From Subtrees to Supertrees

Sebastian Böcker

Universität Bielefeld
Graduiertenkolleg
Strukturbildungs–
prozesse

Dissertation
zur Erlangung des Doktorgrades
der Fakultät für Mathematik
Universität Bielefeld

Bielefeld, im August 1999

1. Gutachter: Prof. Andreas W.M. Dress
2. Gutachter: Prof. Michael A. Steel (Christchurch, Neuseeland)
3. Gutachter: Prof. Thomas Zink

Datum der mündlichen Prüfung: 28. Januar 2000

Gedruckt auf alterungsbeständigem Papier ∞ ISO 9706.

Acknowledgments

This thesis was written under the supervision of Prof. Andreas W.M. Dress.

Parts of this thesis have been published in advance [BD00b, BDS99], and further papers are forthcoming [BBDS, BD, Böca, Böcb].

The work was carried out while being supported by the "Graduiertenkolleg Strukturbildungsprozesse" that is part of the Center for Interdisciplinary Research on Structure Formation (FSPM) at the University of Bielefeld, Germany.

First and foremost, I wish to thank my supervisor Professor Andreas Dress. Without his ideas, comments, and contributions this thesis would not exist. I also thank him for bringing me into contact with the topic, for fruitful discussions, and for maintaining my enthusiasm. His promotion made it possible for me to go to Christchurch, New Zealand, for three months.

Special thanks go to Dr. Mike Steel who asked the right question at the right time and thereby gave me the chance to write this thesis. He is not only my "unofficial supervisor" but also my hiking trainer, and he could doubtlessly make a third career as a cartographer. Thank you very much for the hospitality in Christchurch, and for the maps!

Thanks (without any special order) go to: My girlfriend Claudia Hüttner for listening and not complaining; Anke Schnitker for the chats; my roommates at university, Christian Alscher and Swen Osterkamp, for keeping me sane most of the time; everybody else in the Graduiertenkolleg; my primary school teacher Friedel Karaschewski for telling me to study mathematics; Kordula Döring for the needles; Thomas Müller, Nikolaus Wiesner, and the city of Hamburg for brain relaxation; Cake for three great CDs; Professor Peter Slodowy for showing me the fun and beauty of mathematics; the Christchurchians for a wonderful time; Bernhard Balkenhol, Holger Paschke, and Karl-Hermann Wieners for computer support; Marc Kupietz for cigarettes, movies, and music; my Parents for everything.

In accordance with mathematics protocol, I will endeavor to use the personal pronoun "we" to indicate the reader and the writer, rather than the "royal we."

Contents

CHAPTER 1

Introduction

1.1. The problem of tree amalgamation

Given a collection $\mathcal{T}$ of leaf labeled trees with generally distinct, though not necessarily disjoint label sets, one might want to *amalgamate* these trees into one leaf labeled "supertree" T so that all trees in $\mathcal{T}$ are "induced" subtrees of T. Hence, one might want to know whether such a supertree T exists at all, and if so, whether it is uniquely determined by $\mathcal{T}$. The special case of amalgamating two trees into one supertree is illustrated in Fig. 1.1, and a complete answer regarding the amalgamation of exactly *two* trees will be given in Chapter 4.

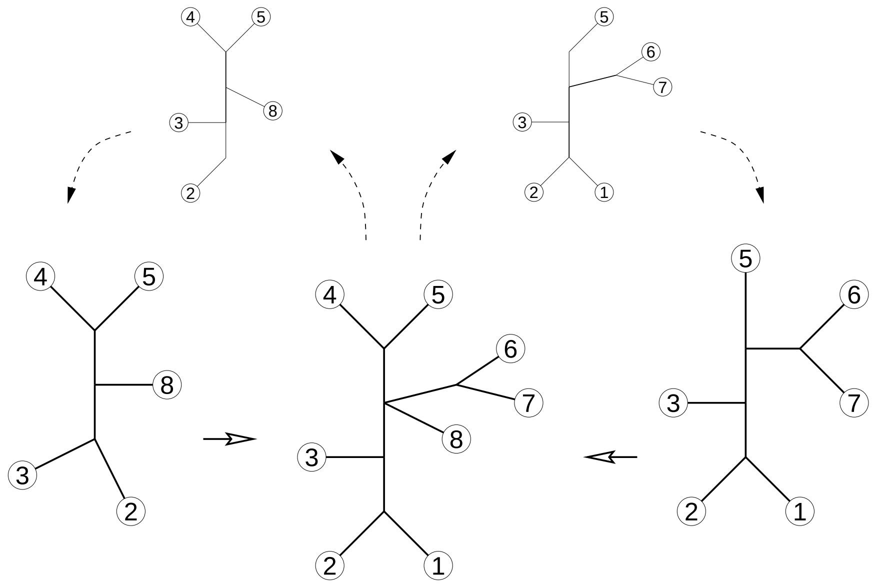


FIGURE 1.1. How to amalgamate two trees into one: The two trees to be amalgamated are depicted bottom left and right, a "supertree" is depicted bottom center, and the two original input trees resulting as "induced" subtrees of this supertree are depicted top left and right.

In order to search for a supertree of a given collection $\mathcal{T}$ of leaf labeled trees, we could simply examine all possible (appropriately leaf labeled) tree structures and determine which (if any) of them "induces" every tree in $\mathcal{T}$. However, this is computationally infeasible since even the number of non-isomorphic *binary* leaf labeled trees grows super-exponentially with the number of leaves: Indeed, this number is precisely the product $(2n-5)!! = 1 \cdot 3 \cdots (2n-5)$ of the first $n-2$ odd numbers where n equals the number of leaves — a result established already by Schröder [Sch70] back in 1870.

In fact, even the problem of determining whether there exists *at least one* supertree of $\mathcal{T}$ was proven to be NP-complete by Steel [Ste92]; and it remains NP-complete if we assume — without loss of generality, as we will see in Section 3.5 — that all trees in $\mathcal{T}$ have precisely four leaves. The complexity of determining whether some (known!) tree is the *unique* supertree of a given collection is still unknown, but strongly conjectured to be NP-complete as well. And if the trees in $\mathcal{T}$ can be amalgamated, there may be (exponentially) many supertrees; see Example 19 on page 81 for a "non-trivial" case where *all* of these (exponentially many) supertrees are binary.

1.2. Motivation via applications

In various fields of classification such as biology [CSW87, SPH98, SOWH96], linguistics [War97], or stemmatology [BBHR98, Rob96], trees are used to represent evolutionary, historical, or hierarchical relationships. In biology, such trees ("phylogenies") typically represent the evolutionary history of a collection of extant species, or the line of descent of some gene. In most applications, the objects of interest are found at the tips (leaves) of the tree, while all interior vertices correspond to a branching (or speciation) event.

There are several reasons why amalgamation problems as described above arise naturally in these fields. Firstly, we may wish to combine trees that have been reconstructed using distinct, though overlapping collections of objects (usually by different researchers using different data and, as often as not, different reconstruction methods). A second reason is that, in general, it is difficult to accurately reconstruct large trees directly, and we may choose instead to reconstruct "trustworthy" trees for small subsets, and then combine these in a supertree (or supertrees), see [BD86, HNP+98, HVW99, HW99, SvH96, Wil98]. An additional reason for this approach is that, for genetic data, the number of sites that can be accurately aligned across a small number of closely related sequences is often much larger than the corresponding number of sites for a set that is large and includes rather diverse sequences. Also, the accuracy of polynomial time tree reconstruction methods (like *neighbor joining* or heuristics for *maximum parsimony*) appears to deteriorate with the maximum evolutionary distance in a phylogeny [RW97, HNR+98].

1.3. The main result

Although the problems of (a) finding a supertree and (b) testing whether it is uniquely determined are known to be hard, a full answer can be given to the second question (implying as well an efficient way to eventually construct this *unique* supertree) in the following, particularly interesting case: Given a collection of binary (or non-degenerated) trees $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$, let X_j denote the set of leaves (or pending vertices) of the tree T_j for $j = 1, \ldots, k$, let X denote the set of leaves of a binary tree T, and assume that X coincides with the union of the (not necessarily disjoint) sets $X_1, \ldots, X_k$. If T is the unique supertree of the collection $\mathcal{T}$, then

$$\#X - 3 \leq \sum_{j=1}^{k} (\#X_j - 3)$$

must always hold, suggesting that particular attention should be devoted to the *minimum*[1] case — that is, the case where equality holds. To this end, we define the excess of the collection $\mathcal{T}$ by

$$\mathrm{exc}(\mathcal{T}) := \#\left(\bigcup\nolimits_{j=1}^{k} X_j\right) - \sum_{j=1}^{k}(\#X_j - 3) - 3\,,$$

and we define $\mathcal{T}$ to be excess-free if $\mathrm{exc}(\mathcal{T}) = 0$ holds. Now, our main result can be stated as follows:

THEOREM 1. *Given a non-empty collection of binary trees $\mathcal{T}$ that is excess-free. Then, $\mathcal{T}$ uniquely defines a supertree T if and only if $\#\mathcal{T} = 1$ or there exists a bipartition of $\mathcal{T}$ into two proper, disjoint subsets $\mathcal{T}_1$, $\mathcal{T}_2 \subset \mathcal{T}$ such that $\mathcal{T}_j$ is excess-free and uniquely defines a supertree T_j for $j = 1, 2$, and T is the unique supertree of T_1, T_2.*

The non-trivial part of this theorem is equivalent to the following corollary:

COROLLARY 1.1. *Given a collection of binary trees $\mathcal{T}$ that is excess-free, uniquely defines a supertree, and has cardinality at least two. Then, there exist two distinct trees $T, T' \in \mathcal{T}$ such that the collection $\{T, T'\}$ is excess-free and uniquely defines a supertree.*

We defer the proofs to Chapter 8, see Theorem 11 and Corollary 8.4 on page 84. Corollary 1.1 will enable us to construct a polynomial time algorithm to check whether a collection of trees uniquely defines a supertree, provided this collection is "sufficiently similar" to a minimum one.

1.4. Structure of this thesis

Firstly, we present some basic definitions and derive some basic results needed later on (Chapters 2 and 3). Then, we give a formal proof of a result regarding the amalgamation of exactly two trees that might be considered "folklore" (even though the actual proof of this result turned out to be unexpectedly cumbersome, see Chapter 4). The first fundamental result of this thesis is the *characterization of ample patchworks* (Chapter 5); the second one is the *formal proof of Theorem 8* via quartet encodings (Chapters 6 and 7). Finally, we derive some applications of Theorem 8, including Theorem 1 stated above (Chapter 8).

The structure of this thesis in detail:

In CHAPTER 2, we recall the prerequisites from graph theory that are needed throughout this thesis. We introduce the concept of "patches" (the set of edges of a subtree of the given tree), and we define two equivalence relations on the set of vertices of a tree, enabling us to remove (arbitrary subsets of) vertices from a tree in a formally adequate way. Because of the scope of this thesis, we give several definitions only for trees (for example, "median" or "patch"), although these definitions could be easily generalized to larger classes of graphs.

The link between phylogenies (biology) and trees (mathematics) is given in CHAPTER 3 by introducing X-trees. First, we recall the correspondence between X-trees

[1]To clarify the necessary distinction between *minimum* and *minimal* in this context, see Example 4 on page 29.

and (compatible) split systems established by Buneman [Bun71]. Then, we describe several ways of computing induced subtrees from a given X-tree: While it is well-known (and somehow straightforward) how to induce a subtree "via a subset of taxa," we present two more constructions (via subsets of edges) that will prove useful in Chapter 7. Altogether, this provides the mathematical vocabulary that is necessary to (at least) formally *state* the questions raised in Section 1.1. In addition, we show that we can restrict ourselves (without loss of generality) to trees with exactly four leaves, or *quartet splits*, when tackling those questions.

As mentioned above, CHAPTER 4 deals with the existence and uniqueness of amalgamations for exactly *two* trees T_1, T_2 (whether or not the collection $\mathcal{T} := \{T_1, T_2\}$ is excess-free) and provides a formal proof of this pertinent and probably well known, yet — to the best of my knowledge — never rigorously established result.

In CHAPTER 5, we introduce the concept of *patchworks* that exhibits an interesting relationship with hierarchies. In particular, we are interested in ample patchworks or — (almost) equivalently — in patchworks that contain a maximal hierarchy. While it is comparatively difficult to check whether an *arbitrary* set system is ample or contains a maximal hierarchy, this task becomes easier when limiting ourselves to patchworks in which case a fast algorithm for deciding this question can be given. Chapter 5 is based on [BD00b]; further results regarding maximal and ample hierarchies can be found in [BD00a].

Quartet encodings are introduced in CHAPTER 6. They will be a useful tool for proving the main theorem in Chapter 7. We also study the excess of a family $\mathcal{T}$ of quartet trees, and we show that excess-free collections of quartet trees are particularly relevant in this context. In particular, the families under consideration form a patchwork, thus enabling us to use the results of Chapter 5.

In CHAPTER 7, we state our main result in terms of quartet encodings (Theorem 8). More precisely, we will show that a "tight" quartet encoding q of a binary tree T uniquely determines T if and only if the patchwork consisting of all excess-free collections of quartet trees derived from q is an ample patchwork. The rest of Chapter 7 is devoted to a rigorous proof of Theorem 8. This theorem, together with a short sketch of its proof (Section 7.3), was already announced in [BDS99].

CHAPTER 8 is dedicated to applications of Theorem 8: We restate this theorem in different ways, showing how it can be applied in "real world situations;" and we give several algorithms with quadratic runtime related to the task of (re-)constructing a unique supertree. Furthermore, we discuss the concept of "dyadic closure" and indicate how our results can be used in this context (see also [BBDS]). Finally, we explicitely construct a collection of quartet trees of "minimal cardinality" that has exponentially many supertrees, every one of them necessarily *binary* (see also [Böca]).

As yet another application of the ideas presented in this thesis, simple, though fundamental restrictions regarding the existence of "unbiased" consensus and supertree methods have been deduced in [SDB00].

1.5. Further steps

One might think of several ways of extending the results of this thesis. Here, we list only a few:

- Since the proof of Theorem 8 is constructive (in the sense that, when the conditions of the theorem are violated, we take one supertree and explicitly construct a second one), we can possibly use our methods to obtain a “fast” algorithm for finding *all* supertrees of (reasonably structured) collections of trees — where “fast” simply means “faster than checking all possible trees.”
- The concept of patchworks — and the problem of deciding whether or not a given patchwork is ample — is so natural that it might well be applicable in other areas of combinatorics.
- The implications of our results regarding the dyadic closure construction must be examined in detail.

1.6. Basic notations

Throughout this thesis, $\#X \in \mathbb{N}_0 \cup \{\infty\}$ denotes the cardinality and $\mathcal{P}(X)$ the set of subsets of a set X, while $\binom{X}{k}$ denotes the set of all subsets of X of cardinality k. The union $A \cup B$ of two disjoint sets A, B will also be denoted by “$A \mathbin{\dot\cup} B$”, and the symmetric difference $(A \cup B) - (A \cap B) = (A \setminus B) \mathbin{\dot\cup} (B \setminus A)$ of two arbitrary sets A, B by “$A \vartriangle B$”. We write “$A \subseteq B$” if the set A is a subset of or equal to B, while “$A \subset B$” or “$A \subsetneqq B$” indicates that A is a proper subset of (and not equal to) B.

For a collection $\mathcal{C} \subseteq \mathcal{P}(X)$ of subsets of X — whose members might also be called ($\mathcal{C}$-)*clusters* — we define

$$\bigcup \mathcal{C} := \bigcup_{C \in \mathcal{C}} C \quad \text{and} \quad \bigcap \mathcal{C} := \bigcap_{C \in \mathcal{C}} C\,, \tag{1.1}$$

$$\text{and} \quad \mathcal{C}^X := \mathcal{C} \cup \{\emptyset, X\} \cup \{\{x\} \mid x \in X\}\,. \tag{1.2}$$

A fuller list of symbols appears on page 95.

CHAPTER 2

Graphs, Trees, and Patches

After recalling (respectively introducing) the necessary prerequisites from graph theory, we will be particularly interested in two things: Firstly, given a tree T, we consider the family of (T-)*patches*, that is, those collections of edges of T that form the complete set of edges of a subtree of T; and second, to deal with trees broken apart by removing a subset of vertices in a formally adequate way, we define two equivalence relations on the set of vertices of a tree in Section 2.4.

2.1. Graphs and paths

A (simple) *graph* $G = (V, E)$ is a pair consisting of a set $V = V(G)$ and a subset $E = E(G)$ of $\binom{V}{2}$. The elements of $V = V(G)$ are called the *vertices* of G, and the elements of $E = E(G)$ are called its *edges*. By (this) definition, a graph does not contain *loops* or *multiple edges*. A graph G is called *finite* if its set of vertices V is finite. The *degree* of a vertex $v \in V$ is defined by

$$\deg(v) = \deg_G(v) := \#\{e \in E \mid v \in e\} \in \mathbb{N}_0 \cup \{\infty\}\,.$$

An edge $e \in E$ is said to be *incident* with a vertex v if $v \in e$ holds. We denote the maximal degree of a graph G by

$$\Delta = \Delta(G) := \sup\{\deg(v) \mid v \in V\}\,. \tag{2.1}$$

For $k = 0, 1, \ldots, \Delta$, let $V_k = V_k(G) := \{v \in V \mid \deg(v) = k\}$ denote the set of vertices of G of degree k. Obviously, $V = V_0 \uplus V_1 \uplus \cdots \uplus V_\Delta$ holds. The vertices in V_1 are called the *leaves* of G. Counting the edges of G twice (or, rather, counting the pairs $(v, e) \in V \times E$ with $v \in e$ in two distinct ways) leads to:

$$2\,\#E = 1\,\#V_1 + 2\,\#V_2 + 3\,\#V_3 + \cdots + \Delta\,\#V_\Delta \tag{2.2}$$

A *G-path* — or just a *path* if the reference to G is clear — is a finite sequence $p = (p_0, p_1, \ldots, p_n)$ of elements from V with

$$\begin{aligned} \{p_{i-1}, p_i\} \in E \quad &\text{for all} \quad i = 1, \ldots, n \\ \text{and} \quad p_{i-1} \neq p_{i+1} \quad &\text{for all} \quad i = 1, \ldots, n-1\,. \end{aligned}$$

The vertex p_i will also be denoted by $v_i(p)$ for all $i = 0, \ldots, n$ while the pair $\{p_{i-1}, p_i\}$ will be denoted by $e_i(p)$ for all $i = 1, \ldots, n$. The number n occurring in the definition of a path p is defined to be its *length*; it is denoted by $l(p)$. We denote the vertex and edge set of a path p by

$$[p] := \{v_i(p) \mid i = 0, 1, \ldots, l(p)\} \quad \text{and} \quad \langle p\rangle := \{e_i(p) \mid i = 1, \ldots, l(p)\}\,. \tag{2.3}$$

The point p_0 is called the *starting point* of p; it is also denoted by $\alpha(p)$. The point $p_n = p_{l(p)}$ is called its *end point*; it is also denoted by $\omega(p)$. Given any two vertices

u, $v \in V$, every path p with $\alpha(p) = u$ and $\omega(p) = v$ is also called a path *from* u *to* v. The graph G is said to be *connected* if there exists a path from u to v for any two vertices u, $v \in V$. A *connected component* of a graph G is a maximal connected subgraph of G, where a graph $G' = (U, F)$ is defined to be a *subgraph* of a graph $G = (V, E)$ if $U \subseteq V$ and $F \subseteq E \cap \binom{U}{2}$ holds.

We denote the set of all paths in G by $\mathbb{P}(G)$.

Given two paths p, $q \in \mathbb{P}(G)$ with $\omega(p) = \alpha(q)$, we define their *concatenation* $p \circ q$ as follows: With $i_0 = i_0(p, q)$ denoting the largest index $i \in \{0, \ldots, \min(l(p), l(q))\}$ with $p_{l(p)-j} = q_j$ for all $j \in \{0, 1, \ldots, i\}$, we set

$$p \circ q := \left(p_0, p_1, \ldots, p_{l(p)-i_0-1}, p_{l(p)-i_0} = q_{i_0}, q_{i_0+1}, \ldots, q_{l(q)}\right). \tag{2.4}$$

If $i_0(p, q) = 0$ holds, we will also write $p \odot q$ instead of $p \circ q$.

Recall that $\mathbb{P}(G)$ forms a groupoid (cf. [Bra27, Bro87]) relative to "$\circ$", the inverse p^{-1} of a path p in $\mathbb{P}(G)$ being the path $(p_{l(p)}, p_{l(p)-1}, \ldots, p_1, p_0)$, while the neutral elements are exactly the paths of length 0. In particular, $(p \circ q) \circ r = p \circ (q \circ r)$ as well as $(p \circ q)^{-1} = q^{-1} \circ p^{-1}$ holds for all p, q, $r \in \mathbb{P}(G)$ with $\omega(p) = \alpha(q)$ and $\omega(q) = \alpha(r)$.

2.2. Trees

A *tree* $T = (V, E)$ is a finite[1] graph such that, for any two vertices u, $v \in V$, there exists a *unique* path from u to v. We will denote this unique path from u to v by $p_T(u \to v)$, or just by $p(u \to v)$, if reference to T is clear. Note that

$$p(u \to w) = p(u \to v) \circ p(v \to w) \tag{2.5}$$

must hold for arbitrary vertices u, v, $w \in V$.

A tree T is called *binary* if every vertex of T is of degree 1 or 3.

It is well known that $\#E = \#V - 1$ must hold for any tree $T = (V, E)$. Together with (2.2), this implies

$$\begin{aligned}
\#E &= 3\,\#E - 2\,\#E \\
&= 3\,\#V - 3 - 2\,\#E \\
&= 3\,\#V_1 + 3\,\#V_2 + \cdots + 3\,\#V_\Delta - 3 - 1\,\#V_1 - 2\,\#V_2 - \cdots - \Delta\#V_\Delta \\
&= 2\,\#V_1 + \#V_2 - 3 - \sum_{k=1}^{\Delta-3} k \cdot \#V_{k+3}
\end{aligned} \tag{2.6}$$

and hence

$$\#E \leq 2\,\#V_1 + \#V_2 - 3\,, \tag{2.7}$$

with equality if and only if $\deg(v) \leq 3$ for all $v \in V$. We will denote the *inner edges* of T — that is, those edges of T that are not incident with a leaf — by

$$\mathring{E} = \mathrm{Int}(T) := \{e \in E \mid e \cap V_1 = \emptyset\}$$

[1] Obviously, this condition is not required in general. In this thesis however, we will limit our attention to finite trees.

and we denote by $\mathring{T}$ the subtree $(V - V_1, \mathring{E})$ of T. Then, as $\#\mathring{E} = \#E - \#V_1$ obviously holds, (2.7) implies

$$\#\mathring{E} \leq \#V_1 + \#V_2 - 3\,, \tag{2.8}$$

again with equality if and only if $\deg(v) \leq 3$ for all $v \in V$.

For $T = (V, E)$ a tree and $u,\, v \in V$ arbitrary, we define:

$$\begin{aligned} d(u,v) = d_T(u,v) &:= l(p(u{\to}v))\,, \\ [u,v] = [u,v]_T &:= \big[p(u{\to}v)\big]\,, \\ v_i(u{\to}v) &:= v_i(p(u{\to}v)) \quad \text{for all} \quad i = 0,\, \ldots,\, d(u,v)\,, \\ \langle u,v\rangle = \langle u,v\rangle_T &:= \langle p(u{\to}v)\rangle\,, \\ e_i(u{\to}v) &:= e_i(p(u{\to}v)) \quad \text{for all} \quad i = 1,\, \ldots,\, d(u,v)\,. \end{aligned}$$

Clearly, the subgraph $\big([u,v], \langle u,v\rangle\big)$ of T is a tree with maximal degree Δ at most 2.[2] Furthermore, we have $[u,v] = [v,u]$, $\langle u,v\rangle = \langle v,u\rangle$, and

$$\#\,\langle u,v\rangle = d(u,v) = \#[u,v] - 1 \tag{2.9}$$

for all $u,\, v \in V$.

It is well known that $d : V \times V \to \mathbb{N}_{\geq 0}$ defines a metric — in fact, this metric is easily seen to coincide with the so-called *shortest path metric*, defined on arbitrary connected graphs.

Example.

1. Let $T = (V, E)$ denote the tree depicted in Fig. 2.1, with vertex set $V = \{1, \ldots, 9, a, \ldots, e\}$ and edge set $E = \{\{1,a\}, \{2,a\}, \ldots, \{c,d\}, \{c,e\}\}$.

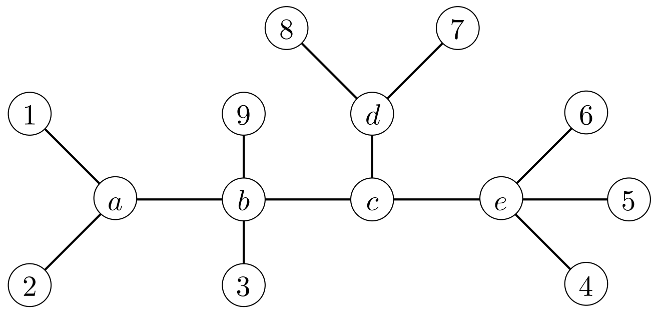


FIGURE 2.1. A tree.

Then T is not binary in view of $\deg(b) = \deg(e) = 4$. Furthermore, we have

$$\begin{gathered} V_1 = \{1, \ldots, 9\}, \quad V_2 = \emptyset, \quad \mathring{E} = \big\{\{a,b\},\ \{b,c\},\ \{c,d\},\ \{c,e\}\big\}, \\ p(1{\to}7) = (1, a, b, c, d, 7), \quad d(a,5) = 4, \quad [3,d] = \{3, b, c, d\}, \\ v_4(1{\to}7) = d, \quad \langle 1,2\rangle = \big\{\{1,a\},\ \{a,2\}\big\}, \quad \text{and} \quad e_3(1{\to}7) = \{b,c\}\,. \end{gathered}$$

[2] In the literature, a "path" is often defined to be a tree with this property, see for example [Die97].

REMARK 2.1. Let $T = (V, E)$ denote a tree, and let p, $q \in \mathbb{P}(T)$ denote two paths with $\omega(p) = \alpha(q)$. Then $i_0(p, q) = 0$ holds if and only if the only element in the intersection $[p] \cap [q]$ is $\omega(p) = \alpha(q)$ if and only if the intersection $\langle p\rangle \cap \langle q\rangle$ is empty.

LEMMA 2.2. *Let $T = (V, E)$ denote a tree. For arbitrary u, v, $w \in V$, we have*

$$[u, w] \subseteq [u, v] \cup [v, w] \quad \text{and} \quad \langle u, w\rangle = \langle u, v\rangle \,\triangle\, \langle v, w\rangle \subseteq \langle u, v\rangle \cup \langle v, w\rangle \ .$$

PROOF. This is a simple consequence of the fact that, by definition of $p \circ q$, the inclusions $[p\circ q] \subseteq [p]\cup[q]$ and $\langle p \circ q\rangle \subseteq \langle p\rangle\cup\langle q\rangle$ hold for all paths p, q with $\omega(p) = \alpha(q)$ in a graph G while the more precise relationship $\langle p \circ q\rangle = \langle p\rangle \,\triangle\, \langle q\rangle$ must clearly hold if G is a tree. □

LEMMA 2.3. *Given a tree $T = (V, E)$ and vertices u, v, $w \in V$, one has*

$$v \in [u, w] \iff [u, v] \cap [v, w] = \{v\} \iff [u, v] \cup [v, w] = [u, w] \iff$$
$$\langle u, v\rangle \cap \langle v, w\rangle = \emptyset \iff \langle u, v\rangle \cup \langle v, w\rangle = \langle u, w\rangle \iff d(u, v) + d(v, w) = d(u, w)\,.$$

PROOF. This is obvious. □

COROLLARY 2.4. *Given a tree $T = (V, E)$ and vertices u, v, $w \in V$ with $v \in [u, w]$, one has*

$$p(u\to w) = p(u\to v) \circ p(v\to w)\,.$$

For three vertices v_1, v_2, $v_3 \in V$ of a tree $T = (V, E)$, we define the *median* $\mathrm{med}(v_1, v_2, v_3)$ of v_1, v_2, and v_3 to be the unique vertex m of T with

$$\{m\} = [v_1, v_2] \cap [v_2, v_3] \cap [v_3, v_1]\,.$$

We have to show that, this way, the median is well-defined. Firstly, note that the set

$$M := [v_1, v_2] \cap [v_2, v_3] \cap [v_3, v_1]$$

is non-empty: With $i_0 := i_0\big(p(v_1\to v_2), p(v_2\to v_3)\big)$ and $m := v_{(d(v_1,v_2)-i_0)}(v_1\to v_2) = v_{i_0}(v_2\to v_3)$ we have

$$p(v_1\to v_3) = p(v_1\to v_2) \circ p(v_2\to v_3) = p(v_1\to m) \circ p(m\to v_3)$$

by definition of i_0 and, hence, $m \in M$. Moreover, for $1 \le i < j \le 3$, we have $m \in [v_i, v_j]$ and therefore $[v_i, v_j] = [v_i, m] \cup [m, v_j]$ as well as $[v_i, m] \cap [m, v_j] = \{m\}$ which in turn implies

$$\begin{aligned}
&[v_1, v_2] \cap [v_2, v_3] \cap [v_3, v_1] = \\
&\qquad = \big([v_1, m] \cup [v_2, m]\big) \cap \big([v_2, m] \cup [v_3, m]\big) \cap \big([v_3, m] \cup [v_1, m]\big) \\
&\qquad = \big([v_1, m] \cap [v_2, m]\big) \cup \big([v_2, m] \cap [v_3, m]\big) \cup \big([v_3, m] \cap [v_1, m]\big) \\
&\qquad = \{m\}\,.
\end{aligned}$$

Consequently, $\mathrm{med}(v_1, v_2, v_3)$ is well-defined.

Clearly, the median does not depend on the order of v_1, v_2, v_3. Moreover, it allows us to improve the formula for $[u, w]$ in Lemma 2.2: For u, v, $w \in V$, we always have

$$[u, w] = [u, v] \,\triangle\, \{\mathrm{med}(u, v, w)\} \,\triangle\, [v, w]\,. \tag{2.10}$$

The following lemma is another consequence of the above arguments:

LEMMA 2.5. *Let $T = (V, E)$ denote a tree, and assume v_1, v_2, $v_3 \in V$. Choose vertices u_1, u_2, u_3 satisfying*

$$u_j \in [v_j, \operatorname{med}(v_1, v_2, v_3)] \quad \textit{for } j = 1, 2, 3 \tag{2.11}$$

then $\operatorname{med}(u_1, u_2, u_3)$ *coincides with* $\operatorname{med}(v_1, v_2, v_3)$.

PROOF. Let $m := \operatorname{med}(v_1, v_2, v_3)$. In view of Lemma 2.2 and 2.3, we have

$$[u_i, u_j] \subseteq [u_i, m] \cup [m, u_j] \subseteq [v_i, m] \cup [m, v_j] = [v_i, v_j]$$

for $i \neq j$ and, hence,

$$\begin{aligned}\{\operatorname{med}(u_1, u_2, u_3)\} &= [u_1, u_2] \cap [u_2, u_3] \cap [u_3, u_1] \\ &\subseteq [v_1, v_2] \cap [v_2, v_3] \cap [v_3, v_1] \\ &= \{\operatorname{med}(v_1, v_2, v_3)\}\,.\end{aligned}$$

Clearly, this implies $\operatorname{med}(u_1, u_2, u_3) = \operatorname{med}(v_1, v_2, v_3)$. □

2.3. Patches

The purpose of this section is to establish convenient notations when dealing with subtrees of a given tree. For the rest of this section, suppose that $T = (V, E)$ is a (finite) tree. Given a subset $U \subseteq V$, we define

$$[U] = [U]_T := \bigcup_{u,v \in U} [u, v] \quad \text{and} \quad \langle U \rangle = \langle U \rangle_T := \bigcup_{u,v \in U} \langle u, v \rangle\ . \tag{2.12}$$

The set $[U] \subseteq V$ is called the *vertex closure* of U, while $\langle U \rangle \subseteq E$ is called its *edge closure*. Similarly, we define $[F] := [\bigcup F]$ and $\langle F \rangle := \langle \bigcup F \rangle$ for every subset $F \subseteq \mathring{E}$.

LEMMA 2.6. *Given a tree $T = (V, E)$ and a subset $U \subseteq V$, we have*

$$[[U]] = [U] \quad \textit{as well as} \quad \langle [U] \rangle = \langle U \rangle\ ,$$

and either $\bigcup \langle U \rangle = [U]$ *or* $\#U = 1$.

PROOF. For every vertex $w \in [[U]]$, there must exist u, $v \in [U]$ and u_1, u_2, v_1, $v_2 \in U$ with $w \in [u, v]$, $u \in [u_1, u_2]$, and $v \in [v_1, v_2]$. In view of the above lemmata, this implies

$$\begin{aligned}w \in [u, v] \subseteq [u, v_1] \cup [v_1, v] &\subseteq [u, u_1] \cup [u_1, v_1] \cup [v_1, v] \\ &\subseteq [u_2, u_1] \cup [u_1, v_1] \cup [v_1, v_2] \subseteq [U]\end{aligned}$$

as claimed.

Similarly, $e \in \langle u, v \rangle$ for some u, $v \in [U]$, together with $u \in [u_1, u_2]$ and $v \in [v_1, v_2]$ for some u_1, u_2, v_1, $v_2 \in U$ as above, implies

$$\begin{aligned}e \in \langle u, v \rangle \subseteq \langle u, v_1 \rangle \cup \langle v_1, v \rangle &\subseteq \langle u, u_1 \rangle \cup \langle u_1, v_1 \rangle \cup \langle v_1, v \rangle \\ &\subseteq \langle u_2, u_1 \rangle \cup \langle u_1, v_1 \rangle \cup \langle v_1, v_2 \rangle \subseteq \langle U \rangle\end{aligned}$$

also as claimed. □

DEFINITION 2.1. Given a tree $T = (V, E)$, we say that a non-empty subset $U \subseteq V$ of vertices is a (T-)*vertex patch* if $[U] = U$ holds. Similarly, a non-empty subset $F \subseteq E$ of edges is called a (T-)*patch* if $\langle \bigcup F \rangle = F$ holds.

REMARK 2.7. Given a tree $T = (V, E)$ and a non-empty subset $F \subseteq E$, then F is a patch if and only if $(\bigcup F, F)$ is a subtree of T if and only if there exists a vertex patch U with $F = \langle U \rangle$ if and only if $F = \langle U \rangle$ holds for some arbitrary subset U of V.

Example.

2. In Fig. 2.2, we have depicted all T-patches (right) of the (up to isomorphism) unique binary tree T with six vertices (left).

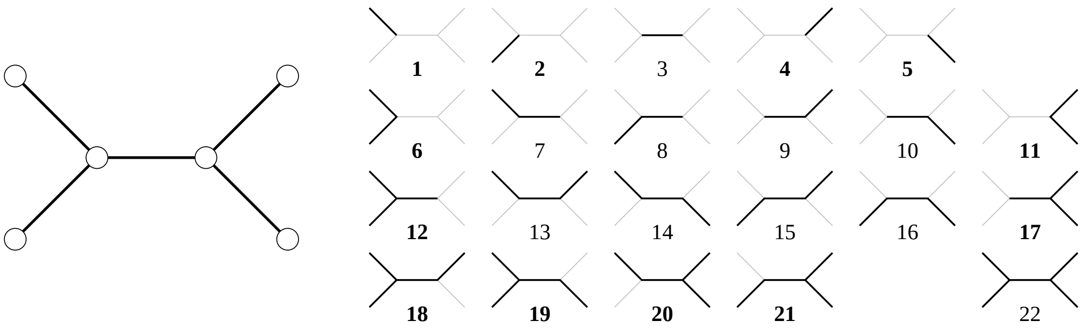


FIGURE 2.2. All patches (depicted right) of a given tree (depicted left).

There are many ways to generalize the concept of "patches" from trees to arbitrary connected graphs, see Example 6 for one such generalization.

LEMMA 2.8. *Let $T = (V, E)$ denote a tree. Given two vertex patches U_1, $U_2 \subseteq V$ with $U_1 \cap U_2 \neq \emptyset$, then both, the intersection $U_1 \cap U_2$ and the union $U_1 \cup U_2$ of U_1, U_2 are vertex patches, too. Similarly, $E_1 \cap E_2$ and $E_1 \cup E_2$ are both patches for any two patches E_1, $E_2 \subseteq E$ with $E_1 \cap E_2 \neq \emptyset$.*

PROOF. If u, $v \in U_j$ holds for some $j \in \{1, 2\}$ then $[u, v] \subseteq U_j \subseteq U_1 \cup U_2$ follows; while for $u_1 \in U_1 \setminus U_2$ and $u_2 \in U_2 \setminus U_1$ we choose an arbitrary vertex $w \in U_1 \cap U_2$, then $[u_1, w] \subseteq U_1$ and $[u_2, w] \subseteq U_2$ implies $[u_1, u_2] \subseteq [u_1, w] \cup [w, u_2] \subseteq U_1 \cup U_2$ by Lemma 2.2. The second part of the lemma follows immediately. $\square$

For a vertex $v \in V$ and a vertex patch $U \subseteq V$, we define $p(v \to U) = p_T(v \to U)$ to be the unique T-path p with

$$\alpha(p) = v \quad \text{and} \quad [p] \cap U = \{\omega(p)\}\,. \tag{2.13}$$

It is clear that such a path always exists. The uniqueness of $p(v \to U)$ is trivial in case $\#U = 1$, and it follows directly from the fact that $\#U \geq 2$ and, therefore, $\bigcup \langle U \rangle = U$ as well as p, $p' \in \mathbb{P}(T)$, $\alpha(p) = \alpha(p') = v$, $[p] \cap U = \{\omega(p)\}$, and $[p'] \cap U = \{\omega(p')\}$ implies $\langle p \rangle \cap \langle U \rangle = \langle p' \rangle \cap \langle U \rangle = \emptyset$ and, therefore,

$$\begin{aligned}
\emptyset &= \left\langle p \circ (p')^{-1} \right\rangle \cap \langle U \rangle \\
&= \Big\langle p(\omega(p) \to v) \circ p(v \to \omega(p')) \Big\rangle \cap \langle U \rangle \\
&= \left\langle p(\omega(p) \to \omega(p')) \right\rangle \cap \langle U \rangle \\
&= \left\langle p(\omega(p) \to \omega(p')) \right\rangle ,
\end{aligned}$$

that is, $\omega(p) = \omega(p')$.

We define

$$d(v,U) = d_T(v,U) := l(p(v\to U))\,,$$
$$\mathrm{gate}(v\to U) = \mathrm{gate}_T(v\to U) := \omega(p(v\to U))\,,$$

and, in case $d(v,U) > 0$ holds, we also define

$$\mathrm{edge}(v\to U) = \mathrm{edge}_T(v\to U) := e_{d(v,U)}(p(v\to U))$$

for v, U as above.

Note that $\mathrm{gate}(v\to U) \in [v,u]$ must hold for all such v, U and all vertices $u \in U$; this is a consequence of:

LEMMA 2.9. *Given a tree $T = (V,E)$, a vertex patch $U \subseteq V$, and vertices $v \in V$, $u \in U$; then*

$$\mathrm{gate}(v\to U) = \mathrm{med}(\mathrm{gate}(v\to U), v, u) \in [v,u]\,.$$

PROOF. The claim follows from

$$\begin{aligned}\{\mathrm{med}(\mathrm{gate}(v\to U), v, u)\} &\subseteq [v, \mathrm{gate}(v\to U)] \cap [\mathrm{gate}(v\to U), u]\\ &\subseteq [v, \mathrm{gate}(v\to U)] \cap U\\ &= \{\mathrm{gate}(v\to U)\}\,.\end{aligned}$$

□

For a patch $F \subseteq E$, we define $p(v\to F) := p(v\to \bigcup F)$, and $d(v\to F)$, $\mathrm{gate}(v\to F)$, and $\mathrm{edge}(v\to F)$ (in case $v \notin \bigcup F$) accordingly.

DEFINITION 2.2. Let $T = (V,E)$ denote a tree. A subset $F \subseteq \mathring{E}$ is called a (T-)*superpatch* if F and $\mathring{E} - F$ are T-patches.

In Example 2, we have numbered exactly those patches with bold numbers that also form a superpatch of the unique binary tree T' satisfying $\mathring{T}' = T$.

Given a superpatch $F \subseteq \mathring{E}$ of a tree $T = (V,E)$, there exists a unique vertex, called the *touching point* of F (and $\mathring{E} - F$) and denoted v_F, such that

$$\bigcup F \cap \bigcup(\mathring{E} - F) = \{v_F\}$$

holds. The existence and uniqueness of v_F can be deduced easily from the existence and uniqueness of a path in $\mathring{T} = (\bigcup \mathring{E}, \mathring{E})$ connecting any two vertices u, $v \in \bigcup \mathring{E}$.

We now define two binary relations on the subsets of V: Given U_1, $U_2 \subseteq V$, we define

$$U_1 \mid U_2 \iff \langle U_1\rangle \cap \langle U_2\rangle = \emptyset \quad\text{and}\quad U_1 \parallel U_2 \iff [U_1] \cap [U_2] = \emptyset\,.$$

We say that U_1 and U_2 are (*strongly*) *separated* if $U_1 \mid U_2$ (or $U_1 \parallel U_2$, respectively) holds. Obviously, $U_1 \mid U_2$ if and only if either $U_1 \parallel U_2$ or $\#([U_1] \cap [U_2]) = 1$ holds. We define $F_1 \mid F_2$ and $F_1 \parallel F_2$ for subsets F_1, $F_2 \subseteq E$ accordingly.

REMARK 2.10. Given a tree $T = (V,E)$, a vertex patch $U \subseteq V$, and two vertices v_1, $v_2 \in V$. Then $\mathrm{gate}(v_1\to U) = \mathrm{gate}(v_2\to U)$ holds if and only if $[v_1, v_2] \mid U$; and if v_1, $v_2 \notin U$ then $\mathrm{edge}(v_1\to U) = \mathrm{edge}(v_2\to U)$ holds if and only if $[v_1, v_2] \parallel U$.

2.4. Two equivalence relations on the set of vertices

DEFINITION 2.3. Given a tree $T = (V, E)$ and a subset $U \subseteq V$ of vertices, we can define an *equivalence relation* "$\approx_{(T,U)}$" (or "$\approx_U$" for short) on the set of vertices V by

$$u \approx_U v \iff [u, v]_T \cap U = \emptyset \quad \text{or} \quad u = v$$

for all u, $v \in V$.

The relation "$\approx_U$" is in fact an equivalence relation in view of $[u, v]_T = [v, u]_T$ and Lemma 2.2. For $v \in V$, let $[v]_{\approx U} \subseteq V$ denote the equivalence class (with respect to "$\approx_U$") containing v; we can easily see that $[v]_{\approx U}$ is a vertex patch. For any subset $V' \subseteq V$, let $V'/{\approx_U} := \{[v]_{\approx U} \mid v \in V'\}$ denote the corresponding set of equivalence classes; note that the equivalence classes in $V/{\approx_U}$ are exactly the vertices of the connected components of the graph $\left(V, E \cap \binom{V-U}{2}\right)$.

For a single vertex $v \in V$, we define $\approx_v := \approx_{\{v\}}$ and get

$$\#(V/{\approx_v}) = \deg_T(v) + 1 \quad \text{and} \quad [v]_{\approx v} = \{v\}\,.$$

Given two subsets U, $U' \subseteq V$, then obviously

$$u \approx_{(U \cup U')} v \iff u \approx_U v \quad \text{and} \quad u \approx_{U'} v \tag{2.14}$$

for all u, $v \in V$. Equation (2.14) implies:

REMARK 2.11. For a tree $T = (V, E)$, two sets U, U' with $U \subseteq U' \subseteq V$, and every vertex $v \in V$, we have $[v]_{\approx U'} \subseteq [v]_{\approx U}$.

In the special case that $U \subseteq V$ is a vertex patch, we infer that $u \approx_U v$ if and only if $u = v$ or $\{u, v\} \parallel U$ holds (cf. Remark 2.10). This implies:

REMARK 2.12. For a tree $T = (V, E)$, a vertex patch $U \subseteq V$, and vertices u, $v \in V - U$, we have

$$u \approx_U v \iff \mathrm{edge}(u \to U) = \mathrm{edge}(v \to U)\,.$$

LEMMA 2.13. *Given a tree $T = (V, E)$, two vertex patches U, $U' \subseteq V$ with non-empty intersection, and a vertex $v \in V$, then*

$$[v]_{\approx U} \subseteq [v]_{\approx U'} \quad \textit{or} \quad [v]_{\approx U'} \subseteq [v]_{\approx U} \quad \textit{(or both).}$$

PROOF. We know from Lemma 2.8 that $U \cup U'$ is a vertex patch. For arbitrary $v \in V$, we may suppose without loss of generality that

$$u := \mathrm{gate}(v \to U \cup U') \in U$$

holds. Given any $w \in [v]_{\approx U}$, it will be enough to show that $w \in [v]_{\approx U'}$ holds. Suppose $[v, w] \cap U' \neq \emptyset$; this implies $[v, w] \cap (U \cup U') \neq \emptyset$, and we infer from Lemma 2.9 that

$$u = \mathrm{gate}(v \to U \cup U') \in [v, w]$$

because for each $u' \in [v, w] \cap (U \cup U')$ we have

$$p(v \to w) = p(v \to u') \circ p(u' \to w)\,.$$

But $u \in [v, w] \cap U$ is a contradiction to $w \in [v]_{\approx U}$, so we are done. $\square$

For $F \subseteq E$, we define $\approx_F := \approx_{(\bigcup F)}$. In the special case of an inner edge $e \in \mathring{E}$ of a binary tree $T = (V, E)$, we note that $e = \{u, w\}$ induces a partition of V into four non-trivial sets, that is

$$\#(V/\approx_e) = 6 \quad \text{and} \quad [u]_{\approx e} = \{u\}, \quad [w]_{\approx e} = \{w\}\,.$$

In the other extreme $F = \mathring{E}$, we easily get $u \approx_{\mathring{E}} v$ if and only if $u = v$ for all vertices u, $v \in V$; and $\mathring{E}$ is the unique "inner patch" with this property, because $v \in \bigcup \mathring{E} - \bigcup F$ for some patch $F \subsetneqq \mathring{E}$ implies $[v]_{\approx F} \neq \{v\}$.

In case the subset F of $\mathring{E}$ is a superpatch, we can derive the following additional results:

REMARK 2.14. Let $T = (V, E)$ denote a tree and $F \subseteq \mathring{E}$ a superpatch. Then,

(i) there exists some equivalence class $\bar{v} \in V/\approx_F$ satisfying $\#\bar{v} > 1$;
(ii) given $v \in V$ with $\#[v]_{\approx F} > 1$, we must have $\#[v]_{\approx(\mathring{E}-F)} = 1$;
(iii) if v_F denotes the touching point of F, then

$$\big\{\bar{v} \in V/\approx_F \;\big|\; \#\bar{v} > 1\big\} \subseteq V/\approx_{v_F} - \big\{\{v_F\}\big\}\,;$$

(iv) if T is binary, then there exist exactly one or two equivalence classes $\bar{v} \in V/\approx_F$ satisfying $\#\bar{v} > 1$.

LEMMA 2.15. *Given a vertex patch $U \subseteq V$ of a binary tree $T = (V, E)$ and three vertices v_1, v_2, $v_3 \in V$ satisfying:*

$$v_i \not\approx_U v_j \quad \textit{for all} \quad 1 \leq i < j \leq 3 \tag{2.15}$$

Then $\operatorname{med}(v_1, v_2, v_3) \in U$ *holds.*

COROLLARY 2.16. *Given a superpatch $F \subseteq \mathring{E}$ of a binary tree $T = (V, E)$ and three vertices v_1, v_2, $v_3 \in V$ satisfying:*

$$v_i \not\approx_F v_j \quad \textit{and} \quad v_i \not\approx_{(\mathring{E}-F)} v_j \quad \textit{for all} \quad 1 \leq i < j \leq 3 \tag{2.16}$$

Then $\operatorname{med}(v_1, v_2, v_3) = v_F$ *holds.*

PROOF OF LEMMA 2.15. Let $m := \operatorname{med}(v_1, v_2, v_3)$. If there exist distinct u, $u' \in \{v_1, v_2, v_3\}$ with $[u, m] \cap U \neq \emptyset$ and $[u', m] \cap U \neq \emptyset$ then

$$m \in [\operatorname{gate}(u \to U), \operatorname{gate}(u' \to U)]$$

follows from Lemma 2.5 and hence, $m \in U$ since U is a vertex patch. Otherwise, there exist distinct u, $u' \in \{v_1, v_2, v_3\}$ with $[u, m] \cap U = \emptyset = [u', m] \cap U$, and

$$[u, u'] \cap U \subseteq ([u, m] \cup [m, u']) \cap U = ([u, m] \cap U) \cup ([u', m] \cap U) = \emptyset$$

implies $u \approx_U u'$ in contradiction to the assertion (2.15). $\square$

DEFINITION 2.4. Given a tree $T = (V, E)$ and a subset $F \subseteq E$ of edges we can define an *equivalence relation* "$\sim_{(T,F)}$" (or "$\sim_F$" for short) on the set of vertices V by

$$u \sim_F v \iff \langle u, v \rangle_T \cap F = \emptyset$$

for all u, $v \in V$.

The relation "$\sim_F$" is in fact an equivalence relation in view of $\langle v, v\rangle_T = \emptyset$, $\langle u, v\rangle_T = \langle v, u\rangle_T$, and again Lemma 2.2. For $v \in V$ we denote the equivalence class (with respect to "$\sim_F$") containing v by $[v]_{\sim F} \subseteq V$; note that $[v]_{\sim F}$ is a vertex patch. For some subset $U \subseteq V$ let $U/{\sim_F} := \big\{[v]_{\sim F} \,\big|\, v \in U\big\}$ denote the corresponding set of equivalence classes; note that the equivalence classes in $V/{\sim_F}$ are exactly the vertices of the connected components of the graph $\big(V, E - F\big)$.

Clearly, $u \approx_F v$ implies $u \sim_F v$ for vertices u, $v \in V$; given two subset F, $F' \subseteq E$ we also have

$$u \sim_{(F \cup F')} v \iff u \sim_F v \quad \text{and} \quad u \sim_{F'} v\,. \tag{2.17}$$

In the special case that F is a patch we note that $u \sim_F v$ holds if and only if $\{u, v\} \mid F$.

Remark 2.17. Given a patch $F \subseteq \mathring{E}$ of a binary tree $T = (V, E)$ and an equivalence class $\bar{v} \in V/{\sim_F}$ with $(\bar{v} - \{v\}) \notin V/{\approx_F}$ then $\bar{v}$ is the disjoint union of $\{v\}$ and exactly two equivalence classes $\bar{v}_1$, $\bar{v}_2 \in V/{\approx_F}$.

Remark 2.18. Given a superpatch $F \subseteq \mathring{E}$ of a tree $T = (V, E)$ then there exists a unique equivalence class $\bar{v} \in V/{\sim_F}$ with $\binom{\bar{v}}{2} \cap F \neq \emptyset$. If T is binary then this $\bar{v}$ is the unique equivalence class of cardinality greater or equal 4.

For a tree $T = (V, E)$ and vertices u, $v \in V$ with $\{u, v\} \in E$ we use the notation

$$V(u, v) := [v]_{\sim \{u,v\}} \tag{2.18}$$

where $V(u, v)$ contains all vertices of that connected component of $(V, E - \{u, v\})$ containing v. We easily infer

$$V = V(u, v) \uplus V(v, u) \quad \text{and} \quad E = \langle V(u, v)\rangle \uplus \big\{\{u, v\}\big\} \uplus \langle V(v, u)\rangle\ . \tag{2.19}$$

Remark 2.19. Given a tree $T = (V, E)$, an edge $\{u, v\} \in E$, and disjoint vertex patches U_1, $U_2 \in V$ with

$$V = U_1 \uplus U_2 \quad \text{and} \quad E = \langle U_1\rangle \uplus \big\{\{u, v\}\big\} \uplus \langle U_2\rangle$$

then $\{U_1, U_2\} = \{V(u, v), V(v, u)\}$ holds.

CHAPTER 3

X-Trees and Split Systems

As mentioned in Section 1.2, we want to "attach" the objects of interests to some tree: To this end, we introduce X-trees and their equivalent, compatible split systems. We have indicated in the introduction that, in order to formally define tree amalgamation, we must first define how to induce subtrees. It is well-known (and somehow straightforward) how to obtain an induced subtree via a subset of taxa. In Section 3.4, we present a different approach by restricting the "information" of an X-tree: We choose a subset of inner edges and construct an induced tree representing exactly the information that corresponds to these edges. Finally, we present a third approach that somehow combines the previous two constructions.

In Section 3.5, we show that the problem of tree amalgamation is in fact equivalent to reconstructing trees from quartet splits.

3.1. Definitions

An X-*tree* is a triple $T = (V, E; \phi)$ such that X is a finite set, (V, E) is a tree, and $\phi : X \to V$ is a map satisfying the condition

$$v \in \phi(X) \quad \text{holds for all} \quad v \in V \quad \text{with} \quad \deg_T(v) \leq 2 \tag{3.1}$$

where $\deg_T(v)$ denotes the degree of the vertex v in (V, E).[1] If ϕ is a bijection from X onto the set V_1 of leaves of T, we say that T is a *phylogenetic* X-tree; if, in addition, the underlying tree (V, E) is binary then we will say that T is a *binary* X-tree. Two X-trees $T = (V, E; \phi)$ and $T' = (V', E'; \phi')$ are *isomorphic* if there exists a bijection $\alpha : V \xrightarrow{\sim} V'$ inducing a bijection between E and E' and satisfying $\phi' = \alpha \circ \phi$ (in which case there is exactly one such map α). From (2.8) we infer that

$$\#\mathring{E} \leq \#X - 3 \tag{3.2}$$

must hold for a phylogenetic X-tree $T = (V, E; \phi)$ with equality if and only if T is binary.

Example.

3. Let $X := \{1, \ldots, 9\}$; the tree (V, E) depicted in Fig. 2.1 on page 9 together with the mapping $\phi := id_X$ is a phylogenetic X-tree. It is not binary because of the inner vertices b and e.

[1]In the following, we will denote an X-tree by $T = (V, E; \phi)$, while the underlying tree will be denoted (V, E). It is obvious how definitions of other mathematical objects like paths, distances etc. transfer from trees to X-trees.

Clearly, every X-tree $T = (V, E; \phi)$ gives rise to a unique phylogenetic X-tree $T_{\text{phyl}} = (V_{\text{phyl}}, E_{\text{phyl}}; \phi_{\text{phyl}})$ defined by

$$\begin{aligned} Y_{\text{phyl}} &:= \left\{ x \in X \;\middle|\; \deg_T(\phi(x)) \neq 1 \text{ or } \phi(x) \in \phi(X - \{x\}) \right\}, \\ V_{\text{phyl}} &:= V \,\dot\cup\, Y_{\text{phyl}}, \\ E_{\text{phyl}} &:= E \,\dot\cup\, \left\{ \{\phi(x), x\} \;\middle|\; x \in Y_{\text{phyl}} \right\}, \end{aligned} \tag{3.3}$$

$$\text{and} \quad \phi_{\text{phyl}} : X \to V_{\text{phyl}} \;:\; x \mapsto \begin{cases} x & \text{if } x \in Y_{\text{phyl}}, \\ \phi(x) & \text{otherwise.} \end{cases}$$

A *split* of X is a subset $\{A, B\} \subseteq \mathcal{P}(X)$ such that A, B is a bipartition of X into two non-empty, disjoint subsets; a *partial split* of X is a split of some non-empty subset of X. We let $\mathcal{S}(X)$ (respectively $\mathcal{S}_{\text{part}}(X)$) denote the set of splits of X (respectively partial splits of X). A split $\{A, B\}$ is *trivial* if $\min\{\#A, \#B\} = 1$. A partial split $S_1 = \{A_1, B_1\}$ is said to *extend* a partial split $S_2 = \{A_2, B_2\}$ if

$$S_2 = \{(A_2 \cup B_2) \cap A_1, (A_2 \cup B_2) \cap B_1\}$$

holds (that is, $A_2 \subseteq A_1$ and $B_2 \subseteq B_1$, or $A_2 \subseteq B_1$ and $B_2 \subseteq A_1$) in which case we will also write $S_2 \leq S_1$.

Two splits S_1, S_2 are *compatible* if there exists $A_1 \in S_1$, $A_2 \in S_2$ with $A_1 \cap A_2 = \emptyset$. A collection of splits (or *split system*) $\mathcal{S} \subseteq \mathcal{S}(X)$ is *compatible* if any two splits from $\mathcal{S}$ are compatible.

Given a partial split $S = \{A, B\}$, we define the *support* of S by $\underline{S} := A \cup B$, and for every $x \in \underline{S}$, we define $S(x) := A$ if $x \in A$ and $S(x) := B$ otherwise; for a collection of partial splits $\mathcal{S} \subseteq \mathcal{S}_{\text{part}}(X)$, we define $\underline{\mathcal{S}} := \bigcup_{S \in \mathcal{S}} \underline{S}$.

For natural numbers i, j, let

$$\mathcal{S}_{i,j}(X) := \left\{ \{A, B\} \in \mathcal{S}_{\text{part}}(X) \;\middle|\; \{\#A, \#B\} = \{i, j\} \right\}. \tag{3.4}$$

A partial split $Q \in \mathcal{Q}(X) := \mathcal{S}_{2,2}(X)$ is called a *quartet split*. We use $Q = xy|wz$ as shorthand for $Q = \{\{x, y\}, \{w, z\}\}$; more generally, we use $S = x_1 \ldots x_j | y_1 \ldots y_k$ as shorthand for a split $S = \{\{x_1, \ldots, x_j\}, \{y_1, \ldots, y_k\}\}$.

3.2. Basic observations

There is a canonical and useful one-to-one correspondence between (isomorphism classes of) X-trees and set systems, due to Buneman [Bun71], which we now recall: Each edge e in an X-tree $T = (V, E; \phi)$ gives rise to a split of X — simply delete e from E and apply ϕ^{-1} to the two connected components of the resulting graph $(V, E - \{e\})$ (or, equivalently, apply ϕ^{-1} to the two equivalence classes in $V/\!\sim_e$) to obtain a split, which we will call a T-split and denote by $S[e] = S_T[e]$. Using the notation of (2.18), we get for $e = \{u, v\}$

$$S_T[e] = \left\{ \phi^{-1}(V(u, v)), \phi^{-1}(V(v, u)) \right\}. \tag{3.5}$$

Let $\mathcal{S}[T]$ denote the set of all T-splits. It is easily checked that distinct edges induce distinct splits and that the set $\mathcal{S}[T]$ is is compatible. Furthermore, we have $\#\mathcal{S}[T] \leq 2\#X - 3$ with equality precisely if T is a binary X-tree.

Lemma 3.1 (Buneman). *The map $T \rightsquigarrow \mathcal{S}[T]$ induces bijections between:*

(i) *the set of (isomorphism classes of) X-trees and the set of compatible split systems $\mathcal{S} \subseteq \mathcal{S}(X)$;*
(ii) *the set of (isomorphism classes of) phylogenetic X-trees and the set of all those compatible split systems $\mathcal{S} \subseteq \mathcal{S}(X)$ that contain all trivial splits of X;*
(iii) *the set of (isomorphism classes of) binary X-trees and the set of compatible split systems $\mathcal{S} \subseteq \mathcal{S}(X)$ for which $\#\mathcal{S} = 2\#X - 3$ holds, or — equivalently — the set of maximal compatible split systems $\mathcal{S} \subseteq \mathcal{S}(X)$.*

This correspondence between X-trees and compatible split systems implies $\#\mathcal{S} \leq 2\#X - 3$ for every compatible split system $\mathcal{S} \subseteq \mathcal{S}(X)$, and $\mathcal{S}$ is maximal (due to inclusion) if and only if equality holds.

Lemma 3.1 also provides a convenient partial order on the set of (isomorphism classes of) X-trees: We write $T' \leq T$ for two X-trees T, T' precisely if $\mathcal{S}[T'] \subseteq \mathcal{S}[T]$. Loosely speaking, $T' \leq T$ states that T' can be obtained from T by contracting edges, and identifying corresponding vertices. To obtain a formal description of this fact, we define:

Definition 3.1. Given two X-trees $T = (V, E; \phi)$ and $T' = (V', E'; \phi')$, a *contraction map* α from T to T' is a surjective map $\alpha : V \to V'$ satisfying (i) $\phi' = \alpha \circ \phi$, (ii) $\alpha(u) = \alpha(w)$ for $u, w \in V$ implies $\alpha(u) = \alpha(v)$ for all $v \in [u, w]_T$, and (iii)

$$E' = \big\{\{\alpha(u), \alpha(v)\} \bigm| \{u, v\} \in E \quad \text{and} \quad \alpha(u) \neq \alpha(v)\big\}\,. \tag{3.6}$$

Clearly, a surjective map $\alpha : V \to V'$ satisfying $\phi' = \alpha \circ \phi$ and (3.6) is a contraction map from T to T' if and only if $\alpha^{-1}(v')$ is a T-vertex patch for all $v' \in V'$ if and only if $\alpha^{-1}(U')$ is a T-vertex patch for all T'-vertex patches $U' \subseteq V'$.

Lemma 3.2. *Given two X-trees T and T', then $T' \leq T$ holds if and only if there exists a contraction map from T to T'.*

Proof. Let $T = (V, E; \phi)$ and $T' = (V', E'; \phi')$ denote two X-trees with $T' \leq T$. We suppose that $\#(\mathcal{S}[T] - \mathcal{S}[T']) = 1$ holds; let $e = \{u_1, u_2\} \in E(T)$ denote the unique edge with $S[e] \notin \mathcal{S}[T']$. For $V^* := V - \{u_1, u_2\} \,\dot\cup\, \{w\}$ we define a map

$$\psi : V \to V^* \;:\; v \mapsto \begin{cases} v & \text{if } v \notin \{u_1, u_2\} \\ w & \text{otherwise} \end{cases}$$

and an X-tree $T^* = (V^*, E^*; \phi^*)$ by

$$\begin{aligned} &E^* := \big\{\{\psi(v_1), \psi(v_2)\} \bigm| \{v_1, v_2\} \in E - \{e\}\big\} \\ \text{and} \quad &\phi^* : X \to V^* \;:\; x \mapsto \psi(\phi(x))\,. \end{aligned}$$

It is obvious from the construction that T^* is an X-tree, and that $\mathcal{S}[T^*] = \mathcal{S}[T] - \{S_T[e]\}$ and hence, $T^* \cong T'$ holds. So, there exists a bijection $\alpha^* : V^* \xrightarrow{\sim} V'$ inducing a bijection between E^* and E' and satisfying $\phi' = \alpha^* \circ \phi^*$. We define a map

$$\alpha : V \to V' \;:\; v \mapsto \begin{cases} \alpha^*(v) & \text{if } v \notin \{u_1, u_2\}, \\ \alpha^*(w) & \text{otherwise.} \end{cases}$$

This map satisfies $\phi' = \alpha \circ \phi$ as well as (3.6), and

$$\{\alpha^{-1}(v') \bigm| v' \in V'\} = \big\{\{v\} \bigm| v \in V - \{u_1, u_2\}\big\} \,\dot\cup\, \big\{\{u_1, u_2\}\big\}$$

finally proves that α is a contraction map from T to T'.

For the first part of the proof, it remains to show that, given X-trees T_1, T_2, T_3 with $T_3 \leq T_2 \leq T_1$ and associated contraction maps $\alpha_1 : V(T_1) \to V(T_2)$ and $\alpha_2 : V(T_2) \to V(T_3)$, then $\alpha := \alpha_2 \circ \alpha_1$ satisfies (3.6) for $E = E(T_1)$ and $E' = E(T_3)$. Indeed,

$$\begin{aligned} E(T_3) &= \big\{\{\alpha_2(u'), \alpha_2(v')\} \bigm| \{u', v'\} \in E(T_2) \text{ and } \alpha_2(u') \neq \alpha_2(v')\big\} \\ &= \big\{\{\alpha_2(\alpha_1(u)), \alpha_2(\alpha_1(v))\} \bigm| \{u, v\} \in E(T_1) \text{ and } \alpha_1(u) \neq \alpha_1(v) \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{and } \alpha_2(\alpha_1(u)) \neq \alpha_2(\alpha_1(v))\big\} \\ &= \big\{\{\alpha(u), \alpha(v)\} \bigm| \{u, v\} \in E(T_1) \text{ and } \alpha(u) \neq \alpha(v)\big\} \end{aligned}$$

since $\alpha_1(u) = \alpha_1(v)$ obviously implies $\alpha_2(\alpha_1(u)) = \alpha_2(\alpha_1(v))$ for u, $v \in V(T_1)$. Using induction, this proves the first part of the lemma.

Vice versa, suppose that there exists a contraction map α from $T = (V, E; \phi)$ to $T' = (V', E'; \phi')$. Let e' denote an arbitrary edge of T', then we easily see that there exists an edge e of T with $e' = \alpha(e)$. We claim that $S_T[e] = S_{T'}[e']$ holds: We can easily see that $[u, v]_T$ is mapped to $[\alpha(u), \alpha(v)]_{T'}$ by α for arbitrary vertices u, v of T, so we infer that a vertex patch U of T is mapped to a vertex patch $\alpha(U)$ of T'. We choose u, $v \in V$ with $e = \{u, v\}$; using the notation of (2.18) we define

$$U_1 := V(u, v) \quad \text{and} \quad U_2 := V(v, u)$$

then $U'_j := \alpha(U_j)$ for $j = 1, 2$ are vertex patches of T' with $U'_1 \uplus U'_2 = V'$. We easily see $e' \notin \langle U'_1 \rangle \cup \langle U'_2 \rangle$ what, together with Remark 2.19, implies

$$S_{T'}[e'] = \big\{(\phi')^{-1}(U'_1), (\phi')^{-1}(U'_2)\big\} = \big\{\phi^{-1}(U_1), \phi^{-1}(U_2)\big\} = S_T[e]$$

because

$$x \in \phi^{-1}(U_j) \iff \phi(x) \in U_j \iff \alpha \circ \phi(x) \in U'_j \iff x \in (\phi')^{-1}(U'_j)$$

holds for $x \in X$ and $j = 1, 2$. This proves that $T' \leq T$ must hold. $\square$

3.3. Induced Y-trees and split systems

Given an X-tree $T = (V, E; \phi)$ and a non-empty subset $Y \subseteq X$, we obtain an induced Y-tree as follows: First construct the minimal subtree of (V, E) that connects all vertices from $\phi(Y)$. Then make this tree "homeomorphically irreducible" by replacing each maximal path of degree-two vertices from $V - \phi(Y)$ by a single edge (and deleting the superfluous vertices and edges) to obtain a tree. Formally, we define the subtree $T' := \big([\phi(Y)], \langle\phi(Y)\rangle\big)$ and, to remove the superfluous vertices and edges,

$$\begin{aligned} V_Y &:= \big\{v \in V(T') \bigm| \deg_{T'}(v) \geq 3 \text{ or } v \in \phi(Y)\big\} \\ \text{and} \quad E_Y &:= \Big\{\{u, v\} \in \tbinom{V_Y}{2} \Bigm| [u, v]_{T'} \cap V_Y = \{u, v\}\Big\} . \end{aligned}$$

The restriction $\phi_Y := \phi|_Y : Y \to V_Y$ satisfies condition (3.1) with X, T replaced by Y, $T|_Y := (V_Y, E_Y; \phi_Y)$. We call the resulting Y-tree $T|_Y$ the *induced* (Y-)subtree of T. Note that $V(T|_Y) \subseteq V(T)$ holds according to this definition.

REMARK 3.3. Given a binary X-tree $T = (V, E; \phi)$ and a subset $Y = \{x_1, x_2, x_3\}$ of X of cardinality three, we construct the binary Y-tree $T|_Y$ and get

$$V(T|_Y) = \phi(Y) \cup \{\mathrm{med}(\phi(x_1), \phi(x_2), \phi(x_3))\}.$$

For a split $S = \{A, B\} \in \mathcal{S}(X)$ and a non-empty subset $Y \subseteq X$ we define $S|_Y := \{A \cap Y, B \cap Y\}$; note that $S|_Y$ is not a split in general because $A \cap Y = \emptyset$ or $B \cap Y = \emptyset$ might hold. For a split system $\mathcal{S} \subseteq \mathcal{S}(X)$ we define the *induced split system*

$$\mathcal{S}|_Y := \bigl\{S|_Y \bigm| S \in \mathcal{S}\bigr\} \setminus \bigl\{\{Y, \emptyset\}\bigr\},$$

and it is easily seen that this set is in fact a split system.

With these definitions, we can extend the partial order "$\leq$" on X-trees in the following way: Given an X-tree T and a Y-tree T' with $Y \subseteq X$, we write $T' \leq T$ if $T' \leq T|_Y$ holds, and we will say that T *displays* T' in this case.

In the following, we will formally state (and prove) some of the properties of induced trees and split systems that are frequently used in this paper. To begin with, it is easy to see that $(S|_Y)\big|_Z = S|_Z$ holds for sets $Z \subseteq Y \subseteq X$ and a split $S \in \mathcal{S}(X)$; analogously, $(\mathcal{S}|_Y)\big|_Z = \mathcal{S}|_Z$ holds for a split system $\mathcal{S} \subseteq \mathcal{S}(X)$. Furthermore, given an X-tree T and a non-empty subset $Y \subseteq X$ then

$$\mathcal{S}[T|_Y] = \mathcal{S}[T]\big|_Y = \{S' \in \mathcal{S}(Y) \bigm| S' \leq S \text{ for some } S \in \mathcal{S}[T]\}. \tag{3.7}$$

Finally, the restriction preserves the order of X-trees, that is, $T'|_Y \leq T|_Y$ holds for X-trees T, T' with $T' \leq T$ and a non-empty subset $Y \subseteq X$.

REMARK 3.4. Given an X-tree T, a non-empty subset $Y \subseteq X$, and an edge $\{u, v\}$ of $T|_Y$. Then

$$\bigl(S_T[e]\bigr)\big|_Y = S_{T|_Y}[\{u, v\}]$$

holds for an edge e of T if and only if $e \in \langle u, v\rangle_T$.

LEMMA 3.5. *Given a finite set X, a compatible split system $\mathcal{S} \subseteq \mathcal{S}(X)$, and a subset $Y \subseteq X$ then*

$$\#\mathcal{S}|_Y \geq \#\mathcal{S} - 2(\#X - \#Y).$$

COROLLARY 3.6. *Let X, Y denote finite set with $Y \subseteq X$. If $\mathcal{S} \subseteq \mathcal{S}(X)$ is a maximal compatible split system then $\mathcal{S}|_Y$ is maximal, too. If T is a binary X-tree then $T|_Y$ is binary.*

For the proof of Lemma 3.5 we use the following

LEMMA 3.7. *Given a finite set X, a compatible split system $\mathcal{S} \subseteq \mathcal{S}(X)$, and an element $x \in X$ we define $Y := X - \{x\}$. Then, there exists at most one pair of splits S_1, $S_2 \in \mathcal{S}$ such that $S_1\big|_Y = S_2\big|_Y$ holds.*

PROOF. Suppose there exist splits S_1, S_2, S_3, $S_4 \in \mathcal{S}$ with $S_1 \neq S_2$, $S_3 \neq S_4$ such that $S_1\big|_Y = S_2\big|_Y$ and $S_3\big|_Y = S_4\big|_Y$ holds. Then, there must exist subsets A, B, C, $D \subseteq Y$ such that

$$S_1 = \{A, B \cup \{x\}\}, \qquad S_2 = \{A \cup \{x\}, B\},$$
$$S_3 = \{C, D \cup \{x\}\}, \qquad \text{and} \quad S_4 = \{C \cup \{x\}, D\}.$$

Since S_1 and S_3 are compatible, one of the intersections $A\cap C$, $A\cap D$, $B\cap C$ must be empty. Suppose that $A\cap C=\emptyset$, then $A\cap D\neq\emptyset$ and $B\cap C\neq\emptyset$, and we infer from the compatibility of S_2 and S_4 that $B\cap D=\emptyset$ must hold — but this already implies $A=D$ and $B=C$. For the remaining two cases, an analogous argumentation finally leads to $\#\{S_1,S_2,S_3,S_4\}=2$. □

PROOF OF LEMMA 3.5. Using induction, we may assume that $Y=X-\{x\}$ holds for some $x\in X$. By Lemma 3.7 there are at most two distinct splits in $\mathcal{S}$ inducing the same split in Y. We infer

$$\begin{aligned}\#\mathcal{S}|_Y &\geq \#\mathcal{S}-\#\big\{(S,S')\in\tbinom{\mathcal{S}}{2}\;\big|\;S|_Y=S'|_Y\big\}-\#\big\{S\in\mathcal{S}\;\big|\;S|_Y=\{Y,\emptyset\}\big\}\\ &\geq \#\mathcal{S}-2\end{aligned}\tag{3.8}$$

by definition of $\mathcal{S}|_Y$. □

The following lemma will be useful for the proof of the Tree Joining Theorem in Chapter 4 — the claim is quite intuitive when dealing with X-trees instead of split systems.

LEMMA 3.8. *Given finite sets X,Y with $Y\subseteq X$, let $\mathcal{S}\subseteq\mathcal{S}(X)$ and $\mathcal{S}'\subseteq\mathcal{S}(Y)$ denote two compatible split systems with $\mathcal{S}\big|_Y\subseteq\mathcal{S}'$. Then, there exists a compatible split system $\mathcal{S}_0\subseteq\mathcal{S}(X)$ with*

$$\mathcal{S}\subseteq\mathcal{S}_0\quad\textit{and}\quad\mathcal{S}_0\big|_Y=\mathcal{S}'\,.\tag{3.9}$$

PROOF. Suppose there exists some split $S'=\{A',B'\}\in\mathcal{S}'$ with $S'\notin\mathcal{S}\big|_Y$. We define the collection of clusters

$$\mathcal{C}:=\big\{C\in\bigcup\mathcal{S}\;\big|\;A'\cap C\neq\emptyset\quad\text{and}\quad B'\cap C=\emptyset\big\}\tag{3.10}$$

and set $A:=(\bigcup\mathcal{C})\cup A'$, $B:=X-A$. Then $S:=\{A,B\}$ is a split of X with $S|_Y=S'$ what implies $S\notin\mathcal{S}$.

We claim that $\mathcal{S}\cup\{S\}$ is compatible: To this end, choose an arbitrary $S_1\in\mathcal{S}$. If there exists $A_1\in S_1$ with $A_1\cap B'=\emptyset$ then either $A_1\cap A'\neq\emptyset$ or $A_1\cap Y=\emptyset$ must hold. In the first case, we infer $A_1\in\mathcal{C}$ and, hence, $A_1\subseteq\bigcup\mathcal{C}\subseteq A$, so S_1 and S are compatible and we are done. In the second case, suppose that $A_1\cap A\neq\emptyset$ holds. Then, there must exist some cluster $C\in\mathcal{C}$ with $A_1\cap C\neq\emptyset$. We know $B_1\cap C\neq\emptyset$ as well $B_1\cap(X-C)\neq\emptyset$ in view of $Y\subseteq B_1$, so $A_1\cap(X-C)=\emptyset$ must hold, since $\{A_1,B_1\}$ and $\{C,X-C\}$ are compatible. This implies $A_1\subseteq C\subseteq A$ and, hence, we are done as well.

In the following, we may suppose that

$$A_1\cap B'\neq\emptyset\quad\text{as well as}\quad B_1\cap B'\neq\emptyset$$

holds for $S_1=\{A_1,B_1\}$; this implies $S_1\big|_Y\neq\{Y,\emptyset\}$ and, hence, $S_1\big|_Y\in\mathcal{S}'$ in view of $\mathcal{S}\big|_Y\subseteq\mathcal{S}'$. We conclude that — without loss of generality — $B_1\cap A'=(B_1\cap Y)\cap A'=\emptyset$ holds, because $S_1\big|_Y$ and S' are compatible. For every cluster $C\in\mathcal{C}$ we get $A_1\cap C\neq\emptyset$ in view of $A'\subseteq A_1$, and $A_1\cap(X-C)\neq\emptyset$ as well as $B_1\cap(X-C)\neq\emptyset$ in view of $B'\subseteq(X-C)$, so $B_1\cap C=\emptyset$ must hold since $\{A_1,B_2\}$ and $\{C,X-C\}$ are compatible. This implies $B_1\cap\bigcup\mathcal{C}=\emptyset$ and $B_1\cap A=B_1\cap(\bigcup\mathcal{C}\cup A')=\emptyset$ what finally shows that S_1 and S are compatible.

Replacing $\mathcal{S}$ by $\mathcal{S} \cup \{S\}$ and iterating this process, we end up with a compatible split system $\mathcal{S}_0 \subseteq \mathcal{S}(X)$ meeting (3.9). $\square$

Following the line of though of this proof, we may even suppose that $\mathcal{S}_0$ of Lemma 3.8 satisfies:

$$\#\mathcal{S}_0 - \#\mathcal{S}_0|_Y = \#\mathcal{S} - \#\mathcal{S}|_Y \tag{3.11}$$

COROLLARY 3.9. *Given finite sets X, Y with $Y \subseteq X$, let $\mathcal{S} \subseteq \mathcal{S}(X)$ and $\mathcal{S}' \subseteq \mathcal{S}(Y)$ denote two compatible split systems such that $\mathcal{S}\big|_Y \subseteq \mathcal{S}'$. If we further suppose $\mathcal{S}'$ to be maximal, then there exists a maximal compatible split system $\mathcal{S}_0 \subseteq \mathcal{S}(X)$ satisfying* (3.9)*.*

PROOF. By Lemma 3.8 there exists a compatible split system $\mathcal{S}_0^* \subseteq \mathcal{S}(X)$ with $\mathcal{S} \subseteq \mathcal{S}_0^*$ and $\mathcal{S}_0^*|_Y = \mathcal{S}'$. Let $\mathcal{S}_0 \subseteq \mathcal{S}(X)$ denote an arbitrary maximal compatible split system with $\mathcal{S}_0^* \subseteq \mathcal{S}_0$. We know that $\mathcal{S}_0\big|_Y$ is compatible, but $\mathcal{S}_0^*\big|_Y = \mathcal{S}'$ is a maximal compatible split system and, hence, $\mathcal{S}_0\big|_Y = \mathcal{S}'$. $\square$

3.4. Inducing subtrees via patches and subsets of edges

We have seen in the previous section how to derive an induced Y-tree $T|_Y$ from an X-tree T via a subset of taxa $Y \subseteq X$. While this construction is well known and somehow straightforward, we will now show how to construct an induced tree via a *patch of edges*.

Let $T = (V, E; \phi)$ denote a phylogenetic X-tree, and $F \subseteq \mathring{E}$ an arbitrary subset. In Section 2.4, we have defined an equivalence relation "$\approx_F$" on the set of vertices of (V, E). It is straightforward to induce an equivalence relation on X (also denoted "$\approx_F$") via

$$x \approx_F y \iff \phi(x) \approx_F \phi(y)$$

for all x, $y \in X$. For any $x \in X$, let $[x]_{\approx F} \subseteq X$ denote the equivalence class of $\approx_F$ containing x, and for some subset $Y \subseteq X$ let $Y/{\approx_F} := \big\{[x]_{\approx F} \bigm| x \in Y\big\}$ denote the corresponding set of equivalence classes.

The following is a simple consequence of Remark 2.14:

REMARK 3.10. Let $T = (V, E; \phi)$ denote an X-tree and $F \subseteq \mathring{E}$ a superpatch. Then,

(i) there exists some equivalence class $\bar{x} \in X/{\approx_F}$ satisfying $\#\bar{x} > 1$;
(ii) given $x \in X$ with $\#[x]_{\approx F} > 1$ we must have $\#[x]_{\approx(\mathring{E}-F)} = 1$;
(iii) if v_F denotes the touching point of F then

$$\big\{\bar{x} \in X/{\approx_F} \bigm| \#\bar{x} > 1\big\} \subseteq X/{\approx_{v_F}};$$

(iv) if T is binary then there exists exactly one or two equivalence classes $\bar{x} \in X/{\approx_F}$ such that $\#\bar{x} > 1$.

For the moment, we will limit our attention to patches $F \subseteq \mathring{E}$ and how we can induce subtrees via those patches. We will see later on how to generalize these constructions for *arbitrary* sets $F \subseteq \mathring{E}$.

DEFINITION 3.2. Given a phylogenetic X-tree $T = (V, E; \phi)$ and a patch $F \subseteq \mathring{E}$, we define the induced X_F-tree $T_F = (V_F, E_F; \phi_F)$ for $X_F := X/{\approx_F}$ by:

$$V_F := \bigcup F \cup X_F \tag{3.12}$$

$$E_F := F \cup \Big\{ \big\{[x]_{\approx F}, \mathrm{gate}(\phi(x) \rightarrow F)\big\} \;\big|\; x \in X \Big\} \tag{3.13}$$

$$\phi_F := id_{X_F} \tag{3.14}$$

Since T is phylogenetic we infer from Remark 2.12 that $\mathrm{gate}(\phi(y) \rightarrow F) = \mathrm{gate}(\phi(x) \rightarrow F)$ must hold for all $y \in [x]_{\approx F}$, so (V_F, E_F) is a tree; it is easy to check that T_F is in fact a phylogenetic X_F-tree, and that $\mathrm{Int}(T_F) = F$ holds. In fact, $\mathrm{edge}(\phi(y) \rightarrow F) = \mathrm{edge}(\phi(x) \rightarrow F)$ if and only if $x \approx_F y$ for $x, y \in X$ by Remark 2.12 and we infer:

LEMMA 3.11. *Given a binary X-tree T and a patch $F \subseteq \mathrm{Int}(T)$ then T_F is a binary X_F-tree.*

More generally, we can "combine" the two methods of deriving induced subtrees as described above into one:

DEFINITION 3.3. Given a phylogenetic X-tree $T = (V, E; \phi)$, a patch $F \subseteq \mathring{E}$, and a subset $Y \subseteq X$, we define $T_{(F,Y)} = (V_{(F,Y)}, E_{(F,Y)}; \phi_{(F,Y)})$ by:

$$V_{(F,Y)} := \bigcup F \cup Y \tag{3.15}$$

$$E_{(F,Y)} := F \cup \Big\{ \big\{x, \mathrm{gate}(\phi(x) \rightarrow F)\big\} \;\big|\; x \in Y \Big\} \tag{3.16}$$

$$\phi_{(F,Y)} := id_Y \tag{3.17}$$

It is easy to see that $(V_{(F,Y)}, E_{(F,Y)})$ is a tree. But $T_{(F,Y)}$ is not a Y-tree in general, because there might be inner vertices of degree one or two not contained in $\phi(Y) = Y$. But if

$$\#\{e \in F \mid v \in e\} + \#\big([v]_{\sim F} \cap \phi(Y)\big) \geq 3 \tag{3.18}$$

holds for all "inner" vertices $v \in \bigcup F$ then $T_{(F,Y)}$ is in fact a *phylogenetic* Y-tree; where the left side of the inequality equals the degree of v in $T_{(F,Y)}$. A useful sufficient condition for $T_{(F,Y)}$ to be a phylogenetic Y-tree will be given in Lemma 3.12 below.

If $T_{(F,Y)}$ is a phylogenetic Y-tree, then it is easy to check that $\mathrm{Int}(T_{(F,Y)}) = F$ holds; but the fact that T is binary does obviously not imply that $T_{(F,Y)}$ is binary. Finally, $T_{(F,Y)} \leq T$ or, equivalently, $T_{(F,Y)} \leq T|_Y$ must hold by construction of $T_{(F,Y)}$.

Under what conditions do all three methods of constructing induced subtrees lead to the "same" tree? For the Y-trees $T|_Y$ and $T_{(F,Y)}$ only, this question is comparatively easy to answer, as we will see in Lemma 3.13 below. It is clear that there cannot exist an isomorphism between T_F and $T|_Y$ (or T_F and $T_{(F,Y)}$, respectively), since $T|_Y$ and $T_{(F,Y)}$ are Y-trees, while T_F is an X_F-tree. But if we consider the canonical projection $\beta_F : X \rightarrow X_F \;:\; x \mapsto [x]_{\approx F}$, under what conditions do we get canonical 1-to-1 correspondences between $T|_Y$, $T_{(F,Y)}$, and T_F simultaneously?

Obviously, $Y/{\approx_F} = X_F$ must hold to satisfy $\beta_F(Y) = X_F$. The above correspondences exist if and only if the subset $Y \subseteq X$ contains exactly one representative

$x \in X$ for every equivalence class $[x]_{\approx F}$ of X_F or, equivalently, if $Y \subseteq X$ is a *minimal* set with the property $Y/{\approx_F} = X_F$. Then, we can easily infer

$$T|_Y \cong T_{(F,Y)} \cong (V_F, E_F; \phi_F \circ \beta_F)\,, \tag{3.19}$$

where the isomorphism $\alpha_1 : V_{(F,Y)} \to V_Y$ is given by $\alpha_1|_F := id_F$ and $\alpha_1|_Y := \phi_Y$; while the isomorphism $\alpha_2 : V_{(F,Y)} \to V_F$ is $\alpha_2|_F := id_F$, and $\alpha_2(x) := \beta_F(x)$ for all $x \in Y$. If $Y \subseteq X$ is not a minimal set with $Y/{\approx_F} = X_F$ then we infer $\#X_F < \#Y$ and $\#F = \#\operatorname{Int}(T_{(F,Y)}) < \#\operatorname{Int}(T|_Y)$ what surely implies that there exist no such 1-to-1 correspondences.

Now, suppose that $F \subseteq \mathring{E}$ is an *arbitrary* non-empty subset and not necessarily a patch.[2] Then, we can still define induced subtrees T_F and $T_{(F,Y)}$, replacing equations (3.12) and (3.13) by

$$V_F := V/{\sim_F} \,\dot\cup\, X/{\approx_F} \tag{3.20}$$

$$E_F := \Big\{\big\{[u]_{\sim F}, [v]_{\sim F}\big\} \mid \{u,v\} \in F\Big\} \,\dot\cup\, \Big\{\big\{[x]_{\approx F}, [\phi(x)]_{\sim F}\big\} \mid x \in X\Big\} \tag{3.21}$$

and equations (3.15) and (3.16) by

$$V_{(F,Y)} := V/{\sim_F} \,\dot\cup\, Y \tag{3.22}$$

$$E_{(F,Y)} := \Big\{\big\{[u]_{\sim F}, [v]_{\sim F}\big\} \mid \{u,v\} \in F\Big\} \,\dot\cup\, \Big\{\big\{y, [\phi(y)]_{\sim F}\big\} \mid y \in Y\Big\} \tag{3.23}$$

It is easy to check that the resulting tree T_F is in fact a phylogenetic X_F-tree, and that the canonical projection $v \mapsto [v]_{\sim F}$ induces a bijection between F and $\operatorname{Int}(T_F)$. But, the fact that T is a binary X-tree does in general not imply that T_F is also binary; more precisely, this is the case if and only if F is a patch.

As above, $T_{(F,Y)}$ is not a Y-tree in general; but if

$$\#\{e \in F \mid \bar{v} \cap e \neq \emptyset\} + \#\big(\bar{v} \cap \phi(Y)\big) \geq 3 \tag{3.24}$$

holds for all "inner" vertices $\bar{v} \in V/{\sim_F}$ of $T_{(F,Y)}$, then $T_{(F,Y)}$ is in fact a *phylogenetic* Y-tree and the left side of the inequality equals the degree of $\bar{v}$ in $T_{(F,Y)}$. Again, we can easily check that the canonical projection $v \mapsto [v]_{\sim F}$ induces a bijection between F and $\operatorname{Int}(T_{(F,Y)})$, and that $T_{(F,Y)} \leq T|_Y$ holds. Confer Fig. 3.1 on page 26 for an example of the construction of T_F and $T_{(F,Y)}$.

Lemma 3.12. *Given a phylogenetic X-tree $T = (V, E; \phi)$ and subsets $F \subseteq \mathring{E}$, $Y \subseteq X$ such that $Y/{\approx_F} = X/{\approx_F}$ holds, then $T_{(F,Y)}$ is a phylogenetic Y-tree.*

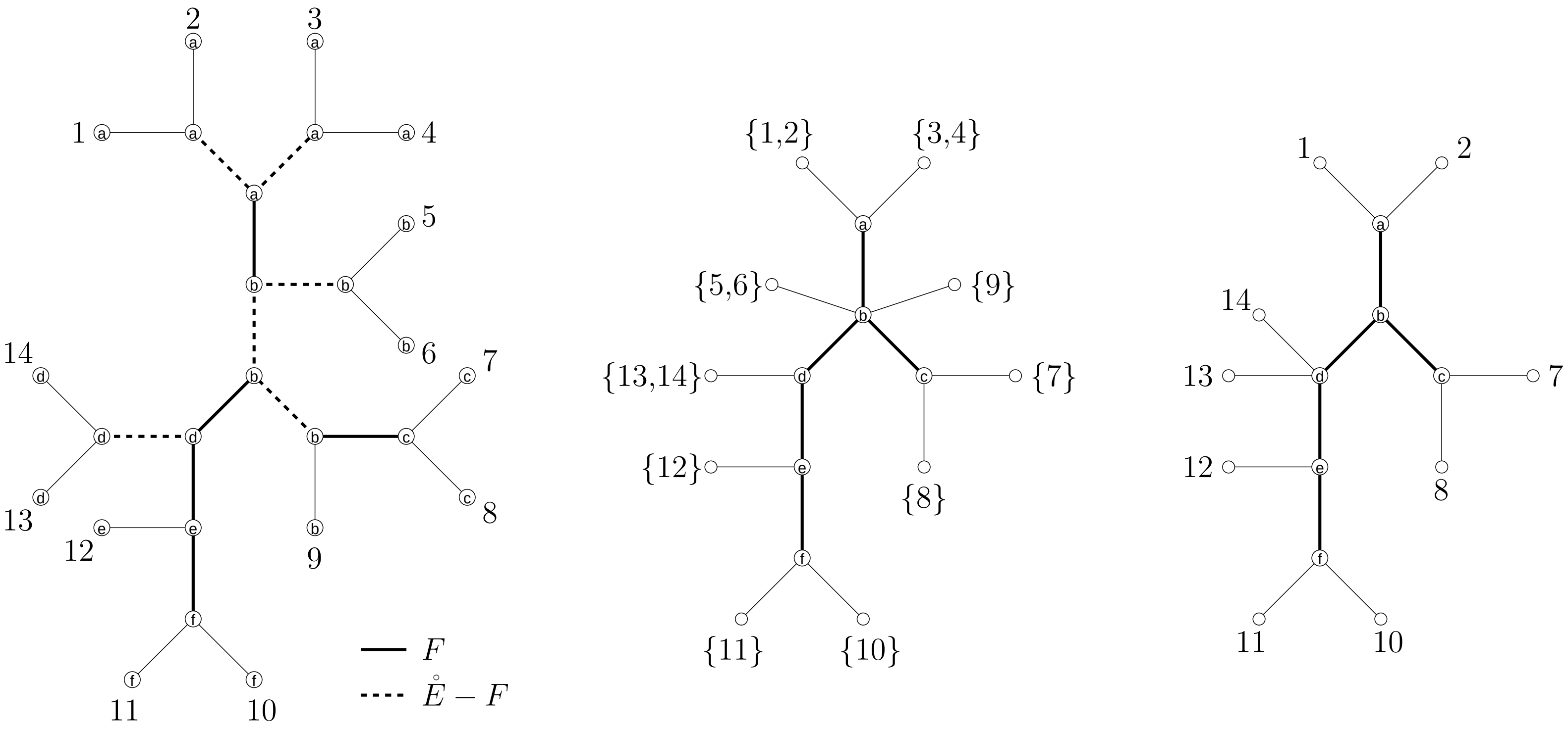
Proof. To show that $T_{(F,Y)}$ is a phylogenetic Y-tree, we show that (3.24) holds for every $\bar{v} \in V/{\sim_F}$. Let $F' := \{e \in F \mid \bar{v} \cap e \neq \emptyset\}$; for $\#F' \geq 3$ we are done.

Second, suppose $F' = \{e_1, e_2\}$ for distinct edges e_1, e_2. Since there are no vertices of degree two in (V, E) we infer that $\bar{v} \subseteq V$ contains at least one leaf; and since T is phylogenetic we infer further $\phi(X) \cap \bar{v} \neq \emptyset$, so we can choose $x \in X$ with $\phi(x) \in \bar{v}$. Now, the condition of the lemma implies the existence of some $y \in Y$ with

$$[\phi(y)]_{\approx F} = [\phi(x)]_{\approx F} \subseteq [\phi(x)]_{\sim F} = \bar{v}$$

and, finally, $\bar{v} \cap \phi(Y) \neq \emptyset$.

[2]Nevertheless, we will use Definitions 3.2 and 3.3 for the sake of brevity whenever we can be sure that $F \subseteq \mathring{E}$ is a patch.



An X-tree T. The induced X_F-tree T_F. The induced Y-tree $T_{(F,Y)}$.

FIGURE 3.1. Inducing subtrees via arbitrary subsets $F \subseteq \mathring{E}$. Left: An X-tree T for $X := \{1, \ldots, 14\}$. The set F is indicated by thick lines, the set $\mathring{E} - F$ by dashed lines. Two vertices u, v of T are labeled with the same character (from a, … , f) if $u \sim_F v$ holds. Middle: the induced X_F-tree T_F. Here, any character a, … , f indicates the equivalence class of all vertices in T labeled analogously. Right: The induced Y-tree $T_{(F,Y)}$ for $Y := \{1, 2, 7, 8, 10, \ldots, 14\}$.

Third, if $F' = \{e\}$ for some edge e then, by an analogous argumentation to above, we show the existence of $x_1, x_2 \in X$ with $x_1 \not\approx_F x_2$ and $\phi(x_1), \phi(x_2) \in \bar{v}$; so, the condition of the lemma implies the existence of $y_1, y_2 \in Y$ with $y_1 \not\approx_F y_2$ and

$$[\phi(y_j)]_{\approx F} = [\phi(x_j)]_{\approx F} \subseteq [\phi(x_j)]_{\sim F} = \bar{v}$$

for $j = 1, 2$ and, finally, $\#(\bar{v} \cap \phi(Y)) \geq 2$. □

Under what conditions do these (modified) constructions lead to the "same" tree? It is easy to see that $T_{(F,Y)} \cong (V_F, E_F; \phi_F \circ \beta_F)$ still holds true if $Y \subseteq X$ is a minimal set with $Y/{\approx_F} = X/{\approx_F}$; but, under this condition, it is necessary for $T|_Y \cong T_{(F,Y)}$ to hold that F is a patch.

Now, we give the promised condition for $T|_Y$ and $T_{(F,Y)}$ to be the "same" tree:

Lemma 3.13. *Given a phylogenetic X-tree $T = (V, E; \phi)$ and subsets $F \subseteq \mathring{E}$, $Y \subseteq X$ then $T|_Y \cong T_{(F,Y)}$ holds if and only if* (i) $F \subseteq \langle \bigcup \mathrm{Int}(T|_Y) \rangle_T$ *and* (ii) *for every inner edge $\{u, v\}$ of $T|_Y$, we have*

$$\#\big(\langle u, v\rangle_T \cap F\big) = 1\,. \tag{3.25}$$

Note that $\bigcup \mathrm{Int}(T|_Y) = V(T|_Y) - \phi(Y) \subseteq V$ are the inner vertices of the Y-tree $T|_Y$. Furthermore, "$T|_Y \cong T_{(F,Y)}$" shall imply that $T_{(F,Y)}$ is a Y-tree.

Proof. Firstly, suppose that $T|_Y \cong T_{(F,Y)}$ holds. For every edge $e \in F$ inducing the (non-trivial) split $S \in \mathcal{S}(Y)$ in $T_{(F,Y)}$, there exist vertices $u, v \in V$ such that $\{u, v\}$ is an inner edge of $T|_Y$ and induces the same split S. Then

$$\big(S_T[f]\big)\big|_Y = S \iff f \in \langle u, v\rangle_T \tag{3.26}$$

holds for all edges $f \in E$ by Remark 3.4 — in particular, we infer $e \in \langle u, v\rangle_T$. This establishes $F \subseteq \langle V(T|_Y) - \phi(Y)\rangle_T$.

Now, let $e = \{u, v\}$ denote an arbitrary inner edge of $T|_Y$ and define

$$S := S_{T|_Y}[e] = \big(S_T[e]\big)\big|_Y\,;$$

then $\langle u, v\rangle_T \cap F = \emptyset$, together with (3.26) for all edges $f \in E$, would imply $S \notin \mathcal{S}\big[T_{(F,Y)}\big]$ in contradiction to the assumptions. Moreover, $\#(\langle u, v\rangle_T \cap F) \geq 2$ implies that $T_{(F,Y)}$ is not a Y-tree by an analogous argumentation, also in contradiction to the assumptions, what establishes the first part of the proof.

Vice versa, suppose $F \subseteq \langle V(T|_Y) - \phi(Y)\rangle_T$, and that (3.25) holds for all inner edges $\{u, v\}$ of $T|_Y$. We define a map

$$\alpha : V(T|_Y) \to V_{(F,Y)} \;:\; v \mapsto \begin{cases} \phi^{-1}(v) & \text{for all } v \in \phi(Y), \\ [v]_{\sim F} & \text{otherwise} \end{cases}$$

that obviously satisfies $\alpha \circ \phi|_Y = \phi^{-1} \circ \phi|_Y = id_Y$. It remains to show that α is a bijection, and that $\{\alpha(u), \alpha(v)\}$ is an edge of $T_{(F,Y)}$ for every edge $\{u, v\}$ of $T|_Y$; then α induces a bijection between the edges of $T|_Y$ and $T_{(F,Y)}$, because $\big(V_{(F,Y)}, E_{(F,Y)}\big)$ is known to be a tree. Let

$$\mathring{V} := V(T|_Y) - \phi(Y) = \{v \in V(T|_Y) \bigm| \deg(v) > 1\}$$

denote the inner vertices of $T|_Y$; we claim that

$$\#\left(\bar{v}\cap\mathring{V}\right)=1 \quad \text{holds for every equivalence class} \quad \bar{v}\in V/{\sim_F} \tag{3.27}$$

what easily implies $\alpha(\mathring{V}) = V/{\sim_F}$ and, together with $\alpha(\phi(Y)) = Y$, that α is surjective.

To establish (3.27), we first infer for $u,\, w \in \bar{v}\cap\mathring{V}$ that $\langle u,w\rangle_T \cap F = \emptyset$ and, in view of (3.25), $u = w$ must hold. Second, suppose $\bar{v}\cap\mathring{V} = \emptyset$: If there exists a unique edge $e \in F$ with $e\cap\bar{v} \neq \emptyset$ then $e \notin \langle\mathring{V}\rangle_T$ must hold in view of $\#(\bar{v}\cap\phi(Y)) \leq 1$ in contradiction to the assumptions: For $x,\, y \in Y$ with $\phi(x),\, \phi(y) \in \bar{v}$ we have $[\phi(x),\phi(y)]_T \subseteq \bar{v}$, so $\bar{v}\cap\mathring{V} = \emptyset$ implies $x = y$ by construction of $T|_Y$. Otherwise, there exist two distinct edges $e_1,\, e_2 \in F$ with $e_j\cap\bar{v} \neq \emptyset$ for $j = 1, 2$, and by assumption (i) there exist vertices $u,\, w \in \mathring{V}$ such that $e_1,\, e_2 \in \langle u,w\rangle_T$ holds. If we choose such u, w so that $p_T(u{\to}w)$ is of *minimal* length then $\{u,w\}$ is necessarily an edge of $T|_Y$ in contradiction to (3.25).

For two distinct inner vertices $u,\, w \in \mathring{V}$ we infer from (3.27) that $u \not\sim_F w$ must hold. As $\alpha|_{\phi(Y)}$ is surely injective we conclude that α is injective and, finally, bijective.

It remains to show that $\{\alpha(u),\alpha(w)\}$ is an edge of $T_{(F,Y)}$ for every edge $\{u,w\}$ of $T|_Y$: If $u = \phi(x)$ holds for some $x \in Y$, then $[u,w]_T \cap F = \emptyset$ follows from (i) and hence, $u \sim_F w$. We infer that $\{\alpha(u),\alpha(w)\} = \{x, [\phi(x)]_{\sim F}\}$ is an edge of $T_{(F,Y)}$. Given an inner edge $\{u,w\}$ of $T|_Y$, there exists a unique edge $e = \{u',w'\}$ in $F\cap\langle u,w\rangle_T$ by (3.25). We may assume without loss of generality that $u \sim_F u'$ and $w \sim_F w'$ holds, so $\{\alpha(u),\alpha(w)\} = \{[u']_{\sim F}, [w']_{\sim F}\}$ is an edge of $T_{(F,Y)}$ and we are done. ☐

3.5. Connecting partial split systems and X-trees

We have seen in Lemma 3.1 that there exists a fundamental correspondence between split systems and X-trees. Unfortunately, things get more complicated when dealing with partial splits instead of splits.

DEFINITION 3.4. Given a subset $\mathcal{S} \subseteq \mathcal{S}_{\text{part}}(X)$ of partial splits of X and an X-tree T, we say that T is *concordant* with $\mathcal{S}$ if, for every $\{A,B\} \in \mathcal{S}$, there exists at least one edge e of T that separates $\phi(A)$ from $\phi(B)$, that is, with $\{A,B\} \leq S[e]$. Let $\mathcal{T}_X(\mathcal{S}) = \mathcal{T}(\mathcal{S})$ denote the set of all (isomorphism classes of) phylogenetic X-trees concordant with $\mathcal{S}$.

Although $\mathcal{T}(\mathcal{S})$ is formally defined to be a set of isomorphism classes of trees, we will still write, for the sake of brevity, $T \in \mathcal{T}(\mathcal{S})$ to indicate that the isomorphism class of T is an element of $\mathcal{T}(\mathcal{S})$.

Note that $T \in \mathcal{T}(\mathcal{S})$ and $T \leq T'$ implies $T' \in \mathcal{T}(\mathcal{S})$, hence any X-tree T with $\{T\} = \mathcal{T}(\mathcal{S})$ for some set $\mathcal{S} \subseteq \mathcal{S}_{\text{part}}(X)$ must be a binary X-tree.

Note also that a collection $\mathcal{S}$ of partial splits with $\mathcal{T}(\mathcal{S}) = \{T\}$ for some binary X-tree $T = (V,E;\phi)$ must contain at least one partial split for every inner edge that specifically "fits" this edge and, hence, $\mathcal{S}$ must contain at least $\#\mathring{E} = \#X - 3$ distinct non-trivial partial splits in view of inequality (3.2). But even a *minimal* collection of partial splits concordant with a unique tree may contain more than this *minimum* number of partial splits:

Example.

4. Define $X := \{1, \ldots, 7\}$ and $\mathcal{Q} := \{12|35, 24|57, 13|47, 34|56, 15|67\} \subseteq \mathcal{Q}(X)$. The unique X-tree concordant with $\mathcal{Q}$ is the caterpillar with seven leaves (as depicted in Fig. 6.3 on page 60), but, for any subset of $\mathcal{Q}$ of size $7-3=4$, there are at least two X-trees concordant with the resulting collection. (from [Ste92])

We will show in the following chapter how to amalgamate *two* X-trees into one. As outlined in the introduction, the general reconstruction problem is as follows: Given as input a collection of trees, we wish to determine whether we can *amalgamate* them into a common *supertree*, that is, whether there exists some X-tree that displays these trees, and if so, whether there exists exactly one such supertree. Formally, let $(Y_1, \ldots, Y_k)$ be a family of non-empty subsets of X, and consider a family $\mathcal{F} := (T_1, \ldots, T_k)$ where T_j is a Y_j-tree for $j = 1, \ldots, k$. We may wish to consider the set $\mathcal{T}(\mathcal{F})$ of all (isomorphism classes of) phylogenetic X-trees T that display every tree in $\mathcal{F}$.

The problem of computing $\mathcal{T}(\mathcal{F})$ can always be reduced to the (seemingly more special) task of computing $\mathcal{T}(\mathcal{F})$ for a family $\mathcal{F}$ of quartet trees: Bandelt and Dress established in [BD86] (Prop. 2) that, for $\mathcal{S} \subseteq \mathcal{S}_{\text{part}}(X)$ and

$$\mathcal{Q}(\mathcal{S}) := \{Q \in \mathcal{Q}(X) \mid Q \leq S \text{ for some } S \in \mathcal{S}\}, \tag{3.28}$$

the relation

$$\mathcal{T}(\mathcal{S}) = \mathcal{T}(\mathcal{Q}(\mathcal{S}))$$

must hold (see also [Ste92]). Thus, there is no loss of generality in restricting one's attention to quartet splits when reconstructing (phylogenetic) trees from partial splits. Similarly, we have

$$\mathcal{T}(\mathcal{F}) = \mathcal{T}\Big(\bigcup\nolimits_{i=1,\ldots,k} \mathcal{S}[T_i]\Big) = \mathcal{T}\Big(\mathcal{Q}\Big(\bigcup\nolimits_{i=1,\ldots,k} \mathcal{S}[T_i]\Big)\Big) \tag{3.29}$$

for any family $\mathcal{F} = (T_1, \ldots, T_k)$ as above. So, the problem of reconstructing trees from subtrees also reduces to the problem of reconstructing them from (quartet) splits. Further results regarding the problem of reconstructing supertrees from subtrees will be described in Chapter 8.

CHAPTER 4

The Tree Joining Theorem

The purpose of this section will be to establish the following theorem, that gives a full answer to the questions raised in the introduction on Fig. 1.1:

Tree Joining Theorem. Let T_j denote an X_j-tree for $j = 1, 2$, and define $Y := X_1 \cap X_2$ and $X := X_1 \cup X_2$. Then, there exists a Y-tree T' with $T_j|_Y \leq T'$ for $j = 1, 2$ if and only if there exists an X-tree T with

$$T_j \leq T|_{X_j} \quad \text{for} \quad j = 1, 2\,. \tag{4.1}$$

The X-tree T is uniquely defined (up to canonical isomorphisms) if and only if the following three conditions hold:

(i) $\#\Big(\mathcal{S}[T_1|_Y] \cup \mathcal{S}[T_2|_Y]\Big) = 2\,\#Y - 3$;

(ii) $\#\mathcal{S}[T_j] - \#\mathcal{S}[T_j|_Y] = 2\,\#X_j - 2\,\#Y$ for $j = 1, 2$; and

(iii) for every $S' \in \mathcal{S}[T'] = \mathcal{S}[T_1|_Y] \cup \mathcal{S}[T_2|_Y]$ we have

$$\#\{S \in \mathcal{S}[T_1] \mid S' \leq S\} \leq 1 \quad \text{or} \quad \#\{S \in \mathcal{S}[T_2] \mid S' \leq S\} \leq 1\,.$$

Condition (i) states that T' is the unique Y-tree with $T_j|_Y \leq T'$ for $j = 1, 2$ or, equivalently, that T' is necessarily a binary Y-tree. Condition (ii) states that T_1, T_2 must be "binary except for the subtrees $T_1|_Y$, $T_2|_Y$." Condition (iii) says that we shall only attach edges of either T_1 or T_2 to every edge of T'.

4.1. Proof of the existence

To establish the equivalence claimed in the first part of the theorem, we first note that we already know from Section 3.3 that $T_j \leq T|_{X_j}$ implies $T_j|_Y \leq (T|_{X_j})\big|_Y = T|_Y =: T'$ for $j = 1, 2$. To show the second part of the equivalence, we note that we may suppose without loss of generality that T' is a binary tree. In view of Corollary 3.9 we may even suppose that T_1 and T_2 are binary trees, too.

We use induction on $k = \#X_1 + \#X_2$; for $X_1 = X_2 = Y$ we simply set $T := T'$. Now, suppose without loss of generality that $Y \subsetneqq X_1$ holds. Choose a fixed $y \in X_1 - Y$, and define $X_0 := X_1 - \{y\}$ and $T_0 := T_1|_{X_0}$. We denote by v the vertex of $T_1 = (V_1, E_1; \phi_1)$ such that $\{v, \phi_1(y)\} \in E_1$ holds, and we denote the two remaining vertices incident with v by w and w' — so, $\{v, w\}, \{v, w'\} \in E_1$ holds. Since $T_0 = (V_0, E_0; \phi_0)$ is a binary tree in view of Corollary 3.6, we infer that

$$\begin{aligned} &V_0 = V_1 - \{v, \phi_1(y)\} \quad \text{and} \\ &E_0 = E_1 - \big\{\{v, \phi_1(y)\}, \{v, w\}, \{v, w'\}\big\} \cup \big\{\{w, w'\}\big\} \end{aligned} \tag{4.2}$$

must necessarily hold, since $V_0 \subseteq V_1$ by definition. Because $T_0|_Y = (T_1|_{X_0})\big|_Y = T_1|_Y \leq T'$ there exists an $(X-\{y\})$-tree $T_* = (V_*, E_*; \phi_*)$ such that

$$T_0 \leq T_*|_{X_0} \quad \text{and} \quad T_2 \leq T_*|_{X_2}$$

by induction hypothesis. As T_0 is binary, we may in fact assume $T_0 = T_*|_{X_0}$.

From Remark 3.4 we infer that

$$S_{T_*}[e]\big|_{X_0} = S_{T_0}[\{w, w'\}]$$

holds for exactly the edges $e \in \langle w, w'\rangle_{T_*}$ of T_*. Choose an arbitrary edge $\{u, u'\} \in \langle w, w'\rangle_{T_*}$ and two vertices $z,\, z' \notin V_*$. We define an X-tree $T = (V, E; \phi)$ by

$$\begin{aligned} &V := V_* \cup \{z, z'\}\,,\\ &E := E_* - \big\{\{u, u'\}\big\} \cup \big\{\{z, z'\}, \{z, u\}, \{z, u'\}\big\}\,,\\ \text{and}\quad &\phi : X \to V \ : x \mapsto \begin{cases} \phi_*(x) & \text{for } x \in X_0 \cup X_2\,,\\ z' & \text{for } x = y\,. \end{cases} \end{aligned} \tag{4.3}$$

Since $T|_{X_0} = T_0 = T_1|_{X_0}$ holds, the placement of the leaf z' secures that we can define an isomorphism α between $T|_{X_1}$ and T_1. Since obviously $T|_{X_2} = T_*|_{X_2} \cong T_2$ we are done.

4.2. Proof of the uniqueness

Throughout the rest of this section, we will use the notations $\mathcal{S} = \mathcal{S}[T]$, $\mathcal{S}_j = \mathcal{S}[T_j]$ for $j = 1, 2$, and $\mathcal{S}' = \mathcal{S}[T']$. Then $T_j \leq T|_{X_j}$ for $j = 1, 2$ holds if and only if

$$\mathcal{S}_j \subseteq \mathcal{S}|_{X_j} \quad \text{holds for} \quad j = 1, 2\,. \tag{4.4}$$

To prove the second part of the theorem is more sophisticated — first, we will show that condition (i) is necessary: To this end, suppose that there exists exactly one split system $\mathcal{S}$ satisfying (4.4). Clearly, $\mathcal{S}$ must be maximal; then $\mathcal{S}|_Y$ is maximal by Corollary 3.6. We may suppose that $\mathcal{S}' = \mathcal{S}_1|_Y \cup \mathcal{S}_2|_Y$ holds. If there exists a split $S \in \mathcal{S}$ with $S|_Y \notin \mathcal{S}' \cup \{\{Y, \emptyset\}\}$ then $S|_{X_j} \notin \mathcal{S}_j \cup \{\{X_j, \emptyset\}\}$ follows; we can remove S from $\mathcal{S}$, and the resulting split system $\mathcal{S} - \{S\}$ still satisfies (4.4), a contradiction.

In view of Lemma 3.8 and equation (3.11), we may suppose for the rest of this proof that $\mathcal{S}_1|_Y = \mathcal{S}' = \mathcal{S}_2|_Y$ is maximal. For the remaining equivalences, we use induction on $k = \#X_1 + \#X_2$, where the equivalence is obvious for $X_1 = X_2 = Y$, because $T' \cong T_1 \cong T_2 \cong T$ holds in this case. Now suppose — without loss of generality — that $Y \subsetneqq X_1$ holds: We choose $y \in X_1 - Y$ and define

$$X_* := X - \{y\}, \quad X_0 := X_1 - \{y\}, \quad T_0 := T_1|_{X_0}, \quad \text{and} \quad \mathcal{S}_0 := \mathcal{S}_1|_{X_0} = \mathcal{S}[T_0]\,.$$

Then $T_0|_Y \leq T'$ must hold and, by induction hypothesis, conditions (i)–(iii) of the theorem (with T_1 replaced by T_0) are satisfied if and only if there exists a *unique* X_*-tree T_* with

$$T_0 \leq T_*|_{X_0} \quad \text{and} \quad T_2 \leq T_*|_{X_2}\,. \tag{4.5}$$

Now, suppose that T is uniquely determined by (4.1); then $T_* = T|_{X_*}$ is also uniquely determined by (4.5) in view of the first part of the theorem and, by induction hypothesis, conditions (i)–(iii) hold when replacing T_1 by T_0.

If *condition (ii)* is violated for T_1, T_2, then

$$\#\mathcal{S}_1 < 2\,\#X_1 - 2\,\#Y + \#\mathcal{S}' = 2\,\#X_1 - 3$$

must necessarily hold, and we calculate

$$\#\mathcal{S}_1 - \#\mathcal{S}_0 < 2\,\#X_1 - 2\,\#Y + \#\mathcal{S}' - 2\,\#X_0 + 2\,\#Y - \#\mathcal{S}' = 2$$

so there exists at most one split $S \in \mathcal{S}_1$ with

$$(\mathcal{S}_1 - \{S\})\big|_{X_0} = \mathcal{S}_0\,.$$

In case $\{X_0, \{y\}\} \notin \mathcal{S}_1$ we can remove the split $\{X_*, \{y\}\}$ from $\mathcal{S}$; while for $\{X_0, \{y\}\} \in \mathcal{S}_1$ there clearly exist two distinct splits S_1, $S_2 \in \mathcal{S}$ such that $S_1|_{X_*} = S_2|_{X_*}$ but — without loss of generality — $S_2|_{X_1} \notin \mathcal{S}_1$ (see Lemma 3.7), so we can remove S_2 from $\mathcal{S}$; in both cases, we obtain a second split system that still satisfies (4.4) in contradiction to the assumptions.

If *condition (iii)* is violated for T_1, T_2, then there exists a split $S' \in \mathcal{S}'$ with

$$\#\{S \in \mathcal{S}_1 \mid S' \leq S\} \geq 2 \tag{4.6}$$

$$\text{and} \quad \#\{S \in \mathcal{S}_2 \mid S' \leq S\} \geq 2, \tag{4.7}$$

but condition (iii) holds when replacing T_1 by T_0, so

$$\#\{S \in \mathcal{S}_0 \mid S' \leq S\} \leq 1 \tag{4.8}$$

and it follows easily that inequalities (4.6) and (4.8) are in fact equalities. So, there exists exactly one edge e of T_0 with $S' \leq S_{T_0}[e]$, while there exist two edges of T_1 with this property — informally speaking, T_1 can be obtained from T_0 by "attaching a new leaf labeled y to the edge e." For T_*, inequality (4.7) tells us that there exist (at least) two distinct edges e_1, e_2 of $T_* = (V_*, E_*; \phi_*)$ such that $S' \leq S_{T_*}[e_j]$ for $j = 1, 2$ while (4.8) implies

$$S_{T_*}[e_1]\big|_{X_0} = S_{T_*}[e_2]\big|_{X_0} = S_{T_0}[e]\,.$$

Attaching a leaf labeled y to both edges e_1, e_2 via

$$\begin{aligned} V_{(j)} &:= V_* \cup \{z, z'\}\,, \\ E_{(j)} &:= E_* - \{e_j\} \cup \big\{\{z, z'\}, \{z, u_j\}, \{z, v_j\}\big\}\,, \\ \text{and} \quad \phi : X \to V\ &: x \mapsto \begin{cases} \phi_*(x) & \text{for } x \in X_*\,, \\ z' & \text{for } x = y\,. \end{cases} \end{aligned} \tag{4.9}$$

for $e_j = \{u_j, v_j\}$ and for $j = 1, 2$ leads to two (obviously non-isomorphic) X-trees $T_{(j)} = (V_{(j)}, E_{(j)}; \phi)$ for $j = 1, 2$, both meeting condition (4.1): Clearly $T_2 \leq T_{(j)}|_{X_2}$ must hold, but $T_0 \leq T_{(j)}|_{X_0}$ and the placement of the leaf labeled y also secures $T_1 \leq T_{(j)}|_{X_1}$ for $j = 1, 2$ — in contradiction to the assumptions.

Vice versa, suppose that conditions (i)–(iii) hold for T_1, T_2. In view of $\mathcal{S}_j|_Y = \mathcal{S}'$ and

$$\#\mathcal{S}_j = 2\,\#X_j - 2\,\#Y + \#\mathcal{S}' = 2\,\#X_j - 3$$

we infer that T_j is a binary X_j-tree for $j = 1, 2$. Clearly, conditions (i)–(iii) still hold when replacing T_1 by T_0 and, by induction hypothesis, there exists a unique X_*-tree

T_* meeting condition (4.5). Choose two arbitrary non-isomorphic X-trees $T_{(j)}$ for $j = 1, 2$, both satisfying (4.1). Then $T_{(j)}|_{X_*}$ also satisfies (4.5) and, hence,

$$T_{(1)}|_{X_*} \cong T_* \cong T_{(2)}|_{X_*}$$

must hold. For $j = 1, 2$ let e_j denote the edge of T_* "the leaf labeled y is attached to in T_j," that is, the unique edge satisfying

$$\# \left\{ S \in \mathcal{S}\left[T_{(j)}\right] \;\middle|\; S_{T_*}[e_j] \leq S \right\} \geq 2 \tag{4.10}$$

where clearly equality must hold. Then $T_{(1)}|_{X_0} \cong T_0 \cong T_{(2)}|_{X_0}$ implies that e_1, e_2 must induce splits with

$$S_{T_*}[e_1]\big|_{X_0} = S_{T_*}[e_2]\big|_{X_0} \tag{4.11}$$

and we conclude

$$S_{T_*}[e_1]\big|_{X_2} \neq S_{T_*}[e_2]\big|_{X_2} \tag{4.12}$$

because e_1, e_2 are *distinct* edges of the X_*-tree T_*. Let $S' := S_{T_*}[e_1]\big|_Y$ then $S' = S_{T_*}[e_2]\big|_Y$ follows from (4.11), and (4.10) for $j \in \{1, 2\}$ implies

$$\#\{S \in \mathcal{S}_1 \mid S' \leq S\} \geq 2$$

while (4.12) implies

$$\#\{S \in \mathcal{S}_2 \mid S' \leq S\} \geq 2$$

so condition (iii) of the theorem is violated. $\square$

CHAPTER 5

Cluster Systems and Patchworks

In this chapter, we want to introduce the concept of "patchworks" that exhibits an interesting relationship with hierarchies. The results of this chapter (see Section 5.2) will be useful in Chapter 7 for proving the main theorem, because — as we will see in Chapter 6 — the objects of interest in those chapters naturally form such patchwork structures. In Chapter 8, this will not only enable us to easily restate the main theorem in distinct ways, but also lead to fast algorithms to check whether its conditions are fulfilled.

In the following, we will deal with collections $\mathcal{C}$ of subsets of a ground set X; but in all applications that we have in mind in this thesis, the ground set X is in fact a *finite* set, cf. the footnote on page 8. To this end, we will limit our attention to *finite* collections $\mathcal{C} \subseteq \mathcal{P}(X)$ in Sections 5.2 to 5.6, where a finite ground set X surely implies that $\mathcal{C}$ is finite, too. In Section 5.7, we will show how to generalize our results for arbitrary collections $\mathcal{C}$.

Apart from the fact that we will apply the results of this chapter only in case of a finite set $\mathcal{C}$, there are two more good reasons for establishing our results in this case first:

- we do not need to invoke Zorn's Lemma (or, equivalently, the Axiom of Choice) to establish our results;
- while it is possible to organize the proof in the seemingly much simpler case $\#\mathcal{C} < \infty$ in such a way that — apart from invoking Zorn's Lemma here and there — only minor modifications are required to derive the above results without that restriction, presenting the proof in the general case only could create the perfectly wrong impression that only the infinite case needed some sophistication while restriction to the finite case would have made things utterly trivial.

The structure of this chapter is as follows: In Section 5.1, we will give definitions and notations, and establish some useful basic observations. In Section 5.2, we will present our main results on patchworks in case of a finite collection $\mathcal{C}$. Theorem 4 will be established in Section 5.4, and Theorems 5 and 2 in Section 5.5, the latter in case of a finite collection $\mathcal{C}$ only. In Section 5.3, some examples will be presented, and Section 5.6 discusses algorithmic aspects. Finally, in the last section, we will deal with arbitrary, that is, with possibly infinite patchworks: We will prove Theorems 2 and 3 in this more general setting; and we will prove Theorems 6 and 7 that replace Theorems 4 and 5, respectively (the connection between these theorems should be more than obvious).

5.1. Definitions, notations, and basic observations

In this chapter, we consider collections $\mathcal{C} \subseteq \mathcal{P}(X)$ of subsets of X; such collections will also be called *cluster systems*, and the subsets in $\mathcal{C}$ will also be called ($\mathcal{C}$-)*clusters* as mentioned in the introduction. To emphasize the ground set X, we will sometimes also refer to the pair $(X, \mathcal{C})$ rather than to just $\mathcal{C}$ when discussing a cluster system $\mathcal{C}$.

Recall the definitions of $\bigcup \mathcal{C}$, $\bigcap \mathcal{C}$, and $\mathcal{C}^X$ from Section 1.6. We will make use of the following notations, definitions, assertions, and basic observations:

$$\begin{aligned} \max(\mathcal{C}) &:= \{C \in \mathcal{C} \mid C \subseteq C' \in \mathcal{C} \Rightarrow C = C'\} \\ \min(\mathcal{C}) &:= \{C \in \mathcal{C} \mid C \supseteq C' \in \mathcal{C} \Rightarrow C' = C\} \\ \min{}^*(\mathcal{C}) &:= \{C \in \mathcal{C} \setminus \{\emptyset\} \mid C \supseteq C' \in \mathcal{C} \Rightarrow C' = \emptyset \text{ or } C' = C\} \end{aligned}$$

(Hierarchy) $\quad C_1, C_2 \in \mathcal{C} \;\Rightarrow\; C_1 \cap C_2 \in \{\emptyset, C_1, C_2\}$

(Partition) $\quad C_1, C_2 \in \mathcal{C} \;\Rightarrow\; \text{“}C_1 \cap C_2 \neq \emptyset \iff C_1 = C_2\text{”}$

(Chain) $\quad C_1, C_2 \in \mathcal{C} \;\Rightarrow\; C_1 \cap C_2 \in \{C_1, C_2\}$

(Antichain) $\quad C_1, C_2 \in \mathcal{C} \text{ and } C_1 \cap C_2 \in \{C_1, C_2\} \;\Rightarrow\; C_1 = C_2$

(Ample) $\quad C_1, C_2 \in \mathcal{C} \text{ and } \#\{C \in \mathcal{C} \mid C_1 \subseteq C \subseteq C_2\} = 2 \;\Rightarrow\; C_2 - C_1 \in \mathcal{C}$

For $Y \subseteq X$ and $\mathcal{C} \subseteq \mathcal{P}(X)$, we also introduce the notations

$$\begin{aligned} \mathcal{C}_{\subseteq Y} = \mathcal{C}(\subseteq Y) &:= \{C \in \mathcal{C} \mid C \subseteq Y\}, \\ \mathcal{C}_{\supseteq Y} = \mathcal{C}(\supseteq Y) &:= \{C \in \mathcal{C} \mid Y \subseteq C\}, \\ \mathcal{C}_{\subset Y} = \mathcal{C}(\subset Y) &:= \{C \in \mathcal{C}_{\subseteq Y} \mid C \neq Y\}, \\ \text{and} \quad \mathcal{C}_{\supset Y} = \mathcal{C}(\supset Y) &:= \{C \in \mathcal{C}_{\supseteq Y} \mid C \neq Y\}. \end{aligned}$$

Clearly, we have

$$\min{}^*(\mathcal{C}) = \min(\mathcal{C} \setminus \{\emptyset\}); \tag{5.1}$$

furthermore, $\mathcal{C}$ coincides with $\mathcal{C}^X$ if and only if $\max(\mathcal{C}) = \max(\mathcal{P}(X)) = \{X\}$, $\min(\mathcal{C}) = \min(\mathcal{P}(X)) = \{\emptyset\}$, and $\min^*(\mathcal{C}) = \min^*(\mathcal{P}(X)) = \binom{X}{1}$.

We define a cluster system $\mathcal{C} \subseteq \mathcal{P}(X)$ to be an (X-)*hierarchy* if it satisfies (Hierarchy), it is called a *partition* if it satisfies (Partition) in which case it is also called a partition of the subset $Y := \bigcup \mathcal{C}$ of X, and we define it to be a *chain* or an *antichain* if it satisfies (Chain) or (Antichain), respectively. As in [BD00a], we define $\mathcal{C}$ to be *ample* if it satisfies (Ample).

It is easily established that, according to these definitions, a partition never contains the empty set among its members; that any chain is a hierarchy; that every partition $\mathcal{C}$ is an antichain; that $\max(\mathcal{C})$, $\min(\mathcal{C})$, and $\min^*(\mathcal{C})$ are antichains for all $\mathcal{C} \subseteq \mathcal{P}(X)$; that any hierarchy that is simultaneously an antichain is either a partition or coincides with $\{\emptyset\}$; and that any subsystem of a partition, a chain, an antichain, or a hierarchy is itself a partition, a chain, an antichain, or a hierarchy, respectively. In particular, $\max(\mathcal{C})$ and $\min^*(\mathcal{C})$ are partitions for every hierarchy $\mathcal{C} \neq \{\emptyset\}$ — and, hence, for every subset $\mathcal{C} \neq \{\emptyset\}$ of a hierarchy.

Any cluster system $\mathcal{C} \subseteq \mathcal{P}(X)$ with $\#\mathcal{C} = 2$ is always (strictly) either a chain or an antichain — in particular, a cluster system $\mathcal{C}$ that is simultaneously a chain and an antichain, cannot have more than one member — while a cluster system $\mathcal{C} \subseteq \mathcal{P}(X)$ is a hierarchy if and only if any subsystem $\mathcal{C}_0 \subseteq \mathcal{C}$ with $\#\mathcal{C}_0 = 2$ is (strictly) either a chain or a partition.

And note that a *finite* cluster system $\mathcal{C}$ is ample if and only if $\mathcal{C}_{\subseteq C}$ is ample for all $C \in \max(\mathcal{C})$: Firstly, given an ample cluster system $(X, \mathcal{C})$ and an arbitrary subset $Y \subseteq X$ then $\mathcal{C}_{\subseteq Y}$ must be ample, too, by definition of ampleness. The converse follows easily from the fact that, given a non-empty cluster C of a finite cluster system $\mathcal{C}$, there exists some $C_{\max} \in \max(\mathcal{C})$ and some $C_{\min} \in \min^*(\mathcal{C})$ with $C_{\min} \subseteq C \subseteq C_{\max}$.

Given a *finite* ample cluster system $(X, \mathcal{C})$ and a non-empty cluster C in $\mathcal{C} - \min^*(\mathcal{C})$, the following holds:

(Ample 1) there exist disjoint clusters C_1, $C_2 \in \min^*(\mathcal{C}_{\subseteq C})$ with $C_1 \cup C_2 \in \mathcal{C}$;
(Ample 2) C is the union of two proper, disjoint subclusters;
(Ample 3) $C = \bigcup \min^*(\mathcal{C}_{\subseteq C})$.

In particular, all this holds for $C := X$ in case $\emptyset \neq X \in \mathcal{C} - \min^*(\mathcal{C})$.

To prove (Ample 2), choose a maximal subcluster $C_1 \in \mathcal{C}_{\subset C}$, then (Ample) implies that $C_2 := C - C_1$ is an element of $\mathcal{C}$.

To prove (Ample 1), consider $\mathcal{C}^* := \mathcal{C}_{\subseteq C} \setminus (\min^*(\mathcal{C}) \cup \{\emptyset\})$ and choose a minimal cluster $C^* \in \mathcal{C}^*$. In view of (Ample 2) there exist proper, disjoint subclusters C_1, $C_2 \in \mathcal{C}$ with $C^* = C_1 \,\dot\cup\, C_2$, so C_1, $C_2 \in (\mathcal{C}_{\subseteq C} - \mathcal{C}^*) \setminus \{\emptyset\} = \min^*(\mathcal{C}_{\subseteq C})$ as well as $C_1 \cup C_2 \in \mathcal{C}$ must hold.

To prove (Ample 3), note that the inclusion "$\supseteq$" is obvious. To show the converse, assume $x \in C$ and choose $C_1 \in \min\{C' \in \mathcal{C} \mid x \in C' \subseteq C\}$. We claim $C_1 \in \min^*(\mathcal{C}_{\subseteq Y})$: Indeed, if there were any proper subset $C' \subsetneqq C_1$ with $\emptyset \neq C' \in \mathcal{C}$, it would have to be contained in $C_2 := C_1 - \{x\}$. Hence, $C_1 \notin \min^*(\mathcal{C}_{\subseteq Y})$ would imply the existence of a non-empty set $C_3 \in \max(\mathcal{C}_{\subseteq C_2})$, and for this set C_3, we would have $\#\{C' \in \mathcal{C} \mid C_3 \subseteq C' \subseteq C_1\} = 2$ and hence, $C_1 - C_3 \in \mathcal{C}$ in contradiction to $x \in C_1 - C_3 \subsetneqq C_1$.

The following lemma gives a characterization of maximal hierarchies on a *finite* ground set X:

LEMMA 5.1. *Given a finite set X and a hierarchy $\mathcal{C} \subseteq \mathcal{P}(X)$. Then $\#\mathcal{C} \leq 2\,\#X$, and the following four conditions are equivalent:*

(i) $\mathcal{C}$ *is a maximal X-hierarchy;*
(ii) $\mathcal{C}$ *is ample, and* $\mathcal{C} = \mathcal{C}^X$*;*
(iii) $\#\mathcal{C} = 2\,\#X$*;*
(iv) $\emptyset$, $X \in \mathcal{C}$*, and every cluster C with $\#C \geq 2$ is the (necessarily disjoint) union of two proper subclusters.*

We will omit the proof of the well-known equivalences "(i) $\Leftrightarrow$ (iii) $\Leftrightarrow$ (iv)". The implication "(ii) $\Rightarrow$ (iv)" follows directly from (Ample 2), and "(i) $\Rightarrow$ (ii)" is easily established: Given a hierarchy $\mathcal{C} \subseteq \mathcal{P}(X)$ and clusters C_1, $C_2 \in \mathcal{C}$ such that $\#\{C \in \mathcal{C} \mid C_1 \subseteq C \subseteq C_2\} = 2$ then

$$\mathcal{C} = \mathcal{C}(\subseteq C_1) \cup \mathcal{C}(\subseteq (C_2 - C_1)) \,\dot\cup\, \mathcal{C}(\supseteq C_2)$$

must hold because $\mathcal{C}$ is a hierarchy; this implies easily that $\mathcal{C}\cup\{C_2-C_1\}$ is a hierarchy, too. So, if $\mathcal{C}$ is maximal we must have $C_2 - C_1 \in \mathcal{C}$ as required.

In the following, we will be particularly interested in cluster systems $\mathcal{C} \subseteq \mathcal{P}(X)$ that satisfy the assertion

(Patchwork) $\quad \emptyset \neq \mathcal{C}' \subseteq \mathcal{C}$ and $\bigcap \mathcal{C}' \neq \emptyset \Rightarrow \bigcap \mathcal{C}' \in \mathcal{C}$ and $\bigcup \mathcal{C}' \in \mathcal{C}$.

Every such cluster system will henceforth be called a *patchwork* or, more specifically, an X*-patchwork*.

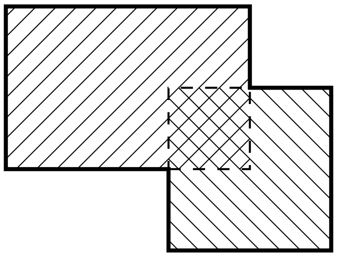


FIGURE 5.1. On the definition of patchworks.

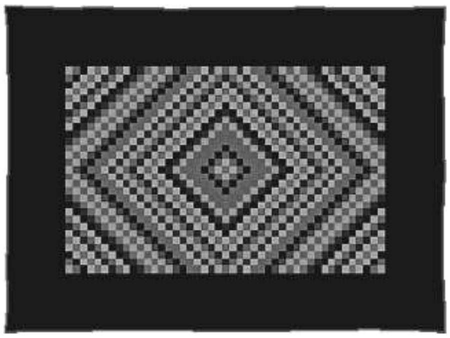


FIGURE 5.2. A patchwork quilt.

Confer Fig. 5.1 on the definition of patchworks. Clearly, a *finite* cluster system $\mathcal{C}$ is a patchwork if and only if it satisfies condition (Patchwork) for all subsets $\mathcal{C}' \subseteq \mathcal{C}$ with $\#\mathcal{C}' = 2$; or equivalently, if

$$A,\, B \in \mathcal{C} \quad \text{and} \quad A \cap B \neq \emptyset \;\Rightarrow\; A \cap B,\, A \cup B \in \mathcal{C} \tag{5.2}$$

holds. In particular, every *finite* hierarchy is also a patchwork. Furthermore, if a cluster system $\mathcal{C} \subseteq \mathcal{P}(X)$ is an X-patchwork or an X-hierarchy, then $\mathcal{C}^X$ is an X-patchwork or an X-hierarchy, too.

REMARK 5.2. Clearly, every patchwork $(X, \mathcal{C})$ with $\emptyset,\, X \in \mathcal{C}$ is a lattice with meet $A \wedge B := A \cap B$ and join $A \vee B := \bigcap\{C \in \mathcal{C} \mid A, B \subseteq C\}$. However, it is — in general — not a sub-lattice of $(X, \mathcal{P}(X))$ because $A \vee B \neq A \cup B$ can hold whenever $A \cap B = \emptyset$.

A (finite or infinite) X-patchwork $\mathcal{C}$ with $X \in \mathcal{C}$ will be called a *connected* patchwork, and it will be called *relatively connected* if its *support* $\bigcup \mathcal{C}$ is a $\mathcal{C}$-cluster.

Clearly, any two distinct maximal clusters in a not relatively connected patchwork $\mathcal{C}$ must be disjoint; so the set of non-empty maximal clusters in a patchwork $\mathcal{C}$ forms a partition of $\bigcup \mathcal{C}$, and $\mathcal{C}$ decomposes into the "essentially disjoint" union of the connected C-patchworks $\mathcal{C}_{\subseteq C}$ where C runs over all non-empty maximal $\mathcal{C}$-clusters.

Similarly, any two distinct clusters from $\min^*(\mathcal{C})$ must have an empty intersection and hence, the clusters in $\min^*(\mathcal{C})$ provide a partition of $\bigcup \min^*(\mathcal{C})$.

For further reference, we collect some of the above observations in

LEMMA 5.3. *Every patchwork* $\mathcal{C}$ *contains two canonical partitions, one is the partition* $\max(\mathcal{C}) \setminus \{\emptyset\}$ *of* $\bigcup \mathcal{C}$ *consisting of all non-empty maximal* $\mathcal{C}$*-clusters, and the* *other one is the partition of* $\bigcup \min^*(\mathcal{C})$ *consisting of all clusters from* $\min^*(\mathcal{C})$*; more precisely, we have* $C \cap C' \in \{\emptyset, C\}$ *and* $C'' \cap C \in \{\emptyset, C''\}$ *for every* $C \in \mathcal{C}$, $C' \in \max(\mathcal{C})$, *and* $C'' \in \min^*(\mathcal{C})$ *— so, in particular, every maximal sub-hierarchy* $\mathcal{C}_0$ *of* $\mathcal{C}$ *must contain* $\max(\mathcal{C}) \cup \min^*(\mathcal{C})$. *Moreover, we have* $\bigcup \min^*(\mathcal{C}) = \bigcup \mathcal{C}$ *if and only if* $\bigcup \min^*(\mathcal{C}_{\subseteq C}) = C$ *holds for every* $C \in \mathcal{C}$*; so, both of these statements must hold in particular if* $\mathcal{C}$ *is a finite and ample patchwork.*

Finally, we introduce the following two concepts: We define a cluster system $(I, \mathcal{J})$ to be a *trace* of a cluster system $(X, \mathcal{C})$ if there exists an I-indexed family

$$\mathcal{F} := (C_i)_{i \in I}$$

of non-empty disjoint subsets of X so that a subset $J \subseteq I$ is an element of $\mathcal{J}$ if and only if

$$C_J = C_{\mathcal{F},J} := \bigcup \{C_i \mid i \in J\}$$

is an element of $\mathcal{C}$, in which case I and $\mathcal{J}$ will also be denoted by $I_{\mathcal{F}}$ and $\mathcal{J}_{\mathcal{F}}$, respectively. And we define $(I, \mathcal{J})$ to be *proper* if

$$\text{(Proper)} \qquad \#I \geq 2, \quad I \in \mathcal{J}, \quad \text{and} \quad \binom{I}{1} \subseteq \mathcal{J}$$

holds.

For later reference, we define a $\mathcal{C}$-*family* to be a family $\mathcal{F} = (C_i)_{i \in I}$ of non-empty, disjoint clusters from $\mathcal{C}$ indexed by some index set I, and we define any such $\mathcal{C}$-family $\mathcal{F}$ to be a *proper* $\mathcal{C}$-*family* if its trace $(I_{\mathcal{F}}, \mathcal{J}_{\mathcal{F}})$ is a proper trace of $(X, \mathcal{C})$, i.e. if $\#I \geq 2$ and $C_I = \bigcup_{i \in I} C_i \in \mathcal{C}$ holds. It is also obvious that in case $\#X \geq 2$, the cluster system $\mathcal{C}$ is proper if and only if $\mathcal{C} \cup \{\emptyset\} = \mathcal{C}^X$ holds if and only if the family $(\{x\})_{x \in X}$ is a proper $\mathcal{C}$-family.

More generally, the following two lemmata hold:

LEMMA 5.4. *Let* $\mathcal{C} \subseteq \mathcal{P}(X)$ *denote a patchwork with* $\bigcup \mathcal{C} = \bigcup \min^*(\mathcal{C})$*, and consider, for any subset* $Y \subseteq X$*, the* $\mathcal{C}$*-family*

$$\mathcal{F} = \mathcal{F}_Y := (C)_{C \in \min^*(\mathcal{C}_{\subseteq Y})}$$

indexed by the set $I_{\mathcal{F}} := \min^*(\mathcal{C}_{\subseteq Y})$ *that is defined by* $C_i := C$ *for every* $i = C \in I_{\mathcal{F}} \subseteq \mathcal{C}$*. Then, the map*

$$\mathcal{J}_{\mathcal{F}} \to \mathcal{C}_{\subseteq Y} \ : J \mapsto C_J$$

defines a canonical 1-1 correspondence between $\mathcal{J}_{\mathcal{F}}$ *and* $\mathcal{C}_{\subseteq Y}$ *and maps* $I_{\mathcal{F}} = \min^*(\mathcal{C}_{\subseteq Y})$ *onto* $\bigcup(\mathcal{C}_{\subseteq Y})$ *in case* $I_{\mathcal{F}}$ *is contained in* $\mathcal{J}_{\mathcal{F}}$*. Moreover,* $\mathcal{F}_Y$ *is a proper* $\mathcal{C}$*-family if and only if we have* $\emptyset \neq \bigcup(\mathcal{C}_{\subseteq Y}) \in \mathcal{C} - \min^*(\mathcal{C})$ *which holds in particular in case* $\emptyset \neq Y \in \mathcal{C} - \min^*(\mathcal{C})$*.*

LEMMA 5.5. *If* $\mathcal{C}$ *is a patchwork, and if* $\mathcal{F} = (C_i)_{i \in I}$ *is any* $\mathcal{C}$*-family, then* $C \in \mathcal{C}(\subseteq C_I)$ *implies*

$$I(C) = I_{\mathcal{F}}(C) := \{i \in I \mid C \cap C_i \neq \emptyset\} \in \mathcal{J}_{\mathcal{F}} \tag{5.3}$$

as well as $C \subseteq C_{I(C)}$*.*

Proof of Lemma 5.5. Clearly, $\{C_i \mid i \in I\}$ is a partition of C_I and we infer

$$C = C \cap C_I = C \cap \bigcup\{C_i \mid i \in I\} = \bigcup\{C \cap C_i \mid i \in I\} = \\ = \bigcup\{C \cap C_i \mid i \in I,\, C \cap C_i \neq \emptyset\} \subseteq \bigcup\{C_i \mid i \in I,\, C \cap C_i \neq \emptyset\} = C_{I(C)}$$

establishing the second part of our claim. Now, given any $i \in I(C)$ we know $C \cap C_i \neq \emptyset$ and hence, $C \cup C_i \in \mathcal{C}$. Consequently,

$$\emptyset \neq C \subseteq \bigcap\{C \cup C_i \mid i \in I(C)\}$$

implies

$$C_{I(C)} = \bigcup\{C \cup C_i \mid i \in I(C)\} \in \mathcal{C}$$

and therefore $I(C) \in \mathcal{I}_{\mathcal{F}}$, establishing the first part of our claim. □

Proof of Lemma 5.4. Since $\min^*(\mathcal{C})$ is a partition of $\bigcup \mathcal{C}$, we infer

$$C = \bigcup\{C \cap C' \in \min^*(\mathcal{C}) \mid C \cap C' \neq \emptyset\} = \bigcup\{C' \in \min^*(\mathcal{C}) \mid C' \subseteq C\}$$

for every cluster $C \in \mathcal{C}_{\subseteq Y}$. So, the map

$$\mathcal{C}_{\subseteq Y} \to \mathcal{I}_{\mathcal{F}} \;: C \mapsto I(C)$$

(where $I(C)$ is defined in (5.3)) is the inverse of the map defined in Lemma 5.4. The remaining claims follow easily. □

Clearly, every trace of a patchwork is a patchwork — just note that

$$C_{\bigcap \mathcal{J}'} = \bigcap\{C_J \mid J \in \mathcal{J}'\} \quad \text{and} \quad C_{\bigcup \mathcal{J}'} = \bigcup\{C_J \mid J \in \mathcal{J}'\}$$

holds for every $\mathcal{C}$-family $\mathcal{F} = (C_i)_{i \in I}$ and every cluster system $\mathcal{J}' \subseteq \mathcal{I}_{\mathcal{F}} \subseteq \mathcal{P}(I)$ — while even a proper trace of an ample cluster system need not be ample. However, we will see in Theorem 3 below that the proper trace of an ample patchwork is always itself an ample patchwork.

5.2. Main results

The main purpose of this chapter is to establish

Theorem 2. *A patchwork $(X, \mathcal{C})$ contains a maximal hierarchy if and only if $\mathcal{C}$ is ample and coincides with $\mathcal{C}^X$.*

As we have announced above, the following holds for proper traces of ample patchworks:

Theorem 3. *Every cluster system $(I, \mathcal{I})$ that is the proper trace of an ample patchwork $(X, \mathcal{C})$ is itself an ample patchwork.*

The following result obviously implies Theorem 3 in case of a *finite* ground set X:

Theorem 4. *For every finite patchwork $(X, \mathcal{C})$, the following assertions are equivalent:*

(i) *$\mathcal{C}$ is ample;*
(ii) *for every non-empty cluster $C \in \mathcal{C} - \min^*(\mathcal{C})$, there exist two disjoint clusters $A, B \in \mathcal{C}_{\subset C}$ with $A \,\dot\cup\, B = C$;*

(iii) *for every non-empty cluster* $C \in \mathcal{C} - \min^*(\mathcal{C})$*, there exist two clusters* $A, B \in \mathcal{C}_{\subset C}$ *with* $A \cup B = C$*;*
(iv) $\bigcup \mathcal{C} = \bigcup \min^*(\mathcal{C})$ *holds and, for every proper trace* $(I, \mathcal{I})$ *of* $(X, \mathcal{C})$ *with* $\mathcal{I} \subseteq \emptyset^I = \{\emptyset\} \cup \binom{I}{1} \cup \{I\}$*, we have* $\#I = 2$*;*
(v) $\bigcup \mathcal{C} = \bigcup \min^*(\mathcal{C})$ *holds, and every proper trace* $(I, \mathcal{I})$ *of* $(X, \mathcal{C})$ *is ample;*
(vi) $\bigcup \mathcal{C} = \bigcup \min^*(\mathcal{C})$ *holds, and for every partition* $\mathcal{C}_0 \subseteq \mathcal{C}$ *with* $\bigcup \mathcal{C}_0 \in \mathcal{C}$ *and* $\#\mathcal{C}_0 \geq 2$*, there exist distinct* $A, B \in \mathcal{C}_0$ *with* $A \cup B \in \mathcal{C}$*.*

Often, the patchwork $\mathcal{C}$ will satisfy the additional condition $\mathcal{C} = \mathcal{C}^X$; then Theorem 4 reduces to:

Corollary 5.6. *For every finite patchwork* $(X, \mathcal{C})$ *satisfying* $\mathcal{C} = \mathcal{C}^X$*, the following assertions are equivalent:*

(i) $\mathcal{C}$ *is ample;*
(ii) *for every cluster* $C \in \mathcal{C}$ *of cardinality* $\#C \geq 2$*, there exist two disjoint clusters* $A, B \in \mathcal{C}_{\subset C}$ *with* $A \cupdot B = C$*;*
(iii) *for every cluster* $C \in \mathcal{C}$ *of cardinality* $\#C \geq 2$*, there exist two clusters* $A, B \in \mathcal{C}_{\subset C}$ *with* $A \cup B = C$*;*
(iv) *for every proper trace* $(I, \mathcal{I})$ *of* $(X, \mathcal{C})$ *with* $\mathcal{I} \subseteq \emptyset^I = \{\emptyset\} \cup \binom{I}{1} \cup \{I\}$*, we have* $\#I = 2$*;*
(v) *every proper trace* $(I, \mathcal{I})$ *of* $(X, \mathcal{C})$ *is ample;*
(vi) *for every partition* $\mathcal{C}_0 \subseteq \mathcal{C}$ *with* $\bigcup \mathcal{C}_0 \in \mathcal{C}$ *and* $\#\mathcal{C}_0 \geq 2$*, there exist distinct* $A, B \in \mathcal{C}_0$ *with* $A \cup B \in \mathcal{C}$*.*

Theorem 4 in turn will allow us to derive the following generalization of Theorem 2:

Theorem 5. *Let* $\mathcal{C}$ *denote a finite patchwork. Then, the following assertions are equivalent:*

(i) $\mathcal{C}$ *is ample;*
(ii) $\mathcal{C}$ *contains an ample hierarchy* $\mathcal{C}_0$ *with* $\max(\mathcal{C}_0) = \max(\mathcal{C})$ *and* $\min^*(\mathcal{C}_0) = \min^*(\mathcal{C})$*;*
(iii) *every maximal sub-hierarchy* $\mathcal{C}_0$ *of* $\mathcal{C}$ *is ample.*

The following corollary is a simple consequence of Theorem 5 (for a finite patchwork $\mathcal{C}$) or Theorem 7 (for an arbitrary patchwork $\mathcal{C}$):

Corollary 5.7. *A patchwork* $\mathcal{C}$ *is ample if and only if it contains an ample patchwork* $\mathcal{C}_0$ *with* $\max(\mathcal{C}_0) = \max(\mathcal{C})$ *and* $\min^*(\mathcal{C}_0) = \min^*(\mathcal{C})$*.*

5.3. Examples

In the context of reconstructing phylogenetic trees from quartet splits, the following examples will be of particular interest to us:

Examples.

5. Let $T = (V, E)$ denote a tree. We define

$$\mathcal{C}_T := \{F \subseteq E \mid F \text{ is a } T\text{-patch}\}, \tag{5.4}$$

so F is in $\mathcal{C}_T$ if and only if $(\bigcup F, F)$ is a subtree of T. We have established in Lemma 2.8 that $\mathcal{C}_T$ is an E-patchwork with $\mathcal{C}_T = (\mathcal{C}_T)^E$ and that this patchwork is ample (in fact, this still holds true in case T is an infinite tree). See Example 2 on page 12 for an explicit example.

6. More generally, let $G = (V, E)$ denote an arbitrary graph. A non-empty subset $F \subseteq E$ is called a *G-patch* if the subgraph $(\bigcup F, F)$ is connected and if, for every two distinct vertices u, $v \in \bigcup F$ and every path p from u to v, we have $\langle p \rangle \subseteq F$. Clearly, if $G = (V, E)$ is a tree, then this definition is in accordance with the definition of Section 2.3. So, generalizing (5.4), we may define

$$\mathcal{C}_G := \{F \subseteq E \mid F \text{ is a } G\text{-patch}\} \tag{5.5}$$

for any graph G; one can check that (as in the case of trees) $\mathcal{C}_G$ is a patchwork, that $\mathcal{C}_G$ is connected if and only if G is connected, and that $\mathcal{C}_G$ is ample. Note also that $F \subseteq E$ is a minimal G-patch if and only if the subgraph $(\bigcup F, F)$ is a 2-connected component of G (provided "bridges of G" are included among the 2-connected components of G, see for example [DMS97] for definitions); in particular, one has $\mathcal{C}_G = (\mathcal{C}_G)^E$ if and only if G is a tree. See Fig. 5.3 for an explicit example.

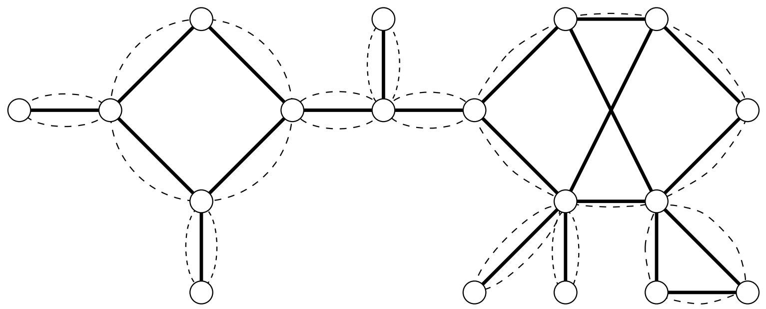


FIGURE 5.3. A graph G and the minimal G-patches.

7. Given a set X, let $\phi : \mathcal{P}(X) \to \mathbb{R}$ denote a map that satisfies

$$\phi(A) + \phi(B) \geq \phi(A \cup B) + \phi(A \cap B) \geq 0 \tag{5.6}$$

for all A, $B \subseteq X$ with $A \cap B \neq \emptyset$. Then

$$\mathcal{C} := \{C \subseteq X \mid \phi(C) \leq 0\}$$

is a patchwork whose clusters will also be called ϕ-clusters. For the proof, we first note that $\phi(A) \geq 0$ must hold for all non-empty subsets $A \subseteq X$, and given two clusters A, $B \subseteq X$ with $A \cap B \neq \emptyset$ and $\phi(A) = \phi(B) = 0$ we infer from (5.6) that $\phi(A \cup B) + \phi(A \cap B) \leq 0$ must hold, so $\phi(A \cup B) = \phi(A \cap B) = 0$.

8. Given finite sets V, E with $\#E \geq 2$ and a relation $R \subseteq E \times V$, we define for every $F \subseteq E$ the set

$$R(F) := \{v \in V \mid \text{there exists } f \in F \text{ with } (f, v) \in R\} \tag{5.7}$$

and

$$r(F) := \#R(F)\,,$$

and for every $e \in E$, we put $r(e) := r(\{e\})$. We suppose $R(E) = V$, and we define a map $\lambda : \mathcal{P}(E) \to \mathbb{R}$ by

$$\lambda(F) := r(F) - \sum_{f\in F} r(f) - (\#F-1)\frac{r(E) - \sum_{e\in E} r(e)}{\#E-1}\,. \tag{5.8}$$

Note that

$$\lambda(F_1) + \lambda(F_2) \geq \lambda(F_1 \cup F_2) + \lambda(F_1 \cap F_2)$$

holds for all $F_1,\ F_2 \subseteq E$; also, we have

$$\lambda(\emptyset) = \frac{r(E) - \sum_{e\in E} r(e)}{\#E-1} \leq 0$$

and $\lambda(E) = 0$, as well as $\lambda(\{e\}) = 0$ for all $e \in E$.

We say that R is *super-additive* if $\lambda(F) \geq 0$ holds for all non-empty subsets $F \subseteq E$, that is, if

$$\frac{\sum_{e\in E} r(e) - r(E)}{\#E-1} \geq \frac{\sum_{f\in F} r(f) - r(F)}{\#F-1} \tag{5.9}$$

holds for all $F \subseteq E$ with $\#F \geq 2$ or, equivalently, if one has

$$r^*(F) \geq \sum_{f\in F} r^*(f) \quad \text{with} \quad r^*(F) := r(F) - \frac{\sum_{e\in E} r(e) - r(E)}{\#E-1} \tag{5.10}$$

for all $F \subseteq E$ with $\#F \geq 1$ (where $r^*(e) := r^*(\{e\})$ for all $e \in E$, of course). Then, the set

$$\mathcal{C}_R := \{F \subseteq E \mid \lambda(F) \leq 0\} = \Big\{F \subseteq E \;\Big|\; r^*(F) \leq \sum_{f\in F} r^*(f)\Big\} \tag{5.11}$$

of *R-patches* is a patchwork satisfying in addition the condition $(\mathcal{C}_R)^E = \mathcal{C}_R$.

9. Note in particular that in case $T = (V, E)$ is a finite tree, we have $\mathcal{C}_R = \mathcal{C}_T$ for

$$R := \{(e, v) \in E \times V \mid v \in e\}\,.$$

Indeed, we have $R(F) = \bigcup F$, $r(F) = \#\bigcup F$ and $r(e) = 2$ for all $e \in E$. From $r(E) = \#V = \#E + 1$, we infer

$$\frac{\sum_{e\in E} r(e) - r(E)}{\#E-1} = \frac{2\,\#E - \#E - 1}{\#E-1} = 1\,.$$

So, if we denote by $\pi_0(V', E')$ the number of connected components of a graph (V', E'), the identity $\#\pi_0(\bigcup F, F) + \#F = \#\bigcup F$ holds for all $F \subseteq E$, and we infer that R is super-additive in view of the inequality

$$\begin{aligned}\frac{\sum_{f\in F} r(f) - r(F)}{\#F-1} &= \frac{2\,\#F - \#F - \#\pi_0(\bigcup F, F)}{\#F-1} \\ &= \frac{\#F - \#\pi_0(\bigcup F, F)}{\#F-1} \leq 1\end{aligned}$$

that holds for all subsets $F \subseteq E$ with $\#F \geq 2$. Finally, to establish $\mathcal{C}_R = \mathcal{C}_T$, we observe that

$$\lambda(F) = \#F + \#\pi_0(\textstyle\bigcup F, F) - 2\,\#F + (\#F - 1) = \#\pi_0(\bigcup F, F) - 1$$

vanishes for some non-empty set $F \subseteq E$ if and only if $(\bigcup F, F)$ is connected.

10. More generally, assume that for some $n \in \mathbb{N}$ we have $r(e) = n+1$ for all $e \in E$, and $\#V = \#E + n$. Then we have

$$\frac{\sum_{e\in E} r(e) - r(E)}{\#E - 1} = n$$

$$\text{and} \quad \frac{\sum_{f\in F} r(f) - r(F)}{\#F - 1} = n + \frac{\#F + n - r(F)}{\#F - 1}$$

for every $F \subseteq E$ with $\#F \geq 2$, so R is super-additive if and only if $r(F) \geq \#F + n$ holds for all non-empty subsets $F \subseteq E$; in this case, a non-empty subset F is an R-patch if and only if $r(F) = \#F + n$ holds. In the following chapters, we will be interested in the particular case $n := 3$.

5.4. Proof of Theorem 4

Throughout this and the next section, we assume $\#\mathcal{C} < \infty$. We will prove Theorem 4 in that case by establishing the implications (i) $\Rightarrow$ (ii) $\Rightarrow$ (iii) $\Rightarrow$ (iv) $\Rightarrow$ (v) $\Rightarrow$ (i) and (v) $\Rightarrow$ (vi) $\Rightarrow$ (iv). Of those, the implications (ii) $\Rightarrow$ (iii) and (vi) $\Rightarrow$ (iv) are trivial, the implication (i) $\Rightarrow$ (ii) follows directly from (Ample 2), and the implication (v) $\Rightarrow$ (vi) follows directly from (Ample 1). The remaining ones will now be established, one by one:

(iii) $\Rightarrow$ (iv):

Obviously, $x \in \bigcup \mathcal{C}$ implies $x \in C_x := \bigcap \mathcal{C}(\supseteq \{x\}) \in \mathcal{C}$ for every patchwork $\mathcal{C}$. However, if (iii) holds, we cannot have $C_x \in \mathcal{C} - \min^*(\mathcal{C})$, so we must have $C_x \in \min^*(\mathcal{C})$ and, hence, $\bigcup \mathcal{C} = \bigcup \min^*(\mathcal{C})$.

Next, assume that $\mathcal{F} = (C_i)_{i\in I}$ is a proper $\mathcal{C}$-family whose trace $(I, \mathcal{I}) = (I_\mathcal{F}, \mathcal{I}_\mathcal{F})$ only involves subsets from $\emptyset^I = \{\emptyset\} \cup \binom{I}{1} \cup \{I\}$. Consider

$$\mathcal{C}^* := \{C \in \mathcal{C}(\subseteq C_I) \mid I(C) = I\},$$

and note that $\mathcal{C}^*$ definitely contains C_I. Now, choose $C^* \in \min(\mathcal{C}^*)$, note that $\emptyset \neq C^* \in \mathcal{C} - \min^*(\mathcal{C})$ must hold in view of $\#I \geq 2$ and $\emptyset \neq C^* \cap C_i$ for all $i \in I$, choose $A, B \in \mathcal{C}_{\subset C^*}$ with $A \cup B = C^*$ and, finally, note that $I = I(C^*) = I(A) \cup I(B)$, $I(A) \neq I \neq I(B)$, and $I(A), I(B) \in \mathcal{I}_\mathcal{F} \subseteq \{\emptyset\} \cup \binom{I}{1} \cup \{I\}$ together implies $2 \leq \#I \leq \#I(A) + \#I(B) \leq 1 + 1 = 2$, as claimed.

(iv) $\Rightarrow$ (v):

Let $\mathcal{F} = (C_i)_{i\in I}$ be a proper $\mathcal{C}$-family with trace $(I, \mathcal{I})$, and assume $J_1, J_2 \in \mathcal{I}$ and $\#\{J \in \mathcal{I} \mid J_1 \subseteq J \subseteq J_2\} = 2$. Recall that $\mathcal{I}(\subseteq J_2 - J_1)$ is a non-empty patchwork, consider the partition $I' := \{J_1\} \cup \max(\mathcal{I}_{\subseteq J_2 - J_1})$, put $\mathcal{F}' := (C_J)_{J\in I'}$, and note that the associated trace $(I', \mathcal{I}') = (I', \mathcal{I}_{\mathcal{F}'})$ must then be a proper trace of $(X, \mathcal{C})$ with $\mathcal{I}' \subseteq \{\emptyset\} \cup \binom{I'}{1} \cup \{I'\}$. So, we must have $\#I' = 2$ which implies $\#\max(\mathcal{I}_{\subseteq J_2 - J_1}) = 1$ and hence, in view of $\{i\} \in \mathcal{I}$ for each $i \in I$,

$$\max(\mathcal{I}_{\subseteq J_2 - J_1}) = \{\bigcup \max(\mathcal{I}_{\subseteq J_2 - J_1})\} = \{\bigcup(\mathcal{I}_{\subseteq J_2 - J_1})\} = \{J_2 - J_1\},$$

that is, $J_2 - J_1 \in \mathcal{I}$, as claimed.

(v) $\Rightarrow$ (i):

Assume $C_1, C_2 \in \mathcal{C}$ and $\#\{C \in \mathcal{C} \mid C_1 \subseteq C \subseteq C_2\} = 2$; put $I := \min^*(\mathcal{C}_{\subseteq C_2})$ and consider the (necessarily proper) $\mathcal{C}$-family $\mathcal{F} = \mathcal{F}_{C_2} := (C)_{C \in I}$ defined in Lemma 5.4. We have $J_1 := \min^*(\mathcal{C}_{\subseteq C_1}) \in \mathcal{I}_{\mathcal{F}}$ and $J_2 := I \in \mathcal{I}_{\mathcal{F}}$ in view of $C_{J_1} = \bigcup \min^*(\mathcal{C}_{\subseteq C_1}) = C_1$ and $C_{J_2} = \bigcup \min^*(\mathcal{C}_{\subseteq C_2}) = C_2$, so we must have $\#\{J \in \mathcal{I} \mid J_1 \subseteq J \subseteq J_2\} = 2$ and hence, $J_2 - J_1 \in \mathcal{I}$ which in turn implies $C_{J_2 - J_1} = C_{J_2} - C_{J_1} = C_2 - C_1 \in \mathcal{C}$, as claimed. □

5.5. Proof of Theorems 5 and 2 in case $\#\mathcal{C} < \infty$

To prove Theorem 5, it is enough to establish the implications (iii) $\Rightarrow$ (ii) $\Rightarrow$ (i) $\Rightarrow$ (iii).

(iii) $\Rightarrow$ (ii):

Let $\mathcal{C}_0 \subseteq \mathcal{C}$ denote a maximal sub-hierarchy of $\mathcal{C}$. Lemma 5.3 states that $\max(\mathcal{C}) \cup \min^*(\mathcal{C}) \subseteq \mathcal{C}_0$ holds. So, we have $\max(\mathcal{C}_0) = \max(\mathcal{C})$ and $\min^*(\mathcal{C}_0) = \min^*(\mathcal{C})$. Moreover, (iii) implies that $\mathcal{C}_0$ must be ample; so, we are done.

(ii) $\Rightarrow$ (i):

Let $\mathcal{C}_0 \subseteq \mathcal{C}$ be an ample hierarchy with $\max(\mathcal{C}_0) = \max(\mathcal{C})$ and $\min^*(\mathcal{C}_0) = \min^*(\mathcal{C})$. In view of Theorem 4: (i) $\Leftrightarrow$ (ii), it is enough to show that, for every non-empty cluster $C \in \mathcal{C} - \min^*(\mathcal{C})$, there exist $A, B \in \mathcal{C}_{\subset C}$ with $A \uplus B = C$. So, for any such C, put $C_0 := \bigcap \mathcal{C}_0(\supseteq C)$ which — in view of $\max(\mathcal{C}_0) = \max(\mathcal{C})$ — must be a member of $\mathcal{C}_0$ that contains C. Moreover, $\emptyset \neq C \notin \min^*(\mathcal{C}) = \min^*(\mathcal{C}_0)$ implies $\emptyset \neq C_0 \notin \min^*(\mathcal{C}_0)$. So, there are $A_0, B_0 \in \mathcal{C}_0(\subset C_0)$ with $A_0 \uplus B_0 = C_0$. Now, putting $A := C \cap A_0$ and $B := C \cap B_0$, we see that $A \uplus B = C$ as well as $A, B \in \mathcal{C}_{\subset C}$ must hold, the latter because of $A_0, B_0 \in \mathcal{C}_0(\subset C_0)$ and $C_0 = \bigcap \mathcal{C}_0(\supseteq C)$ and hence, $C \not\subseteq A_0$ and $C \not\subseteq B_0$.

(i) $\Rightarrow$ (iii):

Assume that $\mathcal{C}_0$ is a maximal sub-hierarchy of some finite ample patchwork $\mathcal{C}$, and note that according to Lemma 5.3, $\mathcal{C}_0$ must contain $\min^*(\mathcal{C})$ and that, hence, $C_0 = \bigcup \min^*(\mathcal{C}_0(\subseteq C_0))$ must hold for all C_0 in $\mathcal{C}_0$. Now, consider some non-empty cluster $C_0 \in \mathcal{C}_0 - \min^*(\mathcal{C}_0)$, and note that

$$\mathcal{C}^* := \max(\mathcal{C}_0(\subset C_0))$$

must be a partition of C_0 and that the associated $\mathcal{C}$-family $\mathcal{F} := (C)_{C \in \mathcal{C}^*}$ is proper and hence, in view of Theorem 4: (i) $\Leftrightarrow$ (v), it must be ample. Moreover, we must have $\#\mathcal{C}^* = 2$ because $\#\mathcal{C}^* \geq 3$ would — in view of Theorem 4: (i) $\Leftrightarrow$ (iv) — imply the existence of some proper subset $\mathcal{J} \subsetneqq \mathcal{C}^*$ with $\#\mathcal{J} > 1$ and hence, of a cluster $C_{\mathcal{J}} = \bigcup_{C^* \in \mathcal{J}} C^* \in \mathcal{C}$ that would not be contained in $\mathcal{C}_0$ (by definition of $\mathcal{C}^*$) while $\mathcal{C}_0 \cup \{C_{\mathcal{J}}\}$ would clearly be a sub-hierarchy of $\mathcal{C}$, contradicting the assumption that $\mathcal{C}_0$ was a maximal sub-hierarchy of $\mathcal{C}$. Hence, C_0 is the disjoint union of its two proper subclusters in $\mathcal{C}^*$.

Consequently, $\mathcal{C}_0$ satisfies condition (ii) in Theorem 4, so according to that theorem, it must be ample. □

Theorem 2, limited to finite sets $\mathcal{C}$, follows immediately from Theorem 5 and Lemma 5.1.

5.6. Applications of patchworks

The following lemma deals with the case that a collection $\mathcal{C}$ of subsets of X is "nearly" an ample patchwork. Recall that, given an ample cluster system $\mathcal{C}$ and a cluster C_0, the cluster system $\mathcal{C}_{\subseteq C_0}$ is ample, too. The following simple observation will be useful later:

LEMMA 5.8. *Let X denote a finite set, and let $\mathcal{C} \subseteq \mathcal{P}(X)$ denote a patchwork with $X = \bigcup \mathcal{C}_{\subset X} \in \mathcal{C}$. Then, the following assertions are equivalent:*

(i) *$\mathcal{C}_{\subset X}$ is ample, yet $\mathcal{C}$ itself is not ample;*
(ii) *$\mathcal{C}_{\subset X}$ is an ample patchwork with $\#\max(\mathcal{C}_{\subset X}) \geq 3$;*
(iii) *$\#\max(\mathcal{C}_{\subset X}) \geq 3$, and $\mathcal{C}_{\subseteq C}$ is ample for all $C \in \max(\mathcal{C}_{\subset X})$.*

PROOF. The equivalence (ii) $\Longleftrightarrow$ (iii) is trivial.
(i) $\Rightarrow$ (ii):

Given clusters C_1, $C_2 \in \mathcal{C}_{\subset X}$ with non-empty intersection, we know that either $C_1 \cup C_2 \in \mathcal{C}_{\subset X}$, or $C_1 \cup C_2 = X$ must hold. But the latter case would imply — in view of Theorem 4: (i) $\Leftrightarrow$ (iii) — that $\mathcal{C}$ itself is ample. So, $\mathcal{C}_{\subset X}$ is a patchwork, and $\#\max(\mathcal{C}_{\subset X}) \geq 3$ follows from Theorem 4: (i) $\Leftrightarrow$ (ii).
(ii) $\Rightarrow$ (i):

Note that $\max(\mathcal{C}_{\subset X})$ is a partition of X, so $\#\max(\mathcal{C}_{\subset X}) \geq 3$ implies that there exist no disjoint clusters C_1, $C_2 \in \mathcal{C}_{\subset X}$ with $C_1 \cup C_2 = X$, and Theorem 4: (i) $\Leftrightarrow$ (ii) states that $\mathcal{C}$ cannot be ample. □

Now, suppose we are given a finite set X and an X-patchwork $\mathcal{C}$ with $\mathcal{C} = \mathcal{C}^X$; we want to check whether this patchwork is ample and, in this case, construct a maximal hierarchy $\mathcal{C}' \subseteq \mathcal{C}$. A simple algorithm to do this makes use of Condition (vi) of Theorem 4: We start by defining the partition $\mathcal{C}_1 := \binom{X}{1} \subseteq \mathcal{C}$. We check whether there exist sets A, $B \in \mathcal{C}_1$ with $A \cup B \in \mathcal{C}$ in which case we put $\mathcal{C}_2 := \mathcal{C}_1 \cup \{A \cup B\} - \{A, B\}$ (since $(C)_{C \in \mathcal{C}_1}$ is a proper $\mathcal{C}$-family, such A, $B \in \mathcal{C}_1$ must exist by Theorem 4: (i) $\Leftrightarrow$ (vi) in case $\mathcal{C}$ is ample); otherwise, we `STOP`. If we iterate this process of replacements $\#X - 1$ times, we eventually end up with the set $\mathcal{C}_{\#X} = \{X\}$. In this case, $\mathcal{C}$ is ample and

$$\mathcal{C}' := \{\emptyset\} \cup \bigcup_{j=1}^{\#X} \mathcal{C}_j \ \subseteq \mathcal{C}$$

is a maximal X-hierarchy contained in $\mathcal{C}$. Otherwise, $\mathcal{C}$ is not ample.

Suppose that we can check whether an arbitrary subset $C \subseteq X$ is a member of $\mathcal{C}$ in constant time, independently of the choice of C. Then, obviously, this algorithm is of order $O((\#X)^3)$, provided that every other command (like forming the union of two disjoint sets) can be carried out in constant time, too. However, an even faster algorithm can be designed:

LEMMA 5.9. *Algorithm 1 computes, given a finite patchwork $\mathcal{C} \subseteq \mathcal{P}(X)$ with $\mathcal{C} = \mathcal{C}^X$ as input, a maximal hierarchy $\mathcal{C}' \subseteq \mathcal{C}$ if $\mathcal{C}$ is ample; otherwise, it returns* `False`. *Given an oracle that, for any two disjoint clusters C_1, $C_2 \in \mathcal{C}$, is able to tell us in constant time whether $C_1 \cup C_2 \in \mathcal{C}$ holds, the algorithm is of order $O((\#X)^2)$, provided*

1 Let $\mathcal{C}' := \binom{X}{1} \cup \{\emptyset\}$, $\mathcal{A} := \binom{X}{1}$, and $\mathcal{B} := \emptyset$;
2 `While` $\mathcal{A} \neq \{X\}$ `And` $\mathcal{A} \neq \emptyset$ `Do`
3   Choose some $A \in \mathcal{A}$;
4   `If` $A \cup B \notin \mathcal{C}$ holds for all $B \in \mathcal{B}$
5   `Then`
6   Let $\mathcal{A} := \mathcal{A} - \{A\}$;
7   Let $\mathcal{B} := \mathcal{B} \cup \{A\}$
8   `Else`
9   Choose some $B \in \mathcal{B}$ with $A \cup B \in \mathcal{C}$;
10   Let $\mathcal{A} := \mathcal{A} \cup \{A \cup B\} - \{A\}$;
11   Let $\mathcal{B} := \mathcal{B} - \{B\}$;
12   Let $\mathcal{C}' := \mathcal{C}' \cup \{A \cup B\}$
13   `End If`
14 `End While`;
15 `If` $\mathcal{A} = \{X\}$ `Then Return(`$\mathcal{C}'$`) Else Return(False)`.

ALGORITHM 1: Constructing a maximal hierarchy $\mathcal{C}' \subseteq \mathcal{C}$ (written in PSEUDO-PASCAL). INPUT: Cluster system $\mathcal{C} \subseteq \mathcal{P}(X)$ with $\mathcal{C} = \mathcal{C}^X$; OUTPUT: Maximal hierarchy $\mathcal{C}' \subseteq \mathcal{C}$ or `False`.

that every other command (like forming that union of those two disjoint sets) can be carried out in constant time.

PROOF. Clearly, at the beginning of each step of the `While`-loop (line 2), the following holds: $\mathcal{A} \cap \mathcal{B} = \emptyset$, $X \notin \mathcal{B}$, $\mathcal{C}'$ is an ample hierarchy with $\binom{X}{1} \subseteq \mathcal{C}'$, $\mathcal{A} \cup \mathcal{B}$ coincides with the associated partition $\max(\mathcal{C}')$ of X, and $B \cup B' \notin \mathcal{C}$ holds for all distinct clusters B, $B' \in \mathcal{B}$.

If the algorithm stops with $\mathcal{A} = \emptyset$, then $\mathcal{B} \subseteq \mathcal{C}'$ forms a partition of X. Since $B \cup B' \notin \mathcal{C}$ holds for all distinct clusters B, $B' \in \mathcal{B} = \max(\mathcal{C}')$, we infer that $\#\mathcal{B} \geq 3$ must hold. Hence, Theorem 4: (i) $\Leftrightarrow$ (vi) implies that $\mathcal{C}$ cannot be ample.

If the algorithm stops with $\mathcal{A} = \{X\}$ then it is clear that the output $\mathcal{C}'$ will always be a maximal hierarchy contained in $\mathcal{C}$, whether or not $\mathcal{C}$ is a patchwork.

To check the runtime behavior of the algorithm, note first that the number $2\,\#\mathcal{A} + \#\mathcal{B}$ is decreased by one in every step of the `While`-loop (lines 2–14), so we need at most $2n-2$ steps of the loop (where $n := \#X$) because after $2n-2$ steps, we have $2\,\#\mathcal{A} + \#\mathcal{B} = 2$ and hence, $\mathcal{A} = \{X\}$ or $\mathcal{A} = \emptyset$. Furthermore, at each step, at most $\#\mathcal{B}$ sets are checked for membership in $\mathcal{C}$ (lines 4 and 9). Let $k := 2n - 2\,\#\mathcal{A} - \#\mathcal{B} + 1$ denote the *step number* of the `While`-loop, and suppose that the loop was left at step K. We denote the cardinality of $\mathcal{B}$ at the beginning of step k of the loop by b_k for $k = 1, \ldots, K$. Then $b_k = 2n + 1 - k - 2\,\#\mathcal{A} \leq 2n - 1 - k$ for $\mathcal{A} \neq \emptyset$ and $b_k \leq k - 1$ holds, the latter in view of $b_1 = 0$ and $b_{i+1} \in \{b_i \pm 1\}$ for $i = 1, \ldots, K-1$. Together, we get

$$b_k \leq \min(2n - 1 - k, k - 1) \quad \text{for} \quad k = 1, \ldots, K$$

and

$$\sum_{k=1}^{K} b_k \leq \sum_{k=1}^{n}(k-1) + \sum_{k=n+1}^{2n-2}(2n-1-k)$$
$$= \frac{n(n-1)}{2} + \frac{(n-1)(n-2)}{2} = (n-1)^2$$

and hence, at most $(n-1)^2 - 1 = n(n-2)$ subsets of X need to be checked for membership in $\mathcal{C}$, since we already know $X \in \mathcal{C}$. □

Note that Algorithm 1 is "best possible" in the sense that *every* algorithm computing a maximal hierarchy $\mathcal{C}' \subseteq \mathcal{C} \subseteq \mathcal{P}(X)$ and depending on our oracle (as described in Lemma 5.9) needs at least $n(n-2)$ calls of the oracle in the worst case, under the assumption that no further information except $\emptyset^X \subseteq \mathcal{C}$ is available: Constructing an example while being questioned, the oracle can decide to claim $A \mathbin{\dot\cup} B \notin \mathcal{C}$ for all but the last pair of disjoint one-element subsets $A,\, B \in \binom{X}{1}$, and it then can go on forcing us, for every subset $A \in \mathcal{C}$ of cardinality i found so far, to ask $n-i$ questions before a (one-element) subset $B \subseteq X - A$ with $A \mathbin{\dot\cup} B \in \mathcal{C}$ is found. In the more general setting of an oracle that is able to tell us whether $C \in \mathcal{C}$ holds for an *arbitrary* subset $C \subseteq X$, we still conjecture:

Conjecture 1. *Given a finite patchwork $\mathcal{C} \subseteq \mathcal{P}(X)$ with $\mathcal{C} = \mathcal{C}^X$; and given an oracle that, for every cluster $C \subseteq X$, is able to tell us whether $C \in \mathcal{C}$ holds. Then every algorithm computing a maximal hierarchy $\mathcal{C}' \subseteq \mathcal{C}$ and depending on this oracle only to check for membership in $\mathcal{C}$ will need at least $\#X(\#X-2)$ calls of the oracle in the worst case.*

Conjecture 1 does *not* state that, for certain instances of the problem, there exist no algorithms computing a maximal hierarchy $\mathcal{C}' \subseteq \mathcal{C}$ in runtime faster than $O((\#X)^2)$, because such algorithms might directly use the structure of the underlying patchwork instead of relying on an "oracle". It is remarkable that apart from Algorithm 1, there exists a differently structured algorithm that in fact will ask the oracle whether $C \in \mathcal{C}$ holds for subsets $C \subseteq X$ with no subclusters $A,\, B \in \mathcal{C}$ satisfying $A \mathbin{\dot\cup} B = C$; still, this algorithm requires only $\#X(\#X-2)$ calls of the oracle in the worst case: Algorithm 2 (using a recursive "top-down" approach) is depicted on page 49.

We will not formally prove that Algorithm 2 returns a maximal hierarchy $\mathcal{C}' \subseteq \mathcal{C}$ in case it does exist, and `False` otherwise. Just note that lines 10–31 do in fact the same as Algorithm 1, where just the order of processing the sets might be changed; while in lines 6–8, we recursively construct a maximal hierarchy contained in $\mathcal{C}_{\subseteq C}$ for a cluster $C \in \mathcal{C}$ of cardinality $\#C = \#X - 1$ if such a cluster exists.

We will also give only a *sketch proof* for the fact that this algorithm uses at most $n(n-2)$ calls of the oracle (where $n := \#X$): Firstly, the `If`-statement in line 4 uses exactly n oracle calls. We use induction to see that the `Then`-part (lines 6–8) uses at most $(n-1)(n-3) = n(n-2) - n - (n-3)$ calls for $n \geq 3$. For the `Else`-part (lines 10–31), we need at most $\binom{n}{2} - 2\binom{a}{2}$ calls to execute line 10 using an approach comparable to Algorithm 1, where $a \in \mathbb{N}$ denotes the cardinality of $\mathcal{A}$ in line 8. It is easily seen that the maximal number of calls in lines 12–28 is obtained when $\mathcal{B}_1$ is

```
Procedure FindHierarchyTD ($X$: Set, $\mathcal{C}$: $X$-Cluster System):
                                                  $X$-Hierarchy Or False;

Let $k := \#X - 1$;
If $\#X = 2$
    Then Return($\emptyset^X$)
Else If $C \in \mathcal{C}$ holds for some $C \in \binom{X}{k}$
Then
    Choose $C \in \binom{X}{k}$ with $C \in \mathcal{C}$;
    Let $\mathcal{C}' :=$ FindHierarchyTD ($C$, $\mathcal{C}_{\subseteq C}$);
    If $\mathcal{C}' \neq$ False Then Return($\mathcal{C}' \cup \{X, X - C\}$) Else Return(False)
Else
    Let $\mathcal{A} \subseteq \binom{X}{2} \cap \mathcal{C}$ denote a maximal subset such that $A \cap A' \neq \emptyset$
        implies $A = A'$ for all $A$, $A' \in \mathcal{A}$;
    Let $\mathcal{C}' := \emptyset^X \cup \mathcal{A}$, $\mathcal{B}_1 := \{\{x\} \mid x \in X - \bigcup\mathcal{A}\}$, and $\mathcal{B}_{2+} := \emptyset$;
    While $\mathcal{A} \neq \emptyset$ Do
        Choose some $A \in \mathcal{A}$;
        If there exists $B \in \mathcal{B}_1$ with $A \cup B \in \mathcal{C}$
        Then
            Choose some $B \in \mathcal{B}_1$ with $A \cup B \in \mathcal{C}$;
            Let $\mathcal{A} := \mathcal{A} \cup \{A \cup B\} - \{A\}$;
            Let $\mathcal{B}_1 := \mathcal{B}_1 - \{B\}$;
            Let $\mathcal{C}' := \mathcal{C}' \cup \{A \cup B\}$
        Else If $(\#\mathcal{A} + \#\mathcal{B}_{2+}) \geq 3$ And there exists $B \in \mathcal{B}_{2+}$
            with $A \cup B \in \mathcal{C}$
        Then
            Choose some $B \in \mathcal{B}_{2+}$ with $A \cup B \in \mathcal{C}$;
            Let $\mathcal{A} := \mathcal{A} \cup \{A \cup B\} - \{A\}$;
            Let $\mathcal{B}_{2+} := \mathcal{B}_{2+} - \{B\}$;
            Let $\mathcal{C}' := \mathcal{C}' \cup \{A \cup B\}$
        Else
            Let $\mathcal{A} := \mathcal{A} - \{A\}$;
            Let $\mathcal{B}_{2+} := \mathcal{B}_{2+} \cup \{A\}$;
        End If
    End While;
    If $\mathcal{B}_1 = \emptyset$ And $\#\mathcal{B}_{2+} = 2$ Then Return($\mathcal{C}'$) Else Return(False)
End If.
```

ALGORITHM 2: An alternative algorithm to construct a maximal hierarchy $\mathcal{C}' \subseteq \mathcal{C}$ (written in PSEUDO-PASCAL) that checks "$C \in \mathcal{C}$" for some set $C \subseteq X$ that is not the disjoint union of two subsets A, $B \in \mathcal{C}_{\subset C}$. INPUT: Set X, cluster system $\mathcal{C} \subseteq \mathcal{P}(X)$ with $\mathcal{C} = \mathcal{C}^X$; OUTPUT: Maximal hierarchy $\mathcal{C}' \subseteq \mathcal{C}$ or False.

holding its initial (and therefore maximal) value as long as possible; to also maximize the cardinality of $\mathcal{B}_{2+}$ we move all but one sets from $\mathcal{A}$ to $\mathcal{B}_{2+}$ (lines 27–28) and, in a second step, build the pairwise union of the remaining set in $\mathcal{A}$ with all but one set from $\mathcal{B}_{2+}$ (lines 20–23). This approach uses

$$(n-2a)+(n-2a+1)+\cdots+(n-2a+a-2)=\binom{n-a-1}{2}-\binom{n-2a}{2}$$

call of the oracle for the first, and

$$(n-a-1)+(n-a-2)+\cdots+(n-2a+2)=\binom{n-a}{2}-\binom{n-2a+2}{2}$$

calls for the second step. In the remaining, we need at most

$$(n-2a)+(n-2a-1)+\cdots+1=\binom{n-2a+1}{2}$$

calls until the algorithms stops. For lines 10–31, this all sums up to

$$\begin{aligned}\text{oracle calls} &\leq \binom{n}{2}-2\binom{a}{2}+\binom{n-a-1}{2}-\binom{n-2a}{2}+\binom{n-a}{2}-\binom{n-2a+2}{2}+\binom{n-2a+1}{2}\\ &= n^2-3n-2a^2+4a\\ &\leq n(n-2)-n\end{aligned}$$

in view of $a \geq 2$, and equality holds when applying the algorithm to the patchwork $(X, \mathcal{C})$ for $X := \{1, \ldots, n\}$ and

$$\mathcal{C} := \emptyset^X \cup \big\{\{1,2\},\{3,k\} \bigm| k=4,\ldots,n\big\}\,.$$

Because the runtime behavior strongly depends on the structure of the underlying patchwork $\mathcal{C}$, it does not make much sense to ask for the average runtime of the algorithms (unless one first specifies a probability distribution on the set of patchworks to be checked for ampleness). Finally, we note that the runtime behavior of Algorithm 1 does not change under the more realistic assumption that forming the union of two disjoint sets can be carried out in linear time in $\#X$, if we further suppose that we can check in constant time whether, given disjoint clusters C_1, $C_2 \in \mathcal{C}$, their union $C_1 \uplus C_2$ is a member of $\mathcal{C}$. We will show in Section 8.1 how to construct a "constant time oracle" of this type for the application we have in mind — that is, checking whether a "minimum" set of quartet splits is concordant with a unique phylogenetic tree.

Of course, if we do not restrict ourselves to patchworks and ask quite generally whether an *arbitrary* cluster system $\mathcal{C} \subseteq \mathcal{P}(X)$ with $\mathcal{C} = \mathcal{C}^X$ is ample or contains a maximal hierarchy, then we cannot anymore apply the above algorithms. Indeed, it is well known that it is quite hard to deal with these questions in a computationally efficient way. For the latter problem, Bryant [Bry96] presents an algorithm of order $O(\#X \cdot \#\mathcal{C}^2)$.

5.7. Dropping the finiteness restriction

We will now show how our results can be established for arbitrary patchworks $\mathcal{C}$. First, recall that — if not explicitly stated otherwise — all results established in Section 5.1 still hold true for an arbitrary (that is, possibly infinite) cluster system $\mathcal{C}$.

Now, we want to characterize those (infinite) cluster systems that act “finite-like”: A cluster system $\mathcal{C} \subseteq \mathcal{P}(X)$ is *finitary* if $\bigcup \mathcal{C}' \in \mathcal{C}$ and $\bigcap \mathcal{C}' \in \mathcal{C}$ holds for all non-empty chains $\mathcal{C}'$ contained in $\mathcal{C}$. Here, we will also consider cluster systems $\mathcal{C}$ for which $\mathcal{C} \cup \{\emptyset\}$ is finitary. Such cluster systems will be called *almost finitary.*

Clearly, every finite cluster system is finitary, because

$$\min(\mathcal{C}') = \left\{\bigcap \mathcal{C}'\right\} \quad \text{and} \quad \max(\mathcal{C}') = \left\{\bigcup \mathcal{C}'\right\}$$

holds for every *finite* chain $\mathcal{C}'$. More generally, the union of any (almost) finitary cluster system with a finite cluster system is (almost) finitary, too. Furthermore, if a cluster system $\mathcal{C} \subseteq \mathcal{P}(X)$ is almost finitary then $\mathcal{C}^X$ is finitary.

We can easily check that $\mathcal{C}_{\subseteq Y}$ and $\mathcal{C}_{\supseteq Y}$ are finitary for every subset $Y \subseteq X$ and every finitary cluster system $\mathcal{C} \subseteq \mathcal{P}(X)$; that for every cluster C in a finitary cluster system $\mathcal{C}$, there exists some $C_{\max} \in \max(\mathcal{C})$ and some $C_{\min} \in \min(\mathcal{C})$ with $C_{\min} \subseteq C \subseteq C_{\max}$; and finally, that (Ample 3) holds for every cluster C in an almost finitary and ample cluster system $\mathcal{C}$, where the proof follows verbatim from the finite case:

LEMMA 5.10. *For every ample and almost finitary set system $\mathcal{C} \subseteq \mathcal{P}(X)$ and every $Y \subseteq X$, one has*

$$\bigcup \mathcal{C}_{\subseteq Y} = \bigcup \min{}^*(\mathcal{C}_{\subseteq Y}).$$

Recalling the definition of patchworks from Section 5.1, we note that every patchwork is almost finitary, and that a hierarchy is a patchwork if and only if it is almost finitary. More generally, a cluster system $\mathcal{C}$ is a patchwork if and only if it is almost finitary and satisfies the condition (Patchwork) for all *finite* non-empty subsets $\mathcal{C}' \subseteq \mathcal{C}$ — or, equivalently, for all subsets $\mathcal{C}' \subseteq \mathcal{C}$ with $\#\mathcal{C}' = 2$:

Suppose that $\mathcal{C}$ satisfies these two conditions, and let $\mathcal{C}'$ denote an arbitrary non-empty subsystem of $\mathcal{C}$ with $\emptyset \neq Z_0 := \bigcap \mathcal{C}'$, then there must exist some $C_0 \in \min(\mathcal{C}_{\supseteq Z_0})$, and we must have $Z_0 \subseteq C_0 \cap C' \in \mathcal{C}$ and hence, $C_0 \cap C' = C_0$ for all $C' \in \mathcal{C}'$. In turn, this implies $Z_0 \subseteq C_0 \subseteq \bigcap \mathcal{C}' = Z_0$ and hence, $Z_0 = C_0 \in \mathcal{C}$. The claim $Z_1 := \bigcup \mathcal{C}' \in \mathcal{C}$ follows analogously by considering some cluster $C_1 \in \max(\mathcal{C}_{\subseteq Z_1} \cap \mathcal{C}_{\supseteq Z_0})$ and observing that $C_1 \cup C' = C_1$ must hold for all $C' \in \mathcal{C}'$.

Finally, we recall from [BD00a] that a hierarchy $\mathcal{C}_0$ is a maximal X-hierarchy if and only if $\mathcal{C}_0$ is finitary and ample and $\mathcal{C}_0 = \mathcal{C}^X$ holds, and that (conversely) every cluster system $\mathcal{C}_0 \subseteq \mathcal{P}(X)$ with $\mathcal{C}_0 = \mathcal{C}_0^X$ that is minimal among all finitary and ample cluster systems $\mathcal{C} \subseteq \mathcal{P}(X)$ with $\mathcal{C} = \mathcal{C}^X$ is necessarily a (maximal) X-hierarchy.

Now, we give some examples showing that the situation can get more complicated for infinite cluster systems $\mathcal{C}$:

Examples.

11. For $X := \mathbb{N}$, the set

$$\mathcal{C} := \left\{\{1, \ldots, n\} \mid n \in \mathbb{N}\right\}$$

is a chain; all cluster of $\mathcal{C}$ are finite and $\mathcal{C}$ is countable infinite; and $\max(\mathcal{C}) = \emptyset$ holds. Note further that $\mathcal{C} \cup \{X, \emptyset\}$ is finitary, and that $\mathcal{C}^{\mathbb{N}}$ is ample and finitary — what is equivalent to the fact that $\mathcal{C}^{\mathbb{N}}$ is a maximal $\mathbb{N}$-hierarchy.

12. For $X := \mathbb{Q}$, the set

$$\mathcal{C} := \Big\{\{q \in \mathbb{Q} \mid q \leq \alpha\} \mid \alpha \in \mathbb{R}\Big\}$$

is a chain, but $\mathcal{C}$ is not countable in view of

$$\alpha = \sup\{q \in \mathbb{Q} \mid q \leq \alpha\} \quad \text{for all } \alpha \in \mathbb{R}\,.$$

Note that $\mathcal{C}$ is ample and that $\max(\mathcal{C}) = \min(\mathcal{C}) = \min^*(\mathcal{C}) = \emptyset$ holds. But the hierarchy $\mathcal{C}^{\mathbb{Q}}$ is neither ample nor finitary: For $\alpha \in \mathbb{Q}$, the one-element subset $\{\alpha\}$ is a maximal $\mathcal{C}^{\mathbb{Q}}$-subcluster of $\{q \in \mathbb{Q} \mid q \leq \alpha\}$; yet, $\{q \in \mathbb{Q} \mid q < \alpha\}$ does not belong to $\mathcal{C}^{\mathbb{Q}}$. And, given $C := \{q \in \mathbb{Q} \mid q \leq \alpha\} \in \mathcal{C}$ for $\alpha \in \mathbb{R}$ then $\mathcal{C}_{\subset C}$ is a chain in $\mathcal{C}^{\mathbb{Q}}$ with $\bigcup \mathcal{C}_{\subset C} = \{q \in \mathbb{Q} \mid q < \alpha\} \notin \mathcal{C}^{\mathbb{Q}}$.

13. For $X := \mathbb{R}$, the set

$$\mathcal{C} := \big\{(-\infty, \alpha),\ (-\infty, \alpha] \mid \alpha \in \mathbb{R}\big\} \cup \{\mathbb{R}, \emptyset\} \cup \tbinom{\mathbb{R}}{1}$$

is an ample and finitary $\mathbb{R}$-hierarchy with $\mathcal{C}^{\mathbb{R}} = \mathcal{C}$ and hence, $\min^*(\mathcal{C}) = \binom{\mathbb{R}}{1}$.

Given a cluster $C_2 = (-\infty, \beta) \in \mathcal{C}$ for some $\beta \in \mathbb{R}$, then $C_2 - C_1 \notin \mathcal{C}$ holds for all proper subclusters $C_1 \in \mathcal{C}_{\subset C_2}$: If $C_1 = (-\infty, \alpha)$ for some $\alpha < \beta$ we get $C_2 - C_1 = [\alpha, \beta) \notin \mathcal{C}$, if $C_1 = (-\infty, \alpha]$ for $\alpha < \beta$ then $C_2 - C_1 = (\alpha, \beta) \notin \mathcal{C}$ and finally, $C_1 = \{\alpha\}$ for $\alpha < \beta$ implies $C_2 - C_1 = (-\infty, \alpha) \mathbin{\dot\cup} (\alpha, \beta) \notin \mathcal{C}$.

Furthermore, given $C \subseteq \mathbb{R}$ with $\#C = 2$ then $C \notin \mathcal{C}$ must hold, because all clusters in $\mathcal{C}$ contain exactly zero, one, or infinitely many elements.

When establishing our results in the infinite case, we will make repeated use of Zorn's Lemma:

Zorn's Lemma. Every inductively ordered set has (at least) one maximal element.

Remark 5.11. Let $\mathcal{C}$ denote an arbitrary cluster system. The set of chains in $\mathcal{C}$,

$$\mathrm{Chain}_{\mathcal{C}} := \{\mathcal{C}' \subseteq \mathcal{C} \mid \mathcal{C}' \text{ is a chain}\}$$

is finitary and hence, inductively ordered by inclusion: Given a chain of chains $\{\mathcal{C}_i\}_{i \in I} \subseteq \mathrm{Chain}_{\mathcal{C}}$, then $C_1, C_2 \in \bigcup(\{\mathcal{C}_i\}_{i \in I}) = \bigcup_{i \in I} \mathcal{C}_i$ implies the existence of indices $j, k \in I$ with $C_1 \in \mathcal{C}_j$ and $C_2 \in \mathcal{C}_k$. As $\{\mathcal{C}_i\}_{i \in I}$ is a chain we may suppose (without loss of generality) that $\mathcal{C}_j \subseteq \mathcal{C}_k$ and hence, $C_1 \in \mathcal{C}_k$ holds — what already implies $C_1 \cap C_2 \in \{C_1, C_2\}$ since $\mathcal{C}_k$ is a chain. This establishes that $\bigcup(\{\mathcal{C}_i\}_{i \in I})$ is a chain, and because we can easily see that $\bigcap(\{\mathcal{C}_i\}_{i \in I}) = \bigcap_{i \in I} \mathcal{C}_i$ is also a chain, we have established that $\mathrm{Chain}_{\mathcal{C}}$ is finitary.

For an arbitrary chain of chains $\{\mathcal{C}_i\}_{i \in I} \subseteq \mathrm{Chain}_{\mathcal{C}}$ we clearly have

$$\mathcal{C}_j \subseteq \bigcup_{i \in I} \mathcal{C}_i = \bigcup\big(\{\mathcal{C}_i\}_{i \in I}\big) \in \mathcal{C}$$

for all indices $j \in I$ what implies that $\mathrm{Chain}_{\mathcal{C}}$ is inductively ordered by inclusion.

In the following, we will show that Theorems 2 and 3 hold *verbatim* for infinite patchworks, while we have to restate Theorems 4 and 5 in case of infinite patchworks as follows:

Theorem 6. *For every patchwork $\mathcal{C}$, the following assertions are equivalent:*

(i) $\mathcal{C}$ *is ample;*
(ii) *for every non-empty cluster* $C \in \mathcal{C} - \min^*(\mathcal{C})$*, there exist either two disjoint clusters* A*,* $B \in \mathcal{C}_{\subset C}$ *with* $A \dot\cup B = C$*, or a (necessarily infinite) chain* $\mathcal{C}' \subseteq \mathcal{C}_{\subset C}$ *with* $\bigcup \mathcal{C}' = C$*;*
(iii) *for every non-empty cluster* $C \in \mathcal{C} - \min^*(\mathcal{C})$*, there exist either two clusters* A*,* $B \in \mathcal{C}_{\subset C}$ *with* $A \cup B = C$*, or a (necessarily infinite) chain* $\mathcal{C}' \subseteq \mathcal{C}_{\subset C}$ *with* $\bigcup \mathcal{C}' = C$*;*
(iv) $\bigcup \mathcal{C} = \bigcup \min^*(\mathcal{C})$ *holds and, for every proper trace* $(I, \mathcal{J})$ *of* $(X, \mathcal{C})$ *with* $\mathcal{J} \subseteq \emptyset^I = \{\emptyset\} \cup \binom{I}{1} \cup \{I\}$*, we have* $\#I = 2$*;*
(v) $\bigcup \mathcal{C} = \bigcup \min^*(\mathcal{C})$ *holds, and every proper trace* $(I, \mathcal{J})$ *of* $(X, \mathcal{C})$ *is ample;*
(vi) $\bigcup \mathcal{C} = \bigcup \min^*(\mathcal{C})$ *holds and, for every proper* $\mathcal{C}$*-family* $\mathcal{F}$ *of* $(X, \mathcal{C})$*, there exist either two distinct indices* i*,* $j \in I_{\mathcal{F}}$ *with* $\{i, j\} \in \mathcal{J}_{\mathcal{F}}$*, or there exists a non-empty sub-chain* $\mathcal{J}' \subseteq \mathcal{J}_{\mathcal{F}}$ *such that every element* $J \in \mathcal{J}'$ *has infinite cardinality and* $\# \bigcap \mathcal{J}' \leq 1$ *holds.*

THEOREM 7. *Let* $(X, \mathcal{C})$ *denote a patchwork. Then, the following assertions are equivalent:*

(i) $\mathcal{C}$ *is ample;*
(ii) $\mathcal{C}$ *contains an ample and almost finitary hierarchy* $\mathcal{C}_0$ *with* $\max(\mathcal{C}_0) = \max(\mathcal{C})$ *and* $\min^*(\mathcal{C}_0) = \min^*(\mathcal{C})$*;*
(iii) *every maximal sub-hierarchy* $\mathcal{C}_0$ *of* $\mathcal{C}$ *is ample.*

REMARK 5.12. It is tempting to speculate about how much of our concepts, arguments, and results carry over from cluster systems to subsets of lattices. Clearly, the problem here is to find a good working definition of ampleness in this more general situation.

The main obstacle for just copying the proofs given above for the case $\#\mathcal{C} < \infty$ is that, unfortunately, the assertions (Ample 1–2) do not hold *verbatim* for ample and finitary cluster systems, see Example 13. Instead, we have:

LEMMA 5.13. *Given an ample, finitary cluster system* $(X, \mathcal{C})$ *and a non-empty cluster* $C_0 \in \mathcal{C} - \min^*(\mathcal{C})$*, the following holds:*

(i) *either, there exist distinct clusters* C_1*,* $C_2 \in \min^*(\mathcal{C}_{\subseteq C_0})$ *with* $C_1 \cup C_2 \in \mathcal{C}$*, or there exists an infinite chain* $\mathcal{C}' \subseteq \mathcal{C}_{\subseteq C_0} \setminus (\min^*(\mathcal{C}) \cup \{\emptyset\})$ *with* $\bigcap \mathcal{C}' \in \min^*(\mathcal{C}) \cup \{\emptyset\}$*;*
(ii) *either,* C_0 *is the union of two proper, disjoint subclusters, or there exists a (necessarily infinite) chain* $\mathcal{C}' \subseteq \mathcal{C}_{\subset C_0}$ *with* $\bigcup \mathcal{C}' = C_0$*.*

PROOF. *(i)* Consider $\mathcal{C}^* := \mathcal{C}_{\subseteq C_0} \setminus (\min^*(\mathcal{C}) \cup \{\emptyset\})$. If $\mathcal{C}^*$ is finitary, there exists some (necessarily non-empty) cluster $C^* \in \min(\mathcal{C}^*)$, and this cluster C^* must contain some $C_1 \in \min^*(\mathcal{C})$ in view of Lemma 5.10. For this C_1, we must have $\#\{C' \in \mathcal{C} \mid C_1 \subseteq C' \subseteq C^*\} = 2$ and hence, $C_2 := C^* - C_1 \in \min^*(\mathcal{C})$; so, the first alternative holds true. Otherwise, there exists an infinite chain $\mathcal{C}' \subseteq \mathcal{C}^*$ with $\bigcap \mathcal{C}' \in \mathcal{C}_{\subseteq C_0} - \mathcal{C}^* \subseteq \min^*(\mathcal{C}) \cup \{\emptyset\}$ as claimed.

(ii) By Remark 5.11, the set of sub-chains of any cluster system $\mathcal{C}$ is inductively ordered, so Zorn's Lemma implies that a maximal chain $\mathcal{C}' \subseteq \mathcal{C}_{\subset C_0}$ must exist. If $C' := \bigcup \mathcal{C}' \subsetneq C_0$, then $\#\{C \in \mathcal{C} \mid C' \subseteq C \subseteq C_0\} = 2$ and hence, $C_0 - C' \in \mathcal{C}$ holds,

so $C' \,\dot\cup\, (C_0 - C')$ is a bipartition of C_0 as desired. If $\bigcup \mathcal{C}' = C_0$, then $\mathcal{C}'$ is obviously infinite. $\square$

We will now establish Theorem 4 without any finiteness restrictions. As above, we will establish the implications (i) $\Rightarrow$ (ii) $\Rightarrow$ (iii) $\Rightarrow$ (iv) $\Rightarrow$ (v) $\Rightarrow$ (i) and (v) $\Rightarrow$ (vi) $\Rightarrow$ (iv). Again, the implications (ii) $\Rightarrow$ (iii) and (vi) $\Rightarrow$ (iv) are trivial, the implication (i) $\Rightarrow$ (ii) follows directly from Lemma 5.13 (i), and the implication (v) $\Rightarrow$ (vi) follows directly from Lemma 5.13 (ii). The implications (iv) $\Rightarrow$ (v) and (v) $\Rightarrow$ (i) follow verbatim as in the finite case. So, it remains to establish

(iii) $\Rightarrow$ (iv):

We may infer $\bigcup \mathcal{C} = \bigcup \min^*(\mathcal{C})$ as in the finite case. Now, let $\mathcal{F} = (C_i)_{i \in I}$ denote a $\mathcal{C}$-family such that $(I_{\mathcal{F}}, \mathcal{I}_{\mathcal{F}})$ is a proper trace of $(X, \mathcal{C})$, and suppose that $\#I \geq 3$ holds. Given a cluster $C \in \mathcal{C}(\subseteq C_I)$, we know $I(C) \in \mathcal{I}_{\mathcal{F}}$ in view of Lemma 5.5; so, if there exists a cluster $C \subseteq C_I$ such that

$$I(C) \notin \emptyset^I = \{I\} \cup \tbinom{I}{1} \cup \{\emptyset\}$$

holds, we are done. To find such a cluster, choose a maximal chain $\mathcal{C}'$ from

$$\mathcal{C}^* := \{C \in \mathcal{C} \bigm| C \subseteq C_I \text{ and } I(C) = I\} \subseteq \mathcal{C} \setminus (\min^*(\mathcal{C}) \cup \{\emptyset\})$$

that must exist in view of Zorn's Lemma. First, suppose $C^* := \bigcap \mathcal{C}' \in \mathcal{C}^*$. Then, condition (iii) implies the existence of some non-empty subcluster $C \in \mathcal{C}_{\subset C^*}$ with $I(C) \notin \emptyset^I$, so we are done. Otherwise, we must have $I(C^*) \in \binom{I}{1} \cup \{\emptyset\}$, that is, there must exist some $i \in I$ with $C^* \subseteq C_i$. Choose $j \in I - \{i\}$ and, noting that $C_i \cap C \neq \emptyset$ and $C_j \cap C \neq \emptyset$ holds for all $C \in \mathcal{C}'$, define

$$\mathcal{C}'' := \{C_i \cup C \cup C_j \bigm| C \in \mathcal{C}'\} \subseteq \mathcal{C} \setminus (\min^*(\mathcal{C}) \cup \{\emptyset\}).$$

We conclude that $C_i \cup C_j = \bigcap \mathcal{C}'' \in \mathcal{C}$ and hence, $\{i, j\} \in \mathcal{I}_{\mathcal{F}}$ holds.

Finally, we establish the implications (iii) $\Rightarrow$ (ii) $\Rightarrow$ (i) $\Rightarrow$ (iii) from Theorem 5 without any finiteness restrictions.

(iii) $\Rightarrow$ (ii):

Let $\mathcal{C}_0 \subseteq \mathcal{C}$ denote a maximal sub-hierarchy of $\mathcal{C}$. We infer from Lemma 5.3 that $\max(\mathcal{C}) \cup \min^*(\mathcal{C}) \subseteq \mathcal{C}_0$ and hence, $\max(\mathcal{C}_0) = \max(\mathcal{C})$ and $\min^*(\mathcal{C}_0) = \min^*(\mathcal{C})$ holds. Since $\mathcal{C}_0$ is ample by assumption, it remains to show that $\mathcal{C}_0$ is almost finitary. Given a non-empty sub-chain $\mathcal{C}' \subseteq \mathcal{C}_0$, it is easy to see that $\mathcal{C}_0 \cup \{\bigcup \mathcal{C}', \bigcap \mathcal{C}'\}$ is a hierarchy as well; from $\{\bigcup \mathcal{C}', \bigcap \mathcal{C}'\} \subseteq \mathcal{C} \cup \{\emptyset\}$ and the fact that $\mathcal{C}_0$ is a maximal sub-hierarchy, we infer that $\{\bigcup \mathcal{C}', \bigcap \mathcal{C}'\} \subseteq \mathcal{C}_0 \cup \{\emptyset\}$ must hold, too.

(ii) $\Rightarrow$ (i):

For a non-empty cluster $C \in \mathcal{C} - \min^*(\mathcal{C})$, we put $C_0 := \bigcap \mathcal{C}_0(\supseteq C)$ and infer easily that C_0 is a non-empty member of $\mathcal{C}_0 - \min^*(\mathcal{C}_0)$. By Theorem 4: (i) $\Leftrightarrow$ (ii), there exist either two disjoint clusters $A_0, B_0 \in \mathcal{C}_0(\subset C_0)$ with $A_0 \,\dot\cup\, B_0 = C_0$, or there exists an infinite chain $\mathcal{C}_0' \subseteq \mathcal{C}_0(\subset C_0)$ with $\bigcup \mathcal{C}_0' = C_0$. In the first case, we put $A := A_0 \cap C$ and $B := B_0 \cap C$, and infer $A \,\dot\cup\, B = C$ and $A, B \in \mathcal{C}_{\subset C}$ as above. In the latter case, we put

$$\mathcal{C}' := \left\{C \cap C' \bigm| C' \in \mathcal{C}_0'\right\} \setminus \{\emptyset\}$$

and infer that $\mathcal{C}'$ is a chain with $\mathcal{C}' \subseteq \mathcal{C}_{\subset C}$ and $\bigcup \mathcal{C}' = C \cap \bigcup \mathcal{C}'_0 = C \cap C_0 = C$.

(i) $\Rightarrow$ (iii):

We want to show that condition (ii) of Theorem 4 holds for $\mathcal{C}_0$. To this end, consider a non-empty cluster $C_0 \in \mathcal{C}_0 - \min^*(\mathcal{C}_0)$. Choose $x \in C_0$ and define

$$\mathcal{C}' := \left\{ C \in \mathcal{C}_0 \;\middle|\; x \in C \subsetneqq C_0 \right\}.$$

Note that $\mathcal{C}'$ is a chain because $\mathcal{C}_0$ is a hierarchy, and that $\mathcal{C}'$ is non-empty, because Lemma 5.3 implies that $\mathcal{C}_0$ must contain $\min^*(\mathcal{C})$ and that, hence, $C_0 = \bigcup \min^*(\mathcal{C}_0(\subseteq C_0))$ must hold. If $\bigcup \mathcal{C}' = C_0$ holds, we are done. Otherwise, $C' := \bigcup \mathcal{C}'$ is a maximal element of $\mathcal{C}_0(\subset C_0)$. Now, note that $\max(\mathcal{C}_0(\subseteq C_0 - C'))$ is a partition of $C_0 - C'$ because — as in the proof of "(iii) $\Rightarrow$ (ii)" — one can show that $\mathcal{C}_0$ must be almost finitary. Altogether, this implies that

$$\mathcal{C}^* := \max(\mathcal{C}_0(\subseteq C_0 - C')) \cup \{C'\} = \max(\mathcal{C}_0(\subset C_0))$$

is a partition of C_0. As above, we can now show that C_0 is the disjoint union of two proper subclusters from $\mathcal{C}^*$.

Consequently, $\mathcal{C}_0$ satisfies condition (ii) in Theorem 4, so it must be ample. $\square$

Theorem 2 follows immediately from Theorem 7 in view of the fact mentioned above that a hierarchy $\mathcal{C} \subseteq \mathcal{P}(X)$ is a maximal sub-hierarchy of $\mathcal{P}(X)$ if and only if $\mathcal{C} = \mathcal{C}^X$ holds and $\mathcal{C}$ is ample and finitary.

Remark 5.14. Note that it is also an immediate consequence of Theorem 7 that any maximal sub-hierarchy of $\mathcal{P}(X)$ must be ample.

CHAPTER 6

Quartet Encodings and Excess

In this chapter, we analyze conditions under which a binary X-tree $T = (V, E; \phi)$ is uniquely determined by selecting, for each inner edge e of T, a corresponding "representative" quartet split Q with $Q \leq S[e]$. Do to so, we introduce a (bijective) mapping q from the inner edges $\mathring{E}$ into the set of quartet splits $\mathcal{Q} \subseteq \mathcal{Q}(X)$ under examination. One can think of other ways to represent this correspondence, e.g. the inverse map q^{-1} or a relation between inner edges and quartet splits. But we are especially interested in such subsets of quartet splits that share an important feature (i.e., they are "excess-free", see Definition 6.2) and it turns out (Lemma 6.6) that only patches can possibly posses this feature — so, when searching for excess-free subsets of $\mathring{E}$ we can limit our search to patches!

In Chapter 8, we will generalize our terminology of "excess" to sets of quartet splits or even of "labeled trees" (as announced in the introduction). But we can easily see that when dealing with *minimum* sets of quartet splits $\mathcal{Q} \subseteq \mathcal{Q}(X)$ of cardinality $\#\mathcal{Q} = \#X - 3$, this is in fact equivalent to dealing with quartet encodings — so, quartet encodings can be seen as a technical tool enabling us to state and prove Theorem 8 in a short and efficient manner.

6.1. Definitions and basic observations

In order to describe our main result (Theorem 8), we will now introduce some more terminology and preliminary results.

Definition 6.1. Let $T = (V, E; \phi)$ denote a phylogenetic X-tree. A map

$$q : \mathring{E} \to \mathcal{Q}(X)$$

is called a *quartet encoding* of T if $S[e]$ extends $q(e)$ for each $e \in \mathring{E}$. For every $e \in \mathring{E}$ and $F \subseteq \mathring{E}$, we define

$$\underline{q}(e) := \underline{q(e)} \subseteq X, \quad q(F) := \{q(e) \mid e \in F\} \subseteq \mathcal{Q}(X), \quad \text{and} \quad \underline{q}(F) := \bigcup_{e \in F} \underline{q}(e)\,.$$

We will say that q *defines* T if T is the only phylogenetic X-tree concordant with $q(\mathring{E})$; in this case, T has to be binary. A quartet encoding q is *tight* if, for each edge $e \in \mathring{E}$, there exists no other edge in $\mathring{E}$ separating the two subsets in $q(e)$.

Given a quartet encoding q of a phylogenetic X-tree $T = (V, E; \phi)$ then $\{\phi(x), \phi(y)\} \mid \{e\}$ holds for all $e \in \mathring{E}$ and both subsets $\{x, y\} \in q(e)$. If q is tight then, obviously, $\#q(\mathring{E}) = \#X - 3$ must hold. It is easy to see that a quartet encoding that defines a tree T is tight, cf. also [Ste92]. It is also easy to see that $\mathcal{T}(\mathcal{Q}) = \{T\}$ holds for some set of quartet splits $\mathcal{Q}$ with $\underline{\mathcal{Q}} = X$, $\#\mathcal{Q} = \#X - 3$, and some (necessarily binary) X-tree T if and only if there exists a quartet encoding q of T with $\mathcal{Q} = q(\mathring{E})$ that defines T.

To exploit inequality (3.2), we introduce the following definition:

DEFINITION 6.2. Given a phylogenetic X-tree $T = (V, E; \phi)$, a quartet encoding $q : \mathring{E} \to \mathcal{Q}(X)$, and a subset $F \subseteq \mathring{E}$, the (q-)*excess* of F is given by

$$\operatorname{exc}(F) = \operatorname{exc}_q(F) := \#\underline{q}(F) - \#F - 3\,. \tag{6.1}$$

We say F is (q-)*excess-free* if $\operatorname{exc}(F) = 0$ holds. We will denote the set of q-excess-free subsets of $\mathring{E}$ by $\mathcal{C}(q) \subseteq \mathcal{P}(\mathring{E})$.

Note that $\operatorname{exc}(\{e\}) = 0$ holds for every $e \in \mathring{E}$, and that $\operatorname{exc}(\emptyset) = -3$.

The next lemma will be used frequently in the following, though the proof is totally combinatorial:

LEMMA 6.1. *Suppose that q is an arbitrary quartet encoding of a phylogenetic X-tree $T = (V, E; \phi)$ and assume $F_1, F_2 \subseteq \mathring{E}$. Then,*

$$\operatorname{exc}(F_1 \cup F_2) + \operatorname{exc}(F_1 \cap F_2) \le \operatorname{exc}(F_1) + \operatorname{exc}(F_2)\,. \tag{6.2}$$

PROOF. In view of $\underline{q}(F_1 \cap F_2) \subseteq \underline{q}(F_1) \cap \underline{q}(F_2)$, we have

$$\begin{aligned}
&\operatorname{exc}(F_1 \cup F_2) + \operatorname{exc}(F_1 \cap F_2) + 6\\
&\qquad = \#\underline{q}(F_1 \cup F_2) - \#(F_1 \cup F_2) + \#\underline{q}(F_1 \cap F_2) - \#(F_1 \cap F_2)\\
&\qquad \le \#\big(\underline{q}(F_1) \cup \underline{q}(F_2)\big) + \#\big(\underline{q}(F_1) \cap \underline{q}(F_2)\big) - \big(\#(F_1 \cup F_2) + \#(F_1 \cap F_2)\big)\\
&\qquad = \#\underline{q}(F_1) + \#\underline{q}(F_2) - \#F_1 - \#F_2\\
&\qquad = \operatorname{exc}(F_1) + \operatorname{exc}(F_2) + 6\,.
\end{aligned}$$

□

More accurately, we have

$$\begin{aligned}
\operatorname{exc}(F_1 \cup F_2) + \operatorname{exc}(F_1 \cap F_2) &= \operatorname{exc}(F_1) + \operatorname{exc}(F_2)\\
&\qquad - \#\big(\underline{q}(F_1) \cap \underline{q}(F_2) - \underline{q}(F_1 \cap F_2)\big)
\end{aligned} \tag{6.3}$$

for all $F_1, F_2 \subseteq \mathring{E}$.

6.2. Examples of quartet encodings

We now present some instructive examples of tight encodings.

Examples.

14. Since $\#X = 4$ already implies that the (unique) quartet encoding of every X-tree must define that tree, the smallest non-trivial case is (without loss of generality) $X := \{1, 2, 3, 4, 5\}$. By exchanging the elements of X accordingly, we can see that the "unique" tight encoding for this ground set is defined by

$$q(e_1) := 12|35 \quad \text{and} \quad q(e_2) := 13|45$$

for the binary X-tree $T = (V, E; \phi := id_X)$ depicted in Fig. 6.1. It is easy to see that q defines T.

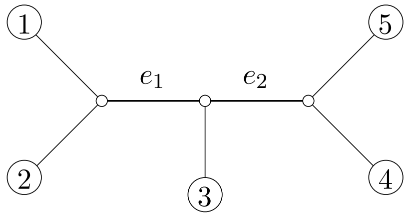


FIGURE 6.1. The "unique" binary X-tree for $X = \{1, 2, 3, 4, 5\}$.

15. For $X := \{1, \dots, 6\}$, consider the binary X-tree $T_1 = (V, E; \phi_1 := id_X)$ with $E := \{e_1, \dots, e_9\}$ having the nontrivial splits

$$S[e_1] = \{\{1,2\}, \{3,4,5,6\}\},$$
$$S[e_2] = \{\{3,4\}, \{1,2,5,6\}\},$$
$$\text{and} \quad S[e_3] = \{\{5,6\}, \{1,2,3,4\}\},$$

plus the six trivial splits as depicted in Fig. 6.2 (a). Consider the quartet encoding $q : \mathring{E} = \{e_1, e_2, e_3\} \to \mathcal{Q}(X)$ defined by

$$q(e_1) := 12|45, \quad q(e_2) := 34|16 \quad \text{and} \quad q(e_3) := 56|23. \tag{6.4}$$

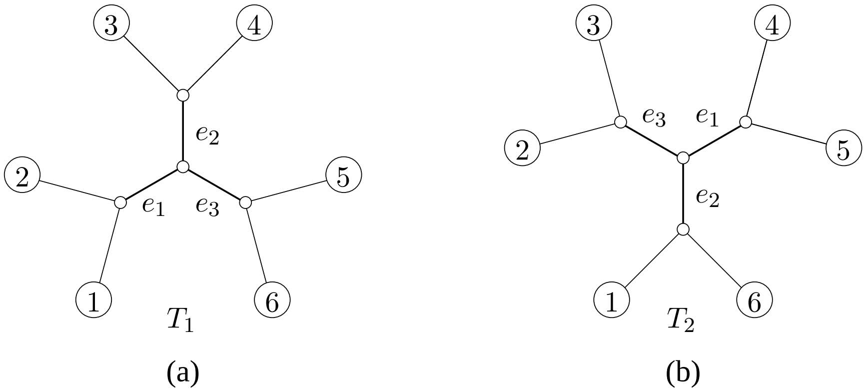


FIGURE 6.2. Two binary X-trees for $X = \{1, \dots, 6\}$ encoded by the same quartet encoding as given in (6.4).

Then q is a tight encoding of T_1, but does not define T_1, as it also encodes the X-tree T_2 depicted in Fig. 6.2 (b).

16. Suppose that $X := \{1, \dots, 7\}$, that $T = (V, E; \phi := id_X)$ is a caterpillar with seven leaves as depicted in Fig. 6.3, and that $q : \mathring{E} \to \mathcal{Q}(\{1, \dots, 7\})$ is the following quartet encoding:

$$q(e_1) := 12|36,\ q(e_2) := 13|46,\ q(e_3) := 24|57, \text{ and } q(e_4) := 25|67.$$

Then it is easy to check that q defines T.

17. Suppose q is any tight quartet encoding of a binary X-tree with

$$\bigcap_{e \in \mathring{E}} \underline{q}(e) \neq \emptyset. \tag{6.5}$$

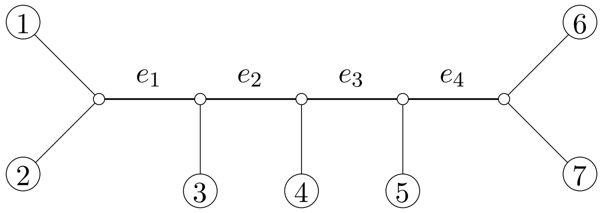


FIGURE 6.3. The caterpillar with seven leaves.

Then it has been established by Steel [Ste92] — and it follows from Theorem 8 below — that q defines T, see Lemma 8.2 on page 78 for the proof.

18. Suppose that $T = (V, E; \phi)$ is a binary X-tree such that the inner edges of T are labeled $\mathring{E} = \{e_1, \dots, e_{n-3}\}$, and that q is a tight quartet encoding of T with

$$\#\Big(\underline{q}(e_i) \setminus \bigcup_{j<i} \underline{q}(e_j)\Big) = 1 \quad \text{for} \quad i = 2, \dots, n-3\,. \tag{6.6}$$

Then it is easily established that q defines T.

6.3. Tight quartet encodings

Recalling the definitions of Section 3.4, we state

REMARK 6.2. Let q denote a quartet encoding of a phylogenetic X-tree $T = (V, E; \phi)$. Then, the following conditions are equivalent:

(i) q is tight;
(ii) $\{\phi(x), \phi(y)\} \nparallel \{e\}$ holds for every $e \in \mathring{E}$ and both sets $\{x, y\} \in q(e)$;
(iii) $[\phi(x), \phi(y)] \cap e \neq \emptyset$ holds for every $e \in \mathring{E}$ and both sets $\{x, y\} \in q(e)$;
(iv) for every edge $e \in \mathring{E}$ such that $q(e) = ab|cd$, the path from $\phi(a)$ to $\phi(b)$ (from $\phi(c)$ to $\phi(d)$, respectively) passes through exactly one vertex of e;
(v) for every edge $e \in \mathring{E}$ we have $x \not\approx_e y$ for all distinct $x, y \in \underline{q}(e)$.

LEMMA 6.3. *Suppose that q is a tight quartet encoding of a binary X-tree $T = (V, E; \phi)$. Then $\mathrm{exc}(\mathring{E}) = 0$, and for every non-empty subset $F \subseteq \mathring{E}$ one has $\mathrm{exc}(F) \geq 0$.*

The proof of this lemma is based on the following observations:

LEMMA 6.4. *Given a tight quartet encoding q of a phylogenetic X-tree $T = (V, E; \phi)$ then*

$$\#\underline{q}(\mathring{E}) \geq \#\mathring{E} + 3\,.$$

This lemma, in turn, is a direct consequence of equation (3.2) and the following lemma:

LEMMA 6.5. *Given a tight quartet encoding q of a phylogenetic X-tree $T = (V, E; \phi)$ and a non-empty subset $F \subseteq \mathring{E}$, we define $Y := \underline{q}(F)$. Then $T_{(F,Y)}$ is a* *phylogenetic Y-tree and*

$$\begin{aligned}\tilde{q} : \operatorname{Int}\big(T_{(F,Y)}\big) &\to \mathcal{Q}(Y)\ : \\ \big\{[u]_{\sim F}, [v]_{\sim F}\big\} &\mapsto q\big(\{u,v\}\big) \quad \textit{for all} \quad \{u,v\} \in F\end{aligned} \tag{6.7}$$

is a tight quartet encoding of $T_{(F,Y)}$.

If F is a patch then (using Definition 3.3) we can replace (6.7) by

$$\tilde{q} : \operatorname{Int}\big(T_{(F,Y)}\big) = F \to \mathcal{Q}(Y)\ :\ e \mapsto q(e)\,. \tag{6.8}$$

Proof. First, we have to show that $T_{(F,Y)}$ is a phylogenetic Y-tree. To do so, we show that (3.24) holds for every $\bar{v} \in V/{\sim_F}$. Let $F' := \{e \in F \mid \bar{v} \cap e \neq \emptyset\}$; for $\#F' \geq 3$ we are done. Second, suppose $F' = \{e_1, e_2\}$ for distinct edges e_1, e_2. If $\bar{v} \cap \phi(Y) = \emptyset$ then e_1, e_2 induce splits S_1, $S_2 \in \mathcal{S}(X)$ with $S_1|_Y = S_2|_Y$ in contradiction to the fact that q is tight. Third, if $F' = \{e\}$ for some edge $e = \{u,v\}$ then we may suppose without loss of generality that $\bar{v} = V(u,v)$; as q is a quartet encoding there must exist distinct x, $y \in X$ with $\{x,y\} \in q(e)$ and $\phi(x)$, $\phi(y) \in V(u,v)$.

It is easily seen that $\tilde{q}$ is well-defined and — in view of $T_{(F,Y)} \leq T|_Y$ — also a quartet encoding of $T_{(F,Y)}$. Together with Remark 6.2 it is clear that $\tilde{q}$ must be tight. ☐

Proof of Lemma 6.3. We infer from Lemma 6.4 that $\underline{q}(\mathring{E}) = X$ and, hence, $\operatorname{exc}(\mathring{E}) = 0$ must hold.

Using the notation of Lemma 6.5 and applying Lemma 6.4 we can calculate

$$\#\underline{q}(F) = \#\underline{\tilde{q}}\big(\operatorname{Int}(T_{(F,Y)})\big) \geq \#\operatorname{Int}\big(T_{(F,Y)}\big) + 3 = \#F + 3$$

and, hence, $\operatorname{exc}(F) = \#\underline{q}(F) - \#F - 3 \geq 0$. ☐

The following lemma shows that it suffices for a subset $F \subseteq \mathring{E}$ to be a q-patch that F is excess-free, in case of a tight encoding q:

Lemma 6.6. *Suppose that q is a tight quartet encoding of a binary X-tree T. If $F \subseteq \operatorname{Int}(T)$ is excess-free then F is a T-patch.*

Proof. Let $F \subseteq \mathring{E}$ denote a subset that is not a patch. Using the notation of Lemma 6.5 it is sufficient to show

$$\#Y = \#\underline{\tilde{q}}\big(\operatorname{Int}(T_{(F,Y)})\big) > \#\operatorname{Int}(T_{(F,Y)}) + 3 = \#F + 3$$

or, equivalently, the existence of a vertex $\bar{v} \in V_{(F,Y)}$ of degree greater than three. To this end, let $v \in V$ denote a vertex with associated equivalence class $\bar{v} := [v]_{\sim F}$ such that $\#(\bar{v} \cap \bigcup F) \geq 2$ holds — such a vertex must exist, because F is no patch. Choose distinct inner vertices v_1, $v_2 \in (\bar{v} \cap \bigcup F)$, then $[v_1, v_2]_T \subseteq \bar{v}$ and, as a consequence, $\langle v_1, v_2\rangle_T \cap F = \emptyset$. We define u_j to be the unique vertex in $[v_1, v_2]_T$ incident with v_j for $j = 1, 2$, that is, $\{v_1, u_1\}$, $\{v_2, u_2\} \in \langle v_1, v_2\rangle_T$. Let $U_j := \bar{v} \cap V(u_j, v_j)$ for $j = 1, 2$. We want to show

$$\#\{e \in F \mid \bar{v} \cap e \neq \emptyset\} + \#\big(\bar{v} \cap \phi(Y)\big) \geq 4$$

and in view of

$$\#\{e \in F \mid \bar{v} \cap e \neq \emptyset\} + \#\big(\bar{v} \cap \phi(Y)\big) \geq \sum_{j=1,2} \Big(\#\{e \in F \mid U_j \cap e \neq \emptyset\} + \#\big(U_j \cap \phi(Y)\big)\Big)$$

it is enough to establish

$$\#\{e \in F \mid U_j \cap e \neq \emptyset\} + \#\big(U_j \cap \phi(Y)\big) \geq 2 \tag{6.9}$$

for $j = 1, 2$. So suppose that $\#\{e \in F \mid U_j \cap e \neq \emptyset\} < 2$ holds; because of $v_j \in U_j \cap \bigcup F$ there exists a unique edge $f_j \in F$ with $U_j \cap f_j \neq \emptyset$. Then $U_j - \{v_j\}$ is an equivalence class of $V/{\approx_{f_j}}$, so $U_j \cap \phi(Y) \neq \emptyset$ follows from Remark 6.2 and the fact that q is tight, so (6.9) holds. $\square$

By Lemmata 6.1 and 6.3, the map $\mathrm{exc} : \mathcal{P}(\mathring{E}) \to \mathbb{Z}$ satisfies the conditions of Example 7 on page 42 and we infer:

Lemma 6.7. *If q is a tight quartet encoding of a binary X-tree $T = (V, E; \phi)$, then $\mathcal{C}(q)$, the set of excess-free subsets of $\mathring{E}$, forms a patchwork.*

6.4. Induced encodings and q-patches

In the following, let q denote a tight quartet encoding of a phylogenetic X-tree $T = (V, E; \phi)$, and $F \subseteq \mathring{E}$ a patch. We have established in Section 3.4 that $T_F = (V_F, E_F; \phi_F)$ is a phylogenetic X_F-tree with $\mathrm{Int}(T_F) = F \subseteq E_F$. For the sake of brevity we set $\bar{x} := [x]_{\approx F}$ for all $x \in X$. Now, we can define a map

$$q_F : F \to \mathcal{Q}(X_F) \ : \ e \mapsto \bar{x}\bar{y}|\bar{w}\bar{z} \quad \text{for} \quad q(e) = xy|wz \tag{6.10}$$

that is well-defined because q is tight — this follows immediately from Remark 6.2.

It is easy to check that q_F is a tight quartet encoding of T_F: For an arbitrary edge $e \in F$ we choose x, y, w, $z \in X$ such that $q(e) = xy|wz$; removing e from T separates $\phi(x)$, $\phi(y)$ from $\phi(w)$, $\phi(z)$, so removing e from T_F separates $\bar{x}$, $\bar{y}$ from $\bar{w}$, $\bar{z}$ as well. Furthermore,

$$\big[\bar{x}, \bar{y}\big]_{T_F} \cap F = \big[\phi(x), \phi(y)\big]_T \cap F$$

for all x, $y \in X$ implies that q_F is tight.

Using this observation and Lemma 6.5 we have established:

Lemma 6.8. *Given a tight quartet encoding q of a binary X-tree T and a patch $F \subseteq \mathrm{Int}(T)$, then*

$$\underline{q}(F)/{\approx_F} = X/{\approx_F}\,. \tag{6.11}$$

Corollary 6.9. *Given a tight quartet encoding q of a binary X-tree T, a patch $F \subseteq \mathrm{Int}(T)$, and $x \in X$ with $x \notin \underline{q}(F)$ then $\#[x]_{\approx F} \geq 2$ must hold.*

The following lemma proofs the utility of these concepts:

Lemma 6.10. *If a quartet encoding q defines an X-tree T, then q_F defines T_F for every patch $F \subseteq \mathrm{Int}(T)$.*

PROOF. Lemma 3.11 states that T_F must be a binary X_F-tree because $T = (V, E; \phi)$ is binary. We suppose that q_F does not define T_F, so there exists a second phylogenetic X_F-tree $T'_F = (V'_F, E'_F; \phi'_F)$ non-isomorphic to T_F that is concordant with $q_F(F)$. In the following, we will construct an X-tree T' non-isomorphic to T that is concordant with $q(\mathring{E})$ in contradiction to the assumption that q defines T.

Let $\bar{x} \in X_F$ denote an arbitrary equivalence class. Firstly, there exists an edge $e_{\bar{x}}$ of T such that $\mathrm{edge}(y \to F) = e_{\bar{x}}$ holds for all $y \in \bar{x}$. Choose $u_{\bar{x}}, v_{\bar{x}} \in V$ such that $e_{\bar{x}} = \{u_{\bar{x}}, v_{\bar{x}}\}$ and $v_{\bar{x}} \in \bigcup F$ holds, and note that $\{\bar{x}, v_{\bar{x}}\}$ is an edge of T_F.

Second, there exists a vertex $w_{\bar{x}}$ of T'_F such that $\{\phi'_F(\bar{x}), w_{\bar{x}}\}$ is an edge of T'_F by analogy.

Third, we define $U_{\bar{x}} := \big[\phi(x)\big]_{\approx F}$ to be a subset of vertices of T; the set is well defined in view of $\phi(y) \approx_F \phi(x)$ for all $y \in \bar{x}$. Clearly, $U_{\bar{x}}$ contains $\phi(\bar{x}) = \{\phi(y) \mid y \in \bar{x}\}$, and $u_{\bar{x}} \in U_{\bar{x}}$, $v_{\bar{x}} \notin U_{\bar{x}}$ holds. We define $E_{\bar{x}} := \binom{U_{\bar{x}}}{2} \cap E$.

Now, we can define an X-tree $T' = (V', E'; \phi')$ by:

$$\begin{aligned} V' &:= \bigcup \mathrm{Int}(T'_F) \cup\!\!\!\!\cdot\;\; \bigcup_{\bar{x} \in X_F} U_{\bar{x}} \\ E' &:= \mathrm{Int}(T'_F) \cup\!\!\!\!\cdot\;\; \bigcup_{\bar{x} \in X_F} E_{\bar{x}} \;\cup\!\!\!\!\cdot\;\; \big\{\{u_{\bar{x}}, w_{\bar{x}}\} \bigm| \bar{x} \in X_F\big\} \\ \phi' &: X \to V' \;:\; x \mapsto \phi(x) \end{aligned}$$

It is easy to check that T' is in fact a phylogenetic X-tree.

Obviously, T' is concordant with $q(F)$, and since every inner edge e of T' that is not an element of $\mathrm{Int}(T'_F)$, is also an edge of T and induces the same split in T and T' by construction of T', we conclude that T' is concordant with $q(\mathring{E})$. As $(T')_F \cong T'_F \ncong T_F$ implies $T' \ncong T$ we are done. ☐

Now, we will deal with the special case that the patch $F \subseteq \mathring{E}$ is excess-free:

DEFINITION 6.3. Given a quartet encoding q of a phylogenetic X-tree $T = (V, E; \phi)$ and a subset $F \subseteq \mathring{E}$, then F is called a *q-patch* if F is a T-patch and $\mathrm{exc}_q(F) = 0$ holds; F is called a *q-superpatch* if F and $\mathring{E} - F$ are q-patches.

Note that a superpatch F that is also a q-patch is not necessarily a q-superpatch, because $\mathrm{exc}_q(\mathring{E} - F) \neq 0$ might hold. For a q-superpatch $F \subseteq \mathring{E}$ we infer from (6.3) that

$$\#\big(\underline{q}(F) \cap \underline{q}(\mathring{E} - F)\big) = 3 \tag{6.12}$$

holds because $F \cap (\mathring{E} - F) = \emptyset$.

LEMMA 6.11. *Given a tight quartet encoding q of a binary X-tree T and a patch $F \subseteq \mathrm{Int}(T)$, then F is q-excess-free if and only if $Y := \underline{q}(F)$ is a* minimal *set with $Y/{\approx_F} = X/{\approx_F}$.*

COROLLARY 6.12. *Suppose that q is a tight quartet encoding of a binary X-tree T. If $F \subseteq \mathrm{Int}(T)$ is excess-free (and therefore a patch), then $T_{(F,Y)} \cong T|_Y$ holds for $Y := \underline{q}(F)$.*

Lemma 6.11 is an easy consequence of the formula

$$\operatorname{exc}_q(F) = \sum_{\bar{x} \in X/\approx_F} \Big(\#\big(\bar{x} \cap \underline{q}(F)\big) - 1 \Big) \tag{6.13}$$

for a tight quartet encoding q of a binary X-tree T and a patch $F \subseteq \operatorname{Int}(T)$; where $\#(\bar{x} \cap \underline{q}(F)) \geq 1$ holds for all $\bar{x} \in X/\approx_F$ in view of Lemma 6.8. To show that equation (6.13) holds we calculate

$$\begin{aligned}
\sum_{\bar{x} \in X/\approx_F} \Big(\#\big(\bar{x} \cap \underline{q}(F)\big) - 1 \Big) &= \#\Big(\bigcup_{\bar{x} \in X/\approx_F} \big(\bar{x} \cap \underline{q}(F)\big) \Big) - \#(X/\approx_F) \\
&= \#\big(X \cap \underline{q}(F)\big) - (\#F + 3) \\
&= \operatorname{exc}_q(F)
\end{aligned}$$

where $\#(X/\approx_F) = \#F + 3$ follows from the fact that T_F is a binary $(X/\approx_F)$-tree with inner edges $\operatorname{Int}(T_F) = F$, and equation (3.2). In case F is a superpatch we have seen in Remark 3.10 that there exist exactly one or two equivalence classes of cardinality greater one, so there are at most two summands in (6.13) not equal to zero. The next lemma states that the "outermost" edges of a q-patch already determine the "behavior" of $\underline{q}(F)$:

Lemma 6.13. *Given a tight quartet encoding q of a binary X-tree $T = (V, E; \phi)$ and a q-patch $F \subseteq \mathring{E}$ we choose $v \in V$, $x \in X$ such that $v = \operatorname{gate}(\phi(x) \to F)$. If $f \in F$ is an edge with $v \in f$ then $[x]_{\approx F} = [x]_{\approx f}$ and*

$$\underline{q}(F) \cap [x]_{\approx F} = \underline{q}(f) \cap [x]_{\approx F}\,.$$

Proof. In view of $v \in f$ and $\operatorname{edge}(\phi(x) \to F) \notin F$ we infer

$$\phi([x]_{\approx F}) = [\phi(x)]_{\approx \{v\}} \cap \phi(X) = \phi([x]_{\approx f})$$

what implies the first part of the lemma. Applying Lemma 6.11 twice we get

$$\#\big(\underline{q}(F) \cap [x]_{\approx F}\big) = 1 = \#\big(\underline{q}(f) \cap [x]_{\approx f}\big)$$

and together with $\underline{q}(f) \subseteq \underline{q}(F)$ the claim of the lemma follows. $\square$

Let q denote a tight quartet encoding of a phylogenetic X-tree T. If $F \subseteq \mathring{E}$ is a patch, then Lemma 6.5 tells us that $q|_F$ is a tight quartet encoding of $T_{(F,Y)}$ for $Y := \underline{q}(F)$. More interesting, in case F is also excess-free, we can modify Lemma 6.10 and get:

Lemma 6.14. *Given a quartet encoding q that defines an X-tree T, and a q-patch $F \subseteq \operatorname{Int}(T)$. Let $Y := \underline{q}(F)$, then $q|_F$ defines $T_{(F,Y)}$ and $T_{(F,Y)} \cong T|_Y$.*

CHAPTER 7

The Main Theorem

It is tempting to conjecture that if q defines a binary X-tree T, then there always exists an edge $e \in \mathring{E}$ such that $\mathring{E} - \{e\}$ is q-excess-free. In some cases (eg. for $\#X \leq 6$ and for encodings constructed as in Example 18 on page 60), this is indeed the case, but in general it is not (Example 16 on page 59 provides a counterexample with seven leaves). Nevertheless, we can still hope that the excess-free subsets of $\mathring{E}$ form at least an ample patchwork, and this turns out to be indeed the case, as we state now as part of our main result:

THEOREM 8. *Given a quartet encoding q of a binary X-tree $T = (V, E; \phi)$, then the following two statements are equivalent:*

(i) *q defines T;*
(ii) *q is tight, and the patchwork $\mathcal{C}(q)$ of q-excess-free subsets of $\mathring{E}$ is ample.*

The remaining part of this chapter consists of a formal proof of this theorem. While it is relatively easy to show that (ii) implies (i) (see Section 7.1) following the train of thought established in the preceding chapters, the implication (i) $\Rightarrow$ (ii), except for the fact that q must be tight, is far from trivial and will be established in Sections 7.2 to 7.5.

7.1. Condition (ii) implies condition (i)

We start with the "trivial" part of the proof. We consider a binary X-tree $T = (V, E; \phi)$ and a tight quartet encoding $q : \mathring{E} \to \mathcal{Q}(X)$ such that $\mathcal{C}(q)$ is an ample patchwork. For $\#X = 4$ an encoding of an X-tree always defines that tree; we proceed by induction and suppose $\#X \geq 5$ in the following. We assume that for every tight quartet encoding q' of an X'-tree T' with $\#X' < \#X$ such that $\mathcal{C}(q')$ forms an ample patchwork, we have established already that q' defines T'. We want to show that q defines T, too.

Let $F_1, F_2 \subseteq \mathring{E}$ denote two disjoint and q-excess-free subsets with $F_1 \cupdot F_2 = \mathring{E}$; such subsets must exist since $\mathcal{C}(q)$ is ample. We define $X_j := \underline{q}(F_j)$ and $T_j := T_{(F_j, X_j)}$ for $j = 1, 2$; we have seen in Lemma 6.5 and Corollary 6.12 that

$$q_j := q|_{F_j} \;:\; \mathrm{Int}(T_j) = F_j \to \mathcal{Q}(X_j)$$

is a tight quartet encoding of $T_j \cong T|_{X_j}$ for $j = 1, 2$. Furthermore,

$$\mathcal{C}(q_j) = \{F \subseteq F_j \bigm| \mathrm{exc}_{q_j}(F) = 0\} = \{F \subseteq F_j \bigm| \mathrm{exc}_q(F) = 0\} = \mathcal{C}(q)_{\subseteq F_j}$$

holds, so $C(q_j)$ is ample for $j = 1, 2$; we conclude that q_j defines T_j for $j = 1, 2$ by induction hypothesis. Let $Y := X_1 \cap X_2$, then (6.12) implies

$$\#Y = \#\big(\underline{q}(F_1) \cap \underline{q}(F_2)\big) = 3\,. \tag{7.1}$$

Obviously $T' := T|_Y = T_1\big|_Y = T_2\big|_Y \cong K_{1,3}$ holds. Applying Lemma 6.11 to the q-patches F_1, F_2 we infer that $x \not\approx_{F_1} y$ and $x \not\approx_{F_2} y$ holds for all distinct $x, y \in Y$, and Corollary 2.16 states that

$$v_{F_1} = \operatorname{med}(\phi(x), \phi(y), \phi(z)) = v_{F_2}$$

must hold for $\{x, y, z\} = Y$; with Remark 3.3 we conclude

$$V(T') = \{\phi(x), \phi(y), \phi(z), \tilde{v}\}$$

for $\tilde{v} := v_{F_1}$. We want to apply the second part of the Tree Joining Theorem to guaranty that T is the *unique* X-tree satisfying $T_1 \leq T$ and $T_2 \leq T$: Then, the existence of an X-tree T^* non-isomorphic to T but concordant with $q(\mathring{E})$ would imply that $T^*|_{X_j}$ is concordant with $q_j(F_j)$ for $j = 1, 2$, while $T^*|_{X_j} \not\cong T_j$ holds for at least one $j \in \{1, 2\}$ in contradiction to the fact that q_j defines T_j.

It is trivial that conditions (i) and (ii) of the Tree Joining Theorem are satisfied. Finally, condition (iii) must hold because q is tight: If there exists an edge $e' = \{\tilde{v}, u\}$ of T' such that

$$\#\{e \in E(T_j) \bigm| S_{T'}(e') \leq S_{T_j}(e)\} \geq 2$$

for $j = 1, 2$, then we define $\tilde{e} := \operatorname{edge}_T(u \to \{\tilde{v}\})$ to be the unique edge of the path connecting $\tilde{v}$ and u in T incident with $\tilde{v}$. Suppose without loss of generality that $\tilde{e} \in F_1$ holds, and that $\phi(x) = u$. We can easily see $[\phi(y), \phi(z)]_T \cap \tilde{e} = \{\tilde{v}\}$, so $[\phi(y), \phi(z)]_T \cap \bigcup F_2 = \emptyset$ or, equivalently, $y \approx_{F_2} z$ holds in contradiction to above.

7.2. Condition (i) implies condition (ii)

We consider a binary X-tree $T = (V, E; \phi)$ such that $\#X \geq 6$ holds, and a quartet encoding $q : \mathring{E} \to \mathcal{Q}(X)$ that defines T (and is therefore tight). We assume that for every X'-tree T' defined by a quartet encoding q' with $\#X' < \#X$ we have established already that $\mathcal{C}(q')$ forms an ample patchwork. We want to show in the following that this holds for $\mathcal{C}(q)$, too.

To show this, it is enough to prove the existence of a q-superpatch $F \subseteq \mathring{E}$: In this case, we can define $F_1 := F$ and $F_2 := \mathring{E} - F$, and again $\#(X_1 \cap X_2) = 3$ holds in view of (7.1), where $X_j := \underline{q}(F_j)$ for $j = 1, 2$. Now, $q_j := q|_{F_j}$ defines $T_j := T_{(F_j, X_j)} \cong T|_{X_j}$ for $j = 1, 2$ as seen in Lemma 6.14. So, $\mathcal{C}(q_1) = \mathcal{C}(q)_{\subseteq F_1}$ and $\mathcal{C}(q_2) = \mathcal{C}(q)_{\subseteq F_2}$ must be ample by induction hypothesis, and

$$\mathcal{C}(q)_{\subseteq F_1} \cup \mathcal{C}(q)_{\subseteq F_2} \cup \{\mathring{E}\} \;\subseteq\; \mathcal{C}(q)$$

already implies that $\mathcal{C}(q)$ is ample, too — where $\mathring{E} \in \mathcal{C}(q)$ holds by Lemma 6.3, of course: It follows from Theorem 2 that $\mathcal{C}(q_j) \cup \{\emptyset\}$ contains a maximal F_j-hierarchy $\mathcal{C}'_j$ for $j = 1, 2$; then $\mathcal{C}'_1 \cup \mathcal{C}'_2 \cup \{\mathring{E}\}$ is a maximal $\mathring{E}$-hierarchy in view of $\mathring{E} = F_1 \uplus F_2$, and invoking Theorem 2 again we conclude from $\mathcal{C}'_1 \cup \mathcal{C}'_2 \cup \{\mathring{E}\} \subseteq \mathcal{C}(q) \cup \{\emptyset\}$ that $\mathcal{C}(q)$ must be ample.

We will use the induction hypothesis in particular in the case where F is a patch of *co-cardinality* one (that is, $\#(\mathring{E} - F) = 1$), and in this special case we also know that the set

$$A_F := \{x \in X \bigm| \#[x]_{\approx_F} \geq 2\}$$

has cardinality two; so, the unique edge $e_F = \mathrm{edge}(\phi(a)\to F) \in \mathring{E} - F$ for $a \in A$ induces the split $\{A_F, X - A_F\}$. Note that $d_T(\phi(a), \phi(a')) = 2$ holds for $A_F = \{a, a'\}$.[1] Let $v_F = \mathrm{gate}(\phi(a)\to F)$ denote the touching point of F and define $u_F := \mathrm{gate}(\phi(a)\to\{e_F\})$ for $a \in A_F$, then $e_F = \{u_F, v_F\}$ holds.

We will derive some basic results for the special case of a patch F of co-cardinality one: A q_F-patch $F' \subseteq F$ is a q-patch if and only if $A_F \not\subseteq \underline{q}(F')$ holds, while we have $\mathrm{exc}_q(F') = 1$ in case $A_F \subseteq \underline{q}(F')$. In general, we have

$$\mathrm{exc}_q(F') = \mathrm{exc}_{q_F}(F') + \delta\big(2, \#\big(\underline{q}(F') \cap A_F\big)\big) \tag{7.2}$$

for an arbitrary subset $F' \subseteq F$. Now, suppose that $q(e_F) = \{A_F, B_F\}$ holds for some set $B_F \in \binom{X - A_F}{2}$. Recall that (6.3) implies

$$\mathrm{exc}(F_1 \cup F_2) = \mathrm{exc}(F_1) + \mathrm{exc}(F_2) + \#(F_1 \cap F_2) - \#\big(\underline{q}(F_1) \cap \underline{q}(F_2)\big) + 3 \tag{7.3}$$

for all $F_1, F_2 \subseteq \mathring{E}$. We infer

$$\begin{aligned}
\mathrm{exc}_q(F' \cup \{e_F\}) &= \mathrm{exc}_q(F') + 3 - \#\big(\underline{q}(F') \cap \underline{q}(e_F)\big) \\
&= \mathrm{exc}_q(F') \ + 1 - \#\big(\underline{q}(F') \cap A_F\big) \ + 2 - \#\big(\underline{q}(F') \cap B_F\big) \\
&= \mathrm{exc}_{q_F}(F') + \begin{cases} 1 & \text{for } \underline{q}(F') \cap A_F = \emptyset, \\ 0 & \text{otherwise} \end{cases} + 2 - \#\big(\underline{q}(F') \cap B_F\big) \qquad (7.4)
\end{aligned}$$

for every subset $F' \subseteq F$; so,

$$\mathrm{exc}_q(F' \cup \{e_F\}) = \mathrm{exc}_{q_F}(F')$$

holds for $F' \subseteq F$ if and only if $\underline{q}(F') \cap A_F \neq \emptyset$ and $B_F \subseteq \underline{q}(F')$. In particular, $F' \cup \{e_F\}$ is a q-patch if and only if F' is a q_F-patch with $\underline{q}(F') \cap A_F \neq \emptyset$ and $B_F \subseteq \underline{q}(F')$. Finally, if $F' \subseteq F$ is a q_F-patch such that $F' \cup \{e_F\}$ is a T-superpatch then

$$\mathrm{exc}_q(F' \cup \{e_F\}) \leq 1 \tag{7.5}$$

must hold: Applying Lemma 6.8 to the quartet encoding q_F we infer with $[A_F]_{\approx F'} = A_F$ that $A_F \in \underline{q_F}(F')$ and, hence, $\underline{q}(F') \cap A_F \neq \emptyset$ holds; while $\underline{q}(F') \cap B_F = \emptyset$ implies $\#[b]_{\approx F'} \geq 2$ for both $b \in B_F$ and hence, $b \approx_{\{v_F\}} b'$ for $B_F = \{b, b'\}$ in contradiction to the fact that q is tight, see Remark 6.2. Note that (7.5) holds in particular if F' is a q_F-superpatch with $v_F \in \bigcup F'$.

We will establish in Sections 7.3 to 7.5 that the following three claims hold:

1. If there exist pairwise distinct patches $F_1, \ldots, F_k \subseteq \mathring{E}$ of co-cardinality one ($k \geq 2$) such that $F_j \cap F_{j+1}$ is a q_{F_j}-patch for $j = 1, \ldots, k$ (identifying indices modulo k), then one of these patches is a q-patch.
2. Every superpatch $F \subseteq \mathring{E}$ with
$$\#F \geq 2, \quad \mathrm{exc}(F) = 0, \quad \text{and} \quad \mathrm{exc}(\mathring{E} - F) = 1$$
is contained in a q-superpatch $F' \subseteq \mathring{E}$.
3. Every superpatch $F_2 \subseteq \mathring{E}$ that (i) is contained in a patch $F \subseteq \mathring{E}$ of co-cardinality one, (ii) forms a q_F-superpatch relative to the induced encoding q_F of T_F, (iii) satisfies $v_F \notin \bigcup F_2$ for the touching point v_F of F, and (iv) is *minimal* with respect to conditions (i) to (iii), is actually a q-patch.

[1] Such elements are called *twins* [BDS99] or a *cherry* [ESSW97b] in the literature.

How do these three claims imply that condition (i) of the theorem must hold? Choose an arbitrary patch $F \subseteq \mathring{E}$ of co-cardinality one. By Lemma 6.10, the induced encoding q_F defines the X_F-tree T_F and, in view of $\#X_F = \#X - 1$, the induction hypothesis guarantees the existence of a q_F-superpatch $F_2 \subseteq F$. By eventually exchanging F_2 with $F - F_2$, we can ensure without loss of generality:

(a) $v_F \in \bigcup(F - F_2)$;
(b) F_2 is chosen minimal; and
(c) $v_F \in \bigcup F_2$ implies $\#F_2 \leq \#(F - F_2)$.

Now suppose that $v_F \notin \bigcup F_2$ holds. We deduce from Claim 3 that $\mathrm{exc}(F_2) = 0$ must hold; furthermore, $F - F_2$ is a q_F-superpatch with $v_F \in \bigcup(F - F_2)$, so we can apply equation (7.5) for $F' := F - F_2$ and infer $\mathrm{exc}_q(F_1) \leq 1$ for $F_1 := \mathring{E} - F_2 = (F - F_2) \cup \{e_F\}$. In case $\#F_2 \geq 2$ we apply Claim 2 to secure the existence of a q-superpatch containing F_2, so $\mathcal{C}(q)$ is in fact an ample patchwork and we are done. To this end, we may suppose in the following that, whenever we are given a patch $F \subseteq \mathring{E}$ of co-cardinality one, then there exists an associated q_F-patch $F_2 \subseteq F$ satisfying (a)–(c) as well as

$$v_F \in \bigcup F_2 \quad \text{or} \quad \#F_2 = 1 \quad \text{(or both)};$$

in particular, $v_F \notin \bigcup F_2$ implies $\#F_2 = 1$.

We will now show that there exists at least one such tuple (F, F_2) satisfying $\#F_2 = 1$. To do so, we consider the "inner tree" $\mathring{T} = (\bigcup \mathring{E}, \mathring{E})$ that is a subtree of (V, E). We distinguish two cases: If there exists a pending edge $e \in \mathring{E} - \mathrm{Int}(\mathring{T})$ that is incident with a vertex of degree two in $\mathring{T}$, then we define $F := \mathring{E} - \{e\}$ and easily check that $v_F \notin \bigcup F'$ must hold for *every* subset $F' \subseteq F$ satisfying $v_F \in \bigcup(F - F')$; in particular, $\#F_2 = 1$ holds for the associated q_F-patch $F_2 \subseteq F$. Otherwise, every pending edge $e \in \mathring{E} - \mathrm{Int}(\mathring{T})$ is incident with a vertex of degree three in $\mathring{T}$; this implies the existence of distinct edges $e_1, e_2 \in \mathring{E} - \mathrm{Int}(\mathring{T})$ such that $e_1 \cap e_2 \neq \emptyset$. Let $F := \mathring{E} - \{e_1\}$; it is easy to check that $v_F \notin \bigcup F'$ holds for *every* proper subset $F' \subsetneqq F - \{e_2\}$. So, the associated q-patch F_2 equals either $\{e_2\}$ or $F - \{e_2\}$; in the latter case, we derive $\#F_2 \leq \#(F - F_2) = 1$.

Let $F \subseteq \mathring{E}$ denote a patch of co-cardinality one with associated q_F-superpatch $F_2 \subseteq F$ of cardinality one. Now, $F' := \mathring{E} - F_2$ is also a patch of co-cardinality one, so there exists a $q_{F'}$-superpatch $F_2' \subseteq F'$ satisfying

$$v_{F'} \in \bigcup F_2' \quad \text{or} \quad \#F_2' = 1 \quad \text{(or both)}.$$

In case $\#F_2' = 1$ holds we can repeat this, constructing a $q_{F''}$-superpatch $F_2'' \subseteq F''$ for $F'' := \mathring{E} - F_2'$, and so on. Since there is only a finite number of T-patches, this either leads to a circle of pairwise distinct patches $F_1, \ldots, F_k$ of co-cardinality one ($k \geq 2$) such that $F_j \setminus F_{j+1}$ and, hence, $F_j \cap F_{j+1}$ is a q_{F_j}-patch for $j = 1, \ldots, k \pmod{k}$; or there exist two distinct patches $\tilde{F}$, $F \subseteq \mathring{E}$ of co-cardinality one such that $\tilde{F} \cap F$ is a $q_{\tilde{F}}$-superpatch, together with the associated q_F-superpatch $F_2 \subseteq F$ satisfying (a)–(c) as well as $v_F \in \bigcup F_2$ and $\#F_2 \geq 2$.

In the first case, we apply Claim 1 to show that one of these patches is actually a q-patch, so $\mathcal{C}(q)$ is ample and we are done. In the second case, we first note that

$$v_F \in \bigcup F_2 \cap \bigcup(F - F_2)\,, \tag{7.6}$$

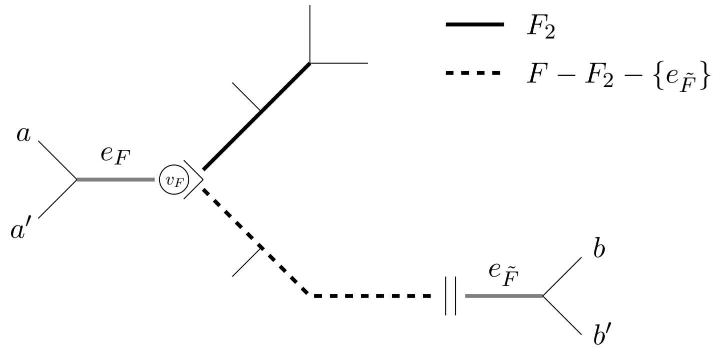


FIGURE 7.1. All Possibilities for the actual positions of $\tilde{F}$, F, F_2, and $A_F = \{a, a'\}$ relative to each other.

and (c) implies $\#(F - F_2) \geq \#F_2 \geq 2$; so, by eventually dropping condition (c) and exchanging F_2 with $\mathring{E} - F_2$, we may suppose for the rest of this paragraph that $e_{\tilde{F}} \in F - F_2$ holds, and the q_F-superpatch F_2 still satisfies $\#F_2 \geq 2$ and (7.6). Since $\tilde{F} \cap F$ is a $q_{\tilde{F}}$-superpatch and $e_{\tilde{F}} \notin \tilde{F} \cap F$ we get $\#(\underline{q}(\tilde{F} \cap F) \cap A_F) \leq 1$ from (6.13); in view of $F_2 \subseteq \tilde{F} \cap F$ we then also have $A_F \not\subseteq \underline{q}(F_2)$, what implies that F_2 is a q-patch as we have seen above. Furthermore, $F - F_2$ is a q_F-superpatch with $v_F \in \bigcup(F - F_2)$, so we can apply equation (7.5) for $F' := F - F_2$ and infer $\mathrm{exc}_q(F_1) \leq 1$ for $F_1 := \mathring{E} - F_2 = (F - F_2) \cup \{e_F\}$. Because $\#F_2 \geq 2$ we can apply Claim 2 and get a q-superpatch containing F_2, so $\mathcal{C}(q)$ is ample and, hence, condition (ii) of the theorem is satisfied. In Fig. 7.1 we have depicted the possible positions of $\tilde{F}$, F, and F_2 relative to each other.

It remains to show that Claims 1 to 3 hold.

7.3. First claim

If there exist pairwise distinct patches $F_1, \ldots, F_k \subseteq \mathring{E}$ of co-cardinality one ($k \geq 2$) such that $F_j \cap F_{j+1}$ is a q_{F_j}-patch for $j = 1, \ldots, k$ (identifying indices modulo k) then one of these patches is a q-patch.

We define $A_j := A_{F_j}$, $B_j := B_{F_j}$, and $e_j := e_{F_j}$ for $j = 1, \ldots, k$. For every $j = 1, \ldots, k \pmod{k}$ it is easy to check that $F_j \cap F_{j+1} = \mathring{E} - \{e_j, e_{j+1}\}$ is a T-patch but, in general, no T-superpatch; that A_j and A_{j+1} are the only two elements in $X/{\approx_{F_j \cap F_{j+1}}}$ of cardinality greater than one; that in view of Lemma 6.11 and the fact that $F_j \cap F_{j+1}$ is a q_{F_j}-patch we can deduce the existence of an element $a_{j+1} \in A_{j+1}$ such that

$$\underline{q}(F_j \cap F_{j+1}) \cup A_j = X - \{a_{j+1}\} \tag{7.7}$$

holds.

If $F_{j+1} = (F_j \cap F_{j+1}) \cup \{e_j\}$ is a q-patch, we are done. Otherwise, we know $\underline{q}(F_j \cap F_{j+1}) \cap A_j \neq \emptyset$ because q is tight, so $B_j \not\subseteq \underline{q}(F_j \cap F_{j+1})$ must hold since $F_j \cap F_{j+1}$ *is* a q_{F_j}-patch. We derive $a_{j+1} \in B_j$ from (7.7) for $j = 1, \ldots, k \pmod{k}$. Furthermore, in view of

$$a_{j+1} \notin \underline{q}(F_j \cap F_{j+1}) \quad \text{and} \quad \{e_1, \ldots, e_{j-1}, e_{j+2}, \ldots, e_k\} \subseteq F_j \cap F_{j+1} \tag{7.8}$$

we conclude that

$$B_j \cap \{a_1, \ldots, a_k\} = \{a_{j+1}\} \tag{7.9}$$

must hold for $j = 1, \ldots, k \pmod k$ and, hence, there exist $z_j \in X - \{a_1, \ldots, a_k\}$ with $B_j = \{a_{j+1}, z_j\}$.

In the following, we will construct a second X-tree T' non-isomorphic to T that is concordant with $q(\mathring{E})$ in contradiction to the fact that q defines T: We cut off, for every $j = 1, \ldots, k \pmod k$, the edge incident with $\phi(a_{j+1})$ and implant it instead into the edge incident with $\phi(z_j)$ as depicted in Fig. 7.2 below.

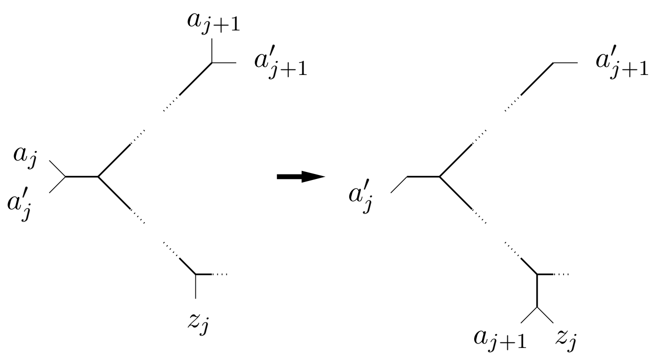


FIGURE 7.2. Cutting off and re-implanting edges.

If $z_j = z_{j+1}$, we have to make sure that the edge leading to $\phi(a_{j+2})$ gets implanted closer to $\phi(z_j)$ than the edge leading to $\phi(a_{j+1})$ — no further special care needs to be taken (and because we cannot have $z_1 = z_2 = \cdots = z_k$ for a tight encoding q, this requirement can always be fulfilled).

Formally, we define

$$\tilde{F} := F_1 \cap F_2 \cap \cdots \cap F_k \subseteq \mathring{E} \quad \text{and} \quad Y := X - \{a_1, \ldots, a_k\}$$

and infer $\underline{q}(\tilde{F}) = Y$ from (7.8) and Lemma 6.3, so $\tilde{F}$ is q-excess-free and $T_{(\tilde{F},Y)} \cong T|_Y$ is concordant with $q(\tilde{F})$ by Lemma 6.14. Suppose without loss of generality that $z_1 \neq z_k$ holds. We define $T_0 = (V_0, E_0; \phi_0) := T_{(\tilde{F},Y)}$ and

$$\begin{aligned} Y_j &:= Y \mathbin{\dot\cup} \{a_{i+1} \mid i = 1, \ldots, j\} \quad \text{for} \quad j = 0, \ldots, k-1, \quad \text{and} \\ Y_k &:= Y \mathbin{\dot\cup} \{a_1, \ldots, a_k\} = X\,; \end{aligned}$$

note that T_0 is a Y_0-tree with $\phi_0 = id_{Y_0}$, and $z_0 := z_k = \phi_0(z_k)$ is a leaf of T_0. In the following, we will iteratively define Y_j-trees $T_j = (V_j, E_j; \phi_j)$ for $j = 1, \ldots, k$ until we end up with an X-tree T_k concordant with $q(\mathring{E})$.

We repeat the following construction step for $j = 0, \ldots, k-1$: Suppose that $z_j = \phi_j(z_j)$ is a leaf of T_j; let v denote the inner vertex of T_j such that $\{v, z_j\}$ is an edge of T_j. We choose $w \notin V_j \cup X$ and define:

$$\begin{aligned} V_{j+1} &:= V_j \mathbin{\dot\cup} \{w, a_{j+1}\} \\ E_{j+1} &:= E_j - \big\{\{v, z_j\}\big\} \mathbin{\dot\cup} \big\{\{v, w\}, \{w, z_j\}, \{w, a_{j+1}\}\big\} \\ \phi_{j+1} &:= id|_{Y_{j+1}} \end{aligned}$$

It is easy to see that $T_{j+1} = (V_{j+1}, E_{j+1}; \phi_{j+1})$ is in fact a Y_{j+1}-tree with $T_{j+1}\big|_{Y_j} = T_j$, and that $z_{j+1} = \phi_{j+1}(z_{j+1})$ is a leaf of T_{j+1}. We denote the inner edge $\{v, w\}$ of T_{j+1} by f_{j+1} and note that $f_1, \ldots, f_j$ are inner edges of T_{j+1}, too.

Finally, we define the X-tree $T' := T_k$ and claim that $S_{T'}[f_j]$ extends $q(e_j)$ for all $j \in \{1, \ldots, k\}$: This follows from the fact that removing f_{j+1} from T' separates $\{a_{j+1}, z_j\}$ from $\{a_j\} \cup Y - \{z_j\}$ for $j = 1, \ldots, k-1$; and removing f_1 separates $\{a_1, z_k\}$ from $\{a_k\} \cup Y - \{z_k\}$ in view of $z_1 \neq z_k$. Since $T_0 \leq T'$ is concordant with $q(\tilde{F})$ we are done. $\square$

7.4. Second claim

Every superpatch $F \subseteq \mathring{E}$ with $\#F \geq 2$, $\mathrm{exc}(F) = 0$, and $\mathrm{exc}(\mathring{E} - F) = 1$ is contained in a q-superpatch $F' \subseteq \mathring{E}$.

We define the superpatch $F_2 := \mathring{E} - F$, and $Y := \underline{q}(F) \cap \underline{q}(F_2)$. Using (7.3), we calculate

$$\begin{aligned} \#Y &= \#\big(\underline{q}(F) \cap \underline{q}(F_2)\big) \\ &= \mathrm{exc}(F) + \mathrm{exc}(F_2) - \mathrm{exc}(\mathring{E}) + 3 \\ &= 0 + 1 - 0 + 3 \;=\; 4\,. \end{aligned}$$

In view of (6.13), we conclude that there exists exactly one equivalence class $\bar{x} \in X/{\approx_{F_2}}$ such that $\#(\bar{x} \cap \underline{q}(F_2)) \geq 2$ holds, where the inequality is indeed an equality. We set $A := \bar{x} \cap \underline{q}(F_2)$ for this $\bar{x}$; applying Remark 3.10, we get $\#[a]_{\approx F} = 1$ for both $a \in A$ and, hence, $A \subseteq \underline{q}(F)$ by Lemma 6.8. Let $B := Y - A$ with $\#B = 2$ and note that, in view of $\mathrm{exc}(F) = 0$, a similar argument implies

$$\big\{x \in \underline{q}(F) \;\big|\; \#(\underline{q}(F) \cap [x]_{\approx F}) \geq 2\big\} \subseteq B\,.$$

Let $\tilde{e} \in \mathring{E}$ denote the edge with $\tilde{e} = \mathrm{edge}(\phi(a) \to F_2)$ for both $a \in A$, set $v := \mathrm{gate}(\phi(a) \to F_2) = v_F$, and choose $u \in V$ such that $\tilde{e} = \{u, v\}$ holds. In Fig. 7.3, we have depicted the possible positions of F, F_2, $\tilde{e}$, and A, B relative to each other.

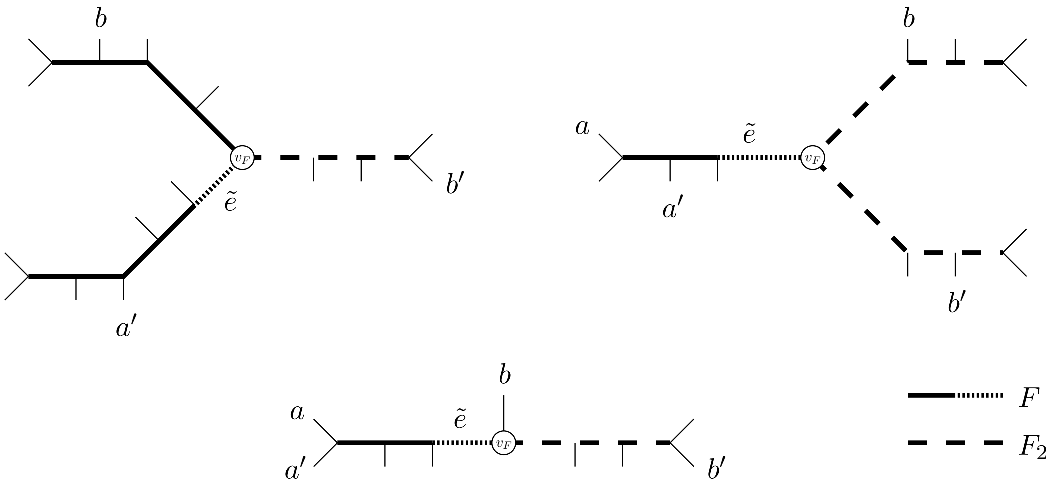


FIGURE 7.3. All three possibilities for the actual positions of F, F_2, $\tilde{e}$, $A = \{a, a'\}$, and $B = \{b, b'\}$ relative to each other.

We define $\tilde{F} := F_2 \cup \{\tilde{e}\}$, $X_2 := \underline{q}(F_2)$, and $T_2 := T_{(\tilde{F},X_2)}$. To see that T_2 is in fact an X_2-tree we only have to check that $\deg_{T_2}(u) = 3$ holds what can be inferred from

$$[u]_{\sim_{\tilde{F}}} \cap \phi(X_2) = \phi(A) \quad \text{and} \quad \#\phi(A) = \#A = 2$$

because $\phi(a) \sim_{\tilde{F}} u \sim_{\tilde{F}} \phi(a')$ is equivalent to $a \approx_{F_2} a'$ for $A = \{a, a'\}$. Now, we define a map

$$\tilde{q} : \mathrm{Int}(T_2) = \tilde{F} \to \mathcal{Q}(X_2) \ : \ e \mapsto \begin{cases} q(e) & \text{for } e \in F_2, \\ \{A, B\} & \text{for } e = \tilde{e} \end{cases} \tag{7.10}$$

and it is easy to check that $\tilde{q}$ is a quartet encoding of T_2, since $\tilde{e}$ separates A from $X_2 - A$ by definition of A, $\tilde{e}$.

We claim that $\tilde{q}$ defines T_2: To this end, we first recall that $T|_Y$ is isomorphic to the binary Y-tree $T_{\{A,B\}}$ inducing the unique non-trivial split $\{A, B\}$; formally, we define

$$T_{\{A,B\}} := \Big(A \cup B \cup \{A, B\}, \big\{\{a, A\}, \{B, b\}, \{A, B\} \bigm| a \in A, b \in B\big\}; id_{A\cup B}\Big). \tag{7.11}$$

Suppose that there exists an X_2-tree T_2' that is concordant with $\tilde{q}(\tilde{F})$ but non-isomorphic to T_2. Obviously $T_2'|_Y \cong T_{\{A,B\}}$ must hold in view of $\tilde{q}(\tilde{e}) = \{A, B\} \in \tilde{q}(\tilde{F})$. We define the binary X_1-tree $T_1 := T_{(F,X_1)} \cong T|_{X_1}$ for $X_1 := \underline{q}(F)$ (see Corollary 6.12) then $T_1|_Y \cong T|_Y \cong T_{\{A,B\}}$ holds as well, so we can use the Tree Joining Theorem to construct an X-tree T' with $T'|_{X_1} \cong T_1$ and $T'|_{X_2} \cong T_2'$ that is obviously not isomorphic to T. As T_1 is concordant with $q(F)$ and T_2' is concordant with $q(F_2)$ we conclude that T' is concordant with $q(\mathring{E})$ in contradiction to the fact that q defines T.

So, $\tilde{q}$ defines T_2 and in view of

$$\#X_2 = \#\tilde{F} + 3 = \#F_2 + 4 = \#\mathring{E} - \#F + 4 \leq \#X - 3 - 2 + 4 = \#X - 1$$

the induction hypothesis secures the existence of $\tilde{q}$-patches $\tilde{F}_1$, $\tilde{F}_2 \subseteq \tilde{F}$ with $\tilde{F}_1 \cup \tilde{F}_2 = \tilde{F}$. We may suppose without loss of generality that $\tilde{e} \in \tilde{F}_1$ holds; then $\tilde{q}(\tilde{F}_2) = q(\tilde{F}_2)$ implies $\mathrm{exc}_q(\tilde{F}_2) = \mathrm{exc}_{\tilde{q}}(\tilde{F}_2) = 0$. Finally, it is easy to see that $F' := F \cup (\tilde{F}_1 - \{\tilde{e}\}) = \mathring{E} - \tilde{F}_2$ is a T-patch and, together with $\underline{q}(F) \cap \underline{\tilde{q}}(\tilde{F}_1) = Y$ and $\underline{q}(F) \cup \underline{\tilde{q}}(\tilde{F}_1) = \underline{q}(F')$, we infer that

$$\begin{aligned} \mathrm{exc}_q(F') &= \#\underline{q}(F') - \#F' - 3 \\ &= \#\underline{q}(F) + \#\underline{\tilde{q}}(\tilde{F}_1) - \#Y - (\#F + \#\tilde{F}_1 - 1) - 3 \\ &= \#\underline{q}(F) - \#F - 3 + \#\underline{\tilde{q}}(\tilde{F}_1) - \#\tilde{F}_1 - 3 \\ &= \mathrm{exc}_q(F) + \mathrm{exc}_{\tilde{q}}(\tilde{F}_1) \\ &= 0 + 0 \end{aligned}$$

holds as claimed. □

7.5. Third claim

Every superpatch $F_2 \subseteq \mathring{E}$ that (i) is contained in a patch $F \subseteq \mathring{E}$ of co-cardinality one, (ii) forms a q_F-superpatch relative to the induced encoding q_F of T_F, (iii) satisfies $v_F \notin \bigcup F_2$ for the touching point v_F of F, and (iv) is *minimal* with respect to conditions (i) to (iii), is actually a q-patch.

Assume that $F \subseteq \mathring{E}$ is a patch of co-cardinality one and that a minimal q_F-superpatch $F_2 \subseteq F$ with $v_F \notin \bigcup F_2$ has been chosen. Put $F_1 := F - F_2$ and define $A := A_F$ and $B := B_F$ with $q(e_F) = \{A, B\}$. Let u_F, u_1, $u_2 \in V$ denote three distinct vertices such that $\{v_F, u_F\} = e_F$, $\{v_F, u_2\} = \mathrm{edge}(v_{F_2} \to \{v_F\})$, and $\{v_F, u_1\} \in \mathring{E}$. If $A \not\subseteq \underline{q}(F_2)$ then $\mathrm{exc}_q(F_2) = 0$ holds in view of (7.2) and we are done, so suppose $A \subseteq \underline{q}(F_2)$ in the following. In view of $v_F \in \bigcup F_2$ we derive $\underline{q}(F_1) \cap A \neq \emptyset$ and, as a consequence,

$$\mathrm{exc}_q(F_i \cup \{e_F\}) = 2 - \big(\#\underline{q}(F_i) \cap B\big) \tag{7.12}$$

for $i = 1, 2$, confer (7.4).

We infer $\mathrm{exc}_q(F_2 \cup \{e_F\}) \geq 1$ from Lemma 6.6 and $\mathrm{exc}_q(F_1 \cup \{e_F\}) \geq 1$ from Claim 2 and the fact that F_2 was chosen minimal: Otherwise, there exists a q-superpatch $F' \subseteq \mathring{E}$ containing $F_1 \cup \{e_F\}$ and, since F_2 is no q-patch, $\mathring{E} - F'$ is a q_F-patch properly contained in F_2. Now, (7.12) implies the existence of b_1, $b_2 \in B$ such that $b_1 \notin \underline{q}(F_2)$ and $b_2 \notin \underline{q}(F_1)$ holds, and together with $B \subseteq X - A \subseteq \underline{q}(F) = \underline{q}(F_1) \cup \underline{q}(F_2)$ (by Lemma 6.8 and Remark 3.10) we conclude $B = \{b_1, b_2\}$, and

$$b_1 \in \underline{q}(F_1) \setminus \underline{q}(F_2) \quad \text{and} \quad b_2 \in \underline{q}(F_2) \setminus \underline{q}(F_1)\,. \tag{7.13}$$

Next, consider the set $\underline{q}(F_1) \cap (\underline{q}(F_2) - A)$. There must exists some set $C \in \binom{X-A}{2}$ with $C = \underline{q}(F_1) \cap (\underline{q}(F_2) - A)$ because $A = [a]_{\approx F} \in \underline{q_F}(F_1) \cap \underline{q_F}(F_2)$ and $\#(\underline{q_F}(F_1) \cap \underline{q_F}(F_2)) = 3$ implies

$$\underline{q_F}(F_1) \cap \underline{q_F}(F_2) = \big\{A, \{c\}, \{c'\}\big\} \tag{7.14}$$

for $C = \{c, c'\}$, see (6.12). Moreover, we clearly have $B \cap C = \emptyset$. Altogether, we infer

$$\underline{q}(F_1) \cap \underline{q}(F_2) = A \cup C \quad \text{and} \quad \underline{q}(F_j) \cap B = \{b_j\} \quad \text{for} \quad j = 1, 2\,. \tag{7.15}$$

Set $\tilde{e} := \mathrm{edge}(v_F \to F_2) \in F_1$ and $\tilde{v} := \mathrm{gate}(v_F \to F_2) = v_{F_2}$, and let $\tilde{u} \in V$ denote the vertex with $\tilde{e} = \{\tilde{u}, \tilde{v}\}$. Applying Lemma 6.11 to equation (7.14), we get

$$\underline{q}(F_2) \cap \phi^{-1}(V(\tilde{v}, \tilde{u})) = A \tag{7.16}$$

and, with $b_2 \notin A$, we derive $\phi(b_2) \in V(\tilde{u}, \tilde{v}) \subseteq V(v_F, u_2)$. As q is tight, we must have $\phi(b_1) \in V(v_F, u_1)$; and with $C \cap A = \emptyset$, we also have $\phi(C) \subseteq V(\tilde{u}, \tilde{v})$. Altogether, we conclude that e_F is the *unique* edge of T separating A from $\{b_1\} \cup C$, while $\tilde{e}$ separates $A \cup \{b_1\}$ from C. In Fig. 7.4, we have depicted the possible positions of F_1, F_2, e_F, $\tilde{e}$, A, B, and C relative to each other.

Analogously to Claim 2, we define $\tilde{F} := F_2 \cup \{\tilde{e}\}$, $\tilde{X} := \underline{q}(F_2)$, and $\tilde{T} := T_{(\tilde{F}, \tilde{X})}$. Again, $\tilde{T}$ is an $\tilde{X}$-tree because $\deg_{\tilde{T}}(\tilde{u}) = 3$ holds in view of

$$[\tilde{u}]_{\sim \tilde{F}} \cap \phi(\tilde{X}) = \phi(A) \quad \text{and} \quad \#\phi(A) = \#A = 2\,.$$

We define the map

$$\tilde{q} : \mathrm{Int}(\tilde{T}) = \tilde{F} \to \mathcal{Q}(\tilde{X}) \ : \ e \mapsto \begin{cases} q(e) & \text{for } e \in F_2, \\ \{A, C\} & \text{for } e = \tilde{e} \end{cases} \tag{7.17}$$

that is a quartet encoding of $\tilde{T}$ because $\tilde{e}$ induces the split $\{A, \tilde{X} - A\}$ in $\tilde{T}$ and $C \subseteq \underline{q}(F_2) - A = \tilde{X} - A$.

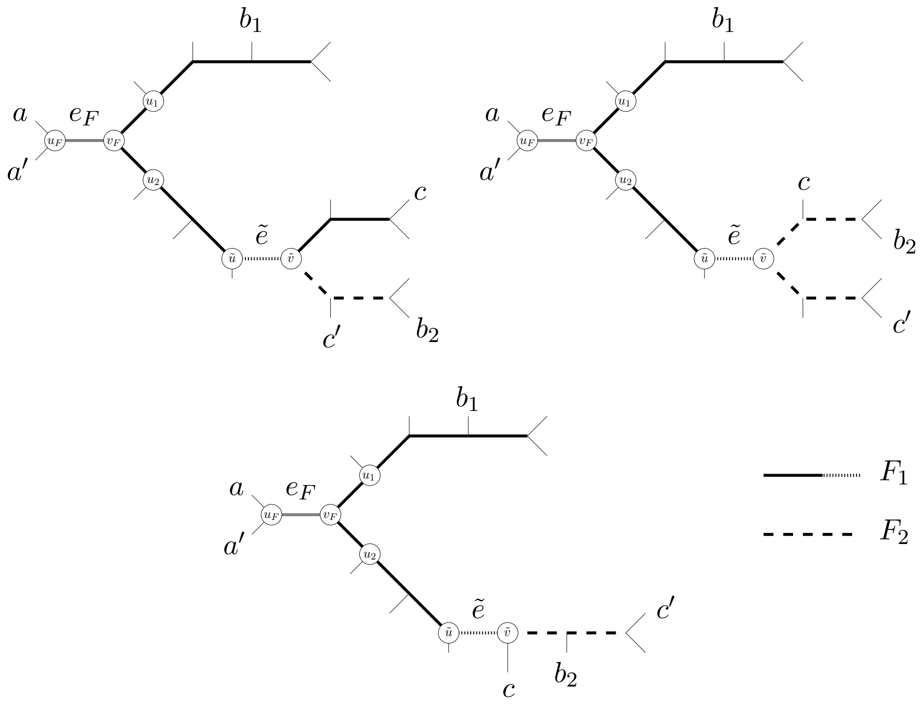


FIGURE 7.4. All three possibilities for the actual positions of F_1, F_2, e_F, $\tilde{e}$, $A = \{a, a'\}$, b_1, b_2, and $C = \{c, c'\}$ relative to each other. Note that $u_2 = \tilde{u}$, or even $v_F = \tilde{u}$ and $u_2 = \tilde{v}$ might hold.

We claim that $\tilde{q}$ defines $\tilde{T}$: Suppose that there exists a binary $\tilde{X}$-tree $\hat{T}$ that is concordant with $\tilde{q}(\tilde{F})$ but non-isomorphic to $\tilde{T}$. Again, we will construct an X-tree T' non-isomorphic to T but concordant with $q(\mathring{E})$. Unfortunately, this gets a little more complicated than in Claim 2.

We start by defining the X_1-tree $T_1 := T_{(F_1, X_1)}$ for $X_1 := \underline{q}(F_1) \cup A$, where $q|_{F_1}$ is a quartet encoding of T_1 (confer Lemma 6.5). Note that T_1 is not binary, and because of $e_F \notin F_1$ we conclude that there is no edge in T_1 separating A from $\{b_1\} \cup C$. So the induced tree $T_1\big|_{A \cup \{b_1\} \cup C}$ (as depicted in Fig. 7.5) is not binary but has exactly one inner edge, and this edge is separating $A \cup \{b_1\}$ from C in view of $\tilde{e} \in F_1$.

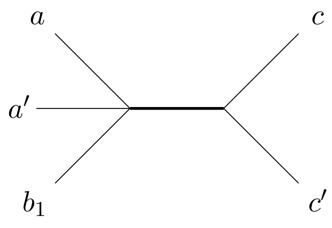


FIGURE 7.5. The tree $T_1\big|_{A \cup \{b_1\} \cup C}$.

Second, we define $X_2 := A \cup \{b_2\} \cup C$ and $T_2 := \hat{T}\big|_{X_2}$; the three possible tree structures of T_2 for $A = \{a, a'\}$ and $C = \{c, c'\}$ are depicted in Fig. 7.6. Note that T_2 must be concordant with $\big\{\{A, C\}\big\}$ in view of $\{A, C\} \in \tilde{q}(\tilde{F})$.

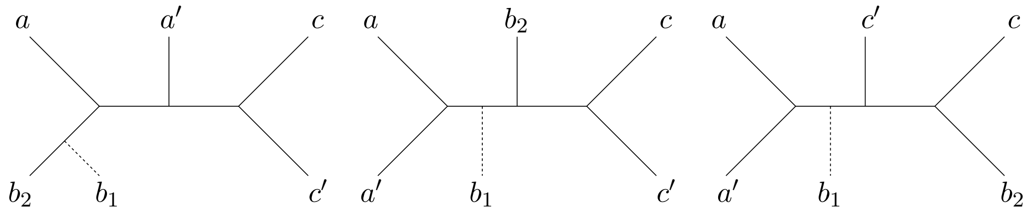


FIGURE 7.6. The three possible tree structures for T_2 (without dotted lines) and T_3 (including dotted lines).

Put $X_3 := A \cup B \cup C = X_2 \uplus \{b_1\}$. Clearly, there exists a X_3-tree T_3 with (a) $T_3\big|_{X_2} \cong T_2$ and (b) T_3 is concordant with $\{aa'b_1|cc', aa'|b_1b_2\}$ for $A = \{a, a'\}$ and $C = \{c, c'\}$ (recall that $x_1x_2x_3|y_1y_2$ denotes the split $\{\{x_1, x_2, x_3\}, \{y_1, y_2\}\}$): If $ab_2|a'cc' \in \mathcal{S}[T_2]$ holds for distinct a, $a' \in A$, then we insert a pending edge leading to b_1 to the pending edge of T_2 leading to b_2. Otherwise we insert the pending edge leading to b_1 to the edge of T_2 inducing the split $aa'|b_2cc'$. It is straightforward to check that the resulting tree satisfies (b). The placement of b_1 and the resulting tree T_3 are also depicted in Fig. 7.6.

Now, we can find a X_4-tree T_4 for $X_4 := \tilde{X} \cup X_3 = \tilde{X} \uplus \{b_1\}$ satisfying $T_4\big|_{X_3} \cong T_3$ and $T_4\big|_{\tilde{X}} \cong \hat{T}$ by invoking the Tree Joining Theorem, because $Y_4 := \tilde{X} \cap X_3 = X_2$ and $T_3\big|_{Y_4} \cong T_2 = \hat{T}\big|_{Y_4}$ holds. Note that there exists an edge in T_4 separating $A \cup \{b_1\}$ from C, and that $T_4 \not\cong T|_{X_4}$ holds in view of $T_4\big|_{\tilde{X}} \cong \hat{T} \not\cong T|_{\tilde{X}}$.

We use the Tree Joining Theorem for a last time to find an X-tree T' with

$$T_1 \leq T'|_{X_1} \quad \text{and} \quad T_4 \cong T'|_{X_4}$$

where the conditions of the theorem are fulfilled in view of $X = X_1 \cup X_4$ and

$$T_1\big|_{Y_5} \leq T_3\big|_{Y_5} \cong T_4\big|_{Y_5}$$

for $Y_5 := X_1 \cap X_4 = A \cup \{b_1\} \cup C$ by construction of T_3. We claim that this X-tree T' (that is obviously not isomorphic to T) is concordant with $q(\mathring{E})$: The tree $T_1 \leq T'$ is concordant with $q(F_1)$, the tree $\hat{T} \leq T_4 \leq T'$ is concordant with $q(F_2)$ and, finally, the tree $T_3 \leq T_4 \leq T'$ is concordant with $\{q(e_F)\} = \{\{A, B\}\}$ by construction. But this is a contradiction to the fact that q defines T and, for this reason, no such tree $\hat{T}$ can exist.

So, $\tilde{q}$ defines $\tilde{T}$ and in view of

$$\#\tilde{X} = \#\tilde{F} + 3 = \#F_2 + 4 \leq \#F + 3 = \#\mathring{E} + 2 = \#X - 1$$

the induction hypothesis secures the existence of $\tilde{q}$-patches $\tilde{F}_1$, $\tilde{F}_2 \subseteq \tilde{F}$ with $\tilde{F}_1 \uplus \tilde{F}_2 = \tilde{F}$ and, say, $\tilde{e} \in \tilde{F}_1$. We will show that $\tilde{F}_2$ is actually a q-patch and $F_1 \cup \tilde{F}_1$ is (at least) a q_F-patch. Then $\tilde{F}_2 \subseteq F_2$ and $v_F \notin \bigcup \tilde{F}_2 \subseteq \bigcup F_2$ implies that the T-superpatch $\tilde{F}_2$ satisfies conditions (i) to (iii) of Claim 3. But (iv) states that F_2 was chosen minimal with respect to these conditions, hence $\tilde{F}_2 = F_2$ holds in contradiction to our assumption $\mathrm{exc}_q(F_2) > 0$.

Here, $\mathrm{exc}_q(\tilde{F}_2) = 0$ is a simple consequence of $\tilde{q}(\tilde{F}_2) = q(\tilde{F}_2)$ since $\tilde{e} \notin \tilde{F}_2$; using

$$\underline{q}(F_1) \cap \underline{q}(F_2) = A \cup C = \underline{\tilde{q}}(\tilde{e}), \quad q|_{\tilde{F}-\{\tilde{e}\}} = \tilde{q}|_{\tilde{F}-\{\tilde{e}\}}, \quad \text{and} \quad q(\tilde{F}_2) \subseteq q(F_2)$$

we infer that

$$\begin{aligned}\underline{q}(F_1 \cup \tilde{F}_1) \cap \underline{q}(\tilde{F}_2) &= \Big(\underline{q}(F_1) \cap \underline{q}(F_2) \cap \underline{q}(\tilde{F}_2)\Big) \cup \Big(\underline{q}(\tilde{F}_1 - \{\tilde{e}\}) \cap \underline{q}(\tilde{F}_2)\Big) \\ &= \Big(\underline{\tilde{q}}(\tilde{e}) \cap \underline{\tilde{q}}(\tilde{F}_2)\Big) \cup \Big(\underline{\tilde{q}}(\tilde{F}_1 - \{\tilde{e}\}) \cap \underline{\tilde{q}}(\tilde{F}_2)\Big) \\ &= \underline{\tilde{q}}(\tilde{F}_1) \cap \underline{\tilde{q}}(\tilde{F}_2)\end{aligned}$$

so we can calculate (using (7.3) and (6.12))

$$\begin{aligned}\mathrm{exc}_{q_F}(F_1 \cup \tilde{F}_1) &= \mathrm{exc}_{q_F}(F_1 \cup \tilde{F}_1 \cup \tilde{F}_2) - \mathrm{exc}_{q_F}(\tilde{F}_2) - \#\Big((F_1 \cup \tilde{F}_1) \cap \tilde{F}_2\Big) \\ &\quad + \#\Big(\underline{q_F}(F_1 \cup \tilde{F}_1) \cap \underline{q_F}(\tilde{F}_2)\Big) - 3 \\ &\leq 0 - 0 - 0 + \#\Big(\underline{q}(F_1 \cup \tilde{F}_1) \cap \underline{q}(\tilde{F}_2)\Big) - 3 \\ &= \#\Big(\underline{\tilde{q}}(\tilde{F}_1) \cap \underline{\tilde{q}}(\tilde{F}_2)\Big) - 3 \\ &= 0\,.\end{aligned}$$

This establishes Claim 3 and, hence, the theorem. ∎

CHAPTER 8

Implications and Applications

In this chapter, we will establish useful results and algorithms when reconstructing trees from quartet splits (Sections 8.1, 8.4) and collections of trees (Sections 8.2, 8.3).

8.1. Reconstructing trees from quartet splits

In order to reconstruct trees from (sets of) quartet splits, we probably want an algorithm that inputs a set of quartet splits $\mathcal{Q} \subseteq \mathcal{Q}(X)$, and outputs the (set of) trees concordant with $\mathcal{Q}$. Although this problem is NP-complete in general (as mentioned in the introduction), Theorem 8 will enable us to find such an algorithm for certain instances of the problem.

In Theorem 8 we used quartet encodings that were useful when proving the theorem, but are somehow clumsy in the framework described above: To apply the theorem, we must already know the "unique" supertree, and then define an appropriate quartet encoding. Luckily, there exists a straightforward generalization of the definitions of "tight" and "excess" given in Chapter 6 to sets of quartet splits (and, in fact, even sets of trees, see Section 8.2):

A phylogenetic X-tree $T = (V, E; \phi)$ is called *tightly concordant* with a set of partial splits $\mathcal{S} \subseteq \mathcal{S}_{\text{part}}(X)$ if, for every inner edge $e \in \mathring{E}$, there exists at least one partial split $S \in \mathcal{S}$ such that e is the *unique* inner edge of T with $S \leq S_T[e]$. So, T is tightly concordant with $\mathcal{S}$ if and only if T is concordant with $\mathcal{S}$, but $T_{(F,X)}$ is not concordant with $\mathcal{S}$ for every proper subset $F \subsetneq \mathring{E}$.

For a set of quartet splits $\mathcal{Q}$ we define the *excess* $\mathcal{Q}$, denoted $\text{exc}(\mathcal{Q})$, by

$$\text{exc}(\mathcal{Q}) := \#\underline{\mathcal{Q}} - 3 - \#\mathcal{Q}$$

and we call $\mathcal{Q}$ *excess-free* if $\text{exc}(\mathcal{Q}) = 0$ holds. We denote the set of excess-free subsets of $\mathcal{Q}$ by $\mathcal{C}(\mathcal{Q}) \subseteq \mathcal{P}(\mathcal{Q})$.

Clearly, given an X-tree $T = (V, E; \phi)$ and a set of quartet splits $\mathcal{Q}$ with $\underline{\mathcal{Q}} = X$ of cardinality $\#\mathcal{Q} = \#X - 3$, then T is (tightly) concordant with $\mathcal{Q}$ if and only if there exist a (tight) quartet encoding q of T with $q(\mathring{E}) = \mathcal{Q}$, see also Section 6.1; if so, then

$$\text{exc}(q(F)) = \text{exc}_q(F) \quad \text{holds for all subsets} \quad F \subseteq \mathring{E}$$

what, in particular, implies $\mathcal{C}(\mathcal{Q}) = \mathcal{C}(q)$.

Now, we can restate Theorem 8 without using quartet encodings; we use Corollary 5.6 to derive the following theorem that is easily seen to be equivalent to Theorem 8:

THEOREM 9. *Given a set of quartet splits $\mathcal{Q}$ with $\underline{\mathcal{Q}} = X$ of cardinality $\#\mathcal{Q} = \#X - 3$ and a binary X-tree $T = (V, E; \phi)$, then T is the unique X-tree concordant* *with $\mathcal{Q}$ if and only T is tightly concordant with $\mathcal{Q}$, and one of the following conditions holds:*

(i) *the patchwork $\mathcal{C}(\mathcal{Q})$ of excess-free subsets of $\mathcal{Q}$ is ample;*
(ii) $\mathcal{C}(\mathcal{Q}) \cup \{\emptyset\}$ *contains a maximal hierarchy* $\mathcal{C}_0$*;*
(iii) *every maximal sub-hierarchy $\mathcal{C}_0$ of $\mathcal{C}(\mathcal{Q})$ is ample;*
(iv) *for every excess-free set $\mathcal{Q}_0 \subseteq \mathcal{Q}$ of cardinality $\#\mathcal{Q}_0 \geq 2$, there exist disjoint and excess-free subsets $\mathcal{Q}_1$, $\mathcal{Q}_2 \subseteq \mathcal{Q}_0$ with $\mathcal{Q}_1 \uplus \mathcal{Q}_2 = \mathcal{Q}_0$;*
(v) *for every excess-free set $\mathcal{Q}_0 \subseteq \mathcal{Q}$ of cardinality $\#\mathcal{Q}_0 \geq 2$, there exist proper excess-free subsets $\mathcal{Q}_1$, $\mathcal{Q}_2 \subset \mathcal{Q}_0$ with $\mathcal{Q}_1 \cup \mathcal{Q}_2 = \mathcal{Q}_0$;*
(vi) *for every proper trace $(I, \mathcal{I})$ of $(X, \mathcal{C}(\mathcal{Q}))$ with $\mathcal{I} \subseteq \emptyset^I = \{\emptyset\} \cup \binom{I}{1} \cup \{I\}$, we have $\#I = 2$;*
(vii) *every proper trace $(I, \mathcal{I})$ of $(X, \mathcal{C}(\mathcal{Q}))$ is ample;*
(viii) *for every partition $\mathcal{C}_0 \subseteq \mathcal{C}(\mathcal{Q})$ with $\bigcup \mathcal{C}_0 \in \mathcal{C}(\mathcal{Q})$ and $\#\mathcal{C}_0 \geq 2$, there exist distinct $\mathcal{Q}_1$, $\mathcal{Q}_2 \in \mathcal{C}_0$ such that $\mathcal{Q}_1 \uplus \mathcal{Q}_2$ is excess-free.*

Note that conditions (i)–(viii) are *not* equivalent in general; but under the assumption that T is tightly concordant with $\mathcal{Q}$, this is indeed the case. The "non-trivial" part of the theorem can be used to deduce a necessary condition applicable without knowing the tree T:

COROLLARY 8.1. *Given a set of quartet splits $\mathcal{Q} \subseteq \mathcal{Q}(X)$ (with $\underline{\mathcal{Q}} = X$) of cardinality $\#\mathcal{Q} = \#X - 3$ such that there is a unique X-tree concordant with $\mathcal{Q}$. Then, the patchwork $\mathcal{C}(\mathcal{Q})$ of excess-free subsets of $\mathcal{Q}$ must be ample (or, equivalently, one (and therefore all) of the conditions* (ii)–(viii) *of Theorem 9 must hold).*

It is important to stress that Theorem 8 does *not* yield an answer to the question "Given a collection of quartet splits $\mathcal{Q}$ with $\underline{\mathcal{Q}} = X$ of cardinality $\#\mathcal{Q} = \#X - 3$, is there (at least one) X-tree T concordant with $\mathcal{Q}$?" in general. Using Theorem 8, we can decide whether there exists at least one X-tree T concordant with $\mathcal{Q}$ only in case T is the *unique* X-tree concordant with $\mathcal{Q}$! So, violation of the necessary condition presented in Corollary 8.1 just tells us $\#\mathcal{T}(\mathcal{Q}) \neq 1$.

We announced in Chapter 6 that Example 17 on page 59 is a simple consequence of our main result:

LEMMA 8.2. *Suppose $\mathcal{Q}$ is a set of quartet splits with $\underline{\mathcal{Q}} = X$ of cardinality $\#\mathcal{Q} = \#X - 3$ such that $\bigcap_{Q \in \mathcal{Q}} \underline{Q} \neq \emptyset$ holds. If there exists a binary X-tree T tightly concordant with $\mathcal{Q}$, then T is the unique tree concordant with $\mathcal{Q}$.*

PROOF. Let $q : \mathring{E} \to \mathcal{Q}$ denote the associated tight quartet encoding of $T = (V, E; \phi)$. Assume $x \in \bigcap_{Q \in \mathcal{Q}} \underline{Q}$ and, for $v \in V$, put

$$\begin{aligned} V(v) &:= \{u \in V \mid v \in [\phi(x), u]_T\}, \\ \mathring{E}_1(v) &:= \{e \in \mathring{E} \mid e \subseteq V(v)\} = \langle V(v) \rangle_T \cap \mathring{E}, \\ \text{and} \quad \mathring{E}_2(v) &:= \mathring{E}_1(v) \uplus \{e \in \mathring{E} \mid e = \mathrm{edge}_T(\phi(x) \to \{v\})\} \end{aligned}$$

Note that $\mathring{E}_1(v)$ and $\mathring{E}_2(v)$ are superpatches of T for all $v \in V$. Now, it is easy to see that

$$\mathcal{C}' := \{\mathring{E}_1(v), \mathring{E}_2(v) \mid v \in V\} \cup \binom{\mathring{E}}{1}$$

is a maximal $\mathring{E}$-hierarchy, and that every non-empty element from $\mathcal{C}'$ is in $\mathcal{C}(q)$: The latter follows, for each $v \in V$ and $F := \mathring{E}_1(v)$, because there exists a unique $\bar{x} \in X/{\approx_F}$ with $\#\bar{x} \geq 2$, and $\bar{x} = \{x\}$ must hold in view of Remark 2.11 and Lemma 6.8; while $F' := \mathring{E}_2(v)$ in case $F' \neq F$ implies $F' = F \mathbin{\dot\cup} \{e\}$ for some $e \in \mathring{E}$, so the claim follows this time because $x \in \underline{q}(e)$.

In view of Theorem 2 this implies that $\mathcal{C}(q)$ is ample. □

We will now present the "constant time oracle" announced in Chapter 5; with the help of this "oracle", we construct an algorithm of order $O((\#X)^2)$ that, given a set of quartet splits $\mathcal{Q}$ with $\underline{\mathcal{Q}} = X$ of cardinality $\#\mathcal{Q} = \#X - 3$, eventually computes $\mathcal{C}' \subseteq \mathcal{C}(\mathcal{Q})$ such that $\mathcal{C}' \cup \{\emptyset\}$ is a maximal hierarchy. The idea behind this algorithm is as follows: Suppose there exists some tree tightly concordant with $\mathcal{Q}$. Given two disjoint subsets $\mathcal{Q}_1,\ \mathcal{Q}_2 \subseteq \mathcal{Q}$ then

$$0 \leq \operatorname{exc}(\mathcal{Q}_1 \mathbin{\dot\cup} \mathcal{Q}_2) = \operatorname{exc}(\mathcal{Q}_1) + \operatorname{exc}(\mathcal{Q}_2) + 3 - \#(\underline{\mathcal{Q}_1} \cap \underline{\mathcal{Q}_2})$$

and for excess-free disjoint subsets $\mathcal{Q}_1,\ \mathcal{Q}_2 \subseteq \mathcal{Q}$ this comes down to

$$0 \leq \operatorname{exc}(\mathcal{Q}_1 \mathbin{\dot\cup} \mathcal{Q}_2) = 3 - \#(\underline{\mathcal{Q}_1} \cap \underline{\mathcal{Q}_2})\,, \tag{8.1}$$

so the sets $\underline{\mathcal{Q}_1}$ and $\underline{\mathcal{Q}_2}$ can intersect in at most three points. So, given three excess-free and disjoint subsets $\mathcal{Q}_1,\ \mathcal{Q}_2\,,\mathcal{Q}_3 \subseteq \mathcal{Q}$ then

$$(\underline{\mathcal{Q}_1} \cup \underline{\mathcal{Q}_2}) \cap \underline{\mathcal{Q}_3} = (\underline{\mathcal{Q}_1} \cap \underline{\mathcal{Q}_3}) \cup (\underline{\mathcal{Q}_2} \cap \underline{\mathcal{Q}_3}) \tag{8.2}$$

can be computed in constant time, provided that we already know $\underline{\mathcal{Q}_i} \cap \underline{\mathcal{Q}_j}$ for $1 \leq i < j \leq 3$.

Now, we can modify Algorithm 1 so that it uses our "constant time oracle" provided by equations (8.1) and (8.2), see Algorithm 3: We use a two-dimensional array of sets $\mathit{Ins}[\cdot,\cdot]$ to store the intersection $\underline{A} \cap \underline{B}$ of subsets $A,\ B \subseteq \mathcal{Q}$, and we take care that $\#\mathit{Ins}[A,B] \leq 3$ holds. We start by calculating $\mathit{Ins}[\cdot,\cdot]$ for all one-element subsets of $\mathcal{Q}$ (lines 1–4) what can be done in time $O((\#\mathcal{Q})^2)$ because, in each step, we calculate the intersection of two sets with exactly four elements. From (8.1) we know that $\operatorname{exc}(\mathcal{Q}_1 \mathbin{\dot\cup} \mathcal{Q}_2) = 0$ if and only if $\#(\underline{\mathcal{Q}_1} \cap \underline{\mathcal{Q}_2}) = 3$ holds (lines 8, 13); and (8.2) allows us to compute $\mathit{Ins}[\cdot,\cdot]$ for the new set $A \mathbin{\dot\cup} B$ versus all remaining sets in time $O(\#\mathcal{Q})$ because, in each step, we calculate the union of two sets with at most three elements (lines 15–18). Then, the following lemma is a consequence of Lemma 5.9 and Corollary 8.1:

LEMMA 8.3. *Given as input a set of quartet splits $\mathcal{Q}$ with $\underline{\mathcal{Q}} = X$ of cardinality $\#\mathcal{Q} = \#X - 3$ such that there is a unique X-tree concordant with $\mathcal{Q}$. Then, Algorithm 3 computes a hierarchy $\mathcal{C}' \subseteq \mathcal{C}(\mathcal{Q})$ such that $\mathcal{C}' \cup \{\emptyset\}$ is a maximal hierarchy. The order of Algorithm 3 is $O((\#X)^2)$.*

We will see in the following section how we can use the information contained in $\mathcal{C}(\mathcal{Q})$ to find an algorithm of order $O((\#X)^2)$ constructing the unique X-tree concordant with $\mathcal{Q}$ if it does exist — the explicit tree amalgamation process can be described more easily on the basis of X-trees instead of splits, so we replace every quartet split $ab|cd$ by the quartet tree $T_{ab|cd}$ defined in (7.11).

For all distinct Q_1, $Q_2 \in \mathcal{Q}$ **Do**
    Let $\mathit{Ins}[\{Q_1\},\{Q_2\}] := \underline{Q_1} \cap \underline{Q_2}$;
    If $\#\mathit{Ins}[\{Q_1\},\{Q_2\}] > 3$ **Then Return(False)** and **Exit**
End For;
Let $\mathcal{C}' := \binom{\mathcal{Q}}{1}$, $\mathcal{A} := \binom{\mathcal{Q}}{1}$, and $\mathcal{B} := \emptyset$;
While $\mathcal{A} \neq \{\mathcal{Q}\}$ **And** $\mathcal{A} \neq \emptyset$ **Do**
    Choose some $A \in \mathcal{A}$;
    If $\#\mathit{Ins}[A, B] < 3$ holds for all $B \in \mathcal{B}$
    Then
        Let $\mathcal{A} := \mathcal{A} - \{A\}$;
        Let $\mathcal{B} := \mathcal{B} \cup \{A\}$
    Else
        Choose some $B \in \mathcal{B}$ with $\#\mathit{Ins}[A, B] = 3$;
        Let $C := A \cup B$;
        For all $D \in \mathcal{A} \cup \mathcal{B} - \{A, B\}$ **Do**
            Let $\mathit{Ins}[C, D] := \mathit{Ins}[A, D] \cup \mathit{Ins}[B, D]$;
            If $\#\mathit{Ins}[C, D] > 3$ **Then Return(False)** and **Exit**
        End For;
        Let $\mathcal{A} := \mathcal{A} \cup \{C\} - \{A\}$;
        Let $\mathcal{B} := \mathcal{B} - \{B\}$;
        Let $\mathcal{C}' := \mathcal{C}' \cup \{C\}$
    End If
End While;
If $\mathcal{A} = \{\mathcal{Q}\}$ **Then Return**($\mathcal{C}'$) **Else Return(False)**.

ALGORITHM 3: Constructing an ample hierarchy $\mathcal{C}' \subseteq \mathcal{C}(\mathcal{Q})$ for a set of quartet splits $\mathcal{Q}$ (written in PSEUDO-PASCAL). INPUT: Set of quartet splits $\mathcal{Q}$ with $\underline{\mathcal{Q}} = X$ of cardinality $\#\mathcal{Q} = \#X - 3$; OUTPUT: Hierarchy $\mathcal{C}' \subseteq \mathcal{C}(\mathcal{Q})$ of excess-free subsets such that $\mathcal{C}' \cup \{\emptyset\}$ is a maximal hierarchy, or `False`.

We end this section with an example showing that there might in fact be exponentially many trees concordant with a collection of quartet splits. This might not be a problem in certain cases: For example, define

$$\mathcal{Q} := \big\{\{\{1,2\},\{j,k\}\} \;\big|\; 3 \leq j < k \leq n\big\} \subseteq \mathcal{Q}(\{1,\dots,n\})$$

for an integer $n \geq 4$, then $\mathcal{T}(\mathcal{Q})$ contains $(2n-7)!!$ many binary X-trees and therefore also super-exponentially many trees in total; but there is exactly *one* tree in $\mathcal{T}(\mathcal{Q})$ that is minimal (with respect to $\leq$); and this tree (having exactly one inner edge, separating $\{1,2\}$ from $\{3,\dots,n\}$) is the one every practitioner would probably be interested in. In the following example, on the contrary, there are not only exponentially many trees concordant with the collection of quartet splits, but also *every* such tree is necessarily binary. Note that this collection of quartet splits is of minimum cardinality.

Example.

19. For $V^* = \{v_1, \ldots, v_7, u_1, \ldots, u_5\}$ let (V^*, E^0) denote the tree depicted in Fig. 8.1 left, and (V^*, E^1) the tree depicted right. Note that, in both trees, the median of v_1, v_2, v_3 is u_4, and the median of v_5, v_6, v_7 is u_1.

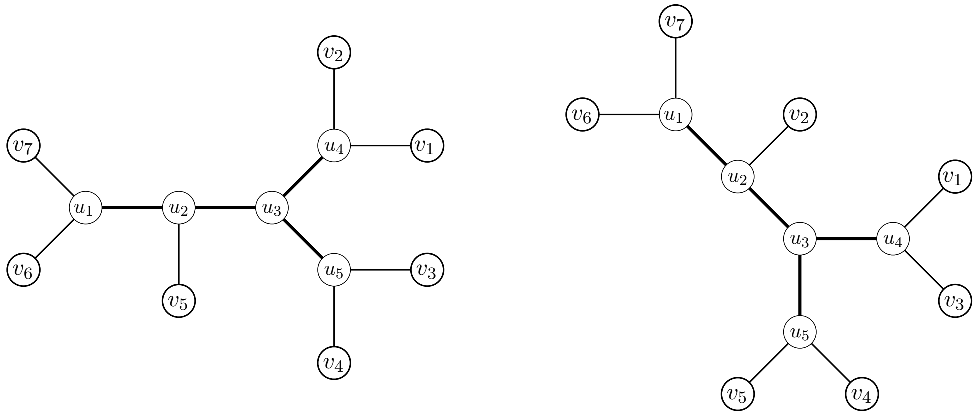


FIGURE 8.1. The trees (V^*, E^0) (left) and (V^*, E^1) (right).

For an integer $k \in \mathbb{N}_0$ we define $Y_k := \{k+1, k+2, \ldots, k+7\}$ and the Y_k-trees $T_k^i := (V^*, E^i; \phi_k)$ for $i = 0, 1$ via

$$\phi_k : Y_k \to V^* \;:\; j \mapsto v_{j-k}$$

(cf. Fig. 8.2 for two explicit examples).

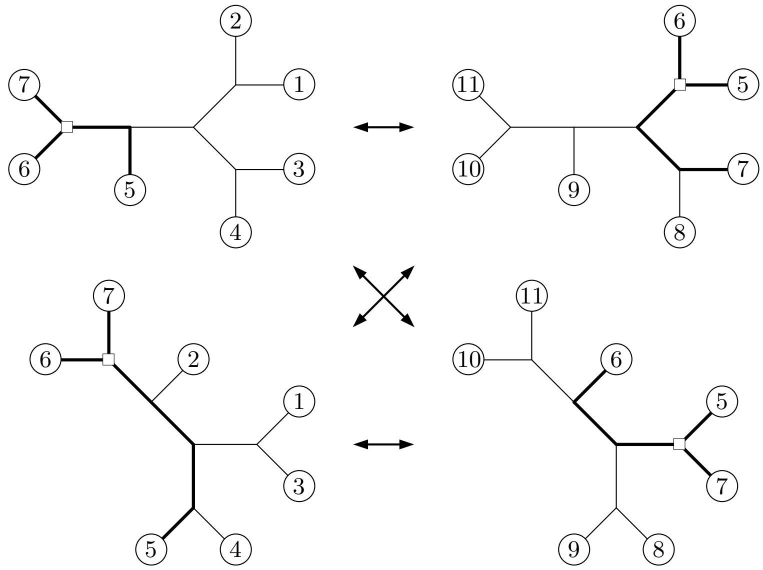


FIGURE 8.2. The four trees T_k^i for $k = 0, 4$ (left, right) and $i = 0, 1$ (top, bottom); the median of the vertices labeled 5, 6, 7 is indicated by a square.

Furthermore, we define the sets of quartet splits

$$\begin{aligned} \mathcal{Q}' &:= \{21|45,\ 34|62,\ 56|13,\ 25|67\} \subseteq \mathcal{Q}(\{1,\dots,7\}) \\ \text{and}\quad \mathcal{Q}'_k &:= \Big\{\{\{a+k, b+k\}, \{c+k, d+k\}\} \;\Big|\; ab|cd \in \mathcal{Q}'\Big\} \subseteq \mathcal{Q}(Y_k) \end{aligned}$$

for all $k \in \mathbb{N}_0$, and we can easily establish (cf. Example 15) that $\mathcal{T}(\mathcal{Q}'_k) = \{T_k^0, T_k^1\}$ holds.

Now, we define

$$X_K := \{1, 2, \dots, 4K+7\} \quad \text{and} \quad \mathcal{Q}_K := \bigcup_{k=0}^{K} \mathcal{Q}'_{4k} \quad \text{for} \quad K \in \mathbb{N}_0$$

and claim that (i) there exist exactly 2^{K+1} X_K-trees concordant with $\mathcal{Q}_K$; that (ii) every tree $T = (V, E; \phi) \in \mathcal{T}(\mathcal{Q}_K)$ is binary; and (iii) the median m_1 of $\phi(4K+5)$, $\phi(4K+6)$, $\phi(4K+7)$ in T is adjacent to $\phi(4K+6)$ and to $\phi(4K+7)$ — or, equivalently, that $\{\{1, 2, \dots, 4K+5\}, \{4K+6, 4K+7\}\}$ is an element of $\mathcal{S}[T]$:

We use induction and note that for $K = 0$ there remains nothing to prove. Suppose the claim holds for $\mathcal{Q}_{K-1}$ and let $k := 4K$. Clearly,

$$X_{K-1} \cap Y_k = \{k+1, k+2, k+3\}\,.$$

Now, choose an arbitrary tree $T = (V, E; \phi) \in \mathcal{T}(\mathcal{Q}_{K-1})$. Firstly, we want to amalgamate T with T_k^0 what is clearly possible; but the resulting tree is also uniquely defined (and, hence, binary): The median m_2 of $\phi_k(k+1)$, $\phi_k(k+2)$, $\phi_k(k+3)$ in T_k^0 is adjacent to $\phi_k(k+1)$, and — together with the induction hypothesis — we see that conditions (i–iii) of the Tree Joining Theorem must hold. Furthermore,

$$\{\{k+1, \dots, k+5\}, \{k+6, k+7\}\} \in \mathcal{S}[T_k^0]$$

implies that $\{\{1, 2, \dots, 4K+5\}, \{4K+6, 4K+7\}\}$ is a split of the resulting tree in view of $4K+6,\ 4K+7 \notin X_{K-1}$. The same holds true when amalgamating T with T_k^1.

Finally, we can define

$$\begin{aligned} \mathcal{T}_K := \big\{T \text{ is an } X_K\text{-tree} \;\big|\; & T|_{X_{K-1}} \in \mathcal{T}(\mathcal{Q}_{K-1}) \quad \text{and} \\ & T|_{Y_k} \cong T_k^i \;\text{ for some }\; i \in \{0,1\}\big\} \end{aligned}$$

(where, in fact, $\mathcal{T}_K$ shall contain the isomorphism classes of X_K-trees) and claim that $\mathcal{T}_K = \mathcal{T}(\mathcal{Q}_K)$ as well as $\#\mathcal{T}_K = 2^{K+1}$ holds: The inclusion "$\subseteq$" follows from the definition of $\mathcal{T}_K$, "$\supseteq$" follows from the above, and because amalgamating distinct trees clearly leads to distinct supertrees, using the induction hypothesis gives us $\#\mathcal{T}(\mathcal{Q}_K) = \#\mathcal{T}_K = 2^{K+1}$.

So, we have constructed sets of quartet splits $\mathcal{Q}_K \subseteq \mathcal{Q}(X_K)$ for all $K \in \mathbb{N}_0$ with $\#\mathcal{Q}_K = 4K+4$ and $\#X_K = 4K+7$, such that $\mathcal{T}(\mathcal{Q}_K)$ contains exactly $2^{(\#X_K - 3)/4}$ (non-isomorphic) trees, all of them binary.

8.2. Reconstructing trees from subtrees

First, we give generalizations of "(tightly) concordant" and "excess" for sets of trees — recall that we announced in Section 1.3 what "excess-free" should mean for a set of trees.

For an X-tree T let $\mathcal{L}(T)$ denote the set of labels X. In the following, an X-tree will also be called a *labeled* tree, thereby not referring to the set of labels.

DEFINITION 8.1. Given a (finite) collection $\mathcal{T}$ of labeled trees, we define

$$\mathcal{L}(\mathcal{T}) := \bigcup\{\mathcal{L}(T) \bigm| T \in \mathcal{T}\}\,.$$

An $\mathcal{L}(\mathcal{T})$-tree $T = (V, E; \phi)$ is *concordant* with $\mathcal{T}$ if $T' \le T$ holds for all $T' \in \mathcal{T}$; and T is called *tightly concordant* with $\mathcal{T}$ if T is concordant with $\mathcal{T}$, but $T_{(F,X)}$ is not concordant with $\mathcal{T}$ for every proper subset $F \subsetneqq \mathring{E}$.

DEFINITION 8.2. Given a (finite) collection $\mathcal{T}$ of binary labeled trees, we define

$$\operatorname{exc}(\mathcal{T}) := \#\mathcal{L}(\mathcal{T}) - 3 - \sum_{T\in\mathcal{T}} \bigl(\#\mathcal{L}(T) - 3\bigr) \tag{8.3}$$

to be the *excess* of $\mathcal{T}$. We say that $\mathcal{T}$ is *excess-free* if $\operatorname{exc}(\mathcal{T}) = 0$ holds, and we denote the set of excess-free subsets of $\mathcal{T}$ by $\mathcal{C}(\mathcal{T}) \subseteq \mathcal{P}(\mathcal{T})$.

Clearly, an X-tree T is (tightly) concordant with a set of quartet splits $\mathcal{Q}$ with $\underline{\mathcal{Q}} = X$ if and only if T is (tightly) concordant with the collection of labeled trees

$$\mathcal{T} := \{T_Q \bigm| Q \in \mathcal{Q}\} \tag{8.4}$$

where T_Q is defined in (7.11) on page 72; and (8.4) also gives rise to a canonical 1-to-1 correspondence between $\mathcal{C}(\mathcal{Q})$ and $\mathcal{C}(\mathcal{T})$ in this case.

The following theorem is a simple consequence of — and together with Corollary 5.6 clearly "equivalent" to — Theorem 9:

THEOREM 10. *Given a collection $\mathcal{T}$ of binary labeled trees that is excess-free, and a binary $\mathcal{L}(\mathcal{T})$-tree T, then T is the unique tree concordant with $\mathcal{T}$ if and only T is tightly concordant with $\mathcal{T}$, and the patchwork $\mathcal{C}(\mathcal{T})$ of excess-free subsets of $\mathcal{T}$ is ample.*

PROOF. For each labeled tree $T = (V, E; \phi) \in \mathcal{T}$ we fix a label $x_T \in \mathcal{L}(T)$. For every inner edge $e = \{u, v\} \in \mathring{E}$ there must exist y_e, w_e, $z_e \in \mathcal{L}(T)$ such that

$$[\phi(x_T), \phi(y_e)] \cap e = \{u\} \quad \text{and} \quad [\phi(w_e), \phi(z_e)] \cap e = \{v\}$$

holds; set $Q_e := x_T y_e | w_e z_e$ and

$$\mathcal{Q}_T := \{Q_e \bigm| e \in \mathring{E}\}$$

then T is clearly tightly concordant with $\mathcal{Q}$, so $\underline{\mathcal{Q}_T} = \mathcal{L}(T)$ must hold. As we also know $\#\mathcal{Q}_T = \#\mathring{E} = \#\mathcal{L}(T) - 3$, Corollary 8.2 secures that T is the unique $\mathcal{L}(\mathcal{T})$-tree concordant with $\mathcal{Q}_T$. We define $\mathcal{Q}$ via the disjoint union

$$\mathcal{Q} := \bigcup\{\mathcal{Q}_T \bigm| T \in \mathcal{T}\} \subseteq \mathcal{Q}(\mathcal{L}(\mathcal{T}))$$

and easily calculate that $\mathcal{Q}$ is excess-free, see also (8.5) below.

The remaining two parts of the proof are straight forward: If T^* is the unique X-tree concordant with $\mathcal{T}$ (for $X := \mathcal{L}(\mathcal{T})$) then T^* is also the unique X-tree concordant with $\mathcal{Q}$. So, by Theorem 9, T^* is tightly concordant with $\mathcal{Q}$ and hence, with $\mathcal{T}$; and the patchwork $\mathcal{Q}$ is ample. For $\#\mathcal{T} = 1$ there remains nothing to prove; otherwise, we apply Theorem 9 (vii) to the proper $\mathcal{C}(\mathcal{Q})$-family $\mathcal{F} := (\mathcal{Q}_T)_{T\in\mathcal{T}}$ with index set $I := \mathcal{T}$ and trace $(I, \mathcal{J})$ defined by

$$\mathcal{J} := \left\{ J \subseteq I \;\middle|\; \bigcup\{\mathcal{Q}_T \mid T \in J\} \in \mathcal{C}(\mathcal{Q}) \right\}.$$

Therefore, $(I, \mathcal{J})$ must be ample and, in view of

$$\begin{aligned}
\operatorname{exc}\left(\bigcup\{\mathcal{Q}_T \mid T \in J\}\right) &= \#\left(\bigcup\nolimits_{T\in J} \underline{\mathcal{Q}_T}\right) - \sum_{T\in J} \#\mathcal{Q}_T \; - 3 \\
&= \#\mathcal{L}(J) - \sum_{T\in J} \left(\#\mathcal{L}(T) - 3\right) - 3 \\
&= \operatorname{exc}(J)
\end{aligned} \tag{8.5}$$

for all $J \subseteq I$ we infer $\mathcal{C}(\mathcal{T}) = \mathcal{J}$ and, finally, that $\mathcal{C}(\mathcal{T})$ is ample.

Vice versa, suppose T^* is tightly concordant with $\mathcal{T}$, and that $\mathcal{C}(\mathcal{T})$ is ample. Clearly T^* is tightly concordant with $\mathcal{Q}$, and since $\mathcal{C}(\mathcal{Q}_T)$ is ample for every $T \in \mathcal{T}$ we conclude that $\mathcal{C}(\mathcal{Q})$ is ample, too. Then Theorem 9 secures that T^* is the unique tree concordant with $\mathcal{Q}$ and, hence, also the unique tree concordant with $\mathcal{T}$. ☐

Using these definitions, we can restate Theorem 1 and Corollary 1.1 from the introduction:

Theorem 11. *Given a collection of binary labeled trees $\mathcal{T}$ that is excess-free and of cardinality greater or equal two. Then, there exists a unique $\mathcal{L}(\mathcal{T})$-tree T concordant with $\mathcal{T}$ if and only if there is a bipartition of $\mathcal{T}$ into two proper, disjoint subsets $\mathcal{T}_1$, $\mathcal{T}_2 \subset \mathcal{T}$ such that*

(i) *$\mathcal{T}_j$ is excess-free for $j = 1, 2$;*
(ii) *there exists a unique $\mathcal{L}(\mathcal{T}_j)$-tree T_j concordant with $\mathcal{T}_j$ for $j = 1, 2$; and*
(iii) *T is the unique $\mathcal{L}(\mathcal{T})$-tree concordant with $\{T_1, T_2\}$.*

Clearly, the "if"-part of the theorem is trivial: The existence of a bipartition $\{\mathcal{T}_1, \mathcal{T}_2\}$ of $\mathcal{T}$ satisfying (ii) and (iii) surely implies that T is the unique $\mathcal{L}(\mathcal{T})$-tree concordant with $\mathcal{T}$.

If we would only know that Theorem 11 holds true, we could easily establish a proof of Theorem 10 by repeatedly using Theorem 11 for the crucial steps. In this sense, Theorems 8–11 are "equivalent", because every one theorem can be used to easily prove the remaining ones.

Corollary 8.4. *Given a collection of binary labeled trees $\mathcal{T}$ that is excess-free and of cardinality greater or equal two such that there exists a unique $\mathcal{L}(\mathcal{T})$-tree T concordant with $\mathcal{T}$. Then, there exist two distinct trees T, $T' \in \mathcal{T}$ such that the collection $\{T, T'\}$ is excess-free, and there exists a unique supertree concordant with $\{T, T'\}$.*

Proof of Theorem 11. Suppose that $\mathcal{T}$ is an excess-free collection of binary labeled trees with cardinality $\#\mathcal{T} \geq 2$, and that T is the unique $\mathcal{L}(\mathcal{T})$-tree concordant with $\mathcal{T}$. Applying Theorem 10 we get a bipartition of $\mathcal{T}$ into two excess-free subsets

$\mathcal{T}_1$, $\mathcal{T}_2 \subset \mathcal{T}$. Then $T_j := T|_{\mathcal{L}(\mathcal{T}_j)}$ is concordant with $\mathcal{T}_j$, and in view of $\#\mathcal{L}(T_1) \cap \mathcal{L}(T_2) = 3$ we can easily check that T_j is tightly concordant with $\mathcal{T}_j$ for $j = 1, 2$. Since also $\mathcal{C}(\mathcal{T}_j)$ is ample for $j = 1, 2$ we can apply Theorem 10 two more times and infer that T_j is the unique tree concordant with $\mathcal{T}_j$ for $j = 1, 2$. But this implies that T must be the unique $\mathcal{L}(\mathcal{T})$-tree concordant with $\{T_1, T_2\}$ what completes the "only if"-part of the theorem, so we are done. ☐

Notice that there is no straightforward generalization of Definition 8.2 to collections of non-binary labeled trees: For a binary X-tree $T = (V, E; \phi)$ set $T_e := T_{(\{e\}, X)}$ for all $e \in \mathring{E}$; then T is obviously the unique X-tree concordant with $\mathcal{T} := \{T_e \mid e \in \mathring{E}\}$. However, using (8.3), all but the one-element subsets of $\mathcal{T}$ have negative excess; while replacing $\#\mathcal{L}(T) - 3$ by $\#\operatorname{Int}(T)$ in the definition of $\operatorname{exc}(\mathcal{T})$ in (8.3) produces positive numbers for all proper and non-empty subsets $\mathcal{T}'$ of $\mathcal{T}$ in view of

$$\#\mathcal{L}(\mathcal{T}) - 3 - \sum_{T \in \mathcal{T}} \#\operatorname{Int}(T) = \#(\mathcal{T} - \mathcal{T}') > 0 .$$

Algorithm 4 is a modification of Algorithm 3 working with an input of labeled trees instead of quartet splits. The only differences between these two algorithms can be found in lines 1–6 and 24, so we easily infer:

LEMMA 8.5. *Given as input an excess-free set $\mathcal{T}$ of binary labeled trees such that there is a unique $\mathcal{L}(\mathcal{T})$-tree concordant with $\mathcal{T}$. Then, Algorithm 4 computes a hierarchy $\mathcal{C}' \subseteq \mathcal{C}(\mathcal{T})$ such that $\mathcal{C}' \cup \{\emptyset\}$ is a maximal hierarchy. The order of Algorithm 4 is $O(n^2)$ for $n := \#\mathcal{L}(\mathcal{T})$.*

To see that Algorithm 4 is of order $O(n^2)$, it is sufficient to check lines 1–4, because the runtime of the remaining part is clearly the same as in Algorithm 3. The time needed for computing the intersection $\mathcal{L}(T_1) \cap \mathcal{L}(T_2)$ in line 2 surely depends on the data structure used to represent the sets $\mathcal{L}(T_1)$, $\mathcal{L}(T_2)$. Using unordered sets this can be carried out in at most $\max(\#\mathcal{L}(T_1), \#\mathcal{L}(T_2))$ comparison steps, but even when using unordered sets, the number of comparison steps cannot exceed $\#\mathcal{L}(T_1) \cdot \#\mathcal{L}(T_2)$. Using the second bound and applying (8.3), we can estimate the total number of comparison steps by

$$\begin{aligned} \text{steps} &\leq \sum_{\substack{T_1, T_2 \in \mathcal{T} \\ T_1 \neq T_2}} \#\mathcal{L}(T_1) \cdot \#\mathcal{L}(T_2) \\ &< \sum_{T_1 \in \mathcal{T}} \#\mathcal{L}(T_1) \cdot \sum_{T_2 \in \mathcal{T}} \#\mathcal{L}(T_2) \\ &= (\#\mathcal{L}(\mathcal{T}) + 3\,\#\mathcal{T} - 3)^2 \\ &< (4\,\#\mathcal{L}(\mathcal{T}))^2 \\ &= 16\,\#\mathcal{L}(\mathcal{T})^2 . \end{aligned}$$

8.3. Fast amalgamation of an excess-free collection of trees

Let $\mathcal{T}$ denote an excess-free collection of binary labeled trees, and let us suppose that, using Algorithm 4 (or 3), we have computed a sub-hierarchy $\mathcal{C}' \subseteq \mathcal{C}(\mathcal{T})$ such that $\mathcal{C} \cup \{\emptyset\}$ is a maximal hierarchy. Can we use the information contained in $\mathcal{C}'$ to

```
For all distinct $T_1, T_2 \in \mathcal{T}$ Do
    Let $Ins[\{T_1\}, \{T_2\}] := \mathcal{L}(T_1) \cap \mathcal{L}(T_2)$;
    If $\#Ins[\{T_1\}, \{T_2\}] > 3$ Then Return(False) and Exit
End For;
Let $\mathcal{C}' := \binom{\mathcal{T}}{1}$, $\mathcal{A} := \binom{\mathcal{T}}{1}$, and $\mathcal{B} := \emptyset$;
While $\mathcal{A} \neq \{\mathcal{T}\}$ And $\mathcal{A} \neq \emptyset$ Do
    Choose some $A \in \mathcal{A}$;
    If $\#Ins[A, B] < 3$ holds for all $B \in \mathcal{B}$
    Then
        Let $\mathcal{A} := \mathcal{A} - \{A\}$;
        Let $\mathcal{B} := \mathcal{B} \cup \{A\}$
    Else
        Choose some $B \in \mathcal{B}$ with $\#Ins[A, B] = 3$;
        Let $C := A \cup B$;
        For all $D \in \mathcal{A} \cup \mathcal{B} - \{A, B\}$ Do
            Let $Ins[C, D] := Ins[A, D] \cup Ins[B, D]$;
            If $\#Ins[C, D] > 3$ Then Return(False) and Exit
        End For;
        Let $\mathcal{A} := \mathcal{A} \cup \{C\} - \{A\}$;
        Let $\mathcal{B} := \mathcal{B} - \{B\}$;
        Let $\mathcal{C}' := \mathcal{C}' \cup \{C\}$
    End If
End While;
If $\mathcal{A} = \{\mathcal{T}\}$ Then Return($\mathcal{C}'$) Else Return(False).
```

Algorithm 4: Constructing an ample hierarchy $\mathcal{C}' \subseteq \mathcal{C}(\mathcal{T})$ for a collection of trees $\mathcal{T}$ (written in Pseudo-Pascal). Input: Excess-free collection $\mathcal{T}$ of binary labeled trees; Output: Hierarchy $\mathcal{C}' \subseteq \mathcal{C}(\mathcal{T})$ of excess-free subsets such that $\mathcal{C}' \cup \{\emptyset\}$ is a maximal hierarchy, or `False`.

efficiently compute the unique $\mathcal{L}(\mathcal{T})$-tree concordant with $\mathcal{T}$ if such a unique tree does exist?

Surely we can, see Algorithm 5 on page 87:

Lemma 8.6. *Given an excess-free collection $\mathcal{T}$ of binary labeled trees and a maximal hierarchy $\mathcal{C}' \subseteq \mathcal{C}(\mathcal{T}) \cup \{\emptyset\}$. If there is a unique $\mathcal{L}(\mathcal{T})$-tree T concordant with $\mathcal{T}$ (or — equivalently — if T is tightly concordant with $\mathcal{T}$), then Algorithm 5 returns the $\mathcal{L}(\mathcal{T})$-tree T; otherwise, the algorithm returns* `False`. *The order of Algorithm 5 is $O(n^2)$ for $n := \#\mathcal{L}(\mathcal{T})$.*

Following the line of thought established in Theorem 11 and Section 7.1 it is clear that Algorithm 5 in fact returns the unique tree concordant with $\mathcal{T}$. For the runtime

`Procedure` AmalgamateCollection ($\mathcal{T}$`: collection of labeled trees,`
$\mathcal{C}'$`: maximal` $\mathcal{T}$`-hierarchy`): $\mathcal{L}(\mathcal{T})$`-Tree Or False;`

1 `If` $\mathcal{T} = \{T\}$ for some binary labeled tree T `Then Return(`T`)` and `Exit`;
2 Choose disjoint non-empty clusters $\mathcal{T}_1$, $\mathcal{T}_2 \in \mathcal{C}'$ with $\mathcal{T}_1 \,\dot\cup\, \mathcal{T}_2 = \mathcal{T}$;
3 Let $T_1 :=$ AmalgamateCollection ($\mathcal{T}_1$, $\mathcal{C}'(\subseteq \mathcal{T}_1)$);
4 Let $T_2 :=$ AmalgamateCollection ($\mathcal{T}_2$, $\mathcal{C}'(\subseteq \mathcal{T}_2)$);
5 `If` $T_1 =$ `False Or` $T_1 =$ `False Then Return(False)` and `Exit`;
6 Let $(V_1, E_1; \phi_1) = T_1$ and $(V_2, E_2; \phi_2) = T_2$;
7 Choose distinct x_1, x_2, $x_3 \in \mathcal{L}(\mathcal{T})$ with $\{x_1, x_2, x_3\} = \mathcal{L}(T_1) \cap \mathcal{L}(T_2)$;
8 Compute $m_1 := \mathrm{med}(\phi_1(x_1), \phi_1(x_2), \phi_1(x_3))$ in T_1;
9 Compute $m_2 := \mathrm{med}(\phi_2(x_1), \phi_2(x_2), \phi_2(x_3))$ in T_2;
10 `For` $k = 1, 2, 3$ `Do`
11   `If` $\phi_1(x_k)$ and m_1 are adjacent in T_1
12   `Then` let $i_k := 1$
13   `Else If` $\phi_2(x_k)$ and m_2 are adjacent in T_2
14   `Then` let $i_k := 2$
15   `Else Return(False)` and `Exit`
16   `End If`
17 `End For`;
18 `For` $k = 1, 2, 3$ `Do`
19   Remove the vertex $\phi_{i_k}(x_k)$ from the vertex set, the edge $\{\phi_{i_k}(x_k), m_{i_k}\}$ from the edge set, and the label x_k from the label set of T_{i_k};
20 `End For`;
21 Identify the vertices m_1 of T_1 and m_2 of T_2, thereby obtaining a $\mathcal{L}(T_1) \cup \mathcal{L}(T_2)$-tree T;
22 `Return(`T`)`.

Algorithm 5: Constructing the unique $\mathcal{L}(\mathcal{T})$-tree concordant with an excess-free collection $\mathcal{T}$ of binary labeled trees. Input: Excess-free collection $\mathcal{T}$ of binary labeled trees, maximal hierarchy $\mathcal{C}' \subseteq \mathcal{C}(\mathcal{T}) \cup \{\emptyset\}$; Output: Unique $\mathcal{L}(\mathcal{T})$-tree concordant with $\mathcal{T}$ or `False`.

behavior, we first note that in view of $\#\mathcal{C}' = 2\,\#\mathcal{T}$ and (8.2),

$$\#\mathcal{T} = \sum_{T \in \mathcal{T}} 1 \leq \sum_{T \in \mathcal{T}} \big(\#\mathcal{L}(T) - 3\big) = \#\mathcal{L}(\mathcal{T}) - 3 = n - 3\,,$$

so the cardinality of $\mathcal{C}'$ (and therefore the number of calls of AmalgamateCollection) is linear in n. If we use a rooted tree to represent the maximal hierarchy $\mathcal{C}'$, then finding non-empty disjoint $\mathcal{T}_1$, $\mathcal{T}_2 \in \mathcal{C}'$ with $\mathcal{T}_1 \,\dot\cup\, \mathcal{T}_2 = \mathcal{T}$ (line 2) can be carried out in constant time, as well as computing $\mathcal{C}'(\subseteq \mathcal{T}_1)$ and $\mathcal{C}'(\subseteq \mathcal{T}_2)$ (lines 3,4). Computation of $\mathcal{L}(T_1) \cap \mathcal{L}(T_2)$ can be done in linear time in n if we use ordered lists to represent $\mathcal{L}(T_1)$ and $\mathcal{L}(T_2)$. Note that when checking the runtime behavior of Algorithm 4, we saw that unordered lists "also do the job," and an analogous argumentation could be used here. But having used Algorithm 4 to compute $\mathcal{C}' \subseteq \mathcal{C}(\mathcal{T}) \cup \{\emptyset\}$, we have in fact already computed

$$\mathcal{L}(T_1) \cap \mathcal{L}(T_2) = \mathcal{L}(\mathcal{T}_1) \cap \mathcal{L}(\mathcal{T}_2) = Ins[\mathcal{T}_1, \mathcal{T}_2]\,.$$

Finding the median of a tree can obviously be done in time linear in the number of vertices, for example by a "flood fill" algorithm. The runtime of lines 10–22 does not depend on n, so we have established that Algorithm 5 has order $O(n^2)$.

Algorithm 5 will return an $\mathcal{L}(\mathcal{T})$-tree T concordant with $\mathcal{T}$ only if T is tightly concordant with $\mathcal{T}$ — and, therefore, the unique tree concordant with $\mathcal{T}$. One can easily modify the algorithm so that it returns one, all, or all minimal $\mathcal{L}(\mathcal{T})$-trees concordant with $\mathcal{T}$ (instead of `False`) if this condition is violated; and with a little more sophisticated modification, we can also use Algorithm 5 to directly compute some consensus tree of all trees concordant with $\mathcal{T}$.

8.4. The dyadic closure

In the last two sections, we always supposed that we were given a *minimum* number of quartet splits or labeled trees when deciding whether there exists a unique supertree. This requirement is obviously too restrictive when it comes to "real world" applications; in this section, we show how to seriously weaken this requirement, and still obtain a fast algorithm to reconstruct the unique supertree.

Suppose that $\mathcal{Q} \subseteq \mathcal{Q}(X)$ is a collection of quartet splits that contains a sub-collection $\mathcal{Q}' \subseteq \mathcal{Q}$ satisfying $\underline{\mathcal{Q}'} = X$ and $\#\mathcal{Q}' = \#X - 3$. Suppose further that there exists a unique phylogenetic X-tree concordant with $\mathcal{Q}'$. Then, only two possibilities remain: Either T is also concordant with $\mathcal{Q}$, or there is no X-tree at all concordant with $\mathcal{Q}$. We can easily check which possibility holds by testing if T is concordant with $\mathcal{Q} - \mathcal{Q}'$.

But even if we know that $\mathcal{Q}$ contains such a minimum collection $\mathcal{Q}'$, how can we possibly find it in reasonable time? To check all subsets of $\mathcal{Q}$ of cardinality $\#X - 3$ is computationally infeasible, because there exist $\binom{\#\mathcal{Q}}{\#X-3}$ many of them — in fact, determining if an arbitrary collection of quartet splits $\mathcal{Q}$ contains such a minimum collection $\mathcal{Q}'$ is NP-hard [BBDS]. But applying the algorithms given above to a set of quartet splits $\mathcal{Q} \subseteq \mathcal{Q}(X)$ with $\#\mathcal{Q} > \#X - 3$ makes no sense, because Algorithm 3 depends on the fact that $\mathcal{C}(\mathcal{Q})$ is a patchwork. We can easily construct examples of arbitrary size where $\mathcal{C}(\mathcal{Q})$ is no patchwork, and the algorithms will surely fail, even though $\mathcal{Q}$ contains such a minimum collection $\mathcal{Q}'$ as a subset.

Thus, we will now tackle this problem based on the notion of "dyadic closure," introduced by Dekker [Dek86] and developed in [ESSW97b, ESSW99]. We will not go into detail, but Theorem 12 below indicates that the dyadic closure might be a useful tool when reconstructing trees from quartet splits.

DEFINITION 8.3. The *dyadic closure* of a collection $\mathcal{Q} \subseteq \mathcal{Q}(X)$ of quartet splits, denoted $cl_2(\mathcal{Q})$, is the minimal subset of $\mathcal{Q}(X)$ that contains $\mathcal{Q}$ and is closed under the following two rules:

(dc1) $\quad ab|cd,\ ab|ce \in cl_2(\mathcal{Q}) \implies ab|de \in cl_2(\mathcal{Q})$

(dc2) $\quad ab|cd,\ ac|de \in cl_2(\mathcal{Q}) \implies ab|ce,\ ab|de,\ bc|de \in cl_2(\mathcal{Q})$

The *semi-dyadic closure* of $\mathcal{Q}$, denoted $scl_2(\mathcal{Q})$, is the minimal subset of $\mathcal{Q}(X)$ that contains $\mathcal{Q}$ and is closed under rule (dc2).

The idea behind the dyadic closure is as follows: Given a phylogenetic X-tree T and a subset $\mathcal{Q}$ contained in

$$\begin{aligned}\mathcal{Q}[T] := \mathcal{Q}(\mathcal{S}[T]) &= \{Q \in \mathcal{Q}(X) \mid Q \leq S \text{ for some } S \in \mathcal{S}[T]\} \\ &= \{Q \in \mathcal{Q}(X) \mid T|_{\underline{Q}} \cong T_Q\},\end{aligned}$$

then the existence of quartet splits $ab|cd$, $ab|ce \in \mathcal{Q}$ implies

$$ab|cde \in \mathcal{S}[T]\big|_{\{a,b,c,d,e\}},$$

so $\mathcal{Q}[T]$ must also contain the quartet split $ab|de$. In case $ab|de \notin \mathcal{Q}$ we can replace $\mathcal{Q}$ by $\mathcal{Q} \cup \{ab|de\} \subseteq \mathcal{Q}[T]$, and using also rule (dc2) we end up (after a finite number of replacement steps) with the set $cl_2(\mathcal{Q})$.

Clearly, the dyadic closure of $\mathcal{Q} \subseteq \mathcal{Q}(X)$ can be computed in polynomial time in $n := \#X$ (see [ESSW97a] and, for a fast algorithm, [BBDS]). It is of course possible that $cl_2(\mathcal{Q})$ may contain both $xy|wz$ and $xw|yz$ for x, y, w, $z \in X$ — such an incompatible pair of quartet splits is said to be *contradictory.* In this case, it is clear that $\mathcal{T}(\mathcal{Q}) = \emptyset$ since any tree that induces the input splits must also induce the output split, and no tree can induce contradictory splits.[1] Note also that T can be reconstructed from $\mathcal{Q}[T]$ in $O(n\log(n))$ steps, see [KLW96, PT86]. Thus, the following theorem (which relies on Theorem 11) provides a polynomial time solution to the tree reconstruction problem under the conditions described at the beginning of this section:

THEOREM 12. *Suppose there exist collections of quartet splits $\mathcal{Q}' \subseteq \mathcal{Q} \subseteq \mathcal{Q}(X)$ with $\underline{\mathcal{Q}'} = X$ and $\#\mathcal{Q}' = \#X - 3$, and a binary X-tree T that is the unique phylogenetic X-tree concordant with $\mathcal{Q}'$. Then, either $cl_2(\mathcal{Q}) = \mathcal{Q}[T]$ and, as a consequence, $\mathcal{T}(\mathcal{Q}) = \{T\}$ holds; or $cl_2(\mathcal{Q})$ contains a contradictory pair of quartet splits what implies $\mathcal{T}(\mathcal{Q}) = \emptyset$.*

The proof of this theorem can be easily established using Theorem 11 and the following lemma that will be proven in [BBDS]:

LEMMA 8.7. *Given sets X, X_1, X_2 such that $X = X_1 \cup X_2$, a binary X-tree T, and a binary X_j-tree T_j for $j = 1, 2$. Then T is the unique tree concordant with $\{T_1, T_2\}$ if and only if $cl_2(\mathcal{Q}[T_1] \cup \mathcal{Q}[T_2]) = \mathcal{Q}[T]$.*

It is conjectured that Lemma 8.7 and Theorem 12 still hold true when replacing the dyadic closure cl_2 by the semi-dyadic closure scl_2; hopefully, this question can be answered in [BBDS].

Example 4 on page 29 shows that even when T is the unique phylogenetic X-tree concordant with $\mathcal{Q} \subseteq \mathcal{Q}(X)$, it may not necessarily be the case that $cl_2(\mathcal{Q}) = \mathcal{Q}[T]$ — otherwise, we would have found a polynomial time solution to the problem whether some known X-tree is the *unique* phylogenetic X-tree concordant with a given collection. Note that, for $\mathcal{Q}$ defined as in Example 4, we even have $cl_2(\mathcal{Q}) = \mathcal{Q}$, so computing the dyadic closure does not give us any new information. In fact, there

[1] In applications, it might still be useful to compute a tree that represents at least some phylogenetic groupings supported by $cl_2(\mathcal{Q})$, see [BG97, Bry97].

exists a straightforward generalization of the dyadic closure to k-adic closures $cl_k(\mathcal{Q})$ for every integer $k \geq 3$ — just note that in (dc1) and (dc2) we inferred a "valid" quartet split from *two* other quartet splits — and Bryant and Steel [BS95] established that $cl_{k+1}(\mathcal{Q}) \neq cl_k(\mathcal{Q})$ for *every* $k \geq 2$ in general.

On the other hand, there exist collections $\mathcal{Q} \subseteq \mathcal{Q}(X)$ with $cl_2(\mathcal{Q}) = \mathcal{Q}[T]$ for some X-tree T even if $\mathcal{Q}$ does *not* contain such a minimum subset $\mathcal{Q}'$ (with $\#\mathcal{Q}' = \#X - 3$, and T the unique X-tree concordant with $\mathcal{Q}'$) as the following example testifies:

Example.

20. Define $X := \{1, \ldots, 6\}$ and $\mathcal{Q} := \{12|35, 14|56, 23|46, 23|56\} \subseteq \mathcal{Q}(X)$. The unique phylogenetic X-tree concordant with $\mathcal{Q}$ is the caterpillar T with six leaves having non-trivial splits $12|3456$, $123|456$, $1234|56 \in \mathcal{S}[T]$; but, for any subset of $\mathcal{Q}$ of size $6 - 3 = 3$, there are at least two phylogenetic X-trees concordant with the resulting collection. Finally, the dyadic closure $cl_2(\mathcal{Q})$ equals $\mathcal{Q}[T]$.

Bibliography

[BBDS] Sebastian Böcker, David Bryant, Andreas W.M. Dress, and Michael A. Steel, *Algorithmic aspects of tree amalgamation*, submitted.

[BBHR98] Adrian C. Barbrook, Norman Blake, Christopher J. Howe, and Peter Robinson, *The phylogeny of The Canterbury Tales*, Nature **394** (1998), 839.

[BD] Sebastian Böcker and Andreas W.M. Dress, *Most parsimonious quartet encodings of binary trees*, in preparation.

[BD86] Hans-Jürgen Bandelt and Andreas Dress, *Reconstructing the shape of a tree from observed dissimilarity data*, Adv. in Appl. Math. **7** (1986), no. 3, 309–343.

[BD00a] Sebastian Böcker and Andreas W.M. Dress, *A note on maximal hierarchies*, to appear in Adv. Math., 2000.

[BD00b] Sebastian Böcker and Andreas W.M. Dress, *Patchworks*, to appear in Adv. Math., 2000.

[BDS99] Sebastian Böcker, Andreas W.M. Dress, and Michael A. Steel, *Patching up X-trees*, Ann. Comb. **3** (1999), 1–12.

[BG97] Vincent Berry and Olivier Gascuel, *Inferring evolutionary trees with strong combinatorial evidence*, Computing and combinatorics (Shanghai, 1997), Springer, Berlin, 1997, pp. 111–123.

[Böca] Sebastian Böcker, *Exponentially many supertrees*, submitted.

[Böcb] Sebastian Böcker, *The Tree Joining Theorem*, in preparation.

[Bra27] H. Brandt, *Über eine Verallgemeinerung des Gruppenbegriffes*, Math. Ann. **96** (1927), 360–366.

[Bro87] Ronald Brown, *From groups to groupoids: A brief survey*, Bull. London Math. Soc. **19** (1987), no. 2, 113–134.

[Bry96] David Bryant, *Hunting for trees in binary character sets: Efficient algorithms for extraction, enumeration, and optimization*, J. Comp. Biol. **3** (1996), no. 2, 275–288.

[Bry97] David Bryant, *Building trees, hunting for trees, and comparing trees: Theory and methods in phylogenetic analysis*, Ph.D. thesis, University of Canterbury, Christchurch, New Zealand, March 1997.

[BS95] David Bryant and Michael A. Steel, *Extension operations on sets of leaf-labelled trees*, Adv. in Appl. Math. **16** (1995), no. 4, 425–453.

[Bun71] Peter Buneman, *The recovery of trees from measures of dissimilarity*, Mathematics in the Archaeological and Historical Sciences (F.R. Hodson, D.G. Kendall, and P. Tautu, eds.), Edinburgh University Press, Edinburgh, 1971, pp. 387–395.

[CSW87] Rebecca L. Cann, Mark Stoneking, and Allan C. Wilson, *Mitocondrial DNA and human evolution*, Nature **325** (1987), 31–36.

[Dek86] M.C.H. Dekker, *Reconstruction methods for derivation trees*, Master's thesis, Vrije Universiteit, Amsterdam, Netherlands, 1986.

[Die97] Reinhard Diestel, *Graph theory*, Springer-Verlag, New York, 1997, Translated from the 1996 German original.

[DMS97] Andreas W.M. Dress, Vincent Moulton, and Michael A. Steel, *Trees, taxonomy, and strongly compatible multi-state characters*, Adv. in Appl. Math. **19** (1997), no. 1, 1–30.

[ESSW97a] Péter L. Erdős, Michael A. Steel, László A. Székely, and Tandy J. Warnow, *Constructing big trees from short sequences*, Automata, languages and programming (Bologna, 1997), Springer-Verlag, Berlin, 1997, pp. 827–837.

[ESSW97b] Péter L. Erdős, Michael A. Steel, László A. Székely, and Tandy J. Warnow, *Local quartet splits of a binary tree infer all quartet splits via one dyadic inference rule*, Comput. Artif. Intell. **16** (1997), no. 2, 217–227, Special issue on parallel and distributed computing (Bratislava, 1996).

[ESSW99] Péter L. Erdős, Michael A. Steel, László A. Székely, and Tandy J. Warnow, *A few logs suffice to build (almost) all trees. I*, Random Structures Algorithms **14** (1999), no. 2, 153–184.

[HNP+98] Daniel H. Huson, Scott Nettles, Laxmi Parida, Tandy J. Warnow, and Shibu Yooseph, *The disk-covering method for tree reconstruction*, Proceedings of "Algorithms and Experiments" (ALEX 98), Trento, Italy, Feb. 9–11 (R. Battiti and A.A. Bertossi, eds.), 1998, pp. 62–75.

[HNR+98] Daniel H. Huson, Scott Nettles, Kenneth Rice, Tandy J. Warnow, and Shibu Yooseph, *Hybrid tree reconstruction methods*, Proceedings of "Workshop on Algorithm Engineering" (WAE 98), Saarbrücken, Germany, Aug. 20–22, 1998.

[HVW99] Daniel H. Huson, Lisa Vawter, and Tandy J. Warnow, *Solving large scale phylogenetic problems using DCM2*, Proceedings of "Intelligent Systems for Molecular Biology" (ISMB 99), Heidelberg, Germany, Aug. 6–10 (T. Lengauer, R. Schneider, P. Bork, D. Brutlag, J. Glasgow, H.-W. Mewes, and R. Zimmer, eds.), 1999, pp. 118–129.

[HW99] Daniel H. Huson and Tandy J. Warnow, *Obtaining highly accurate topology estimates of evolutionary trees from very short sequences*, Proceedings of "Computational Molecular Biology" (RECOMB 99), Lyon, France, Apr. 11–14, 1999.

[KLW96] Sampath K. Kannan, Eugene L. Lawler L., and Tandy J. Warnow, *Determining the evolutionary tree using experiments*, J. Algorithms **21** (1996), no. 1, 26–50.

[PT86] Judea Pearl and Michael Tarsi, *Structuring causal trees*, J. Complexity **2** (1986), no. 1, 60–77, Complexity of approximately solved problems (Morningside Heights, N.Y., 1985).

[Rob96] Peter M.W. Robinson, *Computer-assisted stemmatic analysis and 'best-text' historical editing*, Studies in Stemmatology (P.V. Reenen and M. van Mulken, eds.), John Benjamins Publishing, Amsterdam, 1996, pp. 71–103.

[RW97] Kenneth Rice and Tandy J. Warnow, *Parsimony is hard to beat!*, Computing and combinatorics (Shanghai, 1997), Springer, Berlin, 1997, pp. 124–133.

[Sch70] Ernst Schröder, *Vier combinatorische Probleme.*, Z. Math. Phys. **15** (1870), 361–376.

[SDB00] Michael A. Steel, Andreas W.M. Dress, and Sebastian Böcker, *Simple but fundamental limitations on supertree and consensus tree methods*, to appear in Syst. Biol., 2000.

[SOWH96] David L. Swofford, Gary J. Olsen, Peter J. Waddell, and David M. Hillis, *Phylogenetic inference*, Molecular Systematics (D.M. Hillis, C. Moritz, and B.K. Marble, eds.), Sinauer Associates, second ed., 1996, pp. 407–514.

[SPH98] Michael J. Sanderson, Andy Purvis, and Chris Henze, *Phylogenetic supertrees: Assembling the trees of life*, Trends In Ecology & Evolution **13** (1998), no. 3, 105–109.

[Ste92] Michael A. Steel, *The complexity of reconstructing trees from qualitative characters and subtrees*, J. Classification **9** (1992), no. 1, 91–116.

[SvH96] Korbinian Strimmer and Arndt von Haeseler, *Quartet puzzling: A quartet maximum likelihood method for reconstructing tree topologies*, Mol. Biol. Evol. **13** (1996), no. 7, 964–969.

[War97] Tandy J. Warnow, *Mathematical approaches to comparative linguistics*, Proc. Nat. Acad. Sci. U.S.A. **94** (1997), no. 13, 6585–6590.

[Wil98] Stephen J. Willson, *Measuring inconsistency in phylogenetic trees*, J. Theor. Biol. **190** (1998), no. 1, 15–36.

List of Symbols

Symbol	Page	Explanation
$\mathbb{N}$, $\mathbb{N}_0$		natural numbers (with zero)
$\mathbb{Q}$, $\mathbb{R}$		rational and real numbers
$\#X$	5	the cardinality of a set X, $\#X \in \mathbb{N}_0 \cup \{\infty\}$
$\mathcal{P}(X)$	5	the set of subsets of X
$\binom{X}{k}$	5	the set of subsets of X of cardinality k
$A \mathbin{\dot\cup} B$	5	the disjoint union of two sets A, B
$A \vartriangle B$	5	the symmetric difference of A and B, $A \vartriangle B = (A \cup B) - (A \cap B) = (A \setminus B) \mathbin{\dot\cup} (B \setminus A)$
$A \subseteq B$	5	A is a subset of or equal to B
$A \subset B$, $A \subsetneqq B$	5	A is a proper subset of B, $A \subseteq B$ and $A \neq B$
$\bigcup \mathcal{C}$, $\bigcap \mathcal{C}$	5	$\bigcup \mathcal{C} = \bigcup_{C \in \mathcal{C}} C$ and $\bigcap \mathcal{C} = \bigcap_{C \in \mathcal{C}} C$
$\mathcal{C}^X$	5	$\mathcal{C}^X = \mathcal{C} \cup \{\emptyset, X\} \cup \{\{x\} \mid x \in X\}$
$V(G)$, $E(G)$	7	the vertices/edges of a graph G
$\deg(v) = \deg_G(v)$	7	degree of a vertex v in G, number of edges of G incident with V
$\Delta = \Delta(G)$	7	maximal degree, taken over all vertices of a graph G
$v_i(p)$, $e_i(p)$	7	i-th vertex/edge of a path p
$l(p)$	7	length of a path p
$[p]$, $\langle p \rangle$	7	set of vertices/edges of a path p
$\alpha(p)$, $\omega(p)$	7	starting/end point of a path p
$\mathbb{P}(G)$	8	set of paths of a graph G
$i_0(p,q)$	8	for two paths p, q of a graph G with $\omega(p) = \alpha(q)$, $i_0(p,q)$ denotes the largest index i with $v_{l(p)-j}(p) = v_j(q)$ for all $j \in \{0, 1, \ldots, i\}$
$p \circ q$	8	concatenation of two paths p, q of a graph G with $\omega(p) = \alpha(q)$
$p \circledcirc q$	8	"disjoint" concatenation of p, q with $\omega(p) = \alpha(q)$, $\langle p \rangle \cap \langle q \rangle = \emptyset$
p^{-1}	8	the inverse of a path p, $p^{-1} = (p_{l(p)}, p_{l(p)-1}, \ldots, p_1, p_0)$
$p(u \to v)$	8	unique path from u to v in a tree
$\mathring{E} = \operatorname{Int}(T)$	8	inner edges of a tree $T = (V, E)$
$d(u,v)$	9	distance between vertices u and v
$[u,v]$, $\langle u,v \rangle$	9	set of vertices/edges of the path $p(u \to v)$
$\operatorname{med}(v_1, v_2, v_3)$	10	median of the vertices v_1, v_2, v_3

Symbol	Page	Explanation
$[U]$, $\langle U\rangle$	11	vertex/edge closure of a subset of vertices
$p(v{\to}U)$	12	the unique path from a vertex v to a vertex patch U
$d(v,U)$	13	the distance between a vertex v and a vertex patch U
$\mathrm{gate}(v{\to}U)$	13	the end point of the path from v to U
$\mathrm{edge}(v{\to}U)$	13	the last edge of the path from v to U
v_F	13	the touching point of a superpatch F
$u \approx_U v$	14	given vertices u, v and a subset of vertices U, then $u \approx_U v$ iff $[u,v] \cap U = \emptyset$ or $u = v$
$u \sim_U v$	15	given vertices u, v and a subset of edges F, then $u \sim_F v$ iff $\langle u,v\rangle \cap F = \emptyset$
$[x]_R$, Y/R		for an equivalence relation $R \subseteq X \times X$ and $x \in X$, $Y \subseteq X$ let $[x]_R$ denote the equivalence class with respect to R containing x, and Y/R the corresponding set of equivalence classes
$V(u,v)$	16	the connected component of $(V, E - \{u,v\})$ containing v
$\pi_0(G)$	43	number of connected components of a graph G
$\mathcal{S}(X)$, $\mathcal{S}_{\mathrm{part}}(X)$	18	set of (partial) splits of X
$S_2 \leq S_1$	18	given splits $S_1 = \{A_1, B_1\}$ and $S_2 = \{A_2, B_2\}$ then $S_2 \leq S_1$ iff $S_2 = \{(A_2 \cup B_2) \cap A_1, (A_2 \cup B_2) \cap B_1\}$
$\underline{S}$, $\underline{\mathcal{S}}$	18	support $\underline{S} = A \cup B$ for a split $S = \{A, B\}$ and $\underline{\mathcal{S}} = \bigcup_{S\in\mathcal{S}} \underline{S}$ for a collection of partial splits $\mathcal{S}$
$\mathcal{S}_{i,j}(X)$	18	set of partial splits $S = \{A, B\}$ of X satisfying $\{\#A, \#B\} = \{i, j\}$
$\mathcal{Q}(X)$	18	set of quartet splits of X, $\mathcal{Q}(X) = \mathcal{S}_{2,2}(X)$
$ab\vert cd$	18	the quartet split $\{\{a, b\}, \{c, d\}\}$
$x_1 \dots x_j \vert y_1 \dots y_k$	18	the split $\{\{x_1, \dots, x_j\}, \{y_1, \dots, y_k\}\}$
$S\vert_Y$	21	$S\vert_Y = \{A \cap Y, B \cap Y\}$ for a split $S = \{A, B\}$
$\mathcal{S}\vert_Y$	21	the induced split system $\mathcal{S}\vert_Y := \big\{S\vert_Y \;\big\vert\; S \in \mathcal{S}\big\} \setminus \big\{\{Y, \emptyset\}\big\}$
$S_T[e]$	18	the split of X induced by removing the edge e from the X-tree T
$\mathcal{S}[T]$	18	set of all splits of X induced by removing some edge of the X-tree T
$\mathcal{Q}[T]$	89	set of quartet splits induced by an X-tree T
T_{phyl}	18	the unique phylogenetic tree obtained from T by "pulling out" leaves
$T\vert_Y$	20	the induced Y-tree of an X-tree T for a subset $Y \subseteq X$
$T' \leq T$	19	given X-trees T, T' then $T' \leq T$ iff $\mathcal{S}[T'] \subseteq \mathcal{S}[T]$ holds; given an X-tree T and a Y-tree T' then $T' \leq T$ iff T displays T', that is $Y \subseteq X$ and $T' \leq T\vert_Y$
$x \approx_F y$	23	given an X-tree T, a subset $F \subseteq \mathrm{Int}(T)$, and $x, y \in X$ then $x \approx_F y$ iff $\phi(x) \approx_F \phi(y)$ iff $[\phi(x), \phi(y)]_T \cap \bigcup F = \emptyset$

Symbol	Page	Explanation
T_F	24,25	the induced X_F-tree of an X-tree T for a non-empty subset $F \subseteq \mathring{E}$
$T_{(F,Y)}$	24,25	the induced Y-tree of an X-tree T for non-empty subsets $F \subseteq \mathring{E}$, $Y \subseteq X$
$\mathcal{T}_X(\mathcal{S}) = \mathcal{T}(\mathcal{S})$	28	the set of (isomorphism classes of) phylogenetic X-trees concordant with $\mathcal{S} \subseteq \mathcal{S}_{\text{part}}(X)$
$\mathcal{T}(\mathcal{F})$	29	the set of (isomorphism classes of) phylogenetic X-trees such that each $T \in \mathcal{T}(\mathcal{F})$ displays every tree in $\mathcal{F}$
$\mathcal{Q}(\mathcal{S})$	29	the set of quartet splits $Q \in \mathcal{Q}(X)$ with $Q \leq S$ for some $S \in \mathcal{S}$, for $\mathcal{S} \subseteq \mathcal{S}_{\text{part}}(X)$
$(X, \mathcal{C})$, $(I, \mathcal{I})$	36,39	cluster systems $\mathcal{C} \subseteq \mathcal{P}(X)$ or $\mathcal{I} \subseteq \mathcal{P}(I)$
$\min(\mathcal{C})$, $\max(\mathcal{C})$	36	minimal/maximal elements of $\mathcal{C} \subseteq \mathcal{P}(X)$ due to inclusion
$\min^*(\mathcal{C})$	36	minimal elements of $\mathcal{C} \setminus \{\emptyset\}$
$\mathcal{C}_{\subseteq Y}$, $\mathcal{C}_{\subset Y}$	36	the set of clusters from $\mathcal{C}$ that are (proper) subsets of Y
$\mathcal{C}_{\supseteq Y}$, $\mathcal{C}_{\supset Y}$	36	the set of clusters from $\mathcal{C}$ that are (proper) supersets of Y
$C_J = C_{\mathcal{F},J}$	39	$C_J = \bigcup\{C_i \mid i \in J\}$ for the I-indexed family $\mathcal{F} = (C_i)_{i \in I}$
$I(C) = I_{\mathcal{F}}(C)$	39	$I(C) = \{i \in I \mid C \cap C_i \neq \emptyset\}$ for the I-indexed family $\mathcal{F} = (C_i)_{i \in I}$
q	57	given a phylogenetic X-tree T, a map $q : \text{Int}(T) \to \mathcal{Q}(X)$ with $q(e) \leq S_T[e]$ is a quartet encoding of T
$\underline{q}(e)$, $\underline{q}(F)$	57	$\underline{q}(e) = \underline{q(e)}$ for $e \in \mathring{E}$ and $\underline{q}(F) = \bigcup_{e \in F} \underline{q}(e)$ for $F \subseteq \mathring{E}$
$\text{exc}(F) = \text{exc}_q(F)$	58	q-excess of F, $\text{exc}(F) = \#\underline{q}(F) - \#F - 3$
$\mathcal{C}(q)$	58	given a quartet encoding q of a phylogenetic X-tree T, $\mathcal{C}(q)$ denotes the set of subsets $F \subseteq \text{Int}(T)$ with $\text{exc}_q(F) = 0$
q_F	62	given a tight quartet encoding q of a phylogenetic X-tree T and a patch $F \subseteq \text{Int}(F)$, then $q_F : F \to \mathcal{Q}(X_F)$ is the induced quartet encoding of T_F
$T_{\{A,B\}}$	72	$(A \cup B)$-tree inducing exactly one non-trivial split $\{A, B\}$
$\text{exc}(\mathcal{Q})$	77	excess of a collection of quartet splits $\mathcal{Q}$, $\text{exc}(\mathcal{Q}) = \#\underline{\mathcal{Q}} - \#\mathcal{Q} - 3$
$\mathcal{C}(\mathcal{Q})$	77	collection of excess-free subsets $\mathcal{Q}'$ of $\mathcal{Q}$
$\mathcal{L}(T)$	83	label set of a labeled tree, $\mathcal{L}(T) = X$ for an X-tree T
$\mathcal{L}(\mathcal{T})$	83	label set of a collection of labeled trees, $\mathcal{L}(\mathcal{T}) = \bigcup\{\mathcal{L}(T) \mid T \in \mathcal{T}\}$
$\text{exc}(\mathcal{T})$	83	excess of a collection of binary labeled trees
$\mathcal{C}(\mathcal{T})$	83	collection of excess-free subsets $\mathcal{T}'$ of $\mathcal{T}$
$cl_2(\mathcal{Q})$, $scl_2(\mathcal{Q})$	88	(semi-)dyadic closure of $\mathcal{Q}$

Index