticulations.

Figure 1.1: Examples of a phylogenetic tree and a phylogenetic network. The network contains reticulations and the tree does not. The directions of the arcs are indicated by the arrows.

Phylogenetic trees and their properties have been studied intensively since the 1970s, but the interest in phylogenetic networks has grown more recently. It is of interest how to construct a 'complete' phylogenetic network, displaying all relations between species from a certain genus, family, order, et cetera, from biological data sets (Bapteste et al., 2013), such as DNA data. PhyloNet (Than et al., 2008), PADRE (Lott et al., 2009), TripNet (Poormohammadi et al., 2014), and Dendroscope 3 (Huson and Scornavacca, 2012) are examples of algorithms for constructing phylogenetic networks, including reticulations.

An important result is that a phylogenetic tree is *encoded* by its triplets (see e.g. Dress et al., 2011), where *triplets* are rooted trees on three species. That is, each phylogenetic tree T is the unique phylogenetic tree that contains the set of triplets that can be obtained by restricting T to three of its leaves. The uniqueness is important, because it ensures that T is the only tree that represents the correct evolutionary history. Ranwez et al., 2007 and Scornavacca et al., 2008 present algorithms to construct a phylogenetic tree from its set of triplets. The set of triplets of a phylogenetic network does not necessarily encode the network (Gambette and Huber, 2011). There are examples of algorithms that construct a phylogenetic network, given a set of triplets (Byrka et al., 2010), but the disadvantage is that it is not sure that the constructed phylogenetic network represents the evolutionary history correctly.

In this thesis, we focus on *binary level-k* networks. A phylogenetic network is *binary* if all vertices have indegree and outdegree at most two and a total degree of at least three, and a binary phylogenetic network is *level-k* if each biconnected component has at most k reticulations. TriLoNet (Oldman et al., 2016) is an algorithm that constructs level-1 networks from smaller level-1 networks on three taxa, called trinets. TriL2Net (Kole, 2020) constructs a level-2 phylogenetic network from a set of level-2 trinets. To make sure that such algorithms give the phylogenetic network that represents the evolutionary history correctly, it is of interest to know whether phylogenetic networks are encoded by their set of trinets or other subnets. In Huber and Moulton, 2013 it is proved that level-1 phylogenetic networks are encoded by their trinets. In Van Iersel and Moulton, 2012, it is proved that level-2 phylogenetic networks are encoded by trinets and in Nipius, 2020 it is shown that level-3 phylogenetic networks are *not* encoded by trinets, but *are* encoded by quarnets, where a quarnet is a rooted phylogenetic network on four leaves. In Nipius, 2020, it is also proved that most, but not all level-3 networks are weakly encoded by trinets. It is useful to know which level-k networks *are* encoded by trinets or k-nets.

We see that level-k phylogenetic networks are encoded by their $(k+1)$-nets for $k = 2$ and $k = 3$. We conjecture that this holds for all $k \geq 2$. If we are able to prove this, algorithms can be used to determine the original network from subnetworks for every number of reticulations in its biconnected components, although the more complex the networks are, the larger the subnetworks would need to be. In this way, it is useful to know which other networks are encoded by their subnets, where we want the subnets to be as small as possible. The smaller the subnets are, the less knowledge is needed to determine the phylogenetic network that gives the complete representation of the evolutionary history. The aim of this thesis is to prove as strong as possible encoding results for level-k networks in general.

Important concepts in this thesis to classify rooted phylogenetic networks are biconnected components, generators and symmetry. A *simple* phylogenetic network contains only one *nontrivial biconnected component*, which is a maximal biconnected subgraph that is not an arc for which its deletion results in a disconnected graph. A *generator* is a directed acyclic biconnected multigraph, containing only vertices having either indegree 2 and outdegree at most 1, indegree 1 and outdegree 2, indegree 0 and outdegree 2. The *underlying generator* of a simple network is the graph obtained from the network by deleting all its leaves and suppressing indegree-1 outdegree-1 vertices. In this thesis, we will prove for level-k networks that have no symmetry in the underlying generators of all biconnected components that they are encoded by $(k+1)$-nets. A generator has *symmetry* if there exists a graph automorphism on its vertices such that at least one vertex is not mapped to itself, or the generator contains a pair of parallel arcs. For networks with such symmetry, we will prove different encoding results if a network has leaves on at most p sides of the underlying generators of restrictions to its biconnected components. *Sides* are the arcs and the outdegree-0 vertices of a generator. The advantage of these results is that they hold for all networks, with and without the defined symmetry. We will also prove a more concrete result, namely that level-4 networks are encoded by 6-nets. We conjecture that level-4 networks are even encoded by 5-nets. Lastly, after providing an analysis of the proof in Nipius, 2020 that most level-3 networks are weakly encoded by trinets, we provide an algorithm that may be useful to prove for certain level-k networks that they are encoded by k-nets.

The structure of the thesis is as follows. First, we will formalize the necessary concepts in Chapter 2. This includes several properties and classifications of rooted phylogenetic networks, that we use to prove encoding results. In Chapter 3, we will generalise results from Nipius, 2020 and Van Iersel and Moulton, 2012. It turns out that we can reduce all our problems to simple networks. In Chapter 4, we will prove our first encoding result. It states that all level-k networks without symmetry in the generators of its biconnected components are encoded by $(k+1)$-nets. We prove in Chapter 5 encoding results for networks with leaves on a bounded number of sides of the generators of its biconnected components. In Chapter 6, we prove that all level-4 networks are encoded by 6-nets. In Chapter 7, we analyse how it is proved in Nipius, 2020 that most level-3 networks are encoded by trinets. We give an algorithm that can be used to automate the most laborious part of their proof. Thereafter, we generalise this algorithm, such that it can be used to attempt to prove that certain level-k networks are encoded by k-nets. Finally, we give a conclusion and further research directions in Chapter 8.

2

Preliminaries

We will formalize the concepts that are used in this thesis, give some properties and make some assumptions that hold for the whole thesis. Most definitions are the same as in Van Iersel and Moulton, 2012. The definitions will be useful to decompose phylogenetic networks, determine their subnets and observe the symmetries.

2.1. Phylogenetic networks

A rooted phylogenetic network is defined as follows.

Definition 2.1. For a set of species X, a *rooted phylogenetic network* $N = (V, E)$ on X is a directed acyclic graph with a single indegree-0 vertex, in which its outdegree-0 vertices are bijectively labelled by the elements of X and each vertex has either

- indegree 0 and outdegree at least 2,
- indegree 1 and outdegree at least 2,
- indegree at least 2 and outdegree 1,
- indegree 1 and outdegree 0.

Furthermore, duplicate arcs are not allowed in $E(N)$. The unique indegree-0 vertex is called the *root* and the outdegree-0 vertices are called the *leaves* of N. For an arc (u, v) we say that u is the *tail vertex* and v is the *head vertex*.

Since the leaves are bijectively labelled by the elements of X, we can identify each leaf with its label. We assume for the whole thesis that a phylogenetic network has finitely many vertices and that X is a finite set. We refer in this thesis to a rooted phylogenetic network as a *phylogenetic network* or *network*. In this thesis, the arcs of all networks in all figures are directed from top to bottom. An example of a phylogenetic network can be found in Figure 2.1. The following definition shows the difference between a phylogenetic network and a phylogenetic tree.

Definition 2.2. A *rooted phylogenetic tree* is a rooted phylogenetic network that does not contain vertices with indegree at least 2.

The vertices that are allowed in a phylogenetic network, but not in a phylogenetic tree are called reticulations as in the definition below.

Definition 2.3. In a phylogenetic network, vertices with indegree at least 2 are called *reticulations* or *reticulation vertices*. Vertices with indegree-1 and outdegree at least 2 are called *tree vertices*.

The reticulations in Figure 2.1 are indicated by squares. In this thesis, we will only consider binary phylogenetic networks which are defined as follows.

Definition 2.4. A phylogenetic network is called *binary* if all vertices have indegree and outdegree at most two.

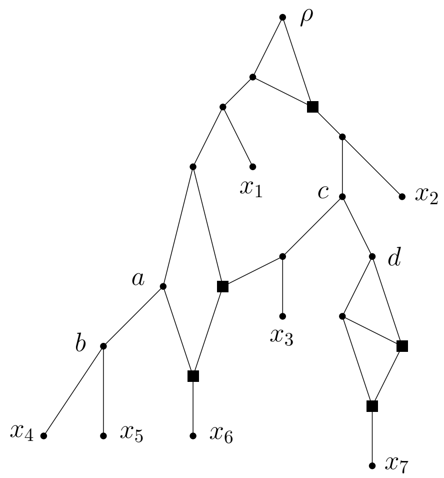


Figure 2.1: Example of a binary rooted phylogenetic network on $X = \{x_1, \ldots, x_7\}$. The root is denoted by ρ and the reticulation vertices are indicated by squares.

2.2. Recoverable networks

In this section, we will define when a phylogenetic network is recoverable. We will first give some other useful definitions.

Definition 2.5. Let $N = (V, E)$ be a phylogenetic network on X and let $u, v \in V(N)$. If $(u, v) \in E(N)$, then u is a *parent* of v and v is a *child* of u. Furthermore, if $u \notin X$, then we say that v is *below* u if $u = v$ or if there exists a directed path from u to v in N. If $u \in X$, then v is *below* u if v is below the parent of u. Lastly, let $e = (u, v) \in E(N)$ and $w \in V(N)$. Then, w is *below* e if w is below v.

Definition 2.6. Let $D = (V, E)$ be a directed graph with and let $D' = (V', E')$ be the underlying undirected graph of D with $V = V'$ and D' connected. A vertex $v \in V$ is called a *cut-vertex* if $D' - v$ is disconnected. Similarly, an arc $a = (u, v) \in E$ is a *cut-arc* if $D' - a'$ is disconnected, with $a' = \{u, v\} \in E'$.

We will use cut-vertices to define a biconnected graph and a biconnected component.

Definition 2.7. A directed graph is *biconnected* if it has no cut-vertices. Furthermore, a *biconnected component* is a maximal biconnected subgraph (i.e. a biconnected subgraph that is not contained in any other biconnected subgraph).

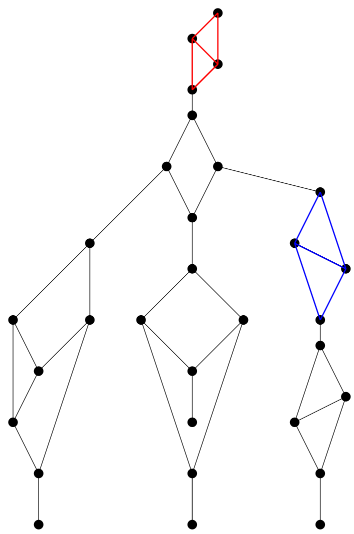
According to this definition, each cut-arc is a biconnected component. We will call such a cut-arc a *trivial biconnected component*. The network in Figure 2.1 contains two nontrivial biconnected components and nine trivial biconnected components. Arcs (a, b) and (c, d) are for example cut-arcs, and each cut-arc is a trivial biconnected component. In this thesis, we will make use of the biconnected components of a network to get results for the whole network. So considering these components makes a lot of problems easier.

In Figure 2.2b each node of the displayed tree represents a nontrivial biconnected component of the network in Figure 2.2a. Trivial biconnected components are represented by edges of the tree, except for ingoing arcs of leaves. We will now define the frequently used definition of a level-k network.

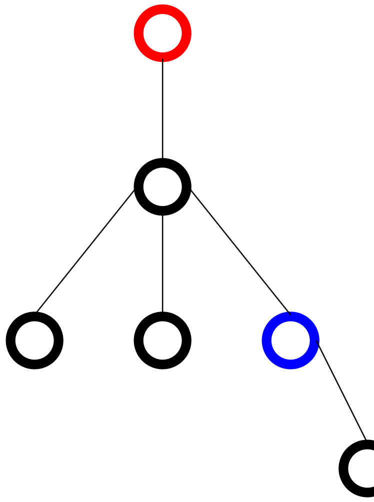
Definition 2.8. A (binary) phylogenetic network is *level-k* if each biconnected component has at most k reticulations.

For example, the networks in Figure 2.1 and 2.2a are both level-3. The following definition defines some special cases of biconnected components.

Definition 2.9. Let N be a network and B be a nontrivial biconnected component. B is *redundant* if it has only one outgoing arc. Furthermore, B is *strongly redundant* if it has only one outgoing arc $a = (u, v)$ and all leaves of N are below v.



(a) Network N with a strongly redundant biconnected component.



(b) A tree where each node represents a biconnected component of N. The red and blue nodes represent the red and blue biconnected components of N. The edges represent cut-arcs of N, not ending in a leaf.

Figure 2.2: A phylogenetic network, with the network displaying its nontrivial biconnected components.

The red arcs in Figure 2.2a form a strongly redundant biconnected component and the red circle in Figure 2.2b represents it. The blue arcs form a redundant biconnected component and the blue circle in Figure 2.2b represents this component. Finally, we can give the definition of a recoverable network.

Definition 2.10. A phylogenetic network is *recoverable* if it has no strongly redundant biconnected components.

The network in Figure 2.1 is recoverable, but the network in Figure 2.2a is not recoverable, since the red arcs form a strongly redundant biconnected component. The importance of this concept will become clear in the next sections.

2.3. Encoding networks with subnets

In this section we will define what it means for a network to be 'determined' or 'encoded' by its subnets. We begin with the definition of a lowest stable ancestor.

Definition 2.11. Let $N = (V, E)$ be a network and let $V' \subseteq V$. A *lowest stable ancestor* $LSA(V')$ is a vertex $w \in V \setminus V'$ for which all paths from the root to any $v \in V'$ pass through w, such that there is no vertex below w for which this property holds. Observe that the lowest stable ancestor is unique.

Observation 1. *Let N be a network on X and let $X' \subseteq X$ with $|X'| \geq 2$. Then there exist two leaves $x_i, x_j \in X'$ for which $LSA(x_i, x_j) = LSA(X')$.*

Proof. First, we will prove the following claim.

Claim: For any $S_1, S_2 \subseteq X'$ such that $S_1 \cap S_2 \neq \emptyset$, $LSA(S_1 \cup S_2)$ equals either $LSA(S_1)$ or $LSA(S_2)$.
Proof: Let $x_1 \in S_1 \cap S_2$. Then, either $LSA(S_1)$ is below $LSA(S_2)$ or $LSA(S_2)$ is below $LSA(S_1)$, otherwise there is a path from the root of N to x_1 that does not pass through $LSA(S_1)$ or does not pass through $LSA(S_2)$. Suppose that $LSA(S_2)$ is below $LSA(S_1)$. Then any path from the root to a leaf in S_2 must pass through $LSA(S_1)$. If not, then there exists a path from the root to x_1 that passes through $LSA(S_2)$, but does not pass through $LSA(S_1)$. This contradicts with the definition, that is, any path from the root of N to a leaf in S_1 passes through $LSA(S_1)$. In the same way, if $LSA(S_1)$ is below $LSA(S_2)$, then any path from the root to a leaf in S_1 passes through $LSA(S_2)$. Then by definition of lowest stable ancestor, $LSA(S_1 \cup S_2)$ equals either $LSA(S_1)$ or $LSA(S_2)$. □

Let $X' = \{x_1, \ldots, x_n\}$. We will prove now by induction on n that $LSA(X') = LSA(\{x_i, x_j\})$ for some $i, j \in \{1, \ldots, n\}$, $i \neq j$. First, suppose that $n = 2$. Then $X' = \{x_1, x_2\}$ and $LSA(X') = LSA(\{x_1, x_2\})$, so the statement holds for $n = 2$. Now, assume that the statement holds for $n = k \geq 2$. That is, if $X' = \{x_1, \ldots, x_k\}$, then $LSA(X') = LSA(\{x_i, x_j\})$ for some $i, j \in \{1, \ldots, k\}$, $i \neq j$. Now, suppose that $X' = \{x_1, \ldots, x_{k+1}\}$ and $S_1 = \{x_1, \ldots, x_k\}$ and $S_2 = \{x_k, x_{k+1}\}$. Note that $X' = S_1 \cup S_2$ and $S_1 \cap S_2 \neq \emptyset$. By the claim, $LSA(S_1 \cup S_2)$ equals either $LSA(\{x_1, \ldots, x_k\})$ or $LSA(\{x_k, x_{k+1}\})$. If $LSA(S_1 \cup S_2) = LSA(\{x_k, x_{k+1}\})$, then $LSA(X') = LSA(\{x_k, x_{k+1}\})$ and we are done. If $LSA(S_1 \cup S_2) = LSA(\{x_1, \ldots, x_k\})$, then by the induction hypothesis $LSA(X') = LSA(S_1 \cup S_2) = LSA(\{x_i, x_j\})$ for some $i, j \in \{1, \ldots, k\}$, $i \neq j$. By induction, it holds for all $n \geq 2$ that if $X' = \{x_1, \ldots, x_n\}$, then $LSA(X') = LSA(\{x_i, x_j\})$ for some $i, j \in \{1, \ldots, n\}$, $i \neq j$. □

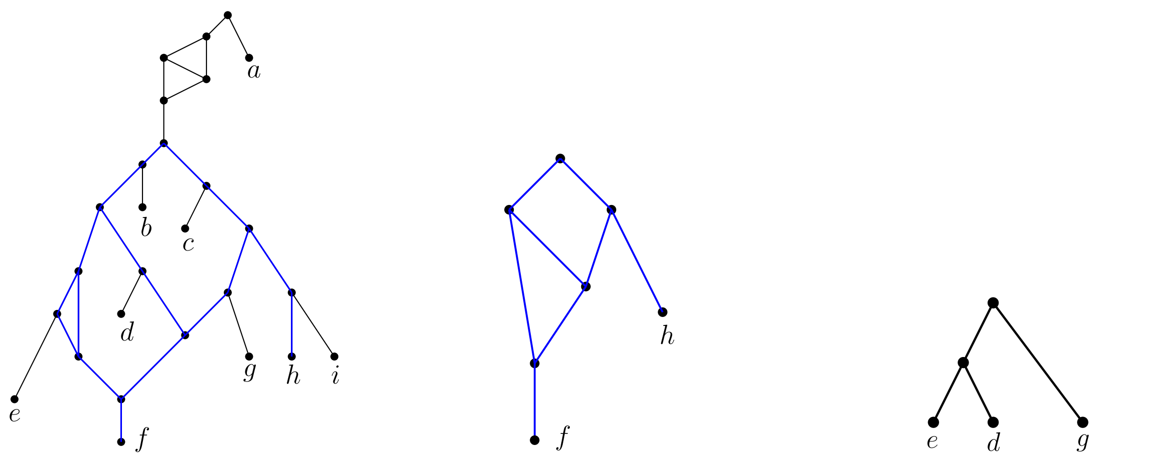
In the network in Figure 2.1, we see for example that $LSA(x_3, x_7) = c$ and $LSA(x_4, x_7) = \rho$. In this thesis we look at subnets of networks where the LSA and the leaves play an important role.

Definition 2.12. A k-net is a binary, rooted phylogenetic network on k leaves.

Since we often use 2-nets and 3-nets, we call them *binets* and *trinets*, respectively. The k-nets are obtained from a network in the following way.

Definition 2.13. Let N be a phylogenetic network on X and let $\{x_1, \ldots, x_p\} \subseteq X$. The p-net on $\{x_1, \ldots, x_p\}$ *exhibited* by N is the p-net obtained from N by deleting all vertices that are not contained in any path from $LSA(\{x_1, \ldots, x_p\})$ to $x_1, \ldots, x_p$ and subsequently suppressing all indegree-1 outdegree-1 vertices and parallel arcs until neither is possible.

Note that a k-net is a rooted phylogenetic network, so it cannot contain indegree-1 outdegree-1 vertices and parallel arcs. Suppressing parallel arcs means replacing by a single arc. An example of a phylogenetic network N on $X = \{a, b, c, d, e, f, g, h, i\}$ is given in Figure 2.3a. The blue arcs represent all paths from $LSA(\{f, h\})$ to f and h. Then, the binet on $\{f, h\}$ exhibited by N is the network in Figure 2.3b. The trinet on $\{d, e, g\}$ exhibited by N is given in Figure 2.3c.



(a) Phylogenetic network N on X. (b) The binet on $\{f, h\}$ exhibited by N. (c) The trinet on $\{d, e, g\}$ exhibited by N.

Figure 2.3: A phylogenetic network with one of its binets and one of its trinets.

We will denote the set of all trinets of a phylogenetic network N by $Tn(N)$ and denote the set of all p-nets of a phylogenetic network N with $S_p(N)$. In Definition 2.16, this notation is used, together with the definition of equal networks. We will first define a graph isomorphism for directed graphs.

Definition 2.14. Let G_1 and G_2 be two directed graphs. A *graph isomorphism* between G_1 and G_2 is a bijection $f : V(G_1) \to V(G_2)$ such that for any $u, v \in V(G_1)$ it holds that (u, v) is an arc of G_1 if and only if $(f(u), f(v))$ is an arc of G_2. Moreover, G_1 and G_2 are *isomorphic* graphs if there exists such a bijection between G_1 and G_2.

Definition 2.15. Let N and N' be two phylogenetic networks on X. N and N' are *equal* networks (or: $N = N'$) if there exists a graph isomorphism f between N and N' such that $f(x) = x$ for each leaf $x \in X$.

Finally, we can define when a network is determined (encoded) by its set of k-nets. We distinguish between weakly and strongly encoded.

Definition 2.16. For $k \geq 2$, a phylogenetic network N is *strongly encoded* or *encoded* by its set of k-nets $\mathcal{S}_k(N)$ if there does not exist a recoverable phylogenetic network N' with $N' \neq N$ such that $\mathcal{S}_k(N) = \mathcal{S}_k(N')$.

Definition 2.17. For $k \geq 2$, a class $\mathcal{C}$ of phylogenetic networks is *weakly encoded* by k-nets if there are no two recoverable networks N, N' in $\mathcal{C}$ with $N \neq N'$ such that $\mathcal{S}_k(N) = \mathcal{S}_k(N')$.

Note that if N' is not recoverable, there is a strongly redundant biconnected component (as in Figure 2.2a), but it is suppressed by constructing any k-net. Therefore we cannot know what this biconnected component is. Therefore, we restrict to recoverable networks in Definition 2.16 and 2.17.

2.4. Generators and symmetry

In many proofs in this thesis, the generator of a phylogenetic network is used. We use that networks can be divided into groups of networks with the same underlying structure. We will call this kind of structure the generator. The symmetries in such a generator turn out to be important. Before we define it formally, we will define what a simple phylogenetic network is. Note the definition of simple in this thesis is different from the common definition of a simple graph.

Definition 2.18. A phylogenetic network is *simple* if the head of each cut-arc is a leaf.

Definition 2.19. A level-k phylogenetic network is a *simple level-k* network if it is simple.

We will now define a level-k generator. Thereafter, we define how to obtain a generator from a network.

Definition 2.20. A level-k generator is a directed acyclic biconnected multigraph with exactly k reticulations. Furthermore, each vertex has either

- indegree 2 and outdegree at most 1,
- indegree 1 and outdegree 2,
- indegree 0 and outdegree 2 (the single root).

Note that we defined a generator as a binary graph. Also, it can contain pairs of parallel arcs. The example of a level-4 generator in Figure 2.4b has one pair of parallel arcs.

Definition 2.21. Let G be a generator. The *sides* of G are the arcs and outdegree-0 vertices of G.

Definition 2.22. Let G be a generator and let A_1 and A_2 be two sides of G that form a pair of parallel arcs. Then, A_1 is the *parallel arc* of A_2, and vice versa.

Each side will be labelled by a capital letter. In Figure 2.4b all sides of this generator are labelled and C_1 is a parallel arc of C_2. Sides L and M are outdegree-0 vertices.

Definition 2.23. Let N be a simple phylogenetic network. The graph obtained from N by deleting all leaves and suppressing all indegree-1 outdegree-1 vertices is called the *underlying generator* of N and is denoted by G_N.

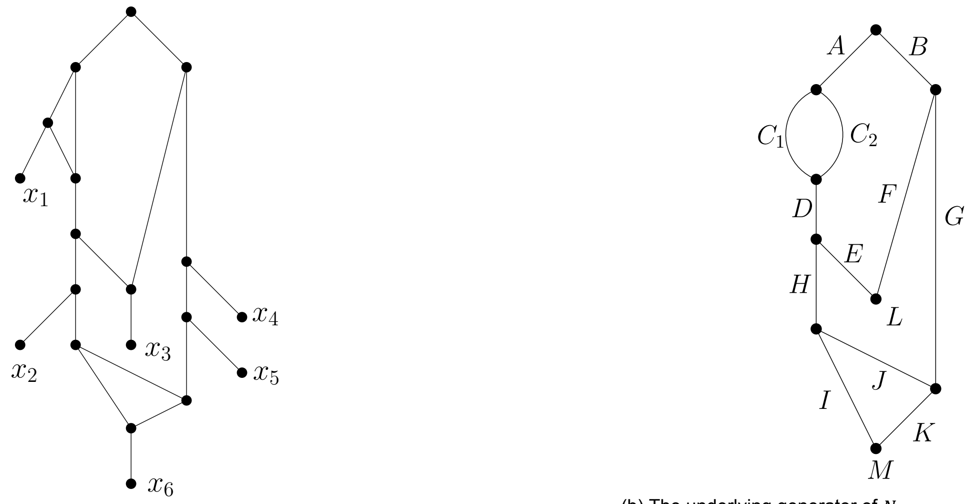
N can be reconstructed from G_N in the following way.

- Replace each arc of G_N by a directed path with $l \geq 0$ internal vertices $v_1, \ldots, v_l$ and, for each such internal vertex v_i, add a leaf $x_i \in X$ and an arc (v_i, x_i);
- for each indegree-2 outdegree-0 vertex v, add a leaf $x \in X$ and an arc (v, x).

It is defined below what it means for a leaf of a network to be on a certain side of the underlying generator.

Definition 2.24. Let N be a simple level-k network on X, let G_N be its underlying generator and let S be a side of G_N. Leaf $x \in X$ *is on side* S if it is 'hung' on side S in this construction of N from G_N. More precisely, x is on side S if

- S is an outdegree-0 vertex in $V(G_N)$, s is the corresponding vertex in $V(N)$ and (s,x) is an arc in $E(N)$,
- or S is an edge $(u,v) \in E(G_N)$ and the parent of x in N is one of the internal nodes of the directed path that is the replacement of (u,v) in the construction of N from G_N.



(a) A simple level-4 network N on X.

(b) The underlying generator of N.

Figure 2.4: Example of a network with its underlying generator.

In Figure 2.4, leaves $x_1, x_2, x_3, x_4, x_5, x_6$ are on sides C_1, H, L, G, G, M of G_N, respectively. G_N is the underlying generator of N, and we see that the sides are labelled in G_N, the leaves are deleted from N to obtain G_N, and all indegree-1 outdegree-1 vertices are suppressed.

In Chapter 3 we will see that for a simple level-k network N on X, there are p-nets exhibited by N, with $p \leq k$, with the same underlying generator as N. This is a very useful result that we prove and use in this thesis. Therefore, we will first define the crucial sides of a generator.

Definition 2.25. Let N be a network on X and G_N its underlying generator. A set of sides of G_N is a *set of crucial sides* if it contains all indegree-2 outdegree-0 vertices together with at least one arc of each pair of parallel arcs. A side is *crucial* if it is contained in a set of crucial sides and *noncrucial* if it is not contained in any set of crucial sides. For a leaf $x \in X$ on side S, x is a *crucial leaf* if S is crucial; x is a *noncrucial leaf* if S is noncrucial.

For example, $\{C_1, L, M\}$ and $\{C_2, L, M\}$ are all sets of crucial sides of G_N in Figure 2.4 and x_1, x_3 and x_6 are crucial leaves. We will now give the definition of a crucial k-net.

Definition 2.26. Let N be a simple level-k network on X and G_N its underlying generator. Let P be a p-net on $X' \subseteq X$ exhibited by N for $p \geq 2$. P is a *crucial p-net* of N if X' contains at least one leaf on each side in some set of crucial sides of G_N.

In Chapter 3 we will see that if $p = k$, then there always exists a crucial p-net. But for a network with a set of crucial sides of size exactly k, it has no crucial p-nets for $p < k$. Furthermore, we will see that a crucial p-net of N has the same underlying generator as N.

We will now concentrate on symmetries in a generator. Having parallel arcs leads for example to a symmetry in a generator. When a generator has symmetry is formally defined below.

Definition 2.27. A generator *has symmetry* if

- there exists a graph automorphism $f : V(G) \to V(G)$ such that $f(v) \neq v$ for at least one $v \in V(G)$, where a *graph automorphism* is a graph isomorphism from a graph to itself,
- or it contains a pair of parallel arcs.

Symmetry in a generator leads to the existence of a relabelling of sides of the generator such that an isomorphic generator is obtained. A relabelling of sides is defined below.

Definition 2.28. Let G be a generator with symmetry and let Z be the set of sides of G. A *relabelling of sides* is a bijective function $f : Z \to Z$ such that $f(s) \neq s$ for some $s \in Z$.

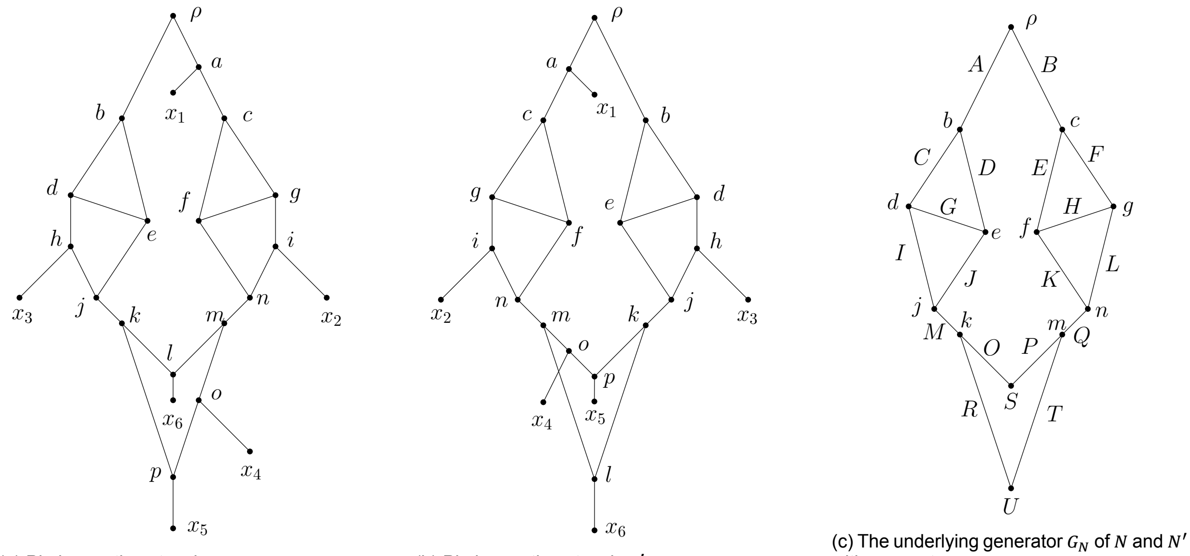
Definition 2.29. Let G be a generator with set of sides Z and let $\{S_1, S_2\} \subseteq Z$ be a set of sides of G which form a pair of parallel arcs. S_1 and S_2 are *switched* if a relabelling of sides f is applied to Z such that $f(S_1) = S_2$, $f(S_2) = S_1$ and $f(s) = s$ for all $s \in Z \setminus \{S_1, S_2\}$.

Definition 2.30. Let N be a network on X and G_N its underlying generator with symmetry. Let $f : V(G_N) \to V(G_N)$ be a graph automorphism such that $f(x) \neq x$ for some $x \in V(G_N)$ giving an isomorphic generator. The relabelling of sides *belonging to* f is the automorphism $f' : Z \to Z$ such that $f(x) = f'(x)$ for all x in the set of outdegree-0 vertices of G_N and $f'(y) = (f(u), f(v))$ for all sides $y \in Z$ that are arcs $(u, v) \in E(G_N)$ for some $u, v \in V(G_N)$.

The network N on X in Figure 2.5a equals the network N' in Figure 2.5b, so $N = N'$. N and N' have the same underlying generator G_N, presented in Figure 2.5c. Note that in the construction of G_N from N, all leaves of X are deleted and vertices a, i, h and o are suppressed. Furthermore, leaves l and p in N and N' are sides in G_N and therefore denoted by capital letters S and U. G_N has symmetry; we will give examples of a graph automorphisms and a relabellings of sides giving an isomorphic generator to G_N.

Let $f : V(G_N) \to V(G_N)$ be the graph automorphism such that $f(S) = U$, $f(U) = S$ and $f(v) = v$ for all $v \in V(G_N) \setminus \{S, U\}$. Then, the relabelling of sides of G_N belonging to f is $f' : Z \to Z$ such that f' maps sides in the following way: $O \Leftrightarrow P; R \Leftrightarrow T; S \Leftrightarrow U$ and $f'(s) = s \; \forall s \in Z \setminus \{O, P, R, S, T, U\}$. This symmetry leads to the possibility for leaves x_5 and x_6 to be on side T and R in N, but on side R and T in N', respectively, while $N = N'$.

Consider leaf x_4 in N. It is on side T in N and on side O in N', while $N = N'$. This can be explained by the existence of another graph isomorphism besides f, $g \neq f$, that gives an isomorphic generator to G_N, too. g maps the vertices of G_N in the following way: $c \Leftrightarrow b; g \Leftrightarrow d; f \Leftrightarrow e; n \Leftrightarrow j; m \Leftrightarrow k$ and ρ, S and U are mapped to itself by g. This symmetry causes for example that x_1 is on side B in N and on side A in N'. The relabelling of sides belonging to the composition of f and g maps T to O. Therefore there exists equal networks with leaf x_4 on the different mentioned sides. In another way, if only network N and G_N are considered, we say that for an simple network N' on the same leaf set with a generator isomorphic to G_N, leaf x_4 is on side O, P, R or T in N', due to the symmetries of G_N.



(a) Phylogenetic network N on X.

(b) Phylogenetic network N' on X.

(c) The underlying generator G_N of N and N' with symmetry.

Figure 2.5: Two equal networks N, N' with their underlying generator G_N having symmetry.

3

Restricting the problem

To determine networks according to their level, it would be helpful if there is a way to reduce the problem to a smaller problem. For example, a level-k network contains at least one biconnected component with k reticulations, but the number of reticulations in the other biconnected components is just at least k, and there can be many nontrivial biconnected components, which makes the network more complex. In this chapter, we will first generalise existing results for crucial subnets. They are of great importance to determine the structure of a simple network. Thereafter, we will show how our determination problem can be reduced to the strongest result, namely encoding networks with p-nets, where p is as small as possible. Finally, it will turn out that we can restrict ourselves to the level-k biconnected component of a level-k network to determine whether it is encoded by subnets. Therefore, we will mostly focus on simple networks in this thesis.

3.1. Crucial subnets

The existence of a crucial k-net for k as small as possible is useful to reconstruct the whole phylogenetic network from its subnets. In Nipius, 2020 it is proved for simple level-3 networks that they have at least one crucial trinet, see the lemma below. This can be generalised for level-k networks; this is stated in Lemma 2.

Lemma 1. (Lemma 5.1 in Nipius, 2020) *If G is a level-3 generator, then it has a set of crucial sides of size at most 3. Hence, every simple level-3 network N has at least one crucial trinet.*

Lemma 2. *Let N be a binary, simple level-k network on X with $|X| \geq c$ where c is the size of a set of crucial sides of G_N, with $c \leq k$. Then N has at least one crucial c-net.*

Proof. Let $\{X_1, \ldots, X_c\}$ be a set of crucial sides of G_N such that each side contains at least one leaf, and let $x_1, \ldots, x_c$ be leaves on sides $X_1, \ldots, X_c$ respectively. Let $C \in \mathcal{S}_c(N)$ be the c-net on $x_1, \ldots, x_c$. Since N is a phylogenetic network and simple, N contains a leaf on each crucial side which is an outdegree-0 vertex in its underlying generator. Also, for a pair of parallel arcs in G_N, at least one of these arcs contains a leaf of N. Therefore, there indeed exists a set of crucial sides of G_N such that each side contains at least one leaf of N. C now contains exactly one leaf on each side in some set of crucial sides of G_N, so C is a crucial c-net. □

What it means for a trinet to be crucial, becomes clear in the following lemma from Van Iersel and Moulton, 2012. It has the same generator as the whole phylogenetic network. Therefore, it is easier to say something about the sides of the leaves, what we need in the next sections. We will generalise this by proving Lemma 4.

Lemma 3. (Lemma 1 in Van Iersel and Moulton, 2012) *Let N be a simple level-k network, G_N its underlying generator and $T \in Tn(N)$. Then, T is a crucial trinet of N if and only if T is a simple level-k network. Moreover, if T is a crucial trinet of N then G_N is its underlying generator.*

Lemma 4. *Let N be a simple level-k network, G_N its underlying generator and P, a p-net of N with $p \geq 2$. Then, P is a crucial p-net of N if and only if P is a simple level-k network. Moreover, if P is a crucial p-net of N, then G_N is the underlying generator of P.*

Proof. Let P be a crucial p-net exhibited by N. By definition, P contains at least one leaf on each side in some set of crucial sides of G_N, the underlying generator of N. Furthermore, N contains k reticulation vertices. The following claim is useful to prove that no reticulations are deleted to obtain P from N.

Claim: Let N be a binary, simple network. There exists an outdegree-0 vertex below each arc in its underlying generator.
Proof: Let G_N be the underlying generator of network N. All vertices in G_N have outdegree $x \in \mathbb{Z}_{\geq 0}$ by construction. Let $a = (e, f)$ be an arc in G_N. Suppose there does not exist an outdegree-0 vertex below a. It follows that each vertex below a has outdegree $x \in \mathbb{Z}_{>0}$. As a consequence, each vertex below a has at least one vertex below it, different from itself. Then $V(G_N) \to \infty$. By construction of G_N, $V(N) \to \infty$, which contradicts with $V(N)$ being finite. So for each arc in G_N, there is an outdegree-0 vertex below it. □

It also follows that there is an outdegree-0 vertex below each vertex of G_N. So P contains a leaf below each reticulation. Therefore, no reticulation vertices are deleted. Also, P contains at least one leaf on one of the arcs forming a pair of parallel arcs in G_N. As a consequence, no parallel arcs are suppressed by constructing P from N. It follows that P contains k reticulation vertices. Hence, P is a simple level-k network. Indeed, if P is not simple, some reticulation vertex must be deleted.

No reticulation vertices are deleted and no parallel arcs are suppressed to obtain P from N. Then, if a vertex is deleted to obtain P from N, then it is a leaf or a parent of a leaf. It follows that G_P, the underlying generator of P, is isomorphic to G_N. We can conclude that G_N is also the underlying generator of P.

To prove the if-direction, assume, for the sake of contradiction, that P is not a crucial p-net. Then for each set of crucial sides of G_N, there is some side in this set for which P does not contain a leaf on that side. Then there are two cases to distinguish. The first case is that there is some leaf, a crucial leaf, that is the child of an indegree-2 outdegree-0 reticulation in N that is not contained in P. It follows that at least one reticulation is deleted to obtain P from N. The second case is that there is some pair of parallel arcs in G_N for which P does not contain a leaf that is on any of these parallel arcs in N. In this case, a pair of parallel arcs is suppressed to obtain P from N. In both cases a reticulation vertex is deleted, so P has less reticulations than N. We can conclude that P is not simple level-k, otherwise P would have exactly k reticulation vertices. Note that this proof holds for all $p \geq 2$ so the lemma follows. □

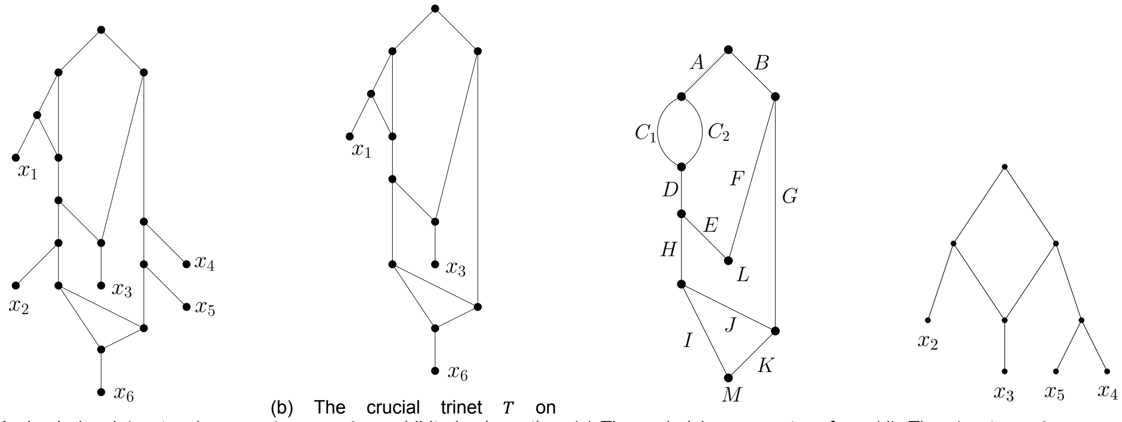
If, for example, a leaf on a side which is an outdegree-0 vertex in a generator is not contained in a p-net exhibited by some network N, then this p-net will not have the same underlying generator. If x_1, x_3 or x_6 is not contained in a p-net exhibited the network N in Figure 3.1a, then G_N is not the underlying generator of the p-net. In Figure 3.1, we see that G_N is the underlying generator of the trinet on $\{x_1, x_3, x_6\}$. Indeed, $\{C_1, L, M\}$ is a set of crucial sides of size three, while N is a level-4 network. The underlying generator of the 4-net on $\{x_2, x_3, x_4, x_5\}$ is not G_N. In fact, the 4-net on $\{x_2, x_3, x_4, x_5\}$ is not simple and an underlying generator is only defined for simple networks. Therefore, the underlying generator of this 4-net is not defined and so it is not G_N, too.

3.2. Getting stronger results

We will now prove that if a network is determined by its set of p-nets, then it is determined by its set of $(p + 1)$-nets. This is very useful because if one is able to prove that a certain network is encoded by its set of p-nets, then the network is encoded by its set of q-nets for $q \geq p$. It also shows that a lower p gives a stronger result.

Lemma 5. *Let N and N' be phylogenetic networks on at least p leaves. If their sets of p-nets are the same, then their sets of $(p - 1)$-nets are the same.*

Proof. Let $x_1, x_2, \dots, x_{p-1}$ be $p-1$ leaves of N. Since $|X| \geq p$, there exists a p-net $P \in \mathcal{S}_p(N)$ containing leaves $x_1, x_2, \dots, x_{p-1}$. Then, the $(p-1)$-net $P_{-1} \in \mathcal{S}_{p-1}(P)$ on $\{x_1, x_2, \dots, x_{p-1}\}$ exhibited by P is, by construction, in the set of $(p-1)$-nets exhibited by N, $\mathcal{S}_{p-1}(N)$. Since $\mathcal{S}_p(N) = \mathcal{S}_p(N')$, P is also a p-net of N', so P_{-1} is, also by construction, contained in the set of $(p-1)$-nets exhibited by N', $\mathcal{S}_{p-1}(N')$. So $P_1 \in \mathcal{S}_{p-1}(N) \implies P_1 \in \mathcal{S}_{p-1}(N')$. By exactly the same reasoning, $P_1 \in \mathcal{S}_{p-1}(N') \implies P_1 \in \mathcal{S}_{p-1}(N)$. $x_1, x_2, \dots, x_{p-1}$ were chosen arbitrarily, so we can conclude that $\mathcal{S}_{p-1}(N) = \mathcal{S}_{p-1}(N')$. □



(a) A simple level-4 network N on X.

(b) The crucial trinet T on $\{x_1, x_3, x_6\}$ exhibited by the network N.

(c) The underlying generator of T, with $G_T = G_N$.

(d) The 4-net on $\{x_2, x_3, x_4, x_5\}$, which has no underlying generator.

Figure 3.1: A crucial trinet with the same underlying generator as the phylogenetic network N, and an example of a 4-net that is not crucial.

Corollary 1. *Let N be a network on at least $p + 1$ leaves. If N is encoded by its set of p-nets, then it is encoded by its set of $(p + 1)$-nets.*

Proof. Let N be a network on at least $p + 1$ leaves which is encoded by its set of p-nets. Suppose N is not encoded by its set of $(p+1)$-nets, then there exists a network $N' \neq N$ for which $\mathcal{S}_{p+1}(N') = \mathcal{S}_{p+1}(N)$. It follows from Lemma 5 that $\mathcal{S}_p(N') = \mathcal{S}_p(N)$, which leads to a contradiction with the assumption that N is encoded by its set of p-nets. We can conclude that N is encoded by its set of $(p + 1)$-nets. □

3.3. Reducing the problem to simple networks

A large number of nontrivial biconnected components in a network can make our determination problem more complicated. In this section, we will generalise Theorem 2 from Van Iersel and Moulton, 2012. This theorem simplifies our problem a lot, since we have only to consider simple networks instead of all networks as a whole. For example, in Nipius, 2020 the problem for level-3 networks is reduced to only 65 level-3 generators. First, we will define cut-arc sets and give a decomposition theorem, Theorem 1, which is proved by Van Iersel and Moulton, 2012.

Definition 3.1. Let N be a phylogenetic network on X and $X' \subseteq X$. X' is a *CA-set (cut-arc set)* of N if there exists a cut-arc $(u, v) \in E(N)$ such that $X' = \{x \in X | x \text{ is below } v\}$.

Note that a CA-set is a set of leaves of the network. For example, $\{x_4, x_5\}$ in the network N in Figure 2.1 is a CA-set because x_4, x_5 are all leaves below b where (a, b) is a cut-arc of N.

Theorem 1. (Theorem 1 in Van Iersel and Moulton, 2012) *Let N be a recoverable binary phylogenetic network on X, and $A \subset X$. Then, A is a CA-set of N if and only if $|A| = 1$ or, for all $z \in X \backslash A$ and $x, y \in A$ with $x \neq y$, $\{x, y\}$ is a CA-set of the trinet on $\{x, y, z\}$ exhibited by N.*

We will prove the following corollary of this and use this in the proof of Lemma 7.

Corollary 2. *Let N and N' be two phylogenetic networks on X. If $\mathcal{S}_k(N) = \mathcal{S}_k(N')$ for some $k \geq 3$, then the CA-sets of N and N' are the same.*

Proof. Let $A \subset X$ be a CA-set of N with $|A| \geq 2$. Then by Theorem 1, for all $z \in X \setminus A$ and $x, y \in A$ with $x \neq y$, $\{x, y\}$ is a CA-set of the trinet T on $\{x, y, z\}$ exhibited by N. $\mathcal{S}_k(N) = \mathcal{S}_k(N')$, so $\mathcal{S}_p(N) = \mathcal{S}_p(N')$ for $2 \leq p \leq k$ by Lemma 5. Let T' be the trinet on $\{x, y, z\}$ exhibited by N'. Then, $T = T'$, so $\{x, y\}$ is a CA-set of T' where z was chosen arbitrarily. Then, A is a CA-set of N' by Theorem 1. If $|A| = 1$, then A is trivially a CA-set of N', too. □

In Van Iersel and Moulton, 2012, the decomposition theorem below, which can only be used for encoding with trinets, is proved. We state this theorem for the sets of k-nets of a network, see Theorem 3, such that we can use it in the whole thesis.

Theorem 2. (Theorem 2 in Van Iersel and Moulton, 2012) *A recoverable phylogenetic network N on X, with $|X| \geq 3$, is encoded by its set of trinets $Tn(N)$ if and only if, for each nontrivial biconnected component B of N with at least four outgoing cut-arcs, N_B is encoded by $Tn(N_B)$.*

Theorem 3. *A recoverable binary phylogenetic network N on X with $|X| \geq k$ is encoded by its set of k-nets $\mathcal{S}_k(N)$ if and only if for each nontrivial biconnected component B of N with at least $k+1$ outgoing cut-arcs, N_B is encoded by $\mathcal{S}_k(N_B)$.*

Proof. In the rest of this section, we will prove Theorem 3.

'Only if' direction

To prove the 'only if' direction, suppose that N is a recoverable binary phylogenetic network on X with $|X| \geq k$ that is encoded by its set of k-nets $\mathcal{S}_k(N)$. For the first part, we will mostly follow the proof of Theorem 2 in Van Iersel and Moulton, 2012. Suppose, for the sake of contradiction, that there exists a nontrivial biconnected component B of N such that N_B is not encoded by its set of k-nets $\mathcal{S}_k(N_B)$, where N_B has at least $k+1$ outgoing cut-arcs. That is, there exists a recoverable network N'_B such that $N'_B \neq N_B$ and $\mathcal{S}_k(N'_B) = \mathcal{S}_k(N_B)$. It follows $Tn(N'_B) = Tn(N_B)$ by Lemma 5 and by Theorem 1 and corollary 2, N'_B has the same CA-sets as N_B. Hence, all CA-sets of N'_B are singletons, because the CA-sets of N_B are. Indeed, N_B is biconnected.

Claim: N'_B has no redundant biconnected components.
Proof: Suppose, for the sake of contradiction, that N'_B has a redundant biconnected component. Then there is only one arc below it by definition. If this is not a cut arc, then this redundant biconnected component is 'embedded' in a bigger biconnected component, so then this is not a biconnected component by definition. So the outgoing arc of the redundant biconnected component must be a cut-arc, and so it's a leaf, say x, by the result in the previous paragraph. Then all trinets containing x have a redundant biconnected component with x directly below it. But since $Tn(N'_B) = Tn(N_B)$ and since B is biconnected, for each leaf x, there exists a trinet in $Tn(N_B)$ with no redundant biconnected component. We have now proved the claim that N'_B has no redundant biconnected components. □

Combining this with the observation that all CA-set of N'_B are singletons, it now follows that N'_B consists of one nontrivial biconnected component with leaves attached to it by cut-arcs, i.e. it is a simple network. We want to contradict that N is encoded by $\mathcal{S}_k(N)$. We can do this by showing that there exist a network $N' \neq N$ with $\mathcal{S}_k(N') = \mathcal{S}_k(N)$. We use $N'_B \neq N_B$ to construct N'. Let B' be the nontrivial biconnected component of which N'_B is the restriction. Let N' be the resulting network after replacing B by B' in N. Now $N' \neq N$, and N' is recoverable, since we only replaced B by B', and N'_B is simple. What is left to show is $\mathcal{S}_k(N') = \mathcal{S}_k(N)$, while $N \neq N'$, which will contradict the fact that N is encoded by $\mathcal{S}_k(N)$.

To show $\mathcal{S}_k(N') = \mathcal{S}_k(N)$, let $K \in \mathcal{S}_k(N)$ be a k-net on $\{x_1, \ldots, x_k\} \subset X$. We consider different places where these leaves can be and prove that $K \in \mathcal{S}_k(N')$ for each case.

First, suppose that $x_1, \ldots, x_k$ are all below the same cut-arc leaving B. Then B is not contained in K by construction. Indeed, $LSA(x_1, \ldots, x_k) \notin V(B)$. Then, since the only difference between N and N' is that B is replaced by B', it holds that $K \in \mathcal{S}_k(N')$.

Second, suppose that $x_1, \ldots, x_k$ are all not below a cut-arc leaving B. Then then $K \in \mathcal{S}_k(N')$ since K contains no arc or vertex of B.

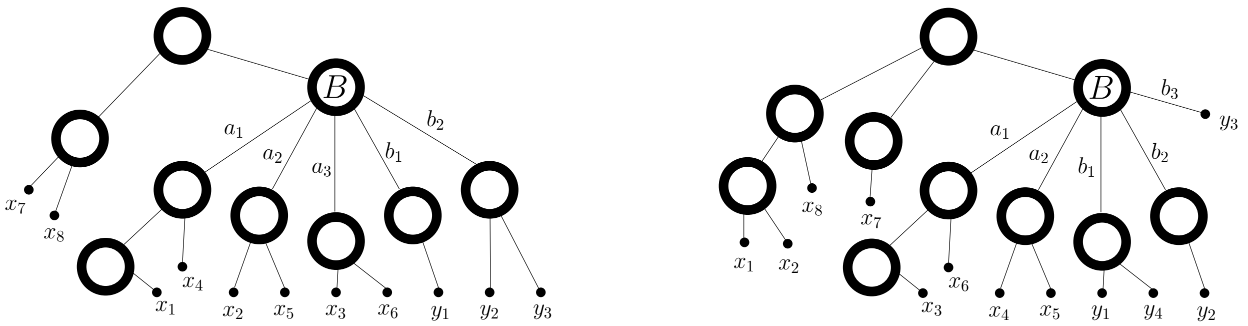
Third, suppose that the leaves are all below different cut-arcs leaving B. Then clearly $K \in \mathcal{S}_k(N')$ since $\mathcal{S}_k(N_B) = \mathcal{S}_k(N'_B)$ and the only difference between N and N' is that B is replaced by B'.

Fourth, suppose that $x_1, \ldots, x_k$ are below q different cut-arcs $a_1, \ldots, a_q$ leaving B, with $q < k$. Also, suppose without loss of generality that $x_1, \ldots, x_q$ are below $a_1, \ldots, a_q$, respectively. Then, consider leaves $y_1, \ldots, y_{k-q}$ below $k-q$ different cut-arcs $b_1, \ldots, b_{k-q}$ leaving B, such that $a_i \neq b_j \; \forall i, j$. Note that these leaves exist because there exists at least one leaf in N below each cut-arc. Since $\mathcal{S}_k(N_B) = \mathcal{S}_k(N'_B)$, the k-nets on $\{x_1, \ldots, x_q, y_1, \ldots, y_{k-q}\}$ exhibited by N and N' are the same, where $x_1, \ldots, x_q$ are below $a_1, \ldots, a_q$, respectively. $\mathcal{S}_q(N_B) = \mathcal{S}_q(N'_B)$ by Lemma 5, so the q-nets on $\{x_1, \ldots, x_q\}$ exhibited by N and N' are the same. Therefore, the k-nets on $\{x_1, \ldots, x_k\}$ exhibited by N and N' are the same. It follows that $K \in \mathcal{S}_k(N')$. Suppose that in the network in Figure 3.2a $k = 4$ and $q = 3$. N_B has five outgoing cut-arcs. Consider for example leaves x_1, x_2, x_3 and x_6. They are below three different cut-arcs leaving B. The 4-nets on $\{x_1, x_2, x_3, y_1\}$, exhibited by N_B and N'_B, respectively, are the same, and therefore, the 4-nets on $\{x_1, x_2, x_3, x_6\}$ are the same.

Fifth, suppose that $x_1, \dots, x_{q'}$ are not below a cut-arc leaving B with $q' < k$, and $x_{q'+1}, \dots, x_k$ are below q different cut-arcs $a_1, \dots, a_q$, with $q \le k - q'$. By the same reasoning as before, the $(k - q')$-nets on $\{x_{q'+1}, \dots, x_k\}$ exhibited by N and N' are the same.

Claim: For a simple network N on X with $|X| \ge 3$, let ρ be the root of N. Then, $LSA(X) = \rho$.
Proof: Suppose for the sake of contradiction that $LSA(X) \neq \rho$. Then $LSA(X)$ is a vertex below ρ, say $w \in V(N)$. All paths from ρ to any $x \in X$ pass through w. w is a lowest stable ancestor, so w is not outdegree-1. Then w is indegree-1 since N is a phylogenetic network. Let u be the only parent of w in N. Then, (u, w) is a cut-arc of N. This contradicts with N being simple, so $LSA(X) = \rho$. □

By the claim and by Observation 1 there exist two leaves x, y of N_B below two different cut-arcs leaving B such that $LSA(x, y)$ is the root of N_B. $\mathcal{S}_k(N_B) = \mathcal{S}_k(N'_B)$ so $LSA(x, y)$ is then also the root of N'_B. Then, the k-net on $x_1, \dots, x_k$ exhibited by N and N' are the same, so $K \in \mathcal{S}_k(N')$. Indeed, extending the $(k - q')$-net on $\{x_{q'+1}, \dots, x_k\}$ to K is independent of B or B'. Lastly, if $k - q' = 1$, then there is only one leaf, say x, below a cut-arc of B. Then, all paths from the root of N_B in N are contained in K. That is also the case for some trinet containing x of which all leaves are below cut-arcs leaving B. We can conclude $K \in \mathcal{S}_k(N')$. Suppose that in the network in Figure 3.2b $k = 4$ and $q' = 2$. We see that x_1, x_2 are not below a cut-arc leaving B, and x_3, x_4 are below a_1 and a_2, respectively. The 4-nets on $\{x_3, x_4, y_1, y_2\}$ exhibited by N_B and N'_B, respectively, are the same, and therefore the binets on $\{x_3, x_4\}$, exhibited by N and N',respectively, are the same. It follows that the 4-nets on, for example, $\{x_1, x_2, x_3, x_4\}$ exhibited by N and N', respectively, are the same.



(a) (b)

Figure 3.2: Examples of phylogenetic networks, where the circles represent biconnected components.

It follows that $K \in \mathcal{S}_k(N')$ for all cases, so $\mathcal{S}_k(N') = \mathcal{S}_k(N)$. So there is a network N' such that $N' \neq N$ and $\mathcal{S}_k(N') = \mathcal{S}_k(N)$. This contradicts with N is encoded by its set of k-nets.

'If' direction

To prove the 'if' direction, let N be a recoverable phylogenetic network on X such that for each nontrivial biconnected component B with at least $k + 1$ outgoing cut-arcs the network N_B is encoded by $\mathcal{S}_k(N_B)$. Let N' be a recoverable network on X with $\mathcal{S}_k(N) = \mathcal{S}_k(N')$. We will show that $N = N'$, so that N is encoded by its set of k-nets, for $k \ge 3$ by following in general the proof of Theorem 2 in Van Iersel and Moulton, 2012. Note, if a biconnected component B has exactly k outgoing cut-arcs, N_B is trivially encoded by $\mathcal{S}_k(N_B)$, since in that case N_B is isomorphic to the single k-net in $\mathcal{S}_k(N_B)$. Therefore we consider 'at least $k + 1$' outgoing cut-arcs.

We will continue with induction on $|X|$. If $|X| = k$, then, since N and N' are recoverable, they are both equal to the single k-net in $\mathcal{S}_k(N)$ and we are done. Assume $|X| \ge k$. Consider the root ρ of N. We shall assume that ρ is in some biconnected component B_ρ and that $a_1 = (u_1, v_1), \dots, a_b = (u_b, v_b)$ are the cut-arcs leaving B_ρ. The network in Figure 3.3a shows an example of a network where B_ρ has two outgoing cut-arcs. If ρ is not in a biconnected component, then you can see the root itself as a component with two outgoing arcs and $b = 2$. Note, if $b = 1$, there is a strongly redundant biconnected component.

Let $N_1, \dots, N_b$ be the networks rooted at $v_1, \dots, v_b$. More precisely, for $i \in \{1, \dots, b\}$, let N_i be the network obtained from N by deleting all vertices that are not below v_i. Suppose that X_i is the leaf-set of N_i. Then, since $b \geq 2$, we have $|X_i| < |X|$. Note that N_i is not necessarily recoverable. In Figure 3.3a, N_1 is the black part of the network, and N_2 is the red part of the network. Note that N_1 is not recoverable and $X_1 = \{x, x_1, x_2, y\}$.

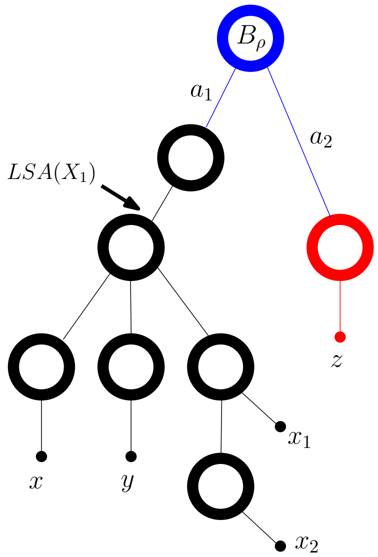
Since the root of N is in some biconnected component B_ρ, it follows that also the root ρ' of N' is in some nontrivial biconnected component $B_{\rho'}$. In the same way as before, let $a'_1 = (u'_1, v'_1), \dots, a'_b = (u'_b, v'_b)$ be the cut-arcs leaving $B_{\rho'}$. Let $N'_1, \dots, N'_b$ be the networks rooted at $v'_1, \dots, v'_b$. Now, by Corollary 2, N' has the same CA-sets as N, because the set of k-nets are the same. Thus, X_i is a CA-set of N' for $i = 1, \dots, b$. So let N'_i be the network on X_i for $i = 1, \dots, b$. To show that $N = N'$, it remains to show that $N_{B_\rho} = N'_{B_{\rho'}}$ and $N_i = N'_i$ for $i = 1, \dots, b$.

To show $N_{B_\rho} = N'_{B_{\rho'}}$, we first observe that $\mathcal{S}_k(N_{B_\rho}) = \mathcal{S}_k(N'_{B_{\rho'}})$. Indeed, if for any k leaves the k-nets exhibited by N_{B_ρ} and $N'_{B_{\rho'}}$ are not the same, then for any k leaves $x_1, \dots, x_k$ below $a_1, \dots, a_k$ respectively, the k-nets exhibited by N and N' are not the same, which is a contradiction. First suppose that $b > k$. Since we assumed N_B encoded by $\mathcal{S}_k(N_B)$ and we observed $\mathcal{S}_k(N_{B_\rho}) = \mathcal{S}_k(N'_{B_{\rho'}})$, it directly follows $N_{B_\rho} = N'_{B_{\rho'}}$. Indeed, for each nontrivial biconnected component B of N with at least $k+1$ outgoing cut-arcs N_B is encoded by $\mathcal{S}_k(N_B)$. Second, suppose that $b = k$. This case is trivial, because there is only one k-net exhibited by N_{B_ρ}. Remember, every cut-arc has become a leaf in N_{B_ρ} so clearly $N_{B_\rho} = N'_{B_{\rho'}}$. Third, suppose that $b < k$. Consider q_i leaves of N below the cut-arc a_i leaving B_ρ, $i = 1, \dots, b$, and $\sum_{i=1}^{b} q_i = k$. These k leaves exists since $|X| \geq k$. Consider the k-net K on the k leaves below the cut-arcs of B_ρ. Note that this k-net is not a restriction of N to a biconnected component, it's just a k-net exhibited by N, i.e. $K \in \mathcal{S}_k(N)$. Let B_ρ^K be the biconnected component of K containing the root of K. Then, $N_{B_\rho^K} = N_{B_\rho}$. Moreover, $K \in \mathcal{S}_k(N')$ since we assumed $\mathcal{S}_k(N') = \mathcal{S}_k(N)$. It follows that $N_{B_\rho^K} = N'_{B_{\rho'}}$ and we can conclude that $N_{B_\rho} = N'_{B_{\rho'}}$ for all $b \geq 2$.

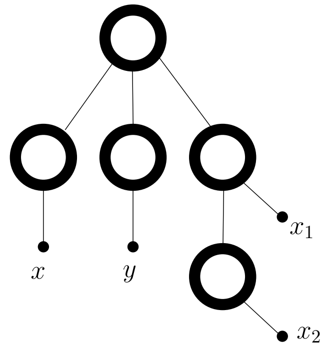
We will show $N_i = N'_i$ for $i = 1, \dots, b$. For the same reasons as in the previous paragraph, $\mathcal{S}_k(N_i) = \mathcal{S}_k(N'_i)$. We will construct recoverable networks from N_i and N'_i and do the proof in two parts. Consider R_i and R'_i obtained from N_i and N'_i respectively by suppressing all strongly redundant biconnected components. Then, R_i and R'_i are recoverable. The network in Figure 3.3b is R_1; it is obtained from the black part of N by suppressing the strongly redundant biconnected component. First, we will prove $R_i = R'_i$. Second, we will prove that N'_i and N_i have the same strongly redundant biconnected components, in the same order.

We distinguish between the number of leaves. Note that we assumed that N_B is encoded by $\mathcal{S}_k(N_B)$ for each biconnected component B and that $\mathcal{S}_k(N) = \mathcal{S}_k(N')$. So first suppose that $|X_i| \geq k$. $|X_i| = k$ is a trivial case. Since we observed that $\mathcal{S}_k(N_i) = \mathcal{S}_k(N'_i)$, it directly follows $N_i = N'_i$. Indeed, for each nontrivial biconnected component B of N with at least $k+1$ outgoing cut-arcs N_B is encoded by $\mathcal{S}_k(N_B)$. Second, suppose that $|X_i| = 1$. It follows that $R_i = R'_i$ because both consist of a single leaf. Third, suppose that $|X_i| = p$ with $2 \leq p < k$. $\mathcal{S}_k(N_i) = \mathcal{S}_k(N'_i)$, so $\mathcal{S}_p(N_i) = \mathcal{S}_p(N'_i)$ for $2 \leq p < k$ by Lemma 5. Let P be the p-net on X_i exhibited by N_i. By construction, P contains no strongly redundant biconnected components. Furthermore, $LSA(X_i)$ is below v_i, so only leaves of X_i are below $LSA(X_i)$. In Figure 3.3a, the location of $LSA(X_1)$ is indicated. In general, the ingoing arc of $LSA(X_i)$ is a cut-arc, and all vertices and arcs below $LSA(X_i)$ are in P. Then $P = R_i$. Let P' be the p-net on X_i exhibited by N'_i. By the same reasoning, $P' = R'_i$. $\mathcal{S}_p(N_i) = \mathcal{S}_p(N'_i)$, so $P = P'$ and therefore $R_i = R'_i$. So in all cases $R_i = R'_i$.

It remains to show that N'_i and N_i have the same strongly redundant biconnected components, in the same order. We distinguish between two cases. First, suppose $|X_i| = 1$ and let $X_i = \{x\}$. For this case, we will follow the proof of Theorem 2 in Van Iersel and Moulton, 2012, too. x is below all strongly redundant biconnected components of N_i, so all leaves of N_i are below them. Then all strongly redundant biconnected components have only one outgoing cut-arc. Let $y, z \in X \backslash X_i$ such that z is below $LSA(u_i)$. Remember $a_i = (u_i, v_i)$ is a cut-arc leaving B_ρ. Note that such a z exists in N since are no cut-arcs above u_i because u_i is contained in B_ρ. Let T be the trinet on $\{x, y, z\}$ exhibited by N. Let a be the cut-arc in T such that x is below a and x, y are not below a and such that there is no such cut-arc above a. Then, arc a corresponds to arc $a_i = (u_i, v_i)$ in N. Indeed, a cannot be above $LSA(u_i)$ since z is below $LSA(u_i)$, and a cannot be between $LSA(u_i)$ and u_i because it is a cut-arc. First, note that $\mathcal{S}_k(N) = \mathcal{S}_k(N')$, so $Tn(N) = Tn(N')$ and $T \in Tn(N')$ by Lemma 5. Now, consider the network T_x obtained from T by deleting all vertices that are not below a. Note that a is in T. Then, $T_x = N_i$ and



(a) The black part is N_1 and the red part is N_2.



(b) R_1 obtained from N_1.

Figure 3.3: Example of a network N with N_1 and N_2 and the construction of R_1, where the circles represent biconnected components.

$T_x = N_i'$. It follows that $N_i = N_i'$.

Second, suppose $|X_i| \geq 2$. Let $x, y \in X_i$ be such that $LSA(x, y) = LSA(X_i)$. These leaves exist by Observation 1. Let $z \in X \setminus X_i$ such that z is below $LSA(u_i)$. z exists by the same reasoning as in the previous paragraph. An example of the locations of x, y and z such that $LSA(x, y) = LSA(X_i)$, $z \in X \setminus X_i$ and z is below $LSA(u_i)$ is displayed in Figure 3.3a. Let T be the trinet on $\{x, y, z\}$ exhibited by N. Then $T \in Tn(N')$ since $Tn(N) = Tn(N')$. Consider the cut-arc a in T such that x and y are below a, z is not below a and such that there is no cut-arc a' with the same properties above a. It follows by the same reasoning as in the previous paragraph that a corresponds to arc $a_i = (u_i, v_i)$ of N. Let D be the directed graph obtained from T by deleting all vertices that are not below a and deleting all vertices that are below $LSA(x, y) = LSA(X_i)$. Then D is isomorphic to the strongly redundant biconnected components of N_i and N_i' because $T \in Tn(N')$ and $T \in Tn(N)$. Therefore, the strongly redundant biconnected components of N_i and N_i' are the same and in the same order.

We have proved that N_i and N_i' have the same strongly redundant biconnected components, in the same order, independent of the number of leaves of N_i. Furthermore, we proved that $R_i = R_i'$, so we can conclude that $N_i = N_i'$ for all $i \in \{1, \dots, b\}$. Together with $N_{B_\rho} = N'_{B_{\rho'}}$ it follows that $N = N'$. So if for each nontrivial biconnected component B of N with at least $k+1$ outgoing cut-arcs N_B is encoded by $\mathcal{S}_k(N_B)$, then N is encoded by its set of k-nets, with N a binary recoverable network on X with $|X| \geq k$. □

4

Encoding level-k networks without symmetry

Underlying generators are used to prove that networks are encoded by subnets. In this thesis, symmetry in underlying generators of simple networks causes some difficulties, that we avoid in this chapter. In Van Iersel and Moulton, 2012 it is proved that level-2 networks are encoded by 4-nets and in Nipius, 2020 it is proved that level-3 networks are encoded by 4-nets. We conjecture that the general case holds. That is, level-k networks are encoded by $(k+1)$-nets. In this chapter we will prove this generalisation for the nonsymmetric case and prove a related result.

4.1. Strongly encoded by $(k+1)$-nets

In this section, we will prove the theorem below from which Corollary 3 will follow. Note that this is a generalisation of Theorem 3 in Van Iersel and Moulton, 2012 for nonsymmetric cases, except for parallel arcs.

Theorem 4. *Every binary, simple level-k network N on X with $|X| \geq k+1$ and without symmetry besides parallel arcs in its underlying generator is encoded by its set of $(k+1)$-nets for $k \geq 2$.*

Proof. Let N be a binary, simple level-k network on X, with $|X| \geq k+1$ and let G_N be its underlying generator, such that G_N has no symmetry other then sets of parallel arcs. Assume that this network is not encoded by its set of $(k+1)$-nets $\mathcal{S}_{k+1}(N)$. Then, there is a recoverable network $N' \neq N$ with $\mathcal{S}_{k+1}(N) = \mathcal{S}_{k+1}(N')$. Then, it follows by Lemma 5 that $\mathcal{S}_p(N) = \mathcal{S}_p(N')$ for $2 \leq p \leq k+1$. We will show that $N = N'$, which is a contradiction, so then the lemma follows.

To prove that $N = N'$, we will first prove that N' is also a binary network. Second, we will prove that N' is a simple network. Third, we will prove that N' is also level-k. Next, we will prove that N and N' have isomorphic generators. Next, we will prove that all leaves in N' are on the same sides of $G_{N'}$ as they are in N. Finally, we will prove that all leaves on each side are in the same order in N and N'. Then, we can conclude that these results lead to $N = N'$.

4.1.1. N' is a binary network

We will prove that N' is a binary network by proving the following lemma.

Lemma 6. *Let N and N' be two phylogenetic networks on X with $|X| \geq 3$. If $Tn(N) = Tn(N')$ and N is binary, then N' is binary.*

Proof. Assume, for the sake of contradiction, that N' contains a vertex v with indegree greater than 2. Let x be a leaf below v and let $y, z \in X$ be leaves such that $LSA(x, y, z) = LSA(X)$. These leaves exist by Observation 1 and since N' is recoverable. In the proof of Theorem 3, we proved the claim that $LSA(X) = \rho$ if N is simple. Then, the trinet on $\{x, y, z\}$ exhibited by N' contains v and all its parents by the claim, the fact that x is below v and the assumption that N' is recoverable. Then, the trinet on $\{x, y, z\}$ is nonbinary. This contradicts with N being binary. Indeed, $Tn(N) = Tn(N')$ by Lemma 5 and $Tn(N)$ contains only binary trinets. So N' has no vertices with indegree greater than 2.

Following the proof in Van Iersel and Moulton, 2012, assume that N' has a vertex v with outdegree greater than 2. Let c_1, c_2 and c_3 be three children of v. Consider three (not necessarily different) leaves x_1, x_2 and x_3 below c_1, c_2 and c_3 respectively. Then, any trinet containing x_1, x_2 and x_3 exhibited by N' is nonbinary. N is a binary network, so $Tn(N)$ contains only binary trinets. $Tn(N) = T(N')$, so $Tn(N')$ contains only binary trinets. This contradicts with N' having a nonbinary trinet. So N' does not contain any vertex with outdegree greater than 2. We can conclude that N' is a binary phylogenetic network, since N' has no vertices with indegree or outdegree greater than 2. □

4.1.2. N' is a simple network

We will prove that N' is a simple network by proving the following lemma.

Lemma 7. *Let N and N' be two phylogenetic networks on X with $|X| \geq 3$. If $Tn(N) = Tn(N')$ and N is simple, then N' is simple.*

Proof. We can follow the proofs of Theorem 2 and 3 in Van Iersel and Moulton, 2012 because $Tn(N) = Tn(N')$ and $|X| \geq 3$ also holds for our case. Since $Tn(N) = Tn(N')$, we have by Corollary 2 that the set of CA-sets of N' equals the set of CA-sets of N. Note that all CA-sets of N, and also of N', are singletons, since N is a simple network. We claim now that N' has no redundant biconnected components. If it has one, then there is only one leaf x below it, otherwise there is a CA-set of two or more leaves. Then all trinets containing leaf x would have a redundant biconnected component with x directly below it. Since $Tn(N) = Tn(N')$ and N is simple, a trinet cannot contain such redundant biconnected component. So N' has no redundant biconnected component and the sets of CA-sets of N and N' are the same. It follows that the head of each cut-arc of N' must be a leaf. N' meets now the definition of a simple network. □

4.1.3. N' is a level-k network

We will prove that N' is a level-k network by proving the following lemma.

Lemma 8. *Let N and N' be binary, simple networks on X with $|X| \geq k+1$. If N is level-k and $\mathcal{S}_{k+1}(N) = \mathcal{S}_{k+1}(N')$, then N' is level-k.*

Proof. N' is a binary simple network on X with $|X| \geq k+1$ and $\mathcal{S}_{k+1}(N) = \mathcal{S}_{k+1}(N')$. Suppose, for the sake of contradiction, that N' is a level-l network with $l > k$. Then, N' has exactly l reticulations, since it is simple. We will now construct a $(k+1)$-net in $\mathcal{S}_{k+1}(N')$ with a level greater than k. We will consider three cases.

First, suppose that there are at least $k+1$ leaves $x_1, \dots, x_{k+1}$ whose parent is a reticulation. Then, let $K_1 \in \mathcal{S}_{k+1}(N')$ be the $(k+1)$-net on $\{x_1, \dots, x_{k+1}\}$.

Second, suppose that there are exactly q leaves $x_1, \dots, x_q$ for $q < k+1$ with a reticulation as parent. If there are at least r leaves $x_{q+1}, \dots, x_{q+r}$ with $q + r = k+1$ on sides that form parallel arcs in the underlying generator, such that we can choose at most one leaf per pair of parallel arcs, let in that case $K_1 \in \mathcal{S}_{k+1}(N')$ be the $(k+1)$-net on $\{x_1, \dots, x_q, x_{q+1}, \dots, x_{q+r}\}$.

Third, suppose that there are exactly q leaves $x_1, \dots, x_q$ for $q < k+1$ with a reticulation as parent. If there are exactly r leaves $x_{q+1}, \dots, x_{q+r}$ for $0 \leq r < (k+1) - q$ on sides that form parallel arcs in the underlying generator, such that we can choose at most one leaf per pair of parallel arcs, let $K_1 \in \mathcal{S}_{k+1}(N')$ be the $(k+1)$-net on $\{x_1, \dots, x_q, x_{q+1}, \dots, x_{q+r}, x_{q+r+1}, \dots, x_{k+1}\}$, where $x_{q+r+1}, \dots, x_{k+1}$ are arbitrary leaves of N'. In Figure 4.1 an example of a level-7 network is given. For this network, K_1 is for example the 8-net on $\{x_1, \dots, x_8\}$, where $q = 2$, $r = 2 < k+1-q = 6$ and $x_5, \dots, x_8$ are chosen arbitrary.

We have chosen the leaves such that K_1 has at least $k+1$ reticulations. Note that if a leaf is chosen on one of the parallel arcs in the underlying generator, the pair of parallel arcs will not be suppressed, so we get a reticulation for that leaf in K_1. As a consequence, $K_1 \in \mathcal{S}_{k+1}(N')$ is a level-l' $(k+1)$-net with $l' > k$.

$\mathcal{S}_{k+1}(N)$ contains only $(k+1)$-nets of level at most k since N is level-k, so $\mathcal{S}_{k+1}(N')$ contains only $(k+1)$-nets of level at most k since $\mathcal{S}_{k+1}(N) = \mathcal{S}_{k+1}(N')$. This contradicts with N' having a level-l' $(k+1)$-net with $l' > k$. So the level of N' cannot be greater than k.

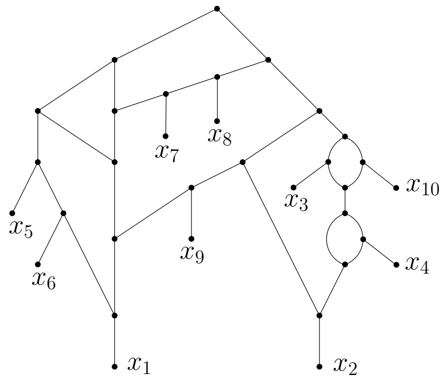


Figure 4.1: A binary, simple level-7 network N on X.

N is level-k with $|X| \geq k+1$. By Lemma 2, N has at least one crucial k-net which is simple level-k by Lemma 4. Since $\mathcal{S}_k(N') = \mathcal{S}_k(N)$, the level of N' is at least k. Combining results gives that N' is a level-k network. □

4.1.4. Isomorphic generators

In this section we will show that G_N is also the underlying generator of N'. First observe that N is a binary, simple level-k network on X with $|X| \geq k+1$. Then, by lemma 2, N has at least one crucial k-net K. So let K be a crucial k-net of N. $K \in \mathcal{K}(N)$ is crucial, so by Lemma 4, K is simple and level-k. Moreover, by Lemma 4, G_N is also the underlying generator of K.

Observe that since $\mathcal{K}(N) = \mathcal{K}(N')$, K is also a k-net of N'. N' is a binary, simple level-k network, $G_{N'}$ its underlying generator and $K \in \mathcal{K}(N')$. K is simple level-k, so by Lemma 4, K is a crucial k-net of N'. Moreover, also by Lemma 4, $G_{N'}$ is the underlying generator of K. G_N and $G_{N'}$ are now both underlying generators of the k-net K. It follows G_N and $G_{N'}$ are isomorphic. We will now prove the following property of generators.

Lemma 9. *Let N and N' be binary, simple level-k networks and let G_N and $G_{N'}$ be their underlying generators, respectively, such that G_N and $G_{N'}$ are isomorphic. Then, G_N and $G_{N'}$ have a set of crucial sides of equal size.*

Proof. Let c be size of a set of crucial sides of G_N. Then, $c \leq k$ because N is level-k. We will prove that a set of crucial sides of $G_{N'}$ is also of size c. N and N' are level-k networks, so G_N and $G_{N'}$ are level-k generators. We have already proved that G_N and $G_{N'}$ are isomorphic, so there exist a bijection $f : V(G_N) \to V(G_{N'})$ such that for any $u, v \in V(G_N)$, (u, v) is an arc of G_N if and only if $(f(u), f(v))$ is an arc of $G_{N'}$. So $|V(G_N)| = |V(G_{N'})|$ and both generators have exactly k reticulation vertices. Let $f : V(G_N) \to V(G_{N'})$ be a bijection as described.

First note that by definition a generator consists of reticulation vertices with indegree 2 and outdegree at most 1, a single vertex with indegree 0 and outdegree 2, and apart from that only tree-vertices with indegree 1 and outdegree 2. Let $v \in V(G_N)$ and suppose that v has indegree-2 and two different parents. If $(u, v), (u', v) \in E(G_N)$ with $u \neq u'$, then $(f(u), f(v)), (f(u'), f(v)) \in E(G_N)$ and $f(u) \neq f(u')$ since f is bijective. So $f(v)$ is indegree-2 and has two different parents in $G_{N'}$. By the same reasoning, the number of parents and the number of children of w and $f(w)$ are the same, as their indegree and outdegree. Therefore, the number of outdegree-0 vertices in G_N is the same as in $G_{N'}$, and the number of indegree-2 outdegree-1 vertices with exactly one parent in G_N is the same as in $G_{N'}$. The latter implies that the number of sets of parallel arcs in G_N is the same as in $G_{N'}$. Then, the size of a set of crucial sides of $G_{N'}$ is c, which proves the lemma. □

4.1.5. The same leaves on the same sides

First observe that $\mathcal{S}_p(N) = \mathcal{S}_p(N')$ for all $2 \leq p \leq k+1$ by Lemma 5. Second, by Lemma 9, we may assume that G_N and $G_{N'}$ have both a set of crucial sides of size c. N and N' are level-k, therefore $c \leq k$ and so $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$. Since we assumed that G_N has no symmetry besides sets of parallel arcs, we are now able to prove that all leaves are on the same sides in N and N'. In this section we will prove the following lemma.

Lemma 10. *Let N and N' be two binary, simple level-k networks on X and let G_N and $G_{N'}$ the isomorphic underlying generators of N and N', respectively, without symmetry besides parallel arcs and with a set of crucial sides of size c. If $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$, then all leaves of N are on the same sides in N'.*

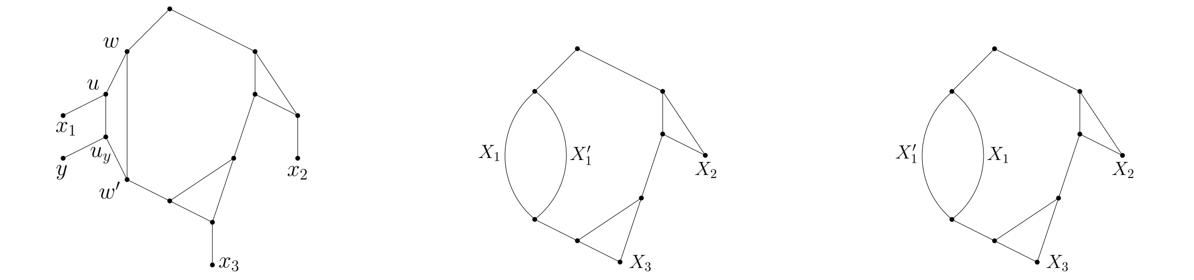
Proof. By Lemma 2, N has a crucial c-net. $|X| \geq c+1$, so N has also a crucial $c+1$-net. Let y be on side Y and let $x_1, \ldots, x_c$ be on sides $X_1, \ldots, X_c$ in N, respectively, such that $\{X_1, \ldots, X_c\}$ forms a set of crucial sides. Let C_1 be the crucial $c+1$-net on $\{x_1, \ldots, x_c, y\} \subseteq X$ exhibited by N. Then, by Lemma 4, leaves $x_1, \ldots, x_c, y$ are also on sides $X_1, \ldots, X_c, Y$ in C_1, respectively. Note that Y can be a noncrucial side, and Y can be an arc in a pair of parallel arcs, too. Let C_1' be the $c+1$-net on $\{x_1, \ldots, x_c, y\}$ exhibited by N'. Since $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$, $C_1' = C_1$ and therefore C_1' is also a simple level-k network. By Lemma 4, C_1' is a crucial $c+1$-net exhibited by N'. Moreover, $G_{N'}$ is the underlying generator of C_1'. To summarize, C_1' equals C_1, G_N is the underlying generator of both N and C_1, $G_{N'}$ is the underlying generator of both N' and C_1', and G_N and $G_{N'}$ are isomorphic generators.

Observe that G_N has no symmetry besides parallel arcs, so there exists no graph automorphism $g : V(G_N) \rightarrow V(G_N)$ such that $g(v) \neq v$ for at least one $v \in V(G_N)$ giving an isomorphic generator. G_N and $G_{N'}$ are isomorphic, so there does not exist such a graph isomorphism $g : V(G_N) \rightarrow V(G_{N'})$ such that $g(v) \neq v$ for at least one $v \in V(G_N)$, too. Furthermore, observe that C_1' and C_1 are equal. Let F be the set of all bijective functions $f : V(C_1) \rightarrow V(C_1')$ such that $f(x) = x$ for each leaf x of C_1 and such that for every $u, v \in V(C_1)$ holds that (u, v) is an arc of C_1 if and only if $(f(u), f(v))$ is an arc of C_1'. Then $F \neq \emptyset$ since $C_1 = C_1'$.

We will begin with proving that each leaf itself is on the same side in N and N'. Suppose that side X_i is an outdegree-0 reticulation in G_N and $u \in V(C_1)$ is the parent of x_i in C_1. Then, since $u \in V(C_1) \cap V(G_N)$ and G_N has no symmetry, it holds that $(u, x_i) \in E(C_1)$, $(f(u), x_i) \in E(C_1')$ and $f(u) = u$ for all $f \in F$. So x_i is on the same side in C_1' as in C_1 and so on the same side in N' as in N. Then, each leaf in N on a side which is an outdegree-0 reticulation in G_N is on the same side in N and N'.

Suppose that side $X_i = (w, w') \in E(G_N)$ is an arc in a pair of parallel arcs and $u \in V(C_1) \setminus V(G_N)$ is the parent of x_i in C_1. Note that $w, w' \in V(G_N) \cap V(C_1)$. By construction of C_1, at least one of the adjacent vertices of u in C_1 is contained in $V(G_N)$. Indeed, x_j is not on side X_i for all $j \in \{1, \ldots, c\}$, $j \neq i$, so only y can possibly be on side X_i. Then, $(f(w), f(u)) \in E(C_1')$ or $(f(u), f(w')) \in E(C_1')$. Also $(f(u), x_i) \in E(C_1')$. Since $f(w) = w$ and $f(w') = w'$ for all $f \in F$, x is again on X_i or its parallel arc in C_1' and so in N'. We can now assume without loss of generality that x_i is on X_i in N', otherwise, we can relabel sides by switching X_i with its parallel arc in $G_{N'}$ such that x_i is on X_i in N' and C_1'. Note that this relabelling is not belonging to an automorphism that maps vertices of G_N. Then, each leaf in N on a side which is in a pair of parallel arcs in G_N is on the same side in N and N'. An example of C_1 is given in Figure 4.2, where x_1 is on side X_1 in N and on its parallel arc, side X_1', in N', as can be seen in Figure 4.2b and 4.2c. Switching side X_1 with X_1' in $G_{N'}$ and mapping all other sides to itself gives that x_1 is also on side X_1 in N'.

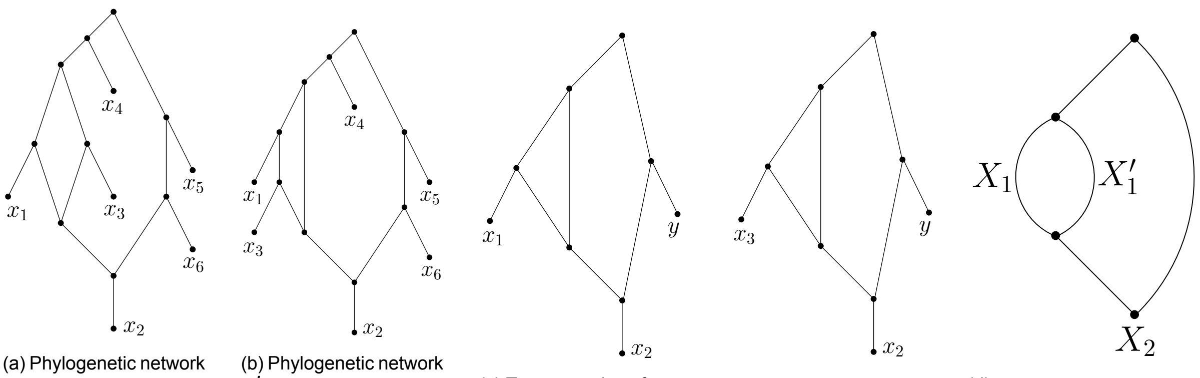
For each pair of parallel arcs of G_N, a relabelling that switches the sides in the set can be applied to A', where A' is the set of sides of $G_{N'}$. A composition of bijective functions is a bijective function, so there exist a relabelling $h : A' \rightarrow A'$ such that after applying h if necessary, each leaf of N' on a side contained in a pair of parallel arcs is on the same side as in N. In order to conclude that not only individual leaves, but all leaves on parallel arcs are on the same sides in N and N', we have prove that after such a relabelling, no leaves that are not considered are moved to a 'wrong' side. So the proof above is not enough to conclude the latter for all leaves on parallel arcs. Figure 4.3 will clarify this. It shows two unequal level-2 networks N and N' on X with isomorphic underlying generators with a set of crucial sides of size 2. If we let C_1 be the trinet on $\{x_1, x_2, y\}$ with $y = x_6$, we can assume by the reasoning above that x_1 is on the same side in N'. The same holds for leaf x_3 by taking the trinet on $\{x_3, x_2, y\}$ with $y = x_6$. But leaf x_3 was not considered if we let C_1 be the trinet on $\{x_1, x_2, y\}$ with $y = x_6$. It turns out that y cannot be taken arbitrarily in this case. A trinet on $\{x_1, x_2, x_6\}$ is not sufficient to conclude that all leaves are on the same sides in N and N', as can be seen for Figure 4.3a and 4.3b. Indeed, if in the network in Figure 4.3a leaf x_1 is considered and is on a different side in N', then sides X_1 and X_1' are switched such that x_1 is on the same side in N and N'. The side of leaf x_3 is then changed, too. To prove that all leaves on parallel arcs, after possibly relabelling sides are on the same sides in N and N', we will prove that two leaves stay together on a side or stay on opposite sides of parallel arcs in N and N', after possibly switching parallel arcs. Only if the latter holds, we can conclude that N and N' have the same leaves on the same sides.



(a) Crucial 4-net C_1 on $\{x_1, x_2, x_3, y\}$. (b) Generator G_N. (c) Generator $G_{N'}$, isomorphic to G_N.

Figure 4.2: Example of $C_1 = C_1'$ with their isomorphic underlying generators, having a pair of parallel arcs and no other symmetry. Leaf x_1 is on a different side in N than in N'.

We will now prove that individual leaves on noncrucial sides are on the same sides in N and N'. Suppose that $Y = (w, w') \in E(G_N)$ is an arc, not in a pair of parallel arcs, with $w, w' \in V(G_N)$. Then, x_j is not on side Y in N for all $j \in \{1, \dots, c\}$. Let $u \in V(C_1)$ be the parent of y. Then, $(w, u), (u, w'), (u, y) \in E(C_1)$, and $(f(w), f(u)), (f(u), f(w')), (f(u), y) \in E(C_1')$. Since $f(w) = w$ and $f(w') = w'$ for all $f \in F$, y is again on Y in C_1' and so in N'. Then, each leaf in N on a side which is an arc, not in a pair of parallel arcs of G_N, is on the same side in N and N'. $\{x_1, \dots, x_c\}$ was chosen as an arbitrary set of crucial leaves and y was chosen arbitrary. Therefore, we can conclude for each leaf of N that it is on the same side in N' as in N, after possibly relabelling sides.



(a) Phylogenetic network N on X. (b) Phylogenetic network N' on X. (c) Two examples of C_1. (d) G_N

Figure 4.3: Example of two unequal networks where we can assume that their individual leaves are on the same sides, using C_1 on two crucial leaves and one arbitrary noncrucial leaf y.

To conclude that all leaves of N' are on the same side as in N, we will now prove that two leaves stay on the same side or, for parallel arcs, stay on opposite sides. We will first prove that if two leaves are together on one arc in a pair of parallel arcs in G_N, they are also together on one arc of $G_{N'}$. Suppose that two leaves are together on one side in N which is contained in a pair of parallel arcs of G_N. Let C_1 be the crucial c-net as before on $\{x_1, \dots, x_c, y\}$ exhibited by N such that x_i and y are together on one arc in a pair of parallel arcs in G_N for some $i \in \{1, \dots, c\}$. Let u_i, u_y be the parents of x_i, y in N respectively. u_i, u_y must be indegree-1 outdegree-2 vertices. Suppose without loss of generality that u_y is below u_i and so that (u_i, u_y) is an arc of C_1. Let C_1' be the $c+1$-net on $\{x_1, \dots, x_c, y\}$ exhibited by N'. We know that C_1 equals C_1', so there exists a bijective function $f : V(C_1) \to V(C_1')$ such that $f(x) = x$ for each leaf x of C_1 and such that for every $u, v \in V(C_1)$ it holds that (u, v) is an arc of C_1 if and only if $(f(u), f(v))$ is an arc of C_1'. Let f be such a function. We know that (u_i, x_i), (u_y, y) and (u_i, u_y) are arcs of C_1, $f(x_i) = x_i$ and $f(y) = y$. Then $(f(u_i), f(x_i)) = (f(u_i), x_i))$, $(f(u_y), f(y)) = (f(u_y), y)$ and $(f(u_i), f(u_y))$ are arcs of C_1' while x_i and y are leaves. So there is an arc from the parent of x_i to the parent of y in C_1', and $f(u_i), f(u_y)$ must also be indegree-1 outdegree-2 in C_1' because f preserves adjacency. It follows that x_i and y are together on one side in C_1'. Furthermore, since G_N and $G_{N'}$ are the generators of C_1 and C_1' respectively, x_i and y are together on one side in N'.

Now we prove that if two leaves are on opposite sides in a pair of parallel arcs in N, then they are also on opposite sides in N'. So suppose that two leaves are on two different sides of a pair of parallel arcs in N. Let C_1 be the crucial $c+1$-net on $\{x_1, \dots, x_c, y\}$ exhibited by N as defined before such that y and x_i are on opposite sides of a pair of parallel arcs of G_N. That is, u_i and u_y are tree-vertices that share their parent and exactly one child in C_1, where u_i and u_y are the parents of x_i and y, respectively. Let $u \in V(C_1)$ be the tail vertex of both parallel arcs and v be the head vertex. Then, $\{(u_i, x_i), (u_y, y), (u, u_i), (u, u_y), (u_i, v), (u_y, v)\} \subset E(C_1)$. Let C_1' be the subnet exhibited by N' as defined before. $C_1 = C_1'$, so there exists again a bijective function $f : V(C_1) \to V(C_1')$ as defined before. Let f_1 be such a function. $f(x_i) = x_i, f(y) = y$ and $f(u) = u$ and $f(v) = v$ since $u, v \in V(G_N) \subset V(C_1)$. Then, $\{(f(u_i), x_i), (f(u_y), y), (u, f(u_i)), (u, f(u_y)), (f(u_i), v), (f(u_y), v)\} \subset E(C_1')$, while x_i and y are leaves. It follows that x_i and y are on different sides in a pair of parallel arcs in C_1' and so in N'.

At last, we will now prove that if two leaves are together on a noncrucial side in N, then they are together on one side in N'. Suppose that $y_1, y_2 \in X$ are together on one side which is not an arc from a pair of parallel arcs in G_N. Let Y be this noncrucial side and let C_{y_1} be the crucial $c+1$-net on $\{x_1, \dots, x_c, y_1\}$ exhibited by N, such that $x_1, \dots, x_c$ are on sides $X_1, \dots, X_c$ respectively, as defined before. Let C_{y_1}' be the crucial $c+1$-net on $x_1, \dots, x_c, y_1$ exhibited by N'. By construction of C_{y_1} and the proof for individual leaves, we know that y_1 is on side Y of the underlying generator of C_1', which is $G_{N'}$. Note that this is independent of the number of sets of parallel arcs in G_N and without any relabelling of sides. Let now C_{y_2} be the crucial $c+1$-net on $\{x_1, \dots, x_c, y_2\}$ exhibited by N, such that $x_1, \dots, x_c$ are on sides $X_1, \dots, X_c$ respectively, as defined before. By exactly the same reasoning, y_2 is on side Y in $G_{N'}$, without any relabelling of sides. We have chosen y_1 and y_2 arbitrarily on side Y, so for every pair of leaves of N that are together on one side which is not an arc from a pair of parallel arcs in G_N, they are together on one side in N'. Then, for a set of leaves $X' \subset X$, it holds that if all leaves of X' are together on one side in N, then they are together on one side in N'.

To summarize, each leaf of N is on the same side in N', each pair of leaves on one side in N is together on one side in N' and leaves on opposite sides in N stay on opposite sides in N'. Now we can conclude that all leaves of N are on the same side in N', which proves Lemma 10. □

4.1.6. The order of the leaves

We will prove that the order of the leaves on each side is the same in N and N' by proving the following Lemma.

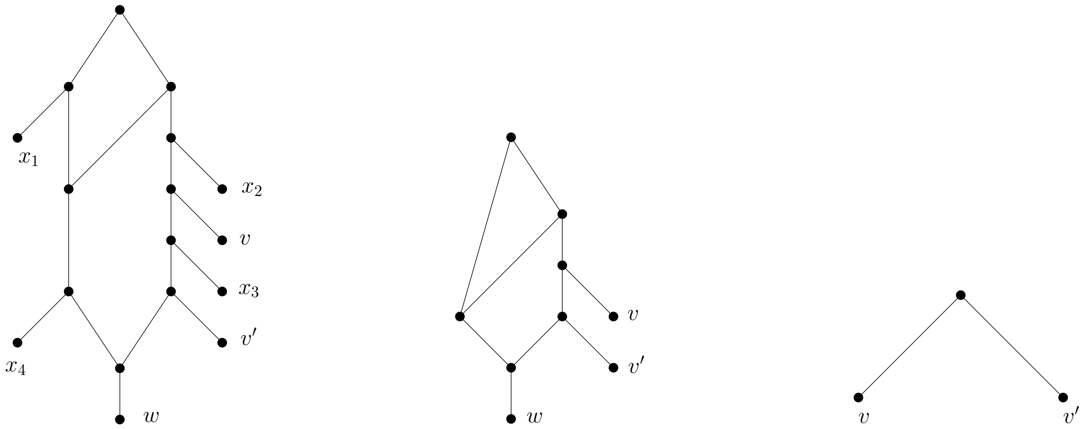
Lemma 11. *Let N and N' be binary, simple networks on X with $|X| \geq 3$, $G_N = G_{N'}$ and $Tn(N) = Tn(N')$. If all leaves are on the same sides in $G_{N'}$ as they are in G_N, then the order of the leaves on each side in N' is the same as in N.*

Proof. Let S be a side in G_N which is an arc. If there is no leaf or only one leaf on S in N, then the order in N' is trivially the same. Suppose there are two or more leaves on side S in N. Note that S is now in $E(G_N)$. Let $v, v' \in X$ be leaves on S in N such that v' is below v. We can suppose this without loss of generality. By the claim in the proof of Lemma 4, there exists an outdegree-0 vertex below each vertex in G_N. So let W be an (arbitrarily) outdegree-0 vertex in G_N below S. Let $w \in X$ be the child of W. Consider the trinet T on $\{v, v', w\}$. Since v' is below v in N and by construction of T (and by definition of 'below'), v' is below v in T. An example of such a trinet exhibited by a network is given in Figure 4.4a. In this trinet, v' is below v, as it is in N. Note that the binet on $\{v, v'\}$ in Figure 4.4c is a so called *cherry*. A cherry is a network consisting of with a root, two leaves, and no other vertices. In this binet, there is no order of the leaves, that is, the leaves are not below each other.

Since $Tn(N) = Tn(N')$, T is a trinet of N'. Given that v and v' are also on side S in N', v' is below v in N'. v and v' are chosen arbitrarily on S, so each pair of leaves on the same side in N' is in the same order as in N. It follows that the order of all leaves on each side in N' is the same as in N. □

4.1.7. Conclusion

We have proved that N and N' are binary, simple level-k networks on X with $|X| \geq k+1$, G_N is isomorphic to $G_{N'}$ and $\mathcal{S}_p(N) = \mathcal{S}_p(N')$ for $2 \leq p \leq k+1$. Furthermore, N and N' contain the same leaves on the same sides. Then, by Lemma 11, the order of the leaves on each side in N' is the same as in N. Now, N and N' have the same underlying generator, the same leaves on each side and on each side the leaves are in the same order. These results imply the existence of a graph isomorphism f between N and N' such that $f(x) = x$ for each leaf $x \in X$, and so $N = N'$. This is a contradiction, so we can



(a) Phylogenetic network N on X. (b) The trinet on $\{v, v', w\}$ exhibited by N. (c) The binet on $\{v, v'\}$ exhibited by N.

Figure 4.4: A trinet where the order of v and v' stays the same as in N, and a cherry where v is not below v' an vice versa.

now conclude that every binary simple level-k network N on X with $|X| \geq k+1$ and without symmetry besides parallel arcs in its underlying generator is encoded by its set of $(k+1)$-nets for $k \geq 2$. This concludes the proof of Theorem 4. □

We can now state Theorem 4 for more general networks.

4.1.8. Corollary

We can now prove Corollary 3, which generalises Theorem 4 from simple to recoverable networks. Note that only the underlying generators of restrictions to biconnected components are required not to have symmetry. There is no such a requirement for a network itself.

Corollary 3. *Every binary, recoverable level-k network N on X, with $|X| \geq k+1$ without symmetry besides parallel arcs in the underlying generator of the restriction of N to any nontrivial biconnected component is encoded by its set of $(k+1)$-nets for $k \geq 2$.*

Proof. The proof follows from Theorem 4, Corollary 1, Theorem 3 and the fact that level-1 networks are encoded by their trinets (Huber and Moulton, 2013). □

4.2. Weakly encoded by smaller subnets

In Section 4.1, we proved that level-k networks are strongly encoded by $(k+1)$-nets, if no biconnected component has symmetry besides parallel arcs. In this section, we will prove that networks in this set are weakly encoded by subnets on a potentially smaller set of leaves. Indeed, for simple networks, we will consider the number of sides c in a set of crucial sides instead of k, the level, with $c \leq k$. The advantage is that these networks can be encoded by smaller subnets if $c < k$. The disadvantage is that we can only prove the weakly encoded version of the theorem, since we do not know how to generalise Lemma 8. Again, we will first prove a theorem for simple networks. Thereafter, we will use Theorem 3 to generalise the result to general networks.

Theorem 5. *Let $c \geq 2$. The class of binary, simple level-k networks with at least $c+1$ leaves, no symmetry in their underlying generator other than sets of parallel arcs, and a set of crucial sides of their underlying generator of size at most c is weakly encoded by $c+1$-nets.*

Proof. Let $c \geq 2$ and let $\mathcal{C}$ be the class of binary, simple level-k networks with at least $c+1$ leaves, no symmetry in their underlying generator other than sets of parallel arcs, and a set of crucial sides of their underlying generator of size at most c. Assume, for the sake of contradiction, that $\mathcal{C}$ is not weakly encoded by $c+1$-nets. Then, there are two networks N and N' on X in $\mathcal{C}$ such that $N \neq N'$ and $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$. We will show that $N = N'$, which is a contradiction.

4.2.1. Isomorphic generators

Let G_N be the underlying generator of N. We will show that G_N is also the underlying generator of N'. First observe that N is a binary, simple level-k network on X with $|X| \geq c+1$ with a set of crucial sides of their underlying generator of size at most c. Suppose G_N contains a set of crucial sides of size q with $q \leq c$. Then, by Lemma 2, N has at least one crucial q-net. So let Q be a crucial q-net exhibited by N.

Observe that since $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$, it follows from Lemma 5 that $\mathcal{S}_q(N) = \mathcal{S}_q(N')$ for $2 \leq q \leq c+1$. We will now follow the proof in Section 4.1.4. It follows that Q is also a q-net exhibited by N'. N' is a binary, simple level-k network, $G_{N'}$ its underlying generator and $Q \in \mathcal{S}_q(N')$. Q is simple level-k, since it is a crucial q-net exhibited by N. Then, by Lemma 4, Q is a crucial q-net exhibited by N'. Moreover, also by Lemma 4, $G_{N'}$ is the underlying generator of Q. G_N and $G_{N'}$ are now both underlying generators of Q. It follows G_N and $G_{N'}$ are isomorphic.

4.2.2. The same leaves on the same sides

First, since we have proved that G_N and $G_{N'}$ are isomorphic, we can deduce that G_N and $G_{N'}$ have a set of crucial sides of equal size by Lemma 9. From now on, we assume that G_N and $G_{N'}$ have a set of crucial sides of size c. Since $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$, it follows that N and N' have the same leaves on the same sides by Lemma 4.1.5

4.2.3. The order of the leaves

Observe again that $\mathcal{S}_q(N) = \mathcal{S}_q(N')$ for $2 \leq q \leq c+1$. N and N' are both binary, simple networks on X. Since $c \geq 2$ and $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$, we know that $|X| \geq 3$ and $Tn(N) = Tn(N')$. Furthermore, G_N and $G_{N'}$ are isomorphic and each leaf in X is on the same side in N' as it is in N. By Lemma 11, it follows that the order of the leaves on each side in N' is the same as in N.

4.2.4. Conclusion

We assumed, for the sake of contradiction, that $\mathcal{C}$ is not weakly encoded by $c+1$-nets with the consequence that there exist two binary, simple level-k networks N and N' on X with $|X| \geq c+1$ with a set of crucial sides of their underlying generator of size at most c such that $N \neq N'$ and $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$. We now have proved, under the assumption that N and N' have a set of crucial sides of size exactly c, that G_N and $G_{N'}$ are isomorphic generators, that the leaves of N are on the same side in N', and that the order of leaves on each side is the same in N' as in N. By Lemma 5, these results still hold if N and N' have a set of crucial sides of size q for $2 \leq q \leq c$, because we only used $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$. These three conditions result in the fact that N and N' must be equal, that is, $N = N'$ which is a contradiction. We can conclude that the class of binary, simple level-k networks with at least $c+1$ leaves and a set of crucial sides of their underlying generator of size at most c is weakly encoded by $c+1$-nets for $c \geq 2$. □

4.2.5. Corollary

The following corollary states the theorem for more general networks. Due to Theorem 3, whether a recoverable network is encoded by subnets depends only on the biconnected components of which the generators have a set of crucial sides of biggest size. This is formulated more precise below.

Corollary 4. *Let $c \geq 2$ and let $\mathcal{C}$ be the class of binary, recoverable level-k networks such that for all networks N in $\mathcal{C}$ it holds that*

- *N has at least $c+1$ leaves;*
- *the underlying generator of the restriction N_B to any nontrivial biconnected component B of N has no symmetry besides parallel arcs;*
- *a set of crucial sides of the underlying generator of the restriction N_B to any biconnected component B is of size at most c.*

Then $\mathcal{C}$ is weakly encoded by $c+1$-nets.

Proof. The proof follows from Theorem 5, Corollary 1, Theorem 3 and the fact that level-1 networks are encoded by their trinets (Huber and Moulton, 2013). □

5

Encoding networks with leaves on a bounded number of sides

In Chapter 4, we focused on the level and the size of a set of crucial sides of a generator to encode nonsymmetric networks by subnets. In this chapter, we will encode networks by subnets on leaves on a bounded number of sides. It turns out that generators with symmetry can be considered in this way, so we get results for a much bigger set of networks. Again, we will first prove theorems for simple networks. Thereafter, we will use Theorem 3 again to generalise our results to recoverable networks.

To encode level-k networks by their subnets by considering the number of sides containing leaves, we will use again crucial subnets of simple networks, as in Chapter 4. Therefore, we prove the following lemma.

Lemma 12. *For a binary, simple level-k network N with $|X| \geq p$ with leaves on p sides of its underlying generator G_N, let $x_1, \ldots, x_p$ be leaves on p different sides in G_N and let $P \in \mathcal{S}_p(N)$ the p-net on $x_1, \ldots, x_p$. Then P is a crucial p-net.*

Proof. Since N is a phylogenetic network and it is simple, N contains a leaf on each crucial side which is an outdegree-0 vertex in its underlying generator. Also, for a pair of parallel arcs in G_N, at least one of these sides contains a leaf in N. Therefore, there exists a set of crucial sides of G_N of size q with $q \leq p$ such that each side contains at least one leaf in N. P is constructed such that it contains exactly one leaf on each side containing at least one leaf in N. So P contains exactly one leaf on each side in some set of crucial sides of G_N, so P is a crucial p-net. □

The following property of simple networks will also be useful in the upcoming sections.

Lemma 13. *Let N and N' be binary, simple level-k networks on X with $|X| \geq p$ and $\mathcal{S}_p(N) = \mathcal{S}_p(N')$. If N and N' contain leaves on q and q' sides of their underlying generator, respectively, with $q, q' \leq p$, then $q = q'$.*

Proof. Let G_N be the underlying generator of N, let $x_1, \ldots, x_q \in X$ such that $x_1, \ldots, x_q$ are on q different sides of G_N and let $Q \in \mathcal{S}_q(N)$ the q-net on $\{x_1, \ldots, x_q\}$. Then Q is a crucial q-net by Lemma 12. Let $P \in \mathcal{S}_p(N)$ be the p-net on $\{x_1, \ldots, x_q, x_{q+1}, \ldots, x_p\}$, where $x_{q+1}, \ldots, x_p$ are arbitrary leaves of X different from $x_1, \ldots, x_q$. These leaves exist since $|X| \geq p$. Then, P is crucial by Lemma 12, so P is a simple level-k network by Lemma 4 and G_N is also the underlying generator of P.

Let P' be the p-net on $\{x_1, \ldots, x_q, x_{q+1}, \ldots, x_p\}$ exhibited by N'. Since $\mathcal{S}_p(N) = \mathcal{S}_p(N')$, it holds that $P = P'$. This means that there exists a bijective function $f : V(P') \to V(P)$ such that $f(x) = x$ for every leaf x of P' and such that for every $u, v \in V(P')$ it holds that (u, v) is an arc of P' if and only if $(f(u), f(v))$ is an arc of P. Furthermore, P' is then also simple level-k, and by Lemma 4, P' is a crucial p-net of N'.

Suppose, for the sake of contradiction, $q \neq q'$. Then, suppose $q > q'$. We can do this without loss of generality, because if $q < q'$, we can interchange the names of N and N' in the theorem. P is crucial and contains, by definition, leaves on q different sides of G_N. Furthermore, $q' < q \leq p$. It follows that there exists at least one pair of vertices in P, say $\{x_i, x_j\} \subset \{x_1, \ldots, x_p\}$ such that x_i and x_j are on two different sides in N and so in P, and together on the same side in N'. Since P' is crucial, x_i and x_j are together on the same side in P' while G_N is the underlying generator of both P and P'. This contradicts with $P = P'$. Therefore, we can conclude that $q = q'$. So if $\mathcal{S}_p(N) = \mathcal{S}_p(N')$ and $q, q' \leq p$, N and N' have leaves on the same number of sides of their underlying generator. □

5.1. Weakly encoded by $(p + 1)$-nets

In this section, we will again concentrate on the generator of a simple network. It turns out that networks with leaves on p sides of their underlying generator are weakly encoded by $(p+1)$-nets for $p \geq 2$. This holds for all level-k networks with $k \geq 1$. Indeed, level-1 networks are encoded by trinets (Huber and Moulton, 2013). The advantage compared to the results in Chapter 4 is that the results hold for generators with symmetry, too. A disadvantage is that the networks are weakly encoded instead of strongly encoded. This is because p can be smaller than k, and we cannot generalise Lemma 8 to this case as we may not have that $\mathcal{S}_{k+1}(N) \neq \mathcal{S}_{k+1}(N')$. Furthermore, we need more leaves in the subnets, making the result also weaker. In Section 5.2 we will prove a stronger variant of the theorem below.

Theorem 6. *Let $p \geq 2$. The class of binary, simple level-k networks with at least $p + 1$ leaves and leaves on at most p sides of their underlying generator is weakly encoded by $(p + 1)$-nets.*

Proof. Let $p \geq 2$ and let $\mathcal{C}$ be the class of binary, simple level-k networks with at least $p + 1$ leaves and leaves on at most p sides of their underlying generator. Assume, for the sake of contradiction, that $\mathcal{C}$ is not weakly encoded by $(p + 1)$-nets. Then, there are two binary, simple level-k networks N and N' on X with $|X| \geq p + 1$ with leaves on at most p sides such that $N \neq N'$ and $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$. We will show that $N = N'$, which is a contradiction, so then the lemma follows.

5.1.1. Leaves on the same number of sides

Let G_N and $G_{N'}$ be the underlying generators of N and N' respectively. Suppose N has leaves on q sides of G_N and N' has leaves on q' sides of $G_{N'}$, with $q, q' \leq p$. $\mathcal{S}_p(N) = \mathcal{S}_p(N')$, so by Lemma 13 it holds that $q = q'$. We can conclude that N and N' have leaves on the same number of sides of their underlying generator.

We may now assume without loss of generality that N and N' have leaves on exactly p sides of their underlying generator. By Lemma 5, $\mathcal{S}_q(N) = \mathcal{S}_q(N')$ holds for $2 \leq q \leq p + 1$, and we will see that therefore the proof of Theorem 6 also holds if N and N' have leaves on q sides of their underlying generator for $2 \leq q \leq p$.

5.1.2. Isomorphic generators

We will show that G_N is also the underlying generator of N'. First observe that N is a binary, simple level-k network on X with $|X| \geq p + 1$ with leaves on p sides. Then, by Lemma 12, N has at least one crucial p-net. So let P be a crucial p-net of N. $P \in \mathcal{S}_p(N)$ is crucial, so by Lemma 4, P is simple and level-k. Moreover, by Lemma 4, G_N is also the underlying generator of P.

Observe that since $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$, it follows from Lemma 5 that $\mathcal{S}_q(N) = \mathcal{S}_q(N')$ for $2 \leq q \leq p + 1$. $\mathcal{S}_p(N) = \mathcal{S}_p(N')$, so P is also a p-net of N'. N' is a binary, simple level-k network, $G_{N'}$ its underlying generator and $P \in \mathcal{S}_p(N')$. P is simple level-k, so by Lemma 4, P is a crucial p-net of N'. Moreover, also by Lemma 4, $G_{N'}$ is the underlying generator of P. G_N and $G_{N'}$ are now both underlying generators of the p-net P. It follows G_N and $G_{N'}$ are isomorphic.

5.1.3. The same leaves on the same sides

We know that $\mathcal{S}_q(N) = \mathcal{S}_q(N')$ for $2 \leq q \leq p + 1$, that G_N and $G_{N'}$ are isomorphic and that both N and N' have leaves on p sides. As in Chapter 4, to conclude $N = N'$, it is left to prove that N and N' have the same leaves on each side and that the leaves on each side of G_N are in the same order in N' as they are in N. In this section, we will prove the first. N on X is binary, simple level-k, with $|X| \geq p + 1$ and has leaves on exactly p sides. Let $x_1, \ldots, x_p$ be p leaves of N on sides $X_1, \ldots, X_p$ respectively, such that $X_i \neq X_j$ for all $i, j \in \{1, \ldots, p\}, i \neq j$. Let P be the p-net on $x_1, \ldots, x_p$ exhibited by N. Then P is a crucial p-net by Lemma 12, so leaves $x_1, \ldots, x_p$ are also on sides $X_1, \ldots, X_p$ in P, respectively.

Since $|X| \geq p + 1$, there exists at least one side in G_N containing two or more leaves in N. Let X_i be such a side for $i \in \{1, \ldots, p\}$. Then there exists a leaf s on side X_i such that $s \neq x_i$. Let P_1 be the $(p+1)$-net on $\{x_1, \ldots, x_p, s\}$. P is crucial and so is P_1. Then, by Lemma 4, P_1 is simple level-k and G_N is also the underlying generator of P_1. Furthermore, it follows that leaves $x_1, \ldots, x_p, s$ are on sides $X_1, \ldots, X_p, X_i$ in P_1, respectively. Let P_1' be the $(p+1)$-net on $x_1, \ldots, x_p, s$ exhibited by N'. Since $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$, P_1' equals P_1 and since P_1 is simple level-k, P_1' is also a simple level-k network. By Lemma 4, P_1' is a crucial $(p+1)$-net of N'. Moreover, $G_{N'}$ is the underlying generator of P_1'. To summarize, P_1' equals P_1, G_N is the underlying generator of both N and P_1, $G_{N'}$ is the underlying generator of both N' and P_1', and G_N and $G_{N'}$ are isomorphic generators.

We will now show that leaves stay together on a side. In the proof of Lemma 10, we proved for generators with no symmetry that two leaves stay together on a side and two leaves stay on opposite sides of parallel arcs in N and in N'. To prove Theorem 6 we prove this result for the more general case. We will now prove that the leaves in a set $S \subset X$ are together on one side in N if and only if these leaves are together on one side in N'.

We begin with the 'only if' direction. First note that N' has also leaves on exactly p sides of $G_{N'}$. Let $x_i \in \{x_1, \ldots, x_p\}$ and suppose that x_i, s are together on side X_i in N, with s as defined before. Note that X_i is then an arc in $E(G_N)$. Suppose without loss of generality that s is below x_i in N and let u and v be the parents of x_i and s respectively. We may assume that (u, v) is an arc in N. Then, (u, v) is an arc of P_1. Note that x_i and s can also be on a side which is an arc in pair of parallel arcs.

We mostly follow the proof of Lemma 10. Observe that (u, v), (u, x_i) and (v, s) are arcs of P_1 and u and v are both indegree-1 outdegree-2. Since $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$, P_1 and P_1' are equal networks, so there exists a bijective function $f : V(P_1) \to V(P_1')$ such that $f(x) = x$ for each leaf $x \in \{x_1, \ldots, x_p, s\}$ and such that for every $a, b \in V(P_1)$ it holds that (a, b) is an arc of P_1 if and only if $(f(a), f(b))$ is an arc of P_1'. So $(f(u), f(v))$, $(f(u), f(x_i))$ and $(f(v), f(s))$ are arcs of P_1' and $f(x_i) = x_i$ and $f(s) = s$. It follows that $f(u)$ and $f(v)$ are the parents of x_i and s in P_1' respectively. Together with the fact that $(f(u), f(v))$ is an arc of P_1' and that $f(v)$ must be outdegree-2 in P_1' gives us that x_i and s are together on one side in N'.

The proof of the 'if' direction works in exactly the same way. As a consequence, by the fact that x_i and s are chosen arbitrarily with the requirement that (u, v) is an arc in N, for a set of leaves $S \subseteq X$ it holds that all leaves in S are together on a unique side in N if and only if all leaves in S are together on a unique side in N'. Observe that it also holds that two leaves are on different sides in N if and only if two leaves are on different sides in N'.

We will now prove that leaves are individually on the same sides. Given that $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$, it follows that $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$ by Lemma 5, where c is the size of a set of crucial sides of G_N. Indeed, $c \leq p$ because each side in a set of crucial sides of G_N contains a leaf in N. Then, by Lemma 10, if G_N has no symmetry besides parallel arcs, then all leaves in N are on the same sides in N'.

From now on, we suppose that G_N has symmetry besides parallel arcs. If G_N has symmetry, then $x_1, \ldots, x_p, s$ are on the same sides in N' as in N, after possibly relabelling sides in $G_{N'}$. We will prove that there always exists such a relabelling. While G_N has symmetry, there is at least one pair of parallel arcs or at least one automorphism $f : V(G_N) \to V(G_N)$ such that $f(v) \neq v$ for at least one $v \in V(G_N)$. That is, there exists at least one relabelling of sides $g : A \to A$, where A is the set of sides in G_N, giving a generator isomorphic to G_N. This leads to the possibility for a number of leaves of $x_1, \ldots, x_p, s$ to be on a different side in N' than in N, while G_N and $G_{N'}$ are isomorphic and P_1 equals P_1'. We know that if a set of leaves is together on one side in N, then these leaves are together on one side in N'. Therefore it is sufficient to look at P_1, containing at least one leaf on each of the p sides containing leaves in N.

If leaves are on different sides in N' as in N and only a strict subset of these sides are arcs that are in a pair of parallel arcs, we come to a relabelling in the following way, which holds for every choice of s. By definition of equal networks, there is a graph isomorphism between P_1 and P_1' that preserves leaf labels, that is, there exists a bijective function $f : V(P_1) \to V(P_1')$ such that $f(x) = x$ for each leaf x of P_1 and such that for every $u, v \in V(P_1)$ it holds that (u, v) is an arc of P_1 if and only if $(f(u), f(v))$ is an arc of P_1'. Let f be such a function. If there is at least one leaf in N' not on the same side as in N which is not a side in a pair of parallel arcs, then there is at least one leaf in P_1' not on the same side as in P_1, because G_N is the generator of both N and P_1, and $G_{N'}$ is the generator of both N' and P_1'. Also note that P_1 and P_1' are also crucial. Since P_1 equals P_1', it follows that $f(v) \neq v$ for some $v \in V(P_1)$. Furthermore, $f(x) = x$ for each leaf x of P_1 and for every $u, v \in V(P_1)$ it holds that (u, v) is an arc of P_1 if and only if $(f(u), f(v))$ is an arc of P_1', so v is not a leaf and not a parent of a leaf in P_1. Then v is also a vertex of G_N by construction. Applying f to the vertex set of G_N gives the generator isomorphic to G_N, which must be the generator of P_1', which is $G_{N'}$. Indeed, f exists because G_N has symmetry other then a pair of parallel arcs, and f is the automorphism $f : V(G_N) \to V(G_N)$ such that $f(v) \neq v$ for at least one $v \in V(G_N)$. The isomorphism $f' : V(G_N) \to V(G_{N'})$ such that $f'(y) = f(y)\ \forall y \in V(G_N)$ represents the relabelling of sides resulting in another side in N' for at least one leaf of N. f' is bijective, so $(f')^{-1}$ exists. Applying $(f')^{-1}$ to the vertex set of $G_{N'}$ gives now the relabelling of sides that belongs to f^{-1} giving for each leaf in N' that it is on the same side of the underlying generator as in N. Together with the result that leaves stay together on a side, it follows for all leaves that they are on the same sides in N and N'.

We see that considering all sides of G_N that contain leaves in N is necessary to determine the sides of noncrucial leaves. In Section 6.1, we did the same for c crucial leaves on c crucial sides, where no more than c crucial sides (except for parallel arcs) were possibilities for the leaves to be on. That is comparable to this case, where we considered p leaves on p sides (and $p + 1$ leaves to prove that leaves stay together on a side), where no more than p sides were possibilities for the leaves to be on.

5.1.4. The order of the leaves

N and N' are both binary, simple networks on X. Since $p \geq 2$ and $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$, we know that $|X| \geq 3$ and $Tn(N) = Tn(N')$ by Lemma 5. Furthermore, G_N and $G_{N'}$ are isomorphic and each leaf of X is on the same side in N' as it is in N. By Lemma 11, it follows that the order of the leaves on each side in N' is the same as in N.

5.1.5. Conclusion

We assumed, for the sake of contradiction, that $\mathcal{C}$ is not weakly encoded by $(p + 1)$-nets with the consequence that there exist two binary, simple level-k networks N and N' on X with $|X| \geq p + 1$ with leaves on at most p sides such that $N \neq N'$ and $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$. We now have proved, under the assumption that N and N' have leaves on exactly p sides, that G_N and $G_{N'}$ are isomorphic generators, that each leaf of N is on the same side in N', that two leaves stay together on a side, and that the order of leaves on each side is the same in N' as in N. By Lemma 5, if N and N' have leaves on q sides with $2 \leq q \leq p$, these results still hold. These conditions result in the fact that N and N' must be equal, that is $N = N'$ which is a contradiction. We can conclude that the class of binary, simple level-k networks with at least $p+1$ leaves and leaves on at most p sides of their underlying generator is weakly encoded by $(p + 1)$-nets for $p \geq 2$. □

Corollary 5. *Let $p \geq 2$ and let $\mathcal{C}$ be the class of binary, recoverable level-k networks on at least $p + 1$ leaves such that for all networks N in $\mathcal{C}$ the restriction N_B to any biconnected component B has leaves on at most p sides of its underlying generator. Then, $\mathcal{C}$ is weakly encoded by $(p + 1)$-nets.*

Proof. The proof follows from Theorem 6, Corollary 1, Theorem 3 and the fact that level-1 networks are encoded by their trinets (Huber and Moulton, 2013). □

5.2. Strongly encoded by $(p + 1)$-nets if $p > k$

It turns out that if $p > k$ in the previous section, then we are able to prove for N and N' being networks as in Section 5.1 that they are both binary and simple level-k. Furthermore, N and N' have both leaves on at most p sides, too. We can now prove that a different set of networks is strongly encoded by their sets of $(p + 1)$-nets.

Theorem 7. *Let $p > k$. Every binary, simple level-k network N on X with $|X| \geq p + 1$ with leaves on at most p sides of its underlying generator is encoded by its set of $(p + 1)$-nets.*

Proof. Let $\mathcal{C}$ be the class of binary, simple level-k networks with at least $p + 1$ leaves and leaves on at most p sides of their underlying generator. By Theorem 6, $\mathcal{C}$ is weakly encoded by $(p + 1)$-nets for $p \geq 2$. We will show that if N is in $\mathcal{C}$, $p > k$, and $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$ for some network N', then N' is contained in $\mathcal{C}$. Namely, if $p > k$, we can prove for N' that it is binary, simple and level-k with leaves on at most p sides of its underlying generator. Thereafter, we can use Theorem 6 to prove the lemma.

Let N be a binary, simple level-k network on X with $|X| \geq p + 1$ with leaves on at most p sides of its underlying generator G_N for $p > k$. Suppose, for the sake of contradiction, that N is not encoded by its set of $(p + 1)$-nets. Then there exists a recoverable network N' such that $N' \neq N$ and $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$. We will deduce a contradiction by showing that $N' = N$.

5.2.1. N' is a binary, simple level-k network

First observe that N and N' are networks on X with $|X| \geq p+1$ and $p \geq 2$. Furthermore, $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$, so $Tn(N) = Tn(N')$ by Lemma 5. It follows that N' is a binary network by Lemma 6. Second, observe again that N and N' are networks on X with $|X| \geq p+1$ and $p \geq 2$. Furthermore, $Tn(N) = Tn(N')$. It follows that N' is a simple network by Lemma 7. Third, observe now that N and N' are binary, simple networks on X with $|X| \geq p+1$. N has leaves on at most p sides. $p > k$, so $|X| > k+1$. $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$, so $\mathcal{S}_{k+1}(N) = \mathcal{S}_{k+1}(N')$ by Lemma 5. It follows that N' is level-k by Lemma 8. Note that if $p \leq k$, there are leaves on at most k sides of G_N. We cannot construct a $(k+1)$-net as in the proof of Lemma 8 and we cannot conclude that N' is level-k. Furthermore, we cannot prove that N is strongly encoded by its set of $(p+1)$-nets.

5.2.2. N' has leaves on at most p sides

N and N' are binary, simple level-k networks on X with $|X| \geq p+1$ and $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$. N has leaves on p sides of G_N. Let $G_{N'}$ be the underlying generator of N'. Suppose, for the sake of contradiction, that N' has leaves on at least $p+1$ sides of $G_{N'}$. Let $x_1, \ldots, x_{p+1}$ be leaves on $p+1$ different sides in $G_{N'}$ and let $P'_1 \in \mathcal{S}_{p+1}(N')$ the $(p+1)$-net on $\{x_1, \ldots, x_{p+1}\}$. Then P'_1 is a crucial $(p+1)$-net by Lemma 12, and P'_1 is level-k by lemma 4. Let P_1 be the $(p+1)$-net on $\{x_1, \ldots, x_{p+1}\}$ exhibited by N. It follows that $P_1 = P'_1$ since $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$. Then P_1 is simple level-k and so it is a crucial $(p+1)$-net by Lemma 4. Then, N has leaves on p sides of G_N and P_1 and P'_1 are crucial. It follows that there exists a pair of leaves $\{x_i, x_j\} \subset \{x_1, \ldots, x_{p+1}\}$ that are together on one side in P_1 and on different sides in P'_1. We will deduce a contradiction.

Since, $P_1 = P'_1$, there exists a bijective function $f : V(P_1) \to V(P'_1)$ such that $f(x) = x$ for each leaf x of P_1 and such that for every $u, v \in V(P_1)$ holds that (u, v) is an arc of P_1 if and only if $(f(u), f(v))$ is an arc of P'_1. Let u_i and u_j be the parents of x_i and x_j in P_1, respectively and suppose without loss of generality that x_j is below x_i in P_1 and that (u_i, u_j) is an arc of P_1. We can suppose the latter without loss of generality too, because every pair of leaves from $\{x_1, \ldots, x_{p+1}\}$ are on different sides in P'_1. Then $(f(u_i), f(u_j))$ is an arc of P'_1 and since $f(x_i) = x_i$ and $f(x_j) = x_j$, it holds that $(f(u_i), x_i)$ and $(f(u_j), x_j)$ are also arcs of P'_1. u_i and u_j are indegree-1 outdegree-2 in P_1, so $f(u_i)$ and $f(u_j)$ are also indegree-1 outdegree-2 in P'_1. It follows that x_i and x_j are on the same side in P'_1 and so in N', since P'_1 is crucial. This contradicts with $x_1, \ldots, x_{p+1}$ being on $p+1$ different sides in $G_{N'}$. So N' cannot have leaves on at least $p+1$ sides. To conclude, N' has leaves on at most p sides of $G_{N'}$.

5.2.3. Conclusion

Given that N is in class $\mathcal{C}$, we have now proved that if $p > k$ and $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$, then N' is in class $\mathcal{C}$. By Theorem 6, $\mathcal{C}$ is weakly encoded by its set of $(p+1)$-nets for $p \geq 2$. It now follows that every binary, simple level-k network N on X with $|X| \geq p+1$ with leaves on at most p sides of its underlying generator is encoded by its set of $(p+1)$-nets for $p > k$. □

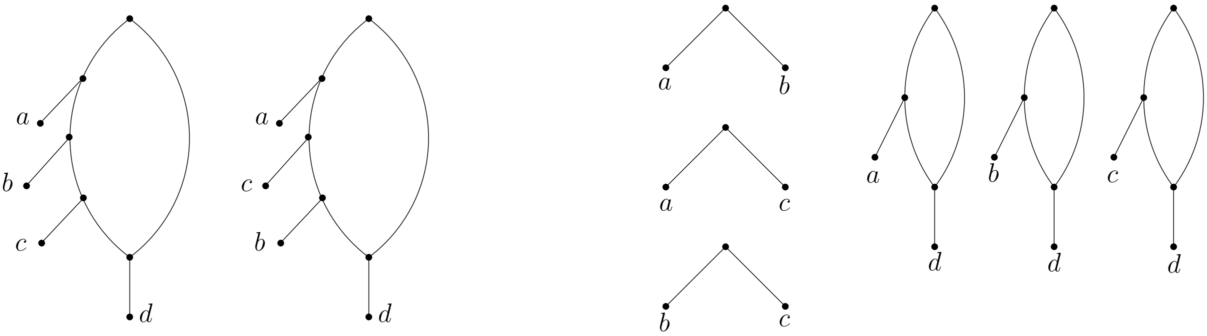
Corollary 6. *Let $p > k$. Every binary, recoverable level-k network N on X with $|X| \geq p+1$ such that the restriction N_B to any biconnected component B has leaves on at most p sides of its underlying generator is encoded by its set of $(p+1)$-nets.*

Proof. The proof follows from Theorem 7, Corollary 1, Theorem 3 and the fact that level-1 networks are encoded by their trinets (Huber and Moulton, 2013). □

5.3. Weakly encoded by p-nets if $p > 2$

Figure 5.1 shows two different networks N, N' with the same set of binets. We see that the order of the leaves is different, but all leaves are on the same side in N and N'. The networks are binary, simple level-1, and have leaves on two sides of their underlying generator. So it turns out that simple level-k networks with leaves on p sides of their underlying generator are not encoded by their set of p-nets. For this counterexample, the binets do not display the order of the leaves on a certain side. In this section, we will prove that if $p > 2$, level-k networks with leaves on p sides are weakly encoded by p-nets.

We will now prove the following theorem which is a strengthening of Theorem 6 and from which Corollary 7 will follow.



(a) Simple level-1 networks N and N', with $N \neq N'$.

(b) The set of six binets of N and N'.

Figure 5.1: Two unequal simple level-1 networks with leaves on two sides of their underlying generator, having the same set of binets.

Theorem 8. *Let $p > 2$. The class of binary, simple level-k networks with at least p leaves with leaves on at most p sides of their underlying generator is weakly encoded by p-nets.*

Proof. We will largely follow the proof in Section 5.1. Let $p > 2$ and let $\mathcal{C}$ be the class of binary, simple level-k networks with at least p leaves and leaves on at most p sides of their underlying generator. Assume, for the sake of contradiction, that $\mathcal{C}$ is not weakly encoded by p-nets. Then, there are two binary, simple level-k networks N and N' on X with $|X| \geq p$ with leaves on at most p sides such that $N \neq N'$ and $\mathcal{S}_p(N) = \mathcal{S}_p(N')$. We will show that $N = N'$, which is a contradiction, so then the lemma follows.

5.3.1. Leaves on the same number of sides and isomorphic generators

Let G_N and $G_{N'}$ be the underlying generators of N and N' respectively. Suppose N has leaves on q sides of G_N and N' has leaves on q' sides of $G_{N'}$, with $q, q' \leq p$. $\mathcal{S}_p(N) = \mathcal{S}_p(N')$, so by Lemma 13 it holds that $q = q'$. We can conclude that N and N' have leaves on the same number of sides of their underlying generator.

We may now assume without loss of generality that N and N' have leaves on exactly p sides of its underlying generator. $\mathcal{S}_q(N) = \mathcal{S}_q(N')$ holds for $2 \leq q \leq p$, and we will see that therefore the proof also holds if N and N' have leaves on q sides of their underlying generator for $3 \leq q \leq p-1$. By following exactly the same reasoning in Section 5.1.2, N and N' have isomorphic underlying generators.

5.3.2. Leaves stay together on a side

We will prove that two leaves are together on one side in N if and only if they are together on one side in N'. To prove this, we will use parts of the proof of Theorem 6, but we cannot use that $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$ since we only assumed $\mathcal{S}_p(N) = \mathcal{S}_p(N')$. Let $x_i, x_j \in X$ be leaves such that they are together on one side in N, and suppose, for the sake of contradiction, that they are on two different sides in N'. Let $P' \in \mathcal{S}_p(N')$ the p-net on $\{x_1, \ldots, x_i, x_j, \ldots, x_p\}$ such that leaves x_α and x_β are on different sides for all $\alpha, \beta \in \{1, \ldots, p\}$, $\alpha \neq \beta$. These leaves exist since $|X| \geq p$, N' contains leaves on exactly p sides, and x_i and x_j are on two different sides in N'. Then, P' is crucial by Lemma 12.

Let P be the p-net on $x_1, \ldots, x_i, x_j, \ldots, x_p$ exhibited by N. Leaves $x_1, \ldots, x_i, x_j, \ldots, x_p$ are on q different sides of G_N for $q < p$, because at least two of these leaves, namely x_i and x_j, are together on one side. $\mathcal{S}_p(N) = \mathcal{S}_p(N')$, so $P = P'$, implying that there exists a bijective function $f : V(P) \rightarrow V(P')$ such that $f(x) = x$ for each leaf x of P and such that for every $u, v \in V(P)$ it holds that (u, v) is an arc of P if and only if $(f(u), f(v))$ is an arc of P'. Note P' is simple level-k by Lemma 4, so P is simple level-k. Then, also by Lemma 4, P is a crucial p-net of N.

Let u_i and u_j be the parents of x_i and x_j in P, respectively. P is crucial, so suppose without loss of generality that x_j is below x_i in P and that (u_i, u_j) is an arc of P. We can suppose the latter without loss of generality too, because every pair of leaves from $\{x_1, \ldots, x_p\}$ are on different sides in P'. Then, $(f(u_i), f(u_j))$ is an arc of P' and since $f(x_i) = x_i$ and $f(x_j) = x_j$, it holds that $(f(u_i), x_i)$ and $(f(u_j), x_j)$ are also arcs of P'. u_i and u_j are indegree-1 outdegree-2 in P, so $f(u_i)$ and $f(u_j)$ are also indegree-1 outdegree-2 in P'. It follows that x_i and x_j are on the same side in P' and so in N', since P' is crucial. This contradicts with $x_1, \ldots, x_p$ being on p different sides of $G_{N'}$.

We can conclude that if two leaves are together on the same side in N, then they are together on the same side in N'. In exactly the same way, if two leaves are together on the same side in N', then they are together on the same side in N, so the result follows. As a consequence, if for a set of leaves $S \subseteq X$ it holds that all leaves in S are together on a unique side in N, then all leaves in S are together on a unique side in N'.

5.3.3. The same leaves on the same sides

Let $x_1, \ldots, x_p$ be p leaves of N on sides $X_1, \ldots, X_p$ respectively, such that $X_i \neq X_j$ for all $i, j \in \{1, \ldots, p\}, i \neq j$. Let P be the p-net on $\{x_1, \ldots, x_p\}$ exhibited by N. Then P is a crucial p-net by Lemma 12, so leaves $x_1, \ldots, x_p$ are also on sides $X_1, \ldots, X_p$ in P, respectively. By Lemma 4, P is simple level-k and G_N is also the underlying generator of P. Let P' be the p-net on $\{x_1, \ldots, x_p\}$ exhibited by N'. Since $\mathcal{S}_p(N) = \mathcal{S}_p(N')$, P' equals P and therefore P' is also a simple level-k network. By Lemma 4, $P' \in \mathcal{S}_p(N')$ is a crucial p-net. Moreover, $G_{N'}$ is the underlying generator of P'.

Furthermore, from the previous subsection, if $\mathcal{S}_p(N) = \mathcal{S}_p(N')$ and if for a set of leaves $S \subseteq X$ it holds that all leaves in S are together on a unique side in N, then all leaves in S are together on a unique side in N'. Since this holds as well while the sets of p-nets are the same, it follows by the proof in Section 5.1.3 that the leaves of N are on the same sides in N', after possibly relabelling sides of $G_{N'}$.

5.3.4. The order of the leaves

N and N' are both binary, simple networks on X. Since $p > 2$ and $\mathcal{S}_p(N) = \mathcal{S}_p(N')$, we know that $|X| \geq 3$ and $Tn(N) = Tn(N')$. Furthermore, G_N and $G_{N'}$ are isomorphic and each leaf of X is on the same side in N' as it is in N. By Lemma 11, it follows that the order of the leaves on each side in N' is the same as in N.

5.3.5. Conclusion

We were able to follow mostly the proof of Theorem 6 while we assumed $\mathcal{S}_p(N) = \mathcal{S}_p(N')$ instead of $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$. However, proving that sets of leaves stay together on one side has been done in a different way. To summarize, we assumed, for the sake of contradiction, that $\mathcal{C}$ is not weakly encoded by p-nets with the consequence that there exist two binary, simple level-k networks N and N' on X with $|X| \geq p$ with leaves on at most p sides such that $N \neq N'$ and $\mathcal{S}_p(N) = \mathcal{S}_p(N')$. We now have proved, under the assumption that N and N' have leaves on exactly p sides, that G_N and $G_{N'}$ are isomorphic generators, that each leaf is individually on the same side in N' as in N, that sets of leaves stay together on one side and that the order of leaves on each side is the same in N' as in N. By Lemma 5, if N and N' have leaves on q sides with $3 \leq q \leq p$, these results still hold. These conditions result in the fact that N and N' must be equal, that is $N = N'$ which is a contradiction. We can conclude that the class of binary, simple level-k networks with at least p leaves and leaves on at most p sides of their underlying generator is weakly encoded by p-nets for $p > 2$. ☐

Corollary 7. *Let $p > 2$ and let $\mathcal{C}$ be the class of binary, recoverable level-k networks on at least p leaves such that for all networks N in $\mathcal{C}$ the restriction N_B to any biconnected component B has leaves on at most p sides of its underlying generator. Then, $\mathcal{C}$ is weakly encoded by p-nets.*

Proof. The proof follows from Theorem 8, Corollary 1, Theorem 3 and the fact that level-1 networks are encoded by their trinets (Huber and Moulton, 2013). ☐

5.4. Strongly encoded by p-nets

It turn out that if we also assume that $p > k + 1$ in the proof of Theorem 8, we can also prove that N' is binary, simple level-k and that N' has leaves on at most p sides of its underlying generator. This is comparable with the aim of Section 5.2. Then, we can prove a variant of Corollary 7, that is Corollary 8. We actually prove that N' from the previous section is in the class $\mathcal{C}$ as defined before. We will prove it in this section. As before, we will use Theorem 3, such that we can reduce the problem first to simple networks. It turns out that if we require $p > k$, as we do in Theorem 7, is not enough here.

Corollary 8. *Let $p > \max\{2, k+1\}$. Every binary, recoverable level-k network N on X with $|X| \geq p$ such that the restriction N_B to any biconnected component B has leaves on at most p sides of its underlying generator is encoded by its set of p-nets.*

Proof. We can mostly follow the proof of Theorem 7 and use Corollary 1, Theorem 3 and the fact that level-1 networks are encoded by their trinets (Huber and Moulton, 2013) to prove this corollary. However, we cannot prove that N' has leaves on at most p sides as in Section 5.2.2. We will first prove that for $p > \max\{2, k+1\}$, every binary, simple level-k network N on X with $|X| \geq p$ with leaves on at most p sides of its underlying generator is encoded by its set of p-nets.

As in the proof of Theorem 7, let $\mathcal{C}$ be the class of binary, simple level-k networks with at least p leaves and leaves on at most p sides of their underlying generator, with $p > \max\{2, k+1\}$. We will show that if N is in $\mathcal{C}$ and $\mathcal{S}_p(N) = \mathcal{S}_p(N')$ for some network N', then N' is contained in $\mathcal{C}$. Let N be a binary, simple level-k network on X with $|X| \geq p$ with leaves on at most p sides of its underlying generator G_N for $p > k+1$ and let N' be a recoverable network with $\mathcal{S}_p(N) = \mathcal{S}_p(N')$. First observe that N and N' are networks on X with $|X| \geq p$ and $p > 2$. Furthermore, $\mathcal{S}_p(N) = \mathcal{S}_p(N')$, so $Tn(N) = Tn(N')$ by Lemma 5. It follows that N' is a binary and simple network by Lemma 6 and 7, respectively. $|X| \geq p$ and $p > k+1$, so $|X| > k+1$. $\mathcal{S}_p(N) = \mathcal{S}_p(N')$, so $\mathcal{S}_{k+1}(N) = \mathcal{S}_{k+1}(N')$ by Lemma 5. It follows that N' is level-k by Lemma 8.

We will now prove that N' has leaves on at most p sides of its underlying generator, not following the proof of Theorem 7. Indeed, we cannot use that $\mathcal{S}_{p+1}(N) = \mathcal{S}_{p+1}(N')$. Suppose, for the sake of contradiction, that N' has leaves on at least $p+1$ sides of $G_{N'}$. N and N' have the same number of leaves because $\mathcal{S}_p(N) = \mathcal{S}_p(N')$. Then, there exist two leaves $y_1, y_2 \in X$ such that y_1 and y_2 are together on one side in N and on two different sides in N'. Let $\{X_1, \ldots, X_c\}$ be a set of crucial sides of G_N such that each side contains at least one leaf, and let $x_1, \ldots, x_c$ be leaves on sides $X_1, \ldots, X_c$ respectively. Let C be the $c+2$-net on $\{x_1, \ldots, x_c, y_1, y_2\}$ exhibited by N. C is crucial by the proof of Lemma 2. Let C' be the $c+2$-net on $\{x_1, \ldots, x_c, y_1, y_2\}$ exhibited by N'. $\mathcal{S}_p(N) = \mathcal{S}_p(N')$ and $p > k+1$, so $\mathcal{S}_{c+2}(N) = \mathcal{S}_{c+2}(N')$ by Lemma 5. Note that this is the reason for taking $p > k+1$ instead of $p > k$ in Theorem 7. Then, $C = C'$ and C' is crucial by Lemma 4 and has $G_{N'}$ as underlying generator. Also, C has G_N as underlying generator. Then y_1, y_2 are on the same side in C and on two different sides in C'. This contradicts with $C = C'$. Note that $y_i = x_j$ for some $i \in \{1, 2\}$ and some $j \in \{1, \ldots, c\}$, then the result follows by the same reasoning, by taking a crucial $c+1$-net. We can conclude that N' has leaves on at most p sides of $G_{N'}$.

It follows that N' is in class $\mathcal{C}$. By Theorem 8, for $p > \max\{2, k+1\}$, every binary, simple level-k network N on X with $|X| \geq p$ with leaves on at most p sides of its underlying generator is encoded by its set of p-nets. Then, the proof follows from Corollary 1, Theorem 3 and the fact that level-1 networks are encoded by their trinets (Huber and Moulton, 2013). □

6

Encoding level-4 networks with 6-nets

If we want to get stronger results for determining networks by their subnets, we can try to use smaller subnets, that is, subnets on a smaller leaf set, determining the whole network. In Chapter 5 we determine networks by p-nets, where p is relatively large. The advantage is that generators of simple networks with symmetry can also be considered. In this chapter, we want to determine level-4 networks with subnets on a leaf set that is as small as possible. We will also consider generators of simple networks with symmetry. Therefore, we have to determine what the symmetries are. Indeed, it is necessary to prove that leaves in N and N' are on the same side, as part of the proof by contradiction, as we did before in the previous chapters. We assume that the symmetries of each generator are known in advance. To clarify, we also say that 'leaves are on the same side in N and N''' if 'we may assume that leaves are on the same sides in N and N'''. This means that there exists a relabelling of the sides of $G_{N'}$, giving an isomorphic generator, such that all leaves in N' are on sides of $G_{N'}$ with the same label as their side in G_N. First, we will give some additional definitions and a useful lemma. Thereafter, we will prove that leaves on crucial sides are, after possibly relabelling sides, on the same sides in N and N', where N and N' are binary, simple level-k networks on X with isomorphic generators such that $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$, with c the size of a set of crucial sides of their underlying generators. Note that the generators of N and N' can have symmetry. So contrary to Chapter 4, we have found a useful result for simple networks with symmetry in their underlying generator. We will use this result to 'fix' crucial leaves in the proof (by contradiction) of the main theorem in this chapter. It states that binary, simple level-4 networks are encoded by their 6-nets.

Definition 6.1. Let N be a binary, simple phylogenetic network on X and G_N its underlying generator. Let $S_1, \dots, S_n$ be sides of G_N with $S_i \neq S_j \; \forall i,j \in \{1, \dots, n\}$ such that for all $i,j \in \{1,\dots,n\}$, $i \neq j$, there exists a relabelling of sides mapping S_i to S_j giving an isomorphic generator. Then, S_i is a *symmetric side* of S_j for all i,j. Furthermore, $\{S_1, \dots, S_n\}$ forms a *set of symmetric sides* if there is no bigger set with this property that contains $\{S_1, \dots, S_n\}$.

In Figure 6.1, $\{A, B\}$ forms a set of symmetric sides and A is a symmetric side of B and vice versa.

Definition 6.2. Let N be binary, simple level-k phylogenetic network with underlying generator G_N. Let A be the set of sides of G_N and let $C \subset A$ be the set of all sides that are outdegree-0 reticulations or the arcs of a pair of parallellel arcs in G_N. Let $f: A \to A$ be a relabelling of sides such that $f(c) = c \; \forall c \in C$ and $f(a) \neq a$ for some $a \in A$ giving an isomorphic generator. Then f is a *noncrucial relabelling*.

Note that a noncrucial relabelling is not mapping sides in a set of crucial sides. In the generator in Figure 6.1, a relabelling of sides of G_N is called a noncrucial relabelling if it maps for example the sides in the following way: $A \Leftrightarrow B; C \Leftrightarrow D; E \Leftrightarrow G; J \Leftrightarrow K; L \Leftrightarrow M$.
The following lemma is useful to determine the minimum number of reticulations below a certain set of vertices in a generator.

Lemma 14. *Let N be a binary, simple network and G_N its underlying generator. Let $x_1, \dots, x_l \in V(G_N)$ with $l \in \mathbb{N}$ be vertices such that there is no path from x_i to x_j for all $i,j \in \{1,\dots,l\}$, with $i \neq j$ and such that $x_1, \dots, x_l$ have outdegree at least 1. Let $a_1, \dots, a_k \in E(G_N)$ with $k \geq l$ be the outgoing arcs of $x_1, \dots, x_l$. Then, there are at least $\left\lceil \frac{1}{2}k \right\rceil$ reticulation vertices below $x_1, \dots, x_l$ in G_N.*

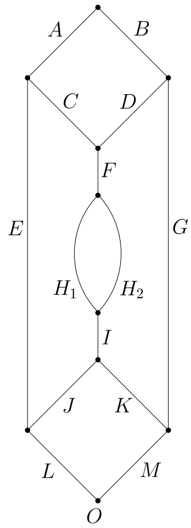


Figure 6.1: The underlying generator G_N with symmetry of some simple level-5 network N.

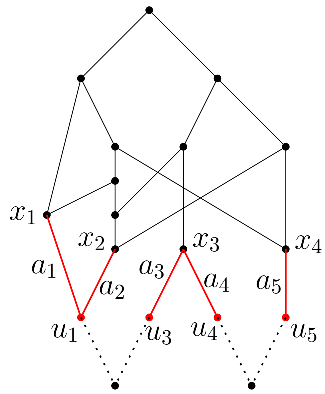


Figure 6.2: An example of a generator used in the proof of Lemma 14. Note that from the fact that $x_1, \dots, x_4$ have five outgoing arcs, it follows that there are at least five reticulations below them.

Proof. Let $a_i = (u_i, v_i)$ for $i = 1, \dots, k$, where u_i is not necessarily different from u_j, and v_i not necessarily different from v_j for $i, j \in \{1, \dots, k\}$. For each reticulation vertex in $V(G_N)$ it holds that its indegree is exactly 2 by definition and its two ingoing arcs are different from each other and different from ingoing arcs of other reticulation vertices. $V(G_N)$ is a finite set, so for each $v \in V(G_N)$ there is a path from v to an outdegree-0 vertex in G_N, so for all $i \in \{1, \dots, k\}$, there is a path from v_i to an outdegree-0 vertex in G_N. Let $P = \{p_1, \dots, p_k\}$ be the set of k different paths where p_i is a path from u_i to a reticulation vertex, such that p_i contains (u_i, v_i) and exactly one reticulation vertex. These exist by the reasoning before. Also observe that if $u_i \neq u_j$, then there is no path from u_i to u_j in G_N for all $i, j \in \{1, \dots, k\}$.

Let r be the reticulation vertex below v_i such that r is not below an other reticulation vertex below v_i. Intuitively, r is the first (and only) reticulation vertex on path p_i. Observe that p_i only contains tree-vertices and reticulation vertices with outdegree at most 1 for all $i \in \{1, \dots, k\}$. At most two paths can contain r. Furthermore, each path of P ends in a reticulation vertex, so the minimum number of reticulations below $a_1, \dots, a_k$ is $\left\lceil \frac{1}{2}k \right\rceil$. Note that a_i was chosen arbitrarily. It follows that there are at least $\frac{1}{2}k$ reticulation vertices below $x_1, \dots, x_k$ if k is even, and there are at least $\frac{k+1}{2}$ reticulation vertices below $x_1, \dots, x_k$ if k is odd. We can conclude that there are at least $\left\lceil \frac{1}{2}k \right\rceil$ reticulation vertices below $x_1, \dots, x_l$.

In Figure 6.2, $a_1, \dots, a_5$ are the outgoing vertices of $x_1, \dots, x_4$, and there is not path from x_i to x_j for some $i \neq j$. Observe that $u_1 = u_2$, and u_1 is the (first) reticulation in the path p_1 and p_2. If for example, $u_4 = u_5$ and it is therefore a reticulation (in case of the dotted lines), then there must be at least one reticulation in path p_3, unequal to u_1 or u_4. Otherwise, the generator is nonbinary or not biconnected. The dotted edges show a possible way to 'finish' the generator. □

6.1. Leaves on crucial sides

By Lemma 10, leaves are on the same sides in N and N', where N and N' are binary, simple level-k networks with isomorphic generators. However, we proved the lemma for simple networks where G_N has no symmetry besides parallel arcs. In this section, we will present a result that holds for networks with symmetry in their underlying generator. That is, we will prove that leaves on crucial sides in N are also on the same sides in N', for N and N' binary, simple level-k networks with isomorphic generators. Furthermore, we will assume $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$, with c the number of crucial sides. The advantage is that we can use a c-net on c crucial leaves, such that each leaf is on a different side in a set of crucial sides. It turns out that the only possibility for for crucial leaves in N and N' is a crucial side. That reduces the problem to the crucial sides of the networks. Considering a crucial $c+1$-net or $c+2$-net would not work for determining the sides of noncrucial leaves in the proof of the lemma below. Indeed, $c+2$ is not the total number of sides and we saw in Chapter 5 that we need in that case a p-net where

p is the number of sides containing leaves. We will now prove the following result and will use it later in this Chapter.

Lemma 15. *Let N and N' be two binary, simple level-k networks on X with isomorphic generators G_N and $G_{N'}$, respectively, such that $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$, with c the size of a set of crucial sides of G_N. Then, all leaves on crucial sides in N are on the same side in N' after possibly relabelling sides.*

Proof. Since G_N and $G_{N'}$ are isomorphic, a set of crucial sides of G_N has the same size as a set of crucial sides of $G_{N'}$ by Lemma 9. Let T be a crucial c-net exhibited by N such that T contains x. T exists by Lemma 2 and by the choice of S. T contains a leaf on each outdegree-0 vertex of G_N, and T contains exactly one leaf on one arc of a pair of parallel arcs. T is a crucial subnet, so x is on side S in T. Let T' be the c-net on the same leaf set as T exhibited by N'. $\mathcal{S}_c(N) = \mathcal{S}_c(N')$, so $T = T'$. Then T' is a simple level-k network and crucial by Lemma 4. T and T' now have the same underlying generator as N and N', respectively. So a set of crucial sides of $G_{N'}$ is of size c and T' is a crucial c-net. It follows that all leaves of T' are on a crucial side in T', and so in N'. So each leaf on a crucial side in N, is on a crucial side in N'.

6.1.1. Individual leaves

We will first prove that each crucial leaf is on the same side in N and N'. To conclude that all crucial leaves are on the same sides in N and N', we have to prove that if two leaves are together on one parallel arc in N, then they are together on one parallel arc in N', what we do in the next section. If G_N has no symmetry besides sets of parallel arcs, all leaves are on the same side in N' as in N by Lemma 10. From now on, we suppose that G_N has symmetry besides sets of parallel arcs. Then, there exists an automorphism $f : V(G_N) \to V(G_N)$ such that $f(y) \neq y$ for some $y \in V(G_N)$ giving an isomorphic generator. Let A and A' be the sets of sides of G_N and $G_{N'}$, respectively. $G_{N'}$ is isomorphic to G_N, so $V(G_N) = V(G_{N'})$ and $A = A'$. $T = T'$, so by definition of equal networks, there exists a bijective function $g : V(T) \to V(T')$ such that $g(l) = l$ for each leaf l of T and such that for every $u, v \in V(T)$ it holds that (u, v) is an arc of T if and only if $(g(u), g(v))$ is an arc of T'. Note that $V(G_N) \subset V(T)$. Since T' is crucial and G_N and $G_{N'}$ are isomorphic, there exists an isomorphism $f : V(G_N) \to V(G_{N'})$ such that $f(x) = g(x)\ \forall x \in V(G_N)$, $f(y) \neq y$ for some $y \in V(G_N)$, and it gives an isomorphic generator. Then, there exists a relabelling of sides belonging to an automorphism with the requirements of f, that can be applied to A' such that leaf x is on side S in T' and so in N' for an arbitrary leaf x on side S in T and so in N. The existence of the relabelling of sides does not depend on the choice of x. We can conclude that each leaf on a crucial side in N is on the same side in N' after a certain relabelling of sides of A'. Since we have chosen T to be a c-net, we have to prove that crucial leaves are together on one side in N which is a parallel arc, if and only if they are together on one side in N'.

6.1.2. Leaves stay together on a side

Let $x, y \in X$ be on side S in N, with S crucial. Then, S is an arc in a pair of parallel arcs in G_N. We will prove that x and y are together on one side in N'. Let C_1 be a crucial $c+1$-net exhibited by N such that it contains leaves x and y. C_1 exists by Lemma 2 since G_N has a set of crucial sides of size c. Then, x, y are on side S in C_1. Suppose without loss of generality that y is below x in N and C_1. Let C_1' be the $c+1$-net on the same leaf set as C_1 exhibited by N'. Since $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$, $C_1 = C_1'$ and C_1' is simple level-k. Then, by Lemma 4, C_1' is crucial.

$C_1 = C_1'$, so there exists a bijective function $f : V(C_1) \to V(C_1')$ such that $f(z) = z$ for each leaf z of N and such that for every $u, v \in V(C_1)$ holds that (u, v) is an arc of C_1 if and only if $(f(u), f(v))$ is an arc of C_1'. Suppose that x and y are on different sides in N'. Then, x and y are on different sides in C_1', but on one side in C_1. Let u, v be the parents of x and y in C_1 respectively. Then $(u, v), (u, x), (v, y) \in E(C_1)$. If there exists a bijective function f as mentioned, then $f(x) = x$ and $f(y) = y$. Furthermore, $(f(u), f(x)) = (f(u), x) \in E(C_1')$ and $(f(v), f(y)) = (f(v), y) \in E(C_1')$, so $f(u), f(v)$ are the parents of x and y in C_1', respectively. Also, $(f(u), f(v)) \in E(C_1')$. Since v and $f(v)$ are tree-vertices, it follows that x and y are on one side, which is an arc, in C_1'. C_1' is crucial so x and y are on one side in N'. In exactly the same way, if $x, y \in X$ are on one the same crucial side in N' then they are together on the same crucial side in N. We can conclude that two or more leaves are together on one side in N if and only if they are together on one side in N'.

6.1.3. Combining results

We have proved that individual leaves on crucial sides are on the same side in N as in N' after possibly relabelling sides by using that $\mathcal{S}_c(N) = \mathcal{S}_c(N')$. Thereafter, we have proved that leaves stay together on one side in N and N' by using that $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$. Then, for all leaves on some crucial side in N it holds that by the choice of T one leaf on this side is on the same side in N' after possibly relabelling sides, and all leaves are together on this side in N'. We can now conclude that all crucial leaves are on the same sides in N and N', which proves the lemma. □

6.2. Level-4 networks

Level-2 networks are encoded by trinets (Van Iersel and Moulton, 2012) and level-3 networks are encoded by 4-nets (Nipius, 2020). We proved that level-4 networks are encoded by 5-nets if underlying generators have no symmetry other than parallel arcs (Corollary 3). We conjecture that recoverable level-4 networks with generators with symmetry are encoded by their 5-nets, but we prove in this section that these are encoded by 6-nets. Again, we first consider simple networks. The difficulties are symmetries in the underlying generators of these. We are able to prove it by focusing on the structure and symmetries of the generators. We will first focus on the k-cycles where vertices of the generator can be mapped in. Thereafter, we will prove results depending on the number of the number of sides in a set of symmetric sides. We will now prove Theorem 9 that states the result for simple level-4 networks. Corollary 9, that follows from this theorem, states it for all recoverable level-4 networks.

Theorem 9. *Every binary, simple level-4 network N on X, with $|X| \geq 6$, is encoded by its 6-nets.*

Proof. Let N be a binary, simple level-4 network on X with $|X| \geq 6$. Suppose, for the sake of contradiction, that N is not encoded by its set of 6-nets. Then there exists a recoverable network N' such that $N' \neq N$ and $\mathcal{S}_6(N) = \mathcal{S}_6(N')$. We will deduce a contradiction by showing that $N' = N$. First observe that $\mathcal{S}_5(N) = \mathcal{S}_5(N')$ and $Tn(N) = Tn(N')$ by Lemma 5. Then, it follows from Lemma 6, 7 and 8 that N' is a binary, simple level-4 network, respectively. Indeed, these lemmas still hold if a generator has symmetry. By the same reasoning as in Section 4.1.4, the underlying generator $G_{N'}$ of N' is isomorphic to G_N. Furthermore, by Lemma 10, if G_N has no symmetry other than parallel arcs, all leaves in N are on the same sides in N'.

We will now prove that if G_N has also other symmetry than parallel arcs, then each leaf in N' is still on the same side as in N, after possibly relabelling sides. $\mathcal{S}_6(N) = \mathcal{S}_6(N')$ and G_N is level-4, so by Lemma 15, leaves on crucial sides in N are on the same sides in N'. That is, there exists some relabelling belonging to an isomorphism $f : V(G_{N'}) \to V(G_N)$, such that all crucial leaves are on the same sides in N and N', after eventually applying such a relabelling. We will now prove that this holds for all leaves on all sides in N and N'.

First, we will prove that if there is an automorphism that maps vertices in $V(G_N)$ in a k-cycle for $k > 2$, then N has only two possibilities for its underlying generator, all leaves are on the same sides in N and N', and $N = N'$. Second, we assume that automorphisms map vertices in $V(G_N)$ in k-cycles for $k \leq 2$ and prove that leaves on noncrucial sides in N are on the same sides in N' if there are more than two possible sides for that leaf to be on, that is, the set of symmetric sides that contains the side of the leaf is of size at least three. Finally, we prove the same if the side of the leaf is contained in a set of symmetric sides of size at most two.

6.3. Cycles

In this section, we will prove the following lemma, stating that if k vertices are mapped in a k-cycle for $k > 2$ under an automorphism giving a generator isomorphic to G_N, there are only two level-4 generators for which this holds. Next, we will prove that $N = N'$ for networks with one of these underlying generators, which proves Theorem 9 for specific cases. Thereafter, we can assume that if there is an automorphism mapping the vertices of G_N such that it gives an isomorphic generator, then it maps vertices in 2-cycles, that is, it only switches pairs of vertices.

Lemma 16. *Let N and N' be binary, simple level-4 networks and G_N and $G_{N'}$ its isomorphic underlying generators, respectively. Let $f : V(G_N) \to V(G_N)$ be an automorphism such that $f(x) \neq x$ for some $x \in V(G_N)$. If $\mathcal{S}_6(N) = \mathcal{S}_6(N')$ and f maps vertices in a k-cycle $x_1 \to x_2 \to \ldots \to x_k \to x_1$ for $k > 2$, then G_N is the generator as in Figure 6.3a or 6.4a.*

Proof. We will first prove that there are no generators of which the vertices can be mapped in 3-cycles by f in the following section.

6.3.1. 3-cycles

Assume without loss of generality that f maps x_1, x_2 and x_3 in the cycle $x_1 \to x_2 \to x_3 \to x_1$ with $\{x_1, x_2, x_3\} \subset X$. f is an automorphism, so for any $u, v \in V(G_N)$, u and v are adjacent in G_N if and only if $f(u)$ and $f(v)$ are adjacent in G_N. Then, x_1, x_2, x_3 must have equal indegree and outdegree and the same distance to the root. Indeed, if one of these requirements is not true, there is some arc $(u, v) \in E(G_N)$ for which $(f(u), f(v)) \notin E(G_N)$.

Suppose x_1, x_2, x_3 are outdegree-1 reticulations. If x_i is a parent of x_j for some $i, j \in \{1, 2, 3\}$, then $(x_k, x_l) \in E(G_N)$ for some $k, l \in \{1, 2, 3\}$. Then $(f(x_k), f(x_l)) \in E(G_N)$ and $(f(f(x_k)), f(f(x_l))) \in E(G_N)$. In that case, G_N contains the cycle $x_1 \to x_2 \to x_3 \to x_1$ or $x_1 \to x_3 \to x_2 \to x_1$. This contradicts with N being acyclic. So x_i is not a parent of x_j for any $i, j \in \{1, 2, 3\}$. G_N is biconnected, so the three children of x_1, x_2, x_3 cannot be cut-vertices. By Lemma 14, G_N must have at least two reticulations below x_1, x_2 and x_3. This contradicts with G_N being level-4, so x_1, x_2 and x_3 are not outdegree-1 reticulations.

Suppose x_1, x_2, x_3 are tree-vertices. By the same reasoning as before, x_i is not a parent of x_j for any $i, j \in \{1, 2, 3\}$. G_N is binary, so x_1, x_2, x_3 cannot have all three the same parent. Suppose without loss of generality that x_1 and x_2 have the same parent y and x_3 has a different parent z. $(y, x_1), (y, x_2), (z, x_3) \in E(G_N)$, $(f(y), f(x_1)) = (f(y), x_2), (f(y), f(x_2)) = (f(y), x_3)$ and $(f(z), f(x_3)) = (f(z), x_1)$. If $(f(y), x_2) \in E(G_N)$, then $f(y) = y$. If $(f(y), x_3) \in E(G_N)$, then $f(y) \neq y$. This is a contradiction, so x_i, x_j cannot have the same parent for $i, j \in \{1, 2, 3\}$, $i \neq j$. Suppose x_1, x_2, x_3 have different parents x, y, z, respectively. Then f must map x, y and z in the cycle $x \to y \to z \to x$. x, y, z cannot be outdegree-0 reticulations and by the previous paragraph, x, y, z cannot be outdegree-1 reticulations, so they are tree-vertices. They cannot be the root or a child of the root since x, y, z are different from each other. The root, x, y, z and x_1, x_2, x_3 are outdegree-2 vertices, so they have together 14 outgoing vertices, all different from each other. x, y, z and x_1, x_2, x_3 are not children of the root, so $|E(G_N)| > 14$. By Gambette et al., 2009, it holds that $|E(G_N)| \leq 14$ since G_N is level-4. We have deduced a contradiction, so x_1, x_2, x_3 cannot be tree-vertices.

Suppose x_1, x_2, x_3 are outdegree-0 reticulations. G_N is binary, so there are at least three and at most six different parents of x_1, x_2 and x_3. Let y_1, y_2 be the parents of x_1 with $y_1 \neq y_2$. Then, $(y_1, x_1), (y_2, x_1) \in E(G_N)$, $(f(y_1), x_2), (f(y_2), x_2) \in E(G_N)$, $(f(f(y_1)), x_3), (f(f(y_2)), x_3) \in E(G_N)$ and $(f(f(f(y_1))), x_1), (f(f(f(y_2))), x_1) \in E(G_N)$. If there are three different parents of x_1, x_2, x_3, then $y_1 \to f(y_1) = y_2 \to f(f(y_1)) \to y_1$ is a 3-cycle under f. These three parents cannot be tree-vertices by the proof before; they cannot be reticulations, too, since G_N is level-4. So x_1, x_2, x_3 have at least four different parents in G_N.

If there are four or five different parents of x_1, x_2 and x_3, then $(y, x_i), (y, f(x_i)) \in E(G_N)$ for some $i \in \{1, 2, 3\}$ and for some parent y of x_1, x_2, x_3, that is, there are two reticulations that share a parent. Then, $(f(y), f(x_i)), (f(y), f(x_{i+1})) \in E(G_N)$. But then, $(f(f(y)), f(f(x_i))), (f(f(y)), f(f(x_{i+1}))) \in E(G_N)$. It follows that each pair of x_1, x_2, x_3 shares a parent, so there are three different parents, which is a contradiction. So x_1, x_2, x_3 cannot have five or less different parents.

If there are six different parents of x_1, x_2 and x_3, let y_1, y_2 be the parents of x_1. Then $y_1 \to f(y_1) \to f(f(y_1)) \to y_1$ is a 3-cycle under f, or $y_1 \to f(y_1) \to f(f(y_1)) \to y_2 \to f(y_2) \to f(f(y_2)) \to y_1$ is a 6-cycle under f. Both cycles lead to a contradiction. Indeed, the vertices in both cycles cannot all be reticulations since G_N is level-4 and they cannot be tree-vertices by Lemma 14 since 12 outgoing arcs lead to at least six reticulations in this way. We can now conclude that there are no automorphisms mapping vertices of G_N in 3-cycles and giving an isomorphic generator.

6.3.2. Generators with 4-cycles

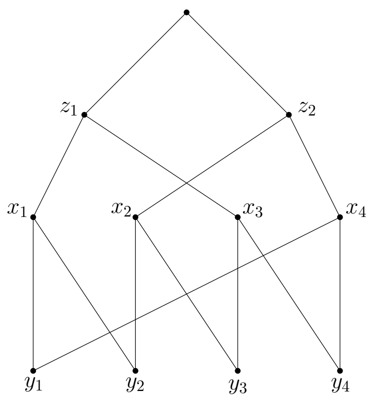
Assume without loss of generality that f maps x_1, x_2, x_3 and x_4 in the cycle $x_1 \to x_2 \to x_3 \to x_4 \to x_1$ with $\{x_1, x_2, x_3, x_4\} \subset X$. First, suppose x_1, x_2, x_3 and x_4 are all outdegree-1 reticulations. By the same reason as in Section 6.3.1, x_i is not a parent of x_j for all $i, j \in \{1, 2, 3, 4\}$ and there is no path from x_i to x_j for all $i, j \in \{1, 2, 3, 4\}$. G_N is biconnected so their children cannot be cut-vertices. By Lemma 14, it follows that G_N has at least six reticulations, which contradicts with G_N being level-4.

Suppose x_1, x_2, x_3 and x_4 are all tree-vertices. Then, they have outdegree-2 so they have eight outgoing arcs. By the same reason as in Section 6.3.1, x_i is not a parent of x_j for all $i, j \in \{1, 2, 3, 4\}$. By Lemma 14, there are at least four reticulations below x_1, x_2, x_3, x_4. Since G_N is level-4, there are exactly four reticulations below them. It follows that x_1, x_2, x_3, x_4 have four children in total, say y_1, y_2, y_3, y_4, which are all outdegree-0 reticulations, also by Lemma 14. Then, the eight outgoing arcs will be of the form (x_i, y_j) with $i, j \in \{1,2,3,4\}$ where y_j is an outdegree-0 reticulation for $j = 1, \dots, 4$.

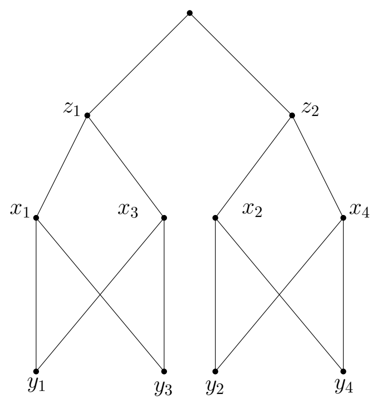
x_1, x_2, x_3 and x_4 have the same distance to the root and they cannot have the root as parent since G_N is binary. Since $|E(G_N)| \leq 14$ and x_1, x_2, x_3, x_4 are indegree-1 outdegree-2, they have a child of the root as parent, say z_1 or z_2. Suppose that $(z_1, x_1), (z_1, x_2) \in E(G_N)$. Then, $(f(z_1), x_2), (f(z_1), x_3) \in E(G_N)$. x_2 is indegree-1 so $f(z_1) = z_1$, but G_N is binary so $(z_1, x_3) \notin E(G_N)$, which is a contradiction. By a similar argument, if $(z_k, x_i) \in E(G_N)$, then $(z_k, f(x_i)) \notin E(G_N)$ for $k \in \{1,2\}, i \in \{1,2,3,4\}$. Then it holds that $(z_k, x_i) \in E(G_N)$ implies $(z_k, f(f(x_i))) \in E(G_N)$ for all $k \in \{1,2\}$, $i \in \{1,2,3,4\}$.

We will now show how x_1, x_2, x_3, x_4 are connected to their four children. Suppose that x_i and $f(x_i)$ have a child in common for some i, say y_j. Then $(x_i, y_j), (f(x_i), y_j) \in E(G_N)$, then $(f(x_i), f(y_j))$, $(f(f(x_i)), f(y_j)) \in E(G_N)$, and so on. It follows that $f(x_i)$ and $f(f(x_i))$ have a child in common. Furthermore, x_i and $f(x_i)$ have a child in common for all i. G_N is now the generator as in Figure 6.3a. Note that f maps vertices in the following 4-cycle: $y_1 \to y_2 \to y_3 \to y_4 \to y_1$.

Suppose that x_i and $f(f(x_i))$ have one child in common, then $f(x_i)$ and $f(f(f(x_i)))$ have a child in common by the same reasoning. In this case, x_i and $f(x_i)$ cannot have a child in common for some i, because x_i and $f(x_i)$ would have a child in common for all i. Then, G_N is the generator as in Figure 6.3b and is not biconnected.



(a) x_i and $f(x_i)$ have a child in common for all i.



(b) x_i and $f(f(x_i))$ have two children in common. As a consequence, this graph is not biconnected.

Figure 6.3: Directed graphs with symmetry and with four vertices that can be mapped in the 4-cycle $x_1 \to x_2 \to x_3 \to x_4 \to x_1$. The graph in (a) is a generator, but the graph in (b) is not because it is not biconnected.

Suppose x_1, x_2, x_3 and x_4 are all outdegree-0 reticulations. Their parents must be tree-vertices since G_N is level-4. If one of their parents has a child different from x_1, x_2, x_3 and x_4, then an extra reticulation is necessary to have no cut-vertices in G_N. So x_1, x_2, x_3 and x_4 have four different parents in total. Let $z_1, z_2, z_3, z_4 \in V(G_N)$ be the parents of x_1, x_2, x_3 and x_4. If $(z_j, x_i), (z_j, f(x_i)) \in E(G_N)$ for some $j, i \in \{1,2,3,4\}$, then $(f(z_j), f(x_i)), (f(z_j), f(f(x_i))) \in E(G_N)$. So if x_i and $f(x_i)$ share a parent, then $f(x_i)$ and $f(f(x_i))$ share a parent. Suppose without loss of generality that $(z_1, x_i), (z_1, f(x_i)) \in E(G_N)$, then f maps vertices z_1, z_2, z_3, z_4 in the 4-cycle $z_1 \to z_2 \to z_3 \to z_4 \to z_1$. In this case, f maps four tree-vertices in a 4-cycle. Since $|E(G_N)| \leq 14$, G_N equals the generator in Figure 6.3a.

It follows that if x_i and $f(f(x_i))$ share a parent in G_N, x_i and $f(x_i)$ do not share a parent for any i because there are exactly four reticulations in G_N. G_N is binary, so x_i and $f(f(x_i))$ must share two parents, and $f(x_i)$ and $f(f(f(x_i)))$ must share two parents, too. $|E(G_N)| \leq 14$ and x_i and x_j have the same distance from the root for all $i, j \in \{1,2,3,4\}$, so the parent of z_i is a child of the root for all $i \in \{1,2,3,4\}$. Then G_N is as in Figure 6.4a and 6.4b, which are the same. Note that the parents of x_1 and x_3 cannot share their parent, because G_N is not biconnected in that case (see Figure 6.3b). So if x_i and $f(f(x_i))$ share a parent in G_N, the generator in Figure 6.4a is the only possibility for G_N.

We can conclude that if four vertices in a symmetric level-4 generator can be mapped in a 4-cycle, giving an isomorphic generator, the generator is as in Figure 6.3a or 6.4a. We will prove in Section 6.3.4 that $N = N'$ if N has one of these generators as underlying generator.

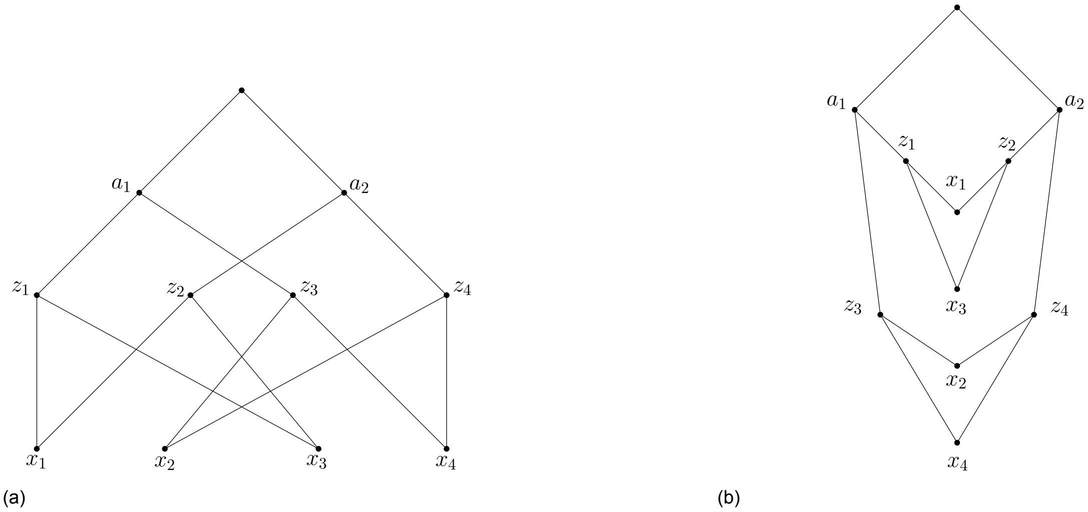


Figure 6.4: Two equal generators with symmetry. They are just drawn differently. Four vertices can be mapped in the 4-cycle $x_1 \to x_2 \to x_3 \to x_4 \to x_1$. x_i and $f(f(x_i))$ share a parent for all i.

6.3.3. k-cycles for k greater than 4

If f maps $k > 4$ vertices in a k-cycle while their in- and outdegree must be equal for each vertex, then these vertices cannot be reticulation vertices since G_N is level-4. If there are k tree-vertices with the same distance to the root, the distance is at least three, while the total number of outgoing arcs of the k tree-vertices is at least ten. There exists no such level-4 generator since $|E(G_N)| \leq 14$. So there exist no symmetric level-4 generators where five or more vertices can be mapped in a k-cycle with $k > 4$, giving an isomorphic generator. We can conclude that there are no k-cycles for $k = 3$ and $k > 4$. Furthermore, if $k = 4$, then N has a generator as in Figure 6.3a or 6.4a. This proves Lemma 16. □

We will prove in Section 6.3.4 that $N = N'$ if G_N is one of these generators. So for the rest of the proof of Theorem 9, we assume that all automorphisms $f : V(G_N) \to V(G_N)$ that give isomorphic generators for which $f(y) \neq y$ for some $y \in V(G_N)$ only map vertices in 2-cycles or 1-cycles.

6.3.4. $N = N'$ for specific generators

In this section, we will prove that $N = N'$ for networks with one of the generators from Figure 6.3a and 6.4a. Suppose that G_N is the generator in Figure 6.3a. $\mathcal{S}_6(N) = \mathcal{S}_6(N')$, so by Lemma 15, we can assume that all leaves on crucial sides are on the same sides in N' as in N, after eventually relabelling the sides of $G_{N'}$. Then, there is no isomorphism $f : V(G_N) \to V(G_{N'})$ such that $f(x_i) = x_i$ for all $i \in \{1, 2, 3, 4\}$ and $f(v) \neq v$ for some $v \in V(G_N)$.

Let $x \in X$ be on noncrucial side S in N. Let T be a crucial 6-net on x and five other leaves of N. T exists by Lemma 2 and by the fact that G_N is level-4. Let T' be the 6-net on the same leaves as T, exhibited by N'. Since $\mathcal{S}_6(N) = \mathcal{S}_6(N')$, $T = T'$ and T' is also simple level-4. By Lemma 4, T' is crucial. Since there exists no such isomorphism f as described, we can say that G_N has no other symmetry than the 4-cycle as described before. The crucial leaves are all on the same sides, $T = T'$, and T and T' have G_N and $G_{N'}$ as underlying generator, respectively. If some leaf is on a different side in T' than in T, while the crucial leaves stay on the same sides, then $T \neq T'$ because there are no other symmetries.

We have proved for individual leaves that we can assume that they are on the same sides in N and N'. We will use the following Lemma to prove that all leaves are on the same sides in N and N'.

Lemma 17. *Let N and N' be binary, simple level-k networks on X with isomorphic generators G_N and $G_{N'}$, respectively. Let $c \geq 1$ be the size of a set of crucial sides of G_N. If $|X| \geq c + 2$ and $\mathcal{S}_{c+2}(N) = \mathcal{S}_{c+2}(N')$, then a set of leaves is on one side in N if and only if they are on one side in N'.*

Proof. We will prove the lemma for two leaves and conclude that it follows that it holds for an arbitrary number of leaves on one side. We will also follow mostly the proof in Section 5.1.3. To prove the 'only if' direction, let $\{X_1, \ldots, X_c\}$ be a set of crucial sides of G_N such that each side contains at least one leaf, and let $x_1, \dots, x_c$ be leaves on sides $X_1, \dots, X_c$ respectively. Let $x, y \in X$ be two leaves that are on one side S in N. Suppose without loss of generality that y is below x in N. Note that S is an arc in G_N. Let $C \in \mathcal{S}_{c+2}(N)$ be the $c+2$-net on $\{x_1, \dots, x_c, x, y\}$ exhibited by N. Then C is a crucial $c+2$-net by the proof of Lemma 2. Note that it is possible that x or y equals x_i for some $i \in \{1, \dots, c\}$. In that case, the proof works the same since $\mathcal{S}_{c+1}(N) = \mathcal{S}_{c+1}(N')$. Let C' be the $c+2$-net on $\{x_1, \dots, x_c, x, y\}$ exhibited by N'. It follows from $\mathcal{S}_{c+2}(N) = \mathcal{S}_{c+2}(N')$ that $C = C'$ and C' is simple level-k. Furthermore, C' is a crucial $c+2$-net by Lemma 4.

Let u and v be the parents of x and y in N, respectively. We may assume that (u, v) is an arc in N. Then (u, v) is an arc in C by construction. Observe that (u, v), (u, x) and (v, y) are arcs of C and u and v are both indegree-1 outdegree-2. Since $\mathcal{S}_{c+2}(N) = \mathcal{S}_{c+2}(N')$, C and C' are equal networks, so there exist a bijective function $f : V(C) \to V(C')$ such that $f(x) = x$ for each leaf $x \in \{x_1, \dots, x_c, x, y\}$ and such that for every $a, b \in V(C)$ it holds that (a, b) is an arc of C if and only if $(f(a), f(b))$ is an arc of C'. So $(f(u), f(v))$, $(f(u), f(x))$ and $(f(v), f(y))$ are arcs of C' and $f(x) = x$ and $f(y) = y$. It follows that $f(u)$ and $f(v)$ are the parents of x and y in C' respectively. Together with the fact that $(f(u), f(v))$ is an arc of C' and that $f(v)$ must be outdegree-2 in C' gives us that x and y are together on one side in N'.

The proof of the 'if' direction works in exactly the same way. Furthermore, the lemma holds for all pairs of leaves on a certain side for which their parents form an arc in the network because we have chosen x and y arbitrary, with these requirements. Then, for a set of leaves $X' \subset X$ it holds that all leaves in X' are together on a unique side in N if and only if all leaves in X' are together on a unique side in N'. □

To conclude that all leaves are on the same sides in N and N', we remark that relabelling the sides of $G_{N'}$ to make sure that all crucial leaves occur on the same side in N' can influence the sides of noncrucial leaves. By the choice of T and T', it is necessary to apply the relabelling of sides to all sides to obtain an isomorphic generator. If G_N has no other symmetries, then the noncrucial leaves must be on the same sides in N' after this relabelling. If G_N has also other symmetry, we eventually have to apply another relabelling of sides, not mapping the crucial sides to different sides, as we will see in the next case. It follows that for simple networks N with an underlying generator as in Figure 6.3a, all leaves are on the same sides in N' as in N.

Suppose that G_N is the generator in Figure 6.4a. Let $x \in X$ be on side S in N. In the same way as for the previous case, we can assume that all leaves on crucial sides are on the same sides in N' as in N, after possibly relabelling sides of $G_{N'}$. Note that there is an isomorphism $f : V(G_N) \to V(G_{N'})$ such that $f(x_i) = x_i$ for all $i \in \{1, 2, 3, 4\}$ and $f(v) \neq v$ for some $v \in V(G_N)$. Indeed, there is one possibility for f and it switches a_1 with a_2, z_1 with z_2 and z_3 with z_4.

Let $x \in X$ be on noncrucial side S in N and let T and T' be as in the previous case. Then, a relabelling that relabels a crucial side cannot be applied to the sides of $G_{N'}$. Let g be the automorphism that switches vertices a_1 with a_2, z_1 with z_2, and z_3 with z_4 in G_N. Note that the relabelling h belonging to g does not relabel a crucial side, and this is the only symmetry that maps the outdegree-0 vertices to themselves. Note that all noncrucial sides are relabelled by h. Let y be the leaf in T on side S_2, with S_2 noncrucial. If y is on side S_2 in T', then x is on side S in T'. If x is not, we have to apply h to $G_{N'}$, relabelling S_2 to $h(S_2) = S_2' \neq S_2$ which is a contradiction. If y is on side S_2' in T', then x is on side $h(S) = S' \neq S$ in T', otherwise $T \neq T'$. In this case, we can apply h to the sides of $G_{N'}$ such that x and y are on sides S and S_2, respectively, in T' and so in N', and the crucial leaves stay on the same side.

By Lemma 17, leaves stay together on a side in N and N'. To conclude that all leaves are on the same sides in N and N', we remark that relabelling the sides of $G_{N'}$ to make sure that all crucial leaves occur on the same side in N' can influence the sides of noncrucial leaves. By the choice of T and T', it is necessary to apply the relabelling of sides to all sides to obtain an isomorphic generator. We saw in this case that after such a relabelling, noncrucial leaves can still be on different sides in N' than in N. Then we have to apply another relabelling. As long as $T = T'$, we can do this. A combination of relabellings is a composition of bijections, which is a bijection again, giving again an isomorphic generator. We can now conclude that all leaves are on the same sides in N and N' for this case, after possibly relabelling sides.

For both cases, the leaves on each side in N and N' are in the same order since $\mathcal{S}_6(N) = \mathcal{S}_6(N')$ and by Lemma 11. It follows that $N = N'$ for networks with one of the generators from Figure 6.3a and 6.4a.

6.4. Set of symmetric sides of size at least three

In the following sections we will prove that we may assume that all noncrucial leaves are on the same sides in N and N'. We will consider two cases. First we will prove this for individual leaves of N of which the side in G_N is in a set of symmetric sides of size at least three, that is, the side has itself and at least two other sides as symmetric side. In Section 6.5, we will prove this if the side of a leaf in G_N is in a set of symmetric sides of size at most two.

By Lemma 15, we can assume that all leaves on crucial sides are on the same sides in N and N'. This means the following. For example, let x be a crucial leaf on side S in N. We may assume that x is on side S in N'. Side S is an arc in a pair of parallel arcs or an outdegree-0 reticulation. It follows that a relabelling belonging to an automorphism f that maps the parent of x to a different vertex in $V(G_N)$ cannot be applied to the set of sides in $G_{N'}$ anymore, otherwise x is not on side S in N'. This means that a leaf on a noncrucial side in N cannot be on its symmetric side in N' if the symmetric side is the image under a relabelling that also maps a crucial side to a different side.

Lemma 18. *Let N and N' be two binary, simple level-4 networks on X such that $\mathcal{S}_6(N) = \mathcal{S}_6(N')$. Let G_N and $G_{N'}$ the isomorphic underlying generators of N and N', respectively. Suppose that all crucial leaves are on the same sides in N and N' after possibly relabelling sides of $G_{N'}$. If there exists a set of symmetric sides in G_N of size at least three, then we may assume that all leaves in N are on the same sides in N'.*

Proof. Let $x \in X$ be on noncrucial side S in N. Then, S is an arc in G_N which is not in a pair of parallel arcs. Let $S = (u, v) \in E(G_N)$ with $u, v \in V(G_N)$. It is given that there are at least three possibilities for sides of x in N' while $G_{N'}$ is still isomorphic to G_N. Remember, each relabelling belongs to an automorphism $f : V(G_N) \to V(G_N)$ for which $f(y) \neq y$ for some $y \in V(G_N)$. We can assume that all crucial leaves are on the same sides in N and N', so y is not an outdegree-0 vertex, not a tree-vertex of which its outgoing arcs are parallel, and not a reticulation vertex of which its ingoing arcs are parallel. Furthermore, we assume that all automorphisms $f : V(G_N) \to V(G_N)$ that give isomorphic generators for which $f(y) \neq y$ for some $y \in V(G_N)$ only map vertices in 2-cycles or 1-cycles. There exists at least two functions $g_1 : V(G_N) \to V(G_N)$ and $g_2 : V(G_N) \to V(G_N)$ such that $g_1 \neq g_2$, $g_1(u) \neq u$ and $g_2(v) \neq v$. Indeed, if there is no such function as g_1 or g_2, then side S has just one symmetric side since G_N is binary. It follows that u cannot be the root $\rho \in V(G_N)$, since $f(\rho) = \rho$ for all automorphisms f.

6.4.1. The tail of S

If u is an outdegree-1 reticulation, then the sides symmetric to S must be outgoing arcs of a reticulation, too, because u must be mapped to an outdegree-1 reticulation for all automorphisms f. Note that there cannot be a path from u' to u'', $u' \neq u''$, where u' and u'' are images of u under some automorphism f. Indeed, applying f must give an isomorphic generator. We can now use Lemma 14. If x can be on at least three different sides in N', S has at least three symmetric sides, so there are at least three outdegree-1 reticulations in G_N. Then, by Lemma 14, there are at least two reticulation vertices below S and its symmetric sides. This contradicts with G_N being level-4, so u is not an outdegree-1 reticulation.

It follows that u is a tree-vertex. By each automorphism giving an isomorphic generator, u is mapped to a tree-vertex with the same distance to the root. Let $d > 0$. u is outdegree-2, so u is in some set of tree-vertices $V' \subset V(G_N)$, $|V'| \geq 3$ for which all $v' \in V'$ have distance d to the root. Suppose without loss of generalisation $V' = \{u_1, u_2, u_3\}$. The distance to the root is then at least two. Since v has outdegree at least one, there does not exist such a generator with at least three such tree-vertices with $|E(G_N)| \leq 14$. Indeed, u_1, u_2, u_3 have in total three ingoing arcs and six outgoing arcs. The ingoing arcs of u_1, u_2, u_3 are not outgoing arcs of the root, so there are two more arcs in G_N. Furthermore, they have at least three children of u_1, u_2, u_3 that can be mapped to each other and that have outdegree at least 1. Then, $|E(G_N)| > 14$ since G_N is biconnected and binary. This is a contradiction. So u can only be mapped to at most one other tree-vertex in $V(G_N)$ by an automorphism. Therefore, S can be mapped to at most four different arcs, which are outgoing arcs of two tree-vertices u, u' that are mapped to each other by some automorphism.

6.4.2. The head of S

Crucial leaves stay on the same sides, so v cannot be outdegree-0. First, suppose that v is a tree-vertex. Then u, u' have four children in total, since v and at least two other children of u and u' must have indegree one. Moreover, all four children of u and u' must have indegree one. Indeed, S has at least three symmetric sides which are outgoing arcs of u and u'. The head of three arcs must be also indegree one. Suppose without loss of generality that the fourth arc is $(u, v') \in E(G_N)$ with v' a reticulation. Then, for the automorphism f mapping u to u' it holds that $f(v') \neq v'$ and $f(v')$ is a tree-vertex since three of the four outgoing arcs of u, u' must have a tree-vertex as head. We can now suppose that all children of u, u' are tree-vertices. The root is also a tree-vertex, so $V(G_N)$ contains seven tree-vertices and $|E(G_N)| = 14$.

S has at least three symmetric sides, so suppose that (u, v) and (u, v') are symmetric sides. We can suppose this without loss of generality because otherwise we can use that the two outgoing arcs of u' are symmetric sides. Then there is some automorphism f such that $f(u) = u$, $f(v) = v'$ and $f(v') = v$ giving an isomorphic generator. Then, v and v' are mapped to each other by f. If v and v' have two children in common, then G_N is the generator in Figure 6.5b and not biconnected. If v and v' have one child in common, then G_N is the generator as in Figure 6.5a, but we assumed that all automorphisms $f : V(G_N) \to V(G_N)$ that give isomorphic generators for which $f(y) \neq y$ for some $y \in V(G_N)$ only map vertices in 2-cycles or 1-cycles, so we have already considered this case (and proved $N = N'$ for networks with this generator) in Section 6.3.4. If v and v' have no children in common, G_N is the generator in Figure 6.4a. We have already considered this case, too.

(a) The generator where v and v' have one child in common.

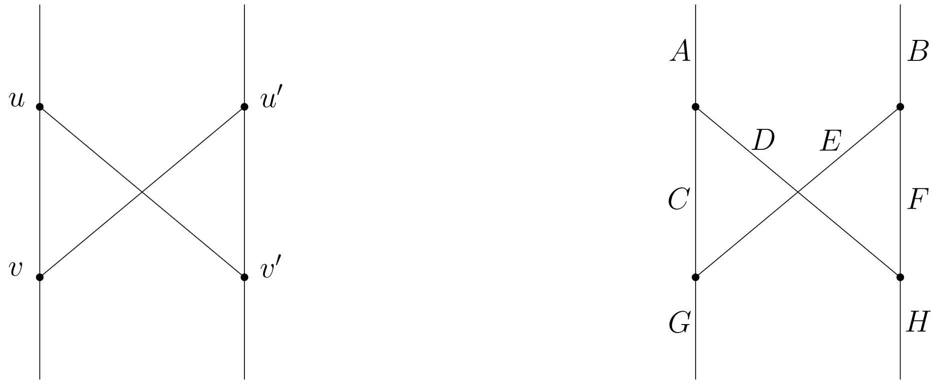
(b) The generator where v and v' have two children in common.

Figure 6.5: Two generators where u, u', and their children are tree-vertices.

Suppose v is an outdegree-1 reticulation. Then all children of u, u' must be indegree-2 outdegree-1. Indeed, S has at least three symmetric sides which are outgoing arcs of u and u'. The head of three arcs must be an outdegree-1 reticulation. Suppose without loss of generality that the fourth arc is $(u, v') \in E(G_N)$ with v' a tree-vertex or outdegree-0 reticulation. Then, for the automorphism f mapping u to u' it holds that $f(v') \neq v'$ and $f(v')$ is indegree-2 outdegree-1 since three of the four outgoing arcs of u, u' must have an outdegree-1 reticulation as head. We can now suppose that all children of u, u' are outdegree-1 reticulations. u and u' have four outgoing arcs in total, so u and u' have at least two children in total. Then u and u' have two common children. Indeed, G_N is level-4 and cannot contain three or more outdegree-1 reticulations that can be mapped to each other by an automorphism giving an isomorphic generator by Lemma 14. So if v is an outdegree-1 reticulation, then u, u' are tree-vertices with two children in common which are outdegree-1 reticulations. Then, G_N must contain a subgraph as in Figure 6.6a.

We see that S has in that case four symmetric sides: $(u, v), (u, v'), (u', v), (u', v')$. We will now prove that G_N must be one of the generators as in Figure 6.7. Let f be the automorphism such that $f(u) = u'$ and $f(u') = u$. Then, $f(v) = v$ and $f(v') = v'$. Suppose u, u' have one parent t. Then $f(t) = t$. If t is the root of G_N, then G_N contains indeed a subnetwork as in Figure 6.6a. If t is not the root of G_N, then the parent of t is the root of G_N. Indeed, G_N is level-4 and biconnected, so the outgoing arcs of v, v' result in at least one reticulation by Lemma 14, and by the same lemma, the outgoing arcs of the root not ending in t result in at least one reticulation, too. Furthermore, since there must be an automorphism mapping v and v' to each other, G_N is as in Figure 6.7a and contains also a subgraph as in Figure 6.6a.

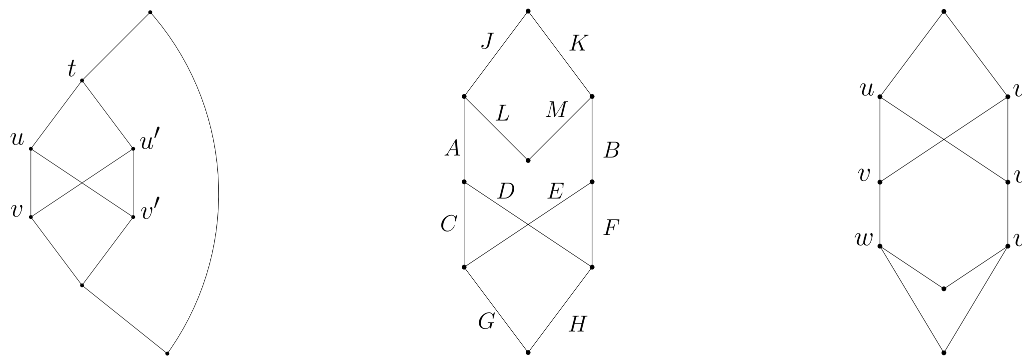
Suppose u and u' have two parents t and t', respectively. Then t and t' must be mapped to each



(a) (b)

Figure 6.6: The subnetwork that G_N must contain if there is a set of symmetric noncrucial sides of size four.

other by f, too. The outgoing arcs of v and v' result in at least one reticulation, so t and t' cannot be reticulations, so they are tree-vertices. Note that there must be an automorphism mapping v and v' to each other after fixing crucial leaves, and G_N is level-4. Then, by Lemma 14, the outgoing vertices of t and t' that have neither u nor u' as head vertices must have the same head vertex. Then G_N is as in Figure 6.7b and contains indeed a subgraph as in Figure 6.6a.



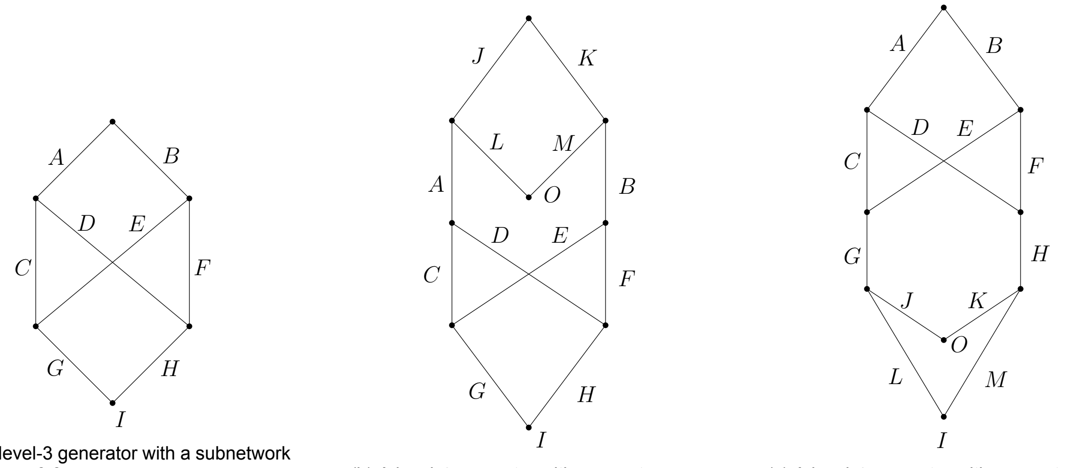
(a) (b) (c)

Figure 6.7: Three possibilities for G_N if it contains a subgraph as in Figure 6.6.

Let $g : V(G_N) \to V(G_N)$ be the automorphism such that $g(v) = v'$ and $g(v') = v$. Then, $g(u) = u$ and $g(u') = u'$. Suppose v and v' share the same child $w \in V(G_N)$, then $g(w) = w$ and G_N contains indeed a subnetwork as in Figure 6.6a. Suppose v and v' have two different children w and w' in $V(G_N)$. Then $g(w) = w'$ and $g(w') = w$. w and w' cannot be outdegree-0 since crucial leaves are fixed and they cannot be outdegree-1 by Lemma 14. So w, w' are tree-vertices and have together four outgoing arcs that lead to at least two more reticulations by Lemma 14. G_N is level-4 and biconnected, so the result that w and w' are tree-vertices leads to one possibility for G_N. Indeed, G_N is the generator in Figure 6.7c and contains a subnetwork as in Figure 6.6a.

6.4.3. The same leaves on the same sides

Suppose that N has a generator with a subnetwork as in Figure 6.6a. Let side I be an outdegree-0 vertex of G_N below v and v' and let i be the leaf on side I in N. The sides in that subnetwork are denoted as in Figure 6.6b. We assumed that $|X| \geq 6$ and $\mathcal{S}_6(N) = \mathcal{S}_6(N')$ and so $Tn(N) = Tn(N')$. Suppose that we consider the possible relabellings of the sides of G_N giving an isomorphic generator as the relabellings belonging to the automorphism that switches u with u', the automorphism that switches v with v', and the automorphism that switches u with u' and v with v'. We will now follow the proof of Lemma 5.2, 'Group 7' in Nipius, 2020 to prove that the leaves on sides C, D, E and F are on the same sides in N and N', after possibly relabelling sides. The prove that is given in Nipius, 2020 holds for the generator in Figure 6.8a, but the proof will hold for all generators in Figure 6.7, because the subnetwork is the same.



(a) The level-3 generator with a subnetwork as in Figure 6.6.

(b) A level-4 generator with symmetry.

(c) A level-4 generator with symmetry.

Figure 6.8: One level-3 and two level-4 generators containing a subnetwork as in Figure 6.6. In all generators, sides C, D, E and F form the considered subnetwork.

Observe that there is some symmetry. Sides A, C, D can be interchanged with sides B, E, F, respectively, to obtain an isomorphic generator. Also, sides C, E, G can be interchanged with sides D, F, H, respectively, again yielding an isomorphic generator. Moreover, sides A, C, D, G can be interchanged with sides B, F, E, H, respectively, again yielding an isomorphic generator. Note, a set of crucial sides has one element, namely side I. Consider the crucial trinet T exhibited by N on $\{i, x, y\}$, where i is on side I and x and y are on noncrucial sides. Let T' be the crucial trinet on $\{i, x, y\}$ exhibited by N'. This trinet exists by Lemma 2 and since $Tn(N) = Tn(N')$. Then $T = T'$. By this trinet, a leaf on side C, D, E or F in N is on side C, D, E or F in N'. Note, we use the symmetry of the generator.

We will now repeat the proof in Nipius, 2020. First, assume that there is at least one leaf on side C in N and that the leaves that are on side C in N are on side C in N'. Let c be such a leaf on side C in N.

Let a be the leaf on side A in N. This leaf is then on side A or B in N'. Let T be now the crucial trinet on $\{a, c, i\}$, which has the same underlying generator as N and N'. Then, leaves a and c are on sides that are arcs of the generator and for which holds that the end point of one of the two sides is the same as the begin point of the other sides. So, since c is on side C in N' and since $T = T'$, a is on side A in N'. In the same way, let b a leaf on side B in N and let T be the trinet on $\{b, c, i\}$. Then, leaves b and c are on sides that are arcs of the generator and for which does not hold that the end point of one of the two sides is the same as the begin point of the other side. By the same reasoning, b is on side B in N'.

Let d be a leaf on side D in N. Earlier we saw that this leaf then is on side C, D, E or F in N'. Let T be now the crucial trinet on $\{c, d, i\}$, which has the same underlying generator as N and N'. Then, leaves c and d are on sides that are arcs of the generator and that have the same begin point but different end points. So, since c is on side C in N' and since $T = T'$, d is on side D in N'. Let e be a leaf on side E in N. Earlier we saw that this leaf then is on side C, D, E or F in N'. Let T be now the crucial trinet on $\{c, e, i\}$, which has the same underlying generator as N and N'. Then, leaves c and e are on sides that are arcs of the generator and that have the same end point but different begin points. So, since c is on side C in N' and since $T = T'$, e is on side E in N'. In the same way, let f be a leaf on side F in N. Earlier we saw that this leaf then is on side C, D, E or F in N'. Let T be now the crucial trinet on $\{c, f, i\}$, which has the same underlying generator as N and N'. Then, leaves c and f are on sides that are arcs of the generator and that have different begin points and different end points. So, since c is on side C in N' and since $T = T'$, f is on side F in N'.

Let g be the leaf on side G in N. This leaf is then on side G or H in N'. Let T be now the crucial trinet on $\{g, c, i\}$, which has the same underlying generator as N and N'. Then, leaves g and c are on sides that are arcs of the generator and for which holds that the end point of one of the two sides is the same as the begin point of the other side. So, since c is on side C in N' and since $T = T'$, g is on side G in N'. In the same way, let h a leaf on side H in N and let T be the trinet on $\{h, c, i\}$. Then, leaves h

and c are on sides that are arcs of the generator and for which does not hold that the end point of one of the two sides is the same as the begin point of the other side. By the same reasoning, h is on side H in N'. We can conclude that all leaves on sides A, B, C, D, E, F, G and H are on the same sides in N' as in N. We can assume by Lemma 15 that leaf i is on side I in N and N'.

Now, assume that the leaves that are on side C are not on side C in N'. Earlier we saw that these leaves then are on side D, E or F in N'. First, if the leaves that are on side C in N are on side D in N', then we can argue in exactly the same way that the leaves that are on sides A, B, D, E, F, G and H are on sides A, B, C, F, E, H and G in N', respectively. Now, relabelling sides C, E, G with sides D, F, H, respectively, gives that all leaves are on the same sides in N' as in N.

Secondly, if the leaves that are on side C in N are on side E in N', then we can argue in exactly the same way that the leaves that are on sides A, B, D, E, F, G, H in N, are on sides A, B, C, F, E, H, G in N', respectively. Now, relabelling sides A, C, D with sides B, E, F, respectively, gives that the leaves are on the same sides in N' as in N. Thirdly, if the leaves that are on side C in N are on side F in N', then we can argue in exactly the same way that the leaves that are on sides A, B, D, E, F, G, H in N, are on sides B, A, E, D, C, H, G in N', respectively. Now, relabelling sides A, C, D, G with sides B, F, E, H, respectively, gives that the leaves are on the same sides in N' as in N.

Finally, suppose there is no leaf on side C in N and there is a leaf on one of the sides A, B, D, E, F, G, H in N. First, assume that there is a leaf on one of the sides D, E or F. Then, we can apply similar arguments based on that leaf as we did for the leaf on side C in order to get that all leaves are on the same side in N' as in N. Now, if there is no leaf on sides C, D, E and F in N, then there is a leaf on one of the sides A, B, G or H in N. Earlier we saw that if a leaf is on side A or B in N, this leaf is on side A or B in N'. Assume that the leaves that are on side A in N are on side A in N' and that the leaves that are on side B in N are on side B in N'. We can assume this without loss of generality, because if it is not the case, we can relabel sides A, C, D with sides B, E, F, respectively. Earlier we also saw that if a leaf is on side G or H in N, this leaf is on side G or H in N'. Assume that the leaves that are on side G in N are on side G in N' and that the leaves that are on side H in N are on side H in N'. We can assume this without loss of generality, because if it is not the case, we can relabel sides C, E, G with sides D, F, H, respectively. So, all leaves are on the same side in N' as in N, after possibly relabelling sides.

The proof for the generator in Figure 6.7a works in the same way as this proof, but side I is then the outdegree-0 vertex and for the noncrucial sides that are not mapped to a different side by any relabelling, it directly follows by $T = T'$ that the leaves on these sides in N must be on the same sides in N'. For the generator in Figure 6.7b, let the sides be labelled as in Figure 6.8b. Then, the proof works the same using $\mathcal{S}_4(N) = \mathcal{S}_4(N')$ if the following two changes are made. First, the relabelling that relabels sides A, C, D with B, E, F, respectively, must be replaced by the relabelling that relabels sides J, L, A, C, D with K, M, B, E, F, respectively. Second, let T be the quarnet on $\{i, o, x, y\}$ instead of $\{i, x, y\}$, where o is the leaf on side O and x, y two noncrucial leaves. Note that a leaf on side J or K in N is on side J or K in N' and a leaf on side L or M in N is on side L or M in N'. For the generator in Figure 6.7c, let the sides be labelled as in Figure 6.8c. Then, the proof works the same using $\mathcal{S}_4(N) = \mathcal{S}_4(N')$ if the following two changes are made. First, the relabelling that relabels sides C, E, G with D, F, H, respectively, must be replaced by the relabelling that relabels sides C, E, G, J, L with D, F, H, K, M, respectively. Second, let T be the quarnet on $\{i, o, x, y\}$ instead of $\{i, x, y\}$, where o is the leaf on side O and x, y two noncrucial leaves.

6.4.4. Conclusion

To conclude, if the side of leaf x is in a set of symmetric sides of size at least three, then this size is four and G_N contains a subnetwork as in Figure 6.6a or G_N is the generator in Figure 6.3a or 6.4b. We have already considered the latter. Moreover, if G_N is the generator as in Figure 6.3a or 6.4b, then $N = N'$ and the lemma follows for this case. If G_N contains a subnetwork as in Figure 6.6a, we have proved that x is on the same side in N'. By Lemma 17, all leaves are on the same sides, so the lemma follows. □

6.5. Set of symmetric sides of size at most two

We have cut the problem of proving that we may assume that all noncrucial leaves are on the same sides in N and N' in two parts. In this section, we will prove the lemma below, which is the second part.

Lemma 18 was the first part, assuming that there exists a set of symmetric sides in G_N of size at least three. Combining Lemma 18 with the following lemma gives that we may assume that all leaves on noncrucial sides are on the same sides in N and N'.

Lemma 19. *Let N and N' be two binary, simple level-4 networks on X such that $\mathcal{S}_6(N) = \mathcal{S}_6(N')$. Let G_N and $G_{N'}$ be the isomorphic underlying generators of N and N', respectively. Suppose that all crucial leaves are on the same sides in N and N' after possibly relabelling sides of $G_{N'}$. If each set of symmetric sides in G_N is of size at most two, then we may assume that all leaves of N are on the same sides in N'.*

Proof. Let $T \in \mathcal{S}_6(N)$ be a crucial 6-net exhibited by N and let $x \in X$ be a leaf on side S_1 in N, with S_1 noncrucial. Suppose that x is a leaf of T. This is possible since a set of crucial sides of G_N is of size at most four. Then, x is on side S_1 in T, since T is crucial. Let T' be the 6-net exhibited by N' on the same leaf set as T. T is simple and level-4 and $T' \in \mathcal{S}_6(N')$. Since $\mathcal{S}_6(N) = \mathcal{S}_6(N')$, T' is also simple and level-4 and therefore it is crucial by Lemma 4. We suppose that all crucial leaves are on the same sides in N and N' after possibly relabelling sides of $G_{N'}$. First, note that if S_1 is in a set of symmetric sides of size one, then x is on side S_1 in N' since $T = T'$ and they have G_N and $G_{N'}$ as underlying generators. Let A and A' be the set of sides of G_N and $G_{N'}$, respectively, let $B \subset V(G_N)$ the set of outdegree-0 vertices and head- and tail-vertices of sets of parallel arcs in G_N and let $C \subset A$ be the set of all outdegree-0 reticulations and all arcs in a pair of parallelel arcs in G_N. Note that $A = A'$ and $V(G_N) = V(G_{N'})$ since G_N and $G_{N'}$ are isomorphic. Let $f : A \to A$ be a noncrucial relabelling. This is a relabelling of sides of G_N giving an isomorphic generator, such that crucial sides keep the same label. Note that there exists an automorphism $g : V(G_N) \to V(G_N)$ such that $g(x) \neq x$ for some $x \in V(G_N)$ and $g(y) = y$ for all $y \in B$ giving an isomorphic generator such that f is the relabelling of sides belonging to g.

6.5.1. The same leaves are on the same sides if there is no symmetry left

Suppose that there does not exist such noncrucial relabelling of sides as in the previous paragraph. That is, there is no symmetry left in G_N after fixing crucial leaves. The generator in Figure 6.9a has no noncrucial relabelling. Indeed, it has symmetry, but after assuming that the leaves on sides X_1, X_2, X_3, X_4 in N are on the same sides in N', the only relabelling cannot be applied anymore. Indeed, that would change the side of the leaves on X_2 and X_4 again. Let T be a crucial 6-net such that it contains x. Then, x is on side S_1 in T. Since $\mathcal{S}_6(N) = \mathcal{S}_6(N')$, all crucial leaves are on the same sides in N and N' after possibly relabelling sides of $G_{N'}$, and since there exists no noncrucial relabelling, x is on side S_1 in T'. T' is crucial so x is on side S_1 in N'. This holds for all noncrucial leaves of N. So all leaves in N are in this case on the same side in N'.

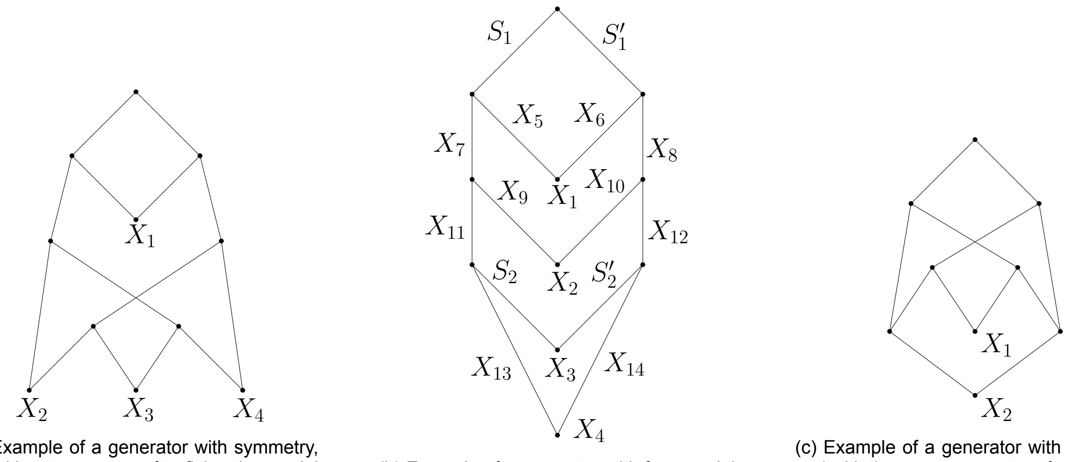
6.5.2. Unique relabelling

Suppose for the rest of the proof that there exists at least one noncrucial relabelling as defined before. If there are two noncrucial relabellings f and g such that $f \neq g$ and $f(S) = g(S) \neq S$ for some $S \in A \setminus C$, then there exists a $P \in A \setminus C$ for which $f(P) = P$ and $g(P) \neq P$. Indeed, P has at most two symmetric sides. Then, applying g to A maps S to $S' \neq S$ and maps P to $P' \neq P$ and gives an isomorphic generator. But applying f to A maps S to S' and maps P to P and gives an isomorphic generator, too. We know that $f(S') = S$. Then, $f \circ g(A)$ gives an isomorphic generator, and $f \circ g(S) = S$ and $f \circ g(P) = P'$. It follows that g must be a composition of noncrucial relabellings $g = f \circ h$ with $D = \{a \in A : f(a) \neq a\}$; $E = \{a \in A : h(a) \neq a\}$; $D \cap E = \emptyset$, that is, a composition of two functions of which the sets of sides that these two functions map to other sides are disjoint. So we can now assume that if a noncrucial relabelling f maps a noncrucial side S to $S' \neq S$, than there exists no noncrucial relabelling $g \neq f$ such that $g(S) = S'$ with $S' \neq S$ and g is not a composition of relabellings of which the sets of sides that are mapped to other sides by these functions are disjoint.

6.5.3. The same leaves are on the same sides if there is symmetry left

Suppose that x is on noncrucial side S_1 in N and x is on side S_1 or $S_1' \neq S_1$ in N'. Then, there exists a noncrucial relabelling $f : A \to A$ for which $f(S_1) = S_1'$. Let $D = \{a \in A : f(a) \neq a\}$ and let the crucial 6-net T be such that x is a leaf of T. Then x is on side S_1 in T and on side S_1' in T' since T and T' are crucial. A set of crucial sides of G_N is of size at most four. Then, T contains at least one noncrucial leaf,

say y for which $y \neq x$. So let $y \in X$ be on noncrucial side S_2 in N and let T be such that T is a 6-net on x, y and four other leaves.
An example is the generator in Figure 6.9b. It contains a set of crucial sides of size four and it holds that $D = A \setminus \{X_1, X_2, X_3, X_4\}$ because the noncrucial relabelling relabels all noncrucial sides (from the left to the right and vice versa), where D is defined as in Section 6.5.2. Also, there is symmetry left after assuming that the leaves on sides X_1, X_2, X_3, X_4 in N are on the same sides in N'. Examples of sides S_1 and S_2 are given, with their symmetric sides S_1' and S_2', respectively.



(a) Example of a generator with symmetry, but without symmetry after fixing the crucial leaves.

(b) Example of a generator with four crucial sides and a noncrucial relabelling.

(c) Example of a generator with symmetry, and with the same symmetry after fixing the crucial leaves.

Figure 6.9: Examples of level-4 generators with sets of symmetric sides of size at most two.

Suppose $S_2 \in D$, $f(S_2) = S_2' \neq S_2$ and y is on side S_2 in N'. If x is on side S_1' in N', then we have to apply the relabelling f to the sides of $G_{N'}$ such that x is on side S_1 in N', but side y will then be on side S_2' in N'. Then, x is on side S_1 in T and on side S_1 in T', and y is on side S_2 in T and on S_2' in T'. Then $T \neq T'$ which is a contradiction. Indeed, $T \neq T'$ because we assumed that there exists no noncrucial relabelling $g \neq f$ such that $g(S_2) = S_2'$ and $f(S_1) = S_1$ and g is not a composition of relabellings. So x is on side S_1 in T' and so in N'. For example, the generator in Figure 6.9c has two crucial sides, and it holds that there is symmetry left after assuming that the leaves on X_1, X_2 are on the same sides in N' as in N. However, to determine the side of x, a crucial 6-net is not necessary, but a crucial 4-net is in this case. To determine the side if the generator is as in Figure 6.9b, a crucial 6-net is necessary, since there are four crucial leaves in T.

Suppose $S_2 \in D$, $f(S_2) = S_2' \neq S_2$ and y is on side S_2' in N'. Then y is on side S_2' in T'. If x is on side S_1 in N' and so in T', then $T \neq T'$ by the same reasoning as in the previous paragraph. So x is on side S_1' in T' and so in N'. Then, we can apply the relabelling f to A' such that x is on the same side in N' as in N. The same holds for y.

Suppose $S_2 \notin D$ and so $f(S_2) = S_2$. If T contains four crucial leaves, then we can assume without loss of generality that x is on side S_1 in T' and so in N'. Indeed, T contains four crucial leaves and $S_2 \notin D$, so if x is on side S_1' in N, we can apply the relabelling f to A' such that x is on the same side in N' as in N, while the sides of y and the crucial leaves of T are not influenced by applying f. If T contains less than four crucial leaves, we can suppose without loss of generality that the sides of the noncrucial leaves of T are not in D, since leaf y was chosen arbitrarily. Then we can assume without loss of generality that x is on side S_1 in T' and so in N'. By the same reasoning as before, if x is on side S_1' in N', we can apply f to A'. Note that x and y were chosen arbitrarily on noncrucial sides. So for each pair of noncrucial sides containing leaves, where a side of these leaves can be mapped to another side by a noncrucial relabelling, the proof holds. By the assumption we made in Section 6.5.2, the applied relabelling is not a composition of relabellings of which the sets of sides that are mapped to other sides by these relabellings are disjoint. We therefore can assume that if one leaf on a side in D is on a different side in N' than in N, all leaves on sides in D are on a different side in N' than in N. So applying a relabelling such that some leaf is on the same side in N as N', gives the result that all leaves on sides in D are on the same sides in N and N'. Also, leaves stay together on one side in N and N' by Lemma 17. We have now proved Lemma 19. □

6.5.4. Combining results

We first proved Lemma 16 and proved for networks with a generator as in Figure 6.3a or 6.4a that $N = N'$. Thereafter, we proved for networks with one of the generators as in Figure 6.7 that we may assume that all leaves are on the same sides. Subsequently, we could assume that for the rest of the level-4 generators with symmetry that automorphisms map only in 1-cycles or 2-cycles, and that sets of symmetric sides are of size at most two. For these generators, we assumed that all leaves on crucial sides are on the same sides and proved that we may assume that leaves on noncrucial sides are on the same sides in N and N'. So for the last group of generators, we may also assume that all leaves are on the same sides in N and N'. Indeed, we considered the noncrucial relabellings and the relabellings that map crucial sides to different sides, eventually with other noncrucial sides in the domain. If there is no noncrucial relabelling, all leaves are on the same side, otherwise there is some crucial leaf on a wrong side. If there is at least one noncrucial relabelling, we proved uniqueness and that all leaves are still on the same sides in N and N'.

By Theorem 4, all leaves are on the same sides in N and N' if G_N has no symmetry besides parallel arcs. Combining the results gives that for all binary, simple level-4 networks N and N' on X with $|X| \geq 6$ and with isomorphic underlying generators the leaves in N are on the same sides in N'. $Tn(N) = Tn(N')$, so by Lemma 11, the order of the leaves on each side is the same in N' as in N. We can conclude that $N = N'$ which contradicts with the assumption. Therefore, we can conclude that every binary, simple level-4 network on X, with $|X| \geq 6$, is encoded by its 6-nets. □

The following Corollary is a corollary of Theorem 9.

Corollary 9. *Every binary, recoverable level-4 network on X, with $|X| \geq 6$, is encoded by its 6-nets.*

Proof. The proof follows from Theorem 9, Corollary 1, Theorem 3 and the fact that level-1 networks are encoded by their trinets (Huber and Moulton, 2013). □

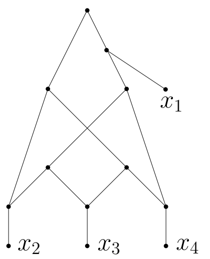
7

Algorithm for proving encoding results

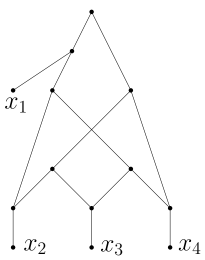
It is proved in Nipius, 2020 that most level-3 networks are encoded by their trinets. In this thesis, all level-3 generators from Gambette et al., 2009 are considered, divided into groups according to the number of sides in a set of crucial sides, the number of sets of parallel arcs, and symmetry. Extending the proof in Nipius, 2020 to higher level generators would be even much more work. Especially the part of the proof where it is proved that leaves in N are on the same side in N', where N and N' are binary, simple level-3 networks with isomorphic generators, given that $Tn(N) = Tn(N')$. Therefore, it is useful to have an algorithm that can be used to determine the side of a leaf in N' given a certain side in N. Note that for proving that binary, simple level-3 networks are encoded by trinets, it is necessary that we can assume that leaves are on a unique side in N'. In this chapter, we first analyse how the sides for a leaf in N' are determined in Nipius, 2020. Thereafter, we present an algorithm for determining the possible sides in N' for a leaf in N. Lastly, we will generalise this algorithm such that it works for simple level-k networks.

7.1. Encoding level-3 networks with trinets

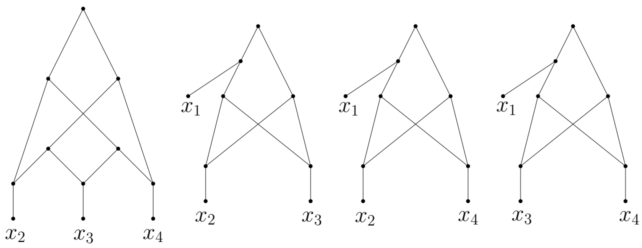
In this subsection, we analyse the proof in Nipius, 2020 of the theorem stating that the class of most binary, simple level-3 networks with at least three leaves is weakly encoded by trinets. There is one generator for which networks with this generator are not encoded by their trinets (Nipius, 2020). This counterexample is given in Figure 7.1. The binary, simple level-3 networks N and N' in Figure 7.1a and 7.1b are unequal. Indeed, the paths from the parent of x_1 to x_2 and x_4 in N have not the same length as these paths in N'. We see that the four different trinets exhibited by N are the same as the four different trinets exhibited by N' in Figure 7.1c. It follows that N and N' are not encoded by their set of trinets.



(a) Network N



(b) Network N'



(c) The set of trinets exhibited by N and N'.

Figure 7.1: Two different simple level-3 networks with the same set of trinets (Nipius, 2020).

Nipius first assumes that the class of binary simple level-3 networks is not encoded by trinets, except for networks with the underlying generator of the network in Figure 7.1a with leaves on the sides that are outgoing arcs of the root, and no leaves on other noncrucial sides. Then there are two binary, simple level-3 networks N and N', not as the exception, such that $Tn(N) = Tn(N')$. Second, in Nipius, 2020 it is proved that N and N' have isomorphic underlying generators G_N and $G_{N'}$, respectively. Then it is shown that for each of the 65 level-3 generators from Gambette et al., 2009 that if G_N and $G_{N'}$ are isomorphic, then $N = N'$. By Lemma 11, if all leaves are on the same sides in N and N', then the order of the leaves on each side is the same in N and N'. First we will prove that leaves are on the same sides, thereafter we will give an example of the proof for one certain generator. The level-2 generators will be underlying generators of noncrucial trinets exhibited by a level-3 network, and they are presented in Figure 7.2.

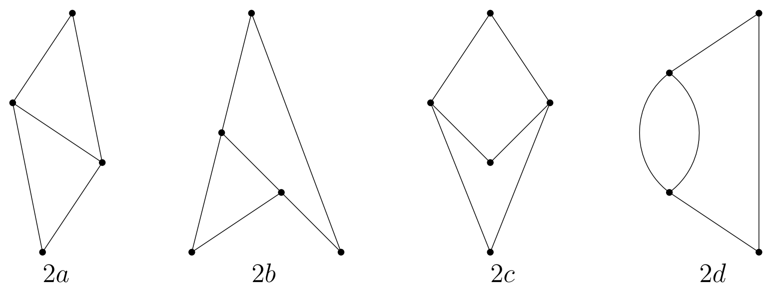


Figure 7.2: The four level-2 generators (Gambette et al., 2009).

7.1.1. Example for a specific generator

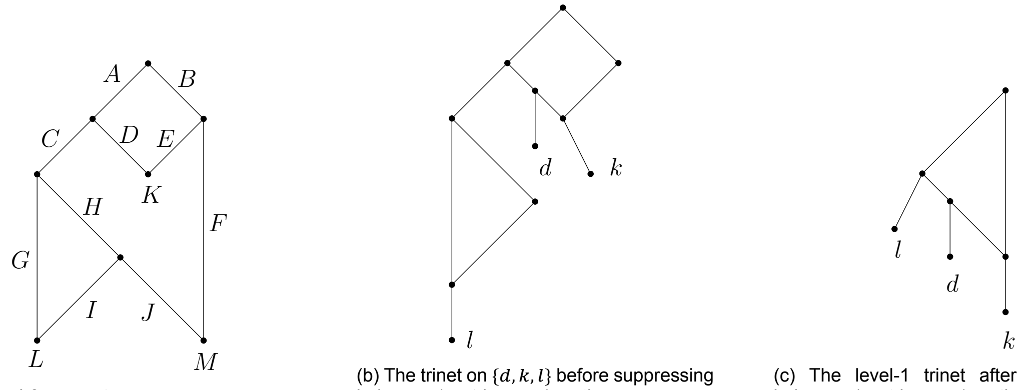
To proof that a leaf is on the same side in N' as it is in N in general, it is often useful to build up crucial k-nets, as we saw before in this thesis. For level-3 networks with one or two sides in a set of crucial sides of their underlying generator, there exists a crucial trinet containing a noncrucial leaf by Lemma 2. If a set of crucial sides is of size three, the single crucial trinet contains only crucial leaves. Then, we cannot conclude anything about the sides of noncrucial leaves in N'. Nipius assumed that the set of trinets of N and N' is the same, but the set of k-nets for $k \geq 4$ is not necessarily the same. Therefore, only trinets are used, which are not necessary crucial. For determining the possible sides for a noncrucial leaf in N', the trinet needs to be noncrucial if there is a set of crucial sides of size three. In this section, we concentrate on generators with three sides in a set of crucial sides.

Let N be a simple level-3 network with three sides in a set of crucial sides of its underlying generator. We want to construct a trinet exhibited by N on a noncrucial leaf, say x. To determine the side in N' for x, it is better to have a maximum number of crucial leaves of N in the trinet. Then, the trinet is on x and two crucial leaves. There are three choices for the sides of the crucial leaves in N. Indeed, we choose two out of three crucial sides containing leaves. Then, we obtain three different noncrucial level-2 or level-1 trinets, that determine the possibilities for the sides that x can be on in N', because $Tn(N) = Tn(N')$. We can also say that we obtain the three trinets by deleting one crucial leaf (for each trinet a different leaf).

We will consider the generator in Figure 7.3a as an example. Suppose N is a network with an underlying generator G_N as in Figure 7.3a and such that N contains a leaf on each side of G_N. Let $a, \dots, m$ be leaves on sides $A, \dots, M$ in N, respectively. Suppose that we want to determine the side of leaf d in N'. Then, as in Nipius, 2020, we let $T_{S,1}$ be the trinet on $\{s, k, l\}$, $T_{S,2}$ the trinet on $\{s, k, m\}$ and $T_{S,3}$ the trinet on $\{s, l, m\}$ for a leaf s on side S. In Figure 7.3c, $T_{D,1}$ is shown. This is a level-1 network, after suppressing indegree-1 outdegree-1 vertices in the graph in Figure 7.3b which has two nontrivial biconnected components, and they turn out to be useless (Nipius, 2020) to determine the side of a leaf in N', so we do not consider level-1 trinets.

Figure 7.4a shows $T_{D,2}$, the trinet on $\{d, k, m\}$ exhibited by N. The generator of this level-2 trinet is generator $2c$ from Figure 7.2. We see some disadvantage, namely the occurrence of symmetry in this generator, while the level-3 generator has no symmetry. Therefore, since $Tn(N) = Tn(N')$, leaf d is on side D or E in N'. Figure 7.4b shows $T_{D,3}$, the trinet on $\{d, l, m\}$ exhibited by N. The generator of this level-2 trinet is generator $2b$ from Figure 7.2. This generator has no symmetry, but sides A, C and D in the level-3 generator are merged to one arc to obtain generator $2b$. Therefore, since $Tn(N) = Tn(N')$, leaf d is on side A, C or D in N'. In Section 7.1.2, we will analyse the process of obtaining a lower level generator from a level-3 generator.

We have considered $T_{D,1}$, $T_{D,2}$ and $T_{D,3}$. $T_{D,2}$ and $T_{D,3}$ give us that leaf d is on side D or E in N', but also on side A, C or D in N'. We see that there is one side in common, namely side D. Therefore, we can conclude that leaf d is on side D in N'. Repeating this process for each noncrucial side containing



(a) A level-3 generator G_N.

(b) The trinet on $\{d, k, l\}$ before suppressing indegree-1 outdegree-1 vertices.

(c) The level-1 trinet after suppressing indegree-1 outdegree-1 vertices.

Figure 7.3: A generator G_N and the construction of a trinet on $\{d, k, l\}$ exhibited by N.

(a) The trinet on $\{d, k, m\}$ exhibited by N.

(b) The trinet on $\{d, l, m\}$ exhibited by N.

Figure 7.4: Two trinets exhibited by N, where N is the network as described in Section 7.1.1 with the underlying generator as in Figure 7.3a.

a leaf, will lead to the possibilities of the sides for all leaves in N. We don't consider leaves on crucial sides in this way, because the sides can be determined by considering a crucial trinet on three crucial leaves.

Nipius presents all possibilities for a side of a leaf for each noncrucial side in Nipius, 2020 for the three different trinets $T_{S,1}$, $T_{S,2}$ and $T_{S,3}$, where each trinet is on exactly two crucial sides. A table displaying these possibilities for networks with a generator as in Figure 7.3a is presented in Table 7.1. We see that all leaves on noncrucial sides are on the same side in N' as in N. Furthermore, Nipius proved this for leaves on crucial sides, too. Thereafter, it is concluded that if N has this generator, then $N = N'$.

7.1.2. Constructing level-2 from level-3 generators

In Gambette et al., 2009, an algorithm for generating level-k generators from level-$(k-1)$ generators is given. In this section, we will explain how a level-2 generator is constructed by deleting a crucial side from a level-3 generator. We do this because level-2 trinets exhibited by a simple level-3 network are considered in Nipius, 2020. Furthermore, it will be useful to determine the sides for a leaf, as in the previous section.

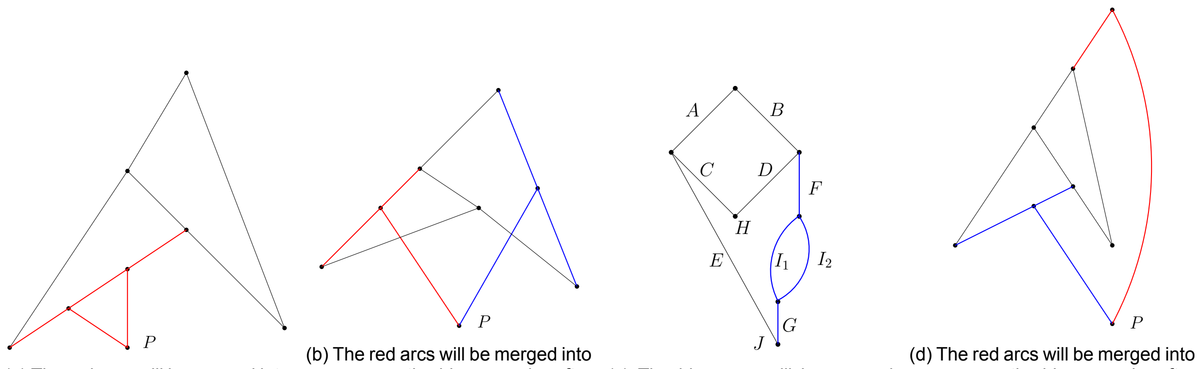
In Figure 7.5a, a level-3 generator G_N is given. It contains three outdegree-0 vertices. Let leaf s be on noncrucial side S of G_N. If we want to determine the side of s in N' as in the previous section, we have to construct a level-2 trinet. Suppose that we indeed obtain a level-2 trinet T on s and on two crucial leaves of N. We will now consider the outdegree-0 vertex, say side P, of G_N that is not contained in the level-2 generator G_T. After deleting P and suppressing indegree-1 outdegree-1 vertices, five sides of G_N (the red arcs in Figure 7.5a) are merged into one side in G_T. This is the case when the parents of P in G_N form an arc. Then, if S is one of the red sides, s is on one of the red sides in N' since $Tn(N) = Tn(N')$. If S is one of the black arcs, then s is on the same side in N' since G_T is generator $2b$

Leaf on side S in N	Trinet $T_{S,1}$	Trinet $T_{S,2}$ Generator $2c$	Trinet $T_{S,3}$ Generator $2b$	Result for N'
A		$A \vee B$	$A \vee C \vee D$	A
B		$A \vee B$	$B \vee E \vee F$	B
C		$C \vee F \vee G \vee H \vee I \vee J$	$A \vee C \vee D$	C
D		$D \vee E$	$A \vee C \vee D$	D
E		$D \vee E$	$B \vee E \vee F$	E
F		$C \vee F \vee G \vee H \vee I \vee J$	$B \vee E \vee F$	F
G		$C \vee F \vee G \vee H \vee I \vee J$	G	G
H		$C \vee F \vee G \vee H \vee I \vee J$	H	H
I		$C \vee F \vee G \vee H \vee I \vee J$	I	I
J		$C \vee F \vee G \vee H \vee I \vee J$	J	J

Table 7.1: Possibilities for a side in N' for a leaf in N, where G_N is the generator in Figure 7.3a.

and has no symmetry.

Let G_N be the generator in Figure 7.5b. After constructing a level-2 trinet T that is not on the crucial leaf on side P, the three red arcs are merged into one arc and the three blue arcs are merged into one arc to obtain G_T from G_N. If leaf s is on side S in N, and S is one of the blue arcs, then s is on one of the blue sides in N' since $Tn(N) = Tn(N')$. The same holds for the red sides. This is the case when the parents of P in G_N do not form an arc, and none of the parents is the root. Note that the level-3 generators, as the level-2 generators after deleting P in Figure 7.5a and 7.5b, do not have symmetry since level-2 generator $2b$ occurs.



(a) The red arcs will be merged into one arc after deleting P.

(b) The red arcs will be merged into one arc, as the blue arcs do, after deleting P.

(c) The blue arcs will be merged into one arc after deleting I_1 or I_2.

(d) The red arcs will be merged into one arc, as the blue arcs do, after deleting P.

Figure 7.5: Generators showing how possibilities for sides in N' occur.

Suppose that N is a network with a generator as in Figure 7.5c. This network contains a pair of parallel arcs. Let s be on noncrucial side $S \in \{A, B, C, D, E, F, G, I_1, I_2\}$ in N. Consider a trinet T on $\{s, h, j\}$, where h and j are on sides H and J in N. The generator of the level-2 trinet is generator $2c$, which has symmetry. The parallel arcs are deleted and the blue arcs are merged into one arc. We see that if S is for example side F, then s is on side E, F, I_1, I_2 or G in N' since $Tn(N) = Tn(N')$.

The fourth example is the generator in Figure 7.5d. In the same way as in the previous examples, if s is on side S in N and S is one of the red arcs of G_N, then s is on one of the red arcs in N'. The same holds for the blue arcs. Indeed, the blue arcs will be merged after deleting P in the same way as the example in Figure 7.5b. The red arcs will not be merged if s is on one of the red arcs in N, but they are deleted if the considered trinet is not on a leaf on one of the red sides. We will see this in the algorithm in the next section.

Before we present the algorithm giving the possible sides for a leaf in N', we observe some useful properties of the sides in level-3 generators. First, in Table 7.1, the sets of possible sides for a leaf in N' occur in disjoint sets for each trinet. Furthermore, for each trinet, these sets form a partition of the sides of $G_{N'}$. This will always be the case, because such set of sides is obtained in one of the following ways.

- Given a level-3 generator with a set of crucial sides of size three, and a level-2 generator obtained from the level-3 generator by deleting one outdegree-0 vertex or a pair of parallel arcs, and suppressing indegree-1 outdegree-1 vertices. If the level-2 generator has no symmetry, then the sets of sides that are merged into one side and the individual arcs that are not merged form disjoint sets. This can be seen in Table 7.1, where trinet $T_{S,3}$ has generator $2b$ as underlying generator.
- Given a level-3 generator with a set of crucial sides of size three, and a level-2 generator obtained from the level-3 generator by deleting one outdegree-0 vertex or a pair of parallel arcs, and suppressing indegree-1 outdegree-1 vertices. If the level-2 generator has symmetry, then the sets of sides that are merged into one side, together with its symmetric side in the level-2 generator, and the sets of symmetric sides with sides that are not merged form sets of disjoint sides. This can be seen in Table 7.1, where trinet $T_{S,2}$ has generator $2c$ as underlying generator.
- Let N be a simple level-3 network. If the trinet on two crucial leaves and one noncrucial leaf of N is not simple, and the parent of one leaf is the root, then the two outgoing arcs of the root of G_N form a set of possible sides in N'. The other sets are formed in the same way as described before. For example, this is the case when one leaf on the red arcs in Figure 7.5d is considered, as in Figure 7.6. The red arcs in Figure 7.5d form a set of possible sides.

Lastly, we observe that after the deletion of an outdegree-0 vertex, say P, from a level-3 generator G_N and after suppressing the indegree-1 outdegree-1 vertices to obtain a level-2 generator, the adjacent arcs of the two parents of P form a disjoint set of sides if the parents of P form an arc in the level-3 generator. For example, the red sides in Figure 7.5a form such a set. If the parents of P do not form an arc in G_N, then the adjacent arcs of each individual parent form a set of possibilities, as in Figure 7.5b happens. If an arc a in a pair of parallel arcs is deleted from G_N, then the adjacent arcs of a and a itself form a set of possibilities. These are for example the blue arcs in Figure 7.5c.

7.1.3. Algorithm for level-3 networks

We will now give Algorithm 1 which has as input a level-3 generator G_N, the underlying generator of network N, having a set of crucial sides of size three, its set of noncrucial sides, leaf s_m on noncrucial side S_m and a set of crucial sides of which each side contains a leaf in N. Level-1 trinets are not considered, as for trinet $T_{S,1}$ in Table 7.1 happens.

Algorithm 1 *DeterminingSides* determines all possible sides to be on for a leaf in N', given the side in N, for all trinets on the leaf and two crucial leaves, where N and N' are binary, simple level-3 networks with isomorphic generators and $Tn(N) = Tn(N')$.

procedure DeterminingSides(G_N)
Result: S, the set of sides where a leaf can be on in N'.
Initialisation: Let G_N be a level-3 generator with a set of crucial sides of size three. Let x_1, x_2, x_3 be the leaves in N on sides X_1, X_2, X_3, respectively, where $\{X_1, X_2, X_3\}$ form a set of crucial sides in G_N. Let $S_1, \dots, S_n$ be the noncrucial sides of G_N and let s_m be a leaf on side S_m where $m \in \{1, \dots, n\}$.
  for $m = 1$ **to** n **do**
    for all x_i, x_j in combinations of two elements of x_1, x_2, x_3 **do** Let T be the trinet on $\{s_m, x_i, x_j\}$ and G_T the underlying generator of T constructed from G_N by deleting one crucial side of $\{X_1, X_2, X_3\}$ and suppressing indegree-1 outdegree-1 vertices and deleting indegree-0 outdegree-1 vertices. Call the deleted crucial side X.
$S \leftarrow \emptyset$
      while G_T is level-2 **do**
        if G_T has no symmetry **then**
          if X is a vertex and its parents are adjacent **then** Let P_1 be the set of sides which are arcs incident to one of the parents of X in G_N
            if $S_m \in P_1$ **then** $S \leftarrow S \cup P_1$
            else $S \leftarrow S \cup S_m$
            end if
          else if X is a vertex and its parents are not adjacent **then** Let P_1 be the set of sides in G_N that are arcs incident to the same parent of X. Let P_2 the set of sides in G_N that are arcs incident to the other parent of X.
            if $S_m \in P_i$ for $i \in \{1, 2\}$ **then**
              for $i = 1$ **to** 2 **do**
                if $S_m \in P_i$ **then** $S \leftarrow S \cup P_i$
                else $S \leftarrow S$
                end if
              end for
            else $S \leftarrow S \cup S_m$
            end if
          else Let P_1 be the set of sides in G_N containing X and its adjacent arcs.
            if $S_m \in P_1$ **then** $S \leftarrow S \cup P_1$
            else $S \leftarrow S \cup S_m$
            end if
          end if

Part 2

else Let A be the set of sides of G_T and let $f : A \to A$ be a relabelling giving an isomorphic generator such that $f(X_i) = X_i$ and $f(X_j) = X_j$.
if X is a vertex and its parents are adjacent **then** Let P be the set of sides which are arcs incident to one of the parents of X in G_N. Let $P' \in G_T$ be the side which is obtained by suppression of the parents of X in G_N.
$Y \leftarrow f(P')$
$P_1 \leftarrow P \cup Y$
Let P_2 and P_3 be sets of sides in G_N such that they are images of each other under f in G_T.
for $i = 1$ **to** 3 **do**
if $S_m \in P_i$ **then** $S \leftarrow S \cup P_i$
else $S \leftarrow S$
end if
end for
else if X is a vertex and its parents are not adjacent **then** Let u and v be the parents of X. Let P and Q be the sets of sides which are arcs incident to u and v, respectively, and let P' and Q' be the sides in G_T obtained by suppression of u and v, respectively.
if $f(P') = Q'$ **then** $P_1 \leftarrow P \cup Q$ and let P_2 and P_3 be sets of sides in G_N such that they are images of each other under f in G_T such that $P_i \cap P_j = \emptyset$ for $i, j \in \{1, 2, 3\}, i \neq j$.
else $Y \leftarrow f(P')$
$Z \leftarrow f(Q')$
$P_1 \leftarrow P \cup Y$
$P_2 \leftarrow Q \cup Z$
Let P_3 be set of sides in G_N such that they are images of each other under f in G_T such that $P_i \cap P_j = \emptyset$ for $i, j \in \{1, 2, 3\}, i \neq j$.
end if
for $i = 1$ **to** 3 **do**
if $S_m \in P_i$ **then** $S \leftarrow S \cup P_i$
else $S \leftarrow S$
end if
end for
else Let P be the set of sides in G_N containing X and its adjacent arcs. Let $P' \in G_T$ be the side which is obtained by suppression of the vertices above and below X in G_N.
$Y \leftarrow f(P')$
$P_1 \leftarrow P \cup Y$
Let P_2 and P_3 be sets of sides in G_N such that they are images of each other under f in G_T such that $P_i \cap P_j = \emptyset$ for $i, j \in \{1, 2, 3\}, i \neq j$.
for $i = 1$ **to** 3 **do**
if $S_m \in P_i$ **then** $S \leftarrow S \cap P_i$
else $S \leftarrow S$
end if
end for
end if
end if
end while
return S
end for
end for
end procedure

7.2. Towards encoding level-k networks with k-nets

Algorithm 1 helps to find the possible sides of leaves in simple level-3 networks. In this section, we present a generalisation of this algorithm. By Theorem 4, every binary, simple level-k network N on X with $|X| \geq k+1$ and without symmetry besides parallel arcs in its underlying generator is encoded by its set of $(k+1)$-nets for $k \geq 2$. To know whether certain simple level-k networks are (weakly) encoded by their k-nets, a generalisation of Algorithm 1 can be useful. In general, the class of binary, simple level-k networks with at least k leaves is not weakly encoded by k-nets for $k \geq 2$. Indeed, we presented a counterexample for $k = 3$ in Section 7.1, but we do not know yet whether the statement is true for $k \geq 4$. We conjecture that, for some $k \geq 4$, most simple level-k networks are weakly encoded by their k-nets.

To prove the statement for certain networks, we can, for the sake of contradiction, consider two binary, simple level-k networks N and N' such that $N \neq N'$, with isomorphic generators and with $S_k(N) = S_k(N')$. Then, we have to prove $N = N'$ and as part of that, prove that we can assume that all leaves are on the same sides in N' as in N. For this part, we will give the generalisation of Algorithm 1, giving for a leaf on noncrucial side $S \in E(G_N)$ the set of possible sides on which it can be on in N'. The algorithm does this with a level-k generator with a set of crucial sides of size k as input, supposing that each side contains a leaf in N and it gives the possible sides for a leaf for different k-nets. Moreover, for each such generator, k different k-nets exhibited by networks with such a generator are considered, each on the leaf on the considered side and on $k-1$ crucial leaves. There are k choices of $k-1$ sides out of a set of k crucial sides. Note that in Algorithm 1 three different trinets are considered for each level-3 generator with a set of crucial sides of size three.

We present Algorithm 2 which actually works in the same way. Indeed, we consider only simple level-$(k-1)$ k-nets with a level-$(k-1)$ underlying generator. For example, let G_K be such a level-$(k-1)$ generator, the underlying generator of K, a level-$(k-1)$ k-net on a noncrucial leaf and $k-1$ crucial leaves. We choose the $k-1$ crucial leaves such that these leaves are on $k-1$ different sides in G_N, the given level-k generator. Then, G_K can be obtained from G_N by deleting a crucial side of G_N and suppressing all indegree-1 outdegree-1 and indegree-0 outdegree-1 vertices.

Note that we call G_K the underlying generator of K. There is one possibility for K for which its underlying generator is not defined according to Definition 2.23. This is when K is not simple, as in the trinet on $\{s, k, l\}$ in Figure 7.6. For this trinet, the level-3 generator G_N is the generator in Figure 7.5d and we see that its blue arcs are merged and the red arcs are not. In this case, the two outgoing arcs of the root of G_N form a set of possible sides and G_K is generator $2b$, because the indegree-0 outdegree-1 vertices are deleted from G_N after deleting a crucial side.

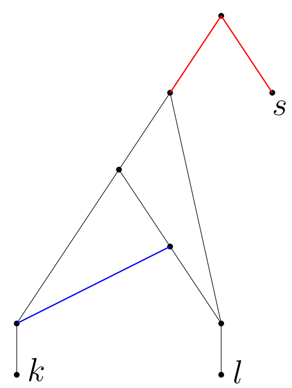


Figure 7.6: A nonsimple trinet.

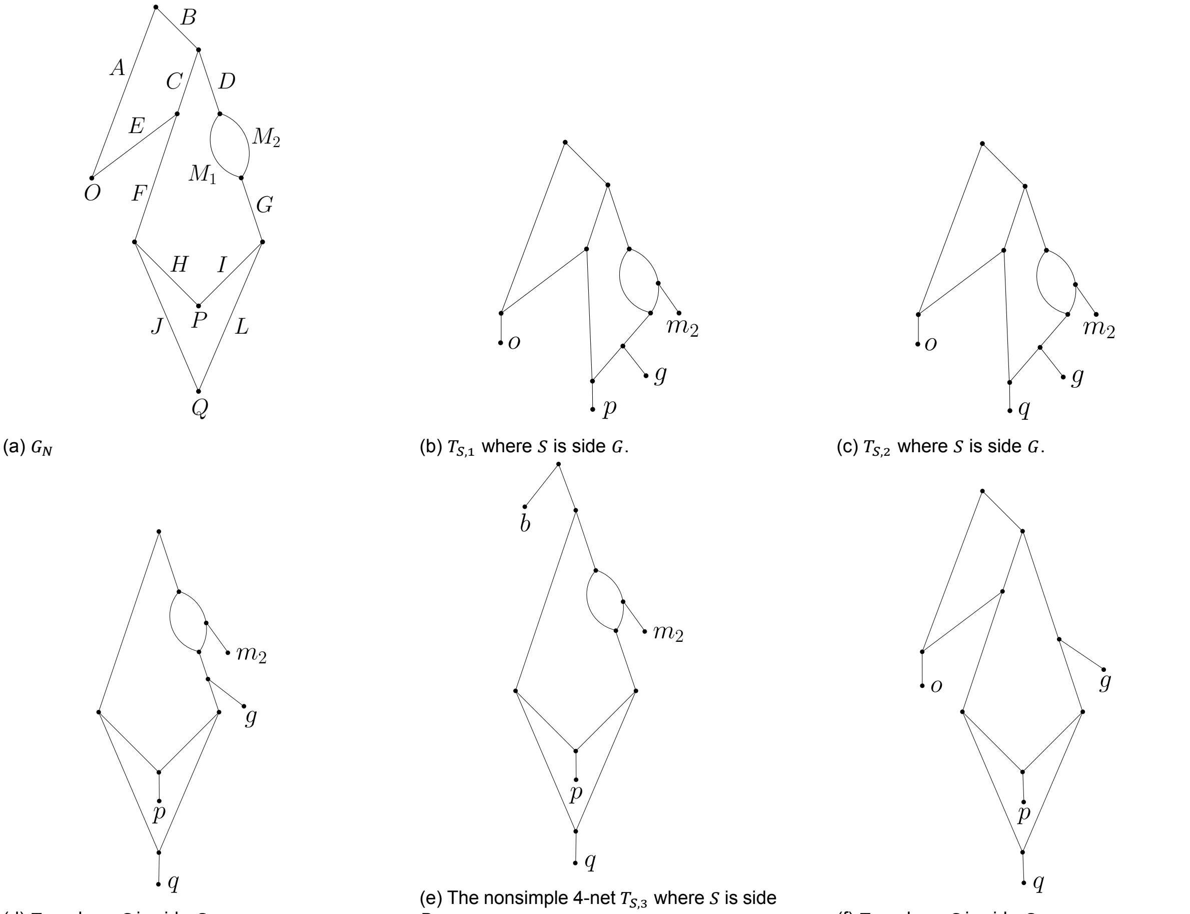
Algorithm 2 *DeterminingSides* determines all possible sides to be on for a leaf in N', given the side in N, for all k-nets on the considered leaf and $k-1$ crucial leaves, where N and N' are binary, simple level-k networks with isomorphic generators and $S_k(N) = S_k(N')$.

procedure DeterminingSides(G_N)
Result: S, the set of sides where a leaf can be on in N'.
Initialisation: Let G_N be a level-k generator with a set of crucial sides of size k. Let $x_1, x_2, \dots, x_k$ be the leaves in N on sides $X_1, X_2, \dots, X_k$, respectively, where $\{X_1, X_2, \dots, X_k\}$ forms a set of crucial sides in G_N. Let $S_1, \dots, S_n$ be the noncrucial sides of G_N and let s_m be a leaf on side S_m where $m \in \{1, \dots, n\}$.
  for $m = 1$ **to** n **do**
    for all $i \in \{1, \dots, k\}$ **do** Let K be the k-net on $\{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_k, s_m\}$ and let G_K be the underlying generator of K constructed from G_N by deleting crucial side X_i and suppressing indegree-1 outdegree-1 vertices and deleting indegree-0 outdegree-1 vertices.
$S \leftarrow \emptyset$
      while G_K is level-$(k-1)$ **do**
        if G_K has no symmetry other than parallel arcs **then**
          if X_i is a vertex and its parents are adjacent **then** Let P_1 be the set of sides which are arcs incident to one of the parents of X_i in G_N.
            if $S_m \in P_1$ **then** $S \leftarrow S \cup P_1$
            else $S \leftarrow S \cup S_m$
            end if
          else if X_i is a vertex and its parents are not adjacent **then** Let P_1 be the set of sides in G_N that are arcs incident to the same parent of X. Let P_2 the set of sides in G_N that are arcs incident to the other parent of X.
            if $S_m \in P_i$ for $i \in \{1, 2\}$ **then**
              for $i = 1$ **to** 2 **do**
                if $S_m \in P_i$ **then** $S \leftarrow S \cup P_i$
                else $S \leftarrow S$
                end if
              end for
            else $S \leftarrow S \cup S_m$
            end if
          else Let P be the set of sides in G_N containing X_i and its adjacent arcs.
            if $S_m \in P_1$ **then** $S \leftarrow S \cup P_1$
            else $S \leftarrow S \cup S_m$
            end if
          end if

Part 2

else Let A be the set of sides of G_K and let $f_1, \dots, f_n$ be all relabellings of sides in A where $f_i : A \to A\ \forall i \in \{1, \dots, n\}$ giving an isomorphic generator.
if X_i is a vertex and its parents are adjacent **then** Let P be the set of sides which are arcs incident to one of the parents of X_i in G_N. Let $P' \in G_K$ be the side which is obtained by suppression of the parents of X_i in G_N.
Let $Y_1, \dots, Y_n$ be the images of P' under relabellings $f_1, \dots f_n$, respectively, and let $Y' = \bigcup_{i=1}^{n} Y_i$.
$P_1 \leftarrow P \cup Y'$
Let $P_2, \dots, P_q$ be sets of sides in G_N such that $|P_j| \geq 2\ \forall j \in \{2, \dots, q\}$ and $\forall Z \in P_j\ \exists f_i$ for some $i \in \{1, \dots, n\}$ such that $f_i(Z) \neq Z$ and $f_i(Z) \in P_j$.
for $i = 1$ to q **do**
if $S_m \in P_i$ **then** $S \leftarrow S \cup P_i$
else $S \leftarrow S$
end if
end for
else if X_i is a vertex and its parents are not adjacent **then** Let u and v be the parents of X. Let P and Q be the sets of sides which are arcs incident to u and v, respectively, and let P' and Q' be the sides in G_T obtained by suppression of u and v, respectively.
if $f(P') = Q'$ **then** $P_1 \leftarrow P \cup Q$ and let P_2 and P_3 be sets of sides in G_N such that they are images of each other under f in G_T such that $P_i \cap P_j = \emptyset$ for $i, j \in \{1, 2, 3\}, i \neq j$.
else $Y \leftarrow f(P')$
$Z \leftarrow f(Q')$
$P_1 \leftarrow P \cup Y$
$P_2 \leftarrow Q \cup Z$
Let P_3 be set of sides in G_N such that they are images of each other under f in G_T such that $P_i \cap P_j = \emptyset$ for $i, j \in \{1, 2, 3\}, i \neq j$.
end if
for $i = 1$ **to** 3 **do**
if $S_m \in P_i$ **then** $S \leftarrow S \cup P_i$
else $S \leftarrow S$
end if
end for
else Let P be the set of sides in G_N containing X and its adjacent arcs. Let $P' \in G_T$ be the side which is obtained by suppression of the vertices above and below X in G_N.
$Y \leftarrow f(P')$
$P_1 \leftarrow P \cup Y$
Let P_2 and P_3 be sets of sides in G_N such that they are images of each other under f in G_T such that $P_i \cap P_j = \emptyset$ for $i, j \in \{1, 2, 3\}, i \neq j$.
for $i = 1$ **to** 3 **do**
if $S_m \in P_i$ **then** $S \leftarrow S \cap P_i$
else $S \leftarrow S$
end if
end for
end if
end if
end while
return S
end for
end for
end procedure

We will give an example of how the algorithm works for a level-4 generator. Let G_N be the level-4 generator in Figure 7.7a. It contains a pair of parallel arcs, and the relabelling of sides H, P, L with J, Q, L gives an isomorphic generator. Let N be a binary, simple level-4 network having G_N as underlying generator. Let $a, \dots, j, l, m_1, m_2, o, p, q$ be arbitrary leaves in N on sides $A, \dots, J, L, M_1, M_2, O, P, Q$, respectively. We consider 4-nets $T_{S,1}$ on $\{s, m_2, o, p\}$, $T_{S,2}$ on $\{s, m_2, o, q\}$, $T_{S,3}$ on $\{s, m_2, p, q\}$ and $T_{S,1}$ on $\{s, o, p, q\}$ where s is a leaf on some noncrucial side S of G_N. Note that o, m_2, p and q are crucial leaves. We have given two examples of $T_{S,3}$, namely the 4-net on $\{g, m_2, p, q\}$ and the 4-net on $\{b, m_2, p, q\}$ to see that if a 4-net contains a leaf that is on a side which is an outgoing arc of the root of G_N, it is possible that the 4-net is not simple. All these 4-nets are given in Figure 7.7. The results for all different sides and different 4-nets are shown in Table 7.2. The overlapping sides are the possible sides for a leaf in N'.



(a) G_N (b) $T_{S,1}$ where S is side G. (c) $T_{S,2}$ where S is side G.

(d) $T_{S,3}$ where S is side G. (e) The nonsimple 4-net $T_{S,3}$ where S is side B. (f) $T_{S,4}$ where S is side G.

Figure 7.7: An example of a level-4 generator with four different 4-nets on three crucial leaves and one noncrucial leaf exhibited by N.

In this thesis we prove encoding results by contradiction. We conjecture that, in general, most simple level-k networks are weakly encoded by their k-nets. To prove which level-k networks are encoded by their k-nets for some k, Algorithm 2 can be used. Implementing the algorithm helps for the most complicated part of the proof, if a proof by contradiction is used, in the same way as the proofs of the other encoding results in this thesis. We conjecture that, as in Nipius, 2020 is done, generators with a certain level have to be considered piece by piece if it has to be proved that all leaves are on the same sides in N and N'. Indeed, if unequal binary, simple level-k networks N and N' are given, then we have proved in Section 4.1.4 that the generators are isomorphic if $\mathcal{S}_k(N) = \mathcal{S}_k(N')$. If all leaves are on the same sides in N and N', then by Lemma 11, the order of the leaves on each side is the same. So proving that leaves are on the same sides in N and N' is the part for which we presented this algorithm.

If a certain level-k generator is given as input, Algorithm 2 gives for leaves on each side the possible sides that the leaves can be on in N', for each k-net as described. The 'overlapping' sides, for example the last column in Table 7.2, is not given by Algorithm 2, so this has to be added in the implementation.

Leaf on side S in N	Trinet $T_{S,1}$	Trinet $T_{S,2}$	Trinet $T_{S,3}$	Trinet $T_{S,4}$	Result for N'
A	A	A	$A \vee B$	A	A
B	B	B	$A \vee B$	B	B
C	C	C	$C \vee E \vee F$	C	C
D	D	D	D	$D \vee M_1 \vee M_2 \vee G$	D
E	E	E	$C \vee E \vee F$	E	E
F	$F \vee H \vee J$	$F \vee H \vee J$	$C \vee E \vee F$	F	F
G	$G \vee I \vee L$	$G \vee I \vee L$	G	$D \vee M_1 \vee M_2 \vee G$	G
H	$F \vee H \vee J$	$F \vee H \vee J$	H	H	H
I	$G \vee I \vee L$	$G \vee I \vee L$	I	I	I
J	$F \vee H \vee J$	$F \vee H \vee J$	J	J	J
L	$G \vee I \vee L$	$G \vee I \vee L$	L	L	L

Table 7.2: Possibilities for a side in N' for a leaf in N, where G_N is the generator in Figure 7.7a.

If the results are all single sides, as in Table 7.2 is the case, then it is proved that if the underlying generator of N is the generator in the input, then all leaves in N are on the same sides in N'. If the results are not all single leaves, then it has still to be proved that for leaves on certain sides that they are on just one and the same side in N', after possibly relabelling sides of $G_{N'}$. We conjecture that this is not possible for every generator, because we conjecture that there are level-k networks that are not encoded by their k-nets for choices of $k > 3$.

8

Conclusion and further research directions

We conjectured that level-k phylogenetic networks are encoded by $(k+1)$-nets. We aimed to prove as strong as possible encoding results that support this conjecture. It turns out that recoverable level-k networks without symmetry besides parallel arcs in the underlying generator of the restriction to any nontrivial biconnected component are encoded by $(k+1)$-nets. A network is recoverable if it contains no biconnected components with one outgoing arc (u, v) and all leaves are below v and a generator has symmetry if it contains a pair of parallel arcs, or if there exists a graph automorphism on its vertices such that at least one vertex is not mapped to itself. This supports our hypothesis for a large set of networks. It generalises the results in Van Iersel and Moulton, 2012 and Nipius, 2020 for cases without such symmetry and builds further on the research on encoding networks having a certain number of reticulations. If some algorithm can build a level-k network without this symmetry in its biconnected components from a set of given $(k+1)$-nets for some $k \geq 2$, we proved that if only the crucial $(k+1)$-nets are known, then this is enough to build the network that represents the complete evolutionary history. The crucial $(k+1)$-nets are subnets on $k+1$ leaves such that they are simple and also level-k.

In Chapter 4, we have also proved that the class of binary, recoverable level-k networks without symmetry besides parallel arcs in the underlying generator of the restriction to any nontrivial biconnected component, with at least $c+1$ leaves and with a set of crucial sides of size at most c in any biconnected component is weakly encoded by $c+1$ nets. A set of crucial sides contains all indegree-2 outdegree-0 vertices together with at least one arc of each pair of parallel arcs of a generator. This means that if some algorithm can build a level-k phylogenetic network without this symmetry from a set of given $c+1$-nets for some $c \geq 2$, then the network on all species that can be build is unique in that class. In this case, if only *crucial* $c+1$-nets are given as input, there is enough information to build the network.

The results in Chapter 4 do not hold in general for level-k networks having symmetry in their underlying generators. We did not find a counterexample for networks having symmetry, so we still conjecture that all level-k networks are encoded by $(k+1)$-nets. This is an open problem. We have proved other encoding results for networks containing such symmetry. In Chapter 5, we proved that the class of binary, recoverable level-k networks on at least p leaves, such that for all networks N in this class the restriction of N to any biconnected component has leaves on at most p sides of its underlying generator, is weakly encoded by p-nets for all $p > 2$. This means that if there is an algorithm that can build the phylogenetic network on all leaves from subnets on p leaves, then the network is unique in the defined class. We actually proved that *crucial* p-nets are enough to give a unique network, and we did this for all results in Chapter 5. The advantage of the results from this chapter is that they hold for networks with symmetry. Also, if $p \leq k+1$, we see that the defined class of level-k networks is weakly encoded by p-nets and therefore weakly encoded by $(k+1)$-nets. This is a result that builds further on the theory and meets our expectations. If $p > k+1$, the results do not imply that level-k networks are encoded by $(k+1)$-nets.

We proved that level-4 phylogenetic networks are encoded by 6-nets. This meets our expectations, because if level-4 networks are encoded by 5-nets, then they are encoded by 6-nets. We actually proved that level-4 networks are encoded by *crucial* 6-nets, since the crucial 6-nets were always enough to determine the networks. In Nipius, 2020, it is proved that the class of most binary recoverable level-3 networks are weakly encoded by trinets by considering groups of generators with different properties. In the proof of our result, we analyse groups of generators, too, especially their symmetries. We prove our result again by contradiction, and the most laborious part is to prove for two binary, simple level-4 networks with the same generator and the same set of 6-nets that the same leaves are on the same sides in both networks. First, we proved for level-4 generators that if there is a symmetry that is an automorphism mapping vertices in a cycle of at least three vertices, then this must be a 4-cycle and there are only two different level-4 generators with this property. For simple networks with one of these generators as underlying generator, we proved that they are encoded by 6-nets. Second, we proved that we may assume that the same leaves are on the same sides if there exists a set of crucial sides in the underlying generator of size at least three. A set of symmetric sides is a set of sides of a generator with symmetry, such that all sides in this set can be mapped to each other, giving an isomorphic generator, and there is no bigger set with this property. We saw that these generators must contain a certain subnetwork. Lastly, we proved that we may assume that the same leaves are on the same sides if there exists a set of crucial sides in the underlying generator of size at most two. Combining the results, together with the result that the order of the leaves on each side is the same, gives that all binary, simple level-4 networks are encoded by 6-nets. As a corollary, this holds for all binary recoverable level-4 networks. The way of analysing the symmetries, that is, looking for cycles and trying to find subnetworks, can help to strengthen this result to 5-nets or extend this result to higher level networks, although the higher the level of generators, the more generators there are (Gambette et al., 2009).

Algorithm 2 can be used to attempt to prove which level-k networks are encoded by their k-nets for some k. If certain level-k networks are encoded by their k-nets instead of $(k+1)$-nets, it is a stronger result. This is still an open problem. We conjecture that not all level-k networks are encoded by their k-nets for each $k > 3$, but the level-3 counterexample cannot be generalised easily. It can be useful to know whether some level-k phylogenetic network (after building it with some algorithm) is encoded by its k-nets. It is therefore useful to have a tool for proving this. Algorithm 2 is such a tool. It determines the possible sides of a leaf in N' if the side in N is given, but it does not ensure that a leaf is on a unique side in N'. An additional proof has to be given if more than one possible side is given by the algorithm. The algorithm works for k-nets, but not for subnets lower than k because it constructs k-nets on $k-1$ crucial leaves, and not less than $k-1$. Extending the algorithm to smaller subnets would make it even stronger.

We will now give some recommendations for further research. We showed that the method of proving that level-k networks are encoded by $(k+1)$-nets for $k=2$ and $k=3$ in Nipius, 2020 and Van Iersel and Moulton, 2012 does also work in this thesis for some general cases. Therefore, we can recommend this method for proving that level-k phylogenetic networks are encoded by p-nets for some $k \geq 4$ and $p \in \mathbb{N}$. An open problem is to prove the conjecture that level-k networks are encoded by $(k+1)$-nets. A different open problem is finding which level-k networks are encoded by k-nets. The bigger the set of networks that are encoded by k-nets, the more useful the result. A related open problem is to find a counterexample showing that not all level-k networks are encoded by k-nets for some $k > 3$.

For proving that leaves are on the same sides in two binary, simple level-k networks with the same generator and the same set of $(k+1)$-nets, an algorithm has to be used. A method that does not consider each generator separately may be used instead. Finding such an algorithm is an open problem.

Algorithms TriLoNet (Oldman et al., 2016) and TriL2Net (Kole, 2020) construct level-1 phylogenetic networks from smaller level-1 networks and construct level-2 networks from level-2 trinets. Our encoding result for level-4 networks makes the development of an algorithm that constructs level-4 phylogenetic networks from its set of 6-nets an open problem. Indeed, because of our encoding results, such an algorithm would give the unique and correct network that represents the complete evolutionary history by a level-4 network. A more general open problem is to develop an algorithm that constructs level-k networks from a set of $(k+1)$-nets or k-nets.

We recommend to implement Algorithm 2. It is a first step to determine whether simple level-k networks with generators with and without symmetry are encoded by their k-nets. In Gambette et al., 2009, an algorithm to construct all level-k generators is presented and can be used. It is an open problem to extend this algorithm to p-nets for $p < k$. In that case, the algorithm has to be extended to the deletion of more than one crucial side from a set of crucial sides.

In applications, nonbinary phylogenetic networks exist (Marcussen et al., 2012, Brassac and Blattner, 2015). Therefore it can be useful to extend the results of this thesis to nonbinary phylogenetic networks. An open problem is therefore proving that (nonbinary) level-k phylogenetic networks are encoded by their $(k+1)$-nets for $k \geq 2$.

Bibliography

Bapteste, E., van Iersel, L., Janke, A., Kelchner, S., Kelk, S., McInerney, J. O., Morrison, D. A., Nakhleh, L., Steel, M., Stougie, L., & Whitfield, J. (2013). Networks: Expanding evolutionary thinking. *Trends in Genetics*, *29*, 439–441. https://doi.org/10.1016/J.TIG.2013.05.007

Brassac, J., & Blattner, F. (2015). Species-level phylogeny and polyploid relationships in hordeum (poaceae) inferred by next-generation sequencing and in silico cloning of multiple nuclear loci. *Systematic biology*, *64*, 792–808. https://doi.org/10.1093/SYSBIO/SYV035

Byrka, J., Guillemot, S., & Jansson, J. (2010). New results on optimizing rooted triplets consistency. *Discrete Applied Mathematics*, *158*, 1136–1147. https://doi.org/10.1016/J.DAM.2010.03.004

Darwin, C. (1859). *On the origin of species by means of natural selection, or preservation of favoured races in the struggle for life*. John Murray.

Dress, A., Huber, K. T., Koolen, J., Moulton, V., & Spillner, A. (2011). *Basic phylogenetic combinatorics.* Cambridge University Press.

Gambette, P., Berry, V., & Paul, C. (2009). The structure of level-k phylogenetic networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5577 LNCS*, 289–300. https://doi.org/10.1007/978-3-642-02441-2_26

Gambette, P., & Huber, K. T. (2011). On encodings of phylogenetic networks of bounded level. *Journal of Mathematical Biology*, *65*, 157–180. https://doi.org/10.1007/S00285-011-0456-Y

Huber, K. T., & Moulton, V. (2013). Encoding and constructing 1-nested phylogenetic networks with trinets. *Algorithmica*, *66*, 714–738. https://doi.org/10.1007/s00453-012-9659-x

Huson, D. H., Rupp, R., & Scornavacca, C. (2010). *Phylogenetic networks: Concepts, algorithms and applications*. Cambridge University Press.

Huson, D. H., & Scornavacca, C. (2012). Dendroscope 3: An interactive tool for rooted phylogenetic trees and networks. *Systematic Biology*, *61*, 1061–1067. https://doi.org/10.1093/SYSBIO/SYS062

Kole, S. (2020). *Constructing level-2 phylogenetic networks from trinets*. http://resolver.tudelft.nl/uuid:c699ea63-f8c8-40f7-8f07-11ac055c42e0

Lott, M., Spillner, A., Huber, K. T., & Moulton, V. (2009). Padre: A package for analyzing and displaying reticulate evolution. *Bioinformatics*, *25*, 1199–1200. https://doi.org/10.1093/BIOINFORMATICS/BTP133

Mallet, J., Besansky, N., & Hahn, M. W. (2016). How reticulated are species? *BioEssays*, *38*, 140–149. https://doi.org/10.1002/BIES.201500149

Marcussen, T., Jakobsen, K. S., Danihelka, J., Ballard, H. E., Blaxland, K., Brysting, A. K., & Oxelman, B. (2012). Inferring species networks from gene trees in high-polyploid north american and hawaiian violets (viola, violaceae). *Systematic Biology*, *61*, 107–126. https://doi.org/10.1093/SYSBIO/SYR096

Marcussen, T., Sandve, S. R., Heier, L., Spannagl, M., Pfeifer, M., Consortium, T. I. W. G. S., Jakobsen, K. S., Wulff, B. B. H., Steuernagel, B., Mayer, K. F. X., & Olsen, O.-A. (2014). Ancient hybridizations among the ancestral genomes of bread wheat. *Science*, *345*. https://doi.org/10.1126/SCIENCE.1250092

Mcvay, J. D., Hipp, A. L., & Manos, P. S. (2017). A genetic legacy of introgression confounds phylogeny and biogeography in oaks. *Proc. R. Soc. B*, *284*. https://doi.org/10.1098/rspb.2017.0300

Nipius, L. (2020). *Rooted binary level-3 phylogenetic networks are encoded by quarnets*. https://repository.tudelft.nl/islandora/object/uuid%3Aa9c5a8d4-bc8b-4d15-bdbb-3ed35a9fb75d

Oldman, J., Wu, T., van Iersel, L., & Moulton, V. (2016). Trilonet: Piecing together small networks to reconstruct reticulate evolutionary histories. *Molecular Biology and Evolution*, *33*, 2151–2162. https://doi.org/10.1093/MOLBEV/MSW068

Poormohammadi, H., Eslahchi, C., & Tusserkani, R. (2014). Tripnet: A method for constructing rooted phylogenetic networks from rooted triplets. *PLoS ONE*, *9*, e106531. https://doi.org/10.1371/JOURNAL.PONE.0106531

Ranwez, V., Berry, V., Criscuolo, A., Fabre, P.-H., Guillemot, S., Scornavacca, C., & Douzery, E. J. P. (2007). Physic: A veto supertree method with desirable properties. *Systematic Biology*, *56*, 798–817. https://doi.org/10.1080/10635150701639754

Scornavacca, C., Berry, V., Lefort, V., Douzery, E. J., & Ranwez, V. (2008). Physicist: Cleaning source trees to infer more informative supertrees. *BMC Bioinformatics*, *9*, 1–17. https://doi.org/10.1186/1471-2105-9-413

Soucy, S. M., Huang, J., & Gogarten, J. P. (2015). Horizontal gene transfer: Building the web of life. *Nature Reviews Genetics*, *16*, 472–482. https://doi.org/10.1038/nrg3962

Than, C., Ruths, D., & Nakhleh, L. (2008). Phylonet: A software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinformatics*, *9*, 1–16. https://doi.org/10.1186/1471-2105-9-322

Van Iersel, L., & Moulton, V. (2012). Trinets encode tree-child and level-2 phylogenetic networks. *Journal of Mathematical Biology*, *68*, 1707–1729. http://arxiv.org/abs/1210.0362