Properties of Subtree-Prune-and-Regraft Operations on Totally-Ordered Phylogenetic Trees

Yun S. Song**

Department of Statistics, University of Oxford, 1 South Parks Road, Oxford, OX1 3TG, UK
song@stats.ox.ac.uk

The date of receipt and acceptance will be inserted by the editor

AMS Subject Classification: 05C05, 92D15

Abstract. We study some properties of subtree-prune-and-regraft (SPR) operations on leaf-labelled rooted binary trees in which internal vertices are *totally* ordered. Since biological events occur with certain time ordering, sometimes such totally-ordered trees must be used to avoid possible contradictions in representing evolutionary histories of biological sequences. Compared to the case of plain leaf-labelled rooted binary trees where internal vertices are only *partially* ordered, SPR operations on totally-ordered trees are more constrained and therefore more difficult to study. In this paper, we investigate the unit-neighbourhood $U(T)$, defined as the set of totally-ordered trees one SPR operation away from a given totally-ordered tree T. We construct a recursion relation for $|U(T)|$ and thereby arrive at an efficient method of determining $|U(T)|$. In contrast to the case of plain rooted trees, where the unit-neighbourhood size grows quadratically with respect to the number n of leaves, for totally-ordered trees $|U(T)|$ grows like $O(n^3)$. For some special topology types, we are able to obtain simple closed-form formulae for $|U(T)|$. Using these results, we find a sharp upper bound on $|U(T)|$ and conjecture a formula for a sharp lower bound. Lastly, we study the diameter of the space of totally-ordered trees measured using the induced SPR-metric.

Keywords: subtree prune regraft, ordered trees, neighbourhood, recombination

1. Introduction

Leaf-labelled binary trees, also known as binary phylogenetic trees, are widely used in representing evolutionary histories of biological sequences. Furthermore, some problems in evolutionary genetics naturally involve considering more than a single tree, and it therefore is important to have a method of comparing trees. Exactly how dissimilar-

** Present address: Department of Computer Science, University of California, Davis, CA 95616, USA. E-mail: yssong@cs.ucdavis.edu

ities between trees should be measured depends on the kind of trees being considered, as well as on the underlying problem.

A case in which one is inevitably led to use a collection of trees is the study of genealogies subject to recombination. Recombination can cause different regions in DNA sequences to have different evolutionary histories. In such a case, one is often interested in knowing how two trees differ in their topologies, the main idea being that a quantitative measure of the difference in tree topologies should reflect the number of detectable recombination events.

A type of tree rearrangement operation useful for studying topology changes due to recombination is the so-called *subtree-prune-and-regraft* (SPR) operation [1,6]. As the name indicates, an SPR operation roughly involves cutting an edge from a tree T, thus "pruning" a subtree t from T, and then "regrafting" t onto somewhere in the remaining part of T. One important point to note is that the precise definition of an SPR operation depends on the type of the tree on which the operation is being performed.

In [5], SPR operations on trees were used to address the problem of determining the minimum number of recombination events while constructing possible minimal evolutionary histories. This problem was first considered by Hein in [2], where unrooted trees were used. The main result of the work in [5] is that if the minimum number of recombination events is to be determined correctly for any data, then the right kind of trees and the right kind of distance between trees must be used. More exactly, the kind of trees which should be used are leaf-labelled rooted binary trees called *ordered* trees, in which internal vertices are *totally* ordered. Moreover, the induced SPR-metric on the space of ordered trees correctly quantifies the number of recombination events, and therefore it is of interest to study SPR operations on ordered trees.

The focus of this paper is to study some properties of SPR operations on ordered trees. This paper is a sequel to our earlier paper [4], which considered similar questions for SPR operations on plain leaf-labelled rooted binary trees where internal vertices are only *partially* ordered. In comparison to the case of plain rooted trees, SPR operations on ordered trees are more difficult to study, and this fact is clearly reflected in our work. As in [4], we investigate the unit-neighbourhood $U(T)$, defined as the set of trees one SPR operation away from a given tree T. In relation to recombination, the unit-neighbourhood size $|U(T)|$ gives the number of trees one recombination event away from T. To obtain an efficient method of determining $|U(T)|$ for arbitrary tree topology, we construct a recursion relation, from which we are able to derive simple closed-form formulae for $|U(T)|$ for some special topology types. In [4], it was shown that for plain rooted trees the unit-neighbourhood size grows quadratically with respect to the number n of leaves. In this paper, we show that $|U(T)|$ grows like $O(n^3)$ for ordered trees. We find a sharp upper bound on $|U(T)|$, and using this result we construct bounds on the diameter of the space of ordered trees. In addition, we conjecture a formula for a sharp lower bound on $|U(T)|$.

This paper is organised as follows. In Section 2, we define some notations and terminologies to be used throughout the paper. A more precise definition of SPR operations is provided there as well. The aforementioned recursion relation for $|U(T)|$ is constructed in Section 3, whereas special topology types with closed-form formulae for $|U(T)|$ are discussed in Section 4. In Section 5, we consider sharp bounds on $|U(T)|$. We conclude in Section 6 with a brief look into the diameter of the space of ordered trees.

Note. We have written a computer program to check all our results for $n \leq 8$.

2. Preliminaries

In this section, we describe the kind of trees to be considered in this paper and introduce the notion of SPR operations on such trees. Some definitions are taken verbatim from [4].

2.1. Trees

In this paper, we consider leaf-labelled rooted binary trees whose branch lengths are not specified. The discrete space of leaf-labelled rooted binary trees with n leaves is denoted by $\mathscr{T}_n^{\mathrm{r}}$. To be distinguished from *ordered* trees, which we presently define, a tree in $\mathscr{T}_n^{\mathrm{r}}$ is sometimes called a *plain* rooted tree. For $n \geq 2$, a tree in $\mathscr{T}_n^{\mathrm{r}}$ has n labelled degree-1 vertices called *leaves*; $n-2$ unlabelled degree-3 vertices; and a distinguished vertex of degree 2 called the *root*. A 1-leaved tree consists of a single labelled degree-0 vertex which serves as both the root and the leaf. The leaves of an n-leaved tree are bijectively labelled by a finite set $\mathscr{L}$ of n elements. In the remainder of this paper, when we say a tree without any qualification, we shall mean a leaf-labelled rooted binary tree.

An n-leaved rooted binary tree contains $2n-2$ edges. A *pendant* edge is an edge incident with a leaf, and a *cherry* is a 2-leaved subtree. For any (sub)tree s, we denote by $\ell(s)$ the number of leaves in s. It was shown by Schröder [3] that the number of inequivalent leaf-labelled rooted binary trees with n leaves is

$$R(n) := |\mathscr{T}_n^{\mathrm{r}}| = (2n-3)!! = (2n-3) \times (2n-5) \times \cdots \times 3 \times 1 = \frac{(2n-2)!}{2^{n-1}(n-1)!}\,.$$

In a rooted tree $T \in \mathscr{T}_n^{\mathrm{r}}$, time flows vertically from the root to the leaves, and we use $t(v)$ to denote the time associated to a vertex v. We say that a vertex $v \in T$ is a *descendant* of a vertex $u \in T$ if there exists a path from u to v which goes strictly forward in time; u is called an *ancestor* of v. The set $\{v_1, v_2, \ldots, v_{n-2}\}$ of degree-3 vertices in $T \in \mathscr{T}_n^{\mathrm{r}}$ is a partially ordered set whose binary relation denoted $\preceq$ is given by ancestral relation; we say that $v_i \prec v_j$ if v_i is an ancestor of v_j. Two degree-3 vertices v_i and v_j in $T \in \mathscr{T}_n^{\mathrm{r}}$ are *incomparable* if v_i is not in the path to the root from v_j and vice versa. An *ordered* tree is a leaf-labelled rooted binary tree whose corresponding set $\{v_1, v_2, \ldots, v_{n-2}\}$ of degree-3 vertices is a *totally* ordered set under the binary relation $\preceq_a$ defined by age ordering; we say that $u \prec_a v$ if and only if $t(u) < t(v)$. In an ordered tree, $t(v_i) \neq t(v_j)$ if $i \neq j$. Note that $v_i \prec_a v_j$ if v_j is a descendant of v_i. If there exists no ancestral relation between v_i and v_j, then either $v_i \prec_a v_j$ or $v_j \prec_a v_i$ is allowed. If $t(u) < t(v)$, or equivalently $u \prec_a v$, we say that v is *younger* than u. All degree-3 vertices in a tree are younger than the root. Two trees equivalent as plain rooted trees are distinct as ordered trees if the age ordering of their degree-3 vertices are different. For example, the two trees shown in Figure 1(a) are inequivalent ordered trees which are equivalent as plain rooted trees.

The *parent* $p(v)$ of a vertex v is an ancestor of v which is adjacent to v. Let $I(v)$ be the number of degree-3 vertices whose associated times lie between $t(v)$ and $t(p(v))$, i.e. $I(v)$ counts the number of *intermediate* degree-3 vertices between $t(v)$ and $t(p(v))$. For example, $I(u) = 0$ and $I(u') = 1$, where u and u' are as indicated in Figure 1(a). Let

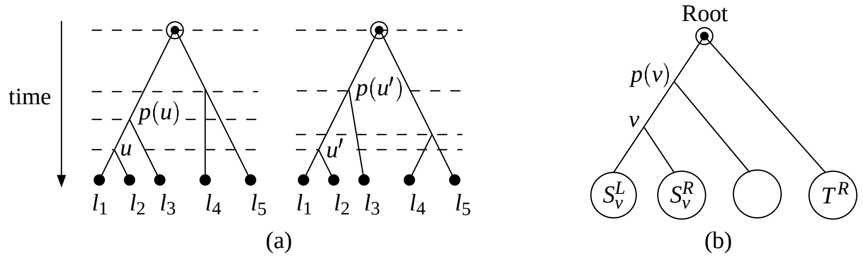

Figure 1: (a) Inequivalent ordered trees which are equivalent as plain rooted trees. (b) Illustration of some notations defined in the text.

v be a vertex in T of degree 2 or higher. Then, we use S_v to denote the subtree whose root is v. Furthermore, we let S_v^L and S_v^R denote the two subtrees of S_v whose roots are adjacent to v. By convention, S_v^L (resp. S_v^R) is drawn on the left (resp. right). If v is the root of T, then we define $T^L := S_v^L$ and $T^R := S_v^R$. See Figure 1(b) for a schematic depiction of these definitions.

The space of ordered trees with n leaves is denoted by $\mathscr{T}_n^{\mathrm{o}}$, and the number of inequivalent ordered trees with n leaves, for $n \geq 2$, is

$$D(n) := |\mathscr{T}_n^{\mathrm{o}}| = \prod_{k=2}^{n} \binom{k}{2} = \frac{n!(n-1)!}{2^{n-1}}. \tag{2.1}$$

This formula can be proved using induction on the number of leaves as follows. There exists a unique 2-leaved ordered tree and $D(2) = 1$. Given n labelled leaves, consider going backwards in time until exactly two leaves find a common ancestor. There are $\binom{n}{2}$ inequivalent such configurations. The number of inequivalent configurations further back in time is just the number of ordered trees with $n-1$ leaves. In summary, $D(n) = \binom{n}{2} D(n-1) = \binom{n}{2} \prod_{k=2}^{n-1} \binom{k}{2} = \prod_{k=2}^{n} \binom{k}{2}$.

The number $\Omega(T)$ of ordered trees corresponding to a plain rooted tree T depends on the topology of T. More exactly, the correspondence goes as follows. Let $d_L(v)$ (resp. $d_R(v)$) denote the number of degree-3 vertices which are left (resp. right) descendants of v. (Here, "left" and "right" refer to whether a descendant vertex is contained in S_v^L or in S_v^R. In the tree shown on the left hand side of Figure 1(a), for example, $d_L(\text{root}) = 2$ and $d_R(\text{root}) = 1$, whereas $d_L(p(u)) = 1$ and $d_R(p(u)) = 0$.) Then, it is not difficult to show that

$$\Omega(T) = \prod_{v, \text{vertices in } T \text{ of degree 2 or higher}} \Delta(v),$$

where $\Delta(v) := (d_L(v) + d_R(v))!/[d_L(v)!d_R(v)!]$.

2.2. SPR Operations on Plain Rooted Trees

The precise definition of an SPR operation depends on the type of the tree on which the operation is being performed. In general, the more constraints a tree has, the more restrictive an SPR operation has to be. We begin our discussion with plain leaf-labelled rooted trees. There are three kinds of SPR operations that can be performed on leaf-labelled rooted trees. Illustration of these operations is shown in Figure 2. In what follows, let T (resp. T') denote a tree before (resp. after) an SPR operation. The notation

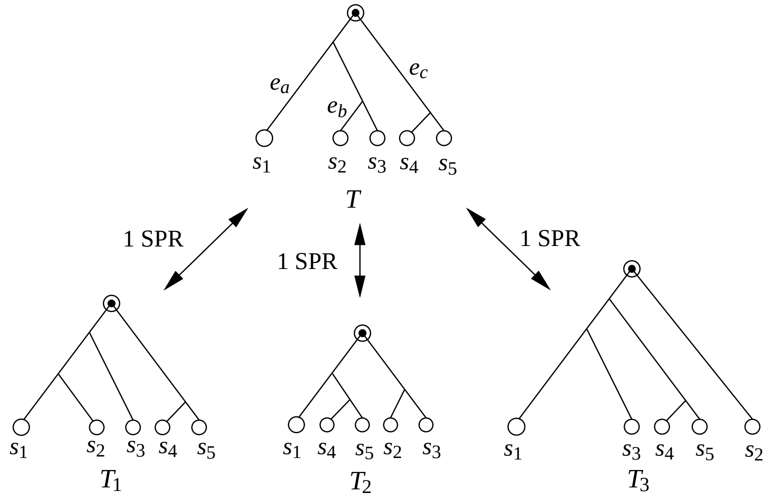

Figure 2: Illustration of SPR operations on plain rooted trees.

$T \setminus s$ denotes the part of T obtained from removing a subtree s and the edge incident with the root of s but not contained in s. In words the three SPR operations are as follows.

(1) An edge e is cut to prune a subtree s, not T^L or T^R, and s is regrafted onto a pre-existing edge in the remaining part $T \setminus s$ of T, thus creating a new degree-3 vertex. The vertex in $T \setminus s$ where e used to be incident gets removed. The root of T remains the root of T'. (In Figure 2, $T \to T_1$ is an example of this kind. The edge e_b is cut and then regrafted onto the edge e_a.)

(2) Let e_L and e_R, respectively, be the edges which join T^L and T^R to the root of T. The notations L and R can be interchanged in the following description: The edge e_L is cut to prune T^L, and T^L is regrafted onto a pre-existing edge in T^R. The edge e_R gets removed and the vertex which used to be joined to the root of T via e_R becomes the root of T'. (In Figure 2, $T \to T_2$ is an example of this kind. The edge e_c can be cut and regrafted onto e_a. The root of T^L containing s_1, s_2 and s_3 before the SPR operation then becomes the root of T_2.)

(3) Let r denote the root of T. An edge e is cut to prune a subtree s, not T^L or T^R. A new root r' is created and an edge is formed from r' to r. Lastly, s is joined to r'. (In Figure 2, $T \to T_3$ is an example of this kind. The edge e_b is cut and the pruned subtree s_2 gets joined to the new root.)

2.3. SPR Operations on Ordered Trees

For ordered trees, we impose an additional restriction on the definition of SPR operations. Consider a subtree s of an ordered tree $T \in \mathscr{T}_n^o$. Let u be the parent of the root of s. An operation which prunes and regrafts s to transform $T \in \mathscr{T}_n^o$ into $T' \in \mathscr{T}_n^o$ is defined to satisfy the following additional property: For any two vertices $v_i, v_j \in T$ neither being u, if $v_i \prec_a v_j$ before the SPR operation, then $v_i \prec_a v_j$ after the SPR operation, and vice versa.

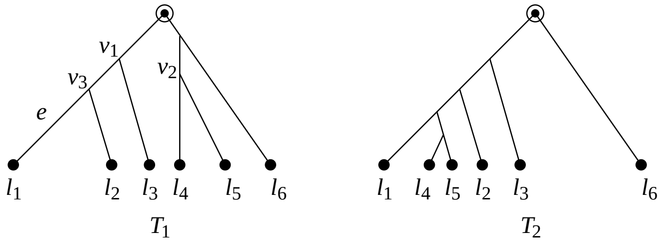

Figure 3: An example of ordered trees which are more than one SPR operation apart.

The two ordered trees T_1 and T_2 shown in Figure 3 are more than one SPR operation away from each other. If T_1 and T_2 were plain rooted trees, then the subtree containing l_4 and l_5 could be pruned and regrafted onto the edge e to transform T_1 into T_2. Such an operation is forbidden for ordered trees, however, because the ordering of the degree-3 vertices v_2 and v_3 would change by the operation. As a result, at least 2 SPR operations are required to transform T_1 into T_2, and vice versa.

3. The Unit-Neighbourhood of an Ordered Tree

Let $d_{SPR}(T,T')$ denote the minimum number of SPR operations required to transform T into T'. The unit-neighbourhood of an n-leaved ordered tree T is defined as

$$U(T) = \{T' \in \mathscr{T}_n^{\mathrm{o}} \,|\, d_{SPR}(T,T') = 1\}.$$

In this section, we discuss how the size $|U(T)|$ can be computed in a systematic way.

3.1. Moving Below the Youngest Degree-3 Vertex

Consider an n-leaved ordered tree T. The part of the tree below the youngest degree-3 vertex is illustrated in Figure 4. We are interested in knowing how many inequivalent ordered trees can be obtained by pruning a leaf labelled $l_i \in \mathcal{L} \setminus \{l_{n-1}, l_n\}$ and regrafting it onto a pendant edge *below* level t but not on the cherry containing l_{n-1} and l_n. There are $(n-2)$ ways of choosing a leaf for pruning and $(n-3)$ ways of choosing a pendant edge for regrafting. But, not all $(n-2)(n-3)$ such SPR operations lead to distinct ordered trees, and the resulting number of inequivalent ordered trees depends on the topology of the original tree T. Let us examine how over-counting can arise.

Perhaps the best way to illustrate how the above-mentioned counting should work is through an explicit example. Consider the example shown in Figure 5, where, for ease

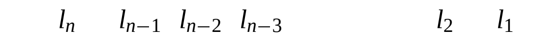

Figure 4: Schematic depiction of the bottom of an n-leaved ordered tree. The youngest degree-3 vertex in the tree is the one to which l_{n-1} and l_n are joined to form a cherry.

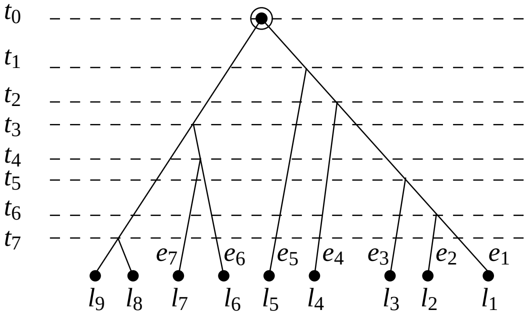

Figure 5: An example of a 9-leaved ordered tree where the problem of over-counting described in Section 3.1 can arise in several ways. For some pendant edges, the part below t_7 is labelled according to the leaf label.

of discussion, we have given labels to the parts below t_7 for some pendant edges. To avoid being long-winded, let $l_i \rightsquigarrow e_j$ denote pruning l_i and regrafting it onto e_j below level t_7. Then, note that $l_6 \rightsquigarrow e_7$ leads to the same ordered tree as that obtained from $l_7 \rightsquigarrow e_6$. Likewise, $l_1 \rightsquigarrow e_2$ is equivalent to $l_2 \rightsquigarrow e_1$. More generally, the number of over-counting due to this kind of symmetry is $c(T)-1$, where $c(T)$ is the number of cherries in T; here, -1 is for the cherry containing l_n and l_{n-1}.

Over-counting can arise in another way as well. For example, $l_4 \rightsquigarrow e_5$ is equivalent to $l_5 \rightsquigarrow e_4$. Similarly, $l_1 \rightsquigarrow e_3$ is equivalent to $l_3 \rightsquigarrow e_1$ and $l_2 \rightsquigarrow e_3$ is equivalent to $l_3 \rightsquigarrow e_2$. It is important to note, however, that $l_3 \rightsquigarrow e_4$ is *not* equivalent to $l_4 \rightsquigarrow e_3$. The presence of intermediate degree-3 vertices between t_2 and t_5 distinguishes the ordered tree obtained by $l_3 \rightsquigarrow e_4$ from that obtained by $l_4 \rightsquigarrow e_3$. More generally, the number of over-counting due to this kind of symmetry is equal to the number of leaves, other than l_n and l_{n-1}, satisfying the following: Let v denote the parent vertex of the leaf. Then, (i) the parent $p(v)$ of v is incident with a pendant edge, and (ii) there is no intermediate degree-3 vertex between $t(v)$ and $t(p(v))$. The number of such leaves is given by summing the following quantity over all degree-3 vertices v except for the youngest one, which is the root of the cherry containing l_n and l_{n-1}:

$$w(v) := \delta_{I(v),0}\, \delta_{\ell(S_{p(v)}\backslash S_v),1} \left[\delta_{\ell(S_v^L),1} + \delta_{\ell(S_v^R),1}\right]. \tag{3.2}$$

Here, $\delta_{a,b}$ is the Kronecker delta function and the remaining notations have been defined in Section 2.1. The first delta function in (3.2) makes sure that there are no intermediate degree-3 vertices between $t(v)$ and $t(p(v))$; the second delta function makes sure that the parent vertex $p(v)$ is incident with a pendant edge; the last delta function counts the number of pendant edges incident with v.

It is straightforward to check that there are no other sources of over-counting. In summary, the number of inequivalent ordered trees obtained from the SPR operations being considered here is

$$(n-2)(n-3) - \left[c(T) - 1 + \sum_{v\neq\text{youngest}} w(v)\right] =: (n-2)(n-3) - b(T), \tag{3.3}$$

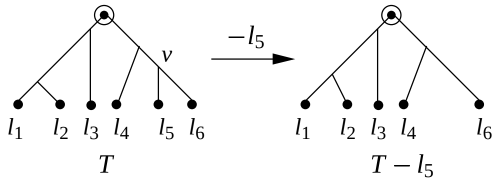
v
$-l_5$
l_1 l_2 l_3 l_4 l_5 l_6
T
l_1 l_2 l_3 l_4 l_6
$T - l_5$

Figure 6: An example of pruning without regrafting for a 6-leaved ordered tree.

where the sum is over all degree-3 vertices except for the youngest one. Just to demonstrate how this works, let us return to the example shown in Figure 5. Let v_i denote the vertex at t_i. Then, one obtains

$$w(v_1) = 0,\ \ w(v_2) = 1,\ \ w(v_3) = 0,\ \ w(v_4) = 0,\ \ w(v_5) = 0,\ \ w(v_6) = 2.$$

Therefore, since $n = 9$ and $C(T) = 3$ in the present example, $(n-2)(n-3) - b(T) = (9-2)(9-3) - (3-1+1+2) = 37$, which, as one can check explicitly, is the correct answer.

3.2. Pruning without Regrafting

We here define an operation which reduces the number of leaves in a tree by one. In an n-leaved ordered tree T, let v denote a degree-3 vertex with a leaf labelled l_k adjacent to it. Then, $T - l_k$ is defined as an $(n-1)$-leaved tree obtained by removing v, l_k and the edge joining them, and then connecting the two other edges which used to be incident with v into a single edge. An example is shown in Figure 6.

3.3. A Recursion for $|U(T)|$: The Bottom-Up Approach

Proposition 3.1. *For $n \geq 4$ and $T \in \mathscr{T}_n^{\mathrm{o}}$, the size of the unit-neighbourhood $U(T)$ satisfies the recursion relation*

$$|U(T)| = 2(n^2 - 2n - 2) - (h-1)(1 - \delta_{h,0}) - b(T) + |U(T - l_n)|, \qquad (3.4)$$

where l_n is a leaf with its parent vertex $p(l_n)$ being the youngest degree-3 vertex in T, $b(T)$ is defined as in (3.3), and $h := I(p(l_n))$, i.e. the number of intermediate degree-3 vertices between $t(p(l_n))$ and $t(p(p(l_n)))$.

Remark. Note that $|U(T)| = 2$, for all $T \in \mathscr{T}_3^{\mathrm{o}}$, serves as the boundary condition for the recursion.

Proof. There are 3 distinct cases we need to consider. These cases are illustrated in Figure 7. In each case, let v be the youngest degree-3 vertex in T; that is, let v be the parent vertex of l_n. We first do a “coarse counting” of the number of additional moves which are made possible because of the presence of v, i.e. we want to count the number of moves which would be absent if the cherry containing l_{n-1} and l_n were instead a single-leaved subtree. Then, for each of the 3 cases illustrated in Figure 7, we shall analyse which moves included in the “coarse counting” lead to equivalent ordered trees, thus eliminating all over-counting.

The following “coarse counting” of moves is common to all three cases:

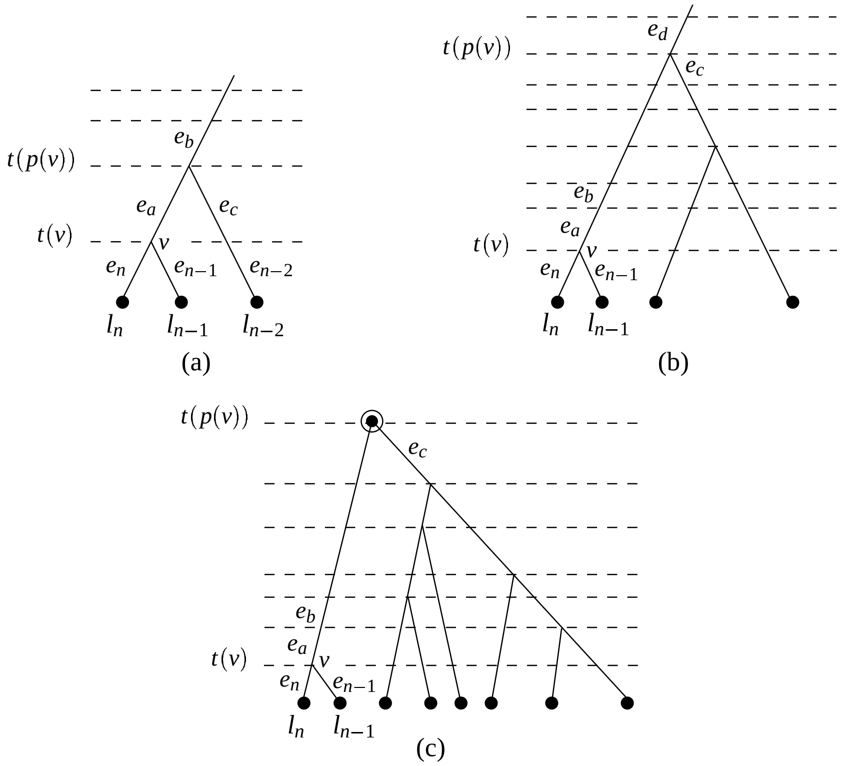

Figure 7: Illustration of the three possible cases in the proof of Proposition 3.1. (a) $h = 0$. (b) $h \neq 0$ and $p(v)$ is not the root. (c) $h \neq 0$ and $p(v)$ is the root.

(a) $+2(n-2)$ from moving other pendant edges to e_n or e_{n-1}.

(b) $+(n^2-n-2)$ from cutting e_n or e_{n-1} and then attaching it somewhere above $t(v)$. There are $\sum_{k=1}^{n-2}(k+1) = (n^2-n-2)/2$ distinct regions above $t(v)$, depending on which edge and which time interval is chosen. Note that moving e_n or e_{n-1} to somewhere below $t(v)$ leads to a tree topology already included here.

(c) $+2$ from cutting e_n or e_{n-1} and then attaching it to the root.

(d) $+(n-2)(n-3)-b(T)$ *inequivalent* moves from pruning any of $l_1,\ldots,l_{n-2}$ and then regrafting it somewhere below $t(v)$ but not onto e_n or e_{n-1}. These moves have been discussed in Section 3.1.

So far we have counted $2(n-2)+(n^2-n-2)+2+(n-2)(n-3)-b(T) = 2(n-1)^2-b(T)$ moves, not all of which are inequivalent. We now account for possible over-counting. In each case, the notation used conforms to the corresponding figure.

Case 1 ($h = 0$):
In this case, since v is the youngest degree-3 vertex, the descendant subtree of $p(v)$ not containing v must contain exactly one leaf. Moreover, since $n \geq 4$, $p(v)$ cannot be the root. We refer to Figure 7(a) in the following discussion:

(1-1) -2: Cutting e_n or e_{n-1} and then attaching it to e_a does not change the topology.

(1-2) -4: Moving e_n (resp. e_{n-1}) to e_c is equivalent to pruning l_{n-2} and regrafting it onto e_n (resp. e_{n-1}). Also, moving e_n (resp. e_{n-1}) to e_b is equivalent to pruning l_{n-2} and then regrafting it onto e_{n-1} (resp. e_n). These lead to double-counting in (a) and (b) of the above list.

Case 2 ($h \neq 0$ and $p(v)$ is not the root):
Shown in Figure 7(b) is a partial depiction of an ordered tree which falls into this case.

(2-1) -2: Cutting e_n or e_{n-1} and then attaching it to e_a does not change the topology.

(2-2) -2: Cutting e_n or e_{n-1} and then attaching it to e_b leads to a tree topology also included in part (d) of the above list.

(2-3) -2: Moving e_n (resp. e_{n-1}) to e_c is equivalent to moving e_{n-1} (resp. e_n) to e_d.

(2-4) $-(h-1)$: Let E be the edge joining v with $p(v)$. Pruning l_n and then regrafting it somewhere on E is equivalent to doing the same thing to l_{n-1}. Other than e_a and e_b, which we have already considered above, there are $h-1$ intervals in E.

Case 3 ($h \neq 0$ and $p(v)$ is the root):
An example of this case is shown in Figure 7(c).

(3-1) -2: Cutting l_n or l_{n-1} and then attaching it to e_a does not change the topology.

(3-2) -2: Cutting e_n or e_{n-1} and then attaching it to e_b leads to a tree topology also included in part (d) of the above list.

(3-3) -2: Moving e_n (resp. e_{n-1}) to e_c is equivalent to moving e_{n-1} (resp. e_n) to the root. This leads to double-counting in (b) and (c) of the above list.

(3-4) $-(h-1)$: Let E be the edge joining v with $p(v)$. Pruning l_n and then regrafting it somewhere on E is equivalent to doing the same thing to l_{n-1}. Other than e_a and e_b, which we have already considered above, there are $h-1$ intervals in E.

In summary, over-counting in each case contributes $-[6+(h-1)(1-\delta_{h,0})]$, and thus the number of inequivalent ordered trees in $U(T)$ which arise due to the presence of v is $2(n-1)^2 - b(T) - [6+(h-1)(1-\delta_{h,0})]$, which can be written as $2(n^2-2n-2)$ $-(h-1)(1-\delta_{h,0})] - b(T)$. The remaining number of inequivalent ordered trees in $U(T)$ is given by $|U(T-l_n)|$. This completes our proof of the proposition. ∎

3.4. Solving for $|U(T)|$

The recursion shown in (3.4) can be carried out sequentially until a 3-leaved ordered tree is reached. Many terms in such an expansion can be summed explicitly to obtain a simpler formula for $|U(T)|$. Before we proceed, we introduce an additional notation which will shortly prove convenient. Let l be a leaf adjacent with the youngest vertex in an ordered tree T. Then, $\mathcal{P}(T)$ is defined as the ordered tree $T-l$ obtained by *pruning* l from T. In a similar vein, $\mathcal{P}^k(T)$ is recursively defined by pruning from $\mathcal{P}^{k-1}(T)$ a leaf adjacent to the youngest vertex in $\mathcal{P}^{k-1}(T)$. The initial condition is $\mathcal{P}^0(T) = T$.

Now, since $|U(T')| = 2$ for all 3-leaved ordered trees T' and $2+\sum_{k=4}^{n} 2(k^2-2k-2)$ $= \frac{1}{3}(n+3)(n-2)(2n-5)$, the unit-neighbourhood size of T with degree-3 vertices $\{v_1,\dots,v_{n-2}\}$ can be written as

$$|U(T)| = \frac{1}{3}(n+3)(n-2)(2n-5) - \sum_{i=1}^{n-2}(I(v_i)-1)(1-\delta_{I(v_i),0}) - \sum_{k=0}^{n-4}\left[c(\mathcal{P}^k(T)) - 1 + w(\mathcal{P}^k(T))\right]. \tag{3.5}$$

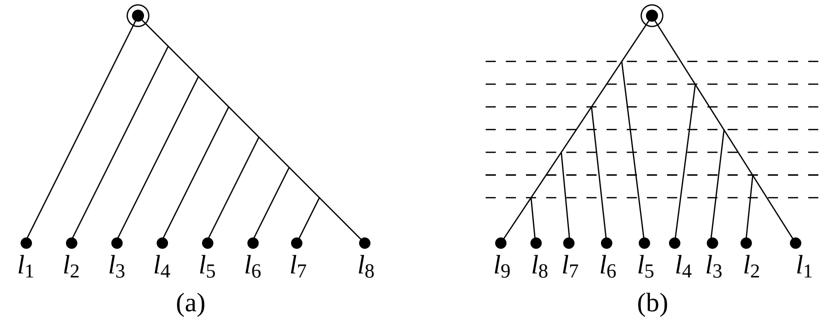

Figure 8: Examples of special topology types with closed-form formulae for $|U(T)|$. (a) An 8-leaved caterpillar tree. (b) A 9-leaved alternating-comb tree.

Here, $c(\mathcal{P}^k(T))$ is the number of cherries in $\mathcal{P}^k(T)$ and $w(\mathcal{P}^k(T))$ denotes

$$\sum_{v,\text{ not the youngest in }\mathcal{P}^k(T)} w(v),$$

where $w(v)$ is defined as in (3.2). We observe that (3.5) is considerably more complicated than the corresponding formula for plain rooted trees (c.f. Proposition 3.2. of [4]). Also, because the negative terms in (3.5) are at most of $O(n^2)$, the unit-neighbourhood size $|U(T)|$ for an ordered tree T is of $O(n^3)$. This is in contrast to the case of plain rooted trees, where the unit-neighbourhood size grows quadratically with respect to n.

4. Some Special Cases with Closed-Form Formulae

The formula for $|U(T)|$ shown in (3.5) is not in closed-form, but it may take on a simple form if we focus on a specific topology type. In this section, we consider three special topology types with closed-form formulae for $|U(T)|$. Results from this section will be used in the next section.

4.1. Caterpillar Trees

A caterpillar tree is a tree of the type illustrated in Figure 8(a). We have the following result for the size of the unit-neighbourhood:

Proposition 4.1. *Let $n \geq 3$. For an n-leaved caterpillar tree T,*

$$|U(T)| = \frac{1}{6}(4n^3 - 9n^2 - 19n + 42). \tag{4.6}$$

Proof. Let us analyse the unevaluated sums appearing in (3.5). Note that $I(v_i) = 0$, and therefore $(1 - \delta_{I(v_i),0}) = 0$, for all degree-3 vertices v_i in a caterpillar tree. Moreover, since a caterpillar tree contains exactly one cherry and since if T is a caterpillar tree, then so is $\mathcal{P}^k(T)$, we conclude that $c(\mathcal{P}^k(T)) = 1$ for all $0 \leq k \leq n-4$. Finally, since $w(T') = j-3$ for a j-leaved caterpillar tree, (3.5) is equal to $\frac{1}{3}(n+3)(n-2)(2n-5) - \sum_{j=4}^{n}(j-3) = \frac{1}{6}(4n^3 - 9n^2 - 19n + 42)$, which is our desired result. ∎

In [4], it was shown that, for τ an n-leaved caterpillar tree regarded as a *plain* rooted tree, the unit-neighbourhood size $|U(\tau)|$ is given by $3n^2-13n+14$.

4.2. Alternating-Comb Trees

We here define a new topology type which contains two caterpillar trees as subtrees. An example of this new topology type is shown in Figure 8(b). More exactly, an *alternating-comb* tree T is defined as an ordered tree such that T^L (resp. T^R) is a caterpillar tree with $\lceil \frac{n}{2} \rceil$ (resp. $\lfloor \frac{n}{2} \rfloor$) leaves, and the age-ordered sequence of degree-3 vertices in T alternates between T^L and T^R. Here, $\lceil \cdot \rceil$ is the ceiling function, whereas $\lfloor \cdot \rfloor$ is the floor function. The following proposition shows that alternating-comb trees also admit a very simple closed-form formula for the unit-neighbourhood size.

Proposition 4.2. *Let $n \geq 3$. For an n-leaved alternating-comb tree T,*

$$|U(T)| = \frac{1}{3}(2n^3-3n^2-20n+39). \qquad (4.7)$$

Proof. Let $4 \leq m \leq n$ and let $\{v_1, v_2, \ldots, v_{m-2}\}$ label the set of all degree-3 vertices in an m-leaved alternating-comb tree T' so that $t(v_1) < t(v_2) < \cdots < t(v_{m-2})$. Then, $I(v_1) = 0$ and $I(v_i) = 1$ for all $2 \leq i \leq m-2$, and therefore $(I(v_i)-1)(1-\delta_{I(v_i),0}) = 0$ for all $1 \leq i \leq m-2$. Furthermore, from the definition of $w(v)$ in (3.2), we conclude that $w(v_1) = 0$, since $\ell(S_{p(v_1)} \setminus S_{v_1}) > 1$ for an m-leaved alternating-comb tree where $m \geq 4$. Also, for all $2 \leq i \leq m-2$, we have $w(v_i) = 0$ because $\delta_{I(v_i),0} = 0$. Lastly, we note that $c(T') = 2$.

Now, since $\mathcal{P}^k(T)$, for all $0 \leq k \leq n-4$, also is an alternating-comb tree if T is an alternating-comb tree, the above discussion leads to the following non-vanishing contributions to (3.5):

$$\begin{aligned} |U(T)| &= \frac{1}{3}(n+3)(n-2)(2n-5) - \left[\sum_{k=0}^{n-4} c(\mathcal{P}^k(T)) - 1\right] \\ &= \frac{1}{3}(n+3)(n-2)(2n-5) - (n-3), \end{aligned}$$

which is equal to the (4.7). ∎

4.3. Cherry-Descending Trees

Consider the n-leaved ordered tree shown in Figure 9. It contains $\lfloor \frac{n-1}{2} \rfloor$ cherries, of which $\lfloor \frac{n-1}{2} \rfloor - 1$ are ordered in sequentially decreasing order as shown in the figure. Note that the value of the expression $n+1-2\lfloor \frac{n-1}{2} \rfloor$ shown in the figure is 2 if n is odd or 3 if n is even. For $n=3$ and $n=4$, cherry-descending trees and caterpillar trees are the same.

Proposition 4.3. *Let $n \geq 3$. For an n-leaved cherry-descending tree T,*

$$|U(T)| = \frac{1}{6}\left\{4n^3-9n^2-13n+42-3(2n+3)\left\lfloor \frac{n-1}{2} \right\rfloor + 9\left(\left\lfloor \frac{n-1}{2} \right\rfloor\right)^2\right\}. \qquad (4.8)$$

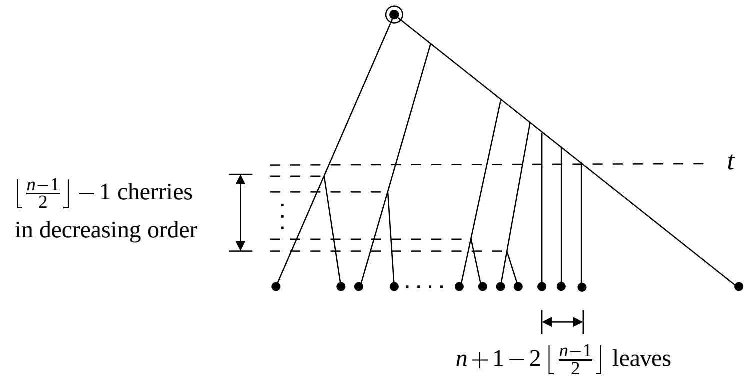
$\left\lfloor\frac{n-1}{2}\right\rfloor - 1$ cherries
in decreasing order
t
$n+1-2\left\lfloor\frac{n-1}{2}\right\rfloor$ leaves

Figure 9: An n-leaved cherry-descending tree. There are $\left\lfloor\frac{n-1}{2}\right\rfloor$ cherries in this tree.

Proof. We apply the main recursion (3.4) in the bottom-up fashion until what is left is a caterpillar tree T' with $n-\left(\left\lfloor\frac{n-1}{2}\right\rfloor-1\right)$ leaves. From Proposition 4.1, we know $|U(T')|$. We wish to examine what the other terms in the recursion contribute to $|U(T)|$. In the following discussion, we keep in mind that the number of times that the recursion must be applied before T' is reached is $\left(\left\lfloor\frac{n-1}{2}\right\rfloor-1\right)$. Adding up the terms like the first one on the right hand side of (3.4) gives

$$\sum_{i=n-\left(\left\lfloor\frac{n-1}{2}\right\rfloor-1\right)+1}^{n} 2(i^2-2i-2). \tag{4.9}$$

Furthermore, since $I(v)=n-\left\lfloor\frac{n-1}{2}\right\rfloor-1$, for all degree-3 vertices v below t, the sum of the terms like $-(h-1)(1-\delta_{h,0})$ gives

$$-\left(n-\left\lfloor\frac{n-1}{2}\right\rfloor-2\right)\left(\left\lfloor\frac{n-1}{2}\right\rfloor-1\right). \tag{4.10}$$

Lastly, one can show that $w(\mathcal{P}^k(T))=n-2\left\lfloor\frac{n-1}{2}\right\rfloor+1+k$ and that $c(\mathcal{P}^k(T))-1=\left\lfloor\frac{n-1}{2}\right\rfloor-1-k$, thus yielding

$$-\sum_{k=0}^{\left\lfloor\frac{n-1}{2}\right\rfloor-2} b(\mathcal{P}^k(T)) = -\left(n-\left\lfloor\frac{n-1}{2}\right\rfloor\right)\left(\left\lfloor\frac{n-1}{2}\right\rfloor-1\right). \tag{4.11}$$

Summing $|U(T')|$, (4.9), (4.10) and (4.11) gives (4.8). ∎

5. Sharp Bounds on $|U(T)|$

In this section, we study sharp lower and upper bounds on $|U(T)|$. We define

$$\delta_{\min}(n)=\min_{T\in\mathscr{T}_n}|U(T)| \qquad \text{and} \qquad \delta_{\max}(n)=\max_{T\in\mathscr{T}_n}|U(T)|,$$

	Plain Rooted Trees			Ordered Trees		
n	$R(n)$	$\delta_{\min}(n)$	$\delta_{\max}(n)$	$D(n)$	$\delta_{\min}(n)$	$\delta_{\max}(n)$
3	3	2	2	3	2	2
4	15	10	12	18	13	13
5	105	24	28	180	35	38
6	945	44	52	2,700	75	81
7	10,395	70	84	56,700	135	146
8	135,135	102	124	1,587,600	220	237

Table 1. The minimum and the maximum values of $|U(T)|$ for plain rooted trees and for ordered trees. These have been determined via computer-aided exhaustive search.

where $\mathscr{T}_n$ is either $\mathscr{T}_n^{\mathrm{o}}$ or $\mathscr{T}_n^{\mathrm{r}}$, depending on whether plain rooted trees or ordered trees are being considered. We have written a computer program which computes the unit-neighbourhood size $|U(T)|$ through exhaustive comparison of trees, and therefore, for small number n of leaves, the minimum and the maximum values of $|U(T)|$ can be determined via explicit computation. Table 1 shows the result of such computation for $n \leq 8$.

In [4], closed-form formulae for $\delta_{\min}(n)$ and $\delta_{\max}(n)$ were derived for plain rooted trees, and they agree with our exhaustive search results summarised in the first part of Table 1. The goal of this section is to construct closed-form formulae for the numbers shown in the second part of Table 1 pertaining to ordered trees.[1]

5.1. The Maximum Unit-Neighbourhood Size

We first establish the following proposition regarding alternating-comb trees defined in Section 4.2:

Proposition 5.1. *In the case of ordered trees with n leaves, where $n \geq 3$, alternating-comb trees have the maximum unit-neighbourhood size.*

Proof. We shall prove this statement by induction on n. For $n = 3$ and $n = 4$, all ordered trees have $|U(T)| = 2$ and $|U(T)| = 13$, respectively. For $n = 5$, one can explicitly show that a 5-leaved alternating-comb tree T has $|U(T)| = 38$, which is the maximum value. Suppose that the statement in the proposition is true for all $3 \leq n \leq k-1$, where $k \geq 6$. We now use (3.4) to compute $|U(T)|$, where T is a k-leaved alternating-comb tree. Let l_k be a leaf adjacent with the youngest vertex in T. Then, we note that $T - l_k$ also is an alternating-comb tree if T is an alternating-comb tree. It therefore follows from the induction hypothesis that $|U(T - l_k)| = \delta_{\max}(k-1)$. Furthermore, because $I(p(l_k)) = 1$ and $b(T) = 1$ if T is a k-leaved alternating-comb tree, the remaining terms in (3.4) contribute $2(k^2 - 2k - 2) - 1$. In summary, $|U(T)| = \delta_{\max}(k-1) + 2(k^2 - 2k - 2) - 1$.

Now, suppose that T' is an arbitrary k-leaved ordered tree and that l_k be a leaf adjacent with the youngest vertex in T'. Then, it follows from (3.4) that

$$|U(T')| = |U(T' - l_k)| + 2(k^2 - 2k - 2) - (I(p(l_k)) - 1)(1 - \delta_{I(p(l_k)),0}) - b(T').$$

[1] Incidentally, we take this opportunity to report an error in Table 1 of [5], which shows incorrect values of $\delta_{\min}(n)$ and $\delta_{\max}(n)$ for ordered trees. In that work, ordered trees in $|U(T)|$ which are equivalent to T as plain rooted trees were accidentally omitted in the exhaustive search.

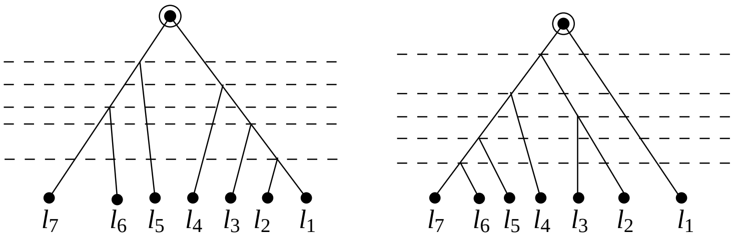

Figure 10: 7-leaved ordered trees which are not alternating-comb trees but still with the maximum unit-neighbourhood size.

Clearly, $(I(p(l_k))-1)(1-\delta_{I(p(l_k)),0})$ is non-negative. Furthermore, every tree contains at least one cherry, so $c(T')\geq 1$. Suppose that $c(T')=1$. Then, T' must be a caterpillar tree, so $b(T')=w(T')=k-3$, which is greater than 1 since $k\geq 6$. Suppose that $c(T')>1$. Then, clearly $b(T')=c(T')-1+w(T')\geq 1$, since $w(T')$ is non-negative. Thus, for all k-leaved ordered trees T', we must have $b(T')\geq 1$. Combining this result with the fact that $|U(T'-l_k)|\leq\delta_{\max}(k-1)$, we conclude that $|U(T')|\leq|U(T)|$. In other words, $|U(T)|=\delta_{\max}(k)$. This completes our induction. ▮

The following sharp upper bound on $|U(T)|$ now follows straightforwardly from Proposition 4.2 and Proposition 5.1:

Corollary 5.2. *For ordered trees with n-leaves, where $n\geq 3$,*

$$\delta_{\max}(n)=\frac{1}{3}(2n^3-3n^2-20n+39). \tag{5.12}$$

As it should be, the formula given in (5.12) is consistent with the numerical values obtained from our computer-aided exhaustive search (c.f. Table 1). Also, we point out that alternating-comb trees are not the only type of trees with the maximum unit-neighbourhood size. For example, the 7-leaved ordered trees shown in Figure 10 also have $\delta_{\max}(7)=146$ as their unit-neighbourhood size.

5.2. The Minimum Unit-Neighbourhood Size

For plain rooted trees, it was shown in [4] that caterpillar trees have the minimum unit-neighbourhood size. For ordered trees, however, that no longer holds true. For example, for $n\geq 5$, the formula for cherry-descending trees (c.f. (4.8)) always leads to a smaller value than that for caterpillar trees (c.f. (4.6)). In fact, the formula shown in (4.8) agrees with our exhaustively-determined numerical values of $\delta_{\min}(n)$ for ordered trees shown in Table 1. Moreover, for $n\leq 8$, we have explicitly checked that an ordered tree has the minimum unit-neighbourhood size if and only if it is a cherry-descending tree. This is in contrast with the maximum size case, where several different topology types can have the maximum unit-neighbourhood size.

It does not seem straightforward to show that cherry-descending trees have the minimum unit-neighbourhood size for all $n\geq 3$. Hence, based on the successful match, for $n\leq 8$, with the numerical values shown in Table 1, we propose the following conjecture:

Conjecture 5.3. *For ordered trees with n-leaves, where* $n \geq 3$,

$$\delta_{\min}(n) = \frac{1}{6}\left\{4n^3 - 9n^2 - 13n + 42 - 3(2n+3)\left\lfloor\frac{n-1}{2}\right\rfloor + 9\left(\left\lfloor\frac{n-1}{2}\right\rfloor\right)^2\right\}.$$

Note that this would imply that the difference between $\delta_{\max}(n)$ and $\delta_{\min}(n)$ for ordered trees grows like $O(n^2)$, although both $\delta_{\max}(n)$ and $\delta_{\min}(n)$ are of $O(n^3)$.

6. Diameter of $\mathscr{T}_n^{\text{o}}$

In the same spirit as the work done in [1] and in [4] for unrooted trees and plain rooted trees, respectively, we obtain the following result for ordered trees:

Proposition 6.1. *Let* $\text{diam}_{SPR}(\mathscr{T}_n^{\text{o}})$ *denote the diameter of* $\mathscr{T}_n^{\text{o}}$*, defined as the maximum value of* $d_{SPR}(T,T')$ *over all trees* $T,T' \in \mathscr{T}_n^{\text{o}}$*. Then,*

$$\frac{2}{3}n - o(n) \leq \text{diam}_{SPR}(\mathscr{T}_n^{\text{o}}) \leq n-2.$$

Proof. Following [1], we can use $[\delta_{\max}(n)]^{\text{diam}_{SPR}(\mathscr{T}_n^{\text{o}})} \geq D(n)$ to obtain the above lower bound for $\text{diam}_{SPR}(\mathscr{T}_n^{\text{o}})$. Here, $D(n)$ is the number of ordered trees (c.f. (2.1)), whereas $\delta_{\max}(n)$ is given by (5.12). Using Stirling's approximation for the factorial function and carrying out a similar set of steps as in [1], one can show that $\lim_{n\to\infty} \text{diam}_{SPR}(\mathscr{T}_n^{\text{o}})/n = 2/3$ and thus obtain the proposed lower bound. For the upper bound, the proof from [4] for plain rooted trees can be applied to ordered trees as well. ∎

Note. For plain rooted trees, the lower bound on $\text{diam}_{SPR}(\mathscr{T}_n^{\text{r}})$ obtained using the same approach as above is $n/2 - o(n)$ [4].

Acknowledgments. The author thanks Jotun Hein and Derek Phillips for useful discussion. Two anonymous referees are also thanked for helpful comments on improving the exposition. This research is supported by EPSRC under grant HAMJW and by MRC under grant HAMKA.

References

1. B.L. Allen and M. Steel, Subtree transfer operations and their induced metrics on evolutionary trees, Ann. Combin. **5** (2001) 1–13.
2. J. Hein, A heuristic method to reconstruct the history of sequences subject to recombination, J. Mol. Evol. **36** (1993) 396-405.
3. E. Schröder, Vier combinatorische probleme, Zeit. für. Math. Phys. **15** (1870) 361-376.
4. Y.S. Song, On the combinatorics of rooted binary phylogenetic trees, Ann. Combin. **7** (2003) 365–379.
5. Y.S. Song and J. Hein, Parsimonious reconstruction of sequence evolution and haplotype blocks: Finding the minimum number of recombination events, In: Algorithms in Bioinformatics (Proceedings of WABI 2003), G. Benson and R. Page, Eds., Springer-Verlag, Berlin, 2003, pp. 287-302.
6. D.L. Swofford and G.J. Olsen, Phylogeny reconstruction, In: Molecular Systematics, D.M. Hillis et al., Eds., Sinauer Associates, Massachusetts, 1990, pp. 411-501.