Improved Approximation Algorithms for Reconstructing the History of Tandem Repeats

Zhi-Zhong Chen* Lusheng Wang†

Abstract

Some genetic diseases in human beings are dominated by short sequences repeated consecutively called tandem repeats. Once a region containing tandem repeats is found, it is of great interest to study the history of creating the repeats. The computational problem of reconstructing the duplication history of tandem repeats has been studied extensively in the literature. Almost all previous studies focused on the simplest case where the size of each duplication block is 1. Only recently we succeeded in giving the first polynomial-time approximation algorithm with a guaranteed ratio for a more general case where the size of each duplication block is at most 2; the algorithm achieves a ratio of 6 and runs in $O(n^{11})$ time. In this paper, we present two new polynomial-time approximation algorithms for this more general case. One of them achieves a ratio of 5 and runs in $O(n^9)$ time while the other achieves a ratio of $2.5 + \epsilon$ for any constant $\epsilon > 0$ but runs slower.

Keywords: Computational biology, approximation algorithms.

1 Introduction

The genomes of many species are dominated by short segments repeated consecutively. It is estimated that over 10% of the human genome consists of repeated segments. About 10-25% of all known proteins have some form of repeated structures. Computing the duplication history of a tandem repeated region is a very important problem in computational biology [3, 5, 9]. A number of papers related to this problem have been published [1, 5, 6, 7, 8, 9, 12].

1.1 The Duplication Model

The model for the duplication history of tandem repeated segments was proposed by Fitch in 1977 [3] and re-proposed by Tang et al. [9] and Jaitly et al. [5]. The model captures both the evolutionary history and the observed order of segments on a chromosome. Let $S = s_1 s_2 \dots s_n$ be an observed string consisting of n segments of the same length m. Let $t_i t_{i+1} \dots t_{i+k-1}$ be k consecutive segments in an ancestor string of S in the evolutionary history. A *duplication event* generates $2k$ consecutive segments $l_c(t_i) l_c(t_{i+1}) \dots l_c(t_{i+k-1}) r_c(t_i) r_c(t_{i+1}) \dots r_c(t_{i+k-1})$ by (approximately) copying the k segments $t_i t_{i+1} \dots t_{i+k-1}$ twice, where both $l_c(t_{i+j})$ and $r_c(t_{i+j})$ are approximate copies

*Department of Mathematical Sciences, Tokyo Denki University, Hatoyama, Saitama 350-0394, Japan. Email: chen@r.dendai.ac.jp.

†Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong.

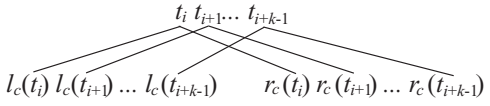

Figure 1: A k-duplication, where both $l_c(t_{i+j})$ and $r_c(t_{i+j})$ are approximate copies of t_{i+j}.

of t_{i+j} (see Figure 1). Assume that the n segments s_1, s_2, ... s_n were formed from a locus by tandem duplications. Then, the locus had grown from a single copy through a series of duplications. A duplication replaces a stretch of DNA consisting of several segments with two (approximately) identical and adjacent copies of itself. If the stretch contains k segments, the duplication is called a *k-duplication*.

Recall that in a rooted binary tree T, each vertex may have at most one parent and either zero or two children. There is only one vertex, called the *root* of T, that has no parent. Those vertices with no children are called the *leaves* of T while the others are called the *nonleaves* of T. The two children of each nonleaf v in T are distinguished as the *left child* and the *right child* of v in T, respectively. If a vertex v_1 appears on the path from the root to another vertex v_2 in T, then v_1 is an *ancestor* of v_2 in T while v_2 is a *descendant* of v_1 in T. For convenience, we view each vertex as a descendant and ancestor of itself. Two vertices are *incomparable* in T if neither of them is an ancestor or descendant of the other in T. Moreover, the edge between a nonleaf u and a child v of u is denoted by (u, v).

Let $S = \langle s_1, s_2, \ldots, s_n \rangle$ be a list of strings of the same length m. A *duplication model* for S is a rooted binary tree M embedded in the plane and armed with a partition $\mathcal{B}$ of the set of nonleaves of M into disjoint lists such that the following conditions are satisfied (cf. Figure 2):

1. Each vertex of M is a point in the plane while each edge of M is a straight-line segment in the plane.

2. The root of M appears at the top while the leaves of M appear at the bottom (at the same height).

3. The left child of each nonleaf v in M appears below and on the left of v, while the right child appears below and on the right of v.

4. Each vertex of M is labeled by a string of length m. In particular, the leaves of M are labeled, from left to right, by s_1, s_2, ..., s_n, respectively.

5. For every list $\langle v_1, \ldots, v_k \rangle \in \mathcal{B}$ with $k \geq 2$, the following hold:

   (a) v_1, v_2, ..., v_k are pairwise incomparable in M.

   (b) If we draw a line segment ℓ from v_1 to v_k in the plane, then ℓ is horizontal, v_1, v_2, ..., v_k appear on ℓ from left to right in this order, and no other vertices of M appear on ℓ.

   (c) For every two integers i and j with $1 \leq i < j \leq k$, the edge from v_i to its right child crosses the edge from v_j to its left child in the plane.

6. Two edges of M cross each other only if the crossing is specified in Condition 5c.

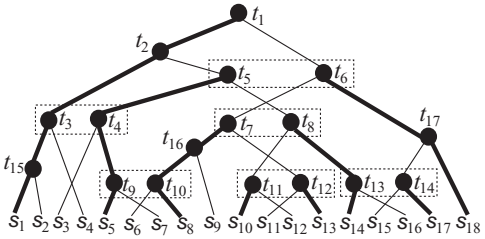

Figure 2: A duplication model M.

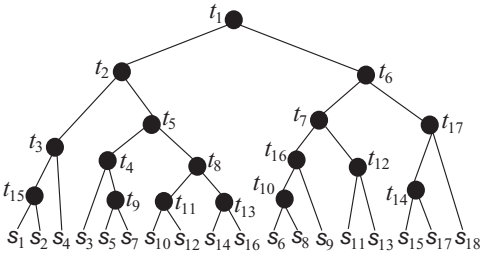

Figure 3: The associated phylogeny of M in Figure 2.

We call each list in $\mathcal{B}$ a *block* of M. The *size* of a block B is the number of vertices in B and is denoted by $|B|$. For each integer $k \geq 1$, a *k-block* is a block of size k. When we depict M via a figure, we show each block B of M with $|B| \geq 2$ by drawing a rectangle to enclose the vertices of B. Hence, if a vertex is not enclosed by a rectangle in the figure, then it alone forms a block of M.

Each edge of M carries a *cost* which is simply the hamming distance between the two segments associated with the two endpoints of the edge. The *cost* of M, denoted by $c(M)$, is the total cost of edges of M. We remark that all our results apply to other distance measures satisfying the triangle inequality.

Since M is a tree, we can re-embed it in the plane without edge crossings and without violating the first three conditions above. The new embedded tree T_M is called the *associated phylogeny* for M (see Figure 3 for an example). Clearly, T_M is unique, its vertices one-to-one correspond to those of M, and its cost equals that of M.

Obviously, if every block in M is of size 1, then T_M and M are identical. However, if one or more blocks in M are of size larger than 1, then the left-to-right order of the labels of the leaves of T_M is not s_1, s_2, ..., s_n.

1.2 The Problem, Previous Results, and Our Results

Now, we are ready to state the problem considered in the paper:

Duplication History Reconstruction (DHR):

- **Input:** A list $S = \langle s_1, s_2, \ldots, s_n \rangle$ of strings of the same length m.
- **Output:** A duplication model for S with the smallest cost.

For each integer $k \geq 1$, let k-DHR denote the special case of DHR where the size of each duplication block is at most k. 1-DHR and its variants have been studied extensively in the literature [1, 2, 5, 9, 10, 11]. In particular, Jaitly *et al.* proved the NP-hardness of 1-DHR and

designed a PTAS for it. At present, the best PTAS for 1-DHR was given in [2]. Benson and Dong [1] and Tang *et al.* [9] designed exact algorithms for 1-DHR that run in time exponential in m but polynomial in n.

Unlike 1-DHR, k-DHR with $k \geq 2$ is much harder to approximate. Indeed, as observed in [4], we can design a trivial 2-approximation algorithm for 1-DHR as follows: Given $\langle s_1, s_2, \ldots, s_n \rangle$, first construct a rooted path P with n vertices, next label the vertices of P with s_1, s_2, $\ldots$, s_n in this order from bottom to top, and finally add a new child with label s_i to the vertex of P with label s_i for every $i \in \{2, 3, \ldots, n\}$. However, this simple algorithm does not work for 2-DHR. To see this, consider the case where n is even, $s_1 = s_3 = \cdots = s_{n-1} = 0$, and $s_2 = s_4 = \cdots = s_n = 1$. In this case, the cost of the optimal duplication model for $\langle s_1, s_2, \ldots, s_n \rangle$ is 1 while the cost of the duplication model constructed by the simple algorithm is $n - 1$. Not only this simple algorithm but also the other known algorithms for 1-DHR do not work for 2-DHR. In fact, it had been elusive for a while to design a polynomial-time approximation algorithm for 2-DHR that achieves a guaranteed ratio. Only very recently, Chen *et al.* [2] succeeded in designing the first polynomial-time approximation algorithm with a guaranteed ratio for 2-DHR; it runs in $O(n^{11} + n^2 m)$ time and achieves a ratio of 6 [1]. The main ideas behind the algorithm can be summarized as follows:

1. Each duplication model M for $S = \langle s_1, s_2, \ldots, s_n \rangle$ can be decomposed into smaller components which can be organized into a tree called the *component tree* of M.

2. The component tree of M can be transformed into a new model M' for S in $O(nm)$ time with $c(M') \leq 3 \cdot c(M)$.

3. We can find the best component tree of a *lifted model* for S in $O(n^{11} + n^2 m)$ time via dynamic programming, where a lifted model for S is a model whose vertices are assigned strings in S.

In this paper, we design two better approximation algorithms for 2-DHR. One of our algorithms runs in $O(n^9 + n^2 m)$ time and achieves a ratio of 5. The other runs in polynomial time and achieves a ratio of $2.5 + \epsilon$ for any constant $\epsilon > 0$. Our algorithms are of purely theoretical interest because of their high complexity. Besides the above old ideas used in [2], our algorithms have two new important ideas. The main new idea is to show the existence of a 0.75-*separator* in a duplication model M, which is a set $\mathcal{P}$ of edge-disjoint paths such that the total weight of edges in paths in $\mathcal{P}$ is at most $0.75 \cdot c(M)$ and the paths in $\mathcal{P}$ can be used to decompose M into smaller components that can be organized into a tree (still called the *component tree* of M as before). The other new idea is to look at *r-lifted models* which are less restricted than lifted models. Basically, r-lifted models are similar to r-lifted phylogenies in [11]. We believe that the two new ideas will eventually lead to a PTAS for 2-DHR.

Throughout the remainder of this paper, a duplication model means one in which each block is of size at most 2.

1.3 Organization of the Paper

The rest of this paper is organized as follows. In Section 2, we give several definitions and prove several lemmas for duplication models. In Section 3, we generalize duplication models to multiroot

[1] In the conference version of [2], the authors wrongly claimed a ratio-2 polynomial-time approximation algorithm.

models and define splitting vertices for them. In Section 4, we show how to use splitting vertices to split multiroot models to smaller multiroot models; we also define separators and show their relations to splitting vertices. In Section 5, we prove our main lemma that every multiroot model has a 0.75-separator. In Section 6, we use a 0.75-separator Γ of a multiroot model M to split M into smaller multiroot models and organize them into a tree $\mathcal{D}(M, \Gamma)$ called the *component tree* of M *associated with* Γ. In Section 7, we show how to construct a new model M' from $\mathcal{D}(M, \Gamma)$ with $c(M') \leq 2.5 \cdot c(M)$. In Section 8, we define abstract component trees for a list $\mathcal{L}$ of strings in such a way that the component tree of each multiroot model for $\mathcal{L}$ is always an abstract component tree for $\mathcal{L}$. The crucial point is that the best abstract component tree for $\mathcal{L}$ can be computed in polynomial time via dynamic programming (cf. Section 9). The best abstract component tree for $\mathcal{L}$ can then be used to construct a real multiroot model for $\mathcal{L}$ whose cost is close to optimal. The ratio-5 and the ratio-$(2.5 + \epsilon)$ approximation algorithms are summarized in Sections 10 and 11, respectively. Section 12 concludes the paper with several remarks.

Throughout the remainder of this paper, let $S = \langle s_1, s_2, \ldots, s_n \rangle$ be a list of strings of the same length m. Our goal is to show how to construct a good duplication model for this list.

2 Preliminaries

Fix two integers i and j with $1 \leq i \leq j \leq n$. Let M be a duplication model for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$. Note that two strings in $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ may be identical. So, for clarity, we use ℓ_h to denote the $(h - i + 1)$st leftmost leaf in M for every integer $h \in \{i, i + 1, \ldots, j\}$. Obviously, the label of ℓ_h in M is s_h.

An edge in M is *planar* if it is not crossed by another edge in M. A path in M is *planar* if it traverses planar edges only.

Lemma 2.1 *For each vertex u of M, there is a planar path P in M from u down to a leaf.*

Proof. Suppose that we start at a vertex u and go down to a leaf based on the following rule: Assume that we are now at a nonleaf v. If the edge from v to its left child v_1 is planar, then we next move to v_1; otherwise, we move to its right child v_2. Since M is a duplication model, at least one of (v, v_1) and (v, v_2) is planar. So, we can always go down to a leaf by traversing planar edges only. □

Lemma 2.2 *Consider an arbitrary nonleaf u of M. Let v_1 and v_2 be the left and the right child of v in M, respectively. Then, all descendants of v_2 in M appear on the right of each planar path from v_1 to a leaf in M. Similarly, all descendants of v_1 in M appear on the left of each planar path from v_2 to a leaf in M.*

Proof. Since v_1 appears on the left of v while v_2 appears on the right of v in M, the lemma follows from the planarity of planar paths immediately. □

A *left* (respectively, *right*) *edge* in M is an edge between a nonleaf and its left (respectively, right) child in M. A *left* (respectively, *right*) *path* in M is a path in M that traverses only left (respectively, right) edges.

Lemma 2.3 *If two edges cross each other in M, then one of them is a left edge and the other is a right edge.*

PROOF. Suppose that two edges (u_1, u_2) and (v_1, v_2) cross each other in M. Then, u_1 and v_1 together form a 2-block B. Moreover, either (u_1, u_2) is a right edge and (v_1, v_2) is a left edge, or (u_1, u_2) is a left edge and (v_1, v_2) is a right edge. Thus, the lemma holds. □

Consider a 2-block $B = \langle u_1, u_2 \rangle$ in M. We call u_1 the *left vertex* in B and call u_2 the *right vertex* in B. Note that the edge between u_1 and its left child in M is planar, while the edge between u_1 and its right child in M is not. Similarly, the edge between u_2 and its right child in M is planar, while the edge between u_2 and its left child in M is not.

For each nonleaf u of M, we use $I_M(u)$ (respectively, $J_M(u)$) to denote the smallest (respectively, largest) integer h such that ℓ_h is a leaf descendant of u in M. The following lemma follows from Lemmas 2.1 and 2.2 immediately:

Lemma 2.4 *Let u be a nonleaf of M. Then, the path from u to $\ell_{I_M(u)}$ in M is a left path and the path from u to $\ell_{J_M(u)}$ in M is a right path. Moreover, the two paths do not cross each other.*

Two nonleaves u and v are *unrelated* in M if $J_M(u) < I_M(v)$ or $J_M(v) < I_M(u)$. A nonleaf u *crosses* another nonleaf v in M if $I_M(u) < I_M(v) < J_M(u) < J_M(v)$. Note that if u crosses v in M, then v does not cross u in M. A nonleaf u *covers* another nonleaf v in M if $I_M(u) \leq I_M(v) < J_M(v) \leq J_M(u)$. Note that if u is an ancestor of v in M, then u covers v in M. However, a nonleaf may cover another nonleaf in M even if they are incomparable in M. Two nonleaves of M are *unnested* if neither of them covers the other in M. A nonleaf u is *on the left* of another vertex v in M if u and v are unnested and $I_M(u) < I_M(v)$ (or equivalently, $J_M(u) < J_M(v)$).

The next lemma helps the reader understand how a nonleaf covers another in M or how two nonleaves become unrelated in M.

Lemma 2.5 *Let u and v be two nonleaves in M. Let P_u (respectively, Q_u) be the path from u to $\ell_{I_M(u)}$ (respectively, $\ell_{J_M(u)}$). Let P_v (respectively, Q_v) be the path from v to $\ell_{I_M(v)}$ (respectively, $\ell_{J_M(v)}$). Suppose that u covers v in M or u and v are unrelated in M. Then, neither P_u nor Q_u crosses P_v or Q_v in M.*

PROOF. Without loss of generality, we may assume that $I_M(u) < I_M(v)$. By Lemmas 2.3 and 2.4, P_u and P_v do not cross each other and neither do Q_u and Q_v. Consequently, P_u and Q_v cannot cross each other, because P_u appears on the left of P_v while Q_v appears on the right of P_v. For a similar reason, Q_u and P_v cannot cross each other if u covers v in M. If u and v are unrelated in M, then Q_u and P_v cannot cross each other either, because $\ell_{J_M(u)}$ is on the left of $\ell_{I_M(v)}$ and Q_u starts at $\ell_{J_M(u)}$ and goes up all the way to the left while P_v starts at $\ell_{I_M(v)}$ and goes up all the way to the right. □

The next lemma helps the reader understand how a nonleaf crosses another in M.

Lemma 2.6 *Suppose that a nonleaf u crosses another v in M. Then, the path Q_u from u to $\ell_{J_M(u)}$ in M and the path P_v from v to $\ell_{I_M(v)}$ in M cross each other exactly once and hence there is exactly one 2-block $B_{u,v}$ in M whose left vertex is on Q_u and whose right vertex is on P_v.*

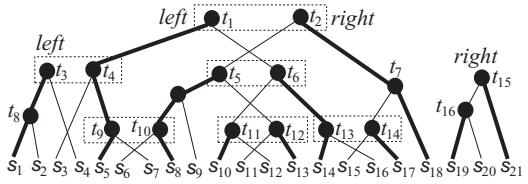

Figure 4: A 4-root model M for $\langle s_1, s_2, \ldots, s_{21} \rangle$ whose roots are marked $left$ or $right$.

PROOF. Let P_u (respectively, Q_v) be the path from u (respectively, v) to $\ell_{I_M(u)}$ (respectively, $\ell_{J_M(v)}$) in M. Since $I_M(u) < I_M(v) < J_M(u) < J_M(v)$, at least one of P_u and Q_u has to cross at least one of P_v and Q_v in M. By Lemmas 2.3 and 2.4, P_u and P_v do not cross each other and neither do Q_u and Q_v. Thus, Q_u and P_v cross each other or P_u and Q_v cross each other. As in the proof of Lemma 2.5, we can show that P_u and Q_v do not cross each other. Hence, Q_u and P_v cross each other. We also know that they can cross each other at most once because Q_u goes down all the way to the right while P_v goes down all the way to the left. Thus, they cross each other exactly once. □

We call the block $B_{u,v}$ in Lemma 2.6 the *witness block for the* (u, v)*-crossing* in M.

The following two lemmas help the reader understand the relations between unnested nonleaves in M and have been proved in [2]:

Lemma 2.7 *There do not exist three pairwise unnested nonleaves x, y, and z in M such that both x and z cross y in M.*

Lemma 2.8 *Suppose that x, y, and z are three pairwise unnested nonleaves in M such that x crosses y in M and y crosses z in M. Then, $I_M(y) < J_M(x) < I_M(z) < J_M(y)$ and hence x does not cross z in M.*

3 Multiroot Models and Splitting Vertices

For technical reasons, we generalize duplication models to multiroot models. Fix two integers i and j with $1 \leq i \leq j \leq n$. For convenience, we also call a duplication model for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ a *1-root model* for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$. For an integer $k \geq 2$, a *k-root model* for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ is obtained from a duplication model M for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ by deleting a subtree T with exactly $k-1$ vertices such that

- T contains the root of M but contains no leaf of M.
- For every 2-block B in M, T contains either both or neither of the vertices in B.

Figure 4 depicts a 4-root model.

For convenience, we define a *multiroot model* for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ to be a k-root model for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ with $k \geq 1$. Obviously, all definitions, notations, and lemmas for (1-root) models given in Section 2 still make sense for multiroot models. Besides, we have the following lemma:

Lemma 3.1 *Every two roots in a multiroot model M for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ are unnested.*

PROOF. The lemma is trivially true when M has only one root. So, suppose that M has two or more roots. Then, M is obtained from a (1-root) model M' for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ by deleting the root x and possibly some other nonleaves.

Suppose that u_1 and u_2 are two roots in M. Toward a contradiction, assume that u_1 covers u_2. Then, by Lemma 2.5, u_2 appears below u_1 in M (and hence appears below u_1 in M', too). On the other hand, x appears above u_1 in M'. Thus, the path from x to u_2 in M' has to cross the path from u_1 to $\ell_{I_M(u_1)}$ or the path from u_1 to $\ell_{J_M(u_1)}$. In either case, some edge (y_1, y_2) in M' but not in M is crossed by some edge (z_1, z_2) of M in M'. Obviously, y_1 and z_1 together form a 2-block in M'. Moreover, z_1 is in M but y_1 is not. However, when we obtained M from M', we would have deleted either both or none of y_1 and z_1 because they together form a 2-block. □

For a multiroot model M for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$, the *left* (respectively, *right*) *boundary* of M is the path that starts at the leftmost (respectively, rightmost) leaf of M and then repeats moving up to the parent of the current vertex until a root is reached. By Lemma 2.4, the left (respectively, right) boundary of M is a left (respectively, right) path in M. For example, the left boundary of the multiroot model in Figure 4 is the path: s_1, t_8, t_3, while the right boundary is the path: s_{21}, t_{15}.

A multiroot model for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ is *root-marked* if each root u of M is marked either $left$ or $right$ such that the following hold:

- If u is the left vertex of a 2-block in M, or u is on the left boundary of M but is not on the right boundary of M, then u is marked $left$.
- If u is the right vertex of a 2-block in M, or u is on the right boundary of M but is not on the left boundary of M, then u is marked $right$.

Throughout the remainder of this section, let M be a root-marked multiroot model for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$. Let V be the vertex set of M. We define five functions P_M, L_M, R_M, U_M, $D_M : V \to V \cup \{\perp\}$ as follows:

- For each $v \in V$, $P_M(v)$ is the parent of v in M if v is not a root of M; otherwise, $P_M(v) = \perp$.
- For each $v \in V$, $L_M(v)$ is the left child of v in M if v is not a leaf of M; otherwise, $L_M(v) = \perp$.
- For each $v \in V$, $R_M(v)$ is the right child of v in M if v is not a leaf of M; otherwise, $R_M(v) = \perp$.
- For each $v \in V$,
  - $D_M(v) = \perp$ if v is a leaf of M;
  - $D_M(v) = L_M(v)$ if (1) v is a root marked $left$, (2) v is the left child of its parent in M and $(v, L_M(v))$ is not crossed in M, or (3) the edge $(v, R_M(v))$ is crossed in M;
  - $D_M(v) = R_M(v)$ otherwise.
- For each $v \in V$, $U_M(v) = \perp$ if there is no $u \in V$ with $D_M(u) = v$; otherwise, $U_M(v) = P_M(v)$.

Intuitively speaking, function D_M tells us the direction when we move down from a vertex in tree M, while function U_M tells us to stop or continue when we move up from a vertex in tree M. As the following fact shows, U_M is simply the reverse function of D_M and *vice versa*.

Fact 3.2 *If u is a vertex of M with $U_M(u) \neq \perp$, then $D_M(U_M(u)) = u$. Similarly, if v is a vertex of M with $D_M(v) \neq \perp$, then $U_M(D_M(v)) = v$.*

PROOF. Obvious from the definitions of functions U_M and D_M. □

For each vertex v of M, consider the planar path that starts at v and then repeats moving down to $D_M(u)$ from the current vertex u until a leaf of M is reached. We denote this path by $\overrightarrow{D_M}(v)$. As an example, for the duplication model M in Figure 4, $\overrightarrow{D_M}(t_1)$, ..., $\overrightarrow{D_M}(t_3)$, $\overrightarrow{D_M}(t_5)$, $\overrightarrow{D_M}(t_6)$, $\overrightarrow{D_M}(t_{11})$, $\overrightarrow{D_M}(t_{12})$, $\overrightarrow{D_M}(t_{14})$, ..., $\overrightarrow{D_M}(t_{16})$ are the bold paths in the figure. The following fact is clear from the definition of D_M:

Fact 3.3 *If u and v are incomparable nonleaves of M, then $\overrightarrow{D_M}(u)$ and $\overrightarrow{D_M}(v)$ are vertex-disjoint paths.*

Since the left boundary of M is a left path and the right boundary of M is a right path, the following fact holds:

Fact 3.4 *The following statements hold:*

1. *For every nonroot vertex u on the left (respectively, right) boundary of M, $\overrightarrow{D_M}(u)$ is a subpath of the left (respectively, right) boundary of M.*
2. *If M has at least two roots, then for the root x on the left (respectively, right) boundary of M, $\overrightarrow{D_M}(x)$ is the left (respectively, right) boundary of M.*
3. *If M has only one root, then for the root x of M, $\overrightarrow{D_M}(x)$ is either the left or the right boundary of M.*

For each leaf v of M, consider the planar path that starts at v and then repeats moving up to $U_M(u)$ from the current vertex u until a vertex u with $U_M(u) = \perp$ is reached. We denote this path by $\overleftarrow{U_M}(v)$. Since D_M is a function, the following fact is clear from Fact 3.2:

Fact 3.5 *If u and v are distinct leaves of M, then $\overleftarrow{U_M}(u)$ and $\overleftarrow{U_M}(v)$ are vertex-disjoint paths.*

By Lemma 3.1, there exists an obvious left-to-right order of the roots in M, namely, a root x is on the left of another y in M if $I_M(x) < I_M(y)$ (or equivalently, $J_M(x) < J_M(y)$). Two roots in M are *consecutive* if they appear consecutively in this order among the roots.

Let the left-to-right order of the roots in M be $x_1, \ldots, x_k$. A *splitting vertex* of M is a vertex $v \notin \{x_1, \ldots, x_k\}$ such that for some integer $h \in \{1, \ldots, k-1\}$, x_h crosses x_{h+1} in M and the witness block for the (x_h, x_{h+1})-crossing in M contains v. For example, the multiroot model in Figure 4 has exactly one splitting vertex, namely, t_4.

Lemma 3.6 *Suppose that $k \geq 3$ and for some integer h with $1 < h < k$, x_{h-1} crosses x_h and x_h crosses x_{h+1} in M. Then, the right vertex in the witness block for the (x_{h-1}, x_h)-crossing in M or the left vertex in the witness block for the (x_h, x_{h+1})-crossing in M is a splitting vertex of M.*

PROOF. If the right vertex y in the witness block for the (x_{h-1}, x_h)-crossing in M is not a splitting vertex of M, then $y = x_h$ and so x_h cannot be the left vertex z in the witness block for the (x_h, x_{h+1})-crossing in M, implying z is a splitting vertex of M. □

Lemma 3.7 *For every splitting vertex v of M, no vertex of $\overrightarrow{D_M}(v)$ appears on the left or right boundary of M.*

PROOF. Let x_h be the root in M that is also an ancestor of v. We assume that v is the left vertex in a 2-block; the case where v is the right vertex in a 2-block is similar. Then, $D_M(v)$ is the left child of v in M. So, by Lemmas 2.1 and 2.2, $\overrightarrow{D_M}(v)$ appears on the left of the path from v to $\ell_{J_M(v)}$ in M. Consequently, $\overrightarrow{D_M}(v)$ cannot pass a vertex on the right boundary of M.

By Lemma 2.6, v is a descendant of the right child of x_h in M. So, by Lemma 2.2, $\overrightarrow{D_M}(v)$ appears on the right of the path from v to $\ell_{I_M(x_h)}$ in M. Consequently, $\overrightarrow{D_M}(v)$ cannot pass a vertex on the left boundary of M. □

4 Splitting Multiroot Models

Throughout this section, let i and j be two integers with $1 \leq i \leq j \leq n$, let M be a multiroot model for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$, let v be a splitting vertex v of M, and let ℓ_b be the leaf of M at which $\overrightarrow{D_M}(v)$ ends. Let M_l be the graph obtained from M as follows (cf. Figure 5):

1. Delete every vertex that lies on the right of $\overrightarrow{D_M}(v)$. (*Comment:* If v is the right vertex in a 2-block before this step, then v becomes a new root after this step. Moreover, some nonroot vertices may have only one child after this step.)

2. If v is the right vertex in a 2-block, then mark v *right*.

Note that M_l is a multiroot model for $\langle s_i, s_{i+1}, \ldots, s_b \rangle$ if no vertex in M_l has only one child. However, some vertices in M_l may have only one child. So, we call M_l the *left root-marked multiroot semi-model* obtained by splitting M along $\overrightarrow{D_M}(v)$. Similarly, we can define M_r, the *right root-marked multiroot semi-model* obtained by splitting M along $\overrightarrow{D_M}(v)$. In detail, M_r is obtained from M as follows (cf. Figure 5):

1. Delete every vertex that lies on the left of $\overrightarrow{D_M}(v)$.

2. If v is the left vertex in a 2-block, then mark v $left$.

Note that M_r is a multiroot model for $\langle s_b, s_{b+1}, \ldots, s_j \rangle$ if no vertex in M_r has only one child.

We can obtain a root-marked multiroot model for $\langle s_i, s_{i+1}, \ldots, s_b \rangle$ (respectively, $\langle s_b, s_{b+1}, \ldots, s_j \rangle$) from M_l (respectively, M_r) by repeating the following step (cf. Figure 6):

- If some vertex u has only one child in M_l (respectively, M_r), then add a new edge from the parent of u to the child of u and further delete u together with the two edges incident to it.

We call the multiroot model obtained from M_l (respectively, M_r) as above the *left* (respectively, *right*) *root-marked multiroot model* obtained by splitting M along $\overrightarrow{D_M}(v)$, and use $\widetilde{M_l}$ (respectively, $\widetilde{M_r}$) to denote it.

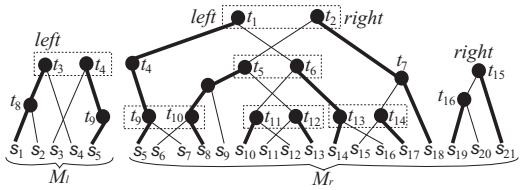

Figure 5: The left and the right root-marked multiroot semi-models obtained by splitting the multiroot model M in Figure 4 along $\overrightarrow{D_M}(t_4)$.

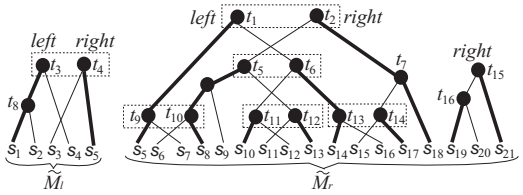

Figure 6: The left and the right root-marked multiroot models obtained by splitting the multiroot model M in Figure 4 along $\overrightarrow{D_M}(t_4)$.

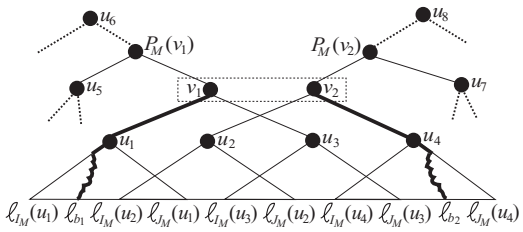

Figure 7: A sketch of a 2-root model where the bold paths are $D_M(v_1)$ and $D_M(v_2)$.

Figure 7 gives a sketch of a 2-root model which may help the reader understand the proofs of Lemmas 4.1, 4.2, and 4.3. Intuitively speaking, v_1 and v_2 in the figure correspond to v in the proofs, u_1 through u_8 in the figure correspond to u in the proofs, and ℓ_{b_1} and ℓ_{b_2} in the figure correspond to ℓ_b in the proofs.

Lemma 4.1 *For each vertex u of $\widetilde{M}_l$ that is not an ancestor of ℓ_b, $D_M(u) = D_{\widetilde{M}_l}(u)$. Similarly, for each vertex u of $\widetilde{M}_r$ that is not an ancestor of ℓ_b, $D_M(u) = D_{\widetilde{M}_r}(u)$.*

Proof. We only prove the first assertion; the other proof is similar. Let u be a vertex of $\widetilde{M}_l$ that is not an ancestor of ℓ_b. If u is a leaf in $\widetilde{M}_l$, then clearly $D_M(u) = D_{\widetilde{M}_l}(u) = \bot$. So, assume that u is a nonleaf in $\widetilde{M}_l$. Then, since u is not an ancestor of ℓ_b, neither child of u is on $\overrightarrow{D_M}(v)$. So, $L_M(u) = L_{\widetilde{M}_l}(u)$ and $R_M(u) = R_{\widetilde{M}_l}(u)$. Moreover, since $\overrightarrow{D_M}(v)$ is a planar path, $(u, L_M(u))$ is a planar edge in M if and only if $(u, L_{\widetilde{M}_l}(u))$ is a planar edge in $\widetilde{M}_l$. Similarly, $(u, R_M(u))$ is a planar edge in M if and only if $(u, R_{\widetilde{M}_l}(u))$ is a planar edge in $\widetilde{M}_l$. Now, if u is a root in M, then $D_M(u) = D_{\widetilde{M}_l}(u)$ because u is marked $left$ (respectively, $right$) in M if and only u is marked $left$ (respectively, $right$) in $\widetilde{M}_l$. On the other hand, if u is not a root in M, $P_M(u)$ may appear on $\overrightarrow{D_M}(v)$ or not. An obvious but crucial point is that if $P_M(u)$ appears on $\overrightarrow{D_M}(v)$, then u is the left child of $P_M(u)$ in both M and $\widetilde{M}_l$. So, no matter whether $P_M(u)$ appears on $\overrightarrow{D_M}(v)$ or not, one

can easily verify that $P_M(u) = P_{\widetilde{M}_l}(u)$ and that u is the left (respectively, right) child of its parent in M if and only if u is the left (respectively, right) child of its parent in $\widetilde{M}_l$. Thus, we also have $D_M(u) = D_{\widetilde{M}_l}(u)$ when u is not a root in M. □

Lemma 4.2 *For each vertex u of M that is also a vertex of $\widetilde{M}_l$, $\overrightarrow{D_M}(u)$ and $\overrightarrow{D_{\widetilde{M}_l}}(u)$ end at the same leaf. Similarly, for each vertex u of M that is also a vertex of $\widetilde{M}_r$, $\overrightarrow{D_M}(u)$ and $\overrightarrow{D_{\widetilde{M}_r}}(u)$ end at the same leaf.*

Proof. We only prove the first assertion; the other proof is similar. Recall that ℓ_b is the leaf of M at which $\overrightarrow{D_M}(v)$ ends. Let u be a vertex of M that is also a vertex of $\widetilde{M}_l$. If u is not an ancestor of ℓ_b in $\widetilde{M}_l$, then by Lemma 4.1, $\overrightarrow{D_M}(u)$ and $\overrightarrow{D_{\widetilde{M}_l}}(u)$ are identical and hence end at the same leaf. Moreover, if u is an ancestor of ℓ_b in $\widetilde{M}_l$ and is also a descendant of v in M, then $\overrightarrow{D_M}(u)$ clearly ends at ℓ_b and so does $\overrightarrow{D_{\widetilde{M}_l}}(u)$ (by Fact 3.4) because u is on the right boundary of $\widetilde{M}_l$. So, assume that u is an ancestor of ℓ_b in $\widetilde{M}_l$ and is also an ancestor of v in M. We distinguish two cases as follows:

Case 1: v is the right vertex of a 2-block in M. In this case, v is a root in $\widetilde{M}_l$ and is marked *right*. Thus, $u = v$ and in turn both $\overrightarrow{D_M}(u)$ and $\overrightarrow{D_{\widetilde{M}_l}}(u)$ end at ℓ_b.

Case 2: v is the left vertex of a 2-block in M. In this case, v does not appear in $\widetilde{M}_l$ but $P_M(v)$ remains in $\widetilde{M}_l$. We further distinguish three subcases as follows:

Subcase 2.1: u is not a root in M. Then, by Lemmas 2.4 and 2.6, the path from x to v in M is a right path, where x is the root of M that is also an ancestor of u. So, $\overrightarrow{D_M}(u)$ passes v and hence ends at ℓ_b. Moreover, u is not a root in $\widetilde{M}_l$ and appears on the right boundary of $\widetilde{M}_l$, implying that $\overrightarrow{D_{\widetilde{M}_l}}(u)$ ends at ℓ_b by Fact 3.4.

Subcase 2.2: u is a root marked *right* in M. In this case, $D_M(u) = R_M(u)$ and $D_{\widetilde{M}_l}(u) = R_{\widetilde{M}_l}(u)$. Since $D_M(u) = R_M(u)$, $\overrightarrow{D_M}(u)$ passes v by Lemmas 2.4 and 2.6, implying that $\overrightarrow{D_M}(u)$ ends at ℓ_b. On the other hand, since $D_{\widetilde{M}_l}(u) = R_{\widetilde{M}_l}(u)$ and $R_{\widetilde{M}_l}(u)$ is a nonroot vertex on the right boundary of $\widetilde{M}_l$, $\overrightarrow{D_{\widetilde{M}_l}}(u)$ ends at ℓ_b by Fact 3.4.

Subcase 2.3: u is a root marked $left$ in M. In this case, $D_{\widetilde{M}_l}(u)$ is the same as $D_M(u)$ and is not an ancestor of ℓ_b in $\widetilde{M}_l$. Thus, by Lemma 4.1, $\overrightarrow{D_{\widetilde{M}_l}}(u)$ and $\overrightarrow{D_M}(u)$ end at the same leaf. □

Lemma 4.3 *For each vertex u of $\widetilde{M}_l$ that is not an ancestor of ℓ_b, $U_M(u) = U_{\widetilde{M}_l}(u)$. Similarly, for each vertex u of $\widetilde{M}_r$ that is not an ancestor of ℓ_b, $U_M(u) = U_{\widetilde{M}_r}(u)$.*

Proof. We only prove the first assertion; the other proof is similar. Let u be a vertex of $\widetilde{M}_l$ that is not an ancestor of ℓ_b. If $P_{\widetilde{M}_l}(u)$ exists and is not an ancestor of ℓ_b, then by Lemma 4.1, $U_M(u) = U_{\widetilde{M}_l}(u)$. Moreover, if u is a root in $\widetilde{M}_l$, then u is also a root in M (because u cannot be v), implying that $U_M(u) = U_{\widetilde{M}_l}(u) = \perp$. So, assume that u is not a root in $\widetilde{M}_l$ and $P_{\widetilde{M}_l}(u)$ is an ancestor of ℓ_b. Then, since u is not an ancestor of ℓ_b, u is the left child of $P_{\widetilde{M}_l}(u)$ in both $\widetilde{M}_l$ and M. We distinguish three cases as follows:

Case 1: $P_{\widetilde{M}_l}(u)$ is not a root in $\widetilde{M}_l$. In this case, $P_{\widetilde{M}_l}(u)$ is a nonroot vertex on the right boundary of $\widetilde{M}_l$. Thus, $\overrightarrow{D_{\widetilde{M}_l}}(P_{\widetilde{M}_l}(u))$ ends at ℓ_b by Fact 3.4. So, by Lemma 4.2, $\overrightarrow{D_M}(P_{\widetilde{M}_l}(u))$

ends at ℓ_b, too. Now, since u is not an ancestor of ℓ_b in both $\widetilde{M}_l$ and M, $D_{\widetilde{M}_l}(P_{\widetilde{M}_l}(u)) \neq u$ and $D_M(P_{\widetilde{M}_l}(u)) \neq u$. Consequently, $U_{\widetilde{M}_l}(u) = \perp$ and $U_M(u) = \perp$.

Case 2: $P_{\widetilde{M}_l}(u)$ is a root marked *right* in $\widetilde{M}_l$. In this case, clearly $U_{\widetilde{M}_l}(u) = \perp$. Moreover, either $P_{\widetilde{M}_l}(u) = v$, or $\overrightarrow{D_M}(P_{\widetilde{M}_l}(u))$ passes v by Lemmas 2.4 and 2.6. In either case, $\overrightarrow{D_M}(P_{\widetilde{M}_l}(u))$ ends at ℓ_b, implying that $U_M(u) = \perp$.

Case 3: $P_{\widetilde{M}_l}(u)$ is a root marked $left$ in $\widetilde{M}_l$. In this case, clearly $U_{\widetilde{M}_l}(u) = P_{\widetilde{M}_l}(u) = P_M(u) = U_M(u)$. □

Lemma 4.4 *For every leaf $\ell_h \notin \{\ell_i, \ell_b\}$ in $\widetilde{M}_l$, $\overleftarrow{U_M}(\ell_h) = \overleftarrow{U_{\widetilde{M}_l}}(\ell_h)$. Similarly, for every leaf $\ell_h \notin \{\ell_b, \ell_j\}$ in $\widetilde{M}_r$, $\overleftarrow{U_M}(\ell_h) = \overleftarrow{U_{\widetilde{M}_r}}(\ell_h)$.*

Proof. We only prove the first assertion; the other proof is similar. Consider a leaf $\ell_h \notin \{\ell_i, \ell_b\}$ in $\widetilde{M}_l$. By Fact 3.2, there is a vertex u in $\widetilde{M}_l$ with $U_{\widetilde{M}_l}(u) = \perp$ such that $\overleftarrow{U_{\widetilde{M}_l}}(\ell_h) = \overrightarrow{D_{\widetilde{M}_l}}(u)$. If u were an ancestor of ℓ_b in $\widetilde{M}_l$, then $\overrightarrow{D_{\widetilde{M}_l}}(u)$ would end at ℓ_i or ℓ_b by Fact 3.4, no matter whether u is a root in $\widetilde{M}_l$ or not. Thus, u is not an ancestor of ℓ_b in $\widetilde{M}_l$. Consequently, by Lemma 4.3, $\overleftarrow{U_M}(\ell_h) = \overleftarrow{U_{\widetilde{M}_l}}(\ell_h)$. □

For a multiroot model N, we use $c(N)$ to denote the total cost of edges in N. Similarly, for a path P in N, we use $c(P)$ to denote the total cost of edges on P. The following lemma will be very useful:

Lemma 4.5 *$c(\widetilde{M}_l) + c(\widetilde{M}_r) \leq c(M) + c(\overleftarrow{U_M}(\ell_b))$. Moreover, the hamming distance between s_b and $s(v)$ does not exceed $c(\overleftarrow{U_M}(\ell_b))$, where $s(v)$ is the string assigned to v in M.*

Proof. Recall that $\overrightarrow{D_M}(v)$ starts at v and ends at ℓ_b. So, by the triangle inequality, the hamming distance between s_b and $s(v)$ does not exceed $c(\overrightarrow{D_M}(v))$. Moreover, since $\overrightarrow{D_M}(v)$ ends at ℓ_b, $\overrightarrow{D_M}(v)$ is a subpath of $\overleftarrow{U_M}(\ell_b)$ by Fact 3.2, implying that $c(\overrightarrow{D_M}(v)) \leq c(\overleftarrow{U_M}(\ell_b))$. Thus, the second assertion in the lemma clearly holds.

To prove the first assertion, first note that $c(M_l) + c(M_r) = c(M) + c(\overrightarrow{D_M}(v))$. Moreover, by the triangle inequality, $c(\widetilde{M}_l) \leq c(M_l)$ and $c(\widetilde{M}_r) \leq c(M_r)$. Now, by the last inequality in the last paragraph, the first assertion in the lemma also holds. □

A *separator* of M is a set Γ of leaves such that for every 2-block in M consisting of two vertices u_1 and u_2, at least one of $\overrightarrow{D_M}(u_1)$ and $\overrightarrow{D_M}(u_2)$ ends at a leaf in Γ. For example, for the duplication model in Figure 2, $\{\ell_2, \ell_3, \ell_5, \ell_7, \ell_9, \ell_{10}, \ell_{12}, \ell_{14}, \ell_{15}, \ell_{17}\}$ is a separator. Moreover, for the multiroot model in Figure 4, $\{\ell_1, \ell_6, \ell_8, \ell_9, \ell_{11}, \ell_{13}, \ell_{14}, \ell_{16}, \ell_{18}\}$ is a separator. Note that the set of all leaves of M is a trivial separator of M. However, what we want is a separator Γ of M such that the total cost of the paths in $\{\overleftarrow{U_M}(v) \mid v \in \Gamma\}$ is small.

By Lemma 4.2, we have the following corollary immediately:

Corollary 4.6 *For every separator Γ of M and for every splitting vertex v of M, Γ_1 (respectively, Γ_2) is a separator of the left (respectively, right) root-marked multiroot model obtained by splitting M along $\overrightarrow{D_M}(v)$, where Γ_1 (respectively, Γ_2) is the set of those leaves in Γ that are also leaves of $\widetilde{M}_l$ (respectively, $\widetilde{M}_r$).*

Lemma 4.7 *Suppose that M has two consecutive roots x_1 and x_2 such that x_1 crosses x_2 in M and the witness block for the (x_1, x_2)-crossing in M contains neither x_1 nor x_2. Let Γ be a separator of M. Then, there is a splitting vertex u such that $\overrightarrow{D_M}(u)$ ends at a leaf in Γ.*

PROOF. Let u_1 and u_2 be the left and the right vertex in the witness block for the (x_1, x_2)-crossing in M, respectively. Since Γ is a separator, at least one of $\overrightarrow{D_M}(u_1)$ and $\overrightarrow{D_M}(u_2)$ ends at a leaf in Γ. If $\overrightarrow{D_M}(u_1)$ ends at a leaf in Γ, then u_1 is a desired splitting vertex. Similarly, if $\overrightarrow{D_M}(u_2)$ ends at a leaf in Γ, then u_2 is a desired splitting vertex. □

Lemma 4.8 *Suppose that M has at least four roots and every two consecutive roots in M cross each other. Let Γ be a separator of M. Then, there is a splitting vertex u such that $\overrightarrow{D_M}(u)$ ends at a leaf in Γ.*

PROOF. Let $x_1, \ldots, x_k$ be the roots in M. Let u_2 and u_3 be the left and the right vertex in the witness block for the (x_2, x_3)-crossing in M, respectively. Since Γ is a separator, at least one of the following cases occurs:

Case 1: $\overrightarrow{D_M}(u_2)$ ends at a leaf in Γ. If $u_2 \neq x_2$, then u_2 is a desired splitting vertex. If $u_2 = x_2$, consider the right vertex y in the witness block for the (x_1, x_2)-crossing in M. Obviously, $y \neq x_2$ and $D_M(x_2) = L_M(x_2)$. Thus, applying Lemma 2.6 to the (x_1, x_2)-crossing, we see that $\overrightarrow{D_M}(x_2)$ must pass y by Lemma 2.4. Hence, $\overrightarrow{D_M}(y)$ ends at a leaf in Γ. Consequently, y is a desired splitting vertex.

Case 2: $\overrightarrow{D_M}(u_3)$ ends at a leaf in Γ. If $u_3 \neq x_3$, then u_3 is a desired splitting vertex. If $u_3 = x_3$, consider the left vertex y in the witness block for the (x_3, x_4)-crossing in M. Obviously, $y \neq x_3$ and $D_M(x_3) = L_M(x_3)$. Thus, applying Lemma 2.6 to the (x_3, x_4)-crossing, we see that $\overrightarrow{D_M}(x_3)$ must pass y by Lemma 2.4. Hence, $\overrightarrow{D_M}(y)$ ends at a leaf in Γ. Consequently, y is a desired splitting vertex. □

For a constant δ, a *δ-separator* of M is a separator Γ of M such that the total cost of the paths in $\{\overleftarrow{U_M}(v) \mid v \in \Gamma\}$ is at most $\delta \cdot c(M)$.

5 The Existence of 0.75-Separators

For convenience, we represent each bijection G from a set Y to a set Z as a set consisting of all pairs $(y, G(y))$ with $y \in Y$. Moreover, if x is a root but not a leaf in a root-marked multiroot model M, then *omitting x from M* means the operation of modifying M as follows:

1. Delete x and the two edges incident to it from M.
2. If $L_M(x)$ is the right vertex of some 2-block in M, then mark $L_M(x)$ $right$; otherwise, mark $L_M(x)$ $left$.
3. If $R_M(x)$ is the left vertex of some 2-block in M, then mark $R_M(x)$ $left$; otherwise, mark $R_M(x)$ $right$.

Lemma 5.1 *For every two integers i and j with $1 \leq i \leq j \leq n$ and for every root-marked model M for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ with one or two roots, we can compute three disjoint subsets X, Y, and Z of $\{i, i+1, \ldots, j\}$ and a bijection $G : Y \to Z$ satisfying the following conditions:*

1. *Both $i \in X$ and $j \in X$.*

2. *For every multiset $\mathcal{L}$ that can be obtained from $X \cup Y \cup Z$ by adding either y or $G(y)$ for every $y \in Y$, if we sort $\mathcal{L}$ in nondecreasing order, then for every 2-block in M consisting of two vertices v_1 and v_2, there are two integers b_1 and b_2 of different parity such that $\overrightarrow{D_M}(v_1)$ ends at ℓ_{k_1} and $\overrightarrow{D_M}(v_2)$ ends at ℓ_{k_2}, where k_1 and k_2 are the b_1th and the b_2th integer in $\mathcal{L}$, respectively.*

PROOF. By induction on the total number of vertices and edges in M. In the base case, $i = j$ and there is only one vertex in M; we just let $X = \{i\}$, $Y = Z = \emptyset$, and $G = \emptyset$ which clearly satisfy the conditions in the lemma. So, suppose that M has two or more vertices. We distinguish three cases as follows.

Case 1: M has only one root. Let u be the root of M. Consider the root-marked 2-root model N for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ obtained from M by omitting u. By the inductive hypothesis, we can compute three subsets X_N, Y_N, Z_N and a bijection $G_N : Y_N \to Z_N$ for N. We let $X = X_N$, $Y = Y_N$, $Z = Z_N$, and $G = G_N$. Obviously, X, Y, Z, and G satisfy the conditions in the lemma.

Case 2: M has two roots but they together do not form a 2-block in M. Let v_1 and v_2 be the left and the right root of M, respectively. One of the following three subcases must occur:

Subcase 2.1: v_1 does not cross v_2 in M. In this subcase, there is an integer $k \in \{i, i+1, \ldots, j-1\}$ such that ℓ_i through ℓ_k are the leaf descendants of v_1 in M while ℓ_{k+1} through ℓ_j are the leaf descendants of v_2 in M. Let M_1 be the root-marked model for $\langle s_i, s_{i+1}, \ldots, s_k \rangle$ obtained from M by deleting the descendants of v_2 in M. Similarly, let M_2 be the root-marked model for $\langle s_{k+1}, s_{k+2}, \ldots, s_j \rangle$ obtained from M by deleting the descendants of v_1 in M. For each $h \in \{1, 2\}$, let X_h, Y_h, Z_h, and G_h be the subsets and the function computed for M_h. Obviously, $X = X_1 \cup X_2$, $Y = Y_1 \cup Y_2$, $Z = Z_1 \cup Z_2$, and $G = G_1 \cup G_2$ satisfy the conditions in the lemma.

Subcase 2.2: There is a 2-block in M consisting of v_1 and a vertex $u_2 \neq v_2$ such that u_2 is a descendant of v_2 in M. Let M_1 and M_2 be the left and the right root-marked multiroot model obtained by splitting M along $\overrightarrow{D_M}(u_2)$, respectively. For each $h \in \{1, 2\}$, let X_h, Y_h, Z_h, and G_h be the subsets and the function computed for M_h. By Lemma 4.2, $X = X_1 \cup X_2$, $Y = Y_1 \cup Y_2$, $Z = Z_1 \cup Z_2$, and $G = G_1 \cup G_2$ satisfy the conditions in the lemma.

Subcase 2.3: v_1 crosses v_2 in M but no 2-block in M contains v_1. Let u_1 be the left vertex in the witness block for (v_1, v_2)-crossing in M. Let M_1 and M_2 be the left and the right root-marked multiroot model obtained by splitting M along $\overrightarrow{D_M}(u_1)$, respectively. For each $h \in \{1, 2\}$, let X_h, Y_h, Z_h, and G_h be the subsets and the function computed for M_h. By Lemma 4.2, $X = X_1 \cup X_2$, $Y = Y_1 \cup Y_2$, $Z = Z_1 \cup Z_2$, and $G = G_1 \cup G_2$ satisfy the conditions in the lemma.

Case 3: M has two roots and they together form a 2-block in M. Let v_1 and v_2 be the left and the right root of M, respectively. Let N be the root-marked multiroot model obtained from M by omitting both v_1 and v_2. Let u_1 and u_3 be the left and the right child of v_1, respectively. Let u_2 and u_4 be the left and the right child of v_2, respectively. One of the following five subcases must occur:

Subcase 3.1: u_2 does not cross u_3 in N. In this subcase, there is an integer k such that ℓ_i, ℓ_{i+1}, $\ldots$, ℓ_k are the leaf descendants of u_1 or u_2 in N. Let N_1 be the root-marked 2-root model for $\langle s_i, s_{i+1}, \ldots, s_k \rangle$ obtained from N by deleting all vertices that are descendants of u_3 or u_4 in N. Let N_2 be the root-marked 2-root model for $\langle s_{k+1}, s_{k+2}, \ldots, s_j \rangle$ obtained from N by deleting all vertices

that are descendants of u_1 or u_2 in N. For each $h \in \{1,2\}$, let X_h, Y_h, Z_h, and G_h be the subsets and the function computed for N_h. If $\sum_{h=1}^{2}(|X_h|+|Y_h|*3)$ is even, then let $X = \bigcup_{h=1}^{2} X_h$, $Y = \bigcup_{h=1}^{2} Y_h$, $Z = \bigcup_{h=1}^{2} Z_h$, and $G = \bigcup_{h=1}^{2} G_h$; otherwise, let $X = \bigcup_{h=1}^{2} X_h - \{k, k+1\}$, $Y = \bigcup_{h=1}^{2} Y_h \bigcup \{k\}$, $Z = \bigcup_{h=1}^{2} Z_h \bigcup \{k+1\}$, and $G = \bigcup_{h=1}^{2} G_h \bigcup \{(k, k+1)\}$. Obviously, X, Y, Z, and G satisfy the conditions in the lemma.

Subcase 3.2: u_2 crosses u_3 in N, u_1 does not cross u_2 in N, and u_3 does not cross u_4 in N. In this subcase, there are two distinct integers k_1 and k_2 such that the leaf descendants of u_1 in N are ℓ_i, ℓ_{i+1}, ..., ℓ_{k_1} and the leaf descendants of u_4 in N are ℓ_{k_2}, ℓ_{k_2+1}, ..., ℓ_j. Let N_1 be the root-marked model for $\langle s_i, s_{i+1}, \ldots, s_{k_1} \rangle$ obtained from N by deleting all vertices that are not descendants of u_1 in N. Let N_2 be the root-marked 2-root model for $\langle s_{k_1+1}, s_{k_1+2}, \ldots, s_{k_2-1} \rangle$ obtained from N by deleting all vertices that are descendants of u_1 or u_4 in N. Let N_3 be the root-marked model for $\langle s_{k_2}, s_{k_2+1}, \ldots, s_j \rangle$ obtained from N by deleting all vertices that are not descendants of u_4 in N. For each $h \in \{1,2,3\}$, let X_h, Y_h, Z_h, and G_h be the subsets and the function computed for N_h. If $\sum_{h=1}^{3}(|X_h|+|Y_h|*3)$ is even, then let $X = \bigcup_{h=1}^{3} X_h$, $Y = \bigcup_{h=1}^{3} Y_h$, $Z = \bigcup_{h=1}^{3} Z_h$, and $G = \bigcup_{h=1}^{3} G_h$; otherwise, let $X = \bigcup_{h=1}^{3} X_h - \{k_1+1, k_2-1\}$, $Y = \bigcup_{h=1}^{3} Y_h \bigcup \{k_1+1\}$, $Z = \bigcup_{h=1}^{3} Z_h \bigcup \{k_2-1\}$, and $G = \bigcup_{h=1}^{3} G_h \bigcup \{(k_1+1, k_2-1)\}$. Obviously, X, Y, Z, and G satisfy the conditions in the lemma.

Subcase 3.3: u_1 crosses u_2 in N and u_2 crosses u_3 but u_3 does not cross u_4 in N. In this subcase, there is an integer k_2 such that ℓ_{k_2}, ℓ_{k_2+1}, ..., ℓ_j are the leaf descendants of u_4 in N. Let N' be the multiroot model for $\langle s_i, s_{i+1}, \ldots, s_{k_2-1} \rangle$ obtained from N by deleting the descendants of u_4 in N. By Lemma 3.6, u_2 has a descendant x_2 in N' that is a splitting vertex of N'. Let ℓ_{k_1} be the leaf at which $\overrightarrow{D_{N'}}(x_2)$ ends. Let N_1 and N_2 be the left and the right root-marked multiroot model obtained by splitting N' along $\overrightarrow{D_{N'}}(x_2)$. Moreover, let N_3 be the root-marked model for $\langle s_{k_2}, s_{k_2+1}, \ldots, s_j \rangle$ obtained from N by deleting all vertices that are not descendants of u_4 in N. For each $h \in \{1,2,3\}$, let X_h, Y_h, Z_h, and G_h be the subsets and the function computed for N_h. If $\sum_{h=1}^{3}(|X_h|+|Y_h|*3)$ is even, then let $X = \bigcup_{h=1}^{3} X_h$, $Y = \bigcup_{h=1}^{3} Y_h$, $Z = \bigcup_{h=1}^{3} Z_h$, and $G = \bigcup_{h=1}^{3} G_h$; otherwise, let $X = \bigcup_{h=1}^{3} X_h - \{k_1, k_2-1\}$, $Y = \bigcup_{h=1}^{3} Y_h \bigcup \{k_1\}$, $Z = \bigcup_{h=1}^{3} Z_h \bigcup \{k_2-1\}$, and $G = \bigcup_{h=1}^{3} G_h \bigcup \{(k_1, k_2-1)\}$. By Lemma 4.2, X, Y, Z, and G satisfy the conditions in the lemma.

Subcase 3.4: u_2 crosses u_3 in N and u_3 crosses u_4 but u_1 does not cross u_2 in N. This subcase is similar to Subcase 3.3.

Subcase 3.5: For each $h \in \{1,2,3\}$, u_h crosses u_{h+1} in N. By Lemma 3.6, u_2 has a descendant x_2 in N that is a splitting vertex of N. Let ℓ_{k_2} be the leaf at which $\overrightarrow{D_N}(x_2)$ ends. Let N_1 and N' be the left and the right root-marked multiroot model obtained by splitting N along $\overrightarrow{D_N}(x_2)$. Note that N' has three roots, u_3 is the middle root of N', and each pair of consecutive roots cross in N'. So, by Lemma 3.6, u_3 has a descendant x_3 in N' that is a splitting vertex of N'. Let ℓ_{k_3} be the leaf at which $\overrightarrow{D_{N'}}(x_3)$ ends. Let N_2 and N_3 be the left and the right root-marked multiroot model obtained by splitting N' along $\overrightarrow{D_{N'}}(x_3)$. For each $h \in \{1,2,3\}$, let X_h, Y_h, Z_h, and G_h be the subsets and the function computed for N_h. If $\sum_{h=1}^{3}(|X_h| + |Y_h| * 3)$ is even, then let $X = \bigcup_{h=1}^{3} X_h$, $Y = \bigcup_{h=1}^{3} Y_h$, $Z = \bigcup_{h=1}^{3} Z_h$, and $G = \bigcup_{h=1}^{3} G_h$; otherwise, let $X = \bigcup_{h=1}^{3} X_h - \{k_2, k_3\}$, $Y = \bigcup_{h=1}^{3} Y_h \bigcup \{k_2\}$, $Z = \bigcup_{h=1}^{3} Z_h \bigcup \{k_3\}$, and $G = \bigcup_{h=1}^{3} G_h \bigcup \{(k_2, k_3)\}$. By Lemma 4.2, X, Y, Z, and G satisfy the conditions in the lemma. □

For example, for the duplication model M in Figure 2, the sets X, Y, Z, and the bijection G constructed in the proof of Lemma 5.1 are $\{1, 2, 5, 6, 7, 9, 11, 12, 15, \ldots, 18\}$, $\{3, 8, 10\}$, $\{4, 13, 14\}$, and $\{(3,4), (8,14), (10,13)\}$, respectively.

Lemma 5.2 *Every duplication model M for $\langle s_1, s_2, \ldots, s_n\rangle$ has a 0.75-separator.*

PROOF. Consider the three subsets X, Y, Z and the bijection G obtained by applying Lemma 5.1 with $i = 1$, $j = 1$, and the root of M marked *left*. We construct a multiset $\mathcal{L}$ as follows:

1. Initialize $\mathcal{L} = X \cup Y \cup Z$.
2. For each $h \in Y$, if $c(\overleftarrow{U_M}(\ell_h)) \leq c(\overleftarrow{U_M}(\ell_{G(h)}))$, then we add h to $\mathcal{L}$; otherwise, we add $G(h)$ to $\mathcal{L}$. (*Comment:* We call each integer added to $\mathcal{L}$ in this step a *duplicated* integer.)

By Fact 3.5, the total cost of paths in the set $\{\overleftarrow{U_M}(\ell_h) \mid h \in X \cup Y \cup Z\}$ does not exceed $c(M)$. Moreover, the total cost of paths in the set $\{\overleftarrow{U_M}(\ell_h) \mid h$ is a duplicated integer$\}$ is at most half the total cost of paths in the set $\{\overleftarrow{U_M}(\ell_h) \mid h \in Y \cup Z\}$. Thus, the total cost of the paths in the multiset $\{\overleftarrow{U_M}(\ell_h) \mid h \in \mathcal{L}\}$ is at most $1.5 \cdot c(M)$.

We are now ready to construct Γ from $\mathcal{L}$ as follows:

1. Sort the integers in $\mathcal{L}$ in nondecreasing order.
2. Let $\mathcal{L}_1$ (respectively, $\mathcal{L}_2$) be the set of integers that appear in odd (respectively, even) positions in $\mathcal{L}$.
3. If the total cost of paths in the set $\{\overleftarrow{U_M}(\ell_h) \mid h \in \mathcal{L}_1\}$ does not exceed the total cost of paths in the set $\{\overleftarrow{U_M}(\ell_h) \mid h \in \mathcal{L}_2\}$, then set $\Gamma = \{\ell_h \mid h \in \mathcal{L}_1\}$; otherwise, set $\Gamma = \{\ell_h \mid h \in \mathcal{L}_2\}$.

Clearly, Γ is a 0.75-separator of M. □

For example, if the cost of every edge is 1 in the duplication model M in Figure 2, then the 0.75-separator constructed in the proof of Lemma 5.2 is $\{\ell_2, \ell_3, \ell_5, \ell_7, \ell_9, \ell_{10}, \ell_{12}, \ell_{14}, \ell_{15}, \ell_{17}\}$.

6 The Component Tree of a Multiroot Model

For each vertex v of M, let $s(v)$ denote the string assigned to v in M. Moreover, for a list $\mathcal{L}$ of vertices in M, let $s(\mathcal{L})$ denote the list of strings assigned to the vertices in $\mathcal{L}$. Furthermore, for two strings s' and s'', let $d(s', s'')$ denote the hamming distance between them.

Let M be a root-marked multiroot model with at most five roots for $\langle s_i, s_{i+1}, \ldots, s_j\rangle$. Let Γ be a separator of M. We will use Γ to decompose M into components. Each component N will be a root-marked multiroot model with at most five roots for a list $\langle s_{i'}, s_{i'+1}, \ldots, s_{j'}\rangle$ with $i \leq i' \leq j' \leq j$. We call the triple $(s(\mathcal{L}), i', j')$ the *signature* of N, where $\mathcal{L}$ is the list of roots in N (ordered from left to right). If $j' > i'$, then N will be decomposed into smaller components. In summary, we will start with M and obtain a lot of components. These components will then be organized into a tree $\mathcal{D}(M, \Gamma)$. Each node of $\mathcal{D}(M, \Gamma)$ corresponds to a component N, is labeled with the signature of N, and is given a type which roughly shows how N is obtained. We call $\mathcal{D}(M, \Gamma)$ the *component tree* of M associated with Γ (see Figure 8 for an example). We construct $\mathcal{D}(M, \Gamma)$ by induction on the total number of vertices and edges in M as follows.

In the base case, $j = i$ and M has only one vertex; we let $\mathcal{D}(M, \Gamma)$ have only one node, label the node with the signature of M, and call it a *type-0* node.

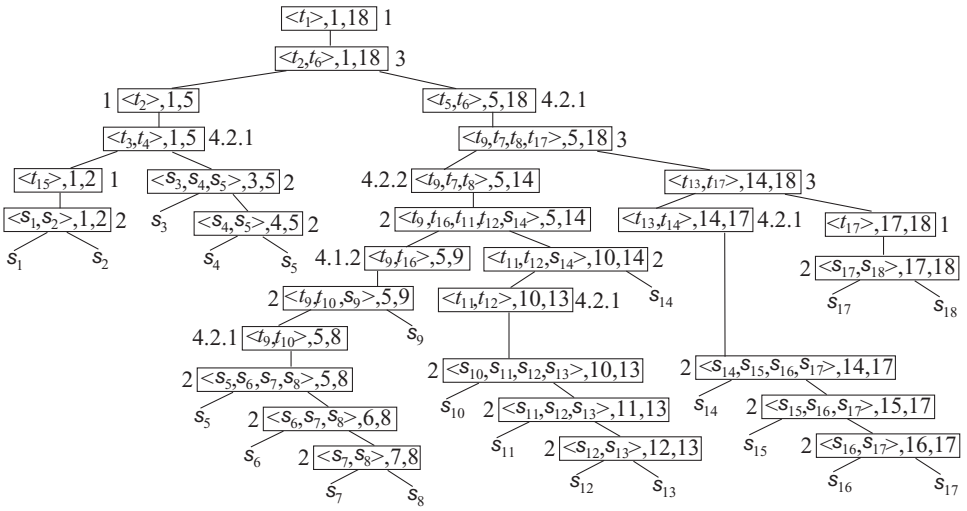

Figure 8: The component tree of the duplication model M in Figure 2 associated with $\Gamma = \{\ell_2, \ell_3, \ell_5, \ell_7, \ell_9, \ell_{10}, \ell_{12}, \ell_{14}, \ell_{15}, \ell_{17}\}$, where the type of each node is given near the node, the weight of each edge is omitted, and the label $(\langle s_i \rangle, i, i)$ of each leaf is simplified to s_i.

Suppose that M has two or more vertices. Then, depending on how many roots are in M and whether they cross each other in M, we distinguish four cases. In each case, we first create a root node α for $\mathcal{D}(M, \Gamma)$ and label it with the signature of M. Then, we proceed to grow $\mathcal{D}(M, \Gamma)$ in each case as follows:

Case 1: M has only one root. Let u be the root of M. Consider the root-marked 2-root model N for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ obtained from M by omitting its root. We construct $\mathcal{D}(N, \Gamma)$ recursively, then let the root of $\mathcal{D}(N, \Gamma)$ be the unique child of α, and further let the weight of the edge between α and its child be $d\,(s(u), s(L_M(u))) + d\,(s(u), s(R_M(u)))$. We also call α a *type-1* node.

Case 2: M has two or more roots and there are two consecutive roots which do not cross each other in M. Let $v_1, \ldots, v_k$ be the roots in M (ordered from left to right). Let h be the smallest integer in $\{1, \ldots, k-1\}$ such that v_h does not cross v_{h+1} in M. There is an integer $b \in \{i, i+1, \ldots, j-1\}$ such that ℓ_i through ℓ_b are the leaf descendants of v_1 through v_h in M while ℓ_{b+1} through ℓ_j are the leaf descendants of v_{h+1} through v_k in M. Let M_1 be the root-marked multiroot model for $\langle s_i, s_{i+1}, \ldots, s_b \rangle$ obtained from M by deleting the descendants of v_{h+1} through v_k in M. Similarly, let M_2 be the root-marked model for $\langle s_{b+1}, s_{b+2}, \ldots, s_j \rangle$ obtained from M by deleting the descendants of v_1 through v_h in M. For each $h \in \{1, 2\}$, let Γ_h be the set of those $v \in \Gamma$ such that v is also a leaf in M_h. We construct $\mathcal{D}(M_1, \Gamma_1)$ and $\mathcal{D}(M_2, \Gamma_2)$ recursively, then let the root of $\mathcal{D}(M_1, \Gamma_1)$ be the left child of α while let the root of $\mathcal{D}(M_2, \Gamma_2)$ be the right child of α, and further let the weight of each edge between α and its child be 0. We also call α a *type-2* node.

Case 3: M has two or more roots, every two consecutive roots in M cross each other, and there is a splitting vertex v in M such that $\overrightarrow{D_M}(v)$ ends at a leaf vertex in Γ. Let v be the leftmost splitting vertex in M such that $\overrightarrow{D_M}(v)$ ends at a leaf in Γ. Let ℓ_b be the leaf at which $\overrightarrow{D_M}(v)$ ends. Let $\widetilde{M}_l$ and $\widetilde{M}_r$ be the left and the right root-marked multiroot models obtained by splitting M along $\overrightarrow{D_M}(v)$, respectively. We call $\overrightarrow{D_M}(v)$ a *splitting path*. Let Γ_l (respectively, Γ_r) be the set of those $v \in \Gamma$ such that v is also a leaf in $\widetilde{M}_l$ (respectively, $\widetilde{M}_r$). We construct $\mathcal{D}(\widetilde{M}_l, \Gamma_l)$ and

$\mathcal{D}(\widetilde{M}_r, \Gamma_r)$ recursively, then let the root of $\mathcal{D}(\widetilde{M}_l, \Gamma_l)$ be the left child of α while let the root of $\mathcal{D}(\widetilde{M}_r, \Gamma_r)$ be the right child of α, and further let the weight of each edge between α and its child be $\frac{1}{2}d(s_b, s(v))$. Note that v is either the rightmost root in $\widetilde{M}_l$ or the leftmost root in $\widetilde{M}_r$. In the former case, we call α a *type-3.1* node while in the latter case, we call α a *type-3.2* node.

Case 4: M has two or more roots, every two consecutive roots in M cross each other, and there is no splitting vertex v in M such that $\overrightarrow{D_M}(v)$ ends at a leaf in Γ. By Lemma 4.8, M has either two or three roots. We distinguish two subcases as follows:

Subcase 4.1: M has a root contained in no 2-block in M. In this subcase, we can use Lemma 4.7 to show that exactly one root u in M is not contained in a 2-block in M. Consider the root-marked multiroot model N for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ obtained from M by omitting u. We construct $\mathcal{D}(N, \Gamma)$ recursively, then let the root of $\mathcal{D}(N, \Gamma)$ be the unique child of α, and further let the weight of the edge between α and its child be $d\left(s(u), s(L_M(u))\right) + d\left(s(u), s(R_M(u))\right)$. We also call α a *type-4.1.h* node if u is the hth leftmost root in M. Note that $1 \leq h \leq 3$.

Subcase 4.2: Every root in M is contained in a 2-block in M. In this subcase, there is a unique pair (v_1, v_2) of consecutive roots in M such that some 2-block in M contains both v_1 and v_2. Consider the root-marked multiroot model N for $\langle s_i, s_{i+1}, \ldots, s_j \rangle$ obtained from M by omitting v_1 and v_2. We construct $\mathcal{D}(N, \Gamma)$ recursively, then let the root of $\mathcal{D}(N, \Gamma)$ be the unique child of α, and further let the weight of the edge between α and its child be $\sum_{h=1}^{2}\left(d\left(s(v_h), s(L_M(v_h))\right) + d\left(s(v_h), s(R_M(v_h))\right)\right)$. We also call α a *type-4.2.h* node if v_1 is the hth leftmost root in M. Note that $1 \leq h \leq 2$.

Fact 6.1 *The component tree $\mathcal{D}(M, \Gamma)$ of M is unique.*

PROOF. Immediate from the construction of $\mathcal{D}(M, \Gamma)$ from M. □

We use $c(\mathcal{D}(M, \Gamma))$ to denote the total weight of edges in $\mathcal{D}(M, \Gamma)$.

Lemma 6.2 $c(\mathcal{D}(M, \Gamma)) \leq c(M) + 2\sum_{u \in \Gamma - \{\ell_i, \ell_j\}} c(\overleftarrow{U_M}(u))$.

PROOF. By induction on the total number of vertices and edges in M. The proof is in parallel with the construction of $\mathcal{D}(M, \Gamma)$. So, we will inherit the notations used in the construction.

The base case corresponds to the base case in the construction of $\mathcal{D}(M, \Gamma)$. In this case, the lemma is clearly true because $c(M) = 0$ and $c(\mathcal{D}(M, \Gamma)) = 0$. So, suppose that M has at least two vertices. Then, the root α of $\mathcal{D}(M, \Gamma)$ may have one or two children.

Case I: α has only one child in $\mathcal{D}(M, \Gamma)$. This case corresponds to Case 1, 4.1, or 4.2 in the construction of $\mathcal{D}(M, \Gamma)$. By inspecting these cases, one can easily see that $c(M) = c(N) + w$ and $c(\mathcal{D}(M, \Gamma)) = c(\mathcal{D}(N, \Gamma)) + w$, where w is the weight assigned to the edge between α and its unique child in $\mathcal{D}(M, \Gamma)$. So, by the inductive hypothesis,

$$c(\mathcal{D}(M, \Gamma)) = w + c(\mathcal{D}(N, \Gamma)) \leq w + c(N) + 2 \cdot \sum_{u \in \Gamma - \{\ell_i, \ell_j\}} c(\overleftarrow{U_N}(u)) = c(M) + 2 \cdot \sum_{u \in \Gamma - \{\ell_i, \ell_j\}} c(\overleftarrow{U_N}(u)).$$

For each $u \in \Gamma - \{\ell_i, \ell_j\}$, $c(\overleftarrow{U_N}(u)) \leq c(\overleftarrow{U_M}(u))$ because $\overleftarrow{U_N}(u)$ is clearly a subpath of $\overleftarrow{U_M}(u)$. Thus, $c(\mathcal{D}(M, \Gamma)) \leq c(M) + 2\sum_{u \in \Gamma - \{\ell_i, \ell_j\}} c(\overleftarrow{U_M}(u))$.

Case II: α has two children in $\mathcal{D}(M, \Gamma)$. This case corresponds to Case 2 or 3 in the construction of $\mathcal{D}(M, \Gamma)$. By inspecting Case 2, $c(\mathcal{D}(M, \Gamma)) = c(\mathcal{D}(M_1, \Gamma_1)) + c(\mathcal{D}(M_2, \Gamma_2) \leq c(M_1) + 2\sum_{u \in \Gamma_1 - \{\ell_i, \ell_b\}} c(\overleftarrow{U_{M_1}}(u)) + c(M_2) + 2\sum_{u \in \Gamma_2 - \{\ell_{b+1}, \ell_j\}} c(\overleftarrow{U_{M_2}}(u)) \leq c(M) + 2\sum_{u \in \Gamma - \{\ell_i, \ell_j\}} c(\overleftarrow{U_M}(u))$,

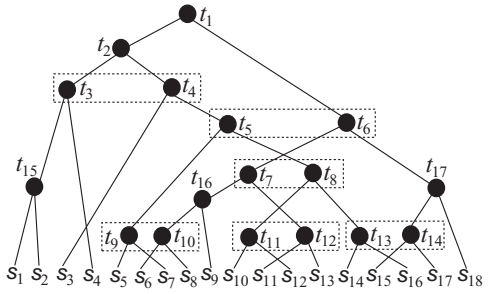

Figure 9: The new duplication model M' constructed from the component tree in Figure 8.

where the first inequality follows from the inductive hypothesis and the second follows from the fact that $\Gamma = \Gamma_1 \cup \Gamma_2$, $\Gamma_1 \cap \Gamma_2 = \emptyset$, and $c(\overleftarrow{U_{M_h}}(u)) = c(\overleftarrow{U_M}(u))$ for each $h \in \{1, 2\}$ and each $u \in \Gamma_h$.

Next consider Case 3. Obviously, $c(\mathcal{D}(M, \Gamma)) = d(s_b, s(v)) + c(\mathcal{D}(\widetilde{M}_l, \Gamma_l)) + c(\mathcal{D}(\widetilde{M}_r, \Gamma_r)$. So, by the inductive hypothesis, $c(\mathcal{D}(M, \Gamma)) \leq d(s_b, s(v)) + c(\widetilde{M}_l) + 2\sum_{u \in \Gamma_l - \{\ell_i, \ell_b\}} c(\overleftarrow{U_{\widetilde{M}_l}}(u)) + c(\widetilde{M}_r) + 2\sum_{u \in \Gamma_r - \{\ell_b, \ell_j\}} c(\overleftarrow{U_{\widetilde{M}_r}}(u))$. Moreover, by Lemma 4.4 and the fact that $\ell_b \in \Gamma - \{\ell_i, \ell_j\}$, we have

$$\sum_{u \in \Gamma_l - \{\ell_i, \ell_b\}} c(\overleftarrow{U_{\widetilde{M}_l}}(u)) + \sum_{u \in \Gamma_r - \{\ell_b, \ell_j\}} c(\overleftarrow{U_{\widetilde{M}_r}}(u)) = \sum_{u \in \Gamma - \{\ell_i, \ell_j\}} c(\overleftarrow{U_M}(u)) - c(\overleftarrow{U_M}(\ell_b)).$$

Hence, by Lemma 4.5, $c(\mathcal{D}(M, \Gamma)) \leq c(M) + 2\sum_{u \in \Gamma - \{\ell_i, \ell_j\}} c(\overleftarrow{U_M}(u))$. □

Lemma 6.2 implies the following corollary immediately:

Corollary 6.3 *If Γ is a 0.75-separator of M, then $c(\mathcal{D}(M, \Gamma)) \leq 2.5 \cdot c(M)$.*

7 Constructing Models from Component Trees

We inherit the notations in Sections 2 and 6. Recall that the label of each node β in $\mathcal{D}(M, \Gamma)$ is a triple $(\mathcal{S}, i, j)$, where $\mathcal{S}$ is an ordered nonempty list of at most five (possibly not distinct) strings and i and j are two integers with $1 \leq i \leq j \leq n$. For convenience, we call $\mathcal{S}$ the *string list* of β.

We show how to use $\mathcal{D}(M, \Gamma)$ to construct a duplication model M' for $\langle s_1, s_2, \ldots, s_n \rangle$ such that $c(M') \leq c(\mathcal{D}(M, \Gamma))$ (see Figure 9 for an example). In the construction of M', we will only use the label and the type of each node in $\mathcal{D}(M, \Gamma)$, i.e., we will not look at the topology of M and will not look at Γ, either.

The construction of M' indeed involves constructing a multiroot model $M'(\beta)$ for each node β of $\mathcal{D}(M, \Gamma)$. We will maintain the invariant that $M'(\beta)$ has $|\mathcal{S}|$ roots labeled by the strings in $\mathcal{S}$, where $\mathcal{S}$ is the string list of β.

We next detail the construction of M'. We construct M' by processing the nodes of $\mathcal{D}(M, \Gamma)$ in a bottom-up fashion. We first process each leaf β in $\mathcal{D}(M, \Gamma)$ by constructing $M'(\beta)$ as follows: Create a new vertex and assign it the unique string in the string list of β.

Now, consider the processing of a nonleaf β in $\mathcal{D}(M, \Gamma)$. Let $\gamma_1, \ldots, \gamma_h$ be the children of β in $\mathcal{D}(M, \Gamma)$, where the ordering is from left to right. Suppose that $M'(\gamma_1), \ldots, M'(\gamma_h)$ have been constructed by processing $\gamma_1, \ldots, \gamma_h$. The processing of β will depend on its type. Since the possible types of β one-to-one correspond to the cases in Section 6, we construct $M'(\beta)$ by distinguishing several cases as follows:

Type 1 (cf. Case 1 in Section 6): We create a new root for $M'(\beta)$, assign it the unique string in the string list of β, and connect it to the roots of $M'(\gamma_1)$ by two new edges. Note that the total cost of the two new edges is exactly the weight of the edge between β and γ_1 in $\mathcal{D}(M, \Gamma)$.

Type 2 (cf. Case 2 in Section 6): We just put $M'(\gamma_1)$ on the left of $M'(\gamma_2)$.

Type 3.1 (cf. Case 3 in Section 6): Let u be the parent of the leftmost leaf in $M'(\gamma_2)$. We connect $M'(\gamma_1)$ and $M'(\gamma_2)$ by deleting the left child of u in $M'(\gamma_2)$ and then making the rightmost root of $M'(\gamma_1)$ be the left child of u. Note that the cost of the new edge is exactly the total weight of the edges between β and its children in $\mathcal{D}(M, \Gamma)$.

Type 3.2 (cf. Case 3 in Section 6): Let u be the parent of the rightmost leaf in $M'(\gamma_1)$. We connect $M'(\gamma_1)$ and $M'(\gamma_2)$ by deleting the right child of u in $M'(\gamma_1)$ and then making the leftmost root of $M'(\gamma_2)$ be the right child of u. Note that the cost of the new edge is exactly the total weight of the edges between β and its children in $\mathcal{D}(M, \Gamma)$.

Type 4.1.h (cf. Subcase 4.1 in Section 6): We create a new root for $M'(\beta)$, assign it the hth string in the string list of β, and connect it to the hth and the $(h+1)$st roots of $M'(\gamma_1)$ by two new edges. Note that the total cost of the two new edges is exactly the weight of the edge between β and γ_1 in $\mathcal{D}(M, \Gamma)$.

Type 4.2.h (cf. Subcase 4.2 in Section 6): We create two new roots v_1 and v_2 for $M'(\beta)$, assign v_1 the hth string in the string list of β, assign v_2 the $(h+1)$st string in the string list of β, connect v_1 to the hth and the $(h+2)$nd roots of $M'(\gamma_1)$ by two new edges, and connect v_2 to the $(h+1)$st and the $(h+3)$rd roots of $M'(\gamma_1)$ by two new edges. Note that the total cost of the two new edges is exactly the weight of the edge between β and γ_1 in $\mathcal{D}(M, \Gamma)$.

Lemma 7.1 *A new duplication model M' for $\langle s_1, s_2, \ldots, s_n \rangle$ can be constructed from $\mathcal{D}(M, \Gamma)$ as above in $O(n)$ time. Moreover, $c(M') \leq c(\mathcal{D}(M, \Gamma))$. Consequently, if Γ is a 0.75-separator of M, then $c(M') \leq 2.5 \cdot c(M)$.*

Proof. Immediate from the above construction of M' and Lemma 6.2. □

8 Abstract Component Trees

An obvious but very crucial property in the construction of M' from $\mathcal{D}(M, \Gamma)$ given in Section 7 is that the construction of M' only depends on the label and the type of each node in $\mathcal{D}(M, \Gamma)$. This motivates us to define abstract component trees (independently of duplication models) for $\langle s_1, \ldots, s_n \rangle$ in which each node will have a label and a type just like a node in $\mathcal{D}(M, \Gamma)$.

In order to define abstract component trees for $\langle s_1, \ldots, s_n \rangle$, we need to define several other terms first. Let $\mathcal{S}$ be a set of strings such that $\{s_1, s_2, \ldots, s_n\} \subseteq \mathcal{S}$. An *$\mathcal{S}$-quadruple* is a quadruple $(\mathcal{L}, i, j, t)$, where $\mathcal{L}$ is a nonempty list of at most five strings in $\mathcal{S}$, i and j are two integers with $1 \leq i \leq j \leq n$ and $j - i + 1 \geq |\mathcal{L}|$, and t satisfies the following conditions:

- $t \in \{0, 1, 2, 3.1, 3.2, 4.1.1, 4.1.2, 4.1.3, 4.2.1, 4.2.2\}$.
- If $t = 0$, then $i = j$ and $\mathcal{L} = \{s_i\}$.
- If $t = 1$, then $|\mathcal{L}| = 1$.

- If $t \in \{2, 3.1, 3.2\}$, then $2 \leq |\mathcal{L}| \leq 5$.
- If $t \in \{4.1.1, 4.1.2, 4.1.3, 4.2.1, 4.2.2\}$, then $2 \leq |\mathcal{L}| \leq 3$.

We call t the *type* of the $\mathcal{S}$-quadruple $(\mathcal{L}, i, j, t)$.

For a list $\mathcal{L}$ and an integer $k \geq 1$, let $\mathcal{L}[k]$ denote the kth element in $\mathcal{L}$. Moreover, for two lists $\mathcal{L}_1$ and $\mathcal{L}_2$, let $\mathcal{L}_1 \cdot \mathcal{L}_2$ denote the concatenation of $\mathcal{L}_1$ and $\mathcal{L}_2$. That is, $\mathcal{L}[h] = \mathcal{L}_1[h]$ for each $h \in \{1, \ldots, |\mathcal{L}_1|\}$ and $\mathcal{L}[|\mathcal{L}_1| + h] = \mathcal{L}_2[h]$ for each $h \in \{1, \ldots, |\mathcal{L}_2|\}$.

We define abstract component trees for $\mathcal{S}$-quadruples $(\mathcal{L}, i, j, t)$ by induction on $|\mathcal{L}| + j - i$. In the base case where $|\mathcal{L}| + j - i = 1$, an *abstract component tree* for $(\mathcal{L}, i, j, t)$ is a rooted tree with only one node (the root) which is labeled with $(\mathcal{L}, i, j)$ and is given a type of t.

Consider the case where $|\mathcal{L}| + j - i \geq 2$. An *abstract component tree* for $(\mathcal{L}, i, j, t)$ is a rooted ordered tree $\mathcal{D}$ such that the root α is labeled with $(\mathcal{L}, i, j)$ and is given a type of t, and the following conditions are satisfied:

- If $t = 1$, then α has only one child β in $\mathcal{D}$, the subtree rooted at β in $\mathcal{D}$ is an abstract component tree for some $\mathcal{S}$-quadruple $(\mathcal{L}', i, j, t')$ with $|\mathcal{L}'| = 2$, and the edge (α, β) is given a weight of $d(\mathcal{L}[1], \mathcal{L}'[1]) + d(\mathcal{L}[1], \mathcal{L}'[2])$.
- If $t = 2$, then α has two children β_1 and β_2 in $\mathcal{D}$, the subtree rooted at β_1 in $\mathcal{D}$ is an abstract component tree for some $\mathcal{S}$-quadruple $(\mathcal{L}_1, i, k, t_1)$ with $k < j$, the subtree rooted at β_2 in $\mathcal{D}$ is an abstract component tree for some $\mathcal{S}$-quadruple $(\mathcal{L}_2, k + 1, j, t_2)$, $\mathcal{L} = \mathcal{L}_1 \cdot \mathcal{L}_2$, and both edges (α, β_1) and (α, β_2) are given a weight of 0.
- If $t = 3.1$, then α has two children β_1 and β_2 in $\mathcal{D}$, the subtree rooted at β_1 in $\mathcal{D}$ is an abstract component tree for some $\mathcal{S}$-quadruple $(\mathcal{L}_1, i, k, t_1)$ with $i < k < j$, the subtree rooted at β_2 in $\mathcal{D}$ is an abstract component tree for some $\mathcal{S}$-quadruple $(\mathcal{L}_2, k, j, t_2)$, $\mathcal{L} = \mathcal{L}_1' \cdot \mathcal{L}_2$, and both edges (α, β_1) and (α, β_2) are given a weight of $\frac{1}{2} d(s_k, \mathcal{L}_1[|\mathcal{L}_1|])$, where $\mathcal{L}_1'$ is obtained from $\mathcal{L}_1$ by deleting the last element.
- If $t = 3.2$, then α has two children β_1 and β_2 in $\mathcal{D}$, the subtree rooted at β_1 in $\mathcal{D}$ is an abstract component tree for some $\mathcal{S}$-quadruple $(\mathcal{L}_1, i, k, t_1)$ with $i < k < j$, the subtree rooted at β_2 in $\mathcal{D}$ is an abstract component tree for some $\mathcal{S}$-quadruple $(\mathcal{L}_2, k, j, t_2)$, $\mathcal{L} = \mathcal{L}_1 \cdot \mathcal{L}_2'$, and both edges (α, β_1) and (α, β_2) are given a weight of $\frac{1}{2} d(s_k, \mathcal{L}_2[1])$, where $\mathcal{L}_2'$ is obtained from $\mathcal{L}_2$ by deleting the first element.
- If $t = 4.1.k$ with $k \in \{1, 2, 3\}$, then α has one child β_1 in $\mathcal{D}$, the subtree rooted at β_1 in $\mathcal{D}$ is an abstract component tree for some $\mathcal{S}$-quadruple $(\mathcal{L}_1, i, j, t_1)$ with $|\mathcal{L}_1| = |\mathcal{L}| + 1$, $\mathcal{L}[b] = \mathcal{L}_1[b]$ for each $1 \leq b \leq k - 1$, $\mathcal{L}[b] = \mathcal{L}_1[b + 1]$ for each $k + 1 \leq b \leq |\mathcal{L}|$, and the edge (α, β_1) is given a weight of $d(\mathcal{L}[k], \mathcal{L}_1[k]) + d(\mathcal{L}[k], \mathcal{L}_1[k + 1])$.
- If $t = 4.2.k$ with $k \in \{1, 2\}$, then α has one child β_1 in $\mathcal{D}$, the subtree rooted at β_1 in $\mathcal{D}$ is an abstract component tree for some $\mathcal{S}$-quadruple $(\mathcal{L}_1, i, j, t_1)$ with $|\mathcal{L}_1| = |\mathcal{L}| + 2$, $\mathcal{L}[b] = \mathcal{L}_1[b]$ for each $1 \leq b \leq k - 1$, $\mathcal{L}[b] = \mathcal{L}_1[b + 2]$ for each $k + 2 \leq b \leq |\mathcal{L}|$, and the edge (α, β_1) is given a weight of $\sum_{h=0}^{1} (d(\mathcal{L}[k + h], \mathcal{L}_1[k + h]) + d(\mathcal{L}[k + h], \mathcal{L}_1[k + h + 2]))$.

Let $(\mathcal{L}, i, j, t)$ be an $\mathcal{S}$-quadruple, and let $\mathcal{D}$ be an abstract component tree for $(\mathcal{L}, i, j, t)$. Note that each node in $\mathcal{D}$ is labeled with a triple $(\mathcal{L}', h, k)$ and is given a type t', where $\mathcal{L}'$ is a nonempty list of strings in $\mathcal{S}$, h and k are integers with $1 \leq h \leq k \leq n$, and $t \in \{0, 1, 2, 3.1, 3.2, 4.1.1, 4.1.2, 4.1.3, 4.2.1, 4.2.2\}$. Thus, we can use $\mathcal{D}$ to construct a multiroot model $M'_{\mathcal{D}}$ with $|\mathcal{L}|$ roots as described in Section 7. We define the *weight* of $\mathcal{D}$ to be the total weight of its edges. An abstract component tree for $(\mathcal{L}, i, j, t)$ is *optimal* if its weight is minimized over all abstract component trees for $(\mathcal{L}, i, j, t)$.

An *$\mathcal{S}$-abstract component tree* for $\langle s_1, s_2, \ldots, s_n \rangle$ is an abstract component tree for some $\mathcal{S}$-quadruple $(\mathcal{L}, 1, n, 1)$. An $\mathcal{S}$-abstract component tree for $\langle s_1, s_2, \ldots, s_n \rangle$ is *optimal* if its weight is minimized over all $\mathcal{S}$-abstract component trees for $\langle s_1, s_2, \ldots, s_n \rangle$.

9 Computing an Optimal $\mathcal{S}$-Abstract Component Tree

We now use dynamic programming to compute an optimal abstract component tree for each $\mathcal{S}$-quadruple $(\mathcal{L}, i, j, t)$. For simplicity, we only explicitly give formulas for computing the minimum weight $W(\mathcal{L}, i, j, t)$ of an abstract component tree for each $\mathcal{S}$-quadruple $(\mathcal{L}, i, j, t)$ as follows.

- For each $\mathcal{S}$-quadruple $q = (\mathcal{L}, i, j, t)$ with $t = 0$, $W(q) = 0$.
- For each $\mathcal{S}$-quadruple $q = (\mathcal{L}, i, j, t)$ with $t = 1$,

  $$W(q) = \min_{q'} \; W(q') + d(\mathcal{L}[1], \mathcal{L}'[1]) + d(\mathcal{L}[1], \mathcal{L}'[2]),$$

  where q' ranges over all $\mathcal{S}$-quadruples $(\mathcal{L}', i, j, t')$ with $|\mathcal{L}'| = 2$.
- For each $\mathcal{S}$-quadruple $q = (\mathcal{L}, i, j, t)$ with $t = 2$,

  $$W(q) = \min_{i \leq k < j} \; \min_{q_1} \; \min_{q_2} \; W(q_1) + W(q_2),$$

  where q_1 ranges over all $\mathcal{S}$-quadruples $(\mathcal{L}_1, i, k, t_1)$ and q_2 ranges over all $\mathcal{S}$-quadruples $(\mathcal{L}_2, k+1, j, t_2)$ such that $\mathcal{L} = \mathcal{L}_1 \cdot \mathcal{L}_2$.
- For each $\mathcal{S}$-quadruple $q = (\mathcal{L}, i, j, t)$ with $t = 3.1$,

  $$W(q) = \min_{i < k < j} \; \min_{q_1} \; \min_{q_2} \; W(q_1) + W(q_2) + d(s_k, \mathcal{L}_1[|\mathcal{L}_1|]),$$

  where q_1 ranges over all $\mathcal{S}$-quadruples $(\mathcal{L}_1, i, k, t_1)$ and q_2 ranges over all $\mathcal{S}$-quadruples $(\mathcal{L}_2, k, j, t_2)$ such that $\mathcal{L} = \mathcal{L}'_1 \cdot \mathcal{L}_2$ and $\mathcal{L}'_1$ is obtained from $\mathcal{L}_1$ by deleting the last element.
- For each $\mathcal{S}$-quadruple $q = (\mathcal{L}, i, j, t)$ with $t = 3.2$,

  $$W(q) = \min_{i < k < j} \; \min_{q_1} \; \min_{q_2} \; W(q_1) + W(q_2) + d(s_k, \mathcal{L}_2[1]),$$

  where q_1 ranges over all $\mathcal{S}$-quadruples $(\mathcal{L}_1, i, k, t_1)$ and q_2 ranges over all $\mathcal{S}$-quadruples $(\mathcal{L}_2, k, j, t_2)$ such that $\mathcal{L} = \mathcal{L}_1 \cdot \mathcal{L}'_2$ and $\mathcal{L}'_2$ is obtained from $\mathcal{L}_2$ by deleting the first element.

- For each $\mathcal{S}$-quadruple $q = (\mathcal{L}, i, j, t)$ with $t = 4.1.k$ and $k \in \{1, 2, 3\}$,

$$W(q) = \min_{q_1} W(q_1) + d(\mathcal{L}[k], \mathcal{L}_1[k]) + d(\mathcal{L}[k], \mathcal{L}_1[k+1]),$$

  where q_1 ranges over all $\mathcal{S}$-quadruples $(\mathcal{L}_1, i, j, t_1)$ such that $|\mathcal{L}_1| = |\mathcal{L}| + 1$, $\mathcal{L}[b] = \mathcal{L}_1[b]$ for each $1 \leq b \leq k-1$, and $\mathcal{L}[b] = \mathcal{L}_1[b+1]$ for each $k+1 \leq b \leq |\mathcal{L}|$.

- For each $\mathcal{S}$-quadruple $q = (\mathcal{L}, i, j, t)$ with $t = 4.2.k$ for some $k \in \{1, 2\}$,

$$W(q) = \min_{q_1} W(q_1) + \sum_{h=0}^{1} \left(d(\mathcal{L}[k+h], \mathcal{L}_1[k+h]) + d(\mathcal{L}[k+h], \mathcal{L}_1[k+h+2]) \right),$$

  where q_1 ranges over all $\mathcal{S}$-quadruples $(\mathcal{L}_1, i, j, t_1)$ such that $|\mathcal{L}_1| = |\mathcal{L}| + 2$, $\mathcal{L}[b] = \mathcal{L}_1[b]$ for each $1 \leq b \leq k-1$, and $\mathcal{L}[b] = \mathcal{L}_1[b+2]$ for each $k+2 \leq b \leq |\mathcal{L}|$.

Clearly, the weight of an optimal $\mathcal{S}$-abstract component tree for $\langle s_1, \ldots, s_n \rangle$ is $\min_q W(q)$, where q ranges over all $\mathcal{S}$-quadruples $(\mathcal{L}, 1, n, 1)$. Moreover, the total time needed for finding an optimal $\mathcal{S}$-abstract component tree for $\langle s_1, s_2, \ldots, s_n \rangle$ is $O(|\mathcal{S}|^6 n^3 + |\mathcal{S}|^7 n^2 + |\mathcal{S}|^2 m)$. Since $|\mathcal{S}| \geq n$, the total time needed is $O(|\mathcal{S}|^7 n^2 + |\mathcal{S}|^2 m)$. Thus, we have the following lemma:

Lemma 9.1 *Given a set $\mathcal{S}$ with $\{s_1, \ldots, s_n\} \subseteq \mathcal{S}$, we can compute an optimal $\mathcal{S}$-abstract component tree for $\langle s_1, \ldots, s_n \rangle$ in $O(|\mathcal{S}|^7 n^2 + |\mathcal{S}|^2 m)$ time.*

10 A Ratio-5 Approximation Algorithm

In the remainder of this paper, let M_{opt} be an optimal duplication model for $\langle s_1, s_2, \ldots, s_n \rangle$.

By Lemmas 7.1 and 9.1, if we know the set $\mathcal{S}$ of strings assigned to the vertices of an optimal duplication model for $\langle s_1, s_2, \ldots, s_n \rangle$, then we would have obtained an approximation algorithm for 2-DHR which achieves a ratio of 2.5 and runs in $O(n^9 + n^2 m)$ time.

Unfortunately, it seems difficult to know the set of strings assigned to the vertices of an optimal duplication model for $\langle s_1, s_2, \ldots, s_n \rangle$. As suggested in [2], one idea to get around this difficulty is to look for a restricted type of duplication models called *lifted* duplication models. In a lifted duplication model, the label assigned to each nonleaf is a string in $\{s_1, \ldots, s_n\}$. Based on a result in [10], the following lemma has been proved in [2]:

Lemma 10.1 *There is a lifted duplication model M for $\langle s_1, \ldots, s_n \rangle$ with $c(M) \leq 2 \cdot c(M_{opt})$.*

By Lemmas 7.1, 9.1, and 10.1, we can construct a duplication model N for $\langle s_1, s_2, \ldots, s_n \rangle$ with $c(N) \leq 5 \cdot c(M_{opt})$ as follows:

1. Let $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ (cf. Lemma 10.1).
2. Compute an optimal $\mathcal{S}$-abstract component tree $\mathcal{D}$ for $\langle s_1, s_2, \ldots, s_n \rangle$ (cf. Lemma 9.1).
3. Use $\mathcal{D}$ to construct a duplication model N for $\langle s_1, s_2, \ldots, s_n \rangle$ (cf. Lemma 7.1).

Theorem 10.2 *There is an approximation algorithm for 2-DHR that achieves a ratio of 5 and runs in $O(n^9 + n^2 m)$ time.*

11 A Ratio-$(2.5+\epsilon)$ Approximation Algorithm

To achieve an approximation ratio better than 5, we cannot restrict our attention to lifted duplication models. In other words, we cannot require that all strings assigned to vertices of a duplication model for $\langle s_1, \ldots, s_n \rangle$ be a string in $\{s_1, \ldots, s_n\}$. Instead, we just require that only a constant fraction of strings assigned to vertices of a duplication model for $\langle s_1, \ldots, s_n \rangle$ be a string in $\{s_1, \ldots, s_n\}$. We detail the idea below.

Suppose that T is a phylogeny for a permutation $\langle s_{i_1}, s_{i_2}, \ldots, s_{i_n} \rangle$ of $\langle s_1, s_2, \ldots, s_n \rangle$. A vertex u of T is *lifted* if the string assigned to u in T is the same as the string assigned to some leaf descendant of u in T. If a vertex of T is not lifted then it is *free*. By default, a leaf is a lifted vertex. A *lifted component* of T is a maximal subtree C of T such that

- the root and the leaves of C are lifted vertices of T while the other vertices of C are free vertices of T, and
- the root of C has one child in C.

Note that each nonleaf of a lifted component C other than the root of C has exactly two children in C. It is also clear that no two lifted components of T share an edge. For an integer $r \geq 2$, T is *r-lifted* if each lifted component of T has no more than $r-1$ leaves. The following lemma has been proved in [11]:

Lemma 11.1 *For every integer $t \geq 2$ and every phylogeny T for a list $\mathcal{L}$ of strings, there is a $(2^{t-1}+1)$-lifted phylogeny T' for $\mathcal{L}$ such that $c(T') \leq \left(1+\frac{2}{t+1}\right) \cdot c(T)$ and the topology of T' is the same as that of T.*

For convenience, we define a *partially labeled semi-binary tree* to be a rooted tree C satisfying the following conditions:

- The root of C has only one child while each nonleaf of C other than the root has two children (the left and the right children).
- The root of C is assigned a string in $\{s_1, \ldots, s_n\}$ and so is every leaf of C.
- No string is assigned to a nonleaf of C other than the root.

Let C be a partially labeled semi-binary tree. *Fully labeling* C is the operation of assigning one string of length m to each nonleaf of C other than the root of C. *Optimally fully labeling* C is to fully label C so that the cost of the resulting tree C_{opt} is minimized over all trees that can be obtained by fully labeling C.

Lemma 11.2 *For every constant $\epsilon > 0$, we can compute a set $\mathcal{S}$ of $O(n^{4^{1/\epsilon}})$ strings in $O(mn^{4^{1/\epsilon}})$ time such that there is a duplication model M for $\langle s_1, s_2, \ldots, s_n \rangle$ such that $c(M) \leq (1+\epsilon) \cdot c(M_{opt})$ and each string assigned to a vertex of M is in $\mathcal{S}$.*

Proof. Fix a constant $\epsilon > 0$. Let $t = \lceil \frac{2}{\epsilon} \rceil - 1$ and $r = 2^{t-1}+1$. Obviously, there are only $O(n^r)$ partially labeled semi-binary trees with at most $r-1$ leaves each. We compute the required set $\mathcal{S}$ of strings as follows:

1. For each partially labeled semi-binary tree C with at most $r-1$ leaves, optimally fully label C to obtain C_{opt}. (*Comment:* C_{opt} can be obtained from C in constant time via dynamic programming because C has only a constant number of vertices.)

2. Let $\mathcal{S}$ be the set of all strings assigned to vertices of the trees C_{opt}.

Obviously, $\mathcal{S}$ contains $O(n^r)$ strings and can be computed in $O(n^r)$ time. It remains to show that there is a duplication model M for $\langle s_1, s_2, \ldots, s_n \rangle$ such that $c(M) \leq (1+\epsilon) \cdot c(M_{opt})$ and each string assigned to a vertex of M is in $\mathcal{S}$.

Let T be the associated phylogeny for M_{opt}. Clearly, $c(M_{opt}) = c(T)$. For each $j \in \{1, \ldots, n\}$, let s_{i_j} be the string assigned to the jth leftmost leaf in T. Note that $\langle s_{i_1}, s_{i_2}, \ldots, s_{i_n} \rangle$ is a permutation of $\langle s_1, s_2, \ldots, s_n \rangle$. By Lemma 11.1, there is an r-lifted phylogeny T' for $\langle s_{i_1}, s_{i_2}, \ldots, s_{i_n} \rangle$ such that $c(T') \leq (1+\epsilon) \cdot c(T)$ and the topology of T' is the same as that of T.

For each lifted component C of T', if we ignore the strings assigned to the nonleaves of C other than the root of C, then we obtain a partially labeled semi-binary tree with at most $r-1$ leaves and so we must have optimally fully labeled it to obtain a tree C_{opt} in Step 1 above (when computing $\mathcal{S}$). The crucial point is that modifying T' by replacing C with C_{opt} neither changes the topology of T' nor increases the cost of T'. Suppose that we modify T' by replacing every lifted component C of T' with C_{opt}. Then, $c(T') \leq (1+\epsilon) \cdot c(T)$, each string assigned to a vertex of T' is in $\mathcal{S}$, the topology of T' is the same as that of T, and the jth leftmost leaf is assigned s_{i_j} for each $j \in \{1, \ldots, n\}$. Now, since the vertices of T' one-to-one correspond to those of T and the vertices of T one-to-one correspond to those of M_{opt}, we can obtain a new duplication model M for $\langle s_1, \ldots, s_n \rangle$ from M_{opt} by simply changing the label of each vertex in M_{opt} to that of the corresponding vertex in T'. Obviously, $c(M) \leq (1+\epsilon) \cdot c(M_{opt})$ and each string assigned to a vertex of M is in $\mathcal{S}$. □

By Lemmas 7.1, 9.1, and 11.2, we can construct a duplication model N for $\langle s_1, s_2, \ldots, s_n \rangle$ with $c(N) \leq (2.5+\epsilon) \cdot c(M_{opt})$ as follows:

1. Compute $\mathcal{S}$ as in Lemma 11.2.

2. Compute an optimal $\mathcal{S}$-abstract component tree $\mathcal{D}$ for $\langle s_1, s_2, \ldots, s_n \rangle$ (cf. Lemma 9.1).

3. Use $\mathcal{D}$ to construct a duplication model N for $\langle s_1, s_2, \ldots, s_n \rangle$ (cf. Lemma 7.1).

Theorem 11.3 *For any constant $\epsilon > 0$, there is an approximation algorithm for 2-DHR that achieves a ratio of $(2.5+\epsilon)$ and runs in $O(n^{2+7 \cdot 4^{1/\epsilon}} + mn^{2 \cdot 4^{1/\epsilon}})$ time.*

12 Concluding Remarks

The results presented in this paper are of purely theoretical interest. The ratio-5 approximation algorithm takes $O(n^9)$ time which is too high for large n. The running time of the other algorithm is even worse so that it is impossible to implement the algorithm even for small n. It is of interest to reduce the time complexity of the algorithms.

We conjecture that there is a PTAS for 2-DHR. Indeed, we suspect that the ideas presented in this paper will give some insight for finding such a PTAS.

Acknowledgment

Zhi-Zhong Chen is supported in part by the Grant-in-Aid for Scientific Research of the Ministry of Education, Science, Sports and Culture of Japan, under Grant No. 20500021. Lusheng Wang is fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 120905].

References

[1] G. Benson and L. Dong, "Reconstructing the Duplication History of a Tandem Repeat," Proc. The 7th International Conference on Intelligent Systems for Molecular Biology (ISMB '99), pp. 44-53, 1999.

[2] Z.-Z. Chen, L. Wang, and Z. Wang, "Approximation Algorithms for Reconstructing the Duplication History of Tandem Repeats," Algorithmica, to appear. A preliminary version appeared in "Proc. The 13th Annual International Computing and Combinatorics Conference (COCOON '07), " Lecture Notes in Computer Science, vol. 4598, pp. 493-503, 2007.

[3] W. Fitch, "Phylogenies Constrained by Cross-over Process as Illustrated by Human Hemoglobins in a Thirteen Cycle, Eleven Amino-Acid Repeat in Human Apolipoprotein A-I," Genetics, vol. 86, pp. 623-644, 1977.

[4] O. Gascuel, D. Bertrand, and O. Elemento, "Reconstructing the Duplication History of Tandemly Repeated Sequences," in *Mathematics of Evolution and Phylogeny*, pp. 205-235, Oxford University Press, New York, 2005.

[5] D. Jaitly, P. E. Kearney, G. Lin, and B. Ma, "Methods for Reconstructing the History of Tandem Repeats and Their Application to the Human Genome," J. Comput. Syst. Sci., vol. 65 , pp. 494-507, 2002.

[6] F. Lillo, S. Basile, and R. N. Mantegna, "Comparative Genomics Study of Inverted Repeats in Bacteria," Bioinformatics, vol. 18, pp. 971 - 979, 2002.

[7] J. Macas, T. Mszros, and M. Nouzov, "PlantSat: A Specialized Database for Plant Satellite Repeats," Bioinformatics, vol. 18, pp. 28 - 35, 2002.

[8] H. H. Otu and K. Sayood, "A New Sequence Distance Measure for Phylogenetic Tree Construction," Bioinformatics, vol. 19, pp. 2122 - 2130, 2004.

[9] M. Tang, M. S. Waterman, and S. Yooseph, "Zinc Finger Gene Clusters and Tandem Gene Duplication," Journal of Computational Biology, vol. 9, pp. 429-446, 2002.

[10] L. Wang, T. Jiang, and E. L. Lawler, "Approximation Algorithms for Tree Alignment with a Given Phylogeny," Algorithmica, vol. 16, pp. 302-315, 1996.

[11] L. Wang, T. Jiang, and D. Gusfield, "A More Efficient Approximation Scheme for Tree Alignment," SIAM Journal on Computing, vol. 30, pp. 283-299, 2000.

[12] L. Zhang, B. Ma, L. Wang, and Y. Xu, "Greedy Method for Inferring Tandem Duplication History," Bioinformatics, vol. 19, pp. 1497-1504, 2003.

13 List of Notations

- $c(M)$: Cost of model M (cf. Page 3).
- $c(P)$: Cost of path P in M (cf. Page 13).
- $c(\mathcal{D}(M,\Gamma))$: Total weight of edges in component tree $\mathcal{D}(M,\Gamma)$ (cf. Page 19).
- $d(s',s'')$: Hamming distance between strings s' and s'' (cf. Page 17).
- $\mathcal{D}(M,\Gamma)$: Component tree of model M associated with separator Γ (cf. Page 17).
- $D_M(u)$: The child of vertex u to which we move when going down from u (cf. Page 8).
- $\overrightarrow{D_M}(u)$: The path we trace when going down from vertex u to a leaf guided by function D_M (cf. Page 9).
- $I_M(u)$: The $I_M(u)$-th input string $s_{I_M(u)}$ is the leftmost leaf descendant of u in model M (cf. Page 6).
- $J_M(u)$: The $J_M(u)$-th input string $s_{J_M(u)}$ is the rightmost leaf descendant of u in model M (cf. Page 6).
- ℓ_i: The leaf of model M labeled with the ith input string s_i (cf. Page 5).
- $L_M(u)$: The left child of vertex u in model M (cf. Page 8).
- $\mathcal{L}[i]$: The ith element in list $\mathcal{L}$ (cf. Page 22).
- M_l: The left root-marked multiroot semi-model obtained by splitting model M along path $\overrightarrow{D_M}(v)$ for a chosen splitting vertex v (cf. Page 10).
- $\widetilde{M_l}$: The left root-marked multiroot model obtained by splitting model M along path $\overrightarrow{D_M}(v)$ for a chosen splitting vertex v (cf. Page 10).
- M_{opt}: An optimal duplication model for the input list of strings (cf. Page 24).
- M_r: The right root-marked multiroot semi-model obtained by splitting model M along path $\overrightarrow{D_M}(v)$ for a chosen splitting vertex v (cf. Page 10).
- $\widetilde{M_r}$: The right root-marked multiroot model obtained by splitting model M along path $\overrightarrow{D_M}(v)$ for a chosen splitting vertex v (cf. Page 10).
- $P_M(u)$: The parent of vertex u in model M (cf. Page 8).
- $R_M(u)$: The right child of vertex u in model M (cf. Page 8).
- $s(\mathcal{L})$: The list of strings assigned to vertices u in list $\mathcal{L}$ (cf. Page 17).
- $s(u)$: The string assigned to vertex u in model M (cf. Page 17).
- $U_M(u)$: The parent v of vertex u in model M if $D_M(v)=u$; otherwise, undefined (cf. Page 8).
- $\overleftarrow{U_M}(u)$: The path we trace when going up from leaf u to an ancestor guided by function U_M (cf. Page 9).

14 List of definitions