Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 359 (2006) 378–399

Theoretical Computer Science

www.elsevier.com/locate/tcs

DLS-trees: A model of evolutionary scenarios☆

Paweł Górecki*, Jerzy Tiuryn

Warsaw University, Institute of Informatics, Banacha 2, 02-097 Warsaw, Poland

Received 10 June 2005; accepted 5 May 2006

Communicated by M. Crochemore

Abstract

We present a model of evolution of gene trees in the context of species evolution. Its concept is similar to reconciliation models. We assume that the gene evolution is modelled by duplications and losses. Evolution of species is modelled by speciation events. We define an evolutionary scenario (called a DLS-tree) which can represent an evolution of genes in species. We are interested in all scenarios for a given species tree and a given gene tree—not only parsimonious ones. We propose a rewrite system for transforming the scenarios. We prove that the system is confluent, sound and strongly normalizing. We show that a scenario in normal form (i.e., non-reducible) is unique and minimal in the sense of the cost computed as the total number of gene duplications and losses (mutation cost). We present a classification of the scenarios and analyze their hierarchy. Finally, we prove that the reconciled tree can be easily transformed into DLS-tree in normal form. This solves some open problems for reconciled trees.

Keywords: Rewrite system; Molecular evolution; Phylogenetic tree; Duplication-loss model; Reconciled tree

1. Introduction

Reconstruction of species relationships from a set of gene family trees is a difficult task. Hardness of this problem is caused by dissimilarities which usually occur between gene trees. Those inconsistencies are due to gene losses, gene duplications, gene convergence, horizontal gene transfers or errors in sequencing. They lead to two important problems: reconstruction of the species tree from a family of possibly different gene trees and reconciling a given gene tree with a given species tree.

These problems have been studied by Goodman [6] and then in the nineties [4,9,12–15]. The concepts of *mapping* and *reconciling trees* were introduced. They inspired research on *duplication-loss models* (we call them DL-models) and their extensions, for instance, models with a horizontal gene transfer [3,7,10]. All DL-models are believed to be biologically meaningful [13].

Most approaches to reconstruction of evolution history are *parsimonious*, that is, it is assumed that the solution with the minimal cost is the most likely one. There are several possible cost functions: size of the reconstructed tree, or the number of specified evolutionary events, e.g. gene duplications, or the total number of gene duplications and gene losses. The latter measure, called *mutation cost*, was particularly popular among researchers [11,15]. One of the crucial

☆ A preliminary version of this paper was presented in [8].

* Corresponding author. Tel.: +48 225544401; fax: +48 225544200.
E-mail addresses: gorecki@mimuw.edu.pl (P. Górecki), tiuryn@mimuw.edu.pl (J. Tiuryn).

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.05.019

terms in the DL-models is that of *reconciled tree* which represents the common evolutionary history of genes and species. In [2], authors present several definitions of a reconciled tree which were used recently in the literature. They prove that the definitions are equivalent and that the tree is minimal with respect to the size. Paper [2] still left open questions: (see [16]) *is the reconciled tree minimal with respect to the mutation cost* or *is it minimal with respect to the total number of gene duplications* (*duplication cost*). Also the question of uniqueness of such a tree was left open. In the present paper, we answer all these questions. We build a formal framework of evolutionary scenarios which represents a common history of genes and species under the assumption that only gene duplications, losses and speciations may occur. These scenarios are called here DLS-trees. [1] We claim that a DLS-tree can be used to represent all possible evolutionary scenarios under the above assumptions. Given a DLS-tree T, we show how to retrieve from T a gene tree $gene(T)$, as well a species tree $spec(T)$. A DLS-tree is similar to the concept of a reconciliation (see [1]) for a given species tree and a gene tree. Although, in the definition of a DLS-tree we do not use any particular gene or species tree.

We introduce a system of rules for transforming DLS-trees. This is a certain kind of a term rewrite system. It has pleasing mathematical properties: soundness, [2] confluence and strong normalization. We prove that, a DLS-tree in normal form has minimal size, minimal mutation cost, and minimal duplication cost. It follows from our theory that for every DLS-tree T in normal form, if D_T is the set of all DLS-trees which have the normal form T, then T is the unique tree in D_T among all trees in D_T having the same mutation cost. We show an example that the uniqueness property fails when mutation cost is replaced by duplication cost. We show a one-to-one correspondence between the reconciled trees and the DLS-trees in normal form. Thus, the theory build in this paper is immediately applicable to reconciled trees. We obtain a formula for computing the total number of duplications and losses in a reconciled tree, as a function of a gene tree and a species tree.

A formal analysis of these formulas in the context of reconciled trees can be found in [5,17].

First we define basic terms and DLS-trees. Then we show how to extract a gene and a species tree from a DLS-tree. In Sections 5 and 6, we present the system of rules and prove soundness, completeness and confluence. In Section 6.7, we present an example of a hierarchy of DLS-trees together with all their reductions (Fig. 13). In Section 7, we present formulas for computing the tree in normal form (for a given species tree and a given gene tree) and the number of duplications and losses. Finally, we show a one-to-one correspondence between the reconciled trees and the DLS-trees in normal form.

2. Gene and species trees

Let $\mathcal{I}$ be a set of species. *A gene tree* is a rooted binary directed tree whose leaves are labelled by the elements from $\mathcal{I}$. The labelling need not be one to one. *A species tree* is a gene tree [3] whose leaves are uniquely labelled.

Let T be a gene tree. For a node v of T, by $T(v)$ we denote the subtree of T rooted in v. For each node v of a gene tree T, we define a multiset $\mathfrak{m}_v^T = \{x_1^{i_1}, x_2^{i_2}, \ldots, x_k^{i_k}\}$, where $i_j > 0$ is the number of leaves labelled x_j in $T(v)$. Similarly, for v, we define *a cluster* as a set $m_v^T = \{x_1, x_2, \ldots, x_m\}$. Let $\mathfrak{M}^T$ denote the multiset $\{\mathfrak{m}_v^T | v \in V\}$. In order to make the notation more readable, m_v^T will be denoted by $x_1x_2 \ldots x_m$. Note that if T is a species tree, then $\mathfrak{m}_v^T = m_v^T$. We denote by $root(T)$ the root of T, and by $L(T)$ the set of all labels (i.e., species) in T. For example, see Fig. 1.

We use the standard *nested-parenthesis notation* for trees:

- The empty tree will be denoted by $\emptyset$.
- An one-element tree, whose node is labelled by a is denoted by a.
- If T_p and T_q are two non-empty trees with roots p and q, respectively, then (T_p, T_q) is a tree whose root has two children: p and q. The trees T_p and T_q are rooted in (T_p, T_q) at the nodes p and q, respectively.

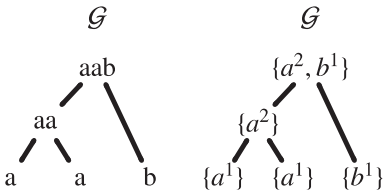
Lemma 1. *If T and S are species trees and $\mathfrak{M}^T = \mathfrak{M}^S$, then $T = S$.*

Proof. It follows easily from the definition of the species tree. □

[1] DLS stands for *D*uplication, *L*oss and *S*peciation.

[2] If T reduces to T' then $gene(T) = gene(T')$ and $spec(T) = spec(T')$.

[3] A species tree is a special case of a gene tree in the model.



$\mathfrak{M}^{\mathcal{G}} = \{\{a^1\}^2, \{b^1\}^1, \{a^2\}^1, \{a^2, b^1\}^1\}$
$L(\mathcal{G}) = \{a, b\}$

Fig. 1. Gene tree $\mathcal{G} = ((a, a), b)$.

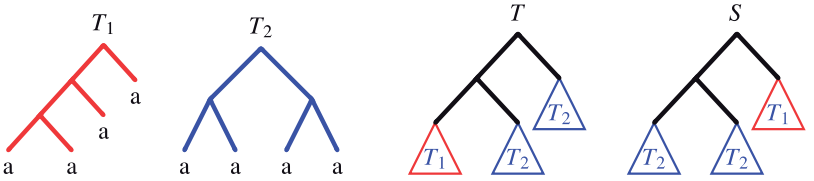


Fig. 2. Counterexample for gene trees.

Lemma 1 fails for gene trees. Fig. 2 presents the counterexample. Let $T_1 = (a, (a, (a, a)))$, $T_2 = ((a, a), (a, a))$, $T = ((T_1, T_2), T_2)$ and $S = ((T_2, T_2), T_1)$. We can easily check that $\mathfrak{M}^T = \mathfrak{M}^S = \{\{a^1\}^{12}, \{a^2\}^5, \{a^3\}^1, \{a^4\}^3, \{a^8\}^1, \{a^{12}\}^1\}$ but the gene tree are different.

A multiset $\mathcal{M}$ is said to *determine a species tree* if $\mathfrak{M}^S = \mathcal{M}$, for some species tree $\mathcal{S}$. By $\mathcal{T}(\mathcal{M})$ we denote the tree determined by $\mathcal{M}$.

Theorem 2. *A multiset $\mathcal{M}$ determines a species tree if and only if*
(M1) *if $\mathcal{M}$ is non-empty, then $\bigcup \mathcal{M} \in \mathcal{M}$,*
(M2) *for all $a \in \bigcup \mathcal{M}$, $\{a\} \in \mathcal{M}$,*
(M3) *for all $A \in \mathcal{M}$ such that A is not a singleton:*

$$\{X \mid X \in \mathcal{M} \text{ and } X \subsetneq A\}$$

contains exactly two maximal (in the sense of inclusion) disjoint sets.

Proof. (=>) Let us assume that T is a species tree determined by $\mathcal{M}$. (M1) is satisfied by the root of T. (M2) obviously refers to the leaves of T. The last condition is satisfied by the internal nodes of T, that is, if a and b are children of an internal node, then a and b determine the two maximal disjoint sets.

(<=) We proceed by induction on the size of $\mathcal{M}$. If $\mathcal{M}$ is an empty set, take the empty tree. If $\mathcal{M} = \{\{x\}\}$, then the tree is x, that is, a tree with one node labelled x. Let us assume that, for all $\mathcal{M}$ satisfying (M1–3) and $|\mathcal{M}| < n$, we can construct the tree $\mathcal{T}(\mathcal{M})$. Consider $\mathcal{M}$ such that $|\mathcal{M}| = n$. We show the construction of the tree $\mathcal{T}(\mathcal{M})$. We consider: $A = \bigcup \mathcal{M}$ (note that $\mathcal{M}$ is not a singleton). Let B and C be the two maximal disjoint sets B and C obtained from (M3) for A. It is easy to show that $B \cup C = A$. We define $\mathcal{M}_B = \{X \mid X \in \mathcal{M} \text{ and } X \subseteq B\}$ and similarly $\mathcal{M}_C$. $\{\{A\}, \mathcal{M}_B, \mathcal{M}_C\}$ is a partition of $\mathcal{M}$ and (M1)–(M3) are satisfied for $\mathcal{M}_B$ and $\mathcal{M}_C$. Now, by the induction hypothesis, we can obtain $\mathcal{T}(\mathcal{M}_B)$ and $\mathcal{T}(\mathcal{M}_C)$. Finally, the species tree determined by $\mathcal{M}$ is given by $(\mathcal{T}(\mathcal{M}_B), \mathcal{T}(\mathcal{M}_C))$. □

3. DLS trees

Now, we define a crucial notion of a DLS-tree. Such a tree could be interpreted as "an evolutionary scenario representing history of genes in the context of species evolution".

First, we start with some biological motivations. "Evolution" part of Fig. 3 presents all aspects of the common evolution of genes and species under assumption that only gene duplications, gene losses or speciations are allowed.

The left tree presents an evolutionary species tree and its interpretation is clear. The rightmost tree presents an evolution of a family of genes which are related to the three species (i.e., each name denotes a species from which the

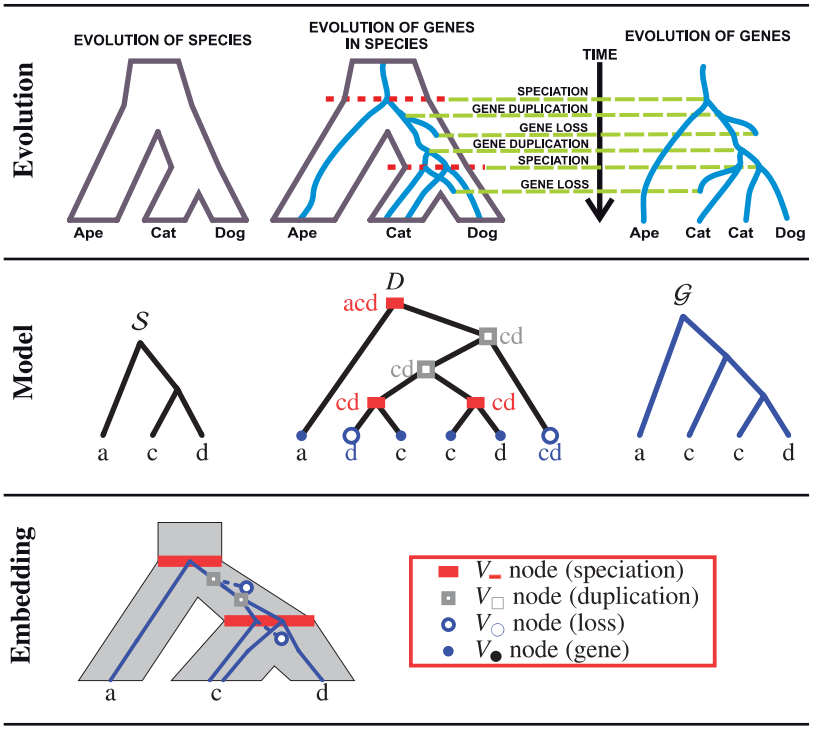


Fig. 3. Evolution, model and illustration of the embedding (a—ape, c—cat, d—dog).

sequence was obtained). We have four genes (called *homologs* related through common ancestry). We have two genes labelled by the species *cat*. Both genes are currently present in cat. This situation is a consequence of the second *gene duplication*. These genes are *paralogs* (they are most closely related through a duplication). For instance, genes labelled by *ape* and *dog* are *orthologs* (they are most closely related through a speciation). We see also that some of the gene lineages are lost. Here, we have two gene losses. Current methods of gene tree reconstruction (from gene sequences) cannot detect this kind of losses which are shown in Fig. 3. However, if we know the species tree and the gene tree, we can find evolutionary scenarios which explain the differences between them in terms of gene duplications and losses. One of them is shown in the middle tree. We see the embedding of the gene tree (right) into the species tree (left). It is should be clear that this kind of embedding is biologically correct. Note that the internal nodes of the gene tree are related either to speciations or to gene duplications.

Our goal is to present a mathematical model of the evolutionary scenario. Let us adopt the following symbols □ (duplication), ◯ (loss), ▬ (speciation) and • (gene).

A *DLS-tree* is either an empty tree, or a binary rooted tree $T = (V, E)$ such that the elements of V are labelled by non-empty subsets of $\mathcal{I}$. For $v \in V$, let Λ_v denote the label of v. V is divided into four disjoint sets: $V_\bullet$, $V_\bigcirc$, $V_\square$ and $V_{\blacksquare}$ such that [4]

(D1) if $v \in V_\bullet$, then v is leaf in T labelled by a species a (v is called a gene node),
(D2) if $v \in V_\bigcirc$, then v is leaf in T (v is called a loss node),
(D3) if $v \in V_\square$, then v has two children a and b such that $\Lambda_a = \Lambda_b = \Lambda_v$ (v is called a duplication node),
(D4) if $v \in V_{\blacksquare}$, then v has two children a and b such that $\Lambda_a \cup \Lambda_b = \Lambda_v$ and $\Lambda_a \cap \Lambda_b = \emptyset$ (v is called a speciation node),
(D5) for all $v, w \in V$ such that $\Lambda_v \cap \Lambda_w \neq \emptyset$, we have either $\Lambda_v \subseteq \Lambda_w$ or $\Lambda_v \supseteq \Lambda_w$.

By $\mathfrak{Labels}^T$ we denote the set of all labels in T. Let $\Lambda(T)$ denotes the label of the root of T.

[4] Sometimes we use an upper index to distinguish objects from different trees.

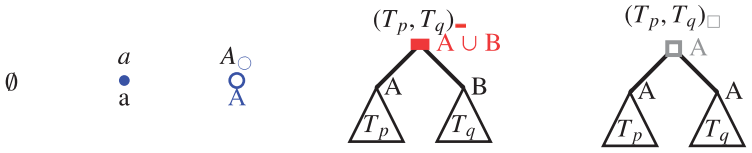


Fig. 4. DLS-trees and its notation.

With a DLS-tree T we associate *a cost* which is the total number of gene duplications and losses in T. This cost is known in the literature as *a mutation cost* [11].

In "model" part of Fig. 3 we present a DLS-tree D. *Embedding* is "evolutionary interpretation" of D in the context of the species tree $\mathcal{S}$. We do not define formally embeddings. It should be clear that every DLS-tree whose labels are clusters in a species tree $\mathcal{S}$ can be embedded into $\mathcal{S}$.

Sometimes we will use a linear (term-like) representation of DLS-trees similar to nested-parenthesis notation. The following productions define the terms:

$$T \to \emptyset \mid a \mid A_{\bigcirc} \mid (T, T)_{\blacksquare} \mid (T, T)_{\square},$$

where $a \in \mathcal{I}$, A is non-empty set of species, and the rest of the symbols except T are terminals. The interpretation of the terms is as follows (see Fig. 4):

- $\emptyset$ is the empty DLS-tree,
- a denotes a DLS-tree with a single gene node labelled by $\{a\}$,
- $A_{\bigcirc}$ denotes a DLS-tree with a single loss node labelled by A,
- if T_p and T_q are two non-empty DLS-trees with roots p and q, respectively, then $(T_p, T_q)_{\square}$ is a DLS-tree whose root is a duplication node and has two children: p and q. The trees T_p and T_q are rooted in $(T_p, T_q)_{\square}$ at the nodes p and q, respectively,
- $(T_p, T_q)_{\blacksquare}$ is defined analogously to the previous case (here the root is a speciation node).

For example, the tree D in Fig. 3 can be described as $(a, (((d_{\bigcirc}, c)_{\blacksquare}, (c, d)_{\blacksquare})_{\square}, cd_{\bigcirc})_{\square})_{\blacksquare}$.

4. Extracting evolutionary information from DLS-trees

We explain how to extract, from a given DLS-tree, a gene tree and a species tree, relying on information contained in its labels. By $\mathbb{DLS}$ we denote the set of all DLS-trees.

4.1. Extracting gene trees

We start with the gene tree. For a set of leaves L in T, let T^L be the smallest subtree of T containing L as its set of leaves. The *homomorphic tree* $T|_L$ of T induced by L is the tree obtained from T^L by contracting all nodes of degree 2 except for its root (i.e., for each such a node x: create an edge connecting the parent of x with the child of x; remove x and all edges incident on it) [2,12]. Now, we can use the homomorphic tree to get the gene tree from a DLS-tree. Let T be a DLS-tree. We set $gene(T)$ to be the gene tree defined by $T|_{V_\bullet}$. The labels of leaves in $gene(T)$ are inherited from T. One can easily check that for the trees in Fig. 3, we have $gene(D) = \mathcal{G}$.

This operation could be also defined equivalently by structural induction:

(1) $gene(\emptyset) = \emptyset$,
(2) $gene(a) = a$,
(3) $gene(A_{\bigcirc}) = \emptyset$,
(4)
$$gene((T_1, T_2)_*) = \begin{cases} \emptyset & \text{if } gene(T_1) = \emptyset = gene(T_2), \\ gene(T_1) & \text{if } gene(T_1) \neq \emptyset = gene(T_2), \\ gene(T_2) & \text{if } gene(T_1) = \emptyset \neq gene(T_2), \\ (gene(T_1), gene(T_2)) & \text{otherwise,} \end{cases}$$

where $* \in \{\blacksquare, \square\}$.

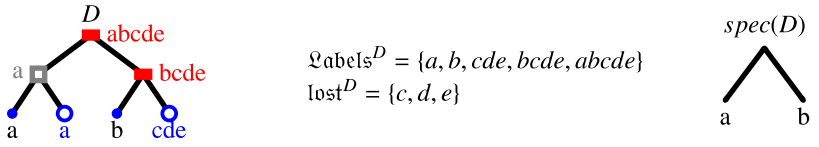


Fig. 5. Incomplete DLS-tree D.

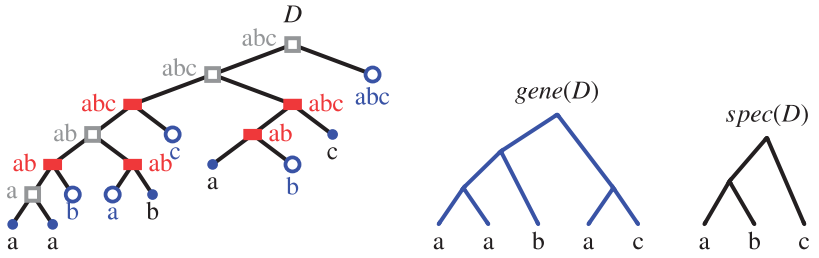


Fig. 6. A complete DLS-tree D and extracted gene and species trees. The cost equals 9.

It can be shown that $L(gene(D)) = \{\Lambda_v \mid v \in V_\bullet^D\}$ for D a DLS-tree. We omit the easy proof. For example, see Fig. 6.

4.2. Extracting species trees

In this section, we present the extraction of the species tree. The natural question is whether the set $\mathfrak{Labels}$ determines a species tree. Fig. 5 presents a DLS-tree which does not satisfy this property. We see that the tree contains an incomplete information on a species relationship due to the loss nodes.

To solve this problem we have to identify species for which the reconstruction (from labels) will give a species tree. We call a species s *lost in* T, if s occurs only in loss nodes of T. Formally, the set of lost species can be defined by

$$\mathfrak{lost}^T = \Lambda(T) \setminus \bigcup\{\Lambda_v^T \mid v \in V_\bullet^T\}. \tag{1}$$

For example, see Figs. 5 and 6. We claim that if we remove lost species from all labels of a DLS-tree, then we will be able to reconstruct the species tree.

Lemma 3. *Let T be a DLS-tree. Then*

$$\{\Lambda_v \setminus \mathfrak{lost}^T \mid v \in T\} \setminus \{\emptyset\} \tag{2}$$

determines a species tree.

Proof. Let $\mathcal{M}$ be the set defined in (2). We prove that M satisfies the properties (M1–3) from Theorem 2.

(M1) Let us assume that $\mathcal{M}$ is non-empty. It should be noted that $\bigcup \mathfrak{Labels}^T$ is the label of the root of T (i.e., $\Lambda(T)$). Thus, $\bigcup \mathcal{M} = (\bigcup\{\Lambda_v \mid v \in T\}) \setminus \mathfrak{lost}^T = \Lambda(T) \setminus \mathfrak{lost}^T \in \mathcal{M}$.

(M2) Let $a \in \bigcup \mathcal{M}$. Then $a \in \Lambda(T) \setminus \mathfrak{lost}^T$. Thus, from the definition of $\mathfrak{lost}^T$ (see (1)), there exists a gene node in T labelled by $\{a\}$. Finally, we obtain $\{a\} = \{a\} \setminus \mathfrak{lost}^T \in \mathcal{M}$.

(M3) Let $A \in \mathcal{M}$, not a singleton. Let $\tilde{A}$ be the least set in the sense of inclusion such that there exists a node v in T such that $\tilde{A} = \Lambda_v$ and

$$A = \Lambda_v \setminus \mathfrak{lost}^T. \tag{3}$$

Since $A \in \mathcal{M}$ there exists at least one node in T which satisfies (3). If v and v' satisfy (3), then by (D5) we have $\Lambda_v \subseteq \Lambda_{v'}$ or $\Lambda_v \supseteq \Lambda_{v'}$. Hence $\tilde{A}$ exists and is well defined.

$\tilde{A}$ does not contain only lost species. Thus, from (D3–5), we conclude that, for each $a \in A$, there exists a path in T whose labels can be presented as a sequence $\tilde{A} \supset P_1^a \supset P_2^a \supset \cdots \supseteq \{a\}$. Let us consider $\mathcal{P} = \{P_1^a\}_{a \in A}$. By (D4) and (D5), $\mathcal{P}$ contains two disjoint elements $\tilde{B}$ and $\tilde{C}$ such that $\tilde{B} \cup \tilde{C} = \tilde{A}$. Thus, we proved that the set

$$\{\Lambda_v \mid v \in T, \Lambda_v \subset \tilde{A}\}$$

contains two maximal sets in the sense of inclusion (i.e., $\tilde{B}$ and $\tilde{C}$).

Let $B = \tilde{B} \setminus \mathrm{lost}^T$ and $C = \tilde{C} \setminus \mathrm{lost}^T$. This is obvious from the construction that A and B are non-empty. Having this, we conclude that B and C are the two maximal sets in

$$\{\Lambda_v \setminus \mathrm{lost}^T \mid v \in T, \Lambda_v \setminus \mathrm{lost}^T \subset A\} \setminus \{\emptyset\}.$$

Note that the above set is the set considered in (M3). □

We denote the species tree determined by (2) by $spec(T)$. Let us notice that, for a DLS tree T, $L(gene(T)) = L(spec(T))$.

The species tree for the incomplete DLS-tree D is presented in Fig. 5. Another example is presented in Fig. 7 (tree T).

We call a DLS-tree *complete*, if it has no lost species. Fig. 6 presents an example of a complete DLS-tree.

4.3. Completion of DLS-trees

In this subsection we show how to transform an incomplete DLS-tree into a complete one. Informally, this transformation gives the largest complete DLS-tree which is included (as a subtree) in a given DLS-tree.

First, we define a mapping $cmpl_X : \mathbb{DLS} \to \mathbb{DLS}$ for $X \subseteq \mathcal{I}$:

(1) $cmpl_X(\emptyset) = \emptyset$,

(2) $cmpl_X(a) = \begin{cases} \emptyset & \text{if } a \in X, \\ a & \text{otherwise.} \end{cases}$

(3) $cmpl_X(A_{\bigcirc}) = \begin{cases} \emptyset & \text{if } A \subseteq X, \\ (A \setminus X)_{\bigcirc} & \text{otherwise.} \end{cases}$

(4) $cmpl_X((T_1, T_2)_*) = \begin{cases} \emptyset & \text{if } cmpl_X(T_1) = \emptyset = cmpl_X(T_2), \\ cmpl_X(T_1) & \text{if } cmpl_X(T_1) \neq \emptyset = cmpl_X(T_2), \\ cmpl_X(T_2) & \text{if } cmpl_X(T_1) = \emptyset \neq cmpl_X(T_2), \\ (cmpl_X(T_1), cmpl_X(T_2))_* & \text{otherwise,} \end{cases}$

where $* \in \{$▬, □$\}$.

Lemma 4. *Let $T \in \mathbb{DLS}$ and $X \subseteq \mathcal{I}$. Then $cmpl_X(T)$ is a DLS-tree, and if this tree is non-empty, then the label of its root is $\Lambda(T) \setminus X$.*

Proof. We proceed by induction on the structure of T. It is clear that the properties hold for the empty and one-element trees. We assume that the properties are satisfied for the trees of the size less than n. Now, we prove the thesis for the trees of the size n. It is sufficient to consider the last case of fourth item in the definition of *cmpl*. The rest is clear.

First, we show that (D3)–(D5) are satisfied.

(D5) is satisfied for $cmpl_X(T_1)$ and $cmpl_X(T_2)$, separately, by the induction hypothesis. For $i = 1, 2$, let v_i be a node in $cmpl_X(T_i)$. Then, by the induction hypothesis, there exists a node w_i in T_i such that

$$\Lambda_{w_i}^{T_i} \setminus X = \Lambda_{v_i}^{cmpl_X(T_i)}.$$

Let us assume that $\Lambda_{v_1}^{cmpl_X(T_1)} \cap \Lambda_{v_2}^{cmpl_X(T_2)} \neq \emptyset$. Without loss of generality, by (D5) for T, we may assume that

$$\Lambda_{w_1}^{T_1} \subseteq \Lambda_{w_2}^{T_2}.$$

Thus, we have

$$\Lambda^{T_1}_{w_1} \setminus X \subseteq \Lambda^{T_2}_{w_2} \setminus X.$$

This proves (D5), for all nodes in $cmpl_X(T)$ except its root. Note that from (D3) and (D4) we have

$$\Lambda(cmpl_X(T)) = \Lambda(cmpl_X(T_1)) \cup \Lambda(cmpl_X(T_2))$$

and, for all nodes, the condition (D5) is satisfied.

Let us assume that $* = \square$. We show that (D3) holds for the root of $cmpl_X(T)$:

$$\begin{aligned} \Lambda(cmpl_X(T_1)) &= \Lambda(T_1) \setminus X && \text{by the induction hypothesis} \\ &= \Lambda(T_2) \setminus X && \text{by (D3) for } T \\ &= \Lambda(cmpl_X(T_2)) && \text{by the induction hypothesis.} \end{aligned}$$

Finally, we have $\Lambda(cmpl_X(T_1)) = \Lambda(cmpl_X(T))$. It is also clear that the label of the root of $cmpl_X(T)$ equals $\Lambda(T) \setminus X$.

Now, we consider the second case, that is, $* = -$. We show that (D4) holds for the root of $cmpl_X(T)$. We need to prove that

$$\Lambda(cmpl_X(T_1)) \cap \Lambda(cmpl_X(T_2)) = \emptyset.$$

The left side of the above equation gives

$$(\Lambda(T_1) \setminus X) \cap (\Lambda(T_2) \setminus X) = (\Lambda(T_1) \cap \Lambda(T_2)) \setminus X.$$

By (D4) for T, this is $\emptyset$. Finally, for the speciation case we have

$$\begin{aligned} \Lambda(T) \setminus X &= (\Lambda(T_1) \cup \Lambda(T_2)) \setminus X && \text{by (D4) for } T \\ &= (\Lambda(T_1) \setminus X) \cup (\Lambda(T_2) \setminus X) \\ &= \Lambda(cmpl_X(T_1)) \cup \Lambda(cmpl_X(T_2)) && \text{by the induction hypothesis} \\ &= \Lambda(cmpl_X(T)). \qquad \square \end{aligned}$$

Note that if T is a DLS-tree, then $cmpl_\emptyset(T) = T$. Now, we define the operation $cmpl : \mathbb{DLS} \to \mathbb{DLS}$. For a DLS-tree T, let $cmpl(T) = cmpl_{\mathfrak{lost}^T}(T)$. The most important property of this function is stated below:

Proposition 5. *If T is a DLS-tree, then $cmpl(T)$ is a complete DLS-tree.*

Proof. We need to prove $\mathfrak{lost}^{cmpl(T)} = \emptyset$. By the definition of $\mathfrak{lost}$, it is sufficient to show

$$\bigcup_{v \in V_\bullet^{cmpl(T)}} \Lambda_v^{cmpl(T)} = \Lambda(cmpl(T)).$$

The inclusion $\subseteq$ is clear from (D3) and (D4).

("$\supseteq$") Let $a \in \Lambda(cmpl(T))$. We notice that $a \notin \mathfrak{lost}^T$. Thus, there exists a gene node in T labelled by $\{a\}$. So $cmpl_{\mathfrak{lost}^T}(a) = a$, hence there exists a gene node in $cmpl(T)$ labelled by $\{a\}$. $\square$

Moreover, this operations preserves *spec*:

Proposition 6. *If T is a DLS-tree, then $spec(T) = spec(cmpl(T))$.*

Proof. It follows easily from Lemma 4 and the definition of $cmpl(T)$. $\square$

Fig. 7 presents an example of an incomplete DLS-tree T with lost species d and e. Also, we present an extraction of its complete DLS-tree $cmpl(T)$ and a species tree $spec(T)$.

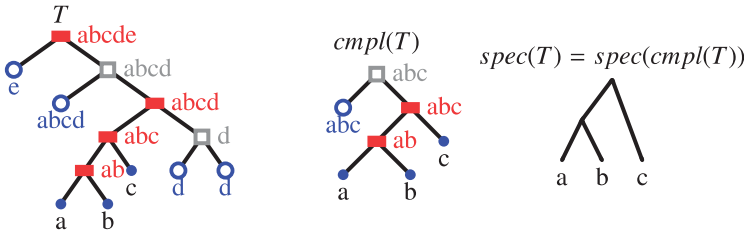


Fig. 7. Extraction of a complete DLS-tree.

SPEC	$\dfrac{(A_{\circ}, B_{\circ})_{-}}{A \cup B_{\circ}}$	Type I
DUP	$\dfrac{(\mathcal{R}, \Lambda(\mathcal{R})_{\circ})_{\square}}{\mathcal{R}}$	Type I
TMOVE	$\dfrac{((C_{\circ}, \mathcal{P})_{-}, (C_{\circ}, \mathcal{Q})_{-})_{\square}}{(C_{\circ}, (\mathcal{P}, \mathcal{Q})_{\square})_{-}}$	Type II
CLOST	$\dfrac{((\mathcal{P}, \Lambda(\mathcal{Q})_{\circ})_{-}, (\Lambda(\mathcal{P})_{\circ}, \mathcal{Q})_{-})_{\square}}{(\mathcal{P}, \mathcal{Q})_{-}}$	Type II

Fig. 8. DLS rules.

5. DLS rules

We define DLS rules (we call them rules). They will be used to transform DLS-trees. Each rule is defined by P/Q, where P (premise) and Q (conclusion) are DLS-trees. By a redex of a rule R, we mean a node v in a tree to which the premise of R is applicable. A DLS-tree T can be transformed into T' by a rule R in node v if and only if

- P equals $T(v)$,
- T' is constructed from T by replacing this subtree by the tree Q.

We denote by $R(T, v)$ the result of reduction.

If T is reduced to T' in one step, this is indicated $T \to T'$. If T is reduced to T' in zero or more steps, this is indicated $T \twoheadrightarrow T'$. We use $\to^{-1}$ to denote backward transformation, that is, $T' \to^{-1} T$ if and only if $T \to T'$. The rules are presented in Fig. 8 and its biological interpretation in Fig. 9.

It should be clear that an application of any rule to a DLS-tree yields a DLS-tree and the reduction decreases the cost.

For a DLS-tree T and a node v in T, we call a subtree T' of $T(v)$ *principal for a rule R in v* if v is a redex of R and in an application of R in v

- if R is DUP, then T' is the tree defined by $\mathcal{R}$,
- if R has type II, then T' is the tree defined by $\mathcal{P}$ or $\mathcal{Q}$.

6. Properties of the system and DLS-trees

In this section, we present important properties of DLS-trees and the system. First prove that our system is sound, then we define semi-normal and fat DLS-trees. Finally, we prove completeness and confluency of the system.

6.1. Soundness

The following Proposition shows soundness of the system:

Proposition 7 (*Soundness*). *If $T \to T'$, then $gene(T) = gene(T')$ and $spec(T) = spec(T')$.*

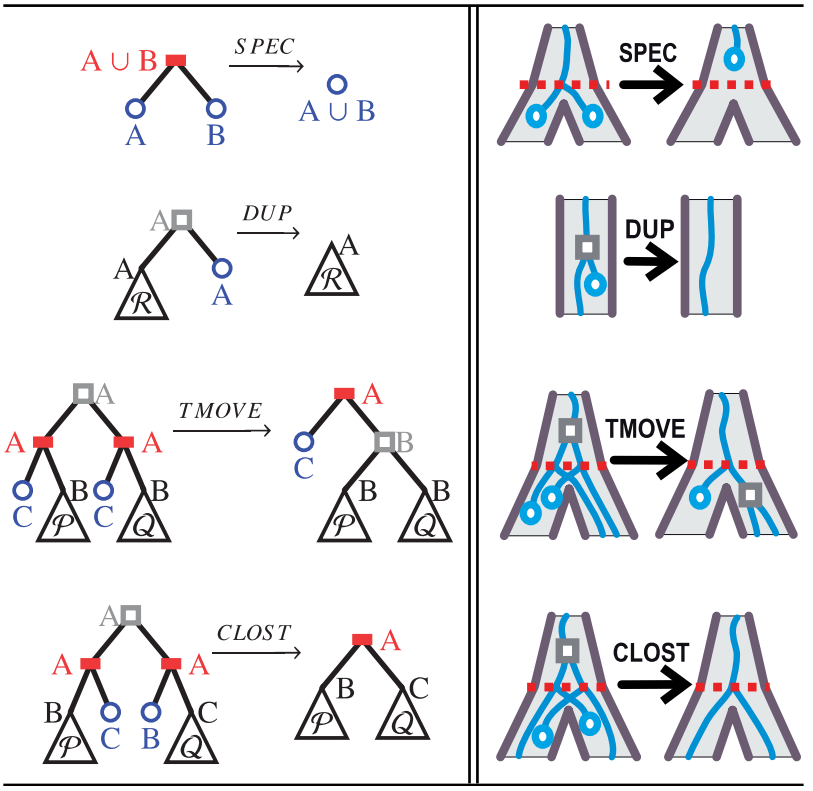


Fig. 9. Rules and their biological interpretations.

Proof. We proceed by induction. First, for all DLS rules, we prove that $gene(P) = gene(Q)$, where P is a premise and Q a consequence. We use the nested parenthesis notation adapted for gene trees.

SPEC $gene((A_{\bigcirc}, B_{\bigcirc})_{\blacksquare}) = \emptyset = gene(A \cup B_{\bigcirc})$,
DUP $gene((\mathcal{R}, \Lambda(\mathcal{R})_{\bigcirc})\square) = gene(\mathcal{R})$,
TMOVE $gene(((\mathcal{C}_{\bigcirc}, \mathcal{P})_{\blacksquare}, (\mathcal{C}_{\bigcirc}, \mathcal{Q})_{\blacksquare})\square) = \tau(gene(\mathcal{P}), gene(\mathcal{Q})) = gene((C_{\bigcirc}, (\mathcal{P}, \mathcal{Q})\square)_{\blacksquare})$,
CLOST $gene(((\mathcal{P}, \Lambda(\mathcal{Q})_{\bigcirc})_{\blacksquare}, (\Lambda(\mathcal{P})_{\bigcirc}, \mathcal{Q})_{\blacksquare})\square) = \tau(gene(\mathcal{P}), gene(\mathcal{Q})) = gene((\mathcal{P}, \mathcal{Q})_{\blacksquare})$,
where

$$\tau(T_1, T_2) = \begin{cases} \emptyset & \text{if } T_1 = \emptyset = T_2, \\ T_1 & \text{if } T_1 \neq \emptyset = T_2, \\ T_2 & \text{if } T_1 = \emptyset \neq T_2, \\ (T_1, T_2) & \text{otherwise.} \end{cases}$$

This completes the first part of the proof.

Now we prove the second equation. Let us notice that $\Lambda(T) = \Lambda(T')$ and

$$\bigcup\{\Lambda_v^T \mid v \in V_\bullet^T\} = \bigcup\{\Lambda_v^{T'} \mid v \in V_\bullet^{T'}\}.$$

Easy proof is omitted. This yields

$$\mathfrak{lost}^T = \mathfrak{lost}^{T'}.$$

First, we consider an application of the rules CLOST, TMOVE and DUP. In these cases we have

$$\mathfrak{Labels}^T = \mathfrak{Labels}^{T'}. \tag{4}$$

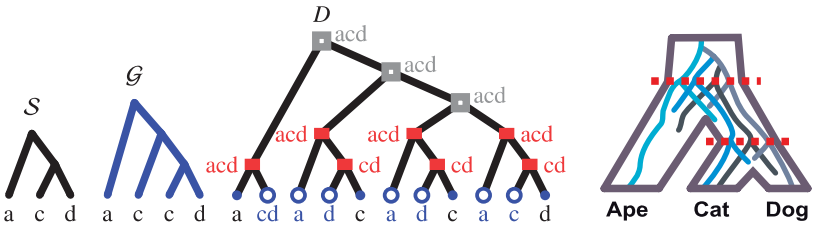


Fig. 10. A fat DLS-tree D, its gene tree $\mathcal{G}$ and its species tree $\mathcal{S}$, and an evolutionary interpretation of D (see Fig. 3).

For a tree T, let $\mathcal{M}(T)$ be the set defined by (2) (see Lemma 3). By (4), we have

$$\mathcal{M}(T) = \mathcal{M}(T'). \tag{5}$$

Now, we show that the above equality holds for the rule SPEC.

(1) If $A \subseteq \mathfrak{lost}^T$, then there is no subset of A, which will be present in $\mathcal{M}(T)$ or $\mathcal{M}(T')$. Thus, in this case, (5) is satisfied.

(2) If $B \not\subset \mathfrak{lost}^T$ and $C \subseteq \mathfrak{lost}^T$, then $B \cap \mathfrak{lost}^T = A \cap \mathfrak{lost}^T$ and there is no subset of C, which will present in $\mathcal{M}(T)$ or $\mathcal{M}(T')$. Again, (5) is satisfied.

(3) If $B \not\subset \mathfrak{lost}^T$ and $C \not\subset \mathfrak{lost}^T$, then there exists a speciation node (or nodes) in T (and in T') labelled by A. By (D4) and (D5), the children of this node are labelled by B and C. Thus, $B \cap \mathfrak{lost}^T$ and $C \cap \mathfrak{lost}^T$ occur in $\mathcal{M}(T)$ and $\mathcal{M}(T')$, this yields (5).

For each rule, from Lemma 3 and (5), we have $spec(T) = spec(T')$. □

6.2. *Semi-normal and fat trees*

A DLS-tree containing no type I redexes is called *a semi-normal* tree. A semi-normal tree T is called *fat*, if the following conditions are satisfied:

- every duplication node has label $\Lambda(T)$,
- each speciation node has exactly one lost child.

Fig. 10 presents an example of a fat DLS-tree D. $\mathcal{G}$ and $\mathcal{S}$ are taken from Fig. 3.

Lemma 8. *If T is fat, then each child of a duplication node is either a duplication node or is the root of the tree*

$$(B_{1\bigcirc}, (B_{2\bigcirc}, \ldots (B_{k\bigcirc}, a)_{\blacksquare} \ldots)_{\blacksquare})_{\blacksquare}, \tag{6}$$

where $a \in \mathcal{I}$ and $k \geqslant 0$.

Proof. If v is a duplication node, then its children are labelled by $\Lambda(T)$. Let w be a child of v. We have to consider three cases:

- w is a duplication node (obvious),
- w is a leaf and the tree for v is a, where a is the label of w (also $\Lambda(T) = \{a\}$).
- w is a speciation node and there are no more duplication nodes in $T(w)$ (their labels do not equal $\Lambda(T)$); thus, all internal nodes in the subtree $T(w)$ are speciations.

This completes the proof. □

We call the tree defined by (6) *a chain tree*. The label of the only gene node will be called *a target*.

Extracting gene and species trees from fat trees is quite natural (see Fig. 10):

Proposition 9. *Let us assume that T is fat. Then*

(1) *$gene(T)$ is constructed from T by replacing each chain tree in T by a single node labelled by its target,*

(2) *if there is no lost species in T, then $spec(T)$ is determined by $\bigcup_{C \in \mathcal{C}(T)} \mathfrak{M}^C$, where $\mathcal{C}(T)$ denotes the set of all chain trees in T.*

Proof. (1) It is easy to notice that if C is a chain tree with a target a, then $gene(C) = a$. By Lemma 8, we conclude that all duplication nodes will be transformed by $gene$ into internal nodes of the gene tree.

(2) It should be noted that the root of each chain tree is labelled by $\Lambda(T)$. So

$$\mathfrak{Labels}^T = \bigcup_{C \in \mathcal{C}(T)} \mathfrak{M}^C.$$

The above set equals the set defined by (2) from Lemma 3. From this lemma, we get the species tree. □

Also converse holds:

Proposition 10. *Given a gene tree $\mathcal{G}$ and a species tree $\mathcal{S}$ such that $L(\mathcal{G}) \subseteq L(\mathcal{S})$. There exists a unique fat tree T such that $gene(T) = \mathcal{G}$, $\mathfrak{Labels}^T \subseteq \mathfrak{M}^{\mathcal{S}}$ and $L(\mathcal{S}) = \Lambda(T)$.*

Proof. For each label $a \in L(\mathcal{G})$, we define the chain tree $\mathcal{S}_a$ with a target a. Let $p_0 p_1 \dots p_k$ be the unique path in $\mathcal{S}$, where $p_0 = root(\mathcal{S})$ and p_n has label a. Let $\mathcal{S}_a$ be the chain tree (6), where $B_i = m^{\mathcal{S}}_{p_{i-1}} \setminus m^{\mathcal{S}}_{p_i}$ for $i = 1, 2, \dots, k$. Note that $\Lambda(\mathcal{S}_a) = L(\mathcal{S})$.

We show the construction of the fat tree T. Set each internal node of the gene tree be a duplication node labelled by $L(\mathcal{G})$. Each leaf labelled by a in this tree is replaced by the chain tree $\mathcal{S}_a$. It is quite easy to check that T is well defined. It follows from the uniqueness of $\mathcal{S}_a$ that T is unique. □

Corollary 11. *By Proposition* 10, *the duplication cost of T equals the number of the internal nodes in $\mathcal{G}$. It also follows that the number of gene losses in T equals*

$$\sum_{a \in \mathfrak{L}(\mathcal{G})} len(\mathcal{S}, a),$$

where $\mathfrak{L}(\mathcal{G})$ is the multiset of all leaf labels in $\mathcal{G}$ and $len(\mathcal{S}, a)$ denotes the length of the path in $\mathcal{S}$, whose start is the root of $\mathcal{S}$ and the end is the (*unique*) *node labelled by a in $\mathcal{S}$.*

We define $\sim$ to be the least equivalence relation on the set of DLS-trees which contains relation $\rightarrow$. Thus, if $T \sim T'$, then T can be transformed into T' by applying DLS rules zero or more times in any direction.

Proposition 12. *Every DLS-tree is equivalent to a fat tree.*

Proof. Let T be a DLS-tree. We consider the following rewrite rules:

- type I in the direction $\rightarrow$,
- type II in the direction $\rightarrow^{-1}$.

(A1) First, we eliminate iteratively all redexes of the rules DUP and SPEC. We get a semi-normal tree.

(A2) Let $\#_X(T)$ equals the number of duplication nodes in T labelled by $X \subseteq \mathcal{I}$. Let v be a speciation node in T such that its children are not lost. Let $T' = \mathrm{CLOST}^{-1}(T, v)$. Thus, we have

$$\#_A(T') > \#_A(T),$$
$$\#_X(T') = \#_X(T) \quad \text{for } X \neq A.$$

Apply CLOST^{-1}. After this step all speciation nodes have exactly one lost child. Note that the final tree is semi-normal.

(A3) Let us assume that v is a speciation node in T labelled by A and its child is a duplication node labelled by B. Note that the second child is lost (after (A2)). Thus, we can apply TMOVE^{-1}. Let $T' = TMOVE^{-1}(T, v)$. We have

$$\#_A(T') > \#_A(T),$$
$$\#_B(T') < \#_B(T),$$
$$\#_X(T') = \#_X(T) \quad \text{for } X \neq A, B.$$

Applying TMOVE^{-1} preserves the property that speciation nodes have exactly one lost child and does not introduce redexes of type I.

After finite number of steps, we get $\#_X = 0$ for all $X \neq \Lambda(T)$. It means that in the final tree all duplication nodes are labelled by $\Lambda(T)$. We have shown that T can be transformed into a fat tree. □

From the proof, we conclude that the procedure presented below will produce a fat tree from any DLS-tree:
- eliminate iteratively all redexes DUP and SPEC,
- eliminate all redexes of TMOVE^{-1},
- eliminate all redexes of CLOST^{-1}.

From this construction, we have the following property:

Corollary 13. *Every complete DLS-tree is equivalent to a complete fat tree.*

Observe that we can increase the cost of a fat tree by applying SPEC in direction $\rightarrow^{-1}$; in this way, we increase each $B_{\bigcirc}$ by at most $|B| - 1$, or by applying DUP in direction $\rightarrow^{-1}$; this can be done an unbounded number of times, increasing the number of the duplication nodes and introducing spurious loss nodes.

Note that applying transformations (A2) and (A3) (see the proof of Proposition 12) we get a tree with larger size. Thus, we conclude that a fat tree is the heaviest (in the sense of size) among all equivalent semi-normal trees.

Recall that a complete DLS-tree is a tree without lost species.

Proposition 14. *For complete DLS-trees T_1 and T_2, if $T_1 \sim T_2$, then there exists a unique complete fat tree equivalent to T_1 and T_2.*

Proof. By Corollary 13, there exist complete fat trees F_1 and F_2 equivalent to T_1 and T_2, respectively. Let $spec(T_1) = \mathcal{S}$. Notice that $\mathfrak{Labels}^{T_1} = \mathfrak{Labels}^{F_1} = \mathfrak{M}^{\mathcal{S}} = \mathfrak{Labels}^{T_2} = \mathfrak{Labels}^{F_2}$. By Propositions 7 and 10, we obtain uniqueness. □

6.3. Completeness

We can also prove completeness of the system.

Proposition 15 (*Completeness*). *Let T_1 and T_2 be complete DLS-trees such that $gene(T_1) = gene(T_2)$ and $spec(T_1) = spec(T_2)$. Then $T_1 \sim T_2$.*

Proof. By Corollary 13, there exist complete fat trees F_1 and F_2 equivalent to T_1 and T_2, respectively. By Proposition 7, for $i = 1, 2$, we have $gene(T_i) = gene(F_i)$ and $spec(T_i) = spec(F_i)$. Moreover, $\mathfrak{Labels}^{T_2} = \mathfrak{Labels}^{F_2} = \mathfrak{M}^{\mathcal{S}} = \mathfrak{Labels}^{T_2} = \mathfrak{Labels}^{F_2}$. By Proposition 10, we get $F_1 = F_2$. Hence $T_1 \sim T_2$. □

6.4. Confluency and normalization

We use $T \xrightarrow{=} T'$ if T' can be obtained from T by at most one reduction. The following proposition states that the system is weakly confluent.

Proposition 16 (*Weak confluence*). *Let T be a DLS-tree. Then, for each T_1 and T_2 such that $T \rightarrow T_1$ and $T \rightarrow T_2$, there exists T_3 such that $T_1 \xrightarrow{=} T_3$ and $T_2 \xrightarrow{=} T_3$.*

Proof. For $i = 1, 2$, let $T_i = R_i(T, v_i)$. Without loss of generality we may assume that both reductions are different.
- $R_1 = DUP$: We consider a pattern $P = (\mathcal{R}, \Lambda(\mathcal{R})_{\bigcirc})_{\square}$ rooted in v_1. Then, v_2 is
  - either outside of the subtree rooted by v_1 in T; in this case the pattern P is present in $R_2(T, v_2)$,
  - or in the subtree $\mathcal{R}$.

  We see that the applications are independent, i.e., $R_2(R_1(T, v_1), v_2) = R_1(R_2(T, v_2), v_1)$.

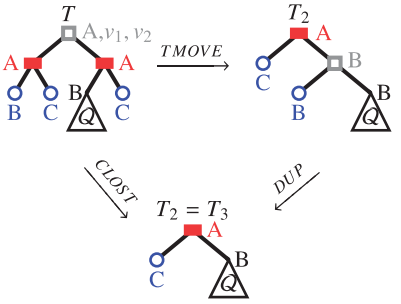


Fig. 11. TMOVE and CLOST case.

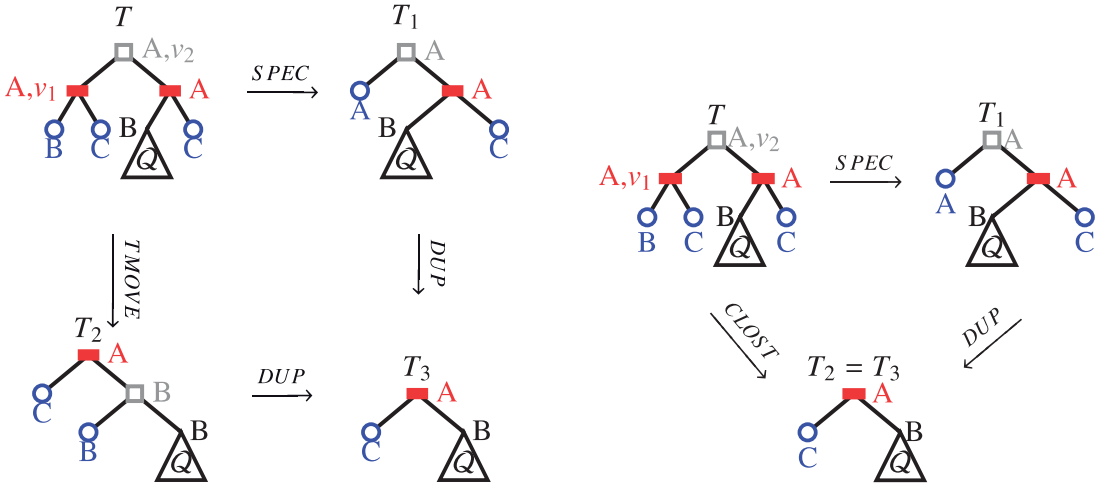


Fig. 12. SPEC and TMOVE, SPEC and CLOST cases.

- $R_1 = TMOVE$ *and* $R_2 = CLOST$ *case*: The dependence may occur only if the rules can be applied to the same node. In such a case, at least one of the principal trees for R_2 in v_2 is a one-element tree with a loss node. Without loss of generality we may assume that $B_{\bigcirc}$ is the principal tree with a loss node. The reductions are presented in Fig. 11.
- $R_1 = SPEC$ *and* $R_2 = TMOVE$: The dependence may occur only if at least one of the principal trees for R_2 in v_2 is a one-element tree with a loss node. Without loss of generality we may assume that $B_{\bigcirc}$ is the principal tree with a loss node. The reductions are presented in the left part of Fig. 12.
- $R_1 = SPEC$ *and* $R_2 = CLOST$ *case*: This case is similar to the previous one (see the right diagram in Fig. 12).
- $R_1 = R_2$: If the applied rules are equal, then their redexes are different. Simple analysis leads to the conclusion that they have to be independent. □

The following theorem states that our system is confluent. Recall that a DLS-tree in normal form is non-reducible.

Theorem 17 (*Confluence*). *Take a DLS-tree T. There exists a unique DLS-tree T^* (in normal form) such that every sequence of reductions in direction $\rightarrow$, which starts in T and terminates in normal form, yields T^*.*

Proof. The termination follows from the fact that every application of rules reduces the cost.

Let us assume that T_k^0 and T_0^n are in normal form such that $T_0^0 \rightarrow T_1^0 \rightarrow T_2^0 \rightarrow \cdots \rightarrow T_k^0$ and $T_0^0 \rightarrow T_0^1 \rightarrow T_0^2 \rightarrow \cdots \rightarrow T_0^n$, where $T_0^0 = T$. It follows from Proposition 16, that the diagram of reductions presented in (7) is well defined. The proof is straightforward (by induction).

$$
\begin{array}{ccccccc}
T_0^0 & \to & T_1^0 & \to & T_2^0 & \to \cdots \to & T_k^0 \\
\downarrow & & \downarrow^{\parallel} & & \downarrow^{\parallel} & & \downarrow^{\parallel} \\
T_0^1 & \stackrel{=}{\to} & T_1^1 & \stackrel{=}{\to} & T_2^1 & \stackrel{=}{\to} \cdots \to & T_k^1 \\
\downarrow & & \downarrow^{\parallel} & & \downarrow^{\parallel} & & \downarrow^{\parallel} \\
T_0^2 & \stackrel{=}{\to} & T_1^2 & \stackrel{=}{\to} & T_2^2 & \stackrel{=}{\to} \cdots \to & T_k^2 \\
\downarrow & & \downarrow^{\parallel} & & \downarrow^{\parallel} & & \downarrow^{\parallel} \\
\vdots & & \vdots & & \vdots & \ddots & \vdots \\
\downarrow & & \downarrow^{\parallel} & & \downarrow^{\parallel} & & \downarrow^{\parallel} \\
T_0^n & \stackrel{=}{\to} & T_1^n & \stackrel{=}{\to} & T_2^n & \stackrel{=}{\to} \cdots \stackrel{=}{\to} & T_k^n
\end{array}
\tag{7}
$$

By induction, we show that, for $i = 1, 2, \ldots, n$, T_k^i is in normal form and equal T_k^0. Analogously, for $i = 1, 2, \ldots, k$, T_j^n is in normal form and equal T_0^n. Finally, $T_k^0 = T_0^n$. □

Theorem 18. *For DLS-trees T_1 and T_2, we have $T_1 \sim T_2$ if and only if $T_1^* = T_2^*$.*

Proof. ($\Rightarrow$). Let $\xrightarrow{-1,1}$ denote the following relation on $\mathbb{DLS}$: $T \xrightarrow{-1,1} T'$ if and only if $T \to T'$ or $T' \to T$. Let $\| (T_1, T_2) \|$ denote the minimal number of reductions $\to$ and $\to^{-1}$ required to transform T_1 into T_2. We show by induction that, for each $d \in \{0, 1, \ldots\}$, if $T_1 \sim T_2$ and $\| (T_1, T_2) \| = d$ then $T_1^* = T_2^*$.

Let $d = 0$. We have $T_1 = T_2$. This case is clear, by Theorem 17.

Let $d > 0$. Assume that, for all T_1 and T_2 such that $\| (T_1, T_2) \| < d$, $T_1^* = T_2^*$.

Consider T_1 and T_2 such that $\| (T_1, T_2) \| = d$. Thus, we have $T_1 \xrightarrow{-1,1} S^1 \xrightarrow{-1,1} S^2 \xrightarrow{-1,1} \cdots \xrightarrow{-1,1} S^d \xrightarrow{-1,1} T_2$. By the induction hypothesis, we have

$$
\begin{array}{ccccccccccc}
T_1 & \xrightarrow{-1,1} & S_1 & \xrightarrow{-1,1} & S_2 & \xrightarrow{-1,1} & \ldots & \xrightarrow{-1,1} & S_d & \xrightarrow{-1,1} & T_2 \\
 & & \downarrow & & \downarrow & & & & \downarrow & & \downarrow \\
 & & S_1^* & = & S_2^* & = & \cdots & = & S_d^* & = & T_2^*
\end{array}
\tag{8}
$$

where, for each $i = 1, 2, \ldots, d$, S_i^* is the normal form of S_i. We have two cases:

- If $T_1 \to S_1$, then $T_1 \twoheadrightarrow S_1^*$. By the uniqueness of the normal form of T_1 we obtain $T_1^* = S_1^* = T_2^*$.
- If $T_1 \to^{-1} S_1$, then $S_1 \twoheadrightarrow T_1^*$. By the uniqueness of the normal form of S_1, we obtain $S_1^* = T_1^* = T_2^*$.

($\Leftarrow$). We have $T_1 \twoheadrightarrow T_1^* = T_2^* \twoheadleftarrow T_2$. Thus, $T_1 \sim T_2$. □

Corollary 19. *For a DLS-tree T, T^* is the unique tree with minimal cost in the set of all trees which are equivalent to T.*

6.5. Computing a fat tree

Algorithm 1. DLS-tree to a fat tree.

1. **Input:** DLS-tree T
2. **Output:** The unique fat tree equivalent to T
3. eliminate all redexes of DUP, SPEC and HGT
4. eliminate all redexes of TMOVE^{-1} and CLOST^{-1}

Algorithm 1 presents an efficient procedure for transforming a given DLS-tree into the (unique) equivalent fat tree.

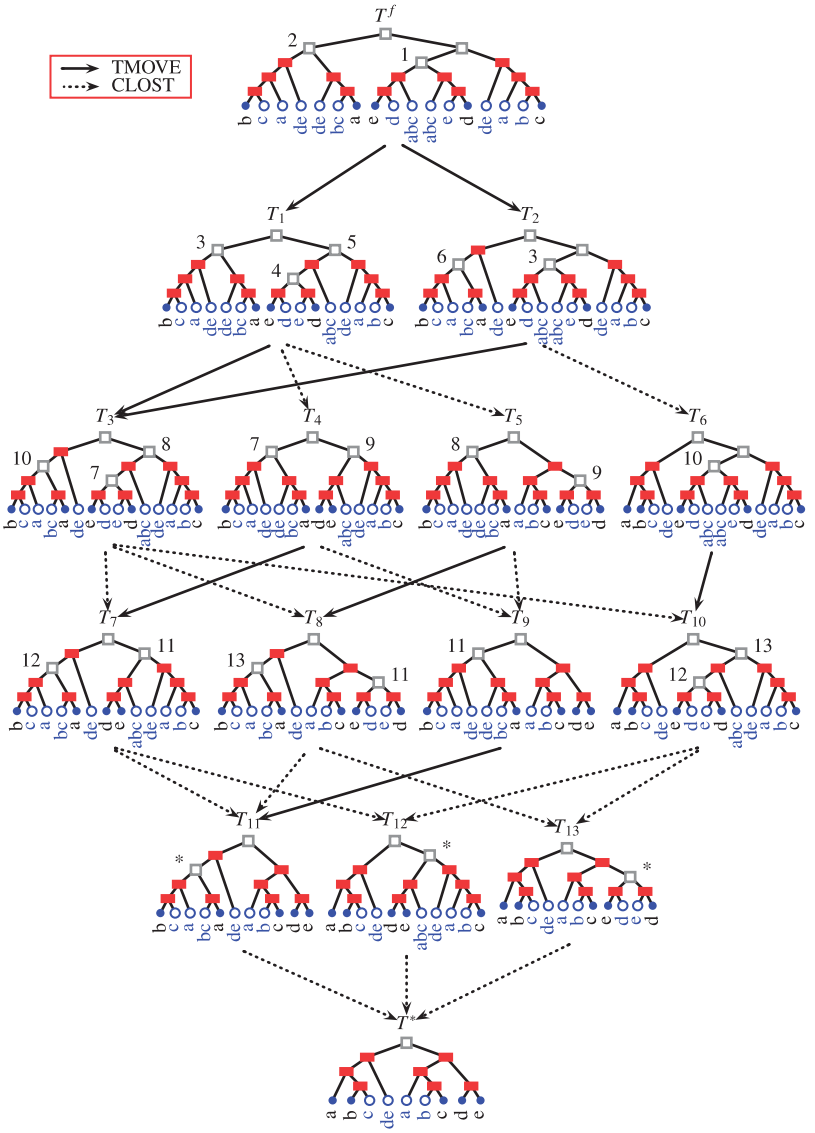


Fig. 13. Example hierarchy of semi-normal trees with all possible reductions.

Theorem 20. *The time and space complexity of Algorithm* 1 *is* $\mathcal{O}(|T|^2)$.

Proof. Assume that T has n nodes. Then T has $k = (n+1)/2$ leaves and at most k gene nodes. Also, the longest path in T has at most k nodes. Hence, the path in the longest chain tree in the final fat tree has at most k nodes. Having, this we conclude that we have at most $2 * k * k + n$ nodes in the fat tree. Each reversed reduction of type II increases the size of the tree, therefore, the time and space complexity of Algorithm 1 complexity is $\mathcal{O}(|T|^2)$. □

6.6. *Computing a tree in normal form*

Similarly to Algorithm 1, we can define an algorithm for computing the normal form of a given DLS-tree (by removing $^{-1}$ in line 4). It is clear that the time complexity of this transformation is $\mathcal{O}(|T|)$ and the space complexity is $\mathcal{O}(1)$ (in each step we decrease the size of T).

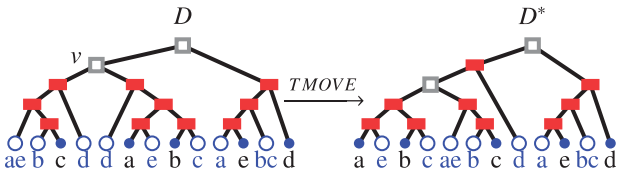


Fig. 14. D (semi-normal) and D^* (in normal form) have the same minimal duplication cost (v is a redex).

6.7. Hierarchy of semi-normal trees

Semi-normal trees are important representants of each class of equivalent DLS-trees. We consider a hierarchy of equivalent semi-normal trees and summarize its properties.

By the proof of Proposition 12 and further discussion, we can transform each semi-normal tree into the unique fat tree in two steps. In the first step, we apply all possible CLOST rules in the reverse direction. Then, we apply TMOVE rules in the reverse direction.

Analogously, we can transform each semi-normal tree into the unique tree in a normal form. First, we apply TMOVE rules, then CLOST in the direction $\rightarrow$.

An example of a hierarchy of semi-normal trees, with all possible reductions, is presented in Fig. 13. In our example, we have all possible 15 semi-normal DLS-trees. T^{f} is the fat tree. T^* is the tree in normal form. The labels of the internal nodes are not shown. They can be easily reconstructed from the labels of the leaves. Dotted and solid arrows denote CLOST and TMOVE reductions, respectively. The nodes marked by a number i (and $*$) are the redexes of the rules which produce T_i (and T^*, respectively). For instance, a redex of $T_5 \rightarrow T_9$ is a node in T_5 marked by 9.

It should be noted that, if trees are in the same row in Fig. 13, then they have the same number of speciations. It follows easily from the fact that the reductions of type II reduce the number of speciations by one.

6.8. Duplication cost and uniqueness

We can also prove that there exists more than one DLS-tree with the same minimal duplication cost. It is clear, if an application of SPEC is possible. For example, there exists a DLS-tree T (not semi-normal) which has the same duplication cost as T^* from Fig. 13 and such that T^* is obtained from T by one application of SPEC (to obtain T replace $de_{\bigcirc}$ in T^* by $(d_{\bigcirc}, e_{\bigcirc})_{\blacksquare}$). Even if we consider only semi-normal trees, the uniqueness of duplication cost is not satisfied. See Fig. 14 for details.

7. From gene and species trees to DLS-trees

In this section, for a given species tree $\mathcal{S}$ and a gene tree $\mathcal{G}$, we present the construction of a DLS-tree $\rho(\mathcal{G}, \mathcal{S})$ in normal form subject to the condition $\emptyset \neq L(\mathcal{G}) \subseteq L(\mathcal{S})$ (this condition is required for the trees in this section). We show how to compute the number of evolutionary events in such a tree. Also, we show a natural transformation from reconciled trees into normal form DLS-trees.

7.1. Normal from trees

Let $\rightsquigarrow$ denote a path existence relation in $\mathcal{S}$, i.e., $a \rightsquigarrow b$ if and only if there exists a path from a to b in $\mathcal{S}$. Let $\twoheadrightarrow$ denote a child relation, i.e., $a \twoheadrightarrow b$ if and only if b is a child of a. Reversed arrows are used to denote the reversed relations.

For a species tree $\mathcal{S}$ and a gene tree $\mathcal{G}$ such that $\emptyset \neq L(\mathcal{G}) \subseteq L(\mathcal{S})$, for each $g \in \mathcal{G}$, by $M(g)$ we denote the node $s \in \mathcal{S}$ such that

$$m_s^{\mathcal{S}} = \bigcap \{m_w^{\mathcal{S}} \mid m_g^{\mathcal{G}} \subseteq m_w^{\mathcal{S}}\}.$$

The obtained function $M : \mathcal{G} \rightarrow \mathcal{S}$ is called in the literature [6,15] *a least common ancestor mapping* or just *lca-mapping* (see Fig. 19).

The definition of $\rho(\mathcal{G},\mathcal{S})$ is by structural induction on the size of $\mathcal{G}$ and $\mathcal{S}$. Let $s = root(\mathcal{S})$ and $g = root(\mathcal{G})$. If $\mathcal{S}$ and $\mathcal{G}$ are leaves, then $\rho = a$, where a is the label of g. Otherwise, let p and q be the children of g, then

$$\rho(\mathcal{G},\mathcal{S}) = \begin{cases} (\rho(\mathcal{G}(p),\mathcal{S}), \rho(\mathcal{G}(q),\mathcal{S}))_{\square} & \text{if } M(g) = s = M(q)\ (R1), \\ (\rho(\mathcal{G}(p),\mathcal{S}(a)), \rho(\mathcal{G}(q),\mathcal{S}(b)))_{\blacksquare} & \text{if } M(p) \leftsquigarrow a \leftarrow s = M(g) \rightarrow b \rightsquigarrow M(q)\ (R2), \\ (\rho(\mathcal{G},\mathcal{S}(a)), (m_b^{\mathcal{S}})_{\bigcirc})_{\blacksquare} & \text{if } M(g) \leftsquigarrow a \leftarrow s \rightarrow b \neq a\ (R3). \end{cases} \tag{9}$$

Lemma 21. *For a species tree $\mathcal{S}$ and a gene tree $\mathcal{G}$ such that $\emptyset \neq L(\mathcal{G}) \subseteq L(\mathcal{S})$, $\rho(\mathcal{G},\mathcal{S})$ is a DLS-tree.*

Proof. The proof follows easily from the definition of the DLS-tree. □

Lemma 22. *Under the assumptions of Lemma* 21
(I) $gene(\rho(\mathcal{G},\mathcal{S})) = \mathcal{G}$,
(II) *labels of $\rho(\mathcal{G},\mathcal{S})$ are clusters in $\mathcal{S}$,*
(III) *if $L(\mathcal{G}) = L(\mathcal{S})$, then $spec(\rho(\mathcal{G},\mathcal{S})) = \mathcal{S}$.*

Proof. (I) It follows by induction on the size of $\mathcal{G}$ and $\mathcal{S}$. If $\mathcal{G} = \mathcal{S} = g$, then $gene(\rho(g,g)) = g$. Let $d > 1$. Assume that, for all $\mathcal{G}$ and $\mathcal{S}$ such that $|\mathcal{G}| + |\mathcal{S}| < d$ and $\emptyset \neq L(\mathcal{G}) \subseteq L(\mathcal{S})$, $gene(\rho(\mathcal{G},\mathcal{S})) = \mathcal{G}$. We proceed with $\mathcal{G}$ and $\mathcal{S}$ such that $|\mathcal{G}| + |\mathcal{S}| = d$. From the definition of *gene* and (9), we have
(R1) $gene((\rho(\mathcal{G}(p),\mathcal{S}), \rho(\mathcal{G}(q),\mathcal{S}))_{\square}) = (gene(\rho(\mathcal{G}(p),\mathcal{S})), gene(\rho(\mathcal{G}(q),\mathcal{S}))) = (\mathcal{G}(p), \mathcal{G}(q)) = \mathcal{G}$,
(R2) $gene((\rho(\mathcal{G}(p),\mathcal{S}(a)), \rho(\mathcal{G}(q),\mathcal{S}(b)))_{\blacksquare}) = (gene(\rho(\mathcal{G}(p),\mathcal{S}(a))), gene(\rho(\mathcal{G}(q),\mathcal{S}(b)))) = (\mathcal{G}(p),\mathcal{G}(q)) = \mathcal{G}$,
(R3) $gene((\rho(\mathcal{G},\mathcal{S}(a)), (m_b^{\mathcal{S}})_{\bigcirc})_{\blacksquare}) = gene(\rho(\mathcal{G},\mathcal{S}(a))) = \mathcal{G}$.
(II) It follows immediately from the fact that $\Lambda(\rho(\mathcal{G},\mathcal{S})) = L(\mathcal{S})$.
(III) Note that $\mathfrak{lost}^{\rho(\mathcal{G},\mathcal{S})} = L(\mathcal{S}) \setminus L(\mathcal{G})$. In this case $\rho(\mathcal{G},\mathcal{S})$ is a complete DLS-tree. By (II) and Lemma 3, we conclude that $spec(\rho(\mathcal{G},\mathcal{S})) = \mathcal{S}$.
This completes the proof. □

One of the most important properties of ρ is stated below:

Lemma 23. *Let $\mathcal{G}$ be a gene tree and $\mathcal{S}$ be a species tree such that $\emptyset \neq L(\mathcal{G}) \subseteq L(\mathcal{S})$. Then $\rho(\mathcal{G},\mathcal{S})$ is in normal form.*

Proof. We show that there are no redexes of the DLS rules in $R = \rho(\mathcal{G},\mathcal{S})$.
(SPEC) Observe that $\rho(\mathcal{G}',\mathcal{S}')$ can never be a lost leaf. Thus, there is no DLS pattern $(A_{\bigcirc}, B_{\bigcirc})_*$ in $\rho(\mathcal{G},\mathcal{S})$. We conclude that SPEC cannot be applicable.
(DUP) Similarly, we obtain that $(A_{\bigcirc}, T)_{\square}$ is not present in $\rho(\mathcal{G},\mathcal{S})$.
(TMOVE) Let us assume that $((S_1,\mathcal{C}_{\bigcirc})_{\blacksquare}, (S_2,\mathcal{C}_{\bigcirc})_{\blacksquare})_{\square}$ (i.e., the premise of this rule) is present in $\rho(\mathcal{G},\mathcal{S})$. Without loss of generality, we may assume that the root of $\rho(\mathcal{G},\mathcal{S})$ is the redex of the rule. The pattern can be obtained only from the first case, that is, from (R1) of (9). So we have

$$M(root(\mathcal{G})) = root(\mathcal{S}) = M(q), \tag{10}$$

where q is a child of the root in $\mathcal{G}$. By (R1), for $i = 1$ or 2, we see that

$$\rho(\mathcal{G}(q),\mathcal{S}) = (S_i,\mathcal{C}_{\bigcirc})_{\blacksquare}.$$

The pattern could be obtained only from the case (R3). However, this requires $M(root(\mathcal{G}(q))) = M(q) \neq root(\mathcal{S})$ which contradicts (10).
(CLOST) The proof is similar to (TMOVE) case. □

Now, we conclude that if T is a complete DLS-tree, then $T^* = \rho(gene(T), spec(T))$, where T^* is the normal form of T.

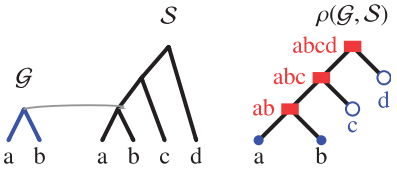


Fig. 15. Example of $\mathcal{G}$ and $\mathcal{S}$ with $\mathbf{loss}_0 = 2$.

7.2. Counting evolutionary events

Having formula (9), we can compute the number of evolutionary events in a tree in normal form.

Lemma 24. *Let $\mathcal{G}$ be a gene tree and $\mathcal{S}$ be a species tree such that $\emptyset \neq L(\mathcal{G}) \subseteq L(\mathcal{S})$. Then the number of duplications in $\rho(\mathcal{G},\mathcal{S})$ equals*

$$\mathbf{dup}(\mathcal{G},\mathcal{S}) = |\{g \mid M(g) = M(p) \text{ where } p \text{ is a child of } g \text{ in } \mathcal{G}\}|.$$

Proof. Follows immediately from (R1) in (9). □

Lemma 25. *Let $\mathcal{G}$ be a gene tree and $\mathcal{S}$ be a species tree such that $\emptyset \neq L(\mathcal{G}) \subseteq L(\mathcal{S})$. For each node g, we define a non-negative integer $\mathbf{loss}_g$ as follows: we set $\mathbf{loss}_g = 0$ if g is leaf in $\mathcal{G}$, if g is an internal node in $\mathcal{G}$ then let p and q denote the two children of g. We define*

$$\mathbf{loss}_g = \begin{cases} d(M(g),M(p)) + 1 & \text{if } M(p) \neq M(g) = M(q), \\ d(M(g),M(p)) + d(M(g),M(q)) & \text{otherwise}, \end{cases}$$

where $d(s,s') = |\{t \mid m^{\mathcal{S}}_{s'} \subset m^{\mathcal{S}}_t \subset m^{\mathcal{S}}_s\}|$.

Let $\mathbf{loss}_0 = |\{s \mid m^{\mathcal{S}}_{M(root(\mathcal{G}))} \subset m^{\mathcal{S}}_s\}|$. Then the number of gene losses in $\rho(\mathcal{G},\mathcal{S})$ is given by

$$\mathbf{loss}_0 + \sum_{g\in\mathcal{G}} \mathbf{loss}_g. \tag{11}$$

Proof. The first element of the final sum (11) is needed, if $M(root(\mathcal{G})) \neq root(\mathcal{S})$. It should be clear from (R3) case of (9) that we obtain $\mathbf{loss}_0$ gene losses in the first applications of ρ until $S' = \mathcal{S}(M(root(\mathcal{G})))$ is reached as the second argument of ρ:

$$\rho(\mathcal{G},\mathcal{S}) = (A^1_{\bigcirc}, \ldots (A^{\mathbf{loss}_0}_{\bigcirc}, \rho(\mathcal{G},S'))_{\blacksquare} \ldots)_{\blacksquare}.$$

Fig. 15 illustrates this property. Note that if $M(root(\mathcal{G})) = root(\mathcal{S})$, then $\mathbf{loss}_0 = 0$.

Let $g \in \mathcal{G}$. It follows from (9) that $\mathcal{S}(M(g))$ is the smallest subtree T of $\mathcal{S}$ such that $\rho(\mathcal{G},\mathcal{S}) = (\ldots \rho(\mathcal{G}(g),T)\ldots)$. For a node g in $\mathcal{G}$, let $\underline{g}$ be the root of $T_g = \rho(\mathcal{G}(g),\mathcal{S}(M(g)))$ in $\rho(\mathcal{G},\mathcal{S})$. Let us assume that g is internal node in $\mathcal{G}$. Let p and q be the two children of g.

We consider paths $\underline{g}p_1 \ldots p_k\underline{p}$ and $\underline{g}q_1 \ldots q_m\underline{q}$ in $\rho(\mathcal{G},\mathcal{S})$. We show that $\mathbf{loss}_g = k + m$.

Without loss of generality we assume that g is the root of $\mathcal{G}$ and $M(g) = root(\mathcal{S})$.

Case A: $M(p) = M(g) = M(q)$. (R1) is applied. So $\rho(\mathcal{G},\mathcal{S}) = (\rho(\mathcal{G}(p),\mathcal{S}),\rho(\mathcal{G}(q),\mathcal{S}))_{\square}$, and hence $T_p = \rho(\mathcal{G}(p),\mathcal{S})$. Similarly, $T_q = \rho(\mathcal{G}(q),\mathcal{S})$. We obtain: $k = m = 0$. See Fig. 16 for details.

Case B: $M(p) \neq M(g) = M(q)$. Again, (R1) is applied. We have $T_q = \rho(\mathcal{G}(q),\mathcal{S})$, and hence $m = 0$. For the second child, we apply a sequence of k (R3) cases. In this way, we reach T_p in $\rho(\mathcal{G},\mathcal{S})$. Fig. 16 presents the details. Note that

$$m^{\mathcal{S}}_{M(g)} = \Lambda_{\underline{g}} = \Lambda_{p_1}. \tag{12}$$

It should be clear that $\Lambda_{p_2}, \Lambda_{p_3}, \ldots, \Lambda_{p_k}$ is a sequence of all intermediate labels which occur between $M(g)$ and $M(p)$ in $\mathcal{S}$. So $d(M(g),M(p)) = k - 1$, and hence the number of gene losses equals $d(M(g),M(p)) + 1$.

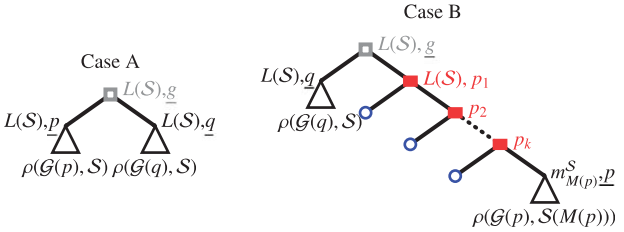


Fig. 16. Cases A and B.

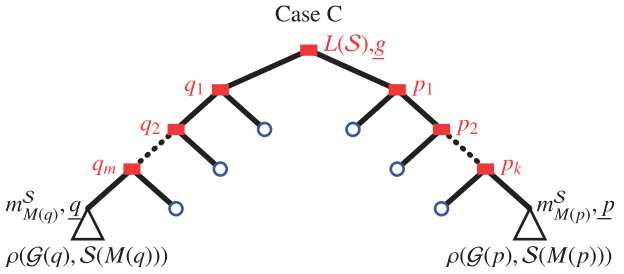


Fig. 17. Case C.

Case C: $M(p) \neq M(g) \neq M(q)$. First, (R2) is applied. So the labels of the children of $\vec{g}$ in T_g (i.e., Λ_{p_1} and Λ_{q_1}) do not equal $m^{\mathcal{S}}_{M(g)}$. Thus, this situation is different, that is, (12) does not hold. We apply a sequence of k (R3) cases to obtain T_p. In this case, $\Lambda_{p_1}, \Lambda_{p_2}, \ldots, m^{\mathcal{S}}_{p_k}$ is a sequence of all intermediate labels which occur between $M(g)$ and $M(p)$ in $\mathcal{S}$. So $d(M(g), M(p)) = k$ and, similarly, $d(M(g), M(q)) = m$.

Fig. 17 presents the details. □

7.3. Embeddings and normal form trees

It should be clear that we can embed a DLS-tree D into a species tree $\mathcal{S}$ if the labels of T are clusters in $\mathcal{S}$. For a gene tree $\mathcal{G}$ and a species tree $\mathcal{S}$, consider the set $\mathcal{P}(\mathcal{G}, \mathcal{S})$ of all DLS-trees in normal form such that if $T \in \mathcal{P}(\mathcal{G}, \mathcal{S})$, then $gene(T) = \mathcal{G}$ and $\mathfrak{Labels}(T) \subseteq \mathfrak{M}^{\mathcal{S}}$.

The following lemma states one of the crucial properties of $\mathcal{P}(\mathcal{G}, \mathcal{S})$.

Lemma 26. *Let $\mathcal{G}$ be a gene tree and $\mathcal{S}$ be a species tree such that $\emptyset \neq L(\mathcal{G}) \subseteq L(\mathcal{S})$. Then*

$$\mathcal{P}(\mathcal{G}, \mathcal{S}) = \{T(s) \mid T = \rho(\mathcal{G}, \mathcal{S}),\ s \in V^T_{\square},\ L(\mathcal{G}) \subseteq \Lambda^T_s \subseteq L(\mathcal{S})\}. \tag{13}$$

Proof. ($\supseteq$) Clear. ($\subseteq$) Assume that $\mathcal{P}(\mathcal{G}, \mathcal{S})$ contains T^*_1 and T^*_2 such that $\Lambda(T^*_1) = \Lambda(T^*_2) = L(\mathcal{S})$. There exist fat trees F_1 and F_2 equivalent to T^*_1 and T^*_2, respectively. By Proposition 10, $F_1 = F_2$. Thus, T^*_1 and T^*_2 are equivalent and by uniqueness of the normal form $T^*_1 = T^*_2$. We conclude that there is exactly one DLS-tree T^* in normal form in $\mathcal{P}(\mathcal{G}, \mathcal{S})$ satisfying $L(\mathcal{S}) = \Lambda(T^*)$. It is defined by $\rho(\mathcal{G}, \mathcal{S})$.

Assume that $\mathcal{P}(\mathcal{G}, \mathcal{S})$ contains T such that $\Lambda(T) \subset L(\mathcal{S})$. It is obvious that $L(\mathcal{G}) \subseteq \Lambda(T)$. Let $T' = (\ldots((T, A^1_{\bigcirc})_{\square}, A^2_{\bigcirc})_{\square} \ldots, A^k_{\bigcirc})_{\square}$ such that $\Lambda(T') = L(\mathcal{S})$ (note that T' is reconstructed uniquely). This yields $T' = \rho(\mathcal{G}, \mathcal{S})$. □

In general, it should be clear that $\mathcal{P}(\mathcal{G}, \mathcal{S})$ contains $\mathbf{loss}_0 + 1$ elements. See Fig. 18 for illustration.

7.4. Reconciled trees

Now, we present a definition of the reconciled tree taken from [2]. We know that this definition is equivalent to the definition given by [15]. Let $s = root(\mathcal{S})$ and $g = root(\mathcal{G})$. *The reconciled tree* $R(\mathcal{G}, \mathcal{S})$ of $\mathcal{G}$ with respect to $\mathcal{S}$ is the

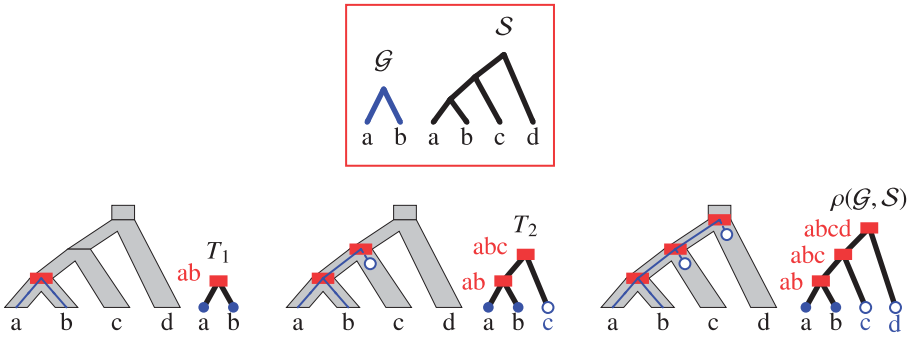


Fig. 18. Elements of $\mathcal{P}(\mathcal{G},\mathcal{S})$ and corresponding embeddings.

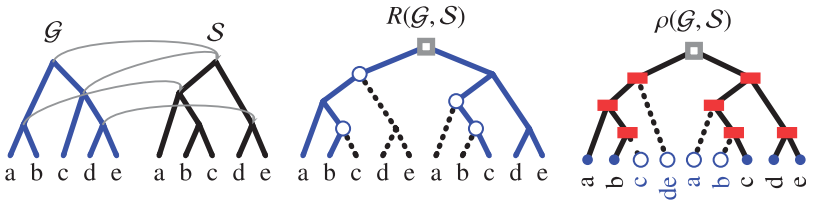


Fig. 19. Example of a mapping M, a reconciled tree $R(\mathcal{G},\mathcal{S})$ and a DLS-tree $\rho(\mathcal{G},\mathcal{S})$.

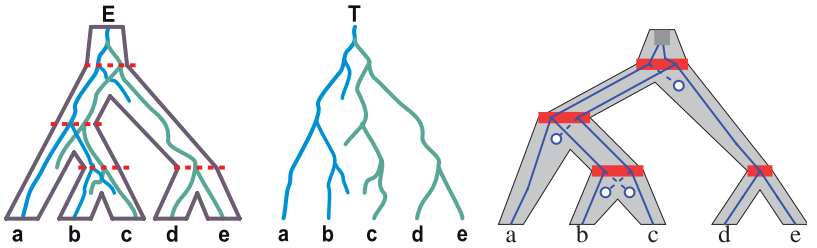


Fig. 20. The evolution of species and genes (cont. example from Fig. 19).

tree $\mathcal{G}$, if $\mathcal{G}$ and $\mathcal{S}$ are leaves. Otherwise, let p and q be the children of g, then

$$R(\mathcal{G},\mathcal{S}) = \begin{cases} (R(\mathcal{G}(p),\mathcal{S}),\, R(\mathcal{G}(q),\mathcal{S})) & \text{if } M(g) = s = M(q)\ (RT1), \\ (R(\mathcal{G}(p),\mathcal{S}(a)),\, R(\mathcal{G}(q),\mathcal{S}(b))) & \text{if } M(p) \leftsquigarrow a \leftarrow s = M(g) \rightarrow b \rightsquigarrow M(q)\ (RT2), \\ (R(\mathcal{G},\mathcal{S}(a)),\, S(b)) & \text{if } M(g) \leftsquigarrow a \leftarrow s \rightarrow b \neq a\ (RT3). \end{cases} \tag{14}$$

Theorem 27. *Let $\mathcal{G}$ be a gene tree and $\mathcal{S}$ be a species tree such that $\emptyset \neq L(\mathcal{G}) \subseteq L(\mathcal{S})$. Let θ be a transformation which takes a DLS-tree and returns a tree with leaves labelled by species*:

$$\theta(T) = \begin{cases} a & \text{if } T = a \text{ and } a \in \mathcal{I}, \\ \mathcal{S}(v) & \text{if } T = A_{\bigcirc} \text{ and } v \text{ in } \mathcal{S} \text{ such that } m_v^{\mathcal{S}} = A, \\ (T_1, T_2)_* & \text{if } T = (T_1, T_2)_*, \text{ where } * \in \{\blacksquare, \square\}. \end{cases} \tag{15}$$

Then $\theta(\rho(\mathcal{G},\mathcal{S})) = R(\mathcal{G},\mathcal{S})$.

Proof. It follows immediately from the definition of the reconciled tree and the definition of ρ. □

Thus, θ is a natural transformation between reconciled trees and DLS-trees in normal form. We claim that the formula for computing the mutation cost is the same for the reconciled tree [11] and the tree in normal form.

Fig. 19 presents an example of a lca-mapping M, for all internal nodes of $\mathcal{G}$. Also, it presents a reconciled tree $R(\mathcal{G}, \mathcal{S})$ and a DLS-tree $\rho(\mathcal{G}, \mathcal{S})$. Note that $\rho(\mathcal{G}, \mathcal{S})$ equals the tree T^* shown in Fig. 13. It is easy to notice that $\theta(\rho(\mathcal{G}, \mathcal{S})) = R(\mathcal{G}, \mathcal{S})$. For a more readable presentation, all the lost gene lineages are shown with dotted lines. The solid lines in $R(\mathcal{G}, \mathcal{S})$ and $\rho(\mathcal{G}, \mathcal{S})$ represent embedded gene trees.

Fig. 20 is a continuation of the example presented in Fig. 19. Tree E is the evolutionary interpretation of our DLS-tree. T presents an extraction of the gene lineages from E. Note that the tree T is equal topologically to the DLS-tree $\rho(\mathcal{G}, \mathcal{S})$. Also the embedding is shown.

8. Final remarks

All algorithms presented here, the system of rules, mappings, reductions, reconciled trees and others for DL-models are included in this software. It will be soon available.

Acknowledgments

We thank Bartosz Wilczynski and Szymon Nowakowski for comments. This research was supported by KBN Grant 4 T11F 020 25.

References

[1] L. Arvestad, A.-C. Berglund, J. Lagergren, B. Sennblad, Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution, in: RECOMB 2004, 2004.
[2] P. Bonizzoni, G. Vedova, R. Dondi, Reconciling gene trees to a species tree, Algorithms and Complexity, Proc. Fifth Italian Conf. (CIAC 2003) Vol. 2653, 2003, pp. 120–131.
[3] M.A. Charleston, Jungles: a new solution to the host/parasite phylogeny reconciliation problem, Math. Biosci. 149 (1998) 191–223.
[4] O. Eulenstein, B. Mirkin, M. Vingron, Duplication-based measures of difference between gene and species trees, J. Comput. Biol. 5 (1) (1998) 135–148.
[5] O. Eulenstein, M. Vingron, On the equivalence of two tree mapping measures, DAMATH: Discrete Appl. Math. Combin. Oper. Res. Comput. Sci. 88 (1995).
[6] M. Goodman, J. Czelusniak, G.W. Moore, A.E. Romero-Harrera, G. Matsuda, Fitting the gene lineage into its species lineage. A parsimony strategy illustrated by cladograms constructed from globin sequences, Syst. Zool. 28 (1979) 132–163.
[7] P. Górecki, Reconciliation problems for duplication, loss and horizontal gene transfer, in: RECOMB 2004, San Diego, 2004, pp. 316–325.
[8] P. Górecki, J. Tiuryn, On the structure of reconciliations, Lecture Notes in Computer Science, Vol. 3388, Springer, Berlin, 2005, pp. 42–54.
[9] R. Guigo, I. Muchnik, T. Smith, Reconstruction of ancient molecular phylogeny, Molecular Phys. Evol. 6 (1996) 189–213.
[10] M. Hallett, J. Lagergren, New algorithms for the duplication-loss model, in: Proc. RECOMB 2000, ACM Press, Tokyo, 2000, pp. 138–146.
[11] B. Ma, M. Li, L. Zhang, On reconstructing species trees from gene trees in term of duplications and losses, in: RECOMB 1998, 1998, pp. 182–191.
[12] B. Ma, M. Li, L. Zhang, From gene trees to species trees, SIAM J. Comput. 30 (2000) 792–852.
[13] B. Mirkin, I. Muchnik, T.F. Smith, A biologically consistent model for comparing molecular phylogenies, J. Comput. Biol. 2 (4) (1995) 493–507.
[14] R.D.M. Page, Component analysis: a valiant failure? Cladistics 6 (1990) 119–136.
[15] R.D.M. Page, Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas, Syst. Biol. 43 (1994) 58–77.
[16] R.D.M. Page, M.A. Charleston, Reconciled trees and incongruent gene and species trees, Mathematical Hierarchies and Biology, DIMACS Series in Mathematics and Theoretical Computers Science 37 (1998) 57–70.
[17] L. Zhang, On a Mirkin-Muchnik-Smith conjecture for comparing molecular phylogenies, J. Comput. Biol. 4 (2) (1997) 177–188.