A decomposition algorithm for noncrossing trees

Lun Lv

School of Sciences
Hebei University of Science and Technology
Shijiazhuang 050018, P.R. China

klunlv@gmail.com

Sabrina X.M. Pang*

College of Mathematics and Statistics
Hebei University of Economics and Business
Shijiazhuang 050061, P.R. China

stpangxingmei@heuet.edu.cn

Submitted: May 7, 2013; Accepted: Dec 29, 2013; Published: Jan 12, 2014
Mathematics Subject Classifications: 05A05, 05A19

Abstract

Based on the classic bijective algorithm for trees due to Chen, we present a decomposition algorithm for noncrossing trees. This leads to a combinatorial interpretation of a formula on noncrossing trees of size n with k descents. We also derive the formula for noncrossing trees of size n with k descents and i leaves, which is a refinement of the formula given by Flajolet and Noy. As an application of our algorithm, we answer a question proposed by Hough, which asks for a bijection between two classes of noncrossing trees with a given number of descents.

Keywords: noncrossing tree, descent, bijection

1 Introduction

A *noncrossing tree* (NC-tree for short) is a tree drawn on n points numbered in counterclockwise order on a circle in such a way that its edges are rectilinear and do not cross. We always consider the points labeled counterclockwise from 1 to n and the root labeled 1. The *size* of a NC-tree is defined as the number of edges. It is well known that the number of NC-trees of size n equals $\frac{1}{2n+1}\binom{3n}{n}$, the generalized Catalan number. Noncrossing configurations have been extensively studied, see, for example, [1, 3, 5–11, 13].

*Corresponding author

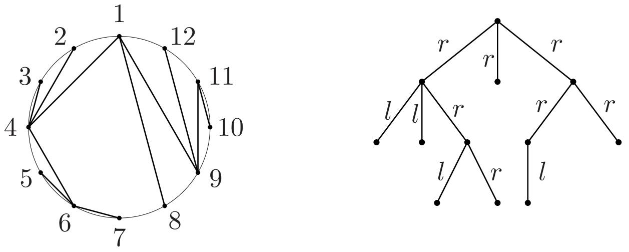

Figure 1: The $(l;r)$-representation of a NC-tree.

In a NC-tree T, a vertex v is *planted* at u if (u,v) is an edge and u lies on the unique path from the root to v. Moreover, (u,v) is a *descent* if $u>v$. Denote $N_{n,k}$ the number of NC-trees of size n with k descents. Using generating function, Hough [8] pointed out the following formula

Theorem 1.1 (Theorem 2.2, [8]).

$$N_{n,k}=\frac{1}{n}\binom{n-1+k}{n-1}\binom{2n-k}{n+1}. \tag{1.1}$$

Hough [8] also observed that $N_{2k+1,k-1}=N_{2k+1,k}$, and proposed the following question.

Question 1.2 (Question 4.2, [8]). *Find a bijection on NC-trees with $2k+1$ edges between those with $k-1$ descents and those with k descents.*

In this paper, we give a combinatorial interpretation of formula (1.1) by introducing a decomposition algorithm for NC-trees. Moreover, we derive a refined formula on NC-trees of size n with respect to the number of descents and the number of leaves. Notice that this formula is also a refinement of a result given by Flajolet and Noy [6]. The second result of this paper is to provide a bijective proof for Question 1.2.

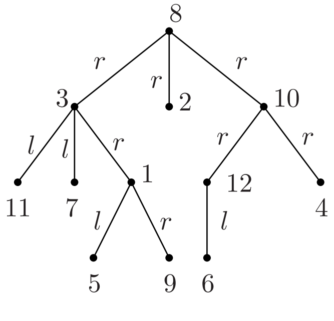

Figure 2: A labeled NC-tree.

Our combinatorial interpretation of formula (1.1) relies on the $(l;r)$-representation of NC-trees introduced by Panholzer and Prodinger [11]. Represent a NC-tree T by a plane tree with each edge labeled by l or r using the following rules: Label an edge by l if it is a descent; Otherwise, label it by r. See Figure 1 for the $(l;r)$-representation of a NC-tree. Briefly, we call an edge labeled by l or r an l-edge or an r-edge respectively. It is obvious that a plane tree obtained in this way has two properties: (i) the edges adjacent to the root are always r-edges; (ii) for any internal vertex v, the vertices planted at v with l-edges are always to the left of the vertices planted at v with r-edges. In addition, these two properties provide a necessary and sufficient condition for a plane tree to become a NC-tree. Note that each l-edge in such a plane tree corresponds to a descent. To construct the decomposition algorithm, we still need the notion of labeled NC-trees. A *labeled NC-tree* of size n is referred to the $(l;r)$-representation of a NC-tree of size n with the label set $[n+1] := \{1, 2, \ldots, n+1\}$. See Figure 2 for a labeled NC-tree. Clearly, the number of labeled NC-trees of size n equals $(n+1)!$ times the number of NC-trees of size n.

2 A decomposition algorithm for labeled NC-trees

The purpose of this section is to give a decomposition algorithm for labeled NC-trees that leads to a combinatorial interpretation of formula (1.1). Our algorithm is a generalization of the bijective algorithm for plane trees in [2]. It should be pointed out that Chen [2] established a more general bijection for trees.

A *planted tree* is a rooted tree with root degree one, and a *planted edge* is the edge adjacent to the root. Define a planted tree r*-unique* if the planted edge is an r-edge and all the other edges are l-edges.

We begin to deal with a labeled NC-tree T of size n with k l-edges. Remind that the label set of T is $[n+1]$. The first procedure of our algorithm is to decompose T into a forest F on $[2n-k]$, where F is composed of $n-k$ r-unique planted trees.

Decomposition algorithm:

Step 1. Find an r-unique planted subtree of T with planted edge (u, v) such that v is the rightmost child of u and the original label i of u is the smallest. Then we may obtain an r-unique planted subtree with planted edge (u, v).

Step 2. Remove the r-unique planted subtree and relabel the vertex u by $n+2$ in T. However, the original label i will be reused for later comparisons of vertices.

Step 3. Repeat the above steps for the remaining tree and subsequently supply $n+3, n+4, \ldots, 2n-k$ to relabel the encountered vertex u.

Figure 3 (a) is an illustration of the first procedure of the decomposition algorithm. The second procedure of our algorithm is to decompose F into a set M of n matches on $[2n]$, where a match is a rooted tree with two vertices.

Step 4. Find an l-edge (u, v) such that v is the rightmost child of u, v is a leaf and the label i of u is the smallest. Then we obtain a match with root u.

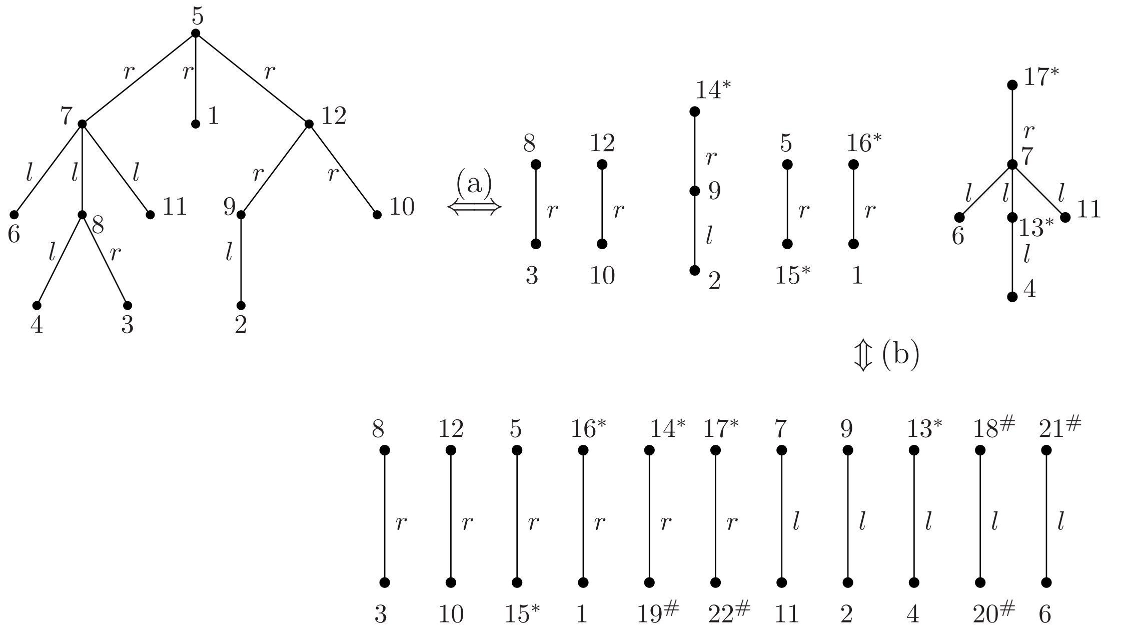

Figure 3: The decomposition algorithm of a labeled NC-tree.

Step 5. Remove the match and relabel the vertex u by $2n - k + 1$ in F. However, the label i is still used for later comparisons of vertices.

Step 6. Repeat Step 4 and Step 5 for the remaining forest and subsequently supply $2n - k + 2, 2n - k + 3, \ldots, 2n$ to relabel the encountered vertex u.

See Figure 3 (b) for an example of the second procedure. It is clear that for $k = 0$, the decomposition algorithm for a labeled NC-tree with k descents reduces to the algorithm for a plane tree introduced by Chen [2].

We call a match labeled by r or l an *r-match* or *l-match* respectively. For $k \geqslant 1$, the resulting n matches on $[2n]$ have the following properties:

(A1) There are k l-matches and $n - k$ r-matches;

(A2) The labels for the roots of r-matches are in the set $[2n - k]$;

(A3) The label $2n$ is a leaf of an r-match.

Let $\mathcal{M}_{n,k} := \{M | M$ is a set of n matches on $[2n]$ satisfying conditions (A1)-(A3)$\}$. Now we are ready to establish the inverse of the decomposition algorithm. Specifically, for $M \in \mathcal{M}_{n,k}$, we need to merge M into a labeled NC-tree.

First, rewrite the label set $[2n]$ of M by the marked set

$$\{1, 2, \ldots, n+1, (n+2)^*, \ldots, (2n-k)^*, (2n-k+1)^\#, \ldots, (2n)^\#\}.$$

A vertex labeled by a mark $*$ or a mark $\#$ is called a $*$-marked or a $\#$-marked vertex respectively.

Merging algorithm:

Step 1. Find the tree T without r-edge and $\#$-marked vertex, such that the root i of T is the smallest.

Step 2. Find the tree $T^{\#}$ with the smallest $\#$-marked vertex. Let $j^{\#}$ be this marked vertex.

Step 3. If $j^{\#}$ is the root of $T^{\#}$, then merge T with $T^{\#}$ by identifying i and $j^{\#}$, put the subtrees of T to the right of $T^{\#}$, and keep i as the label of the combined vertex. This operation is called a *horizontal merge*. If $j^{\#}$ appears as a leaf of $T^{\#}$, then replace $j^{\#}$ by T in $T^{\#}$. This operation is called a *vertical merge*. See Figure 4.

Step 4. Repeat the above steps until no $\#$-marked vertex left.

Step 5. Find the tree T without $*$-marked vertex and the root i of T is the smallest.

Step 6. Find the tree T^* with the smallest $*$-marked vertex j^*. If j^* is the root of T^*, then merge T with T^* by applying the horizontal merge. If j^* appears as a leaf of T^*, then merge T with T^* by applying the vertical merge.

Step 7. Repeat Step 5 and Step 6 until we get a labeled NC-tree.

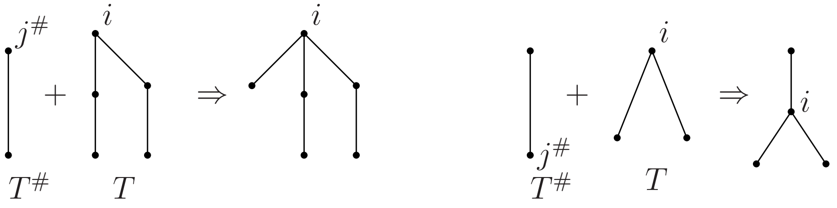

Figure 4: A horizontal merge and a vertical merge.

Relying on the above algorithms, we arrive at the following result.

Theorem 2.1. *The decomposition algorithm and the merging algorithm are inverse to each other. Consequently, for $k \geqslant 1$, there is a bijection between the set of labeled NC-trees of size n with k descents and the set $\mathcal{M}_{n,k}$.*

The proof of Theorem 2.1 is elementary but tedious, and we present it in Appendix A.

Combinatorial proof of Theorem 1.1. A NC-tree of size n with 0 descent reduces to a plane tree with n edges. It is well known that the set of plane trees of n edges is counted by the n-th Catalan number, see, for example, [12]. Hence, the formula (1.1) is clear. We only need to confirm (1.1) for $k \geqslant 1$. Keep in mind that the number of labeled NC-trees of size n with k descents equals $(n+1)!$ times the number of NC-trees of size n with k descents. By Theorem 2.1, it is equivalent to prove the following expression

$$|\mathcal{M}_{n,k}| = (n+1)!N_{n,k} = \frac{(n+1)!}{n}\binom{n-1+k}{n-1}\binom{2n-k}{n+1}. \tag{2.1}$$

Note that the set of matches in $\mathcal{M}_{n,k}$ satisfies conditions (A1)–(A3). Formula (2.1) holds since (i) there are $\binom{2n-k}{n-k}$ ways to choose $n-k$ labels from $[2n-k]$ for the roots of r-matches, (ii) there are $\binom{n-k}{1}$ ways to choose a label to match $2n$ for an r-match, (iii) there are $\binom{n+k-1}{k}$ ways to choose k labels from the remaining label set for the roots of l-matches, and (iv) there are $(n-1)!$ ways to arrange the remaining labels for the leaves of the matches. This gives the number

$$\binom{2n-k}{n-k}\binom{n-k}{1}\binom{n+k-1}{k}(n-1)!,$$

which equals the right hand side of (2.1). □

Observe that in the merging algorithm, a leaf with an unmarked label in M is still a leaf in the corresponding labeled NC-tree. We may derive a refined formula on NC-trees with respect to the number of descents and the number of leaves.

Theorem 2.2. *For $k \geqslant 1$, the number of NC-trees of size n with k descents and i leaves is equal to*

$$\frac{1}{n}\sum_{t=0}^{n-k}\binom{n}{t-1}\binom{n-k}{t}\binom{n+1-t}{i}\binom{k+t-2}{n-1-i}. \tag{2.2}$$

Proof. The proof is similar as Theorem 1.1, and we give a sketch. Note that a labeled NC-tree of size n with k descents and i leaves corresponds to a set of n matches $M \in \mathcal{M}_{n,k}$ with i unmarked leaves. Consider the set of M in $\mathcal{M}_{n,k}$ with exactly i unmarked leaves and t r-matches with unmarked roots, which is counted by

$$\binom{n+1}{t}\binom{n-k-1}{n-k-t}\binom{n-k}{1}\binom{n+1-t}{i}\binom{k+t-2}{n-1-i}(n-1)!.$$

By summing over t, we derive that the number of labeled NC-trees of size n with k descents and i leaves is

$$\begin{aligned}&\sum_{t=0}^{n-k}\binom{n+1}{t}\binom{n-k-1}{n-k-t}\binom{n-k}{1}\binom{n+1-t}{i}\binom{k+t-2}{n-1-i}(n-1)!\\ &= \frac{(n+1)!}{n}\sum_{t=0}^{n-k}\binom{n}{t-1}\binom{n-k}{t}\binom{n+1-t}{i}\binom{k+t-2}{n-1-i}.\end{aligned}$$

This implies expression (2.2). □

We remark that for $k = 0$, the number of NC-trees of size n with k descents and i leaves is given by the Narayana number, see [4]. The formula (2.2) is also a refinement of a formula obtained by Flajolet and Noy [6, Theorem 1].

By arranging the matches in $M \in \mathcal{M}_{n,k}$ in terms of the increasing order of their leaves, we obtain a sequence on the roots of these n matches. Assume that the label for the root of an r-match is colored white, while the label for the root of an l-match is colored black and written in boldface. Therefore, M can be represented by a bicolored sequence consisting of n distinct integers such that

(B1) there are $n - k$ white elements that belong to the set $[2n - k]$,

(B2) there are k black elements that belong to the set $[2n - 1]$,

(B3) the last element is white.

For example, the corresponding bicolored sequence for the labeled NC-tree in Figure 3 is $(16, \mathbf{9}, 8, \mathbf{13}, \mathbf{21}, 12, \mathbf{7}, 5, 14, \mathbf{18}, 17)$.

Denote $\mathcal{P}_{n,k}$ the set of bicolored sequences satisfying conditions (B1)–(B3). It is straightforward to obtain the following correspondence between labeled NC-trees and the bicolored sequences.

Theorem 2.3. *For $k \geqslant 1$, there is a bijection between labeled NC-tree of size n with k descents and the set $\mathcal{P}_{n,k}$.*

3 A bijective proof of Question 1.2

In this section we present a bijective proof of Question 1.2. To describe our bijection, we first notice that it is equivalent to construct a bijection between the set of labeled NC-trees of size $2k + 1$ with k l-edges and the set of labeled NC-trees of size $2k + 1$ with $k - 1$ l-edges. Using Theorem 2.3, it remains to give a one-to-one correspondence between $\mathcal{P}_{2k+1,k}$ and $\mathcal{P}_{2k+1,k-1}$.

Let $\mathcal{A}_{2k+1,k}$ denote the set of bicolored sequences π in $\mathcal{P}_{2k+1,k}$ such that a white element i is underlined and a black element j is double underlined. For example,

$$(16, \mathbf{9}, 8, \underline{\underline{\mathbf{13}}}, \mathbf{21}, 12, \mathbf{7}, \underline{5}, 14, \mathbf{18}, 17)$$

belongs to $\mathcal{A}_{11,5}$. By definition, we have

$$|\mathcal{A}_{2k+1,k}| = (k + 1)k|\mathcal{P}_{2k+1,k}|. \tag{3.1}$$

Let $\mathcal{B}_{2k+1,k-1}$ be the set of bicolored sequences σ in $\mathcal{P}_{2k+1,k-1}$ such that (i) a white element i is underlined, (ii) a white element j different from i is double underlined, (iii) neither i nor j is the last white element in σ. For example,

$$(16, \mathbf{9}, 8, \underline{13}, \mathbf{21}, \underline{\underline{12}}, \mathbf{7}, 5, 14, \mathbf{18}, 17)$$

belongs to $\mathcal{B}_{11,4}$. It follows that

$$|\mathcal{B}_{2k+1,k-1}| = (k+1)k|\mathcal{P}_{2k+1,k-1}|. \tag{3.2}$$

We proceed to construct a bijection $\Phi : \mathcal{A}_{2k+1,k} \to \mathcal{B}_{2k+1,k-1}$, which relies on the partition of $\mathcal{A}_{2k+1,k}$ and $\mathcal{B}_{2k+1,k-1}$. Without loss of generality, we may assume that the elements in an integer set are arranged in the increasing order. As mentioned above, the underlined and double underlined elements are always denoted by i and j respectively. Now we need to consider five cases.
Case 1. $\phi_1 : S_1 \to T_1$, where

- S_1 is the set of bicolored sequences π in $\mathcal{A}_{2k+1,k}$ such that i is not the last white element and $j < 3k+3$,

- T_1 is the set of bicolored sequences σ in $\mathcal{B}_{2k+1,k-1}$ such that $3k+3$ is not a white element in σ.

For $\pi \in S_1$, $\phi_1(\pi)$ can be generated from π by changing the black element j to a white element j. For example, $\pi = (16, \mathbf{9}, 8, \underline{\underline{\mathbf{13}}}, \mathbf{21}, \underline{12}, \mathbf{7}, 5, 14, \mathbf{18}, 17)$ in $\mathcal{A}_{11,5}$ belongs to S_1. Then $\phi_1(\pi) = (16, \mathbf{9}, 8, \underline{\underline{13}}, \mathbf{21}, \underline{12}, \mathbf{7}, 5, 14, \mathbf{18}, 17)$.
Case 2. $\phi_2 : S_2 \to T_2$, where

- S_2 is the set of bicolored sequences π in $\mathcal{A}_{2k+1,k}$ such that i is not the last white element and $j = 3k+3$,

- T_2 is the set of bicolored sequences σ in $\mathcal{B}_{2k+1,k-1}$ such that $3k+3$ is a white element but not the last in σ and $j = 3k+3$.

For $\pi \in S_2$, $\phi_2(\pi)$ can be generated from π by changing the black element $3k+3$ to a white element $3k+3$. For example, $\pi = (16, \mathbf{9}, \underline{8}, \mathbf{13}, \mathbf{21}, 12, \mathbf{7}, 5, 14, \underline{\underline{\mathbf{18}}}, 17)$ in $\mathcal{A}_{11,5}$ belongs to S_2. Then $\phi_2(\pi) = (16, \mathbf{9}, \underline{8}, \mathbf{13}, \mathbf{21}, 12, \mathbf{7}, 5, 14, \underline{\underline{18}}, 17)$.
Case 3. $\phi_3 : S_3 \to T_3$, where

- S_3 is the set of bicolored sequences π in $\mathcal{A}_{2k+1,k}$ such that $3k+3$ is an element in π and i is the last white element,

- T_3 is the set of bicolored sequences σ in $\mathcal{B}_{2k+1,k-1}$ such that $3k+3$ is the last white element in σ.

For $\pi \in S_3$, assume that j is the m-th black element in π. This implies that $1 \leqslant m \leqslant k$. Then $\phi_3(\pi)$ can be generated from π by the following procedures:

Step 1. Remove the double underline from j to the m-th white element;

Step 2. Change the black element $3k+3$ to a white element $3k+3$;

Step 3. Exchange the positions of $3k+3$ and i, where i is still underlined.

For example, $\pi = (16, \mathbf{9}, 8, \mathbf{13}, 12, \mathbf{18}, \underline{\underline{\mathbf{7}}}, 5, 14, \mathbf{21}, \underline{17})$ in $\mathcal{A}_{11,5}$ belongs to S_3. Then $\phi_3(\pi) = (16, \mathbf{9}, 8, \mathbf{13}, 12, \underline{17}, \mathbf{7}, \underline{\underline{5}}, 14, \mathbf{21}, 18)$.
Case 4. $\phi_4 : S_4 \to T_4$, where

- S_4 is the set of bicolored sequences π in $\mathcal{A}_{2k+1,k}$ such that
  - $3k+3$ is not an element in π,
  - i is the last white element,
  - j is the m-th element in the set $([4k+1] - E(\pi) - \{3k+3\}) \cup \{j\}$ with $1 \leqslant m \leqslant k$, where $E(\pi)$ is the set of elements in π,
- T_4 is the set of bicolored sequences σ in $\mathcal{B}_{2k+1,k-1}$ such that $3k+3$ is a white element but not the last in σ and $i = 3k+3$.

For $\pi \in S_4$, $\phi_4(\pi)$ can be generated from π by the following procedures:

Step 1. Remove the double underline from j to the m-th white element;

Step 2. Replace the black element j by a white element $3k+3$;

Step 3. Remove the underline from i to $3k+3$.

For example, $\pi = (16, \mathbf{9}, 8, \mathbf{13}, 12, \mathbf{19}, \underline{\underline{\mathbf{3}}}, 2, 14, \mathbf{21}, \underline{17})$ is in $\mathcal{A}_{11,5}$. By definition, $k = 5$ and $([4k+1] - E(\pi) - \{3k+3\}) \cup \{j\} = \{1, 3, 4, 5, 6, 7, 10, 11, 15, 20\}$. It follows that $m = 2$ and $1 \leqslant m \leqslant k$. One can verify that π belongs to S_4. Then $\phi_4(\pi) = (16, \mathbf{9}, \underline{\underline{8}}, \mathbf{13}, 12, \mathbf{19}, \underline{18}, 2, 14, \mathbf{21}, 17)$.
Case 5. $\phi_5 : S_5 \cup S_6 \cup S_7 \to T_5$, where

- S_5 is the set of bicolored sequences π in $\mathcal{A}_{2k+1,k}$ such that $3k+3$ is an element in π, i is not the last white element and $j > 3k+3$,
- S_6 is the set of bicolored sequences π in $\mathcal{A}_{2k+1,k}$ such that $3k+3$ is not an element in π, i is not the last white element and $j > 3k+3$,
- S_7 is the set of bicolored sequences π in $\mathcal{A}_{2k+1,k}$ such that
  - $3k+3$ is not an element in π,
  - i is the last white element,
  - j is the m-th element in the set $([4k+1] - E(\pi) - \{3k+3\}) \cup \{j\}$ with $k+1 \leqslant m \leqslant 2k$,
- T_5 is the set of bicolored sequences σ in $\mathcal{B}_{2k+1,k-1}$ such that $3k+3$ is a white element but not the last in σ, $i < 3k+3$ and $j < 3k+3$.

For $\pi \in S_5 \cup S_6 \cup S_7$, we have the following three cases.

1. If $\pi \in S_5$, assume that j is m-th element in π greater than $3k+3$. Then $\phi_5(\pi)$ can be generated from π by the following procedures:

   Step 1. Change the black element $3k+3$ to a white element $3k+3$;

   Step 2. Remove the double underline from j to the m-th white element, except for i and $3k+3$.

   For example, $\pi = (16, \mathbf{9}, 8, \mathbf{13}, \underline{\underline{\mathbf{21}}}, 12, \mathbf{7}, \underline{5}, 14, \mathbf{18}, 17)$ in $\mathcal{A}_{11,5}$ belongs to S_5. One sees that $m = 1$. Then $\phi_5(\pi) = (\underline{\underline{16}}, \mathbf{9}, 8, \mathbf{13}, \mathbf{21}, 12, \mathbf{7}, \underline{5}, 14, 18, 17)$.

2. If $\pi \in S_6$, let B denote the set of elements in π, except for j, greater than $3k+3$. Assume that j is the m-th element in the set $\{3k+4, 3k+5, \ldots, 4k+1\} - B$. Then $\phi_5(\pi)$ can be generated from π by the following procedures:

   Step 1. Replace the black element j by a white element $3k+3$;

   Step 2. Remove the double underline to the $(m+|B|)$-th white element, except for i and $3k+3$.

   For example, $\pi = (16, \mathbf{9}, \underline{8}, \mathbf{13}, \underline{\underline{\mathbf{20}}}, 12, \mathbf{7}, 5, 14, \mathbf{21}, 17)$ in $\mathcal{A}_{11,5}$ belongs to S_6. By definition, $B = \{21\}$, $\{3k+4, 3k+5, \ldots, 4k+1\} - B = \{19, 20\}$, $m = 2$ and $|B| = 1$. Hence, $\phi_5(\pi) = (16, \mathbf{9}, \underline{8}, \mathbf{13}, 18, 12, \mathbf{7}, \underline{\underline{5}}, 14, \mathbf{21}, 17)$.

3. If $\pi \in S_7$, $\phi_5(\pi)$ is generated from π by the following procedures:

   Step 1. Remove the underline from i to the $(m-k)$-th white element;

   Step 2. Replace the black element j by a white element $3k+3$;

   Step 3. Remove the double underline to the $(k-1)$-th white element, except for the underlined white element and $3k+3$.

   For example, $\pi = (16, \mathbf{9}, 8, \mathbf{13}, 12, \mathbf{19}, \underline{\underline{\mathbf{15}}}, 5, 14, \mathbf{21}, \underline{17})$ is in $\mathcal{A}_{11,5}$. By definition, $k = 5$, $([4k+1] - E(\pi) - \{3k+3\}) \cup \{j\} = \{1, 2, 3, 4, 6, 7, 10, 11, 15, 20\}$ and thus $m = 9$, which implies that $k+1 \leqslant m \leqslant 2k$. It follows that π belongs to S_7. Therefore $\phi_5(\pi) = (16, \mathbf{9}, 8, \mathbf{13}, 12, \mathbf{19}, 18, \underline{5}, \underline{\underline{14}}, \mathbf{21}, 17)$.

It is easy to see that (i) $S_i \cap S_j = \varnothing$ holds for $1 \leqslant i \neq j \leqslant 7$, (ii) $T_i \cap T_j = \varnothing$ holds for $1 \leqslant i \neq j \leqslant 5$, (iii) $\mathcal{A}_{2k+1,k} = \bigcup_{i=1}^{7} S_i$, and (iv) $\mathcal{B}_{2k+1,k-1} = \bigcup_{i=1}^{5} T_i$. Furthermore, it can be checked that each map ϕ_i is bijective. Given $\pi \in \mathcal{A}_{2k+1,k}$, it is natural to construct the bijection Φ as follows

$$\Phi(\pi) = \begin{cases} \phi_1(\pi), & if\ \pi \in S_1, \\ \phi_2(\pi), & if\ \pi \in S_2, \\ \phi_3(\pi), & if\ \pi \in S_3, \\ \phi_4(\pi), & if\ \pi \in S_4, \\ \phi_5(\pi), & if\ \pi \in S_5 \cup S_6 \cup S_7. \end{cases}$$

This finishes the proof of Question 1.2.

Acknowledgments.

We would like to thank the referees for helpful suggestions to improve the presentation. This work was supported by the National Natural Science Foundation of China (Projects 11001073 and 11101116), the Natural Science Foundation of Hebei Province (Projects A2012207001 and A2014208152), the One-Hundred Outstanding Innovative Talents Scheme of Hebei Province Education Department(No. BR2-231) and the Top Young-aged Talents Program of Hebei Province.

Appendix A: The proof of Theorem 2.1

We shall follow the notation introduced in Section 2. To prove Theorem 2.1, it is sufficient to show the following two lemmas.

Lemma .1. *Step 1 – Step 4 in the merging algorithm and Step 4 – Step 6 in the decomposition algorithm are inverse to each other.*

Lemma .2. *Step 5 – Step 7 in the merging algorithm and Step 1 – Step 3 in the decomposition algorithm are inverse to each other.*

The proof of Lemma .1. First, it is necessary to verify the feasibility of the algorithms. Obviously, Step 4 – Step 6 in the decomposition algorithm are always feasible. It remains to confirm the feasibility of Step 1 – Step 4 in the merging algorithm.

For $M \in \mathcal{M}_{n,k}$, note that M contains k trees without an r-edge and k #-marked vertices. The merging algorithm consists of k merging operations successively by merging the k #-marked vertices in their increasing order.

In order to show the validity of Step 1, we observe that each merging operation decreases the number of trees without an r-edge by 1 and the number of #-marked vertices by 1. Then before the ith ($1 \leqslant i \leqslant k$) merging operation, the number of trees without an r-edge and the number of #-marked vertices are both $k-i+1$. It follows that there will be at least one tree without an r-edge and #-marked vertices since $(2n)^{\#}$ belongs to an r-match.

For the validity of Step 3, we need to show that all the #-marked vertices must appear either as a root or as a leaf. This is obtained by observing that in our merging algorithm, the only trees with more than one #-marked vertex must be the match in M with two #-marked vertices.

To complete the proof of Lemma .1, it suffices to show that

Claim 1. if M is a set of matches obtained from a forest F by applying Step 4 – Step 6 of the decomposition algorithm, then we can recover F from M by applying Step 1 – Step 4 of the merging algorithm;

Claim 2. if F is a forest obtained from a set of matches M by applying Step 1 – Step 4 of the merging algorithm, then we can recover M from F by applying Step 4 – Step 6 of the decomposition algorithm.

We only prove Claim 1 for brevity and similarity. Set F' the corresponding forest by applying Step 1 – Step 4 of the merging algorithm to M. Note that any #-marked label in an r-match of M appears as a leaf. It follows that each r-match with #-marked label will be merged by adopting a vertical merge in the merging algorithm, and the resulting tree is an r-unique planted tree without #-marked label. This implies that F' is composed of r-unique planted trees. To verify Claim 1, it suffices to show that $F' = F$. Remind that the number of l-edges in F is k and the number of trees in F is $n-k$. We proceed by induction on the number k.

If $k = 1$, denote F by $\{T_1, T_2, \ldots, T_{n-1}\}$, where T_1 contains the unique l-edge. It implies that T_1 is composed of two edges. More precisely, an l-edge is attached under an r-edge in T_1. In addition, any of $T_2, \ldots, T_{n-1}$ is an r-match. By applying Step 4 – Step 6 of the decomposition algorithm to F, we get $M = \{T_{11}, T_{12}, T_2, \ldots, T_{n-1}\}$, where T_{11} and T_{12} are the l-match and the r-match respectively decomposed from T_1. Conversely, it is easy to see that $(2n)^{\#}$ is the unique #-marked label in M and $(2n)^{\#}$ appears as the leaf of T_{12}. Thus, by applying Step 1 – Step 4 of the merging algorithm, T_{11} will be merged with T_{12} by a vertical merge. Clearly, this operation recovers F, that is, $F' = F$.

Assume that the result holds with k replaced by $k-1$. Then if there are k l-edges in F, let (i, j) be the first edge decomposed from F at the first running of Step 4 – Step 6 in the decomposition algorithm. Notice that (i, j) is an l-edge, j is the rightmost child of i and j is a leaf. Let F_1 be the forest obtained from F by deleting the edge (i, j) and the vertex j. Besides, denote M_1 the corresponding set of matches by applying Step 4 – Step 6 of the decomposition algorithm to F_1. One sees that there are $k-1$ l-edges in F_1. By the induction hypothesis, we can recover F_1 from M_1 by applying Step 1 – Step 4 of the merging algorithm.

In light of the decomposition algorithm, it is straightforward to derive the following relations between M and M_1.

(1) (i, j) is an l-match in M but not in M_1;

(2) The set of other matches in M is the same as M_1 if we replace the label $(2n-k+1)^{\#}$ in M by i, and replace the #-marked labels in M_1 in increasing order by $(2n-k+2)^{\#}$, $(2n-k+3)^{\#}$, …, $(2n)^{\#}$ respectively.

Observe that (i, j) is still the first edge encountered whenever applying the merging algorithm to M (This is implied by the fact that all the l-matches without #-marked vertex are obtained from those leaves that are the rightmost child. Among such matches, (i, j) has the smallest root label). Furthermore, the first merge step for M is the merging of the match (i, j) and the match with the label $(2n-k+1)^{\#}$. Then clearly, the following merge steps are the same as those for M_1. Moreover, at any step, j is always the rightmost child of i. It follows that the forest F', corresponding to M by Step 1 – Step 4 of the merging algorithm, can be obtained from F_1 by attaching a rightmost child j to i. This gives $F' = F$. □

The proof of Lemma .2. The feasibility of Step 1 – Step 3 in the decomposition algorithm is obvious. It is routine to check the feasibility of Step 5 – Step 7 in the merging algorithm.

Before the first running of Step 5 – Step 7 in the merging algorithm, there are $n-k$ trees and $n-k-1$ $*$-marked vertices in these trees. Thus there must be a tree without $*$-marked vertex. After each merge, both the number of trees and the number of $*$-marked vertices decrease by 1. It means that we can always find a tree without $*$-marked vertex at any step. Therefore, Step 5 – Step 7 in the merging algorithm are feasible.

To complete the proof of Lemma .2, it remains to show that

Claim 3. if F is a forest obtained from a tree T by applying Step 1 – Step 3 of the decomposition algorithm, then we can recover T from F by applying Step 5 – Step 7 of the merging algorithm;

Claim 4. if T is a tree obtained from a forest F by applying Step 5 – Step 7 of the merging algorithm, then we can recover F from T by applying Step 1 – Step 3 of the decomposition algorithm.

The proofs of Claim 3 and Claim 4 are similar to that of Claim 1, and we only present a sketch for the proof of Claim 3. Set T' the corresponding tree by applying Step 5 – Step 7 of the merging algorithm to F. Since each tree in F is an r-unique planted tree, we eventually get a NC-tree T' by adopting merging algorithm. Now, it suffices to show $T' = T$. We proceed by induction on the number of r-edges in T.

If there is only one r-edge in T, T must be an r-unique planted tree. In terms of the decomposition algorithm, F contains a unique tree T. Hence, we can recover T from F trivially.

For a general labeled NC-tree T, let $\widehat{T}$ be the first r-unique planted subtree of T encountered at the first running of Step 1 – Step 3 in the decomposition algorithm. Moreover, let (i, j) be the planted edge of $\widehat{T}$. Notice that (i, j) is an r-edge and j is the rightmost child of i. Let T_1 be the tree obtained from T by deleting $\widehat{T}$ but keeping the vertex i. It is clear that the number of r-edges in T_1 decreases by 1. Denote F_1 the corresponding forest by applying Step 1 – Step 3 of the decomposition algorithm to T_1. By the induction hypothesis, we can recover T_1 from F_1 by applying Step 5 – Step 7 of the merging algorithm.

One can check that the relations between F and F_1 are

(1) $\widehat{T}$ is an r-unique planted tree in F but not in F_1;

(2) the set of other trees in F is the same as F_1 if we replace the labels $(n+2)^*$ in F by i, and replace the $*$-marked labels in F_1 in increasing order by $(n+3)^*$, $(n+4)^*$, …, $(2n-k)^*$ respectively.

In addition, $\widehat{T}$ is still the first tree encountered whenever applying the merging algorithm to F. Furthermore, the first merge step for F is the merging of $\widehat{T}$ and the tree with the label $(n+2)^*$. After that, the merge steps are the same as those for F_1. Finally, T' can be obtained from T_1 by combining the vertex i of T_1 and the vertex i of $\widehat{T}$ such that j is the rightmost child of i, which implies $T' = T$. □

References

[1] A. Asinowski and T. Mansour, Dyck paths with coloured ascents, *European J. Combin.* 29 (2008) 1262–1279.

[2] W.Y.C. Chen, A general bijective algorithm for trees, *Proc. Natl. Acad. Sci.* 87 (1990) 9635-9639.

[3] W.Y.C. Chen and S.H.F. Yan, Noncrossing trees and noncrossing graphs, *Electron. J. Combin.* 13 (2006) #N12.

[4] E. Deutsch, Dyck path enumeration, *Discrete Math.* 204 (1999) 167-202.

[5] E. Deutsch and M. Noy, Statistics on non-crossing trees, *Discrete Math.* 254 (2002) 75-87.

[6] P. Flajolet and M. Noy, Analytic combinatorics of non-crossing configurations, *Discrete Math.* 204 (1999) 203-229.

[7] A. Guo, Cyclic sieving phenomenon in non-crossing connected graphs, *Electron. J. Combin.* 18(1) (2011) #P9.

[8] D.S. Hough, Descents in noncrossing trees, *Electron. J. Combin.* 10 (2003) #N13.

[9] S. Kluge, The cyclic sieving phenomenon for non-crossing forests, *Electron. J. Combin.* 19(3) (2012) #P1.

[10] M. Noy, Enumeration of noncrossing trees on a circle, *Discrete Math.* 180 (1998) 301-313.

[11] A. Panholzer and H. Prodinger, Bijections for ternary trees and noncrossing trees, *Discrete Math.* 250 (2002) 181-195.

[12] R.P. Stanley, Enumerative Combinatorics, Vol. 2, Cambridge University Press, Cambridge, New York, 1999.

[13] Y. Sun and Z. Wang, Consecutive pattern avoidances in non-crossing trees, *Graphs Combin.* 26 (2010) 815–832.