Chapter 2
Provably Robust Simplification of Component Trees of Multidimensional Images

Gabor T. Herman, T. Yung Kong, and Lucas M. Oliveira

Abstract We are interested in translating n-dimensional arrays of real numbers (*image*s) into simpler structures that nevertheless capture the topological/geometrical essence of the objects in the images. In the case $n = 3$ these structures may be used as descriptors of images in macromolecular databases. A *foreground component tree structure* (*FCTS*) contains all the information on the relationships between connected components when the image is thresholded at various levels. But unsimplified FCTSs are too sensitive to errors in the image to be good descriptors. This chapter presents a method of simplifying FCTSs which can be proved to be robust in the sense of producing essentially the same simplifications in the presence of small perturbations. We demonstrate the potential applicability of our methodology to macromolecular databases by showing that the simplified FCTSs can be used to distinguish between two slightly different versions of an adenovirus.

2.1 Introduction

High-level structural information about macromolecules is now being organized into databases. These include EM maps (three-dimensional grayscale image arrays obtained by reconstruction from electron microscopic data) of macromolecular structures. The large size of these image arrays, the arbitrary position and orientation of the macromolecule in the array, and the possibility of non-linear stretching of the

G.T. Herman (✉) · L.M. Oliveira
Computer Science Ph.D. Program, Graduate Center, City University of New York,
365 Fifth Avenue, New York, NY 10016, USA
e-mail: gabortherman@yahoo.com

L.M. Oliveira
e-mail: lmoliveira@gmail.com

T.Y. Kong
Computer Science Department, Queens College, City University of New York,
65-30 Kissena Boulevard, Flushing, NY 11367, USA
e-mail: ykong@cs.qc.cuny.edu

V.E. Brimkov, R.P. Barneva (eds.), *Digital Geometry Algorithms*,
Lecture Notes in Computational Vision and Biomechanics 2,
DOI 10.1007/978-94-007-4174-4_2,

range make standard methods of comparison between database entries infeasible. There is a need for simple robust descriptors that capture the topological/geometrical essence of the macromolecules in the images. We believe that appropriately simplified foreground component tree structures may be suitable for this purpose.

Foreground component trees are well known representations of grayscale images. Given a grayscale image $I : \mathcal{S} \to \mathbb{R}$ whose domain $\mathcal{S}$ is connected, the foreground component tree of I is a rooted tree whose nodes are the connected components of superlevel sets of I. These nodes have sometimes been called *maximum intensity extremal regions* [6]. A node $\mathbf{c}'$ is an ancestor in the tree of a node $\mathbf{c}$ if and only if $\mathbf{c}' \supseteq \mathbf{c}$. The tree can be efficiently constructed using an algorithm which processes the elements of $\mathcal{S}$ in decreasing order of their graylevels and uses Tarjan's union-find algorithm [11] to build the tree from the bottom up. For details, see [1, Alg. 4.1] or [7, Alg. 2]. The latter paper also describes applications of foreground component trees to image processing and gives a bibliography of some relevant literature.

Two related representations of images (contour trees and 0th persistence diagrams) will be described in Sect. 2.7 when we discuss research problems suggested by our work.

Unsimplified foreground component trees are too sensitive to errors in the image to be good descriptors. Accordingly, this chapter presents a new three-step method of simplifying these trees that is provably robust, in the sense that the method produces essentially the same simplified trees when the image is slightly perturbed. This property of our method is precisely stated in our main result, Theorem 1.

Methods of simplifying component trees to suppress features that are likely due to noise or artifacts have previously been considered (see, e.g., [7, 10]). But we are not aware of any previous work in which a tree simplification method has been proved to have a robustness property of the kind stated in Theorem 1.

We believe that the simplified trees produced by our method will be useful image descriptors for the identification and classification of macromolecules. As evidence of this we provide a sample biological application in which they are used to differentiate two versions of an adenovirus.

2.2 Foreground Component Tree Structures (FCTSs)

We use the term *adjacency relation* to mean an irreflexive symmetric binary relation (i.e., a set κ of ordered pairs such that if $(a, b) \in \kappa$ then $a \neq b$ and $(b, a) \in \kappa$). The members of the pairs that belong to any adjacency relation we are using will be called *spels*. (As in, e.g., [5], "spel" is an abbreviation of "spatial element", and we think of spels as generalizations of pixels and voxels.) We use the term *grayscale image* or, more briefly, the term *image*, to mean a real-valued function whose domain is a nonempty set of spels. If $I : \mathcal{S} \to \mathbb{R}$ is any image then for any $s \in \mathcal{S}$ we may refer to the real value $I(s)$ as the *graylevel* of s in I.

In the practical work described in Sect. 2.6, we use the "6-adjacency" relation [5, p. 16] on $\mathbb{Z}^3$ as our adjacency relation, and use grayscale images whose domain is the finite set $\{(x, y, z) \in \mathbb{Z}^3 \mid 0 \leq x \leq 274,\ 0 \leq y \leq 274,\ 0 \leq z \leq 274\}$.

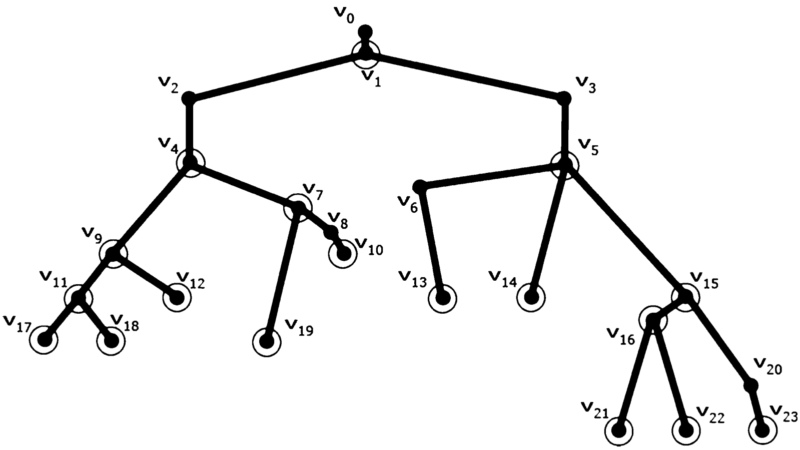


Fig. 2.1 A rooted tree in which the critical nodes have been circled

Let κ be an adjacency relation. We say that two disjoint sets of spels $\mathcal{S}_1$ and $\mathcal{S}_2$ are *κ-adjacent* if there exist $s_1 \in \mathcal{S}_1$ and $s_2 \in \mathcal{S}_2$ such that $(s_1, s_2) \in \kappa$. We call a sequence $s_0, \ldots, s_l$ of $l + 1$ spels a *κ-path* if $l = 0$ or if $l \geq 1$ and $(s_i, s_{i+1}) \in \kappa$ for $0 \leq i < l$. We say that a set $\mathcal{S}$ is *κ-connected* if for all $s, s' \in \mathcal{S}$ there exists a κ-path $s_0, \ldots, s_l$ such that $s_0 = s$, $s_l = s'$, and $s_i \in \mathcal{S}$ for $0 \leq i \leq l$.

Let $I : \mathcal{S} \to \mathbb{R}$ be any image, let $\tau \in \mathbb{R}$, and let $s \in \mathcal{S}$. Then $\mathcal{C}_\kappa(s, I, \tau)$ will denote the set of all $s' \in \mathcal{S}$ for which there exists a κ-path $s_0, \ldots, s_l$ such that $s_0 = s$, $s_l = s'$, and $I(s_i) \geq \tau$ for $0 \leq i \leq l$. Note that $\mathcal{C}_\kappa(s, I, \tau) = \emptyset$ if $\tau > I(s)$, and $s \in \mathcal{C}_\kappa(s, I, \tau)$ if $\tau \leq I(s)$. We write $\mathcal{C}_\kappa(s, I)$ to denote the set $\mathcal{C}_\kappa(s, I, I(s))$. Readily, if $t \in \mathcal{C}_\kappa(s, I)$, then $I(t) \geq I(s)$ and either $\mathcal{C}_\kappa(t, I) = \mathcal{C}_\kappa(s, I)$ or $\mathcal{C}_\kappa(t, I) \subsetneq \mathcal{C}_\kappa(s, I)$ according to whether $I(t) = I(s)$ or $I(t) > I(s)$.

We assume the reader is familiar with the concept of a rooted tree (as defined in, e.g., [3, Appendix B.5.2]). Let $\mathcal{T}$ be any rooted tree. We write $\mathbf{Nodes}(\mathcal{T})$ to denote the (finite) set of all nodes of $\mathcal{T}$, write $\mathbf{root}(\mathcal{T})$ to denote the root of $\mathcal{T}$, and write $\mathbf{Leaves}(\mathcal{T})$ to denote the set of all leaves of $\mathcal{T}$.

Recall that if $\mathbf{u} \in \mathbf{Nodes}(\mathcal{T})$ and $\mathbf{v}$ is a node of the subtree of $\mathcal{T}$ that is rooted at $\mathbf{u}$, then $\mathbf{u}$ is said to be an *ancestor* of $\mathbf{v}$ in $\mathcal{T}$, and $\mathbf{v}$ a *descendant* of $\mathbf{u}$ in $\mathcal{T}$. We write $\mathbf{u} \preceq_{\mathcal{T}} \mathbf{v}$ or $\mathbf{v} \succeq_{\mathcal{T}} \mathbf{u}$ to mean that $\mathbf{u}, \mathbf{v} \in \mathbf{Nodes}(\mathcal{T})$ and $\mathbf{u}$ is an ancestor of $\mathbf{v}$ in $\mathcal{T}$. We write $\mathbf{u} \prec_{\mathcal{T}} \mathbf{v}$ or $\mathbf{v} \succ_{\mathcal{T}} \mathbf{u}$ to mean that $\mathbf{u} \preceq_{\mathcal{T}} \mathbf{v}$ but $\mathbf{u} \neq \mathbf{v}$. If $\mathbf{u} \prec_{\mathcal{T}} \mathbf{v}$ then $\mathbf{u}$ is said to be a *proper* ancestor of $\mathbf{v}$ in $\mathcal{T}$, and $\mathbf{v}$ a *proper* descendant of $\mathbf{u}$ in $\mathcal{T}$.

For $\mathbf{v} \in \mathbf{Nodes}(\mathcal{T})$, we write $\mathbf{Children}_{\mathcal{T}}(\mathbf{v})$ to denote the set of all the children of $\mathbf{v}$ in $\mathcal{T}$, and if $\mathbf{v} \neq \mathbf{root}(\mathcal{T})$ then we write $\mathbf{parent}_{\mathcal{T}}(\mathbf{v})$ to denote the parent of $\mathbf{v}$ in $\mathcal{T}$. A node $\mathbf{v}$ of $\mathcal{T}$ is said to be *critical* if $|\mathbf{Children}_{\mathcal{T}}(\mathbf{v})| \neq 1$; thus $\mathbf{v}$ is a critical node if and only if either $\mathbf{v} \in \mathbf{Leaves}(\mathcal{T})$ or $|\mathbf{Children}_{\mathcal{T}}(\mathbf{v})| \geq 2$. In Fig. 2.1, the critical nodes are circled.

Let κ be any adjacency relation. Then a *κ-foreground component tree structure* or *κ-FCTS* is a pair $(\mathcal{T}, \ell)$ for which there exists a collection $\mathcal{C}$ of nonempty finite κ-connected sets of spels such that the following four conditions hold:

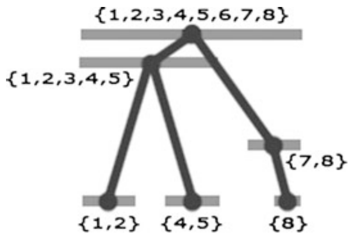


Fig. 2.2 The tree of the FCTS that is defined in Example 1

1. $\bigcup \mathcal{C} \in \mathcal{C}$
2. For all $\mathbf{u}, \mathbf{v} \in \mathcal{C}$, if $\mathbf{u} \not\supseteq \mathbf{v}$ and $\mathbf{v} \not\supseteq \mathbf{u}$ then the sets $\mathbf{u}$ and $\mathbf{v}$ are disjoint and are not κ-adjacent.
3. ℓ is a real-valued function on $\mathcal{C}$ such that, for all $\mathbf{u}, \mathbf{v} \in \mathcal{C}$, $\ell(\mathbf{u}) < \ell(\mathbf{v})$ whenever $\mathbf{u} \supsetneq \mathbf{v}$. (For each $\mathbf{v} \in \mathcal{C}$ we call $\ell(\mathbf{v})$ the *level* of $\mathbf{v}$.)
4. $\mathscr{T}$ is the rooted tree such that $\mathbf{Nodes}(\mathscr{T}) = \mathcal{C}$ and, for all $\mathbf{u}, \mathbf{v} \in \mathcal{C}$, $\mathbf{u} \prec_{\mathscr{T}} \mathbf{v}$ if and only if $\mathbf{u} \supsetneq \mathbf{v}$.

Condition 1 is equivalent to the condition that $\mathcal{C}$ have an element which is a superset of every element of $\mathcal{C}$. Moreover, since every element of $\mathcal{C}$ is required to be a nonempty finite κ-connected set, condition 1 implies that $\bigcup \mathcal{C}$ is a finite κ-connected set. Since $\bigcup \mathcal{C}$ is finite, $\mathcal{C}$ can only be a finite collection.

If $\mathcal{C}$ is *any* collection of nonempty finite κ-connected sets that satisfies conditions 1 and 2, and ℓ any function that satisfies condition 3, then there will exist a unique rooted tree $\mathscr{T}$ that satisfies condition 4 (so that $(\mathscr{T}, \ell)$ is a κ-FCTS); the root of this tree will be $\bigcup \mathcal{C}$.

Example 1 Let κ be the adjacency relation on the integers such that $(n_1, n_2) \in \kappa$ if and only if $|n_1 - n_2| = 1$. Let $\mathcal{C}$ be the following collection of six sets: $\{\{1,2,3,4,5,6,7,8\}, \{1,2,3,4,5\}, \{1,2\}, \{4,5\}, \{7,8\}, \{8\}\}$. Then it is readily confirmed that $\mathcal{C}$ satisfies conditions 1 and 2. Now let $\ell : \mathcal{C} \to \mathbb{R}$ be defined by $\ell(\{1,2,3,4,5,6,7,8\}) = 12$, $\ell(\{1,2,3,4,5\}) = 13$, $\ell(\{7,8\}) = 16$, and $\ell(\{1,2\}) = \ell(\{4,5\}) = \ell(\{8\}) = 18$. Then it is readily confirmed that ℓ satisfies condition 3. Thus there is a κ-FCTS $(\mathscr{T}, \ell)$ for which $\mathbf{Nodes}(\mathscr{T}) = \mathcal{C}$. The tree $\mathscr{T}$ of this κ-FCTS is shown in Fig. 2.2.

If $\mathfrak{F}$ is a κ-FCTS $(\mathscr{T}, \ell)$, then we may use $\mathfrak{F}$ to mean the rooted tree $\mathscr{T}$ in our terminology and notation. As examples of this, nodes and edges of $\mathscr{T}$ may be referred to as nodes and edges of $\mathfrak{F}$, the notations $\mathbf{Nodes}(\mathfrak{F})$, $\mathbf{root}(\mathfrak{F})$, and $\mathbf{Leaves}(\mathfrak{F})$ will have the same meanings as $\mathbf{Nodes}(\mathscr{T})$, $\mathbf{root}(\mathscr{T})$, and $\mathbf{Leaves}(\mathscr{T})$, and $\mathbf{parent}_{\mathfrak{F}}(\mathbf{v})$ will have the same meaning as $\mathbf{parent}_{\mathscr{T}}(\mathbf{v})$ for any $\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}) \setminus \mathbf{root}(\mathscr{T})$.

Let $\mathcal{S}$ be any nonempty finite κ-connected set of spels. Then we associate each image $I : \mathcal{S} \to \mathbb{R}$ with the κ-foreground component tree structure $\mathbf{FCTS}_\kappa(I)$ that is defined by $\mathbf{FCTS}_\kappa(I) = (\mathscr{T}_I, \ell_I)$, where:

(i) $\mathbf{Nodes}(\mathscr{T}_I) = \{\mathcal{C}_\kappa(s, I) \mid s \in \mathcal{S}\}$ and, for all $\mathbf{u}, \mathbf{v} \in \mathbf{Nodes}(\mathscr{T}_I)$, we have that $\mathbf{u} \preceq_{\mathscr{T}_I} \mathbf{v}$ if and only if $\mathbf{u} \supseteq \mathbf{v}$.

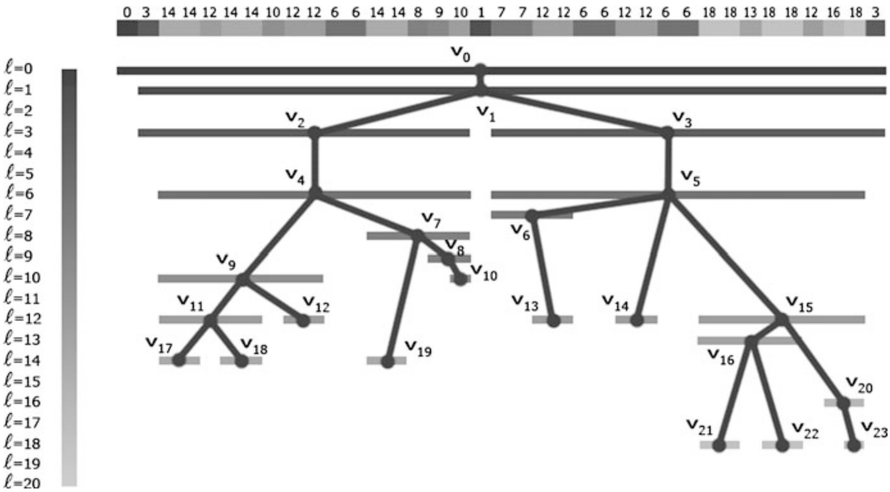


Fig. 2.3 A grayscale image whose domain is a row of 37 pixels is shown at the *top*. Writing I to denote this image, the *numbers* above the image show the graylevel $I(p)$ of each pixel p in the domain; for example, the graylevels of the first, second, third, and fourth pixels on the left are respectively 0, 3, 14, and 14. The κ-FCTS of the image (i.e., $\mathbf{FCTS}_\kappa(I)$) is shown below the image. Here κ is the adjacency relation such that $(p_1, p_2) \in \kappa$ just if p_1 and p_2 are pixels that share an edge. Writing $(\mathscr{T}, \ell)$ for this κ-FCTS, each node of the tree $\mathscr{T}$ is a κ-connected set of pixels whose elements are indicated in the figure by the *horizontal bar* which runs through that node. For example, the root node v_0 of $\mathscr{T}$ consists of all 37 pixels in the domain, the node v_1 consists of all pixels in the domain except the leftmost, and the leaf node v_{17} consists of just the third and the fourth pixels from the left. For each node v, the value of $\ell(\mathsf{v})$ can be read from the *vertical bar* on the left. For example, $\ell(\mathsf{v}_2) = \ell(\mathsf{v}_3) = 3$ and $\ell(\mathsf{v}_4) = \ell(\mathsf{v}_5) = 6$

(ii) For all $s \in \mathcal{S}$, we have that $\ell_I(\mathcal{C}_\kappa(s, I)) = I(s)$. ($\ell_I$ is well defined by this condition, because $I(s) = I(s')$ whenever $\mathcal{C}_\kappa(s, I) = \mathcal{C}_\kappa(s', I)$.)

It is readily confirmed that a κ-FCTS with these two properties exists, because $\mathcal{C} = \{\mathcal{C}_\kappa(s, I) \mid s \in \mathcal{S}\}$ satisfies conditions 1 and 2 in the definition of a κ-FCTS; the root of the tree of this FCTS is $\bigcup \mathcal{C} = \mathcal{S}$. It follows from (ii) that for each $\mathbf{v} \in \mathbf{Leaves}(\mathscr{T}_I)$ the level of $\mathbf{v}$ in $\mathbf{FCTS}_\kappa(I)$ is just the graylevel in I of each spel in $\mathbf{v}$, and that for each $\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}_I)$ the level of $\mathbf{v}$ is just the minimum of the graylevels of the spels in $\mathbf{v}$. We call $\mathbf{FCTS}_\kappa(I)$ the *κ-FCTS of* the image I. Figure 2.3 illustrates this concept.

Conversely, we associate each κ-FCTS $\mathfrak{F} = (\mathscr{T}, \ell)$ with the image $I_{\mathfrak{F}}$ that we now define. For each spel $s \in \mathbf{root}(\mathscr{T})$, conditions 2 and 4 in the definition of a κ-FCTS imply that, among the elements of $\mathbf{Nodes}(\mathscr{T})$ that contain s, there must be a smallest (i.e., a node that is a descendant in $\mathscr{T}$ of every node that contains s); that element will be denoted by $\mathbf{node}_{\mathscr{T}}(s)$. We define $I_{\mathfrak{F}} = I_{(\mathscr{T}, \ell)}$ to be the image whose domain is $\mathbf{root}(\mathscr{T})$, and which satisfies $I_{\mathfrak{F}}(s) = \ell(\mathbf{node}_{\mathscr{T}}(s))$ for all $s \in \mathbf{root}(\mathscr{T})$. We also call $I_{\mathfrak{F}}$ the *image of* the κ-FCTS $\mathfrak{F}$.

Readily, $I_{\mathbf{FCTS}_\kappa(I)} = I$ for any image I whose domain is finite and κ-connected, and $\mathbf{FCTS}_\kappa(I_{\mathfrak{F}}) = \mathfrak{F}$ for every κ-FCTS $\mathfrak{F}$. Thus the maps $I \mapsto \mathbf{FCTS}_\kappa(I)$ and

$\mathfrak{F} \mapsto I_{\mathfrak{F}}$ are mutually inverse bijections between the set of all images with finite κ-connected domains and the set of all κ-FCTSs.

Consequently, a figure (such as Fig. 2.3) that shows an image I and its associated κ-FCTS $\mathbf{FCTS}_{\kappa}(I)$ can also be construed as showing the κ-FCTS $\mathfrak{F} = \mathbf{FCTS}_{\kappa}(I)$ and its associated image $I_{\mathfrak{F}} = I$.

2.3 The (λ, k)-Simplification of a κ-FCTS, Essential Isomorphism, and the Main Theorem

As mentioned earlier, the foreground component tree structure $\mathbf{FCTS}_{\kappa}(I)$ is too sensitive to errors in the image I to be a good descriptor. In this section we propose a method of simplifying $\mathbf{FCTS}_{\kappa}(I)$ that is provably robust, in the sense that the simplified κ-FCTS of I remains essentially the same when I is slightly perturbed. We begin by defining some further terminology and notation.

Let $\mathscr{T}$ be any rooted tree, and $\mathfrak{F} = (\mathscr{T}, \ell)$ a κ-FCTS. Then the set of all critical nodes of $\mathscr{T}$ will be denoted by $\mathbf{Crit}(\mathscr{T})$ or $\mathbf{Crit}(\mathfrak{F})$. The node in $\mathbf{Crit}(\mathscr{T})$ that is an ancestor in $\mathscr{T}$ of every node in $\mathbf{Crit}(\mathscr{T})$ will be called the *lowest critical node* or **LCN** of $\mathscr{T}$ or $\mathfrak{F}$, and denoted by $\mathbf{LCN}(\mathscr{T})$ or $\mathbf{LCN}(\mathfrak{F})$.

For any subset $\mathbf{V}$ of $\mathbf{Nodes}(\mathscr{T})$ that does not contain every ancestor of $\mathbf{LCN}(\mathscr{T})$, there is a κ-FCTS $(\mathscr{T}', \ell')$ such that $\mathbf{Nodes}(\mathscr{T}') = \mathbf{Nodes}(\mathscr{T}) \setminus \mathbf{V}$ and ℓ' is the restriction of ℓ to $\mathbf{Nodes}(\mathscr{T}')$. This κ-FCTS will be denoted by $\mathfrak{F} - \mathbf{V}$.

We write $\mathfrak{F}' \sqsubseteq \mathfrak{F}$ to mean that $\mathfrak{F}' = \mathfrak{F} - \mathbf{V}$ for some $\mathbf{V} \subseteq \mathbf{Nodes}(\mathscr{T}) \setminus \{\mathbf{root}(\mathscr{T})\}$. Thus $\mathfrak{F}' \sqsubseteq \mathfrak{F}$ implies that $\mathbf{root}(\mathfrak{F}') = \mathbf{root}(\mathfrak{F})$ and that $\mathbf{Nodes}(\mathfrak{F}') \subseteq \mathbf{Nodes}(\mathfrak{F})$.

We write $\mathscr{T}^{\mathbf{crit}}$ to denote the rooted tree whose set of nodes is $\mathbf{Crit}(\mathscr{T}) \cup \{\mathbf{root}(\mathscr{T})\}$ in which a node $\mathbf{u}$ is an ancestor of a node $\mathbf{v}$ if and only if $\mathbf{u}$ is an ancestor of $\mathbf{v}$ in $\mathscr{T}$. Thus $\mathbf{root}(\mathscr{T}^{\mathbf{crit}}) = \mathbf{root}(\mathscr{T})$, $\mathbf{LCN}(\mathscr{T}^{\mathbf{crit}}) = \mathbf{LCN}(\mathscr{T})$, and $\mathbf{Crit}(\mathscr{T}^{\mathbf{crit}}) = \mathbf{Crit}(\mathscr{T})$. If $\mathbf{LCN}(\mathscr{T}) \neq \mathbf{root}(\mathscr{T})$ then $\mathbf{LCN}(\mathscr{T}^{\mathbf{crit}}) = \mathbf{LCN}(\mathscr{T})$ is the unique child of $\mathbf{root}(\mathscr{T}^{\mathbf{crit}}) = \mathbf{root}(\mathscr{T})$ in $\mathscr{T}^{\mathbf{crit}}$. The κ-FCTS $(\mathscr{T}^{\mathbf{crit}}, \ell^{\mathbf{crit}})$, where $\ell^{\mathbf{crit}}$ is the restriction of ℓ to $\mathbf{Nodes}(\mathscr{T}^{\mathbf{crit}})$, will be denoted by $\mathfrak{F}^{\mathbf{crit}}$. Note that $\mathfrak{F}^{\mathbf{crit}} \sqsubseteq \mathfrak{F}$. This concept is illustrated in Figs. 2.4 and 2.6.

Using this terminology, our method of simplifying $\mathbf{FCTS}_{\kappa}(I)$ can be stated as follows:

Let $\mathfrak{F}_0 = (\mathscr{T}_0, \ell_0)$ be any κ-FCTS. Then, for every positive real value λ and every nonnegative integer $k < |\mathbf{root}(\mathscr{T}_0)|$, we define the (λ, k)-*simplification* of $\mathfrak{F}_0$ to be the κ-FCTS $\mathfrak{F}_3$ that can be obtained from $\mathfrak{F}_0$ in three steps, as follows:

Step 1: Prune $\mathfrak{F}_0$ by removing nodes of size $\leq k$, to produce $\mathfrak{F}_1 \sqsubseteq \mathfrak{F}_0$.
Step 2: Prune $\mathfrak{F}_1$ by removing branches of length $\leq \lambda$, to produce $\mathfrak{F}_2 \sqsubseteq \mathfrak{F}_1$.
Step 3: Eliminate internal edges of length $\leq \lambda$ from $\mathfrak{F}_2^{\mathbf{crit}}$, to produce the final κ-FCTS $\mathfrak{F}_3 \sqsubseteq \mathfrak{F}_2^{\mathbf{crit}}$.

With the possible exception of the root, every non-leaf node of the final κ-FCTS $\mathfrak{F}_3$ is a critical node both of $\mathfrak{F}_3$ and of the original κ-FCTS $\mathfrak{F}_0$.

Step 1 is one of the filtering methods proposed in Sect. VI of [7]. It is defined as follows: The result of pruning the κ-FCTS $\mathfrak{F}_0 = (\mathscr{T}_0, \ell_0)$ by removing nodes of

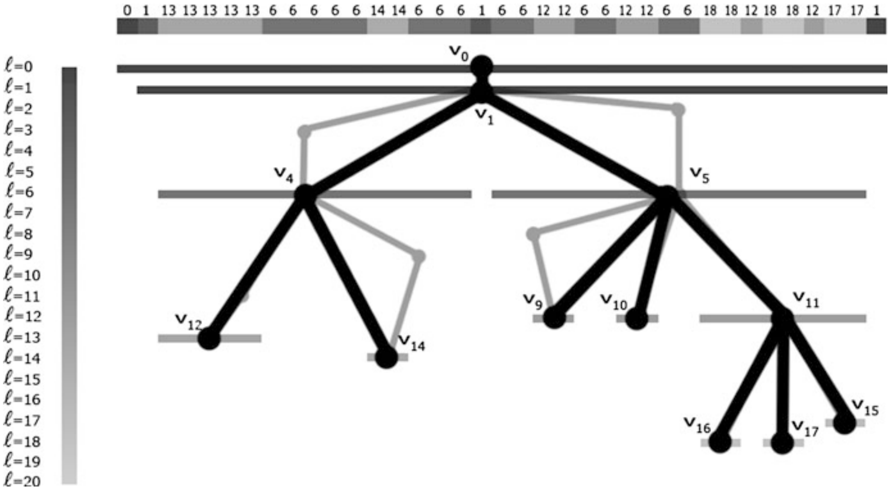


Fig. 2.4 The *thick black edges* are the edges of the FCTS $\mathfrak{F}^{\mathbf{crit}}$, where $\mathfrak{F}$ is the FCTS that is shown in Fig. 2.6 below. Nodes and edges of $\mathfrak{F}$ are shown in *gray*, but may be hidden by nodes and edges of $\mathfrak{F}^{\mathbf{crit}}$—for example, the edges of $\mathfrak{F}$ that join v_4 to v_8 and v_8 to v_{12} in Fig. 2.6 are not visible in this figure because they are hidden by the edge of $\mathfrak{F}^{\mathbf{crit}}$ that joins v_4 to v_{12}. (The image $I_{\mathfrak{F}^{\mathrm{crit}}}$ of $\mathfrak{F}^{\mathbf{crit}}$ is shown at the *top*)

size $\leq k$ is just the κ-FCTS

$$\mathfrak{F}_0 - \left\{\mathbf{v} \in \mathbf{Nodes}(\mathcal{T}_0) \;\middle|\; |\mathbf{v}| \leq k\right\}$$

where, as usual, $|\mathbf{v}|$ denotes the cardinality of the set $\mathbf{v}$—i.e., the number of spels in $\mathbf{v}$. Note that the result is just $\mathfrak{F}_0$ itself if $k = 0$. Figure 2.5 shows an FCTS that has been obtained by pruning the FCTS of Fig. 2.3 in this way.

Precise definitions of steps 2 and 3 of (λ, k)-simplification will be given in Sects. 2.4 and 2.5 below.

While our simplification method is somewhat similar to the method of [10], it has the robustness properties that are stated in Theorem 1 and Corollary 2 below (which the method of [10] does not have). We now introduce terminology and notation that will be used to state these two results.

We say that two κ-FCTSs $\mathfrak{F}_{\mathrm{a}} = (\mathcal{T}_{\mathrm{a}}, \ell_{\mathrm{a}})$ and $\mathfrak{F}_{\mathrm{b}} = (\mathcal{T}_{\mathrm{b}}, \ell_{\mathrm{b}})$ are *essentially isomorphic* if the subtree of $\mathcal{T}_{\mathrm{a}}^{\mathbf{crit}}$ that is rooted at $\mathbf{LCN}(\mathcal{T}_{\mathrm{a}})$ is isomorphic to the subtree of $\mathcal{T}_{\mathrm{b}}^{\mathbf{crit}}$ that is rooted at $\mathbf{LCN}(\mathcal{T}_{\mathrm{b}})$. Thus $\mathfrak{F}_{\mathrm{a}}$ and $\mathfrak{F}_{\mathrm{b}}$ are essentially isomorphic if and only if there exists a mapping $\theta : \mathbf{Crit}(\mathcal{T}_{\mathrm{a}}) \to \mathbf{Crit}(\mathcal{T}_{\mathrm{b}})$ such that $\theta[\mathbf{Crit}(\mathcal{T}_{\mathrm{a}})] = \mathbf{Crit}(\mathcal{T}_{\mathrm{b}})$ and, for all $\mathbf{v}, \mathbf{v}' \in \mathbf{Crit}(\mathcal{T}_{\mathrm{a}})$, $\mathbf{v} \preceq_{\mathcal{T}_{\mathrm{a}}} \mathbf{v}'$ if and only if $\theta(\mathbf{v}) \preceq_{\mathcal{T}_{\mathrm{b}}} \theta(\mathbf{v}')$. (The latter property implies that θ is 1-to-1.) Any such θ will be called an *essential isomorphism* of $\mathfrak{F}_{\mathrm{a}}$ to $\mathfrak{F}_{\mathrm{b}}$.

Note that if the rooted trees $\mathcal{T}_{\mathrm{a}}^{\mathbf{crit}}$ and $\mathcal{T}_{\mathrm{b}}^{\mathbf{crit}}$ are isomorphic, then $\mathfrak{F}_{\mathrm{a}} = (\mathcal{T}_{\mathrm{a}}, \ell_{\mathrm{a}})$ and $\mathfrak{F}_{\mathrm{b}} = (\mathcal{T}_{\mathrm{b}}, \ell_{\mathrm{b}})$ are certainly essentially isomorphic. The converse is almost but not quite true. The only way in which $\mathfrak{F}_{\mathrm{a}} = (\mathcal{T}_{\mathrm{a}}, \ell_{\mathrm{a}})$ and $\mathfrak{F}_{\mathrm{b}} = (\mathcal{T}_{\mathrm{b}}, \ell_{\mathrm{b}})$ could be essentially isomorphic without $\mathcal{T}_{\mathrm{a}}^{\mathbf{crit}}$ and $\mathcal{T}_{\mathrm{b}}^{\mathbf{crit}}$ being isomorphic is if the root is

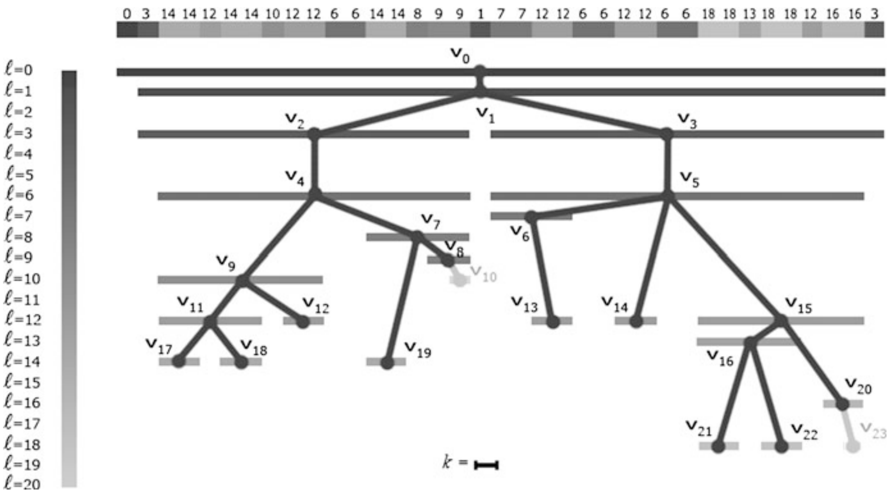


Fig. 2.5 The effect of pruning the FCTS of Fig. 2.3 by removing nodes of size $\leq k$ is shown, in the case $k = 1$; the *black edges* are the edges of the resulting FCTS. Just two nodes (v_{10} and v_{23}) are removed from the tree of Fig. 2.3, as these are the only nodes of that tree that consist of no more than k pixels (i.e., no more than 1 pixel, since $k = 1$). The image of the resulting FCTS is shown at the *top*: Note that the graylevel of the second pixel from the right has changed from 18 in Fig. 2.3 to 16 here; this reflects the removal of v_{23} from the tree. Similarly, the graylevel of the 17th pixel from the left has changed from 10 to 9; this reflects the removal of v_{10}. The graylevels of the other 35 pixels are the same as in Fig. 2.3

the same as the **LCN** in one of the trees but not in the other, and when we remove the root from the latter tree (so its **LCN** becomes its root) it becomes isomorphic to the former tree—e.g., if $\mathscr{T}_{\mathrm{a}}^{\mathbf{crit}}$ has the structure ∧ but $\mathscr{T}_{\mathrm{b}}^{\mathbf{crit}}$ has the structure ⅄.

For any $\delta \geq 0$, if an essential isomorphism θ of $\mathfrak{F}_{\mathrm{a}}$ to $\mathfrak{F}_{\mathrm{b}}$ satisfies the condition $|\ell_{\mathrm{b}}(\theta(\mathbf{x})) - \ell_{\mathrm{a}}(\mathbf{x})| \leq \delta$ for all $\mathbf{x} \in \mathbf{Crit}(\mathfrak{F}_{\mathrm{a}})$, then we say that θ is *level-preserving to within* δ. Evidently, the inverse of any essential isomorphism of $\mathfrak{F}_{\mathrm{a}}$ to $\mathfrak{F}_{\mathrm{b}}$ that is level-preserving to within δ will be an essential isomorphism of $\mathfrak{F}_{\mathrm{b}}$ to $\mathfrak{F}_{\mathrm{a}}$ that is level-preserving to within δ.

If an essential isomorphism θ of $\mathfrak{F}_{\mathrm{a}}$ to $\mathfrak{F}_{\mathrm{b}}$ is level-preserving to within 0 (i.e., if $\ell_{\mathrm{b}}(\theta(\mathbf{x})) = \ell_{\mathrm{a}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbf{Crit}(\mathfrak{F}_{\mathrm{a}})$), then we say that θ is *level-preserving*.

Example 2 The FCTS shown in Fig. 2.6 is essentially isomorphic to the FCTS shown by the thick black edges in Fig. 2.8. Indeed, if $(\mathscr{T}, \ell)$ is the FCTS shown in Fig. 2.6, and $(\mathscr{T}_*, \ell_*)$ is the FCTS shown by the thick black edges in Fig. 2.8, then $(\mathscr{T}, \ell)^{\mathbf{crit}}$ is the FCTS shown in Fig. 2.4, and $(\mathscr{T}_*, \ell_*)^{\mathbf{crit}} = (\mathscr{T}_*, \ell_*)$. It is evident from a quick glance at Figs. 2.4 and 2.8 that $\mathscr{T}^{\mathbf{crit}}$ is isomorphic to $\mathscr{T}_*^{\mathbf{crit}} = \mathscr{T}_*$, so that $(\mathscr{T}, \ell)$ is essentially isomorphic to $(\mathscr{T}_*, \ell_*)$, as we claimed. It is readily confirmed that the mapping $\theta : \mathbf{Crit}(\mathscr{T}) \to \mathbf{Crit}(\mathscr{T}_*)$ which respectively maps

$$\begin{array}{lll} & \mathsf{v}_1,\ \mathsf{v}_4,\ \mathsf{v}_5,\ \mathsf{v}_9,\ \mathsf{v}_{10},\ \mathsf{v}_{11},\ \mathsf{v}_{12},\ \mathsf{v}_{14},\ \mathsf{v}_{15},\ \mathsf{v}_{16},\ \mathsf{v}_{17} & \text{in Fig. 2.6 (or Fig. 2.4)} \\ \text{to} & \mathsf{v}_1,\ \mathsf{v}_4,\ \mathsf{v}_5,\ \mathsf{v}_{13},\ \mathsf{v}_{14},\ \mathsf{v}_{15},\ \mathsf{v}_{17},\ \mathsf{v}_{19},\ \mathsf{v}_{20},\ \mathsf{v}_{21},\ \mathsf{v}_{22} & \text{in Fig. 2.8} \end{array}$$

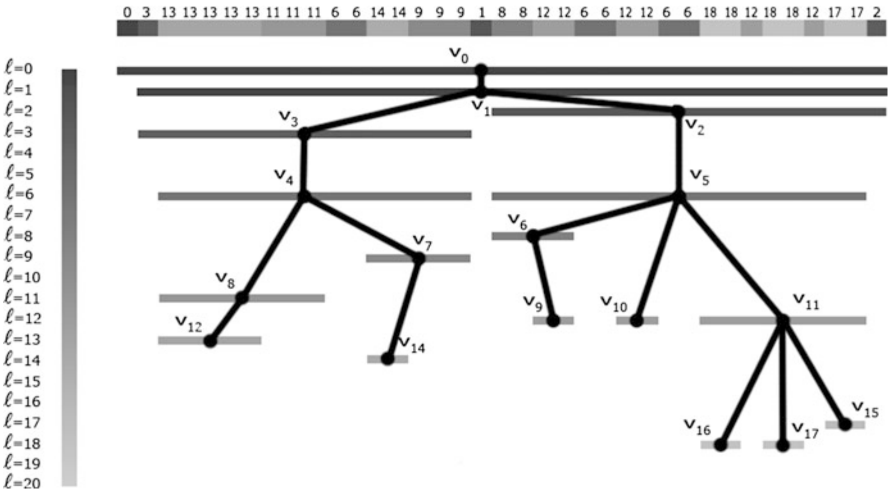


Fig. 2.6 If I is the image at the *top* (and κ is the same adjacency relation as in Figs. 2.3 and 2.5), then $\Lambda_\kappa(I) = 5$ and $K_\kappa(I) = 2$. $\Lambda_\kappa(I) = 5$ because, writing $(\mathscr{T}, \ell)$ for the κ-FCTS of I (which is shown in this figure), $\mathscr{T}$ has critical nodes v_i and v_j such that $\mathsf{v}_i \succ_{\mathscr{T}} \mathsf{v}_j$ and $\ell(\mathsf{v}_i) - \ell(\mathsf{v}_j) = 5$ (e.g., $(\mathsf{v}_i, \mathsf{v}_j) = (\mathsf{v}_4, \mathsf{v}_1)$), but $\mathscr{T}$ has no critical nodes v_i and v_j such that $\mathsf{v}_i \succ_{\mathscr{T}} \mathsf{v}_j$ and $\ell(\mathsf{v}_i) - \ell(\mathsf{v}_j) < 5$. $K_\kappa(I) = 2$ because $\mathscr{T}$ has a node (e.g., v_9) that consists of just 2 pixels, but no node of $\mathscr{T}$ consists of fewer than 2 pixels

is an essential isomorphism of $(\mathscr{T}, \ell)$ to $(\mathscr{T}_*, \ell_*)$. The essential isomorphism θ is not level-preserving, since $|\ell_*(\theta(\mathbf{x})) - \ell(\mathbf{x})| = 1$ when $\mathbf{x} = \mathsf{v}_{12}$ and when $\mathbf{x} = \mathsf{v}_{15}$; indeed, $\ell(\mathsf{v}_{12}) = 13$ but $\ell_*(\theta(\mathsf{v}_{12})) = 14$, and $\ell(\mathsf{v}_{15}) = 17$ but $\ell_*(\theta(\mathsf{v}_{15})) = 16$. But it is readily confirmed that $\ell_*(\theta(\mathbf{x})) = \ell(\mathbf{x})$ for all $\mathbf{x} \in \mathbf{Crit}(\mathscr{T}) \setminus \{\mathsf{v}_{12}, \mathsf{v}_{15}\}$, and so θ is level-preserving to within 1.

Let $I : \mathcal{S} \to \mathbb{R}$ be an image whose domain $\mathcal{S}$ is finite and κ-connected, and let $(\mathscr{T}, \ell) = \mathbf{FCTS}_\kappa(I)$. Then we define:

$$K_\kappa(I) = \min_{s \in \mathcal{S}} \left|\mathcal{C}_\kappa(s, I)\right| = \min_{\mathbf{v} \in \mathbf{Leaves}(\mathscr{T})} |\mathbf{v}|$$

$$\Lambda_\kappa(I) = \min\left\{\ell(\mathbf{u}) - \ell(\mathbf{v}) \;\middle|\; \mathbf{u}, \mathbf{v} \in \mathbf{Crit}(\mathscr{T}) \text{ and } \mathbf{u} \succ_{\mathscr{T}} \mathbf{v}\right\}$$

These concepts are illustrated in Fig. 2.6.

If $I : \mathcal{S} \to \mathbb{R}$ and $I' : \mathcal{S} \to \mathbb{R}$ are two images that have the same domain $\mathcal{S}$, then the value $\max_{s \in \mathcal{S}} |I'(s) - I(s)|$ will be denoted by $\|I' - I\|_\infty$.

Using this notation, we now state our principal robustness result regarding (λ, k)-simplification (a result which we will generalize in Corollary 2):

Theorem 1 (Main Theorem) *Let κ be any adjacency relation, $I : \mathcal{S} \to \mathbb{R}$ any image whose domain $\mathcal{S}$ is finite and κ-connected, k any integer such that $0 \le k < K_\kappa(I)$, and λ any value such that $0 < \lambda < \Lambda_\kappa(I)/2$. Let $I' : \mathcal{S} \to \mathbb{R}$ be an image such that $\|I' - I\|_\infty \le \lambda/2$. Then there is an essential isomorphism of the (λ, k)-simplification of $\mathbf{FCTS}_\kappa(I')$ to $\mathbf{FCTS}_\kappa(I)$ that is level-preserving to within $\lambda/2$.*

A proof of this theorem is given in Appendix B. In the theorem, and in Corollary 2 below, we may think of the image $I : \mathbb{S} \to \mathbb{R}$ as an ideal or perfect image of some object (such as a macromolecule) at a certain level of detail/resolution, and think of the image I' as an imperfect noisy approximation to the ideal image I (such as an EM map of the same object). We may suppose that the ideal image I is not available to us (and we do not know the exact structure of $\mathbf{FCTS}_\kappa(I)$), but the imperfect image I' is available and we can therefore construct $\mathbf{FCTS}_\kappa(I')$. Theorem 1 and Corollary 2 assure us that, if I' is "sufficiently similar" to I, then there will be values of λ and k for which the (λ, k)-simplification of $\mathbf{FCTS}_\kappa(I')$ is essentially isomorphic to $\mathbf{FCTS}_\kappa(I)$.

For this purpose it follows from Theorem 1 that the imperfect noisy approximation I' will be "sufficiently similar" to the ideal image I if there is no spel in $\mathbb{S}$ at which the value of I' differs from the value of I by $\Lambda_\kappa(I)/4$ or more. Additionally, it will follow from Corollary 2 (as we shall explain in Example 4) that I' might be sufficiently similar to I even if this condition is violated at a small number of spels whose values in I and I' may differ by arbitrarily large amounts.

Example 3 To illustrate Theorem 1, let I be the image that is shown in Fig. 2.6, and let I' be the image that is shown in Fig. 2.3. Then $\|I' - I\|_\infty = 1$, because there exists a pixel p (e.g., any of the three rightmost pixels in the domain) for which $|I'(p) - I(p)| = 1$, but there is no pixel p for which $|I'(p) - I(p)| > 1$. Now let $\lambda = 2$ and $k = 1$. As we observe in the caption of Fig. 2.6, $\Lambda_\kappa(I) = 5$ and $K_\kappa(I) = 2$, so the conditions $\lambda < \Lambda_\kappa(I)/2$, $k < K_\kappa(I)$, and $\|I' - I\|_\infty \le \lambda/2$ that appear in Theorem 1 are satisfied. Thus the theorem says that there is an essential isomorphism of the (λ, k)-simplification of $\mathbf{FCTS}_\kappa(I')$ to $\mathbf{FCTS}_\kappa(I)$ that is level-preserving to within $\lambda/2 = 1$. In fact the inverse of the mapping θ defined in Example 2 above is just such an essential isomorphism! That is because (as we will see in Sect. 2.5) the FCTS shown by the thick black edges in Fig. 2.8 is exactly the (λ, k)-simplification of $\mathbf{FCTS}_\kappa(I')$.

From Theorem 1, it is easy to deduce Corollary 2 below. Theorem 1 is essentially the case of Corollary 2 in which $k^* = 0$ and $I^* = I$.

As mentioned above, one can think of I in Theorem 1 and Corollary 2 as a perfect or ideal image, and think of I' as an imperfect approximation to I. Theorem 1 is applicable only if the graylevel of *every* spel in I' is close to (specifically, within less than $\Lambda_\kappa(I)/4$ of) that spel's graylevel in I. Corollary 2 is more general; as we will see in Example 4 below, it may be applicable even if there are exceptional spels at which I''s graylevel is much lower or higher than I's graylevel.

Corollary 2 *Let $I : \mathbb{S} \to \mathbb{R}$ and $I' : \mathbb{S} \to \mathbb{R}$ be images on the same finite κ-connected domain $\mathbb{S}$. For any nonnegative integer $k < |\mathbb{S}|$, let I'_k denote the image of the κ-FCTS that results from pruning $\mathbf{FCTS}_\kappa(I')$ by removing nodes of size $\le k$. Suppose there is an image $I^* : \mathbb{S} \to \mathbb{R}$ such that there exists a level-preserving essential isomorphism of $\mathbf{FCTS}_\kappa(I^*)$ to $\mathbf{FCTS}_\kappa(I)$, and there exists a nonnegative integer $k^* < K_\kappa(I^*)$ for which the image I' satisfies $\|I'_{k^*} - I^*\|_\infty < \Lambda_\kappa(I)/4$. Then,* *for any positive* λ *and integer* k *such that* $2\|I'_{k^*} - I^*\|_\infty \leq \lambda < \Lambda_\kappa(I)/2$ *and* $k^* \leq k < K_\kappa(I^*)$, *there is an essential isomorphism of the* (λ, k)*-simplification of* $\mathbf{FCTS}_\kappa(I')$ *to* $\mathbf{FCTS}_\kappa(I)$ *that is level-preserving to within* $\lambda/2$.

Proof of Corollary 2, assuming Theorem 1 Let k be an integer such that $k^* \leq k < K_\kappa(I^*)$, and λ a positive value such that $2\|I'_{k^*} - I^*\|_\infty \leq \lambda < \Lambda_\kappa(I)/2$.

Now $\mathbf{FCTS}_\kappa(I'_{k^*})$ is the result of applying step 1 of (λ, k^*)-simplification to $\mathbf{FCTS}_\kappa(I')$. It follows that the (λ, k)-simplification of $\mathbf{FCTS}_\kappa(I')$ is the same as the (λ, k)-simplification of $\mathbf{FCTS}_\kappa(I'_{k^*})$ (since applying simplification step 1 twice in succession with parameter k^* and then k has the same effect as applying step 1 just once with the parameter $\max(k^*, k) = k$). To prove the corollary, we need to show that there is an essential isomorphism of this κ-FCTS (i.e., the (λ, k)-simplification of $\mathbf{FCTS}_\kappa(I'_{k^*})$) to $\mathbf{FCTS}_\kappa(I)$ that is level-preserving to within $\lambda/2$.

We have that $\Lambda_\kappa(I) = \Lambda_\kappa(I^*)$, since there is a level-preserving essential isomorphism of $\mathbf{FCTS}_\kappa(I^*)$ to $\mathbf{FCTS}_\kappa(I)$. Thus we have that $\lambda < \Lambda_\kappa(I^*)/2$. Moreover, $\|I'_{k^*} - I^*\|_\infty \leq \lambda/2$ and $k < K_\kappa(I^*)$. So, on applying Theorem 1 to I^* and I'_{k^*}, we see that there is an essential isomorphism of the (λ, k)-simplification of $\mathbf{FCTS}_\kappa(I'_{k^*})$ to $\mathbf{FCTS}_\kappa(I^*)$ that is level-preserving to within $\lambda/2$. Composing this essential isomorphism with the level-preserving essential isomorphism of $\mathbf{FCTS}_\kappa(I^*)$ to $\mathbf{FCTS}_\kappa(I)$ gives an essential isomorphism of the (λ, k)-simplification of $\mathbf{FCTS}_\kappa(I'_{k^*})$ to $\mathbf{FCTS}_\kappa(I)$ that is level-preserving to within $\lambda/2$, as required. □

The following example shows how the condition that I' must satisfy in Corollary 2 is much less restrictive than the condition $\|I' - I\|_\infty < \Lambda_\kappa(I)/4$ that I' needs to satisfy for Theorem 1 to be applicable.

Example 4 Let $\mathcal{S}$ be a 3D rectangular array of voxels, and let κ be the 6-adjacency relation on $\mathcal{S}$. Let $I : \mathcal{S} \to \mathbb{R}$ be an image such that, for each threshold $\tau \leq \max_{s \in \mathcal{S}} I(s)$, the members of $\{\mathcal{C}_\kappa(s, I, \tau) \mid I(s) \geq \tau\}$ have fairly compact shapes and are not very small, and no two of the sets are very close together. (Here "have fairly compact shapes" and "are not very small" imply that: (i) removing a very few randomly chosen voxels from a set $\mathcal{C}_\kappa(s, I, \tau)$ is unlikely to split it into two or more pieces, and unlikely to completely eliminate that set. The "no two of the sets are very close" condition implies that: (ii) adding a very few randomly chosen voxels to a set $\mathcal{C}_\kappa(s, I, \tau)$ is unlikely to connect that set to a different set $\mathcal{C}_\kappa(s', I, \tau)$.) Now let I' be an image on $\mathcal{S}$ that is obtained from I by changing the graylevels of a very small number of randomly chosen voxels by arbitrarily large positive and/or negative amounts. Then $\|I' - I\|_\infty < \Lambda_\kappa(I)/4$ will *not* hold unless every graylevel change is smaller in absolute value than $\Lambda_\kappa(I)/4$. But, regardless of the sizes of the graylevel changes, when k^* is the cardinality of the largest 6-connected subset of the set $\{s \in \mathcal{S} \mid I'(s) > I(s)\}$ it is likely (because of (i) and (ii)) that there will be a level-preserving essential isomorphism of $\mathbf{FCTS}_\kappa(I'_{k^*})$ to $\mathbf{FCTS}_\kappa(I)$, in which case the image I' will satisfy the condition of Corollary 2 with $I^* = I'_{k^*}$.

2.4 Pruning by Removing Branches of Length $\leq \lambda$

Step 2 of (λ, k)-simplification is to prune the FCTS that is the result of step 1 by removing branches of length $\leq \lambda$. We now give a mathematical specification of the output of step 2 (properties P1–P4 below), present a result (Proposition 3) that gives us an easily visualized characterization of the output, and then describe (in Sect. 2.4.3) how step 2 can be efficiently implemented.

2.4.1 Specification of Simplification Step 2

Let $\mathscr{T}$ be any rooted tree and let $\mathbf{x} \in \mathbf{Nodes}(\mathscr{T})$. Then we write $\mathbf{x}{\Downarrow}_{\mathscr{T}}$ to denote the set of all ancestors of $\mathbf{x}$ in $\mathscr{T}$, write $\mathbf{x}{\downarrow}_{\mathscr{T}}$ to denote the set $\mathbf{x}{\Downarrow}_{\mathscr{T}} \setminus \{\mathbf{x}\}$ (i.e., the set of all proper ancestors of $\mathbf{x}$ in $\mathscr{T}$), write $\mathbf{x}{\Uparrow}_{\mathscr{T}}$ to denote the set of all descendants of $\mathbf{x}$ in $\mathscr{T}$, and write $\mathbf{x}{\uparrow}_{\mathscr{T}}$ to denote the set $\mathbf{x}{\Uparrow}_{\mathscr{T}} \setminus \{\mathbf{x}\}$ (i.e., the set of all proper descendants of $\mathbf{x}$ in $\mathscr{T}$).

Now let $\emptyset \neq \mathbf{S} \subseteq \mathbf{Nodes}(\mathscr{T})$. Then we write $\bigwedge_{\mathscr{T}} \mathbf{S}$ to denote the *closest common ancestor* of $\mathbf{S}$, by which we mean the node $\mathbf{v}$ of $\mathscr{T}$ such that $\mathbf{v}{\Downarrow}_{\mathscr{T}} = \bigcap_{\mathbf{u} \in \mathbf{S}} \mathbf{u}{\Downarrow}_{\mathscr{T}}$, or, equivalently, the element of $\bigcap_{\mathbf{u} \in \mathbf{S}} \mathbf{u}{\Downarrow}_{\mathscr{T}}$ that is a descendant in $\mathscr{T}$ of every element of that set.

For any κ-FCTS $\mathfrak{F}_{\text{in}} = (\mathscr{T}_{\text{in}}, \ell_{\text{in}})$, we call a sequence $\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}]$ an ℓ_{in}*-increasing enumeration* of $\mathbf{Leaves}(\mathfrak{F}_{\text{in}})$ if no two of $\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}]$ are the same, $\{\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}]\} = \mathbf{Leaves}(\mathfrak{F}_{\text{in}})$ (so that $\mathsf{n} = |\mathbf{Leaves}(\mathfrak{F}_{\text{in}})|$), and $\ell_{\text{in}}(\mathsf{leaf}[1]) \leq \cdots \leq \ell_{\text{in}}(\mathsf{leaf}[\mathsf{n}])$. Pruning a κ-FCTS $\mathfrak{F}_{\text{in}}$ by removing branches of length $\leq \lambda$ is done using such an enumeration of $\mathbf{Leaves}(\mathfrak{F}_{\text{in}})$.

For any $\lambda > 0$, any κ-FCTS $\mathfrak{F}_{\text{in}} = (\mathscr{T}_{\text{in}}, \ell_{\text{in}})$, and any ℓ_{in}-increasing enumeration $\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}]$ of $\mathbf{Leaves}(\mathfrak{F}_{\text{in}})$, we define the result of pruning $\mathfrak{F}_{\text{in}}$ by removing branches of length $\leq \lambda$ using the leaf enumeration $\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}]$ to be the κ-FCTS $\mathfrak{F}_{\text{out}}$ that has the following four properties:

P1: $\mathfrak{F}_{\text{out}} \sqsubseteq \mathfrak{F}_{\text{in}}$
P2: $\mathsf{leaf}[\mathsf{n}] \in \mathbf{Leaves}(\mathfrak{F}_{\text{out}})$
P3: For $1 \leq i < \mathsf{n}$, $\mathsf{leaf}[i] \in \mathbf{Leaves}(\mathfrak{F}_{\text{out}})$ if and only if there does *not* exist any $j \in \{i+1, \ldots, \mathsf{n}\}$ for which $\ell_{\text{in}}(\mathsf{leaf}[i]) - \ell_{\text{in}}(\bigwedge_{\mathscr{T}_{\text{in}}} \{\mathsf{leaf}[j], \mathsf{leaf}[i]\}) \leq \lambda$.
P4: $\mathbf{Nodes}(\mathfrak{F}_{\text{out}}) = \bigcup \{\mathsf{leaf}[i]{\Downarrow}_{\mathscr{T}_{\text{in}}} \mid 1 \leq i \leq \mathsf{n} \text{ and } \mathsf{leaf}[i] \in \mathbf{Leaves}(\mathfrak{F}_{\text{out}})\}$

Given any κ-FCTS $\mathfrak{F}_{\text{in}} = (\mathscr{T}_{\text{in}}, \ell_{\text{in}})$, any $\lambda > 0$, and any ℓ_{in}-increasing enumeration $\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}]$ of $\mathbf{Leaves}(\mathfrak{F}_{\text{in}})$, it is evident that P1–P4 uniquely determine $\mathfrak{F}_{\text{out}}$. Moreover, even though the result $\mathfrak{F}_{\text{out}}$ of pruning may depend on the leaf enumeration $\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}]$ that is used, we will see from Proposition 3 that, for any given $\mathfrak{F}_{\text{in}}$ and λ, P1–P4 uniquely determine $\mathfrak{F}_{\text{out}}$ *up to a level-preserving essential isomorphism.*

Figure 2.7 shows an FCTS that has been obtained by pruning the FCTS of Fig. 2.5 in this way.

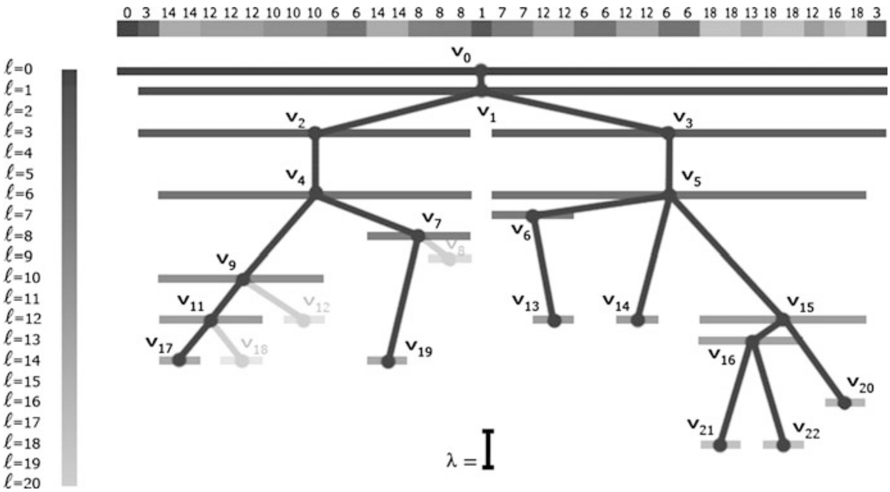


Fig. 2.7 The effect of pruning the FCTS of Fig. 2.5 by removing branches of length λ is shown, in the case $\lambda = 2$; the *black edges* are the edges of the resulting FCTS. Writing $(\mathscr{T}_1, \ell_1)$ for the FCTS of Fig. 2.5, it is assumed that pruning is done using an ℓ_1-increasing leaf enumeration in which the leaf v_{17} of $\mathscr{T}_1$ occurs later than the leaf v_{18}. The leaves v_8, v_{12}, and v_{18} are the only nodes of $\mathscr{T}_1$ that are removed; the leaf v_8 is removed because we have that $\ell_1(\mathsf{v}_8) - \ell_1(\bigwedge_{\mathscr{T}_1}\{\mathsf{v}_{19}, \mathsf{v}_8\}) = \ell_1(\mathsf{v}_8) - \ell_1(\mathsf{v}_7) \le 2 = \lambda$ (and v_{19} occurs later in the ℓ_1-increasing leaf enumeration than v_8 because $\ell_1(\mathsf{v}_8) < \ell_1(\mathsf{v}_{19})$); v_{12} is removed because $\ell_1(\mathsf{v}_{12}) - \ell_1(\bigwedge_{\mathscr{T}_1}\{\mathsf{v}_{17}, \mathsf{v}_{12}\}) = \ell_1(\mathsf{v}_{12}) - \ell_1(\mathsf{v}_9) \le 2 = \lambda$; v_{18} is removed because $\ell_1(\mathsf{v}_{18}) - \ell_1(\bigwedge_{\mathscr{T}_1}\{\mathsf{v}_{17}, \mathsf{v}_{18}\}) = \ell_1(\mathsf{v}_{18}) - \ell_1(\mathsf{v}_{11}) \le 2 = \lambda$ and we are assuming (as mentioned above) that v_{17} occurs later in the ℓ_1-increasing leaf enumeration than v_{18}. In this example no non-leaf nodes of $\mathscr{T}_1$ are removed, as every non-leaf node of $\mathscr{T}_1$ is an ancestor of a leaf of $\mathscr{T}_1$ that is not removed

2.4.2 *An Easily Visualized Characterization of the Output of Simplification Step 2*

The main goal of this section is to present a result (Proposition 3) that is important for the following reasons:

1. It shows that the output of step 2 is independent of the leaf enumeration which is used for pruning (up to a level-preserving essential isomorphism).
2. It gives an easily visualized characterization of the output. (This will be further explained after Proposition 3.)
3. The linear-time implementation of step 2 that is described in Sect. 2.4.3 is based on this result.

For any rooted tree $\mathscr{T}$ and any $\mathbf{x} \in \mathbf{Nodes}(\mathscr{T})$, we write $\mathscr{T}[\mathbf{x}]$ to denote the subtree of $\mathscr{T}$ that is rooted at $\mathbf{x}$.

Now we define some other notation that will be used in Proposition 3. For this purpose, let $\mathfrak{F} = (\mathscr{T}, \ell)$ be any κ-FCTS and λ any positive value. Then we define $\text{depth}_{\mathfrak{F}}(\mathbf{x}) = \max_{\mathbf{y} \in \mathbf{Leaves}(\mathscr{T}[\mathbf{x}])} \ell(\mathbf{y}) - \ell(\mathbf{x})$. Note that $\text{depth}_{\mathfrak{F}}(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathbf{Leaves}(\mathscr{T})$. We also define:

$$\mathbf{U}^\lambda\langle\mathfrak{F}\rangle = \left\{\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}) \,\middle|\, \mathrm{depth}_{\mathfrak{F}}(\mathbf{v}) > \lambda\right\}$$
$$\mathbf{V}^\lambda\langle\mathfrak{F}\rangle = \left\{\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}) \,\middle|\, \mathbf{v} \notin \mathbf{U}^\lambda\langle\mathfrak{F}\rangle \text{ but } \mathbf{v}{\downarrow_{\mathscr{T}}} \subseteq \mathbf{U}^\lambda\langle\mathfrak{F}\rangle\right\}$$

If $\mathbf{U}^\lambda\langle\mathfrak{F}\rangle \neq \emptyset$, then $\mathbf{v} \in \mathbf{V}^\lambda\langle\mathfrak{F}\rangle$ if and only if $\mathbf{v} \in \mathbf{root}(\mathscr{T}){\uparrow_{\mathscr{T}}}$, $\mathrm{depth}_{\mathfrak{F}}(\mathbf{v}) \leq \lambda$, and $\mathrm{depth}_{\mathfrak{F}}(\mathbf{parent}_{\mathscr{T}}(\mathbf{v})) > \lambda$. If $\mathbf{U}^\lambda\langle\mathfrak{F}\rangle = \emptyset$, then $\mathbf{V}^\lambda\langle\mathfrak{F}\rangle = \{\mathbf{root}(\mathscr{T})\}$.

For any $\mathbf{x} \in \mathbf{Nodes}(\mathscr{T})$, either $\mathbf{x} \in \mathbf{U}^\lambda\langle\mathfrak{F}\rangle$ or $\mathbf{x}$ has a unique ancestor in $\mathbf{V}^\lambda\langle\mathfrak{F}\rangle$ (possibly itself), and $\mathbf{x}$ satisfies just one of those conditions. Hence:

$$\mathbf{Nodes}(\mathscr{T}) = \mathbf{U}^\lambda\langle\mathfrak{F}\rangle \cup \bigcup_{\mathbf{v}\in\mathbf{V}^\lambda\langle\mathfrak{F}\rangle} \mathbf{v}{\Uparrow_{\mathscr{T}}} \tag{2.1}$$

If $\mathbf{U}^\lambda\langle\mathfrak{F}\rangle \neq \emptyset$ (so that $\mathbf{root}(\mathscr{T})$ lies in $\mathbf{U}^\lambda\langle\mathfrak{F}\rangle$ and not in $\mathbf{V}^\lambda\langle\mathfrak{F}\rangle$), then we define:

$$\mathbf{V}_1^\lambda\langle\mathfrak{F}\rangle = \left\{\mathbf{v} \in \mathbf{V}^\lambda\langle\mathfrak{F}\rangle \,\middle|\, \mathrm{depth}_{\mathfrak{F}}(\mathbf{v}) + \ell(\mathbf{v}) - \ell\big(\mathbf{parent}_{\mathscr{T}}(\mathbf{v})\big) > \lambda\right\}$$

But if $\mathbf{U}^\lambda\langle\mathfrak{F}\rangle = \emptyset$, then we define $\mathbf{V}_1^\lambda\langle\mathfrak{F}\rangle = \{\mathbf{root}(\mathscr{T})\} = \mathbf{V}^\lambda\langle\mathfrak{F}\rangle$.

Let $\sigma = (\mathsf{leaf}[1], \ldots, \mathsf{leaf}[n])$ be any ℓ-increasing enumeration of the leaves of the tree $\mathscr{T}$, and $\mathbf{v}$ any node of $\mathscr{T}$. Then we define $\mathsf{lastLeaf}_\sigma(\mathbf{v}, \mathscr{T})$ to be the leaf of $\mathscr{T}[\mathbf{v}]$ that occurs later in the ℓ-increasing enumeration σ than all other leaves of $\mathscr{T}[\mathbf{v}]$. (If $\mathscr{T}[\mathbf{v}]$ has just one leaf, then $\mathsf{lastLeaf}_\sigma(\mathbf{v}, \mathscr{T})$ is that leaf.) Thus we have that $\mathrm{depth}_{\mathfrak{F}}(\mathbf{v}) = \ell(\mathsf{lastLeaf}_\sigma(\mathbf{v}, \mathscr{T})) - \ell(\mathbf{v})$. We define $\mathsf{Path}_\sigma(\mathbf{v}, \mathscr{T}) = \{\mathbf{x} \in \mathbf{Nodes}(\mathscr{T}) \mid \mathbf{v} \preceq_{\mathscr{T}} \mathbf{x} \preceq_{\mathscr{T}} \mathsf{lastLeaf}_\sigma(\mathbf{v}, \mathscr{T})\}$. (Note that if $\mathbf{v}'$ is any node of $\mathscr{T}$ that is neither an ancestor nor a descendant of $\mathbf{v}$ in $\mathscr{T}$, then $\mathsf{lastLeaf}_\sigma(\mathbf{v}, \mathscr{T}) \neq \mathsf{lastLeaf}_\sigma(\mathbf{v}', \mathscr{T})$ and $\mathsf{Path}_\sigma(\mathbf{v}, \mathscr{T}) \cap \mathsf{Path}_\sigma(\mathbf{v}', \mathscr{T}) = \emptyset$.)

Using the notation we have just introduced, we now state the main result of this section, which is proved in Appendix A.

Proposition 3 *Let $\mathfrak{F}_{\mathrm{in}} = (\mathscr{T}_{\mathrm{in}}, \ell_{\mathrm{in}})$ be any κ-FCTS, let $\lambda > 0$, and let $\mathfrak{F}_{\mathrm{out}} = (\mathscr{T}_{\mathrm{out}}, \ell_{\mathrm{out}})$ be the κ-FCTS that results from pruning $\mathfrak{F}_{\mathrm{in}}$ by removing branches of length $\leq \lambda$ using an ℓ_{in}-increasing enumeration σ of* $\mathbf{Leaves}(\mathscr{T}_{\mathrm{in}})$. *Then the nodes of $\mathfrak{F}_{\mathrm{out}}$ consist just of*:

(i) *The nodes of* $\mathbf{U}^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle$.
(ii) *The nodes of* $\mathsf{Path}_\sigma(\mathbf{v}, \mathscr{T}_{\mathrm{in}})$ *for each node* $\mathbf{v}$ *in* $\mathbf{V}_1^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle$.

Now let $\mathfrak{F}_{\mathrm{in}} = (\mathscr{T}_{\mathrm{in}}, \ell_{\mathrm{in}})$, λ, σ, and $\mathfrak{F}_{\mathrm{out}} = (\mathscr{T}_{\mathrm{out}}, \ell_{\mathrm{out}})$ be as in Proposition 3. Since $\mathbf{V}_1^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle \subseteq \mathbf{V}^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle$, and since no node in $\mathbf{V}^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle$ is an ancestor in $\mathscr{T}_{\mathrm{in}}$ of a node in $\mathbf{U}^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle$ or of a different node in $\mathbf{V}^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle$, for all $\mathbf{v} \in \mathbf{V}_1^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle$ we have that $\mathsf{Path}_\sigma(\mathbf{v}, \mathscr{T}_{\mathrm{in}}) \cap \mathbf{U}^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle = \emptyset$, and for all distinct $\mathbf{v}, \mathbf{v}' \in \mathbf{V}_1^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle$ we have that $\mathsf{Path}_\sigma(\mathbf{v}, \mathscr{T}_{\mathrm{in}}) \cap \mathsf{Path}_\sigma(\mathbf{v}', \mathscr{T}_{\mathrm{in}}) = \emptyset$.

Thus Proposition 3 gives us an easily visualized characterization of the nodes of the FCTS $\mathfrak{F}_{\mathrm{out}} = (\mathscr{T}_{\mathrm{out}}, \ell_{\mathrm{out}})$ that results from pruning $\mathfrak{F}_{\mathrm{in}}$ by removing branches of length $\leq \lambda$ using the leaf enumeration σ (and hence an easily visualized characterization of $\mathfrak{F}_{\mathrm{out}}$ itself, since $\mathfrak{F}_{\mathrm{out}} \sqsubseteq \mathfrak{F}_{\mathrm{in}}$).

In Proposition 3, $\mathbf{U}^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle$ and $\mathbf{V}_1^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle$ are determined by $\mathfrak{F}_{\mathrm{in}}$ and λ; they do not depend on σ. For any $\mathbf{v}$ in $\mathbf{V}_1^\lambda\langle\mathfrak{F}_{\mathrm{in}}\rangle$, the difference in level between $\mathbf{v}$ and the leaf node of $\mathsf{Path}_\sigma(\mathbf{v}, \mathscr{T}_{\mathrm{in}})$—i.e., the value of $\ell_{\mathrm{out}}(\mathsf{lastLeaf}_\sigma(\mathbf{v}, \mathscr{T}_{\mathrm{in}})) - \ell_{\mathrm{out}}(\mathbf{v}) =$

$\ell_{\text{in}}(\mathsf{lastLeaf}_\sigma(\mathbf{v}, \mathscr{T}_{\text{in}})) - \ell_{\text{in}}(\mathbf{v}) = \text{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{v})$—also does not depend on σ. So even though the sets $\mathsf{Path}_\sigma(\mathbf{v}, \mathscr{T}_{\text{in}})$ may depend on the leaf enumeration σ, we see from Proposition 3 that $\mathfrak{F}_{\text{out}}$ is uniquely determined by $\mathfrak{F}_{\text{in}}$ and λ up to a level-preserving essential isomorphism.

2.4.3 Linear-Time Implementation of Simplification Step 2

In the rest of this chapter we assume that each FCTS $(\mathscr{T}, \ell)$ we use is represented in such a way that we can find the root of $\mathscr{T}$ in $O(1)$ time and can do all of the following in $O(1)$ time for any node $\mathbf{v}$ of $\mathscr{T}$:

- Create a clone of $\mathbf{v}$, and add it to another FCTS (as a new child of some specified node of the latter).
- Find the parent of $\mathbf{v}$ in $\mathscr{T}$, if $\mathbf{v}$ is not the root.
- Determine the value of $\ell(\mathbf{v})$.
- Determine whether or not $\mathbf{v}$ is a leaf of $\mathscr{T}$.

We also assume that, for any non-leaf node $\mathbf{v}$ of $\mathscr{T}$, we can find all the children of $\mathbf{v}$ in $O(|\mathbf{Children}_{\mathscr{T}}(\mathbf{v})|)$ time.

In the rest of this section we describe simple but efficient implementations of step 2 and of a variant of step 2.

Let $\mathfrak{F}_{\text{in}} = (\mathscr{T}_{\text{in}}, \ell_{\text{in}})$ be some κ-FCTS, and let σ be an ℓ_{in}-increasing leaf enumeration of $\mathbf{Leaves}(\mathscr{T}_{\text{in}})$ such that, whenever $\mathbf{x}$ and $\mathbf{y}$ are leaves of $\mathscr{T}_{\text{in}}$, the answer to the question

$$\text{Does } \mathbf{x} \text{ occur later than } \mathbf{y} \text{ in } \sigma? \tag{2.2}$$

can be determined in $O(1)$ time even if $\ell_{\text{in}}(\mathbf{x}) = \ell_{\text{in}}(\mathbf{y})$.

Our implementation of step 2 runs in $O(|\mathbf{Nodes}(\mathscr{T}_{\text{in}})|)$ time, and does *not* require the actual creation of the sequence σ: We allow σ to be *implicitly* defined by some function $f : \mathbf{Leaves}(\mathscr{T}_{\text{in}}) \times \mathbf{Leaves}(\mathscr{T}_{\text{in}}) \to \{\mathbf{Yes}, \mathbf{No}\}$ such that the answer to (2.2) for any two leaves $\mathbf{x}$ and $\mathbf{y}$ of $\mathscr{T}_{\text{in}}$ is $f(\mathbf{x}, \mathbf{y})$ and this can be computed in $O(1)$ time.[1]

For every $\lambda > 0$ let $\mathfrak{F}_{\text{out},\lambda}$ be the FCTS that should result from pruning $\mathfrak{F}_{\text{in}}$ by removing branches of length $\leq \lambda$ using the leaf enumeration σ. We now explain how $\mathfrak{F}_{\text{out},\lambda}$ can be constructed in $O(|\mathbf{Nodes}(\mathscr{T}_{\text{in}})|)$ time.

For each non-leaf node $\mathbf{w}$ of $\mathscr{T}_{\text{in}}$, we define $\mathsf{next}_\sigma(\mathbf{w}, \mathscr{T}_{\text{in}})$ to be the child of $\mathbf{w}$ in $\mathsf{Path}_\sigma(\mathbf{w}, \mathscr{T}_{\text{in}})$ (i.e., the child of $\mathbf{w}$ that is an ancestor of $\mathsf{lastLeaf}_\sigma(\mathbf{w}, \mathscr{T}_{\text{in}})$); if $\mathbf{w}$ is a leaf of $\mathscr{T}_{\text{in}}$ then we define $\mathsf{next}_\sigma(\mathbf{w}, \mathscr{T}_{\text{in}}) = \mathbf{w}$. During a single postorder traversal $\mathsf{next}_\sigma(\mathbf{w}, \mathscr{T}_{\text{in}})$, $\mathsf{lastLeaf}_\sigma(\mathbf{w}, \mathscr{T}_{\text{in}})$, and $\text{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{w})$ can be computed for all nodes $\mathbf{w}$ of $\mathscr{T}_{\text{in}}$ in $\sum_{\mathbf{w}\in\mathbf{Nodes}(\mathscr{T}_{\text{in}})} O(1 + |\mathbf{Children}_{\mathscr{T}_{\text{in}}}(\mathbf{w})|) = O(|\mathbf{Nodes}(\mathscr{T}_{\text{in}})|)$ time. Then,

[1] Note that no algorithm which actually creates the sequence σ that is defined by any such function f can run in $O(|\mathbf{Nodes}(\mathscr{T}_{\text{in}})|)$ time in all cases, because any comparison sort must perform $\Omega(n \log n)$ comparisons to sort a set of n items (here, leaves) in the worst case [3, Thm. 8.1].

for any given node $\mathbf{v}$ of $\mathscr{T}_{\mathrm{in}}$ it is easy to determine in $O(1)$ time whether $\mathbf{v}$ belongs to $\mathbf{U}^{\lambda}\langle\mathfrak{F}_{\mathrm{in}}\rangle$, to $\mathbf{V}_1^{\lambda}\langle\mathfrak{F}_{\mathrm{in}}\rangle$, or to neither of those sets, and it is easy to find all the nodes of $\mathsf{Path}_{\sigma}(\mathbf{v}, \mathscr{T}_{\mathrm{in}})$ by following a chain of $\mathsf{next}_{\sigma}(\mathbf{w}, \mathscr{T}_{\mathrm{in}})$ nodes that starts with $\mathbf{w} = \mathbf{v}$. Hence we can construct $\mathfrak{F}_{\mathrm{out},\lambda}$ in $O(|\mathbf{Nodes}(\mathscr{T}_{\mathrm{in}})|)$ time, for any positive λ that the user may specify, in the following way:

1. Clone $\mathbf{root}(\mathscr{T}_{\mathrm{in}})$, and initialize the output FCTS (i.e., the FCTS that will be output when the algorithm terminates) to be an FCTS whose only node is the clone of $\mathbf{root}(\mathscr{T}_{\mathrm{in}})$.
2. Do a preorder traversal of the subgraph of $\mathscr{T}_{\mathrm{in}}$ that is induced by the set of nodes $\mathbf{U}^{\lambda}\langle\mathfrak{F}_{\mathrm{in}}\rangle \cup \mathbf{V}_1^{\lambda}\langle\mathfrak{F}_{\mathrm{in}}\rangle$. (This is the rooted tree that is derived from $\mathscr{T}_{\mathrm{in}}$ by ignoring all nodes which do not lie in the set $\mathbf{U}^{\lambda}\langle\mathfrak{F}_{\mathrm{in}}\rangle \cup \mathbf{V}_1^{\lambda}\langle\mathfrak{F}_{\mathrm{in}}\rangle$. Note that this set contains $\mathbf{root}(\mathscr{T}_{\mathrm{in}})$ and all the ancestors of each node in the set.) When any node $\mathbf{v}$ is visited during the traversal, do the following:
   (2a) If $\mathbf{v} \in \mathbf{U}^{\lambda}\langle\mathfrak{F}_{\mathrm{in}}\rangle \setminus \{\mathbf{root}(\mathscr{T}_{\mathrm{in}})\}$, then create a clone of $\mathbf{v}$ and add it to the output FCTS.
   (2b) If $\mathbf{v} \in \mathbf{V}_1^{\lambda}\langle\mathfrak{F}_{\mathrm{in}}\rangle$, then find all the nodes of $\mathsf{Path}_{\sigma}(\mathbf{v}, \mathscr{T}_{\mathrm{in}})$ and, for every such node $\mathbf{w}$, create a clone of $\mathbf{w}$ and add it to the output FCTS (unless $\mathbf{w} = \mathbf{root}(\mathscr{T}_{\mathrm{in}})$).

It is evident that $\mathfrak{F}_{\mathrm{out},\lambda}$ can be constructed in this way, since steps (2a) and (2b) will create clones of all nodes of types (i) and (ii) in Proposition 3 (except the root of $\mathscr{T}_{\mathrm{in}}$) and add them to the output FCTS.

Step 3 of (λ, k)-simplification simplifies $\mathfrak{F}^{\mathbf{crit}}$, where $\mathfrak{F}$ is the output of step 2. We can construct $\mathfrak{F}^{\mathbf{crit}}_{\mathrm{out},\lambda}$ directly, without constructing $\mathfrak{F}_{\mathrm{out},\lambda}$, using a modified version of the algorithm described above in which (2a) and (2b) are replaced with:

(2a′) If $\mathbf{v} \in \mathbf{U}^{\lambda}\langle\mathfrak{F}_{\mathrm{in}}\rangle \setminus \{\mathbf{root}(\mathscr{T}_{\mathrm{in}})\}$, and $\mathbf{Children}_{\mathscr{T}_{\mathrm{in}}}(\mathbf{v})$ contains two or more nodes in $\mathbf{U}^{\lambda}\langle\mathfrak{F}_{\mathrm{in}}\rangle \cup \mathbf{V}_1^{\lambda}\langle\mathfrak{F}_{\mathrm{in}}\rangle$, then create a clone of $\mathbf{v}$ and add it to the output FCTS.
(2b′) If $\mathbf{v} \in \mathbf{V}_1^{\lambda}\langle\mathfrak{F}_{\mathrm{in}}\rangle$, then create a clone of the node $\mathsf{lastLeaf}_{\sigma}(\mathbf{v}, \mathscr{T}_{\mathrm{in}})$ and add it to the output FCTS.

Here (2b′) assumes that $\mathscr{T}_{\mathrm{in}}$ has at least two nodes.

2.5 Elimination of Internal Edges of Length $\leq \lambda$ from $\mathfrak{F}^{\mathbf{crit}}$

Step 3 of (λ, k)-simplification is to eliminate internal edges of length $\leq \lambda$ from $\mathfrak{F}^{\mathbf{crit}}$, where $\mathfrak{F}$ is the FCTS that results from step 2 of (λ, k)-simplification. We now mathematically specify the output of step 3, and then present an algorithm which implements step 3.

2.5.1 Specification of Simplification Step 3

Let $\mathfrak{F} = (\mathscr{T}, \ell)$ be any κ-FCTS. Then, for each $\lambda > 0$, the result of eliminating internal edges of length $\le \lambda$ from $\mathfrak{F}^{\mathbf{crit}}$ is the κ-FCTS $\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle$ that we will define below. The definition will use some notation which we now introduce.

The set $\{\ell(\mathbf{c}) - \ell(\mathbf{c}') \mid \mathbf{c}, \mathbf{c}' \in \mathbf{Crit}(\mathfrak{F}) \setminus \mathbf{Leaves}(\mathfrak{F})$ and $\mathbf{c}' \in \mathbf{c}{\downarrow_{\mathfrak{F}}}\}$ will be denoted by $D(\mathfrak{F})$, and $d_1^{\mathfrak{F}} < d_2^{\mathfrak{F}} < \cdots < d_{|D(\mathfrak{F})|}^{\mathfrak{F}}$ will denote the elements of $D(\mathfrak{F})$ in ascending order. (Note that all elements of $D(\mathfrak{F})$ are positive.) We define $d_0^{\mathfrak{F}} = 0$. For any $\lambda > 0$, we define $\mathrm{pred}_{\mathfrak{F}}(\lambda) = \max\{d \in D(\mathfrak{F}) \cup \{0\} \mid d < \lambda\}$.

Example 5 Let $\mathfrak{F}$ be the FCTS shown in Fig. 2.7. Then we see from Fig. 2.7 that $\mathbf{Crit}(\mathfrak{F}) \setminus \mathbf{Leaves}(\mathfrak{F}) = \{\mathsf{v}_1, \mathsf{v}_4, \mathsf{v}_5, \mathsf{v}_{15}, \mathsf{v}_{16}\}$ and $D(\mathfrak{F}) = \{1, 5, 6, 7, 11, 12\}$. It follows, for example, that, $d_1^{\mathfrak{F}} = 1$, $d_2^{\mathfrak{F}} = 5$, and $\mathrm{pred}_{\mathfrak{F}}(\lambda) = 1$ for $1 < \lambda \le 5$.

Now we define $\mathfrak{F}^{\mathbf{crit}}\langle 0\rangle = \mathfrak{F}^{\mathbf{crit}}$ and, for all $\lambda > 0$, we recursively define $\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle$ to be the κ-FCTS that has the following five properties:

E1: $\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle \sqsubseteq \mathfrak{F}^{\mathbf{crit}}$
E2: $\mathbf{LCN}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle) = \mathbf{LCN}(\mathfrak{F})$
E3: $\mathbf{Leaves}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle) = \mathbf{Leaves}(\mathfrak{F})$
E4: If $\lambda \notin D(\mathfrak{F})$, then $\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle = \mathfrak{F}^{\mathbf{crit}}\langle\mathrm{pred}_{\mathfrak{F}}(\lambda)\rangle$.
E5: For every $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}) \setminus (\mathbf{Leaves}(\mathfrak{F}) \cup \{\mathbf{LCN}(\mathfrak{F})\} \cup \{\mathbf{root}(\mathfrak{F})\})$ and every $i \in \{0, \ldots, |D(\mathfrak{F})| - 1\}$, we have that $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle d_{i+1}^{\mathfrak{F}}\rangle)$ if and only if $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle d_i^{\mathfrak{F}}\rangle)$ and $\ell(\mathbf{c}) - \ell(\mathbf{parent}_{\mathfrak{F}^{\mathbf{crit}}\langle d_i^{\mathfrak{F}}\rangle}(\mathbf{c})) > d_{i+1}^{\mathfrak{F}}$.

E1 implies that $\mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle) \subseteq \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}) = \mathbf{Crit}(\mathfrak{F}) \cup \{\mathbf{root}(\mathfrak{F})\}$, and also implies that $\mathbf{root}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle) = \mathbf{root}(\mathfrak{F})$.

Example 6 Figure 2.8 shows the FCTS $\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle$ in the case where $\mathfrak{F}$ is the FCTS that is shown in Fig. 2.7 and $1 \le \lambda < 5$. Here $d_1^{\mathfrak{F}} = 1$ and $d_2^{\mathfrak{F}} = 5$ (as we observed in Example 5). Since $d_1^{\mathfrak{F}} \le \lambda < d_2^{\mathfrak{F}}$, it follows from E4 that $\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle = \mathfrak{F}^{\mathbf{crit}}\langle d_1^{\mathfrak{F}}\rangle = \mathfrak{F}^{\mathbf{crit}}\langle 1\rangle$. The node v_{16} in Fig. 2.7 is not a node of $\mathfrak{F}^{\mathbf{crit}}\langle d_1^{\mathfrak{F}}\rangle$; indeed, when we put $i = 0$ and $\mathbf{c} = \mathsf{v}_{16}$, the condition $\ell(\mathbf{c}) - \ell(\mathbf{parent}_{\mathfrak{F}^{\mathbf{crit}}\langle d_i^{\mathfrak{F}}\rangle}(\mathbf{c})) > d_{i+1}^{\mathfrak{F}}$ in E5 is not met since $\mathbf{parent}_{\mathfrak{F}^{\mathbf{crit}}\langle d_0^{\mathfrak{F}}\rangle}(\mathsf{v}_{16}) = \mathsf{v}_{15}$ and $\ell(\mathsf{v}_{16}) - \ell(\mathsf{v}_{15}) = 1 = d_1^{\mathfrak{F}}$. But E1–E5 imply that the other 12 nodes of $\mathfrak{F}^{\mathbf{crit}}$ are nodes of $\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle$.

2.5.2 Implementation of Simplification Step 3

It is possible to perform simplification step 3 (i.e., to construct $\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle$ from $\mathfrak{F}^{\mathbf{crit}}$) by direct application of E1–E5. However, this would require computation of the sorted sequence $d_1^{\mathfrak{F}} < d_2^{\mathfrak{F}} < \cdots < d_k^{\mathfrak{F}}$, where $d_k^{\mathfrak{F}}$ is λ or $\mathrm{pred}_{\mathfrak{F}}(\lambda)$ according to whether $\lambda \in D(\mathfrak{F})$ or $\lambda \notin D(\mathfrak{F})$, followed by k tree traversals that successively find the nodes of $\mathfrak{F}^{\mathbf{crit}}\langle d_1^{\mathfrak{F}}\rangle, \mathfrak{F}^{\mathbf{crit}}\langle d_2^{\mathfrak{F}}\rangle, \ldots, \mathfrak{F}^{\mathbf{crit}}\langle d_k^{\mathfrak{F}}\rangle$.

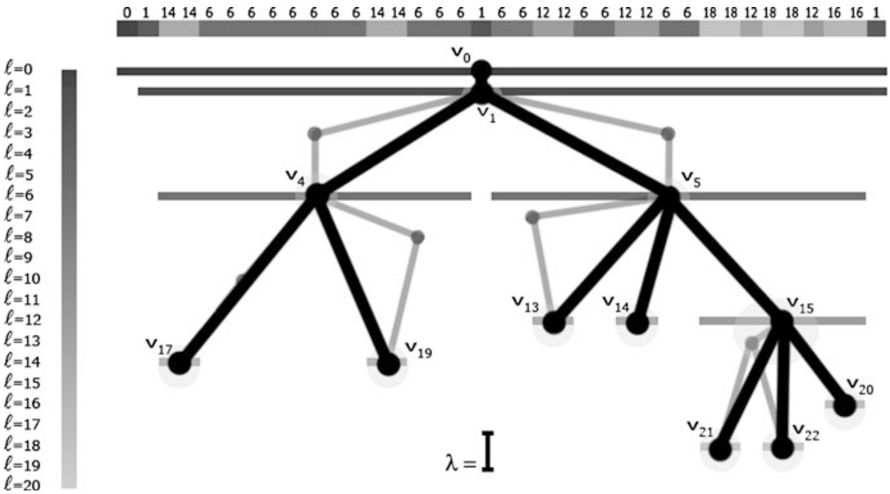


Fig. 2.8 The effect of eliminating internal edges of length $\leq \lambda$ from $\mathfrak{F}^{\mathbf{crit}} = (\mathscr{T}^{\mathbf{crit}}, \ell^{\mathbf{crit}})$ is shown here, in the case where $\mathfrak{F} = (\mathscr{T}, \ell)$ is the FCTS of Fig. 2.7 and $1 \leq \lambda < 5$. The nodes and edges of the resulting FCTS $\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle$ are shown as *fat/thick black nodes and edges*. Other nodes and edges of the tree $\mathscr{T}$ of Fig. 2.7 are colored *gray*, but two of those nodes (v_9 and v_{11} in Fig. 2.7) and three of those edges are partially or completely hidden by the *thick black edge* that joins v_4 to v_{17}. Note that, since 2 is a possible value of λ in this figure, and since $\mathfrak{F}$ is the result of applying steps 1 and 2 of (λ, k)-simplification to the FCTS shown in Fig. 2.3 in the case $\lambda = 2$ and $k = 1$, the FCTS shown in this figure is the (2, 1)-simplification of the FCTS shown in Fig. 2.3

Algorithm 1 below, which performs just one tree traversal after the initial cloning step, will usually be a much more efficient implementation of step 3. It inputs a κ-FCTS $\mathfrak{F}_{\mathrm{in}} = (\mathscr{T}_{\mathrm{in}}, \ell_{\mathrm{in}})$ and a positive λ, and constructs $\mathfrak{F}_{\mathrm{in}}^{\mathbf{crit}}\langle\lambda\rangle$ by creating a clone $(\mathscr{T}, \ell)$ of $\mathfrak{F}_{\mathrm{in}}^{\mathbf{crit}} = (\mathscr{T}_{\mathrm{in}}^{\mathbf{crit}}, \ell_{\mathrm{in}}^{\mathbf{crit}})$ and then labeling each node $\mathbf{c}$ of $\mathscr{T}$ with a value $\mathbf{c}.\mathsf{label}$ such that $\mathfrak{F}_{\mathrm{in}}^{\mathbf{crit}}\langle\lambda\rangle = (\mathscr{T}, \ell) - \{\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}) \mid \mathbf{v}.\mathsf{label} \leq \lambda\}$. The correctness of this algorithm is proved in Appendix A.

If we write $h(\mathfrak{F}_{\mathrm{in}}, \lambda)$ to denote the length $l \geq 1$ of the longest chain of nodes $\mathbf{v}_1 \succ_{\mathscr{T}_{\mathrm{in}}} \cdots \succ_{\mathscr{T}_{\mathrm{in}}} \mathbf{v}_l$ in $\mathbf{Crit}(\mathfrak{F}_{\mathrm{in}})$ for which $\ell_{\mathrm{in}}(\mathbf{v}_1) - \ell_{\mathrm{in}}(\mathbf{v}_l) \leq \lambda$, then we see from the initial step "$(\mathscr{T}, \ell) \longleftarrow$ a clone of $(\mathscr{T}_{\mathrm{in}}^{\mathbf{crit}}, \ell_{\mathrm{in}}^{\mathbf{crit}})$" of Algorithm 1 and from the **repeat** ... **until** loop in `labelDescendants` (Procedure 1) that, under the assumptions which are stated at the beginning of Sect. 2.4.3, the running time of Algorithm 1 is $O(|\mathbf{Nodes}(\mathfrak{F}_{\mathrm{in}})| + h(\mathfrak{F}_{\mathrm{in}}, \lambda)\,|\mathbf{Crit}(\mathfrak{F}_{\mathrm{in}})|)$.

2.6 Demonstration of Potential Biological Applicability

To illustrate the potential usefulness of our simplified FCTSs in identifying structural differences between macromolecules, we looked for two structures that are very similar, but not identical. Appropriate data sets were kindly provided by Roberto Marabini of the Universidad Autónoma de Madrid.

Algorithm 1: Eliminate Internal Edges of Length $\leq \lambda$ from $\mathfrak{F}^{\mathbf{crit}}$

inputs : a κ-FCTS $\mathfrak{F}_{\text{in}} = (\mathscr{T}_{\text{in}}, \ell_{\text{in}})$; a positive real value λ
output: a κ-FCTS $\mathfrak{F}_{\text{out}}$ that satisfies $\mathfrak{F}_{\text{out}} \sqsubseteq \mathfrak{F}^{\mathbf{crit}}_{\text{in}}$

$(\mathscr{T}, \ell) \longleftarrow$ a clone of $(\mathscr{T}^{\mathbf{crit}}_{\text{in}}, \ell^{\mathbf{crit}}_{\text{in}})$;
$\mathbf{root}(\mathscr{T})$.label $\longleftarrow \infty$;
$\mathbf{LCN}(\mathscr{T})$.label $\longleftarrow \infty$;
foreach $\mathbf{x} \in \mathbf{Children}_{\mathscr{T}}(\mathbf{LCN}(\mathscr{T}))$ **do** `labelDescendants`$(\mathbf{x}, \mathscr{T}, \ell, \lambda)$;
$\mathfrak{F}_{\text{out}} \longleftarrow (\mathscr{T}, \ell) - \{\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}) \mid \mathbf{v}.\text{label} \leq \lambda\}$;

Procedure 1: `labelDescendants`$(\mathbf{c}, \mathscr{T}, \ell, \lambda)$

if $\mathbf{c} \notin \mathbf{Leaves}(\mathscr{T})$ **then**
  $\mathbf{c}' \longleftarrow \mathbf{c}$;
  repeat
    $\mathbf{c}' \longleftarrow \mathbf{parent}_{\mathscr{T}}(\mathbf{c}')$;
    $\mathbf{c}$.label $\longleftarrow \ell(\mathbf{c}) - \ell(\mathbf{c}')$;
  until ($\mathbf{c}$.label $> \lambda$ **or** $\mathbf{c}$.label $\leq \mathbf{c}'$.label);
  foreach $\mathbf{x} \in \mathbf{Children}_{\mathscr{T}}(\mathbf{c})$ **do** `labelDescendants`$(\mathbf{x}, \mathscr{T}, \ell, \lambda)$;
else $\mathbf{c}$.label $\longleftarrow \infty$;

These data sets originate from the work of San Martín et al. [9], which investigated some biological questions associated with adenoviruses. These viruses are responsible for a large number of diseases in humans such as gastrointestinal and respiratory infections, but can also be used in gene therapy and vaccine delivery [8]. They have an icosahedral shape with a diameter of approximately 900 Å. At each of the 12 vertices of the icosahedron there is a substructure referred to as a *penton*, and the rest of the surface of the icosahedron consists of 240 *hexons*. To reflect this, our simplified FCTSs of these viruses would be expected to have 252 leaves, one for each penton or hexon. This is indeed the case, as we will see.

In the course of their work, San Martín et al. [9] produced a *mutant* version of the *wildtype* version of the adenovirus they were investigating. The two are identical except for a change in a protein (called IIIa). Surface renderings and central cross-sections of the two versions are shown in Fig. 2.9. We now describe how, in spite of their great similarity, the two versions can be distinguished from each other by an obvious topological difference between their simplified FCTSs.

Each version of the virus studied by San Martín et al. [9] was represented by a grayscale volume image on a $275 \times 275 \times 275$ array of sample points. We further quantized the graylevels in each of these images to a set of just 256 equally spaced values represented by the integers $0, \ldots, 255$, where 0 corresponded to the minimum and 255 the maximum graylevel in the original image. For each resulting image I, we constructed $\mathbf{FCTS}_{\kappa}(I)$ using 6-adjacency as our adjacency relation κ, and computed the (λ, k)-simplification of $\mathbf{FCTS}_{\kappa}(I)$ for various choices of λ and k.

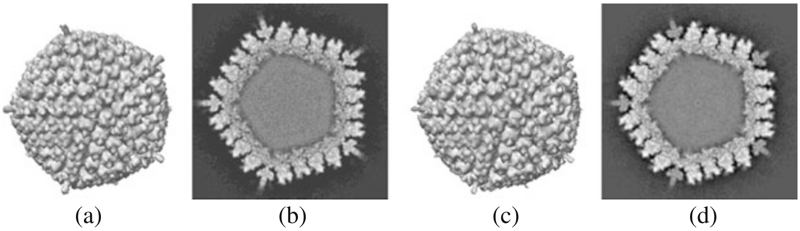


Fig. 2.9 Adenovirus. Surface rendering (**a**) and central cross-section (**b**) of the wildtype version. Surface rendering (**c**) and central cross-section (**d**) of the mutant version

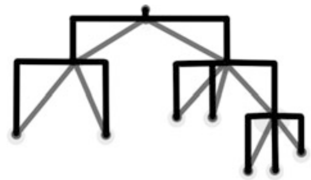
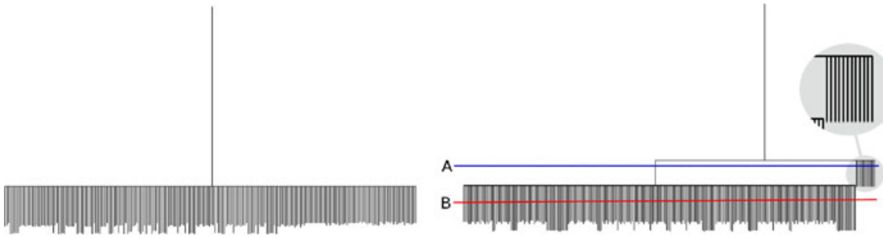


Fig. 2.10 The *gray lines* show an FCTS $(\mathscr{T}, \ell)$ using the tree representation of Figs. 2.1–2.8. The *black lines* show the same FCTS using the tree representation of Fig. 2.11. (In the latter representation, a node that is neither the root nor a leaf is represented by a *horizontal segment*, and an edge from a node $\mathbf{p}$ to one of its children $\mathbf{c}$ is represented by a *vertical segment* of length proportional to $\ell(\mathbf{c}) - \ell(\mathbf{p})$)

We found that $\lambda = 10$ and $k = 799$ were good choices that yielded topologically different simplified FCTSs for the two versions of the virus. These simplified FCTSs are shown in Fig. 2.11, using a tree representation that is explained in Fig. 2.10. Each simplified FCTS has 252 leaves, corresponding to the 12 pentons and 240 hexons. For the wildtype version, the lowest critical node is the parent of all 252 leaves; see Fig. 2.11(a). For the mutant version, the lowest critical node is the parent of the 12 leaves that correspond to pentons, but is the grandparent of the 240 leaves that correspond to hexons; see Fig. 2.11(b). These simplified FCTSs indicate that for the mutant version of the virus there is a substantial range of threshold levels (such as level A in Fig. 2.11(b)) at which the pentons are disconnected from each other and from the hexons, but the hexons are connected to each other; for the wildtype version there is no such range of threshold values. Interestingly, San Martín et al. [9] do not mention this difference between the two versions of the virus, although they do point out that in images of the mutant version pentons have lower graylevels than hexons. (The latter can be seen in Fig. 2.9(d), and is also indicated by Fig. 2.11(b); when the image of the mutant virus is thresholded at the graylevel B in Fig. 2.11(b), hexons are observable but pentons are not.)

So our simplified FCTSs may possibly have revealed a previously unobserved difference between the mutant and the wildtype versions of the virus: for the mutant version, there is a substantial range of threshold values at which the hexons are connected to each other, but no penton is connected to a hexon or to another penton. To investigate whether this is a genuine difference between the two versions of the



(a) (b)

Fig. 2.11 (λ, k)-simplifications of FCTSs of wildtype (**a**) and mutant (**b**) adenoviruses, where $\lambda = 10$ and $k = 799$. (The tree representation used in this figure is explained in Fig. 2.10.) In (**a**), the lowest critical node (represented by the *horizontal line segment*) is the parent of all 252 leaves of the tree. In (**b**), the lowest critical node (represented by the *horizontal line segment* above *line A*) is the parent of the rightmost 12 leaves, which correspond to pentons, but is the grandparent of the other 240 leaves, which correspond to hexons

virus or merely a difference between the specific volume images from which we produced our FCTSs, we carried out a further study.

Ideally, we would have compared simplified FCTSs of, say, 10 independently reconstructed volume images of each version, but such data were not available to us. So we conducted the following approximation of such a study.

For each version of the virus, we randomly selected 2000 out of 3000 available projection images, and used them to reconstruct a volume image on a $275 \times 275 \times 275$ array of points. This was repeated 10 times.

For each of the 20 resulting volume images, we produced a simplified FCTS using the above-mentioned parameters. In each of the 10 simplified FCTSs of the mutant adenovirus, the lowest critical node had 13 children, 12 corresponding to the pentons and the 13th being the root of a subtree whose leaves corresponded to the hexons, as in Fig. 2.11(b). But this was not true of the 10 simplified FCTSs of the wildtype adenovirus; they were all similar to Fig. 2.11(a).

These results provide some evidence to support the hypothesis that images of the mutant version of the virus can be distinguished from images of the wildtype version by the existence in the former (but not the latter) of a substantial range of threshold values with the above-mentioned properties. More investigation would be needed to confirm this hypothesis.

In any event, this example illustrates how our simplified FCTSs may reveal interesting structural differences between two similar macromolecules.

2.7 Possibilities for Future Work

2.7.1 How Can Our Simplification Method and Theorem 1 Be Adapted to Contour Trees?

FCTSs are closely related to contour trees, which are also used to represent images (see, e.g., [12]). Intuitively, a contour tree of an image is an undirected graph each of whose points represents a *contour*—i.e., a connected component of a level set—of a continuous scalar field derived from the image by interpolation. Contours of scalar fields derived from 3D images are often called *isosurfaces*.

To define the contour tree, let $I : \mathcal{S} \to \mathbb{R}$ be an image whose domain $\mathcal{S}$ is a finite set of points in Euclidean n-space $\mathbb{R}^n$ (for some n). As usual, we refer to the elements of $\mathcal{S}$ as spels. For simplicity in defining the contour tree, we require that I be 1-to-1—i.e., we require that no two spels have exactly the same graylevel in I. (This prevents distinct spels from lying on the same contour, and will allow the contour tree to be defined as a graph whose vertices are spels.) Note that a 1-to-1 image can be produced from any image by making arbitrarily small graylevel perturbations.

For any adjacency relation α on $\mathcal{S}$, we write $\mathbf{Graph}(\alpha)$ to denote the undirected simple graph whose vertex set is $\mathcal{S}$ and whose vertex adjacency relation is α. Recall that an undirected graph is said to be a *tree* if it is connected and acyclic.

We will be considering α-FCTSs of I and its negative image $-I$ (which is obtained from I by multiplying each spel's graylevel by -1). For any $x \in \mathcal{S}$, when discussing $\mathbf{FCTS}_\alpha(I)$ and $\mathbf{FCTS}_\alpha(-I)$ we write $\langle x \rangle$ to denote either the node $\mathcal{C}_\alpha(x, I)$ of $\mathbf{FCTS}_\alpha(I)$ or the node $\mathcal{C}_\alpha(x, -I)$ of $\mathbf{FCTS}_\alpha(-I)$.

Now suppose the adjacency relation α is unknown, but we know $\mathbf{Graph}(\alpha)$ is a tree. Then α is uniquely determined by $\mathbf{FCTS}_\alpha(I)$ and $\mathbf{FCTS}_\alpha(-I)$. Indeed, it is not hard to verify that s is an end vertex of $\mathbf{Graph}(\alpha)$ whose only α-neighbor is s' just if in one of $\mathbf{FCTS}_\alpha(I)$ and $\mathbf{FCTS}_\alpha(-I)$ we have that $\langle s \rangle = \{s\}$ is a leaf whose parent is $\langle s' \rangle$, and in the other of $\mathbf{FCTS}_\alpha(I)$ and $\mathbf{FCTS}_\alpha(-I)$ we have that $\langle s \rangle$ has exactly one child. Further, if s is any end vertex of $\mathbf{Graph}(\boldsymbol{\alpha})$ and the restrictions of I and α to $\mathcal{S} \setminus \{s\}$ are denoted by I' and $\boldsymbol{\alpha}'$, then $\mathbf{FCTS}_{\alpha'}(I') = \mathbf{FCTS}_\alpha(I) - \{\langle s \rangle\}$ and $\mathbf{FCTS}_{\alpha'}(-I') = \mathbf{FCTS}_\alpha(-I) - \{\langle s \rangle\}$, from which α' can be computed (e.g., recursively). Algorithm 4.2 in [1], which is based on these two facts, can be used to construct the tree $\mathbf{Graph}(\alpha)$ in $O(|\mathcal{S}|)$ time, given I, $\mathbf{FCTS}_\alpha(I)$, and $\mathbf{FCTS}_\alpha(-I)$.

To define a contour tree of I, we first choose a "good" adjacency relation κ on $\mathcal{S}$. Let $\mathcal{L}$ be a geometric simplicial complex whose vertex set is $\mathcal{S}$ and whose union is connected and simply connected. Let κ be the adjacency relation on $\mathcal{S}$ such that $(s,t) \in \kappa$ if and only if s and t are the endpoints of an edge of the complex $\mathcal{L}$.

Now let $f : \bigcup \mathcal{L} \to \mathbb{R}$ be the continuous scalar field obtained when we extend the image I by linear interpolation over each simplex of $\mathcal{L}$. Let $\lhd$ be the strict partial order on $\mathcal{S}$ such that $s \lhd s'$ if and only if $I(s) < I(s')$ and there is a path in $\bigcup \mathcal{L}$ from s to s' along which f's value increases monotonically from $I(s)$ to $I(s')$. Let $\alpha(I, \mathcal{L})$ be the adjacency relation on $\mathcal{S}$ such that $(s, s') \in \alpha(I, \mathcal{L})$ if and only if one of the spels s and s' is an immediate successor of the other with respect to $\lhd$. (We say y is an *immediate successor* of x with respect to $\lhd$ if $x \lhd y$ and there is no z such that $x \lhd z \lhd y$.) It can be shown, using the linearity of f on each simplex of $\mathcal{L}$ and, e.g., well known properties of Reeb graphs (see [1, 4]), that $\mathbf{FCTS}_{\alpha(I,\mathcal{L})}(I) = \mathbf{FCTS}_\kappa(I)$ and $\mathbf{FCTS}_{\alpha(I,\mathcal{L})}(-I) = \mathbf{FCTS}_\kappa(-I)$, and that $\mathbf{Graph}(\alpha(I, \mathcal{L}))$ is a tree.

We define the *κ-contour tree*[2] of I to be $\mathbf{Graph}(\alpha(I, \mathscr{L}))$. It follows from our remarks above that this tree is uniquely determined by $\mathbf{FCTS}_\kappa(I)$ and $\mathbf{FCTS}_\kappa(-I)$, and that the tree can be constructed in $O(|\mathbb{S}|)$ time from I and these two κ-FCTSs.

In view of the close relationship between contour trees and FCTSs, we are hopeful that it will be possible to formulate a simplification method for contour trees that is similar to our simplification method for FCTSs and is provably robust in the sense that it can be shown to satisfy an analog of Theorem 1.

2.7.2 *Does the Bottleneck Stability Theorem Have an Analog for FCTSs That Implies Theorem 1?*

Let $I : \mathbb{S} \to \mathbb{R}$ be any image whose domain $\mathbb{S}$ is finite, and κ any adjacency relation on $\mathbb{S}$ such that $\mathbb{S}$ is κ-connected. A descriptor of I that is related to (but contains less information than) $\mathbf{FCTS}_\kappa(I)$ is the 0th persistence diagram of $-I$ based on the adjacency relation κ. (Here the minus sign reflects the fact that persistence diagrams are defined in terms of the *sublevel* sets of filter functions[3] whereas FCTSs are defined in terms of the *superlevel* sets of images.) The 0th persistence diagram of $-I$ based on κ is a multiset of points in $\mathbb{R} \times (\mathbb{R} \cup \{+\infty\})$ that contains one point for each leaf of $\mathbf{FCTS}_\kappa(I)$. The diagram is easily computed[4] from $\mathbf{FCTS}_\kappa(I)$, but it is not possible to reconstruct $\mathbf{FCTS}_\kappa(I)$ from the diagram.

Step 2 of our simplification method eliminates those leaves of the FCTS that are represented in the 0th persistence diagram by points (x, y) for which $y - x \leq \lambda$. Moreover, for any two images $I, I' : \mathbb{S} \to \mathbb{R}$, the L_∞-distance between the filter functions used to define the 0th persistence diagrams of $-I$ and $-I'$ is $\|I - I'\|_\infty$. For these reasons, our Theorem 1 is vaguely reminiscent of the $p = 0$ case of the Bottleneck Stability Theorem for persistence diagrams [2], [4, p. 182], which states

[2]The tree defined here is the *augmented contour tree* of [1]. It may have many vertices s that have just two neighbors, of which one neighbor s' satisfies $I(s') < I(s)$ while the other neighbor s'' satisfies $I(s'') > I(s)$. Many authors define the contour tree to not include such vertices.

[3]Persistence diagrams are commonly defined (as in [4, pp. 150–152]) for a filter function $f : \mathscr{K} \to \mathbb{R}$, where $\mathscr{K}$ is a suitable simplicial complex. To define the 0th persistence diagram of $-I$ based on the adjacency relation κ, we can take the simplicial complex $\mathscr{K}$ to be the simple graph whose vertex set is $\mathbb{S}$ and whose edges join κ-adjacent elements of $\mathbb{S}$, and we can use the filter function $f : \mathscr{K} \to \mathbb{R}$ for which $f(v) = -I(v)$ if v is a vertex of $\mathscr{K}$, and $f(e) = -\min(I(x), I(y))$ if e is an edge of $\mathscr{K}$ that joins the vertices x and y.

[4]Let $\mathbf{FCTS}_\kappa(I) = (\mathscr{T}, \ell)$, and let $\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}]$ be any ℓ-increasing enumeration of the leaves of $\mathscr{T}$. For $1 \leq i < \mathsf{n}$, each leaf $\mathsf{leaf}[i]$ is represented in the persistence diagram by a point $(-\ell(\mathsf{leaf}[i]), -\ell(\mathbf{a}))$ where $\mathbf{a}$ is the closest ancestor of $\mathsf{leaf}[i]$ that is an ancestor of at least one of the leaves $\mathsf{leaf}[i+1], \ldots, \mathsf{leaf}[\mathsf{n}]$. The last leaf $\mathsf{leaf}[\mathsf{n}]$ of the ℓ-increasing enumeration is represented in the persistence diagram by the point $(-\ell(\mathsf{leaf}[\mathsf{n}]), +\infty)$. The diagram is defined to also contain, for each $z \in \mathbb{R}$, a point (z, z) with countably infinite multiplicity.

that the bottleneck distance[5] between the pth persistence diagrams of two filter functions cannot exceed the L_∞-distance between those functions.

The Bottleneck Stability Theorem appears not to imply our Theorem 1, because the FCTSs of two images I_1 and I_2 need not be essentially isomorphic even if $-I_1$ and $-I_2$ have the same persistence diagrams. However, it might be possible to prove an analogous stability theorem for FCTSs that does imply Theorem 1.

2.7.3 Can Images Be Simplified Using Variants of Our Method?

In view of the natural bijective correspondence between grayscale images (with finite connected domains) and FCTSs, our method of simplifying FCTSs might also be construed as a method of simplifying images. Unfortunately we have found that, when used for that purpose, it will often be unsatisfactory. (One reason is that the omission of non-critical non-root nodes before performing simplification step 3 may reduce the graylevels of some spels in the resulting image by too much.) Nevertheless, we believe that it may be worthwhile to investigate variants of our method that might be more useful for image simplification.

2.8 Conclusion

FCTSs can be used as descriptors of EM maps and other grayscale images, but unsimplified FCTSs are too sensitive to errors in the image. This chapter has specified a method of simplifying FCTSs that is provably robust (and capable of efficient implementation). Our main theorem and its corollary (Theorem 1 and Corollary 2) conservatively quantify the extent of the method's robustness. We have presented some experimental evidence that the simplified FCTSs produced by our method are useful for the exploration of macromolecular databases. We hope further experimentation will yield more evidence of this or suggest fruitful refinements of our method. Some other avenues for future research have also been discussed.

Acknowledgements We thank Edgar Garduño, Roberto Marabini, and Homeira Pajoohesh for discussions regarding this chapter. The work was supported by awards R01HL070472 from the National Heart, Lung, and Blood Institute and DMS-1114901 from the National Science Foundation. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Heart, Lung, and Blood Institute, the National Institutes of Health, or the National Science Foundation.

[5]The *bottleneck distance* between two persistence diagrams D_1 and D_2 is the infimum of $\sup_{d \in D_1} \|d - \eta(d)\|_\infty$ over all bijections $\eta : D_1 \to D_2$.

Appendix A: Some Properties of Simplification Steps 2 and 3, and a Proof of the Correctness of Algorithm 1

A.1 Properties of Simplification Step 2

Here we prove the main result of Sect. 2.4.2, and establish other properties of simplification step 2 that are used in our proof of the Main Theorem.

Lemma A1 *Let* $\mathfrak{F}_{\mathrm{in}} = (\mathscr{T}_{\mathrm{in}}, \ell_{\mathrm{in}})$ *be any* κ*-FCTS, let* $\lambda > 0$, *and let* $\mathbf{s}$ *and* $\mathbf{s}'$ *be any two distinct leaves of a* κ*-FCTS* $\mathfrak{F}_{\mathrm{out}} = (\mathscr{T}_{\mathrm{out}}, \ell_{\mathrm{out}})$ *that results from pruning* $\mathfrak{F}_{\mathrm{in}}$ *by removing branches of length* $\le \lambda$. *Then* (*regardless of which* ℓ_{in}*-increasing enumeration of* $\mathbf{Leaves}(\mathscr{T}_{\mathrm{in}})$ *is used to perform the pruning*):

(i) $\bigwedge_{\mathscr{T}_{\mathrm{out}}}\{\mathbf{s}, \mathbf{s}'\} = \bigwedge_{\mathscr{T}_{\mathrm{in}}}\{\mathbf{s}, \mathbf{s}'\}$
(ii) $\min(\ell_{\mathrm{out}}(\mathbf{s}), \ell_{\mathrm{out}}(\mathbf{s}')) - \ell_{\mathrm{out}}(\bigwedge_{\mathscr{T}_{\mathrm{out}}}\{\mathbf{s}, \mathbf{s}'\}) > \lambda$

Proof The hypotheses imply that properties P1–P4 hold with respect to some ℓ_{in}-increasing enumeration of $\mathbf{Leaves}(\mathscr{T}_{\mathrm{in}})$. It follows from P4 that, for all $\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}_{\mathrm{out}})$, every node in $\mathbf{v}{\Downarrow}_{\mathscr{T}_{\mathrm{in}}}$ is also a node in $\mathbf{v}{\Downarrow}_{\mathscr{T}_{\mathrm{out}}}$. Therefore $\mathbf{v}{\Downarrow}_{\mathscr{T}}$ is the same set regardless of whether $\mathscr{T} = \mathscr{T}_{\mathrm{out}}$ or $\mathscr{T} = \mathscr{T}_{\mathrm{in}}$. So $\bigwedge_{\mathscr{T}}\{\mathbf{s}, \mathbf{s}'\}$ is the same node regardless of whether $\mathscr{T} = \mathscr{T}_{\mathrm{out}}$ or $\mathscr{T} = \mathscr{T}_{\mathrm{in}}$, since $\bigwedge_{\mathscr{T}}\{\mathbf{s}, \mathbf{s}'\}$ is just the element of $\mathbf{s}{\Downarrow}_{\mathscr{T}} \cap \mathbf{s}'{\Downarrow}_{\mathscr{T}}$ that is a descendant in $\mathscr{T}$ of every element of that set. Hence (i) holds.

To prove (ii), we may assume without loss of generality that, in the ℓ_{in}-increasing leaf enumeration that is used for pruning, $\mathbf{s}$ occurs later than $\mathbf{s}'$. (This assumption implies that $\min(\ell_{\mathrm{in}}(\mathbf{s}), \ell_{\mathrm{in}}(\mathbf{s}')) = \ell_{\mathrm{in}}(\mathbf{s}')$.) Then, since $\mathbf{s}' \in \mathbf{Leaves}(\mathscr{T}_{\mathrm{out}})$, property P3 implies that $\ell_{\mathrm{in}}(\mathbf{s}') - \ell_{\mathrm{in}}(\bigwedge_{\mathscr{T}_{\mathrm{in}}}\{\mathbf{s}, \mathbf{s}'\}) > \lambda$, which is equivalent to:

$$\min\big(\ell_{\mathrm{in}}(\mathbf{s}), \ell_{\mathrm{in}}(\mathbf{s}')\big) - \ell_{\mathrm{in}}\Big(\bigwedge\nolimits_{\mathscr{T}_{\mathrm{in}}}\{\mathbf{s}, \mathbf{s}'\}\Big) > \lambda \tag{A1}$$

But (A1) is equivalent to assertion (ii), because of assertion (i) and the fact that ℓ_{out} is just the restriction of ℓ_{in} to $\mathbf{Nodes}(\mathscr{T}_{\mathrm{out}})$. □

Corollary A2 *Let* λ *be any positive value, and* $\mathfrak{F}_{\mathrm{out}}$ *any* κ*-FCTS that results from pruning a* κ*-FCTS* $\mathfrak{F}_{\mathrm{in}}$ *by removing branches of length* $\le \lambda$. *Then, for all* $\mathbf{v} \in \mathbf{Crit}(\mathfrak{F}_{\mathrm{out}}) \setminus \mathbf{Leaves}(\mathfrak{F}_{\mathrm{out}})$, *we have that* $\mathbf{v} \in \mathbf{Crit}(\mathfrak{F}_{\mathrm{in}}) \setminus \mathbf{Leaves}(\mathfrak{F}_{\mathrm{in}})$ *and* $\mathrm{depth}_{\mathfrak{F}_{\mathrm{out}}}(\mathbf{v}) > \lambda$.

Proof Let $\mathfrak{F}_{\mathrm{out}} = (\mathscr{T}_{\mathrm{out}}, \ell_{\mathrm{out}})$, and let $\mathbf{v} \in \mathbf{Crit}(\mathfrak{F}_{\mathrm{out}}) \setminus \mathbf{Leaves}(\mathfrak{F}_{\mathrm{out}})$. Then $\mathbf{v} = \bigwedge_{\mathscr{T}_{\mathrm{out}}}\{\mathbf{s}, \mathbf{s}'\}$ for some distinct leaves $\mathbf{s}$ and $\mathbf{s}'$ of $\mathfrak{F}_{\mathrm{out}}$. Now $\mathbf{v} = \bigwedge_{\mathscr{T}_{\mathrm{in}}}\{\mathbf{s}, \mathbf{s}'\}$ (by assertion (i) of Lemma A1), and so $\mathbf{v} \in \mathbf{Crit}(\mathfrak{F}_{\mathrm{in}}) \setminus \mathbf{Leaves}(\mathfrak{F}_{\mathrm{in}})$. Moreover, we have that $\mathrm{depth}_{\mathfrak{F}_{\mathrm{out}}}(\mathbf{v}) \ge \ell_{\mathrm{out}}(\mathbf{s}) - \ell_{\mathrm{out}}(\mathbf{v}) = \ell_{\mathrm{out}}(\mathbf{s}) - \ell_{\mathrm{out}}(\bigwedge_{\mathscr{T}_{\mathrm{out}}}\{\mathbf{s}, \mathbf{s}'\}) > \lambda$, where the second inequality follows from assertion (ii) of Lemma A1. □

Lemma A3 *Let* $\mathfrak{F}_{\mathrm{in}} = (\mathscr{T}_{\mathrm{in}}, \ell_{\mathrm{in}})$ *be a* κ*-FCTS, let* $\lambda > 0$, *and let* $\mathfrak{F}_{\mathrm{out}} = (\mathscr{T}_{\mathrm{out}}, \ell_{\mathrm{out}})$ *be the* κ*-FCTS that results from pruning* $\mathfrak{F}_{\mathrm{in}}$ *by removing branches of length* $\le \lambda$

using an ℓ_{in}*-increasing leaf enumeration* $\sigma = (\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}])$ *of* $\mathbf{Leaves}(\mathscr{T}_{\text{in}})$. *Then*:

(a) *For all* $\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}_{\text{in}}) \setminus \mathbf{Nodes}(\mathscr{T}_{\text{out}})$, $\mathbf{v}{\Uparrow}_{\mathscr{T}_{\text{in}}} \cap \mathbf{Nodes}(\mathscr{T}_{\text{out}}) = \emptyset$.
(b) *For all* $\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}_{\text{in}})$, $\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}_{\text{out}})$ *if and only if* $\mathsf{lastLeaf}_\sigma(\mathbf{v}, \mathscr{T}_{\text{in}}) \in \mathbf{Leaves}(\mathscr{T}_{\text{out}})$.
(c) *For all* $\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}_{\text{out}})$, $\text{depth}_{\mathfrak{F}_{\text{out}}}(\mathbf{v}) = \text{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{v})$.

Proof For brevity, we write $\mathsf{lastLeaf}_\sigma(\mathbf{v})$ for $\mathsf{lastLeaf}_\sigma(\mathbf{v}, \mathscr{T}_{\text{in}})$. Evidently, (a) follows from P4, and the "if" part of (b) follows from (a). To establish the "only if" part of (b), let $\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}_{\text{out}})$, and let $\mathsf{leaf}[i] = \mathsf{lastLeaf}_\sigma(\mathbf{v})$. We need to show that $\mathsf{leaf}[i] \in \mathbf{Nodes}(\mathscr{T}_{\text{out}})$. If $i = \mathsf{n}$ then this is true (by property P2), so let us assume $i < \mathsf{n}$. Let j be any element of the set $\{i+1, \ldots, \mathsf{n}\}$ (so that $\mathsf{leaf}[j] \notin \mathbf{Leaves}(\mathscr{T}_{\text{in}}[\mathbf{v}])$). Now we claim that $\mathsf{leaf}[j]$ must satisfy $\ell_{\text{in}}(\mathsf{leaf}[i]) - \ell_{\text{in}}(\bigwedge_{\mathscr{T}_{\text{in}}}\{\mathsf{leaf}[j], \mathsf{leaf}[i]\}) > \lambda$.

To see this, let $\mathsf{leaf}[k]$ be any leaf of $\mathscr{T}_{\text{out}}[\mathbf{v}]$; such a leaf must exist, by P4. As $\mathsf{leaf}[i] = \mathsf{lastLeaf}_\sigma(\mathbf{v})$, we have that $i \geq k$ and $\ell_{\text{in}}(\mathsf{leaf}[i]) \geq \ell_{\text{in}}(\mathsf{leaf}[k])$. As $j \in \{i+1, \ldots, \mathsf{n}\}$, we have that $j \in \{k+1, \ldots, \mathsf{n}\}$. Therefore, since $\mathsf{leaf}[k] \in \mathbf{Leaves}(\mathscr{T}_{\text{out}})$, property P3 implies that:

$$\ell_{\text{in}}\big(\mathsf{leaf}[k]\big) - \ell_{\text{in}}\Big(\bigwedge\nolimits_{\mathscr{T}_{\text{in}}}\big\{\mathsf{leaf}[j], \mathsf{leaf}[k]\big\}\Big) > \lambda \tag{A2}$$

But, since $\mathsf{leaf}[i]$ and $\mathsf{leaf}[k]$ are leaves of $\mathscr{T}_{\text{in}}[\mathbf{v}]$ but $\mathsf{leaf}[j]$ is not,

$$\bigwedge\nolimits_{\mathscr{T}_{\text{in}}}\big\{\mathsf{leaf}[j], \mathsf{leaf}[i]\big\} = \bigwedge\nolimits_{\mathscr{T}_{\text{in}}}\big\{\mathsf{leaf}[j], \mathbf{v}\big\} = \bigwedge\nolimits_{\mathscr{T}_{\text{in}}}\big\{\mathsf{leaf}[j], \mathsf{leaf}[k]\big\}$$

and (since $\ell_{\text{in}}(\mathsf{leaf}[i]) \geq \ell_{\text{in}}(\mathsf{leaf}[k])$) this implies:

$$\begin{aligned} &\ell_{\text{in}}\big(\mathsf{leaf}[i]\big) - \ell_{\text{in}}\Big(\bigwedge\nolimits_{\mathscr{T}_{\text{in}}}\big\{\mathsf{leaf}[j], \mathsf{leaf}[i]\big\}\Big) \\ &\quad \geq \ell_{\text{in}}\big(\mathsf{leaf}[k]\big) - \ell_{\text{in}}\Big(\bigwedge\nolimits_{\mathscr{T}_{\text{in}}}\big\{\mathsf{leaf}[j], \mathsf{leaf}[k]\big\}\Big) \end{aligned}$$

This and (A2) imply that our claim is valid (for any j in $\{i+1, \ldots, \mathsf{n}\}$). The "only if" part of (b) follows from this and property P3.

To prove (c), let $\mathbf{v} \in \mathbf{Nodes}(\mathscr{T}_{\text{out}})$. Then $\mathsf{lastLeaf}_\sigma(\mathbf{v}) \in \mathbf{Leaves}(\mathscr{T}_{\text{out}}[\mathbf{v}])$ (by (b)), and every $\mathbf{w} \in \mathbf{Nodes}(\mathscr{T}_{\text{out}}[\mathbf{v}]) \subseteq \mathbf{Nodes}(\mathscr{T}_{\text{in}}[\mathbf{v}])$ satisfies $\ell_{\text{out}}(\mathbf{w}) = \ell_{\text{in}}(\mathbf{w}) \leq \ell_{\text{in}}(\mathsf{lastLeaf}_\sigma(\mathbf{v})) = \ell_{\text{out}}(\mathsf{lastLeaf}_\sigma(\mathbf{v}))$.

It follows that $\text{depth}_{\mathfrak{F}_{\text{out}}}(\mathbf{v}) = \ell_{\text{out}}(\mathsf{lastLeaf}_\sigma(\mathbf{v})) - \ell_{\text{out}}(\mathbf{v}) = \ell_{\text{in}}(\mathsf{lastLeaf}_\sigma(\mathbf{v})) - \ell_{\text{in}}(\mathbf{v}) = \text{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{v})$. $\square$

Lemma A4 *Let* $\mathfrak{F}_{\text{in}} = (\mathscr{T}_{\text{in}}, \ell_{\text{in}})$ *be a* κ*-FCTS, let* $\lambda > 0$, *and let* $\mathfrak{F}_{\text{out}} = (\mathscr{T}_{\text{out}}, \ell_{\text{out}})$ *be the* κ*-FCTS that results from pruning* $\mathfrak{F}_{\text{in}}$ *by removing branches of length* $\leq \lambda$ *using an* ℓ_{in}*-increasing leaf enumeration* $\sigma = (\mathsf{leaf}[1], \ldots, \mathsf{leaf}[\mathsf{n}])$ *of* $\mathbf{Leaves}(\mathscr{T}_{\text{in}})$. *Then*:

(a) $\mathbf{Nodes}(\mathscr{T}_{\text{out}}) \setminus \mathbf{Leaves}(\mathscr{T}_{\text{out}}) \supseteq \mathbf{U}^\lambda\langle\mathfrak{F}_{\text{in}}\rangle \supseteq \mathbf{Crit}(\mathscr{T}_{\text{out}}) \setminus \mathbf{Leaves}(\mathscr{T}_{\text{out}})$
(b) *For all* $\mathbf{v} \in \mathbf{V}^\lambda\langle\mathfrak{F}_{\text{in}}\rangle \setminus \mathbf{V}_1^\lambda\langle\mathfrak{F}_{\text{in}}\rangle$, $\mathbf{v}{\Uparrow}_{\mathscr{T}_{\text{in}}} \cap \mathbf{Nodes}(\mathscr{T}_{\text{out}}) = \emptyset$.
(c) *For all* $\mathbf{v} \in \mathbf{V}_1^\lambda\langle\mathfrak{F}_{\text{in}}\rangle$, $\mathbf{v}{\Uparrow}_{\mathscr{T}_{\text{in}}} \cap \mathbf{Nodes}(\mathscr{T}_{\text{out}}) = \mathsf{Path}_\sigma(\mathbf{v}, \mathscr{T}_{\text{in}})$.

Proof For brevity, we shall write $\mathbf{U}^\lambda$, $\mathbf{V}^\lambda$, $\mathbf{V}_1^\lambda$, $\mathsf{lastLeaf}_\sigma(\mathbf{v})$, and $\mathsf{Path}_\sigma(\mathbf{v})$ for $\mathbf{U}^\lambda\langle\mathfrak{F}_{\text{in}}\rangle$, $\mathbf{V}^\lambda\langle\mathfrak{F}_{\text{in}}\rangle$, $\mathbf{V}_1^\lambda\langle\mathfrak{F}_{\text{in}}\rangle$, $\mathsf{lastLeaf}_\sigma(\mathbf{v}, \mathscr{T}_{\text{in}})$, and $\mathsf{Path}_\sigma(\mathbf{v}, \mathscr{T}_{\text{in}})$.

First, we prove (a). The inclusion $\mathbf{U}^\lambda \supseteq \mathbf{Crit}(\mathscr{T}_{\text{out}}) \setminus \mathbf{Leaves}(\mathscr{T}_{\text{out}})$ follows from Corollary A2 and Lemma A3(c). Moreover, since P4 implies that $\mathbf{Leaves}(\mathscr{T}_{\text{out}}) \subseteq \mathbf{Leaves}(\mathscr{T}_{\text{in}})$, we have that $\mathbf{u} \notin \mathbf{Leaves}(\mathscr{T}_{\text{out}})$ if $\mathbf{u} \in \mathbf{U}^\lambda$. So the other inclusion of (a) will follow if we can show that $\mathbf{u} \in \mathbf{Nodes}(\mathscr{T}_{\text{out}})$ whenever $\mathbf{u} \in \mathbf{U}^\lambda$.

Let $\mathbf{u}$ be any element of $\mathbf{U}^\lambda$, and let $\mathsf{leaf}[i] = \mathsf{lastLeaf}_\sigma(\mathbf{u})$. If $i = \mathsf{n}$, then $\mathsf{lastLeaf}_\sigma(\mathbf{u}) \in \mathbf{Nodes}(\mathscr{T}_{\text{out}})$ (by property P2) and so $\mathbf{u} \in \mathbf{Nodes}(\mathscr{T}_{\text{out}})$ (because of P4), as required. Now suppose $i < \mathsf{n}$. Let j be any element of the set $\{i+1, \dots, \mathsf{n}\}$ (so $\mathsf{leaf}[j] \notin \mathbf{Leaves}(\mathscr{T}_{\text{in}}[\mathbf{u}])$). Since $\mathsf{leaf}[i]$ is a leaf of $\mathscr{T}_{\text{in}}[\mathbf{u}]$ but $\mathsf{leaf}[j]$ is not, we have that $\bigwedge_{\mathscr{T}_{\text{in}}}\{\mathsf{leaf}[j], \mathsf{leaf}[i]\} \prec_{\mathscr{T}_{\text{in}}} \mathbf{u}$. Hence:

$$\begin{aligned}\ell_{\text{in}}\big(\mathsf{leaf}[i]\big) - \ell_{\text{in}}\Big(\textstyle\bigwedge_{\mathscr{T}_{\text{in}}}\big\{\mathsf{leaf}[j], \mathsf{leaf}[i]\big\}\Big) &> \ell_{\text{in}}\big(\mathsf{leaf}[i]\big) - \ell_{\text{in}}(\mathbf{u})\\ &= \mathrm{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{u}) > \lambda\end{aligned}$$

We see from this and property P3 that $\mathsf{lastLeaf}_\sigma(\mathbf{u}) = \mathsf{leaf}[i] \in \mathbf{Leaves}(\mathscr{T}_{\text{out}})$, and hence (in view of P4) that $\mathbf{u} \in \mathbf{Nodes}(\mathscr{T}_{\text{out}})$, as required. This proves (a).

Next, we prove (b). Let $\mathbf{v}$ be any node in $\mathbf{V}^\lambda \setminus \mathbf{V}_1^\lambda$. Then it follows from the definitions of $\mathbf{V}^\lambda$ and $\mathbf{V}_1^\lambda$ that $\mathbf{v} \neq \mathbf{root}(\mathscr{T}_{\text{in}})$.

Let $\mathbf{p} = \mathbf{parent}_{\mathscr{T}_{\text{in}}}(\mathbf{v})$. Then $\mathbf{p} \in \mathbf{v}{\downarrow}_{\mathscr{T}_{\text{in}}} \subseteq \mathbf{U}^\lambda$, so we have that:

$$\ell_{\text{in}}\big(\mathsf{lastLeaf}_\sigma(\mathbf{p})\big) - \ell_{\text{in}}(\mathbf{p}) = \mathrm{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{p}) > \lambda \tag{A3}$$

Now $\ell_{\text{in}}(\mathbf{d}) - \ell_{\text{in}}(\mathbf{v}) \leq \mathrm{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{v})$ for all $\mathbf{d} \succeq_{\mathscr{T}_{\text{in}}} \mathbf{v}$. Therefore:

$$\ell_{\text{in}}(\mathbf{d}) - \ell_{\text{in}}(\mathbf{p}) \leq \mathrm{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{v}) + \ell_{\text{in}}(\mathbf{v}) - \ell_{\text{in}}(\mathbf{p}) \leq \lambda \quad \text{for all } \mathbf{d} \succeq_{\mathscr{T}_{\text{in}}} \mathbf{v} \tag{A4}$$

Here the second inequality follows from the definition of $\mathbf{V}_1^\lambda$ and the facts that $\mathbf{p} = \mathbf{parent}_{\mathscr{T}_{\text{in}}}(\mathbf{v})$ and $\mathbf{v} \in \mathbf{V}^\lambda \setminus \mathbf{V}_1^\lambda$. It follows from (A3) and (A4) that $\mathsf{lastLeaf}_\sigma(\mathbf{p})$ is not a descendant of $\mathbf{v}$ in $\mathscr{T}_{\text{in}}$, and so

$$\textstyle\bigwedge_{\mathscr{T}_{\text{in}}}\big\{\mathsf{lastLeaf}_\sigma(\mathbf{p}), \mathsf{lastLeaf}_\sigma(\mathbf{v})\big\} = \mathbf{p} \tag{A5}$$

Since $\mathsf{lastLeaf}_\sigma(\mathbf{v}) \succeq_{\mathscr{T}_{\text{in}}} \mathbf{v}$, we deduce from (A4) and (A5) that

$$\ell_{\text{in}}\big(\mathsf{lastLeaf}_\sigma(\mathbf{v})\big) - \ell_{\text{in}}\Big(\textstyle\bigwedge_{\mathscr{T}_{\text{in}}}\big\{\mathsf{lastLeaf}_\sigma(\mathbf{p}), \mathsf{lastLeaf}_\sigma(\mathbf{v})\big\}\Big) \leq \lambda \tag{A6}$$

Since $\mathbf{p} = \mathbf{parent}_{\mathscr{T}_{\text{in}}}(\mathbf{v})$ and $\mathsf{lastLeaf}_\sigma(\mathbf{p}) \neq \mathsf{lastLeaf}_\sigma(\mathbf{v})$ (e.g., by (A5)), the leaf $\mathsf{lastLeaf}_\sigma(\mathbf{p})$ must occur later in the ℓ_{in}-increasing enumeration σ than the leaf $\mathsf{lastLeaf}_\sigma(\mathbf{v})$. This, (A6), and P3 imply that $\mathsf{lastLeaf}_\sigma(\mathbf{v}) \notin \mathbf{Leaves}(\mathscr{T}_{\text{out}})$. It now follows from assertion (b) of Lemma A3 that $\mathbf{v} \notin \mathbf{Nodes}(\mathscr{T}_{\text{out}})$. This and assertion (a) of Lemma A3 imply $\mathbf{v}{\Uparrow}_{\mathscr{T}_{\text{in}}} \cap \mathbf{Nodes}(\mathscr{T}_{\text{out}}) = \emptyset$, which proves (b).

Finally, we prove (c). Let $\mathbf{v}$ be any node in $\mathbf{V}_1^\lambda$. We first make the claim that $\mathsf{lastLeaf}_\sigma(\mathbf{v})$ is a leaf of $\mathscr{T}_{\text{out}}$.

If $\mathbf{v} = \mathbf{root}(\mathscr{T}_{\text{in}})$ then the claim is certainly true (by property P2), so let us assume $\mathbf{v} \neq \mathbf{root}(\mathscr{T}_{\text{in}})$. Let $\mathbf{p} = \mathbf{parent}_{\mathscr{T}_{\text{in}}}(\mathbf{v})$, and let $\mathbf{s}$ be any leaf of $\mathscr{T}_{\text{in}}$ that occurs later in the ℓ_{in}-increasing enumeration σ than $\mathsf{lastLeaf}_\sigma(\mathbf{v})$. Then $\mathbf{s} \notin \mathbf{Leaves}(\mathscr{T}_{\text{in}}[\mathbf{v}])$, and so $\bigwedge_{\mathscr{T}_{\text{in}}}\{\mathbf{s}, \mathsf{lastLeaf}_\sigma(\mathbf{v})\} \preceq_{\mathscr{T}_{\text{in}}} \mathbf{p}$, which implies that:

$$\ell_{\text{in}}\big(\mathsf{lastLeaf}_\sigma(\mathbf{v})\big) - \ell_{\text{in}}\Big(\bigwedge\nolimits_{\mathscr{T}_{\text{in}}}\big\{\mathbf{s}, \mathsf{lastLeaf}_\sigma(\mathbf{v})\big\}\Big) \geq \ell_{\text{in}}\big(\mathsf{lastLeaf}_\sigma(\mathbf{v})\big) - \ell_{\text{in}}(\mathbf{p}) \tag{A7}$$

But, since $\text{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{v}) = \ell_{\text{in}}(\mathsf{lastLeaf}_\sigma(\mathbf{v})) - \ell_{\text{in}}(\mathbf{v})$, we also have that

$$\ell_{\text{in}}\big(\mathsf{lastLeaf}_\sigma(\mathbf{v})\big) - \ell_{\text{in}}(\mathbf{p}) = \text{depth}_{\mathfrak{F}_{\text{in}}}(\mathbf{v}) + \ell_{\text{in}}(\mathbf{v}) - \ell_{\text{in}}(\mathbf{p}) > \lambda \tag{A8}$$

where the inequality follows from the definition of $\mathbf{V}_1^\lambda$ and the facts that $\mathbf{p} = \mathbf{parent}_{\mathscr{T}_{\text{in}}}(\mathbf{v})$ and $\mathbf{v} \in \mathbf{V}_1^\lambda$. Now it follows from (A7) and (A8) that:

$$\ell_{\text{in}}\big(\mathsf{lastLeaf}_\sigma(\mathbf{v})\big) - \ell_{\text{in}}\Big(\bigwedge\nolimits_{\mathscr{T}_{\text{in}}}\big\{\mathbf{s}, \mathsf{lastLeaf}_\sigma(\mathbf{v})\big\}\Big) > \lambda$$

Since this is true for every leaf $\mathbf{s}$ of $\mathscr{T}_{\text{in}}$ that occurs later in the ℓ_{in}-increasing enumeration σ than $\mathsf{lastLeaf}_\sigma(\mathbf{v})$, our claim is justified (by property P3).

If $\mathbf{w}$ is any node in $\mathsf{Path}_\sigma(\mathbf{v})$, then $\mathbf{w} \in \mathsf{lastLeaf}_\sigma(\mathbf{v})\Downarrow_{\mathscr{T}_{\text{in}}}$ and so it follows from our claim (and P4) that $\mathbf{w} \in \mathbf{Nodes}(\mathscr{T}_{\text{out}})$. Thus every node in $\mathsf{Path}_\sigma(\mathbf{v})$ lies in $\mathbf{v}\Uparrow_{\mathscr{T}_{\text{in}}} \cap \mathbf{Nodes}(\mathscr{T}_{\text{out}})$.

It remains only to prove that $\mathbf{v}\Uparrow_{\mathscr{T}_{\text{in}}} \cap \mathbf{Nodes}(\mathscr{T}_{\text{out}}) \setminus \mathsf{Path}_\sigma(\mathbf{v}) = \emptyset$. To do this, we suppose there is a node $\mathbf{x} \in \mathbf{v}\Uparrow_{\mathscr{T}_{\text{in}}} \cap \mathbf{Nodes}(\mathscr{T}_{\text{out}}) \setminus \mathsf{Path}_\sigma(\mathbf{v})$ and deduce a contradiction. As $\mathbf{x} \in \mathbf{v}\Uparrow_{\mathscr{T}_{\text{in}}} \setminus \mathsf{Path}_\sigma(\mathbf{v})$, we have that $\mathbf{x} \notin \mathsf{lastLeaf}_\sigma(\mathbf{v})\Downarrow_{\mathscr{T}_{\text{in}}}$ and so $\mathsf{lastLeaf}_\sigma(\mathbf{v}) \neq \mathsf{lastLeaf}_\sigma(\mathbf{x})$. Moreover, each of the nodes $\mathsf{lastLeaf}_\sigma(\mathbf{x})$ and $\mathsf{lastLeaf}_\sigma(\mathbf{v})$ is a leaf of $\mathscr{T}_{\text{out}}$ (by Lemma A3(b) and our claim).

Let $\mathbf{c} = \bigwedge_{\mathscr{T}_{\text{out}}}\{\mathsf{lastLeaf}_\sigma(\mathbf{x}), \mathsf{lastLeaf}_\sigma(\mathbf{v})\}$. Then we have that $\mathbf{c} \in \mathbf{Crit}(\mathscr{T}_{\text{out}})$, $c \notin \mathbf{Leaves}(\mathscr{T}_{\text{out}})$, and $\mathbf{c} = \bigwedge_{\mathscr{T}_{\text{in}}}\{\mathsf{lastLeaf}_\sigma(\mathbf{x}), \mathsf{lastLeaf}_\sigma(\mathbf{v})\}$ (by assertion (i) of Lemma A1). The latter implies $\mathbf{c} \succeq_{\mathscr{T}_{\text{in}}} \mathbf{v}$ (as $\mathsf{lastLeaf}_\sigma(\mathbf{x}) \succeq_{\mathscr{T}_{\text{in}}} \mathbf{x} \succeq_{\mathscr{T}_{\text{in}}} \mathbf{v}$ and $\mathsf{lastLeaf}_\sigma(\mathbf{v}) \succeq_{\mathscr{T}_{\text{in}}} \mathbf{v}$); and $\mathbf{c} \succeq_{\mathscr{T}_{\text{in}}} \mathbf{v}$ implies $\text{depth}_{\mathfrak{F}_{\text{in}}}\mathbf{c} \leq \text{depth}_{\mathfrak{F}_{\text{in}}}\mathbf{v} \leq \lambda$ (where the second inequality follows from the fact that $\mathbf{v} \in \mathbf{V}_1^\lambda \subseteq \mathbf{V}^\lambda$). Hence $\mathbf{c} \notin \mathbf{U}^\lambda$. But this contradicts assertion (a) (since $\mathbf{c} \in \mathbf{Crit}(\mathscr{T}_{\text{out}}) \setminus \mathbf{Leaves}(\mathscr{T}_{\text{out}})$). It follows that $\mathbf{x}$ cannot exist, and so our proof of (c) is complete. $\square$

We can now prove the main result of Sect. 2.4.2:

Proposition *Let* $\mathfrak{F}_{\text{in}} = (\mathscr{T}_{\text{in}}, \ell_{\text{in}})$ *be a* κ*-FCTS, let* $\lambda > 0$*, and let* $\mathfrak{F}_{\text{out}} = (\mathscr{T}_{\text{out}}, \ell_{\text{out}})$ *be the* κ*-FCTS that results from pruning* $\mathfrak{F}_{\text{in}}$ *by removing branches of length* $\leq \lambda$ *using an* ℓ_{in}*-increasing enumeration* σ *of* $\mathbf{Leaves}(\mathscr{T}_{\text{in}})$*. Then the nodes of* $\mathfrak{F}_{\text{out}}$ *consist just of*:

(i) *The nodes of* $\mathbf{U}^\lambda\langle\mathfrak{F}_{\text{in}}\rangle$.
(ii) *The nodes of* $\mathsf{Path}_\sigma(\mathbf{v}, \mathscr{T}_{\text{in}})$ *for each node* $\mathbf{v}$ *in* $\mathbf{V}_1^\lambda\langle\mathfrak{F}_{\text{in}}\rangle$.

Proof As $\mathbf{U}^\lambda\langle\mathfrak{F}_{\text{in}}\rangle \subseteq \mathbf{Nodes}(\mathscr{T}_{\text{out}})$ by Lemma A4(a), on putting $\mathscr{T} = \mathscr{T}_{\text{in}}$ and $\mathfrak{F} = \mathfrak{F}_{\text{in}}$ in (2.1) and taking the intersection of each side with $\mathbf{Nodes}(\mathscr{T}_{\text{out}})$ we see that:

$$\mathbf{Nodes}(\mathscr{T}_{\text{out}}) = \mathbf{U}^\lambda\langle\mathfrak{F}_{\text{in}}\rangle \cup \bigcup_{\mathbf{v} \in \mathbf{V}^\lambda\langle\mathfrak{F}_{\text{in}}\rangle} \big(\mathbf{v}\Uparrow_{\mathscr{T}_{\text{in}}} \cap \mathbf{Nodes}(\mathscr{T}_{\text{out}})\big)$$

The proposition follows from this and assertions (b) and (c) of Lemma A4. $\square$

A.2 Properties of Simplification Step 3

Here we establish some properties of simplification step 3 that are used in our proof of the Main Theorem and our justification of Algorithm 1.

For all $j \in \{1, \ldots, |D(\mathfrak{F})|\}$, we see from E1–E5 that $\mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\delta\rangle) \subseteq \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\delta'\rangle)$ whenever $\delta \geq \delta'$. It follows that $\mathfrak{F}^{\mathbf{crit}}\langle\cdot\rangle$ has the following monotonicity property:

$$\mathfrak{F}^{\mathbf{crit}}\langle\delta\rangle \sqsubseteq \mathfrak{F}^{\mathbf{crit}}\langle\delta'\rangle \quad \text{whenever } \delta \geq \delta' \tag{A9}$$

In addition, $\mathfrak{F}^{\mathbf{crit}}\langle\cdot\rangle$ has the following four properties for every $\lambda > 0$ (as we will explain below):

E6: For every $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}) \setminus (\mathbf{Leaves}(\mathfrak{F}) \cup \{\mathbf{LCN}(\mathfrak{F})\} \cup \{\mathbf{root}(\mathfrak{F})\})$ and every $i \in \{0, \ldots, |D(\mathfrak{F})| - 1\}$, $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle d^{\mathfrak{F}}_{i+1}\rangle)$ if and only if, for every $j \in \{0, \ldots, i\}$, $\ell(\mathbf{c}) - \ell(\mathbf{parent}_{\mathfrak{F}^{\mathbf{crit}}\langle d^{\mathfrak{F}}_{j}\rangle}(\mathbf{c})) > d^{\mathfrak{F}}_{j+1}$.

E7: For every $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}) \setminus (\mathbf{Leaves}(\mathfrak{F}) \cup \{\mathbf{LCN}(\mathfrak{F})\} \cup \{\mathbf{root}(\mathfrak{F})\})$, $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle)$ if and only if there is no critical proper ancestor $\mathbf{c}'$ of $\mathbf{c}$ in $\mathfrak{F}$ such that $\ell(\mathbf{c}) - \ell(\mathbf{c}') \leq \lambda$ and $\mathbf{c}' \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle \mathrm{pred}_{\mathfrak{F}}(\ell(\mathbf{c}) - \ell(\mathbf{c}'))\rangle)$.

E8: For every $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}) \setminus (\mathbf{Leaves}(\mathfrak{F}) \cup \{\mathbf{LCN}(\mathfrak{F})\} \cup \{\mathbf{root}(\mathfrak{F})\})$, $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle)$ if $\ell(\mathbf{c}) - \ell(\mathbf{parent}_{\mathfrak{F}^{\mathbf{crit}}}(\mathbf{c})) > \lambda$.

E9: For every $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}) \setminus (\mathbf{Leaves}(\mathfrak{F}) \cup \{\mathbf{LCN}(\mathfrak{F})\} \cup \{\mathbf{root}(\mathfrak{F})\})$, if $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle)$ then $\ell(\mathbf{c}) - \ell(\mathbf{parent}_{\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle}(\mathbf{c})) > \lambda$.

Our proof of the correctness of Algorithm 1 will be based on property E7. However, E1–E3, E8, and E9 are the only properties of simplification step 3 that will be used in our proof of the Main Theorem.

E6 is easily deduced from E5 by induction on i. Now we establish E7–E9. Let $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}) \setminus (\mathbf{Leaves}(\mathfrak{F}) \cup \{\mathbf{LCN}(\mathfrak{F})\} \cup \{\mathbf{root}(\mathfrak{F})\})$, and let λ be any positive value. We first claim that, for any critical proper ancestor $\mathbf{c}'$ of $\mathbf{c}$ in $\mathfrak{F}$, the following four conditions are equivalent:

(a) There is some $j \in \{0, \ldots, |D(\mathfrak{F})| - 1\}$ such that $\ell(\mathbf{c}) - \ell(\mathbf{c}') \leq d^{\mathfrak{F}}_{j+1} \leq \lambda$ and $\mathbf{c}' \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle d^{\mathfrak{F}}_{j}\rangle)$.

(b) There is some $j \in \{0, \ldots, |D(\mathfrak{F})| - 1\}$ such that $\ell(\mathbf{c}) - \ell(\mathbf{c}') \leq d^{\mathfrak{F}}_{j+1} \leq \lambda$ and $\mathbf{c}' \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle \mathrm{pred}_{\mathfrak{F}}(\ell(\mathbf{c}) - \ell(\mathbf{c}'))\rangle)$.

(c) $\ell(\mathbf{c}) - \ell(\mathbf{c}') \leq \lambda$ and $\mathbf{c}' \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle \mathrm{pred}_{\mathfrak{F}}(\ell(\mathbf{c}) - \ell(\mathbf{c}'))\rangle)$.

(d) There is some $j \in \{0, \ldots, |D(\mathfrak{F})| - 1\}$ such that $\ell(\mathbf{c}) - \ell(\mathbf{c}') = d^{\mathfrak{F}}_{j+1} \leq \lambda$ and $\mathbf{c}' \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle d^{\mathfrak{F}}_{j}\rangle)$.

Here (a) implies (b) because of the monotonicity property (A9) and the fact that if $\ell(\mathbf{c}) - \ell(\mathbf{c}') \leq d^{\mathfrak{F}}_{j+1}$ then $\mathrm{pred}_{\mathfrak{F}}(\ell(\mathbf{c}) - \ell(\mathbf{c}')) \leq d^{\mathfrak{F}}_{j}$. Evidently, (b) implies (c), and (d) implies (a). For any critical proper ancestor $\mathbf{c}'$ of $\mathbf{c}$ in $\mathfrak{F}$, $\ell(\mathbf{c}) - \ell(\mathbf{c}') = d^{\mathfrak{F}}_{j+1}$ and $\mathrm{pred}_{\mathfrak{F}}(\ell(\mathbf{c}) - \ell(\mathbf{c}')) = d^{\mathfrak{F}}_{j}$ for some $j \in \{0, \ldots, |D(\mathfrak{F})| - 1\}$, and so (c) implies (d). This justifies our claim that (a)–(d) are equivalent.

Next, we observe that $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle)$ holds if and only if $\mathbf{c}$ satisfies $\ell(\mathbf{c}) - \ell(\mathbf{parent}_{\mathfrak{F}^{\mathbf{crit}}\langle d_j^{\mathfrak{F}}\rangle}(\mathbf{c})) > d_{j+1}^{\mathfrak{F}}$ for all $j \in \{0, \ldots, |D(\mathfrak{F})| - 1\}$ such that $d_{j+1}^{\mathfrak{F}} \leq \lambda$. (This follows from E6 when $\lambda \in D(\mathfrak{F})$. It remains true if $\lambda \notin D(\mathfrak{F})$, because of E4.) So $\mathbf{c} \notin \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle)$ just if there is some $j \in \{0, \ldots, |D(\mathfrak{F})| - 1\}$ such that $\ell(\mathbf{c}) - \ell(\mathbf{parent}_{\mathfrak{F}^{\mathbf{crit}}\langle d_j^{\mathfrak{F}}\rangle}(\mathbf{c})) \leq d_{j+1}^{\mathfrak{F}} \leq \lambda$. Thus $\mathbf{c} \notin \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle)$ just if (a) holds for some critical proper ancestor $\mathbf{c}'$ of $\mathbf{c}$ in $\mathfrak{F}$. Equivalently, $\mathbf{c} \notin \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle)$ just if (c) holds for some critical proper ancestor $\mathbf{c}'$ of $\mathbf{c}$ in $\mathfrak{F}$. This proves E7. E8 follows from the "if" part of E7.

Suppose the node $\mathbf{c}$ violated E9. Then $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle)$. Moreover, when $\mathbf{c}' = \mathbf{parent}_{\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle}(\mathbf{c})$ we would have that $\ell(\mathbf{c}) - \ell(\mathbf{c}') \leq \lambda$ and also that $\mathbf{c}' \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle \mathrm{pred}_{\mathfrak{F}}(\ell(\mathbf{c}) - \ell(\mathbf{c}'))\rangle)$, where the latter follows from the former, the fact that $\mathbf{c}' \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\lambda\rangle)$, and the monotonicity property (A9). But this would contradict the "only if" part of E7. So E9 holds.

A.3 Justification of Algorithm 1

The correctness of Algorithm 1 will be deduced from Lemma A5 and Corollary A6 below.

Let $\mathfrak{F} = (\mathscr{T}, \ell)$ be any κ-FCTS, and let $\mathbf{c}$ be any node of $\mathfrak{F}^{\mathbf{crit}}$. Then we define $\delta_\lambda(\mathbf{c}, \mathfrak{F}) = \infty$ if $\mathbf{c} \in \mathbf{Leaves}(\mathfrak{F}) \cup \{\mathbf{LCN}(\mathfrak{F})\} \cup \{\mathbf{root}(\mathfrak{F})\}$, and we define $\delta_\lambda(\mathbf{c}, \mathfrak{F}) = \ell(\mathbf{c}) - \ell(\mathbf{a}_\lambda(\mathbf{c}, \mathfrak{F}))$ otherwise, where $\mathbf{a}_\lambda(\mathbf{c}, \mathfrak{F})$ is the closest critical proper ancestor $\mathbf{c}'$ of $\mathbf{c}$ in $\mathfrak{F}$ such that

$$\begin{array}{ll} \textbf{either} & \ell(\mathbf{c}) - \ell(\mathbf{c}') > \lambda \\ \textbf{or} & \ell(\mathbf{c}) - \ell\big(\mathbf{c}'\big) \leq \lambda \text{ and } \mathbf{c}' \in \mathbf{Nodes}\big(\mathfrak{F}^{\mathbf{crit}}\big\langle \mathrm{pred}_{\mathfrak{F}}\big(\ell(\mathbf{c}) - \ell\big(\mathbf{c}'\big)\big)\big\rangle\big) \end{array}$$

$\mathbf{a}_\lambda(\mathbf{c}, \mathfrak{F})$ exists for all $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}) \setminus (\mathbf{Leaves}(\mathfrak{F}) \cup \{\mathbf{LCN}(\mathfrak{F})\} \cup \{\mathbf{root}(\mathfrak{F})\})$, because when $\mathbf{c}' = \mathbf{LCN}(\mathfrak{F})$ we see from E2 that $\mathbf{c}' \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\mu\rangle)$ for every $\mu \geq 0$ and so $\mathbf{c}'$ must satisfy the "**either**" or the "**or**" condition. Now $\delta_\lambda(\cdot, \mathfrak{F})$ satisfies the following condition:

Lemma A5 *Let $0 \leq \mu \leq \lambda$ and let $\mathfrak{F} = (\mathscr{T}, \ell)$ be any κ-FCTS. Then for all $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}})$ we have that $\delta_\lambda(\mathbf{c}, \mathfrak{F}) > \mu$ if and only if $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\mu\rangle)$.*

Proof Suppose $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}) \setminus (\mathbf{Leaves}(\mathfrak{F}) \cup \{\mathbf{LCN}(\mathfrak{F})\} \cup \{\mathbf{root}(\mathfrak{F})\})$. Then $\delta_\lambda(\mathbf{c}, \mathfrak{F}) > \mu$ holds just if $\ell(\mathbf{c}) - \ell(\mathbf{a}_\lambda(\mathbf{c}, \mathfrak{F})) > \mu$, and since $\mu \leq \lambda$ we see from the definition of $\mathbf{a}_\lambda(\mathbf{c}, \mathfrak{F})$ that this holds just if no critical proper ancestor $\mathbf{c}'$ of $\mathbf{c}$ in $\mathfrak{F}$ satisfies $\ell(\mathbf{c}) - \ell(\mathbf{c}') \leq \mu$ and $\mathbf{c}' \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle \mathrm{pred}_{\mathfrak{F}}(\ell(\mathbf{c}) - \ell(\mathbf{c}'))\rangle)$. So in this case the lemma follows from E7.

The lemma also holds if $\mathbf{c} \in \mathbf{Leaves}(\mathfrak{F}) \cup \{\mathbf{LCN}(\mathfrak{F})\} \cup \{\mathbf{root}(\mathfrak{F})\}$, because in that case $\delta_\lambda(\mathbf{c}, \mathfrak{F}) = \infty > \mu$ and E1–E3 imply $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle\mu\rangle)$. $\square$

Corollary A6 *Let λ be any positive value, let $\mathfrak{F} = (\mathscr{T}, \ell)$ be any κ-FCTS, and let* $\mathbf{c} \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}) \setminus (\mathbf{Leaves}(\mathfrak{F}) \cup \{\mathbf{LCN}(\mathfrak{F})\} \cup \{\mathbf{root}(\mathfrak{F})\})$. *Then* $\delta_\lambda(\mathbf{c}, \mathfrak{F}) = \ell(\mathbf{c}) - \ell(\mathbf{a})$, *where* $\mathbf{a}$ *is the closest critical proper ancestor* $\mathbf{c}'$ *of* $\mathbf{c}$ *in* $\mathfrak{F}$ *such that*

$$\begin{array}{ll} \textbf{\textit{either}} & \ell(\mathbf{c}) - \ell\big(\mathbf{c}'\big) > \lambda \\ \textbf{\textit{or}} & \ell(\mathbf{c}) - \ell\big(\mathbf{c}'\big) \le \lambda \textit{ and } \ell(\mathbf{c}) - \ell\big(\mathbf{c}'\big) \le \delta_\lambda\big(\mathbf{c}', \mathfrak{F}\big) \end{array}$$

Proof We just have to show that $\mathbf{a} = \mathbf{a}_\lambda(\mathbf{c}, \mathfrak{F})$. The definition of $\mathbf{a}_\lambda(\mathbf{c}, \mathfrak{F})$ differs from the definition of $\mathbf{a}$ only in the **or** condition "$\ell(\mathbf{c}) - \ell(\mathbf{c}') \le \lambda$ and $\mathbf{c}' \in \mathbf{Nodes}(\mathfrak{F}^{\mathbf{crit}}\langle \mathrm{pred}_{\mathfrak{F}}(\ell(\mathbf{c}) - \ell(\mathbf{c}'))\rangle)$".

On putting $\mu = \mathrm{pred}_{\mathfrak{F}}(\ell(\mathbf{c}) - \ell(\mathbf{c}'))$ in Lemma A5, we see that this condition holds if and only if $\ell(\mathbf{c}) - \ell(\mathbf{c}') \le \lambda$ and $\mathrm{pred}_{\mathfrak{F}}(\ell(\mathbf{c}) - \ell(\mathbf{c}')) < \delta_\lambda(\mathbf{c}', \mathfrak{F})$, which is equivalent to the **or** condition in the definition of $\mathbf{a}$ (because either $\delta_\lambda(\mathbf{c}', \mathfrak{F}) = \ell(\mathbf{c}) - \ell(\mathbf{a}_\lambda(\mathbf{c}', \mathfrak{F})) \in D(\mathfrak{F})$ or $\delta_\lambda(\mathbf{c}', \mathfrak{F}) = \infty$). So $\mathbf{a} = \mathbf{a}_\lambda(\mathbf{c}, \mathfrak{F})$, as required. ☐

We can now explain why Algorithm 1 is correct. The algorithm sets $(\mathscr{T}, \ell)$ to a clone of $\mathfrak{F}_{\mathrm{in}}^{\mathbf{crit}} = (\mathscr{T}_{\mathrm{in}}^{\mathbf{crit}}, \ell_{\mathrm{in}}^{\mathbf{crit}})$. Writing $\mathfrak{F}$ for $(\mathscr{T}, \ell)$, we claim that the label $\mathbf{c}$.label given by the algorithm to each node $\mathbf{c}$ of $\mathfrak{F} = \mathfrak{F}^{\mathbf{crit}}$ is just the value $\delta_\lambda(\mathbf{c}, \mathfrak{F})$. Assuming this claim is valid, the correctness of the algorithm follows from Lemma A5. So it remains only to verify the claim.

The claim is certainly valid if $\mathbf{c}$ is $\mathbf{root}(\mathfrak{F})$ or $\mathbf{LCN}(\mathfrak{F})$, because those nodes are given the label ∞.

We see that the algorithm does a top-down traversal of $\mathscr{T}[\mathbf{LCN}(\mathfrak{F})]$, during which the procedure `labelDescendants` is executed once for each proper descendant $\mathbf{c}$ of $\mathbf{LCN}(\mathfrak{F})$ in $\mathfrak{F}$. When `labelDescendants` is executed for such a node $\mathbf{c}$ that is a leaf, it gives $\mathbf{c}$ the label ∞. So the claim is valid for each proper descendant $\mathbf{c}$ of $\mathbf{LCN}(\mathfrak{F})$ that is a leaf.

When `labelDescendants` is executed for a proper descendant $\mathbf{c}$ of $\mathbf{LCN}(\mathfrak{F})$ that is not a leaf, the **repeat** loop in the procedure is executed. It follows from Corollary A6 that this loop labels $\mathbf{c}$ with the value $\delta_\lambda(\mathbf{c}, \mathfrak{F})$. (Note that, when the loop is executed, $\mathbf{c}'$.label $= \delta_\lambda(\mathbf{c}', \mathfrak{F})$ for each proper ancestor $\mathbf{c}'$ of $\mathbf{c}$ in $\mathfrak{F}$.) Therefore the claim is also valid for each proper descendant $\mathbf{c}$ of $\mathbf{LCN}(\mathfrak{F})$ that is not a leaf.

Thus the claim is valid for all nodes $\mathbf{c}$ of $\mathfrak{F} = \mathfrak{F}^{\mathbf{crit}}$, and Algorithm 1 is correct.

Appendix B: A Constructive Proof of Theorem 1

For any adjacency relation κ, any image I whose domain is finite and κ-connected, any $\lambda > 0$, and any integer $k \ge 0$, let us say that the image I is (λ, k)-*good* with respect to κ if $\Lambda_\kappa(I) > \lambda$ and $K_\kappa(I) > k$. Also, let us say that an image I' is an ε-*perturbation* of an image I if I' has the same domain as I and $\|I' - I\|_\infty \le \varepsilon$. Then Theorem 1 can be deduced from the following lemma:

Fundamental Lemma *Let κ be any adjacency relation and $I_{\mathbf{good}} : \mathcal{S} \to \mathbb{R}$ an image whose domain $\mathcal{S}$ is finite and κ-connected. Let ε be a positive value, let k be a nonnegative integer for which $I_{\mathbf{good}}$ is $(4\varepsilon, k)$-good with respect to κ, and let I' be an ε-perturbation of $I_{\mathbf{good}}$. Then there is an essential isomorphism of $\mathbf{FCTS}_\kappa(I_{\mathbf{good}})$ to the $(2\varepsilon, k)$-simplification of $\mathbf{FCTS}_\kappa(I')$ that is level-preserving to within ε.*

Proof of Theorem 1, assuming the Fundamental Lemma is valid Suppose I, λ, and k satisfy the hypotheses of Theorem 1, so that $0 < \lambda < \Lambda_\kappa(I)/2$ and $0 \le k < K_\kappa(I)$. Let I' be any image that satisfies the conditions stated in the theorem (i.e., let I' be any image whose domain is the same as that of I and which satisfies the condition $\|I' - I\|_\infty \le \lambda/2$). Then we need to show that the conclusion of Theorem 1 holds—i.e., that there is an essential isomorphism of the (λ, k)-simplification of $\mathbf{FCTS}_\kappa(I')$ to $\mathbf{FCTS}_\kappa(I)$ that is level-preserving to within $\lambda/2$. We now deduce this from the Fundamental Lemma.

Let $I_{\mathbf{good}} = I$, and let $\varepsilon = \lambda/2$. Then $4\varepsilon = 2\lambda < \Lambda_\kappa(I) = \Lambda_\kappa(I_{\mathbf{good}})$ and $k < K_\kappa(I) = K_\kappa(I_{\mathbf{good}})$, so that $I_{\mathbf{good}}$ is $(4\varepsilon, k)$-good with respect to κ. We also have that $\|I' - I_{\mathbf{good}}\|_\infty = \|I' - I\|_\infty \le \lambda/2 = \varepsilon$, so that I' is a ε-perturbation of $I_{\mathbf{good}}$. Thus $I_{\mathbf{good}} = I$ and I' satisfy the hypotheses of the Fundamental Lemma, and must therefore satisfy the conclusion of the lemma, which implies the conclusion of Theorem 1 since $2\varepsilon = \lambda$. $\square$

We now prove the Fundamental Lemma by constructing an explicit essential isomorphism of $\mathbf{FCTS}_\kappa(I_{\mathbf{good}})$ to the $(2\varepsilon, k)$-simplification of $\mathbf{FCTS}_\kappa(I')$ that is level-preserving to within ε.

Let $\mathfrak{F}_{\mathbf{good}} = (\mathscr{T}_{\mathbf{good}}, \ell_{\mathbf{good}}) = \mathbf{FCTS}_\kappa(I_{\mathbf{good}})$, and let $\mathfrak{F}' = (\mathscr{T}', \ell') = \mathbf{FCTS}_\kappa(I')$. Let $\mathfrak{F}_1 = (\mathscr{T}_1, \ell_1)$ be the κ-FCTS that results from pruning $\mathfrak{F}'$ by removing nodes of size $\le k$, and let I_1 be the image $I_{\mathfrak{F}_1}$, so that $\mathfrak{F}_1 = \mathbf{FCTS}_\kappa(I_1)$. Let $\mathfrak{F}_2 = (\mathscr{T}_2, \ell_2)$ be the κ-FCTS that results from pruning $\mathfrak{F}_1$ by removing branches of length $\le 2\varepsilon$, and let $\mathfrak{F}_3 = (\mathscr{T}_3, \ell_3)$ be the κ-FCTS that results from eliminating internal edges of length $\le 2\varepsilon$ from $\mathfrak{F}_2^{\mathbf{crit}}$. Then $\mathfrak{F}_3 = (\mathscr{T}_3, \ell_3)$ is the $(2\varepsilon, k)$-simplification of $\mathbf{FCTS}_\kappa(I')$, so what we want to do is to construct an essential isomorphism of $\mathfrak{F}_{\mathbf{good}}$ to $\mathfrak{F}_3$ that is level-preserving to within ε. We will do this in three steps:

Step 1: We define a suitable mapping $\phi : \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}}) \to \mathbf{Leaves}(\mathscr{T}_1)$.

Step 2: We show that ϕ is 1-to-1, and that the range of the mapping ϕ is exactly the set of all the leaves of the subtree $\mathscr{T}_2$ of $\mathscr{T}_1$. Thereafter, we regard ϕ as a bijection $\phi : \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}}) \to \mathbf{Leaves}(\mathscr{T}_2)$.

Step 3: We extend ϕ to a mapping $\varphi : \mathbf{Crit}(\mathscr{T}_{\mathbf{good}}) \to \mathbf{Crit}(\mathscr{T}_2)$ by defining $\varphi(\mathbf{u}) = \bigwedge_{\mathscr{T}_2} \phi[\mathbf{Leaves}(\mathscr{T}_{\mathbf{good}}[\mathbf{u}])]$. We then establish that, for all $\mathbf{u}, \mathbf{u}' \in \mathbf{Crit}(\mathscr{T}_{\mathbf{good}})$, $\varphi(\mathbf{u}) \preceq_{\mathscr{T}_2} \varphi(\mathbf{u}')$ if and only if $\mathbf{u} \preceq_{\mathscr{T}_{\mathbf{good}}} \mathbf{u}'$, so that φ is 1-to-1 and order-preserving. We also show that the range of φ is the subset $\mathbf{Crit}(\mathscr{T}_3)$ of $\mathbf{Crit}(\mathscr{T}_2)$, and that $|\ell_3(\varphi(\mathbf{u})) - \ell_{\mathbf{good}}(\mathbf{u})| \le \varepsilon$ for every $\mathbf{u} \in \mathbf{Crit}(\mathscr{T}_{\mathbf{good}})$. Hence we can regard φ as a mapping $\varphi : \mathbf{Crit}(\mathscr{T}_{\mathbf{good}}) \to \mathbf{Crit}(\mathscr{T}_3)$ and, when so regarded, φ is an essential isomorphism of $\mathfrak{F}_{\mathbf{good}}$ to $\mathfrak{F}_3$ that is level-preserving to within ε.

Note that the extension of ϕ to φ in step 3 is very natural because, if $\mathscr{T}$ is any rooted tree and $\mathbf{u} \in \mathbf{Crit}(\mathscr{T})$, then $\mathbf{u} = \bigwedge_{\mathscr{T}} \mathbf{Leaves}(\mathscr{T}[\mathbf{u}])$. (In fact $\mathbf{u} \in \mathbf{Crit}(\mathscr{T})$ if and only if $\mathbf{u} \in \mathbf{Nodes}(\mathscr{T})$ and $\mathbf{u} = \bigwedge_{\mathscr{T}} \mathbf{Leaves}(\mathscr{T}[\mathbf{u}])$.)

B.1 Step 1 of the Proof of the Fundamental Lemma

We begin by defining a class of symmetric and transitive relations (on spels) that will be used in our definition of the mapping ϕ.

If $I : \mathcal{S} \to \mathbb{R}$ is an image and $\tau \in \mathbb{R}$, then we write $s \Leftarrow_{I \geq \tau} \Rightarrow t$ to mean that $s, t \in \mathcal{S}$ and $t \in \mathcal{C}_\kappa(s, I, \tau)$. It is readily confirmed that $\Leftarrow_{I \geq \tau} \Rightarrow$ is a symmetric and transitive relation (which depends on κ), and that $s \Leftarrow_{I \geq \tau} \Rightarrow s$ if and only if $I(s) \geq \tau$. Moreover, if $s \Leftarrow_{I \geq \tau_1} \Rightarrow t$ and $t \Leftarrow_{I \geq \tau_2} \Rightarrow u$ then $s \Leftarrow_{I \geq \min(\tau_1, \tau_2)} \Rightarrow u$.

Now let $\mathcal{C}_\kappa(v, I_{\mathbf{good}})$ be any leaf of $\mathscr{T}_{\mathbf{good}}$, and let z be any spel such that

$$z \in \operatorname*{arg\,min}_{u \Leftarrow_{I_{\mathbf{good}} \geq I_{\mathbf{good}}(v) - 2\varepsilon} \Rightarrow v} I_1(u) \tag{B1}$$

It follows from (B1) that:

$$\mathcal{C}_\kappa(z, I_1) \supseteq \left\{ u \;\middle|\; u \Leftarrow_{I_{\mathbf{good}} \geq I_{\mathbf{good}}(v) - 2\varepsilon} \Rightarrow v \right\} = \mathcal{C}_\kappa\left(v, I_{\mathbf{good}}, I_{\mathbf{good}}(v) - 2\varepsilon\right) \tag{B2}$$

Next, we define:

$$\mathcal{M}\big(\mathcal{C}_\kappa(v, I_{\mathbf{good}})\big) = \mathbf{Leaves}\big(\mathscr{T}_1\big[\mathcal{C}_\kappa(z, I_1)\big]\big) \tag{B3}$$

The set $\mathcal{M}(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$ is well defined by (B3) for the following reasons. First, if v' is any spel such that $\mathcal{C}_\kappa(v', I_{\mathbf{good}}) = \mathcal{C}_\kappa(v, I_{\mathbf{good}})$ (so that $I_{\mathbf{good}}(v') = I_{\mathbf{good}}(v)$) then the condition obtained from (B1) when we replace v with v' is equivalent to (B1). Second, if z' is any spel that belongs to the set in (B1), then $\mathcal{C}_\kappa(z', I_1) = \mathcal{C}_\kappa(z, I_1)$ (since $I_1(z') = I_1(z)$, and (B2) implies $z' \in \mathcal{C}_\kappa(z, I_1)$).

We can now define the mapping $\phi : \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}}) \to \mathbf{Leaves}(\mathscr{T}_1)$ by defining $\phi(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$ to be the element of $\mathcal{M}(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$ that occurs later in the ℓ_1-increasing leaf enumeration that is used in pruning $(\mathscr{T}_1, \ell_1)$ (to produce $(\mathscr{T}_2, \ell_2)$) than all other elements of $\mathcal{M}(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$. Note that if $\mathcal{M}(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$ has just one element, then $\phi(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$ is that element.

This completes step 1 of the proof of the Fundamental Lemma.

B.2 Some Useful Observations

Steps 2 and 3 of the proof of the Fundamental Lemma will be based on the following observations:

A. If $(\mathscr{T}, \ell) = \mathbf{FCTS}_\kappa(I)$, where I is an arbitrary image whose domain is finite and κ-connected, and $\emptyset \neq \mathbf{S} \subseteq \mathbf{Nodes}(\mathscr{T})$, then $\ell(\bigwedge_{\mathscr{T}} \mathbf{S})$ is the greatest real value τ such that $s \Leftarrow_{I \geq \tau} \Rightarrow t$ for all spels $s, t \in \bigcup \mathbf{S}$.

B. Whenever $\emptyset \neq \mathbf{L} \subsetneq \mathbf{L}' \subseteq \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})$ and $\bigwedge_{\mathscr{T}_{\mathbf{good}}} \mathbf{L}' \neq \bigwedge_{\mathscr{T}_{\mathbf{good}}} \mathbf{L}$, we have that $\ell_{\mathbf{good}}(\bigwedge_{\mathscr{T}_{\mathbf{good}}} \mathbf{L}') < \ell_{\mathbf{good}}(\bigwedge_{\mathscr{T}_{\mathbf{good}}} \mathbf{L}) - 4\varepsilon$.

C. If $v \in \mathbf{v} \in \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})$, $\mathbf{u} \in \mathbf{Nodes}(\mathscr{T}_{\mathbf{good}})$, and $\mathbf{v} \not\preceq_{\mathscr{T}_{\mathbf{good}}} \mathbf{u}$, then we have that $\ell_{\mathbf{good}}(\bigwedge_{\mathscr{T}_{\mathbf{good}}} \{\mathbf{u}, \mathbf{v}\}) < \ell_{\mathbf{good}}(\mathbf{v}) - 4\varepsilon = I_{\mathbf{good}}(v) - 4\varepsilon$.

D. If $\mathcal{C}_\kappa(v, I_{\mathbf{good}}) \in \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})$ and $u \Leftarrow_{I_{\mathbf{good}} \geq I_{\mathbf{good}}(v)-4\varepsilon} \Rightarrow v$, then we have that $u \Leftarrow_{I_{\mathbf{good}} \geq I_{\mathbf{good}}(u)} \Rightarrow v$ or, equivalently, $\mathcal{C}_\kappa(u, I_{\mathbf{good}}) \supseteq \mathcal{C}_\kappa(v, I_{\mathbf{good}})$.

E. If $\mathcal{C}_\kappa(x, I_1) \in \mathbf{Leaves}(\mathscr{T}_1)$, then $\mathcal{C}_\kappa(x, I_1) \in \mathbf{Leaves}(\mathscr{T}_2)$ if and only if there is no node $\mathcal{C}_\kappa(y, I_1) \in \mathbf{Leaves}(\mathscr{T}_1)$ that satisfies both of the following conditions:
   (i) $x \Leftarrow_{I_1 \geq I_1(x)-2\varepsilon} \Rightarrow y$
   (ii) The leaf $\mathcal{C}_\kappa(y, I_1)$ occurs later in the ℓ_1-increasing leaf enumeration that is used in pruning $(\mathscr{T}_1, \ell_1)$ to produce $(\mathscr{T}_2, \ell_2)$ than the leaf $\mathcal{C}_\kappa(x, I_1)$.

Here A is a consequence of the definitions of $\mathbf{FCTS}_\kappa(I)$ and $\bigwedge_{\mathscr{T}} \mathbf{S}$. (The special case of A in which $\mathbf{S} \subseteq \mathbf{Leaves}(\mathscr{T})$ is of particular interest; note that in this case $s \in \bigcup \mathbf{S}$ if and only if $\mathcal{C}_\kappa(s, I) \in \mathbf{S}$.) B is a consequence of the fact that $\Lambda_\kappa(I_{\mathbf{good}}) > 4\varepsilon$, C can be deduced from B by putting $\mathbf{L} = \{\mathbf{v}\}$ and $\mathbf{L}' = \{\mathbf{v}\} \cup \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}}[\mathbf{u}])$, and D can be deduced from A and C.

Assertion E is a consequence of A and the fact that $(\mathscr{T}_2, \ell_2)$ is the result of pruning $(\mathscr{T}_1, \ell_1)$ by removing branches of length $\leq 2\varepsilon$. In view of assertion (ii) of Lemma A1, we also have the following related fact:

E′. $\ell_1(\bigwedge_{\mathscr{T}_1} \{\mathbf{z}, \mathbf{z}'\}) < \min(\ell_1(\mathbf{z}), \ell_1(\mathbf{z}')) - 2\varepsilon$ whenever $\mathbf{z}$ and $\mathbf{z}'$ are distinct leaves of $\mathscr{T}_2$.

We could of course replace ℓ_1 with ℓ_2 in E′. Moreover, in view of assertion (i) of Lemma A1, we could also replace $\bigwedge_{\mathscr{T}_1}$ with $\bigwedge_{\mathscr{T}_2}$.

Now let x be any spel in $\mathcal{S}$. As $\mathfrak{F}_1$ is the result of pruning $\mathbf{FCTS}_\kappa(I') = (\mathscr{T}', \ell')$ by removing nodes of size $\leq k$, and $I_1 = I_{\mathfrak{F}_1}$, we see from the definition of $I_{\mathfrak{F}_1}$ that $I_1(x) = \max\{\ell'(\mathbf{u}) \mid \mathbf{u} \in \mathbf{Nodes}(\mathscr{T}'), |\mathbf{u}| \geq k+1, \text{ and } x \in \mathbf{u}\}$. This is equivalent to

$$I_1(x) = \max\left\{I'(y) \;\middle|\; y \in \mathcal{S},\ x \in \mathcal{C}_\kappa\left(y, I'\right), \text{ and } \left|\mathcal{C}_\kappa\left(y, I'\right)\right| \geq k+1\right\} \tag{B4}$$

since the nodes $\mathbf{u} \in \mathbf{Nodes}(\mathscr{T}')$ for which $x \in \mathbf{u}$ are just the sets $\mathcal{C}_\kappa(y, I')$ for which $x \in \mathcal{C}_\kappa(y, I')$. Now we claim that:

$$I_1(x) = \max\left\{\tau \;\middle|\; \left|\mathcal{C}_\kappa\left(x, I', \tau\right)\right| \geq k+1\right\} \tag{B5}$$

To see this, we first observe that if y satisfies $x \in \mathcal{C}_\kappa(y, I')$ then y also satisfies $\mathcal{C}_\kappa(y, I') = \mathcal{C}_\kappa(x, I', I'(y))$. It follows from this observation that each element of the set $\{I'(y) \mid y \in \mathcal{S}, x \in \mathcal{C}_\kappa(y, I'), \text{ and } |\mathcal{C}_\kappa(y, I')| \geq k+1\}$ in (B4) belongs to the set $\{I'(y) \mid y \in \mathcal{S} \text{ and } |\mathcal{C}_\kappa(x, I', I'(y))| \geq k+1\}$ and therefore belongs to the set $\{\tau \mid |\mathcal{C}_\kappa(x, I', \tau)| \geq k+1\}$ in our claim (B5). So the right side of (B5) is no less than the right side of (B4); it remains to show that it is no greater.

For every $\tau \leq I'(x)$, let $y(\tau, x)$ be any spel in $\arg\min_{s \in \mathcal{C}_\kappa(x, I', \tau)} I'(s)$, so that $I'(y(\tau, x)) \geq \tau$, and it is easy to see that

$$\mathcal{C}_\kappa\left(y(\tau, x), I'\right) = \mathcal{C}_\kappa\left(x, I', \tau\right) \tag{B6}$$

since $I' \geq I'(y(\tau, x))$ at every spel in $\mathcal{C}_\kappa(x, I', \tau)$. Now if τ_0 is any element of the set $\{\tau \mid |\mathcal{C}_\kappa(x, I', \tau)| \geq k+1\}$, then we have that $I'(y(\tau_0, x)) \geq \tau_0$ and we see from

(B6) that $|\mathcal{C}_\kappa(y(\tau_0, x), I')| \geq k+1$ and $x \in \mathcal{C}_\kappa(y(\tau_0, x), I')$, so that $I'(y(\tau_0, x))$ is an element of $\{I'(y) \mid y \in \mathcal{S}, x \in \mathcal{C}_\kappa(y, I'),$ and $|\mathcal{C}_\kappa(y, I')| \geq k+1\}$ that is no less than τ_0. This shows that the right side of (B4) is no less than the right side of (B5). Hence the right sides of (B4) and (B5) are equal, and so our claim (B5) follows from (B4).

Next, we establish the following properties of I_1:

F. I_1 is an ε-perturbation of $\mathbf{I_{good}}$, and if $(I_\mathrm{a}, I_\mathrm{b}) = (I_1, \mathbf{I_{good}})$ or $(\mathbf{I_{good}}, I_1)$ then for any $\tau, \delta \in \mathbb{R}$ and any spels $s, t, u \in \mathcal{S}$ we have that:
   (i) If $s \Leftarrow I_\mathrm{a} \geq \tau \Rightarrow t$ then $s \Leftarrow I_\mathrm{b} \geq \tau - \varepsilon \Rightarrow t$.
   (ii) If $s \Leftarrow I_\mathrm{a} \geq I_\mathrm{a}(u) - \delta \Rightarrow t$ then $s \Leftarrow I_\mathrm{b} \geq I_\mathrm{b}(u) - \delta - 2\varepsilon \Rightarrow t$.

To see that I_1 has these properties, let x be any spel in $\mathcal{S}$ and note that $\mathcal{C}_\kappa(x, \mathbf{I_{good}}, \tau) \subseteq \mathcal{C}_\kappa(x, I', \tau - \varepsilon)$ for every $\tau \in \mathbb{R}$ since $\|I' - \mathbf{I_{good}}\|_\infty \leq \varepsilon$. On putting $\tau = \mathbf{I_{good}}(x)$, we deduce that $\mathcal{C}_\kappa(x, I', \mathbf{I_{good}}(x) - \varepsilon) \supseteq \mathcal{C}_\kappa(x, \mathbf{I_{good}}, \mathbf{I_{good}}(x)) = \mathcal{C}_\kappa(x, \mathbf{I_{good}})$, whence $|\mathcal{C}_\kappa(x, I', \mathbf{I_{good}}(x) - \varepsilon)| \geq |\mathcal{C}_\kappa(x, \mathbf{I_{good}})| \geq k+1$ (as $K_\kappa(\mathbf{I_{good}}) > k$). It follows from this and (B5) that $I_1(x) \geq \mathbf{I_{good}}(x) - \varepsilon$. On the other hand, whenever $\tau > \mathbf{I_{good}}(x) + \varepsilon$ we have that $I'(x) < \tau$ (as $\|I' - \mathbf{I_{good}}\|_\infty \leq \varepsilon$), which implies that $|\mathcal{C}_\kappa(x, I', \tau)| = 0$ and hence (by (B5)) that $I_1(x) < \tau$. From this it follows that $I_1(x) \leq \mathbf{I_{good}}(x) + \varepsilon$. This shows that I_1 is an ε-perturbation of $\mathbf{I_{good}}$, as F asserts. Now (i) follows immediately, and (ii) can be deduced from (i) by putting $\tau = I_\mathrm{a}(u) - \delta$, since the fact that I_a is an ε-perturbation of I_b implies that $I_\mathrm{a}(u) - \delta \geq I_\mathrm{b}(u) - \delta - \varepsilon$ for every $u \in \mathcal{S}$.

B.3 Step 2 of the Proof of the Fundamental Lemma

The main goals of this step are to show that the mapping ϕ defined in step 1 of the proof is 1-to-1 and that the range of ϕ is exactly the subset $\mathbf{Leaves}(\mathcal{T}_2)$ of $\mathbf{Leaves}(\mathcal{T}_1)$. This will allow us to regard ϕ as a bijection $\phi : \mathbf{Leaves}(\mathcal{T}_\mathbf{good}) \to \mathbf{Leaves}(\mathcal{T}_2)$.

We first state and prove the following easy lemma:

Lemma B1 *Let $\mathcal{C}_\kappa(v, \mathbf{I_{good}})$ be any leaf of $\mathcal{T}_\mathbf{good}$, let x be any spel in $\mathcal{S}$ that satisfies $x \Leftarrow \mathbf{I_{good}} \geq \mathbf{I_{good}}(v) - 2\varepsilon \Rightarrow v$, and let* $\mathbf{s}$ *be any leaf of $\mathcal{T}_1$ such that $\mathbf{s} \succeq_{\mathcal{T}_1} \mathcal{C}_\kappa(x, I_1)$. Then $\mathbf{s} \in \mathcal{M}(\mathcal{C}_\kappa(v, \mathbf{I_{good}}))$.*

Proof Let z be a spel that satisfies (B1) with respect to v. Then (B2) implies that $x \in \mathcal{C}_\kappa(z, I_1)$ and hence that $\mathcal{C}_\kappa(x, I_1) \succeq_{\mathcal{T}_1} \mathcal{C}_\kappa(z, I_1)$. This and (B3) imply $\mathbf{s} \in \mathcal{M}(\mathcal{C}_\kappa(v, \mathbf{I_{good}}))$. □

Next, we establish the following properties of $\mathcal{M}$ and the mapping ϕ:

G. The following are true for any leaf $\mathcal{C}_\kappa(v, \mathbf{I_{good}})$ of $\mathcal{T}_\mathbf{good}$:
   (a) If $\mathcal{C}_\kappa(y, I_1) \in \mathcal{M}(\mathcal{C}_\kappa(v, \mathbf{I_{good}}))$, then:

   (i) $y \Leftarrow I_{\mathbf{good}} \geq I_{\mathbf{good}}(v)-4\varepsilon \Rightarrow v$
   (ii) $y \Leftarrow I_{\mathbf{good}} \geq I_{\mathbf{good}}(y) \Rightarrow v$
   (iii) $y \Leftarrow I_1 \geq I_1(y)-2\varepsilon \Rightarrow v$

(b) If $\mathcal{C}_\kappa(y, I_1) = \phi(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$, then:
   (i) $I_{\mathbf{good}}(v) + \varepsilon \geq I_1(y) \geq I_1(v) \geq I_{\mathbf{good}}(v) - \varepsilon$
   (ii) $y \Leftarrow I_{\mathbf{good}} \geq I_{\mathbf{good}}(v)-2\varepsilon \Rightarrow v$
   (iii) $\mathcal{C}_\kappa(y, I_1) \in \mathbf{Leaves}(\mathscr{T}_2)$

To establish (a), let $\mathcal{C}_\kappa(v, I_{\mathbf{good}})$ be any leaf of $\mathscr{T}_{\mathbf{good}}$ and let $\mathcal{C}_\kappa(y, I_1)$ be an arbitrary element of $\mathcal{M}(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$. Then it follows from the definition of the set $\mathcal{M}(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$ that $\mathcal{C}_\kappa(y, I_1) \subseteq \mathcal{C}_\kappa(z, I_1)$ for some spel z that satisfies the condition $v \Leftarrow I_{\mathbf{good}} \geq I_{\mathbf{good}}(v)-2\varepsilon \Rightarrow z$ (which implies $I_{\mathbf{good}}(z) \geq I_{\mathbf{good}}(v) - 2\varepsilon$). Since $\mathcal{C}_\kappa(y, I_1) \subseteq \mathcal{C}_\kappa(z, I_1)$, we have that $z \Leftarrow I_1 \geq I_1(z) \Rightarrow y$. This implies $z \Leftarrow I_{\mathbf{good}} \geq I_{\mathbf{good}}(z)-2\varepsilon \Rightarrow y$ (in view of assertion (ii) of F), which implies $z \Leftarrow I_{\mathbf{good}} \geq I_{\mathbf{good}}(v)-4\varepsilon \Rightarrow y$ (as $I_{\mathbf{good}}(z) \geq I_{\mathbf{good}}(v) - 2\varepsilon$).

Combining $z \Leftarrow I_{\mathbf{good}} \geq I_{\mathbf{good}}(v)-4\varepsilon \Rightarrow y$ with $v \Leftarrow I_{\mathbf{good}} \geq I_{\mathbf{good}}(v)-2\varepsilon \Rightarrow z$, we deduce assertion (i) of (a). Now (ii) follows from (i) and D because $\mathcal{C}_\kappa(v, I_{\mathbf{good}}) \in \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})$, and (iii) follows from (ii) and F.

Now we establish (b). Suppose $\mathcal{C}_\kappa(y, I_1) = \phi(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$. Consider the node $\mathcal{C}_\kappa(v, I_1)$ of $\mathscr{T}_1$. Let $\mathbf{s}$ be a leaf of $\mathscr{T}_1$ such that $\mathbf{s} \succeq_{\mathscr{T}_1} \mathcal{C}_\kappa(v, I_1)$. Then we have that $\mathbf{s} \in \mathcal{M}(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$, by Lemma B1. Hence $\ell_1(\mathcal{C}_\kappa(y, I_1)) \geq \ell_1(\mathbf{s})$ (as $\mathbf{s}$ cannot occur later in the ℓ_1-increasing leaf enumeration that is used in pruning $(\mathscr{T}_1, \ell_1)$ than $\phi(\mathcal{C}_\kappa(v, I_{\mathbf{good}})) = \mathcal{C}_\kappa(y, I_1)$, by the definition of $\phi(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$). Therefore

$$I_1(y) = \ell_1\big(\mathcal{C}_\kappa(y, I_1)\big) \geq \ell_1(\mathbf{s}) \geq \ell_1\big(\mathcal{C}_\kappa(v, I_1)\big) = I_1(v) \tag{B7}$$

which establishes the second inequality of assertion (i) of (b). The third inequality of (i) follows from F. Now $I_{\mathbf{good}}(v) \geq I_{\mathbf{good}}(y)$ (by assertion (ii) of (a)). This implies $I_{\mathbf{good}}(v) \geq I_1(y) - \varepsilon$ (by F), which is equivalent to the first inequality of assertion (i) of (b). This establishes assertion (i) of (b). It follows from F and assertion (i) of (b) that $I_{\mathbf{good}}(y) \geq I_{\mathbf{good}}(v) - 2\varepsilon$. Assertion (ii) of (b) follows from this and assertion (ii) of (a).

To see that assertion (iii) of (b) holds, let $\mathcal{C}_\kappa(w, I_1)$ be any leaf of $\mathscr{T}_1$ that occurs later in the ℓ_1-increasing leaf enumeration that is used in pruning $(\mathscr{T}_1, \ell_1)$ than $\phi(\mathcal{C}_\kappa(v, I_{\mathbf{good}})) = \mathcal{C}_\kappa(y, I_1)$. Then it follows from the definitions of $\phi(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$ and of an ℓ_1-increasing leaf enumeration that:

- $\mathcal{C}_\kappa(w, I_1) \notin \mathcal{M}(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$
- $I_1(w) = \ell_1(\mathcal{C}_\kappa(w, I_1)) \geq \ell_1(\mathcal{C}_\kappa(y, I_1)) = I_1(y)$

As $I_1(w) \geq I_1(y)$, (B7) implies that $I_1(w) \geq I_1(v)$, and now it follows from F that $I_{\mathbf{good}}(w) \geq I_{\mathbf{good}}(v) - 2\varepsilon$. So $\mathcal{C}_\kappa(v, I_{\mathbf{good}}) \not\succeq_{\mathscr{T}_{\mathbf{good}}} \mathcal{C}_\kappa(w, I_{\mathbf{good}})$; otherwise the spel w would satisfy $w \Leftarrow I_{\mathbf{good}} \geq I_{\mathbf{good}}(w) \Rightarrow v$, which would imply that $w \Leftarrow I_{\mathbf{good}} \geq I_{\mathbf{good}}(v)-2\varepsilon \Rightarrow v$ (since $I_{\mathbf{good}}(w) \geq I_{\mathbf{good}}(v) - 2\varepsilon$), which would in turn imply that $\mathcal{C}_\kappa(w, I_1)$ is an element of $\mathcal{M}(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$ (by Lemma B1), which is false as we saw above.

Since $\mathcal{C}_\kappa(v, I_{\mathbf{good}}) \not\succeq_{\mathscr{T}_{\mathbf{good}}} \mathcal{C}_\kappa(w, I_{\mathbf{good}})$, it follows from C and A that w does *not* satisfy $w \Leftarrow I_{\mathbf{good}} \geq I_{\mathbf{good}}(v)-4\varepsilon \Rightarrow v$. This and assertion (ii) of F imply that w does *not* satisfy $w \Leftarrow I_1 \geq I_1(v)-2\varepsilon \Rightarrow v$, and so (since $I_1(y) \geq I_1(v)$, by (B7)) w does *not* satisfy

$w \Leftarrow_{I_1 \geq I_1(y)-2\varepsilon}\Rightarrow v$. But we know from assertion (iii) of (a) that $y \Leftarrow_{I_1 \geq I_1(y)-2\varepsilon}\Rightarrow v$, so w also does *not* satisfy $w \Leftarrow_{I_1 \geq I_1(y)-2\varepsilon}\Rightarrow y$. As $\mathcal{C}_\kappa(w, I_1)$ is an arbitrary leaf of $\mathscr{T}_1$ that occurs later in the ℓ_1-increasing leaf enumeration used in pruning $(\mathscr{T}_1, \ell_1)$ than the leaf $\phi(\mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}}))$, we see from E that $\phi(\mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}})) \in \mathbf{Leaves}(\mathscr{T}_2)$—i.e., assertion (iii) of (b) holds.

Since $\phi(\mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}})) \in \mathbf{Leaves}(\mathscr{T}_2)$ for every leaf $\mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}})$ of $\mathscr{T}_{\mathbf{good}}$, we can regard ϕ as a mapping $\phi : \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}}) \to \mathbf{Leaves}(\mathscr{T}_2)$, and we will do this from now on.

We next show that $\phi : \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}}) \to \mathbf{Leaves}(\mathscr{T}_2)$ is 1-to-1:

H. $\phi(\mathbf{v}) \neq \phi(\mathbf{v}')$ whenever $\mathbf{v}$ and $\mathbf{v}'$ are distinct leaves of $\mathscr{T}_{\mathbf{good}}$.

Indeed, let $\mathcal{C}_\kappa(v_{\mathrm{a}}, \boldsymbol{I}_{\mathbf{good}})$ and $\mathcal{C}_\kappa(v_{\mathrm{b}}, \boldsymbol{I}_{\mathbf{good}})$ be any two distinct leaves of $\mathscr{T}_{\mathbf{good}}$. To establish H, it is enough to show that $\mathcal{M}(\mathcal{C}_\kappa(v_{\mathrm{a}}, \boldsymbol{I}_{\mathbf{good}}))$ and $\mathcal{M}(\mathcal{C}_\kappa(v_{\mathrm{b}}, \boldsymbol{I}_{\mathbf{good}}))$ are disjoint. Suppose this is not the case. Then there is a leaf $\mathcal{C}_\kappa(x, I_1)$ of $\mathscr{T}_1$ such that $\mathcal{C}_\kappa(x, I_1) \in \mathcal{M}(\mathcal{C}_\kappa(v_{\mathrm{a}}, \boldsymbol{I}_{\mathbf{good}}))$ and $\mathcal{C}_\kappa(x, I_1) \in \mathcal{M}(\mathcal{C}_\kappa(v_{\mathrm{b}}, \boldsymbol{I}_{\mathbf{good}}))$. Now assertion (i) of part (a) of G implies that $v_{\mathrm{a}} \Leftarrow_{\boldsymbol{I}_{\mathbf{good}} \geq \boldsymbol{I}_{\mathbf{good}}(v_{\mathrm{a}})-4\varepsilon}\Rightarrow x$ and that $v_{\mathrm{b}} \Leftarrow_{\boldsymbol{I}_{\mathbf{good}} \geq \boldsymbol{I}_{\mathbf{good}}(v_{\mathrm{b}})-4\varepsilon}\Rightarrow x$.

Assuming without loss of generality that $\boldsymbol{I}_{\mathbf{good}}(v_{\mathrm{a}}) \leq \boldsymbol{I}_{\mathbf{good}}(v_{\mathrm{b}})$, these two properties imply that $v_{\mathrm{a}} \Leftarrow_{\boldsymbol{I}_{\mathbf{good}} \geq \boldsymbol{I}_{\mathbf{good}}(v_{\mathrm{a}})-4\varepsilon}\Rightarrow v_{\mathrm{b}}$, which is impossible in view of C and A. This contradiction establishes H and shows that ϕ is 1-to-1.

Next, we show that:

I. $\mathbf{Leaves}(\mathscr{T}_2) \setminus \phi[\mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})] = \emptyset$

To justify I, let $\mathcal{C}_\kappa(x, I_1)$ be any element of $\mathbf{Leaves}(\mathscr{T}_1) \setminus \phi[\mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})]$. Then what we need to show is that $\mathcal{C}_\kappa(x, I_1) \notin \mathbf{Leaves}(\mathscr{T}_2)$.

Let $\mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}})$ be a leaf of $\mathscr{T}_{\mathbf{good}}$ such that $\mathcal{C}_\kappa(x, \boldsymbol{I}_{\mathbf{good}}) \supseteq \mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}})$. Then $x \Leftarrow_{\boldsymbol{I}_{\mathbf{good}} \geq \boldsymbol{I}_{\mathbf{good}}(x)}\Rightarrow v$ and so it follows from F that $x \Leftarrow_{I_1 \geq I_1(x)-2\varepsilon}\Rightarrow v$. Let $\mathcal{C}_\kappa(y, I_1) = \phi(\mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}}))$. We now claim that:

- $\mathcal{C}_\kappa(y, I_1)$ occurs later in the ℓ_1-increasing leaf enumeration that is used in pruning $(\mathscr{T}_1, \ell_1)$ than $\mathcal{C}_\kappa(x, I_1)$.

Now we justify this claim. Just one of the following is true:

(a) $\boldsymbol{I}_{\mathbf{good}}(v) - 2\varepsilon > \boldsymbol{I}_{\mathbf{good}}(x)$
(b) $\boldsymbol{I}_{\mathbf{good}}(x) \geq \boldsymbol{I}_{\mathbf{good}}(v) - 2\varepsilon$

In case (a) it follows from F that $I_1(v) > I_1(x)$, and so $I_1(y) > I_1(x)$ (since $I_1(y) \geq I_1(v)$, by assertion (i) of part (b) of G); thus our claim is valid.

In case (b), we first observe that, since $x \Leftarrow_{\boldsymbol{I}_{\mathbf{good}} \geq \boldsymbol{I}_{\mathbf{good}}(x)}\Rightarrow v$, (b) implies that $x \Leftarrow_{\boldsymbol{I}_{\mathbf{good}} \geq \boldsymbol{I}_{\mathbf{good}}(v)-2\varepsilon}\Rightarrow v$, so that $\mathcal{C}_\kappa(x, I_1) \in \mathcal{M}(\mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}}))$ (by Lemma B1). Therefore $\mathcal{C}_\kappa(x, I_1) \in \mathcal{M}(\mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}})) \setminus \{\phi(\mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}}))\}$, because $\mathcal{C}_\kappa(x, I_1)$ is an element of $\mathbf{Leaves}(\mathscr{T}_1) \setminus \phi[\mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})]$. As $\mathcal{C}_\kappa(y, I_1) = \phi(\mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}}))$ and $\mathcal{C}_\kappa(x, I_1) \in \mathcal{M}(\mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}})) \setminus \{\phi(\mathcal{C}_\kappa(v, \boldsymbol{I}_{\mathbf{good}}))\}$, it follows from the definition of ϕ that our claim is again valid.

In either case, we have that $x \Leftarrow_{I_1 \geq I_1(x)-2\varepsilon}\Rightarrow v$ (as we saw above), and the claim implies $I_1(y) \geq I_1(x)$. So, since we see from assertion (iii) of part (a) of G that $v \Leftarrow_{I_1 \geq I_1(y)-2\varepsilon}\Rightarrow y$, we also have that $x \Leftarrow_{I_1 \geq I_1(x)-2\varepsilon}\Rightarrow y$. From this, E, and the above claim, we deduce that $\mathcal{C}_\kappa(x, I_1) \notin \mathbf{Leaves}(\mathscr{T}_2)$. This justifies I.

It follows from H and I that $\phi : \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}}) \to \mathbf{Leaves}(\mathscr{T}_2)$ is a bijection. This completes step 2 of the proof of the Fundamental Lemma.

B.4 Step 3 of the Proof of the Fundamental Lemma

We now extend ϕ to a mapping $\varphi : \mathbf{Crit}(\mathscr{T}_{\mathbf{good}}) \to \mathbf{Crit}(\mathscr{T}_2)$ by defining $\varphi(\mathbf{u}) = \bigwedge_{\mathscr{T}_2} \phi[\mathbf{Leaves}(\mathscr{T}_{\mathbf{good}}[\mathbf{u}])]$. We will establish two properties of the mapping φ which together imply that φ is an essential isomorphism of $\mathfrak{F}_{\mathbf{good}}$ to $\mathfrak{F}_3$. The first property is that, for all $\mathbf{u}, \mathbf{u}' \in \mathbf{Crit}(\mathscr{T}_{\mathbf{good}})$, $\varphi(\mathbf{u}) \preceq_{\mathscr{T}_2} \varphi(\mathbf{u}')$ if and only if $\mathbf{u} \preceq_{\mathscr{T}_{\mathbf{good}}} \mathbf{u}'$ (so that φ is an order-preserving injection). The second property is that $\varphi[\mathbf{Crit}(\mathscr{T}_{\mathbf{good}})] = \mathbf{Crit}(\mathscr{T}_3)$. To establish these two properties, we first show that:

J. $|\ell_2(\bigwedge_{\mathscr{T}_2} \phi[\mathbf{L}]) - \ell_{\mathbf{good}}(\bigwedge_{\mathscr{T}_{\mathbf{good}}} \mathbf{L})| \le \varepsilon$ whenever $\emptyset \ne \mathbf{L} \subseteq \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})$.

Indeed, suppose $\emptyset \ne \mathbf{L} \subseteq \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})$. If $|\mathbf{L}| = 1$, then J is an immediate consequence of assertion (i) of part (b) of G, so we will assume $|\mathbf{L}| \ge 2$.

For brevity, we will write $\tau_{\mathbf{L}}$ for $\ell_{\mathbf{good}}(\bigwedge_{\mathscr{T}_{\mathbf{good}}} \mathbf{L})$ and $\tau_{\phi[\mathbf{L}]}$ for $\ell_2(\bigwedge_{\mathscr{T}_2} \phi[\mathbf{L}])$, so that J can be written as $|\tau_{\phi[\mathbf{L}]} - \tau_{\mathbf{L}}| \le \varepsilon$.

We first show that $\tau_{\phi[\mathbf{L}]} \ge \tau_{\mathbf{L}} - \varepsilon$. For this purpose, let $\mathcal{C}_\kappa(x, I_1)$ and $\mathcal{C}_\kappa(y, I_1)$ be any two distinct elements of $\phi[\mathbf{L}]$. Then $\mathcal{C}_\kappa(x, I_1) = \phi(\mathcal{C}_\kappa(u, I_{\mathbf{good}}))$ and $\mathcal{C}_\kappa(y, I_1) = \phi(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$, where $\mathcal{C}_\kappa(u, I_{\mathbf{good}})$ and $\mathcal{C}_\kappa(v, I_{\mathbf{good}})$ are two distinct elements of $\mathbf{L}$. From A and the definition of $\tau_{\mathbf{L}}$ we see that $u \Leftarrow I_{\mathbf{good}} \ge \tau_{\mathbf{L}} \Rightarrow v$. This and F imply that $u \Leftarrow I_1 \ge \tau_{\mathbf{L}} - \varepsilon \Rightarrow v$. We see from the definition of ϕ and assertion (iii) of part (a) of G that $x \Leftarrow I_1 \ge I_1(x) - 2\varepsilon \Rightarrow u$ and $y \Leftarrow I_1 \ge I_1(y) - 2\varepsilon \Rightarrow v$. Combining the last three observations, we deduce that:

$$x \Leftarrow I_1 \ge \min(\tau_{\mathbf{L}} - \varepsilon, I_1(x) - 2\varepsilon, I_1(y) - 2\varepsilon) \Rightarrow y \tag{B8}$$

However, it follows from C and the definition of $\tau_{\mathbf{L}}$ that

$$\begin{aligned}
\tau_{\mathbf{L}} &\le \ell_{\mathbf{good}}\Big(\bigwedge\nolimits_{\mathscr{T}_{\mathbf{good}}} \big\{\mathcal{C}_\kappa(u, I_{\mathbf{good}}), \mathcal{C}_\kappa(v, I_{\mathbf{good}})\big\}\Big) \\
&< \min\Big(\ell_{\mathbf{good}}\big(\mathcal{C}_\kappa(u, I_{\mathbf{good}})\big) - 4\varepsilon, \ell_{\mathbf{good}}\big(\mathcal{C}_\kappa(v, I_{\mathbf{good}})\big) - 4\varepsilon\Big) \\
&= \min\Big(I_{\mathbf{good}}(u) - 4\varepsilon, I_{\mathbf{good}}(v) - 4\varepsilon\Big)
\end{aligned}$$

which implies that $\tau_{\mathbf{L}} - \varepsilon < \min(I_{\mathbf{good}}(u) - 5\varepsilon, I_{\mathbf{good}}(v) - 5\varepsilon)$, which implies that $\tau_{\mathbf{L}} - \varepsilon < \min(I_1(u) - 4\varepsilon, I_1(v) - 4\varepsilon)$ (in view of F), which in turn implies that $\tau_{\mathbf{L}} - \varepsilon < \min(I_1(x) - 4\varepsilon, I_1(y) - 4\varepsilon)$ (by assertion (i) of part (b) of G). So (B8) can be simplified to $x \Leftarrow I_1 \ge \tau_{\mathbf{L}} - \varepsilon \Rightarrow y$. It now follows from A that $\tau_{\phi[\mathbf{L}]} \ge \tau_{\mathbf{L}} - \varepsilon$ (since $\mathcal{C}_\kappa(x, I_1)$ and $\mathcal{C}_\kappa(y, I_1)$ are arbitrary distinct elements of $\phi[\mathbf{L}]$), as required.

To complete the proof of J, we show that $\tau_{\mathbf{L}} \ge \tau_{\phi[\mathbf{L}]} - \varepsilon$. This time we let $\mathcal{C}_\kappa(u, I_{\mathbf{good}})$ and $\mathcal{C}_\kappa(v, I_{\mathbf{good}})$ be any two distinct elements of $\mathbf{L}$, and then define $\mathcal{C}_\kappa(x, I_1) = \phi(\mathcal{C}_\kappa(u, I_{\mathbf{good}}))$ and $\mathcal{C}_\kappa(y, I_1) = \phi(\mathcal{C}_\kappa(v, I_{\mathbf{good}}))$, so that $\mathcal{C}_\kappa(x, I_1)$, $\mathcal{C}_\kappa(y, I_1) \in \phi[\mathbf{L}]$. From A and the definition of $\tau_{\phi[\mathbf{L}]}$ we see that $x \Leftarrow I_1 \ge \tau_{\phi[\mathbf{L}]} \Rightarrow y$. This and F imply that $x \Leftarrow I_{\mathbf{good}} \ge \tau_{\phi[\mathbf{L}]} - \varepsilon \Rightarrow y$. We see from assertion (ii) of part (b) of G that

$u \Leftarrow\!\!I_{\mathbf{good}} \geq I_{\mathbf{good}}(u)-2\varepsilon\!\!\Rightarrow x$; we similarly have that $v \Leftarrow\!\!I_{\mathbf{good}} \geq I_{\mathbf{good}}(v)-2\varepsilon\!\!\Rightarrow y$. Combining the last three observations, we see that:

$$u \Leftarrow\!\!I_{\mathbf{good}} \geq \min(\tau_{\phi[\mathbf{L}]}-\varepsilon, I_{\mathbf{good}}(u)-2\varepsilon, I_{\mathbf{good}}(v)-2\varepsilon)\!\!\Rightarrow v \tag{B9}$$

However, it follows from the definition of $\tau_{\phi[\mathbf{L}]}$ and E$'$ that:

$$\begin{aligned}
\tau_{\phi[\mathbf{L}]} &\leq \ell_2\Big(\bigwedge\nolimits_{\mathscr{T}_2}\{\phi\big(\mathcal{C}_\kappa(u, I_{\mathbf{good}})\big), \phi\big(\mathcal{C}_\kappa(v, I_{\mathbf{good}})\big)\}\Big)\\
&< \min\Big(\ell_2\big(\phi\big(\mathcal{C}_\kappa(u, I_{\mathbf{good}})\big)\big), \ell_2\big(\phi\big(\mathcal{C}_\kappa(v, I_{\mathbf{good}})\big)\big)\Big) - 2\varepsilon\\
&= \min\Big(\ell_2\big(\mathcal{C}_\kappa(x, I_1)\big), \ell_2\big(\mathcal{C}_\kappa(y, I_1)\big)\Big) - 2\varepsilon = \min\big(I_1(x), I_1(y)\big) - 2\varepsilon
\end{aligned}$$

Hence $\tau_{\phi[\mathbf{L}]} - \varepsilon < \min(I_1(x) - 3\varepsilon, I_1(y) - 3\varepsilon)$, which (by assertion (i) of part (b) of G) implies $\tau_{\phi[\mathbf{L}]} - \varepsilon < \min(I_{\mathbf{good}}(u) - 2\varepsilon, I_{\mathbf{good}}(v) - 2\varepsilon)$. We now see from (B9) that $u \Leftarrow\!\!I_{\mathbf{good}} \geq \tau_{\phi[\mathbf{L}]}-\varepsilon\!\!\Rightarrow v$. It follows from this and A that $\tau_{\mathbf{L}} \geq \tau_{\phi[\mathbf{L}]} - \varepsilon$ (since $\mathcal{C}_\kappa(u, I_{\mathbf{good}})$ and $\mathcal{C}_\kappa(v, I_{\mathbf{good}})$ are arbitrary distinct elements of $\mathbf{L}$), as required. Thus we have established J.

From B and J, we deduce:

K. Whenever $\emptyset \neq \mathbf{L} \subseteq \mathbf{L}' \subseteq \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})$, $\bigwedge_{\mathscr{T}_{\mathbf{good}}} \mathbf{L}' = \bigwedge_{\mathscr{T}_{\mathbf{good}}} \mathbf{L}$ if and only if $\ell_2(\bigwedge_{\mathscr{T}_2} \phi[\mathbf{L}]) - \ell_2(\bigwedge_{\mathscr{T}_2} \phi[\mathbf{L}']) \leq 2\varepsilon$.

As we show in Appendix C, it is not difficult to deduce from K that:

L. For all $\mathbf{u} \in \mathbf{Crit}(\mathscr{T}_{\mathbf{good}})$, $\mathbf{Leaves}(\mathscr{T}_2[\varphi(\mathbf{u})]) = \varphi[\mathbf{Leaves}(\mathscr{T}_{\mathbf{good}}[\mathbf{u}])]$.
M. For all $\mathbf{x} \in \varphi[\mathbf{Crit}(\mathscr{T}_{\mathbf{good}})]$, there is no $\mathbf{y} \in \mathbf{x}{\downarrow}_{\mathscr{T}_2} \cap \mathbf{Crit}(\mathscr{T}_2)$ that satisfies the condition $\ell_2(\mathbf{x}) - \ell_2(\mathbf{y}) \leq 2\varepsilon$.
N. For all $\mathbf{x} \in \mathbf{Crit}(\mathscr{T}_2)$, some $\mathbf{z} \in \mathbf{x}{\Downarrow}_{\mathscr{T}_2} \cap \varphi[\mathbf{Crit}(\mathscr{T}_{\mathbf{good}})]$ satisfies the condition $\ell_2(\mathbf{x}) - \ell_2(\mathbf{z}) \leq 2\varepsilon$.

We mention here that N is proved by showing that for every $\mathbf{x} \in \mathbf{Crit}(\mathscr{T}_2)$ the node $\mathbf{z} = \varphi(\bigwedge_{\mathscr{T}_{\mathbf{good}}} \varphi^{-1}[\mathbf{Leaves}(\mathscr{T}_2[\mathbf{x}])])$ has the stated property.

Using L, it is quite easy to show that:

O. For all $\mathbf{u}, \mathbf{u}' \in \mathbf{Crit}(\mathscr{T}_{\mathbf{good}})$, $\varphi(\mathbf{u}) \preceq_{\mathscr{T}_2} \varphi(\mathbf{u}')$ if and only if $\mathbf{u} \preceq_{\mathscr{T}_{\mathbf{good}}} \mathbf{u}'$.

Details of the proof of O are given in Appendix C. It follows from O that φ is an order-preserving injection.

As $\mathfrak{F}_3 = (\mathscr{T}_3, \ell_3)$ is the result of eliminating internal edges of length $\leq 2\varepsilon$ from $\mathfrak{F}_2^{\mathbf{crit}}$, it follows from M and property E8 of simplification step 3 that φ must satisfy $\varphi[\mathbf{Crit}(\mathscr{T}_{\mathbf{good}})] \subseteq \mathbf{Crit}(\mathscr{T}_2) \cap \mathbf{Nodes}(\mathscr{T}_3) = \mathbf{Crit}(\mathscr{T}_3)$. Moreover, N implies that, for all $\mathbf{x} \in \mathbf{Crit}(\mathscr{T}_2) \setminus \varphi[\mathbf{Crit}(\mathscr{T}_{\mathbf{good}})]$, some $\mathbf{z} \in \mathbf{x}{\downarrow}_{\mathscr{T}_2} \cap \varphi[\mathbf{Crit}(\mathscr{T}_{\mathbf{good}})]$ satisfies $\ell_2(\mathbf{x}) - \ell_2(\mathbf{z}) \leq 2\varepsilon$. We therefore have that:

- For all $\mathbf{x} \in \mathbf{Crit}(\mathscr{T}_2) \setminus \varphi[\mathbf{Crit}(\mathscr{T}_{\mathbf{good}})]$, some $\mathbf{z} \in \mathbf{x}{\downarrow}_{\mathscr{T}_2} \cap \mathbf{Crit}(\mathscr{T}_3)$ satisfies the condition $\ell_2(\mathbf{x}) - \ell_2(\mathbf{z}) \leq 2\varepsilon$.

From this and property E9 of simplification step 3 we deduce that φ satisfies $(\mathbf{Crit}(\mathscr{T}_2) \setminus \varphi[\mathbf{Crit}(\mathscr{T}_{\mathbf{good}})]) \cap \mathbf{Nodes}(\mathscr{T}_3) = \emptyset$. Equivalently, φ satisfies the condition $\mathbf{Crit}(\mathscr{T}_3) \setminus \varphi[\mathbf{Crit}(\mathscr{T}_{\mathbf{good}})] = \emptyset$. Thus $\varphi[\mathbf{Crit}(\mathscr{T}_{\mathbf{good}})] = \mathbf{Crit}(\mathscr{T}_3)$. So the order-preserving injection φ can be regarded as a bijection $\varphi : \mathbf{Crit}(\mathscr{T}_{\mathbf{good}}) \to \mathbf{Crit}(\mathscr{T}_3)$. When so regarded, φ is an essential isomorphism of $\mathfrak{F}_{\mathbf{good}}$ to $\mathfrak{F}_3$. Finally, φ is level-preserving to within ε because, for any node $\mathbf{u} \in \mathbf{Crit}(\mathscr{T}_{\mathbf{good}})$, we deduce from J (on putting $\mathbf{L} = \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}}[\mathbf{u}])$, so that $\bigwedge_{\mathscr{T}_{\mathbf{good}}} \mathbf{L} = \mathbf{u}$) that $|\ell_3(\varphi(\mathbf{u})) - \ell_{\mathbf{good}}(\mathbf{u})| \le \varepsilon$.

This completes the proof of the Fundamental Lemma.

Appendix C: Justification of Assertions L, M, N, and O in Step 3 of the Proof of the Fundamental Lemma

For any rooted tree $\mathscr{T}$ and any $\mathbf{u} \in \mathbf{Crit}(\mathscr{T})$, we write $\mathcal{L}_{\mathscr{T}}\mathbf{u}$ to denote the set $\mathbf{Leaves}(\mathscr{T}[\mathbf{u}]) = \{\mathbf{v} \in \mathbf{Leaves}(\mathscr{T}) \mid \mathbf{u} \preceq_{\mathscr{T}} \mathbf{v}\}$. It is readily confirmed that the following are true in any rooted tree $\mathscr{T}$:

$$\text{If } \emptyset \ne \mathbf{L} \subseteq \mathbf{L}' \subseteq \mathbf{Leaves}(\mathscr{T}), \text{ then: } \quad \textstyle\bigwedge_{\mathscr{T}} \mathbf{L}' \preceq_{\mathscr{T}} \bigwedge_{\mathscr{T}} \mathbf{L} \tag{C1}$$

$$\text{If } \emptyset \ne \mathbf{L} \subseteq \mathbf{Leaves}(\mathscr{T}), \text{ then: } \quad \mathcal{L}_{\mathscr{T}} \textstyle\bigwedge_{\mathscr{T}} \mathbf{L} \supseteq \mathbf{L} \tag{C2}$$

$$\text{If } \mathbf{u} \in \mathbf{Crit}(\mathscr{T}), \text{ then: } \quad \textstyle\bigwedge_{\mathscr{T}} \mathcal{L}_{\mathscr{T}}\mathbf{u} = \mathbf{u} \tag{C3}$$

$$\text{If } \mathbf{u} \in \mathbf{Crit}(\mathscr{T}) \text{ and } \mathbf{L} \supsetneq \mathcal{L}_{\mathscr{T}}\mathbf{u}, \text{ then: } \quad \textstyle\bigwedge_{\mathscr{T}} \mathbf{L} \prec_{\mathscr{T}} \mathbf{u} = \bigwedge_{\mathscr{T}} \mathcal{L}_{\mathscr{T}}\mathbf{u} \tag{C4}$$

$$\text{If } \mathbf{u}, \mathbf{v} \in \mathbf{Crit}(\mathscr{T}), \text{ then: } \quad \mathcal{L}_{\mathscr{T}}\mathbf{v} = \mathcal{L}_{\mathscr{T}}\mathbf{u} \text{ if and only if } \mathbf{v} = \mathbf{u} \tag{C5}$$

$$\text{If } \mathbf{u}, \mathbf{v} \in \mathbf{Crit}(\mathscr{T}), \text{ then: } \quad \mathcal{L}_{\mathscr{T}}\mathbf{v} \supsetneq \mathcal{L}_{\mathscr{T}}\mathbf{u} \text{ if and only if } \mathbf{v} \prec_{\mathscr{T}} \mathbf{u} \tag{C6}$$

For all $\mathbf{L} \subseteq \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})$ and all $\mathbf{L}' \subseteq \mathbf{Leaves}(\mathscr{T}_2)$, we write $\phi\mathbf{L}$ to mean $\phi[\mathbf{L}]$ and we write $\phi^{-1}\mathbf{L}$ to mean $\phi^{-1}[\mathbf{L}]$.

If $\mathbf{x} \preceq_{\mathscr{T}_2} \mathbf{y}$ or $\mathbf{y} \preceq_{\mathscr{T}_2} \mathbf{x}$, and λ is any positive value, then we write $\mathbf{x} \approx_\lambda \mathbf{y}$ to mean that $|\ell_2(\mathbf{y}) - \ell_2(\mathbf{x})| \le \lambda$, and write $\mathbf{x} \prec_\lambda \mathbf{y}$ to mean that $\ell_2(\mathbf{y}) - \ell_2(\mathbf{x}) > \lambda$; in the latter case we must have that $\mathbf{x} \prec_{\mathscr{T}_2} \mathbf{y}$. For brevity, we will write $\bigwedge_{\mathbf{good}}$ and $\bigwedge_2$ to mean $\bigwedge_{\mathscr{T}_{\mathbf{good}}}$ and $\bigwedge_{\mathscr{T}_2}$, and write $\mathcal{L}_{\mathbf{good}}$ and $\mathcal{L}_2$ to mean $\mathcal{L}_{\mathscr{T}_{\mathbf{good}}}$ and $\mathcal{L}_{\mathscr{T}_2}$. Note that the definition of the mapping φ can be rewritten in terms of ϕ and $\mathcal{L}_{\mathbf{good}}$ as follows:

$$\varphi(\mathbf{u}) \stackrel{\text{def}}{=} \textstyle\bigwedge_2 \phi\mathcal{L}_{\mathbf{good}}\mathbf{u} \tag{C7}$$

If $\emptyset \ne \mathbf{L} \subseteq \mathbf{L}' \subseteq \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})$, then $\emptyset \ne \phi\mathbf{L} \subseteq \phi\mathbf{L}' \subseteq \mathbf{Leaves}(\mathscr{T}_2)$ and so $\bigwedge_2 \phi\mathbf{L}' \preceq_{\mathscr{T}_2} \bigwedge_2 \phi\mathbf{L}$ (by (C1)). Hence assertion K can be restated as follows (for all nonempty sets $\mathbf{L} \subseteq \mathbf{L}' \subseteq \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})$):

$$\textstyle\bigwedge_2 \phi\mathbf{L}' \approx_{2\varepsilon} \bigwedge_2 \phi\mathbf{L} \quad \text{if and only if} \quad \bigwedge_{\mathbf{good}} \mathbf{L}' = \bigwedge_{\mathbf{good}} \mathbf{L} \tag{C8}$$

When $\emptyset \ne \mathbf{L} \subseteq \mathbf{L}' \subseteq \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})$, the negations of $\bigwedge_2 \phi\mathbf{L} \approx_{2\varepsilon} \bigwedge_2 \phi\mathbf{L}'$ and $\bigwedge_{\mathbf{good}} \mathbf{L}' = \bigwedge_{\mathbf{good}} \mathbf{L}$ are $\bigwedge_2 \phi\mathbf{L}' \prec_{2\varepsilon} \bigwedge_2 \phi\mathbf{L}$ and $\bigwedge_{\mathbf{good}} \mathbf{L}' \prec_{\mathscr{T}_{\mathbf{good}}} \bigwedge_{\mathbf{good}} \mathbf{L}$ respectively (since $\bigwedge_{\mathbf{good}} \mathbf{L}' \preceq_{\mathscr{T}_{\mathbf{good}}} \bigwedge_{\mathbf{good}} \mathbf{L}$ and $\bigwedge_2 \phi\mathbf{L}' \preceq_{\mathscr{T}_2} \bigwedge_2 \phi\mathbf{L}$), so (C8) can also be stated as follows (for all nonempty sets $\mathbf{L} \subseteq \mathbf{L}' \subseteq \mathbf{Leaves}(\mathscr{T}_{\mathbf{good}})$):

$$\textstyle\bigwedge_2 \phi\mathbf{L}' \prec_{2\varepsilon} \bigwedge_2 \phi\mathbf{L} \quad \text{if and only if} \quad \bigwedge_{\mathbf{good}} \mathbf{L}' \prec_{\mathscr{T}_{\mathbf{good}}} \bigwedge_{\mathbf{good}} \mathbf{L} \tag{C9}$$

C.1 Proof of Assertion L

In view of (C7), L can be restated as follows:

- For all $\mathbf{u} \in \mathbf{Crit}(\mathscr{T}_{\mathbf{good}})$, we have that $\mathcal{L}_2 \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} = \phi \mathcal{L}_{\mathbf{good}} \mathbf{u}$. Equivalently, $\phi^{-1} \mathcal{L}_2 \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} = \mathcal{L}_{\mathbf{good}} \mathbf{u}$.

To prove this, let $\mathbf{u} \in \mathbf{Crit}(\mathscr{T}_{\mathbf{good}})$. Then we successively deduce:

$$\mathcal{L}_2 \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \supseteq \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \quad \big[\text{by (C2)}\big]$$

$$\phi^{-1} \mathcal{L}_2 \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \supseteq \phi^{-1} \phi \mathcal{L}_{\mathbf{good}} \mathbf{u}$$

$$\phi^{-1} \mathcal{L}_2 \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \supseteq \mathcal{L}_{\mathbf{good}} \mathbf{u} \tag{C10}$$

The result will follow from (C10) if we can show that the following is *not* true:

$$\phi^{-1} \mathcal{L}_2 \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \supsetneq \mathcal{L}_{\mathbf{good}} \mathbf{u} \tag{C11}$$

To do this, we derive a contradiction from (C11) as follows:

$$\bigwedge_{\mathbf{good}} \phi^{-1} \mathcal{L}_2 \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \prec_{\mathscr{T}_{\mathbf{good}}} \bigwedge_{\mathbf{good}} \mathcal{L}_{\mathbf{good}} \mathbf{u} \quad \big[\text{by (C11) and (C4)}\big]$$

$$\bigwedge_2 \phi \phi^{-1} \mathcal{L}_2 \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \prec_{2\varepsilon} \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \quad \big[\text{by (C9) and (C10)}\big]$$

$$\bigwedge_2 \mathcal{L}_2 \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \prec_{2\varepsilon} \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u}$$

$$\bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \prec_{2\varepsilon} \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \quad \big[\text{by (C3)}\big]$$

C.2 Proof of Assertion M

In view of (C7), M is equivalent to:

- If $\mathbf{x} = \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u}$ for some $\mathbf{u} \in \mathbf{Crit}(\mathscr{T}_{\mathbf{good}})$, and if $\mathbf{y} \in \mathbf{Crit}(\mathscr{T}_2)$ satisfies $\mathbf{y} \prec_{\mathscr{T}_2} \mathbf{x}$, then $\mathbf{y} \prec_{2\varepsilon} \mathbf{x}$.

To prove this, suppose $\mathbf{x} = \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u}$ for some $\mathbf{u} \in \mathbf{Crit}(\mathscr{T}_{\mathbf{good}})$, and $\mathbf{y} \in \mathbf{Crit}(\mathscr{T}_2)$ satisfies $\mathbf{y} \prec_{\mathscr{T}_2} \mathbf{x}$. Then we can successively deduce:

$$\mathbf{y} \prec_{\mathscr{T}_2} \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \quad [\text{because } \mathbf{y} \prec_{\mathscr{T}_2} \mathbf{x}]$$

$$\mathcal{L}_2 \mathbf{y} \supsetneq \mathcal{L}_2 \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \quad \big[\text{by (C6)}\big]$$

$$\mathcal{L}_2 \mathbf{y} \supsetneq \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \quad \big[\text{by (C2)}\big]$$

$$\phi^{-1} \mathcal{L}_2 \mathbf{y} \supsetneq \mathcal{L}_{\mathbf{good}} \mathbf{u} \tag{C12}$$

$$\bigwedge_{\mathbf{good}} \phi^{-1} \mathcal{L}_2 \mathbf{y} \prec_{\mathscr{T}_{\mathbf{good}}} \bigwedge_{\mathbf{good}} \mathcal{L}_{\mathbf{good}} \mathbf{u} \quad \big[\text{by (C4)}\big]$$

$$\bigwedge_2 \phi \phi^{-1} \mathcal{L}_2 \mathbf{y} \prec_{2\varepsilon} \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \quad \big[\text{by (C9) and (C12)}\big]$$

$$\bigwedge_2 \mathcal{L}_2 \mathbf{y} \prec_{2\varepsilon} \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u}$$

$$\mathbf{y} \prec_{2\varepsilon} \bigwedge_2 \phi \mathcal{L}_{\mathbf{good}} \mathbf{u} \quad \big[\text{by (C3)}\big]$$

This proves that $\mathbf{y} \prec_{2\varepsilon} \mathbf{x}$.

C.3 Proof of Assertion N

In view of (C7), $\bigwedge_2 \phi\mathcal{L}_{\mathbf{good}} \bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x} \in \varphi[\mathbf{Crit}(\mathscr{T}_{\mathbf{good}})]$ for every node $\mathbf{x}$ of $\mathscr{T}_2$. So N can be proved by establishing that:

- For all $\mathbf{x} \in \mathbf{Crit}(\mathscr{T}_2)$, the node $\mathbf{z} = \bigwedge_2 \phi\mathcal{L}_{\mathbf{good}} \bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x}$ satisfies $\mathbf{z} \preceq_{\mathscr{T}_2} \mathbf{x}$ and $\mathbf{x} \approx_{2\varepsilon} \mathbf{z}$.

To prove this, let $\mathbf{x} \in \mathbf{Crit}(\mathscr{T}_2)$ and let $\mathbf{z} = \bigwedge_2 \phi\mathcal{L}_{\mathbf{good}} \bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x}$. Then we successively deduce:

$$\begin{aligned} \mathcal{L}_{\mathbf{good}} \textstyle\bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x} &\supseteq \phi^{-1}\mathcal{L}_2\mathbf{x} \quad \big[\text{by (C2)}\big] \\ V\phi\mathcal{L}_{\mathbf{good}} \textstyle\bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x} &\supseteq \phi\phi^{-1}\mathcal{L}_2\mathbf{x} \\ \phi\mathcal{L}_{\mathbf{good}} \textstyle\bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x} &\supseteq \mathcal{L}_2\mathbf{x} \\ \textstyle\bigwedge_2 \phi\mathcal{L}_{\mathbf{good}} \bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x} &\preceq_{\mathscr{T}_2} \textstyle\bigwedge_2 \mathcal{L}_2\mathbf{x} \quad \big[\text{by (C1)}\big] \\ \textstyle\bigwedge_2 \phi\mathcal{L}_{\mathbf{good}} \bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x} &\preceq_{\mathscr{T}_2} \mathbf{x} \quad \big[\text{by (C3)}\big] \end{aligned} \tag{C13}$$

This proves that $\mathbf{z} \preceq_{\mathscr{T}_2} \mathbf{x}$. We can also successively deduce:

$$\begin{aligned} &\textstyle\bigwedge_{\mathbf{good}} \mathcal{L}_{\mathbf{good}} \bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x} = \bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x} \quad \big[\text{by (C3)}\big] \\ &\textstyle\bigwedge_2 \phi\mathcal{L}_{\mathbf{good}} \bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x} \approx_{2\varepsilon} \bigwedge_2 \phi\phi^{-1}\mathcal{L}_2\mathbf{x} \quad \big[\text{by (C8) and (C13)}\big] \\ &\textstyle\bigwedge_2 \phi\mathcal{L}_{\mathbf{good}} \bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x} \approx_{2\varepsilon} \bigwedge_2 \mathcal{L}_2\mathbf{x} \\ &\textstyle\bigwedge_2 \phi\mathcal{L}_{\mathbf{good}} \bigwedge_{\mathbf{good}} \phi^{-1}\mathcal{L}_2\mathbf{x} \approx_{2\varepsilon} \mathbf{x} \quad \big[\text{by (C3)}\big] \end{aligned}$$

This proves that $\mathbf{z} \approx_{2\varepsilon} \mathbf{x}$.

C.4 Proof of Assertion O

Let $\mathbf{u}, \mathbf{u}' \in \mathbf{Crit}(\mathscr{T}_{\mathbf{good}})$. Then:

$$\begin{aligned} \mathbf{u} \preceq_{\mathscr{T}_{\mathbf{good}}} \mathbf{u}' \quad &\text{just if} \quad \mathcal{L}_{\mathbf{good}}\mathbf{u} \supseteq \mathcal{L}_{\mathbf{good}}\mathbf{u}' && \big[\text{by (C5) and (C6)}\big] \\ &\text{just if} \quad \phi\mathcal{L}_{\mathbf{good}}\mathbf{u} \supseteq \phi\mathcal{L}_{\mathbf{good}}\mathbf{u}' \\ &\text{just if} \quad \mathcal{L}_2\varphi(\mathbf{u}) \supseteq \mathcal{L}_2\varphi\big(\mathbf{u}'\big) && [\text{by assertion L}] \\ &\text{just if} \quad \varphi(\mathbf{u}) \preceq_{\mathscr{T}_2} \varphi\big(\mathbf{u}'\big) && \big[\text{by (C5) and (C6)}\big] \end{aligned}$$

References

1. Carr, H., Snoeyink, J., Axen, U.: Computing contour trees in all dimensions. Comput. Geom. **24**, 75–94 (2003)
2. Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Stability of persistence diagrams. Discrete Comput. Geom. **37**, 103–120 (2007)

3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, Cambridge (2001)
4. Edelsbrunner, H., Harer, J.: Computational Topology: An Introduction. Am. Math. Soc., Providence (2010)
5. Herman, G.T.: Geometry of Digital Spaces. Birkhäuser Boston, Boston (1998)
6. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Rosin, P.L., Marshall, D. (eds.) Proceedings of the British Machine Vision Conference, BMVC 2002, pp. 384–393. British Machine Vision Association, Malvern (2002).
7. Najman, L., Couprie, M.: Building the component tree in quasi-linear time. IEEE Trans. Image Process. **15**, 3531–3539 (2006)
8. Russell, W.C.: Update on adenovirus and its vectors. J. Gen. Virol. **81**, 2573–2604 (2000)
9. San Martín, C., Glasgow, J.N., Borovjagin, A., Beatty, M.S., Kashentseva, E.A., Curiel, D.T., Marabini, R., Dmitriev, I.P.: Localization of the N-terminus of minor coat protein IIIa in the adenovirus capsid. J. Mol. Biol. **383**, 923–934 (2008)
10. Sarioz, D., Kong, T.Y., Herman, G.T.: History trees as descriptors of macromolecular structures. In: Bebis, G., et al. (eds.) Advances in Visual Computing: Second International Symposium, ISVC 2006, Proceedings, Part I, pp. 263–272. Springer, Berlin (2006)
11. Tarjan, R.E.: Efficiency of a good but not linear set union algorithm. J. Assoc. Comput. Mach. **22**, 215–225 (1975)
12. Weber, G.H., Dillard, S.E., Carr, H., Pascucci, V., Hamann, B.: Topology controlled volume rendering. IEEE Trans. Vis. Comput. Graph. **13**, 330–341 (2007)

http://www.springer.com/978-94-007-4173-7

Digital Geometry Algorithms
Theoretical Foundations and Applications to Computational Imaging
Brimkov, V.E.; Barneva, R.P. (Eds.)
2012, X, 430 p., Hardcover
ISBN: 978-94-007-4173-7