Journal of Combinatorial Theory, Series A 146 (2017) 239–263

Contents lists available at ScienceDirect

Journal of Combinatorial Theory, Series A

www.elsevier.com/locate/jcta

On the enumeration of tanglegrams and tangled chains [☆]

Sara C. Billey [a], Matjaž Konvalinka [b], Frederick A. Matsen IV [c]

[a] *Department of Mathematics, University of Washington, Seattle, WA 98195, USA*
[b] *Department of Mathematics, University of Ljubljana, and Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia*
[c] *Computational Biology Program, Fred Hutchinson Cancer Research Center, Seattle, WA 98109, USA*

ARTICLE INFO

Article history:
Received 3 November 2015
Available online xxxx

Keywords:
Tanglegrams
Enumeration
Binary trees
Binary partitions

ABSTRACT

Tanglegrams are a class of graphs arising in computer science and in biological research on cospeciation and coevolution. They are formed by identifying the leaves of two rooted binary trees. We give an explicit formula to count the number of distinct binary rooted tanglegrams with n matched leaves, along with a simple asymptotic formula and an algorithm for choosing a tanglegram uniformly at random. The enumeration formula is then extended to count the number of tangled chains of binary trees of any length. This includes a new formula for the number of binary trees with n leaves. We also give a conjecture for the expected number of cherries in a large randomly chosen binary tree and an extension of this conjecture to other types of trees.

[☆] The first author was partially supported by the National Science Foundation grant DMS-1101017. The second author was supported by Research Program Z1-5434 and Research Project BI-US/14-15-026 of the Slovenian Research Agency. The third author was supported by National Science Foundation grants DMS-1223057 and CISE-1564137.

E-mail address: matjaz.konvalinka@fmf.uni-lj.si (M. Konvalinka).

URLs: http://www.math.washington.edu/~billey/ (S.C. Billey), http://www.fmf.uni-lj.si/~konvalinka/ (M. Konvalinka), http://matsen.fredhutch.org/ (F.A. Matsen).

http://dx.doi.org/10.1016/j.jcta.2016.10.003

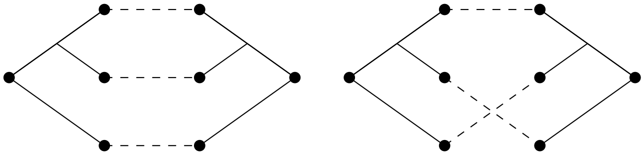

Fig. 1. The tanglegrams of size 3.

1. Introduction

Tanglegrams are graphs obtained by taking two binary rooted trees with the same number of leaves and matching each leaf from the tree on the left with a unique leaf from the tree on the right. This construction is used in the study of cospeciation and coevolution in biology. For example, the tree on the left may represent the phylogeny of a host, such as gopher, while the tree on the right may represent a parasite, such as louse [14], [22, page 71]. One important problem is to reconstruct the historical associations between the phylogenies of host and parasite under a model of parasites switching hosts, which is an instance of the more general problem of *cophylogeny estimation*. See [22–24] for applications in biology. Diaconis and Holmes have previously demonstrated how one can encode a phylogenetic tree as a series of binary matchings [7], which is a distinct use of matchings from that discussed here.

In computer science, the Tanglegram Layout Problem (TL) is to find a drawing of a tanglegram in the plane with the left and right trees both given as planar embeddings with the smallest number of crossings among (straight) edges matching the leaves of the left tree and the right tree [2]. These authors point out that tanglegrams occur in the analysis of software projects and clustering problems.

In this paper, we give the exact enumeration of tanglegrams with n matched pairs of vertices, along with a simple asymptotic formula and an algorithm for choosing a tanglegram uniformly at random. We refer to the number of pairs of matched vertices in a tanglegram as its *size*. Furthermore, two tanglegrams are considered to be equivalent if one is obtained from the other by replacing the tree on the left or the tree on the right by isomorphic trees. For example, in Fig. 1, the two non-equivalent tanglegrams of size 3 are shown.

We state our main results here postponing some definitions until Section 2. The following is our main theorem.

Theorem 1. *The number of tanglegrams of size n is*

$$t_n = \sum_{\lambda} \frac{\prod_{i=2}^{\ell(\lambda)} \left(2(\lambda_i + \cdots + \lambda_{\ell(\lambda)}) - 1\right)^2}{z_\lambda},$$

where the sum is over binary partitions of n and z_λ is defined by Equation (1).

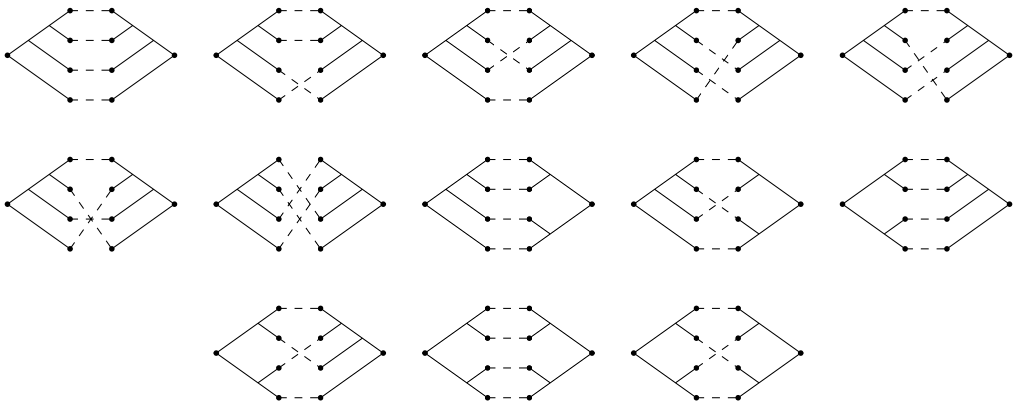

Fig. 2. The 13 tanglegrams of size 4.

The first 10 terms of the sequence t_n starting at $n = 1$ are

$$1, 1, 2, 13, 114, 1509, 25595, 535753, 13305590, 382728552,$$

see [21, A258620] for more terms.

Example. The binary partitions of $n = 4$ are (4), (2, 2), (2, 1, 1) and (1, 1, 1, 1), so

$$t_4 = \frac{1}{4} + \frac{3^2}{8} + \frac{3^2 \cdot 1^2}{4} + \frac{5^2 \cdot 3^2 \cdot 1^2}{24} = 13$$

as shown in Fig. 2. It takes a computer only a moment to compute

$$t_{42} = 33889136420378480492869677415186948305278176263020722832251621520063757$$

and under a minute to compute all 3160 integer digits of t_{1000} using a recurrence based on Theorem 1 given in Section 6.

We use the main theorem to study the asymptotics of the sequence t_n. It turns out that

$$\frac{t_n}{n!} \sim \frac{e^{\frac{1}{8}} 4^{n-1}}{\pi n^3},$$

see Corollary 8 for an explanation and better estimates.

A side result of the proof is a new formula for the number of inequivalent (i.e., non-isomorphic) binary trees, called the Wedderburn–Etherington numbers [21, A001190].

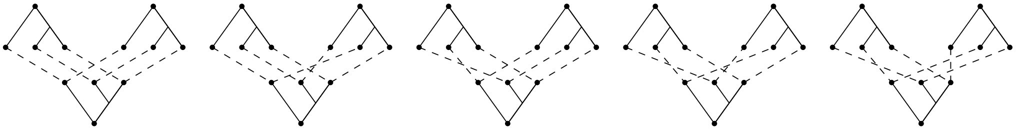

Fig. 3. The tangled chains of length 3 for $n = 3$.

Theorem 2. *The number of inequivalent binary trees with n leaves is*

$$b_n = \sum_{\lambda} \frac{\prod_{i=2}^{\ell(\lambda)} (2(\lambda_i + \cdots + \lambda_{\ell(\lambda)}) - 1)}{z_\lambda},$$

where the sum is over binary partitions of n.

A *tangled chain* is an ordered sequence of k binary trees with matchings between neighboring trees in the sequence. For $k = 1$, these are inequivalent binary trees, and for $k = 2$, these are tanglegrams, so the following generalizes Theorems 1 and 2.

In terms of computational biology, tangled chains of length k formalize the essential input to a variety of problems on k leaf-labeled (phylogenetic) trees (e.g. [28]).

Theorem 3. *The number of ordered tangled chains of length k for n is*

$$\sum_{\lambda} \frac{\prod_{i=2}^{\ell(\lambda)} \left(2(\lambda_i + \cdots + \lambda_{\ell(\lambda)}) - 1\right)^k}{z_\lambda},$$

where the sum is over binary partitions of n.

Example. For $n = k = 3$, we have partitions $(2, 1)$ and $(1, 1, 1)$, and the theorem gives

$$\frac{1^3}{2} + \frac{3^3 \cdot 1^3}{6} = 5,$$

as shown in Fig. 3. For $k = 3$, the number of tangled chains on trees with n leaves gives rise to the sequence starting

$$1, 1, 5, 151, 9944, 1196991, 226435150, 61992679960, 23198439767669, 11380100883484302.$$

See [21, A258486] for more terms.

From the enumerative point of view, it is also quite natural to ask how likely a particular tree T is to appear on one side or the other of a uniformly selected tanglegram. In Section 7, we give a simple explicit conjecture for the asymptotic growth of the expected number of copies of T on one side of a tanglegram as a function of T and the size of the tanglegram. For example, the cherries of a binary tree are pairs of leaves connected by a common parent. We conjecture that the expected number of cherries in one of the binary trees of a tanglegram of size n chosen in the uniform distribution is $n/4$.

Further discussion of the applications of tanglegrams along with several variations on the theme are described in [20]. In particular, tanglegrams can be used to compute the subtree-prune-regraft distance between two binary trees. In a recent follow up paper, Gessel has used the formula given here for binary trees to count several variations on tanglegrams using the theory of species [13].

The paper proceeds as follows. In Section 2, we define our terminology and state the main theorems. We prove the main theorems in Section 3. Section 4 contains an algorithm to choose a tanglegram uniformly at random for a given n. In Section 5, we give several asymptotic approximations to the number of tanglegrams with increasing accuracy and complexity. In Section 6, we give a recursive formula for both the number of tanglegrams and for tangled chains. We conclude with several open problems and conjectures in Section 7.

2. Background

In this section, we recall some vocabulary and notation on partitions and trees. This terminology can also be found in standard textbooks on combinatorics such as [26]. We use these terms to give the formal definition of tanglegrams and the notation used in the main theorems.

A *partition* $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$ is a weakly decreasing sequence of positive integers. The length $\ell(\lambda)$ of a partition is the number of entries in the sequence, and $|\lambda|$ denotes the sum of the entries of λ. We say λ is a *binary partition* if all its parts are equal to a nonnegative power of 2. Binary partitions have appeared in a variety of contexts, see for instance in [17,18,25] and [21, A000123]. When writing partitions, we sometimes omit parentheses and commas.

If λ is a nonempty binary partition with m_i occurrences of the letter 2^i for each i, we also denote λ by $(1^{m_0}, 2^{m_1}, 4^{m_2}, 8^{m_3}, \dots, (2^j)^{m_j})$ where $2^j = \lambda_1$ is the maximum value in λ. Given $\lambda = (1^{m_0}, 2^{m_1}, \dots, (2^j)^{m_j})$, let z_λ denote the product

$$z_\lambda = 1^{m_0} 2^{m_1} \cdots (2^j)^{m_j} m_0! m_1! m_2! \cdots m_j!. \tag{1}$$

The numbers z_λ are well known since the number of permutations in $\mathfrak{S}_n$ with cycle type λ is $n!/z_\lambda$ [26, Prop. 1.3.2]. For example, for $\lambda = 44211 = (1^2, 2^1, 4^2)$, $z_\lambda = 1^2 \cdot 2^1 \cdot 4^2 \cdot 2! \cdot 1! \cdot 2! = 128$.

A rooted tree has one distinguished vertex assumed to be a common ancestor of all other vertices. The neighbors of the root are its *children*. Each vertex other than the root has a unique parent going along the path back to the root, the other neighbors are its children. In a binary tree, each vertex either has two children or no children. A vertex with no children is a *leaf*, and a vertex with two children is an *internal vertex*. Two binary rooted trees with labeled leaves are said to be *equivalent* if there is an isomorphism from one to the other as graphs mapping the root of one to the root of the other. Let B_n be the set of inequivalent binary rooted trees with $n \geq 1$ leaves, and let b_n be the number of elements in the set B_n. The sequence of b_n's for $n \geq 1$ begins

$$1, 1, 1, 2, 3, 6, 11, 23, 46, 98.$$

We can inductively define a linear order on rooted trees as follows. We say that $T > S$ if either:

- T has more leaves than S,
- T and S have the same number of leaves, T has subtrees T_1 and T_2, $T_1 \geq T_2$, S has subtrees S_1 and S_2, $S_1 \geq S_2$, and $T_1 > S_1$ or $T_1 = S_1$ and $T_2 > S_2$.

We assume that every tree T in B_n, $n \geq 2$, is presented so that $T_1 \geq T_2$, where T_1 is the left subtree (or upper subtree if the tree is drawn with the root on the left or on the right) and T_2 is the right (or lower) subtree.

For each tree $T \in B_n$, we can identify its automorphism group $A(T)$ as follows. Fix a labeling on the leaves of T using the numbers $1, 2, \ldots, n$. Label each internal vertex by the union of the labels for each of its children. The edges in T are pairs of subsets from $[n] := \{1, \ldots, n\}$, each representing the label of a child and its parent. Let $v = [v(1), v(2), \ldots, v(n)]$ be a permutation in the symmetric group $\mathfrak{S}_n$. Then, $v \in A(T)$ if permuting the leaf labels by the function $i \mapsto v(i)$, for each i, leaves the set of edges fixed.

A theorem due to Jordan [16] tells us that if T is a tree with subtrees T_1 and T_2, then $A(T)$ is isomorphic to $A(T_1) \times A(T_2)$ if $T_1 \neq T_2$, and to the wreath product $A(T_1) \wr \mathbb{Z}_2$ if $T_1 = T_2$. Since the automorphism group of a tree on one vertex is trivial, this implies that the general $A(T)$ can be obtained from copies of $\mathbb{Z}_2$ by direct and wreath products (see [20] for more details). Furthermore, if $T_1 \neq T_2$, then the conjugacy type of an element of $A(T)$ is $\lambda^1 \cup \lambda^2$, where λ^i is the conjugacy type of an element of $A(T_i)$, $i = 1, 2$, and $\lambda^1 \cup \lambda^2$ is the multiset union of the two sequences written in decreasing order. If $T_1 = T_2$, then for an arbitrary element of $A(T)$ either the leaves in each subtree remain in that subtree, or all leaves are mapped to the other subtree. The conjugacy type of an element of $A(T)$ is then either $\lambda^1 \cup \lambda^2$, where λ^i is the conjugacy type of an element of $A(T_i)$, $i = 1, 2$, or it is $2\lambda^1 = \lambda^1 \cup \lambda^1$, where λ^1 is the conjugacy type of an element of $A(T_1)$. In particular, the conjugacy type of any element of the automorphism group of a binary tree must be a binary partition.

Next, we define tanglegrams. Given a permutation $v \in \mathfrak{S}_n$ along with two trees $T, S \in B_n$ each with leaves labeled $1, \ldots, n$, we construct an *ordered binary rooted tanglegram* (T, v, S) of size n with T as the left tree, S as the right tree, by identifying leaf i in T with leaf $v(i)$ in S. Note, (T, v, S) and (T', v', S') are considered to represent the same tanglegram provided $T = T'$, $S = S'$ as trees and $v' = uvw$ where $u \in A(T)$ and $w \in A(S)$. Let T_n be the set of all ordered binary rooted tanglegrams of size n, and let t_n

be the number of elements in the set T_n. For example, $t_3 = 2$ and $t_4 = 13$. Figs. 1 and 2 show the tanglegrams of sizes 3 and 4 where we draw the leaves of the left and right tree on separate vertical lines and show the matching using dashed lines. The dashed lines are not technically part of the graph, but this visualization allows us to give a planar drawing of the two trees.

We remark that the *plane binary trees* with $n \geq 2$ leaves are a different family of objects from B_n that also come up in this paper. These are trees embedded in the plane so that the left child of a vertex is distinguishable from the right child. The plane binary trees with $n+1$ leaves are well known to be counted by Catalan numbers

$$c_n = \frac{1}{n+1}\binom{2n}{n} = \frac{2^n(2n-1)!!}{(n+1)!}$$

because they clearly satisfy the Catalan recurrence

$$c_n = c_0c_{n-1} + c_1c_{n-2} + c_2c_{n-3} + \cdots + c_{n-1}c_0$$

with $c_0 = c_1 = 1$. For example, there are $c_2 = 2$ distinct plane binary trees with 3 leaves which are mirror images of each other while $b_3 = 1$. The sequence of c_n's for $n \geq 0$ begins

$$1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862,$$

see [21, A000108].

Dulucq and Guibert [8] have studied “twin binary trees”, which are pairs of plane binary trees with matched vertices. This is the plane version of tanglegrams. They show that twin binary trees are in bijection with Baxter permutations. The Baxter permutations in $\mathfrak{S}_n$ are enumerated by a formula due to Chung–Graham–Hoggatt–Kleiman [4]

$$a_n = \frac{\sum_{k=1}^{n}\binom{n+1}{k-1}\binom{n+1}{k}\binom{n+1}{k+1}}{\binom{n+1}{1}\binom{n+2}{2}}.$$

See also the bijective proof by Viennot [27], and further refinements [5,9].

3. Proof of the main theorems

The focus of this section is the proof of Theorem 1, namely that

$$t_n = \sum_{\lambda} \frac{\prod_{i=2}^{\ell(\lambda)} \left(2(\lambda_i + \cdots + \lambda_{\ell(\lambda)}) - 1\right)^2}{z_\lambda},$$

where the sum is over binary partitions of n. The proof of Theorem 1 reflects the chronological steps of discovery. Theorem 2 will follow from an auxiliary result, and the proof of Theorem 3 is similar and is included at the end of the section.

The number of tanglegrams is, by definition, equal to

$$t_n = \sum_T \sum_S |\mathcal{C}(T,S)|,$$

where the sums on the right are over inequivalent binary trees with n leaves, and $\mathcal{C}(T,S)$ is the set of double cosets of the symmetric group $\mathfrak{S}_n$ with respect to the double action of $A(T)$ on the left and $A(S)$ on the right. Let us fix $T \in B_n$ and $S \in B_n$ and write $\mathcal{C} = \mathcal{C}(T,S)$. Then

$$|\mathcal{C}| = \sum_{C\in\mathcal{C}} 1 = \sum_{C\in\mathcal{C}} \frac{|C|}{|C|} = \sum_{C\in\mathcal{C}} \sum_{w\in C} \frac{1}{|C|} = \sum_{w\in\mathfrak{S}_n} \frac{1}{|C_w|},$$

where C_w is the double coset of $\mathfrak{S}_n$ that contains w. It is known (e.g. [15, Theorem 2.5.1 on page 45 and Exercise 40 on page 49]) that the size of the double coset $C_w = A(T)wA(S)$ is the quotient

$$\frac{|A(T)|\cdot|A(S)|}{|A(T)\cap wA(S)w^{-1}|}, \tag{2}$$

and therefore,

$$|\mathcal{C}| = \sum_{w\in\mathfrak{S}_n} \frac{|A(T)\cap wA(S)w^{-1}|}{|A(T)|\cdot|A(S)|}.$$

We have

$$\begin{aligned}\sum_{w\in\mathfrak{S}_n} |A(T)\cap wA(S)w^{-1}| &= \sum_{w\in\mathfrak{S}_n}\sum_{u\in A(T)}\sum_{v\in A(S)} [\![u = wvw^{-1}]\!] \\ &= \sum_{u\in A(T)}\sum_{v\in A(S)}\sum_{w\in\mathfrak{S}_n} [\![u = wvw^{-1}]\!],\end{aligned}$$

where $[\![\cdot]\!]$ is the indicator function. Now $u = wvw^{-1}$ can only be true if u and v are permutations of the same conjugacy type λ, which must necessarily be a binary partition as noted above. Furthermore, if u and v are both of type λ, then there are z_λ permutations w for which $u = wvw^{-1}$. That means that

$$|\mathcal{C}(T,S)| = \frac{\sum_\lambda |A(T)_\lambda|\cdot|A(S)_\lambda|\cdot z_\lambda}{|A(T)|\cdot|A(S)|}, \tag{3}$$

where $A(T)_\lambda$ (respectively, $A(S)_\lambda$) denotes the elements of $A(T)$ (resp., $A(S)$) of type λ.

Equation (3) is already quite useful for computing all tanglegrams with fixed left and right trees. For example, if T and S are both the least symmetric tree with only one cherry, then $A(T) = A(S) = \{\mathrm{id}, (1,2)\}$, the sum is over only two binary partitions of size n, namely $(1,\dots,1)$ and $(2,1,\dots,1)$, and we get

$$|\mathcal{C}| = \frac{n! + 2(n-2)!}{2 \cdot 2} = \frac{(n^2 - n + 2)(n-2)!}{4}.$$

In some other cases the summation is over many more λ's, and can get quite complicated.

However, to get the formula for t_n we want to sum Equation (3) over all pairs of trees, and fortunately a change of the order of summation helps. Indeed, we have

$$t_n = \sum_T \sum_S \frac{\sum_\lambda |A(T)_\lambda| \cdot |A(S)_\lambda| \cdot z_\lambda}{|A(T)| \cdot |A(S)|} = \sum_\lambda z_\lambda \cdot \sum_T \sum_S \frac{|A(T)_\lambda| \cdot |A(S)_\lambda|}{|A(T)| \cdot |A(S)|} \tag{4}$$

$$= \sum_\lambda z_\lambda \cdot \left(\sum_T \frac{|A(T)_\lambda|}{|A(T)|} \right)^2, \tag{5}$$

and the main theorem will be proved once we have shown the following proposition.

Proposition 4. *For a binary partition λ,*

$$\sum_{T \in B_n} \frac{|A(T)_\lambda|}{|A(T)|} = \frac{\prod_{i=2}^{\ell(\lambda)} \left(2\left(\lambda_i + \cdots + \lambda_{\ell(\lambda)}\right) - 1\right)}{z_\lambda},$$

where $A(T)_\lambda$ denotes the elements of $A(T)$ of type λ.

The proposition also implies Theorem 2, as

$$\sum_T 1 = \sum_T \sum_\lambda \frac{|A(T)_\lambda|}{|A(T)|} = \sum_\lambda \sum_T \frac{|A(T)_\lambda|}{|A(T)|}.$$

If $\lambda = 1^n$, then $|A(T)_\lambda| = 1$ for all $T \in B_n$, so the proposition is saying that

$$\sum_T \frac{1}{|A(T)|} = \frac{(2n-3)!!}{n!} = \frac{c_{n-1}}{2^{n-1}}.$$

This is equivalent to $\sum_T 2^{n-1}/|A(T)| = c_{n-1}$. Since $2^{n-1}/|A(T)|$ counts all plane binary trees isomorphic to T, this is just the well-known fact that there are c_{n-1} plane binary trees with n leaves.

For a general λ, however, the proposition is far from obvious. What we need is a recursion satisfied by the expression on the right, analogous to the recursion $c_n = c_0 c_{n-1} + c_1 c_{n-1} + \cdots + c_{n-1} c_0$ for Catalan numbers.

Lemma 5. *For a nonempty subset $S = \{i_1 < i_2 < \ldots < i_k\}$ of positive integers define*

$$r_S(x_1, x_2, \ldots) = (x_{i_2} + \cdots + x_{i_k} - 1)(x_{i_3} + \cdots + x_{i_k} - 1) \cdots (x_{i_{k-1}} + x_{i_k} - 1)(x_{i_k} - 1). \tag{6}$$

Let $n \geq 2$, let $\mathbf{x}$ denote variables $x_1, x_2, \dots$, and let $\mathbf{x}/2$ denote $x_1/2, x_2/2, \dots$. Then

$$r_{[n]}(\mathbf{x}) = 2^{n-1} r_{[n]}(\mathbf{x}/2) + \sum_{1 \in S \subsetneq [n]} r_S(\mathbf{x}) \cdot r_{[n] \setminus S}(\mathbf{x}).$$

Example. For $n = 3$, the lemma says that

$$(x_2 + x_3 - 1)(x_3 - 1) = (x_2 + x_3 - 2)(x_3 - 2) + 1 \cdot (x_3 - 1) + (x_2 - 1) \cdot 1 + (x_3 - 1) \cdot 1,$$

where the last three terms on the right-hand side correspond to subsets $\{1\}$, $\{1, 2\}$, and $\{1, 3\}$, respectively. As another example, take $x_i = 2$ for all i. Then $r_S(\mathbf{x}) = (2|S| - 3)!!$ (where we interpret $(-1)!!$ as 1), $r_S(\mathbf{x}/2) = 0$, and by the obvious symmetry of S and $[n] \setminus S$ the lemma yields

$$2 \cdot (2n - 3)!! = \sum_{k=1}^{n-1} \binom{n}{k} (2k - 3)!!(2n - 2k - 3)!!,$$

which is equivalent to the standard recurrence for Catalan numbers.

Proof of Lemma 5. The proof is by induction on n. For $n = 2$, the statement is simply $x_2 - 1 = (x_2 - 2) + 1 \cdot 1$. Assume that the statement holds for $n - 1$, and let us prove it for n. Both sides are linear functions in x_2, so it is sufficient to prove that they have the same coefficient at x_2 and that they give the same result for one value of x_2.

The coefficient of x_2 in $r_{[n]}(\mathbf{x})$ (resp., $2^{n-1} r_{[n]}(\mathbf{x}/2)$) is clearly $r_{[2,n]}(\mathbf{x})$ (resp., $2^{n-2} r_{[2,n]}(\mathbf{x}/2)$). On the other hand, $r_S(\mathbf{x}) \cdot r_{[n] \setminus S}(\mathbf{x})$ contains x_2 if and only if $2 \in S$, in which case the coefficient at x_2 is $r_{S \setminus \{1\}}(\mathbf{x}) \cdot r_{[2,n] \setminus S}(\mathbf{x})$. The coefficients on both sides are equal by induction.

Plug the value $x_2 = 2 - x_3 - \cdots - x_n$ into both sides. Clearly, the left-hand side becomes $r_{[n] \setminus \{2\}}(\mathbf{x})$. It is easy to see that if $2 \in S$, then $r_S(\mathbf{x}) \cdot r_{[n] \setminus S}(\mathbf{x}) + r_{S \setminus \{2\}}(\mathbf{x}) \cdot r_{([n] \setminus S) \cup \{2\}}(\mathbf{x}) = 0$. That means that all the terms in the summation cancel out except $r_{[n] \setminus \{2\}}(\mathbf{x}) \cdot r_{\{2\}}(\mathbf{x}) = r_{[n] \setminus \{2\}}(\mathbf{x})$. Obviously, $r_{[n]}(\mathbf{x}/2) = 0$, so the right-hand side also equals $r_{[n] \setminus \{2\}}(\mathbf{x})$. □

Proof of Proposition 4. Say λ is a binary partition of n. The proof is by induction on n. For $n = 1$, the statement is obvious. Assume that the statement holds for all binary partitions up to size $n - 1$. Our task is to show

$$\sum_T \frac{|A(T)_\lambda|}{|A(T)|} = \frac{r_{[\ell(\lambda)]}(2\lambda_1, 2\lambda_2, 2\lambda_3, \dots)}{z_\lambda}$$

by showing the left hand side satisfies a recurrence similar to (6).

Given $T \in B_n$, let T_1 and T_2 be the subtrees of the root in T. Fix a labeling on the leaves of T such that the leaves of T_1 are labeled $[1, k]$ and the leaves of T_2 are labeled $[k + 1, n]$. Consider each $A(T_i)$ to be a subgroup of the permutations of the leaf labels for T_i. We can obtain a permutation of type λ in $A(T)$ in one of two ways. First, we can choose permutations $w_1 \in A(T_1), w_2 \in A(T_2)$ of types λ^1 and λ^2, then $w_1 w_2$ is a permutation of $A(T)$ of type λ. Second, if all parts of λ are at least 2 and $T_1 = T_2$ (and in particular $n = 2k$), we can choose an arbitrary permutation $w_1 \in A(T_1)$ and another permutation $w_2 \in A(T_1)$ specifically of type $\lambda/2 := (\lambda_1/2, \lambda_2/2, \dots)$ and construct a permutation $w \in A(T)$ of cycle type λ as follows. Say $f : [1, k] \longrightarrow [k + 1, n]$ mapping i to $i + k$ induces an isomorphism of T_1 and T_2. Define the "tree flip permutation" π to be the product of the transpositions interchanging i with $f(i)$ for all $1 \leq i \leq k$. Now take the product

$$w = \pi w_1 \pi w_1^{-1} \pi w_2.$$

It is clear that $w \in A(T)$ since it is the product of permutations in $A(T)$. Observe also that the cycles of w are constructed so the leaf labels of T_1 interleave the leaf labels of T_2 in the cycles of w_2 so w will have cycle type λ. For example, if $\lambda = (6, 4)$, then $|\lambda| = 10$ and $\pi = (1\ 6)(2\ 7)(3\ 8)(4\ 9)(5\ 10)$. If we choose $w_1 = (1\ 4)(2\ 5)(3)$ and $w_2 = (6\ 9\ 7)(8\ 10)$ then $w = \pi w_1 \pi w_1^{-1} \pi w_2 = (6\ 1\ 9\ 5\ 7\ 4)(8\ 2\ 10\ 3)$, all in cycle notation. Also, every element of $A(T)$ is constructed in one of these two ways.

We need to be careful to differentiate between the cases when the subtrees T_1, T_2 are different and when they are equivalent. We have

$$\begin{aligned}\sum_T \frac{|A(T)_\lambda|}{|A(T)|} &= \sum_{T_1 > T_2} \frac{|A(T)_\lambda|}{|A(T)|} + \sum_{T_1 = T_2} \frac{|A(T)_\lambda|}{|A(T)|} \\ &= \sum_{T_1 > T_2} \left(\sum_{\lambda^1 \cup \lambda^2 = \lambda} \frac{|A(T_1)_{\lambda^1}| \cdot |A(T_2)_{\lambda^2}|}{|A(T_1)| \cdot |A(T_2)|} \right) \\ &\quad + \sum_{T_1} \frac{\left(\sum_{\lambda^1 \cup \lambda^2 = \lambda} |A(T_1)_{\lambda^1}| \cdot |A(T_1)_{\lambda^2}|\right) + |A(T_1)| \cdot |A(T_1)_{\lambda/2}|}{2|A(T_1)|^2}\end{aligned}$$

or equivalently

$$\begin{aligned}2 \sum_{T \in B_n} \frac{|A(T)_\lambda|}{|A(T)|} = &\sum_{T_1 \in B_{n/2}} \frac{|A(T_1)_{\lambda/2}|}{|A(T_1)|} \\ &+ \sum_{\lambda^1 \cup \lambda^2 = \lambda} \left(\sum_{T_1 \in B_{|\lambda^1|}} \frac{|A(T_1)_{\lambda^1}|}{|A(T_1)|} \right) \left(\sum_{T_2 \in B_{|\lambda^2|}} \frac{|A(T_2)_{\lambda^2}|}{|A(T_2)|} \right).\end{aligned} \tag{7}$$

Let

$$q_\lambda = \frac{\prod_{i=2}^{\ell(\lambda)} (2(\lambda_i + \cdots + \lambda_{\ell(\lambda)}) - 1)}{z_\lambda} = \frac{r_{[\ell(\lambda)]}(2\lambda_1, 2\lambda_2, 2\lambda_3, \dots)}{z_\lambda};$$

the notation also makes sense if $\lambda_{\ell(\lambda)} = 1/2$, as in that case $q_\lambda = 0$. By the induction hypothesis and (7), it suffices to prove that

$$2q_\lambda = q_{\lambda/2} + \sum_{\lambda^1 \cup \lambda^2 = \lambda} q_{\lambda^1} \cdot q_{\lambda^2}. \tag{8}$$

After multiplying both sides by z_λ, this is

$$\begin{aligned} 2\prod_{i=2}^{\ell(\lambda)}(2(\lambda_i + \cdots + \lambda_{\ell(\lambda)}) - 1) &= 2^{\ell(\lambda)} \prod_{i=2}^{\ell(\lambda)}(\lambda_i + \cdots + \lambda_{\ell(\lambda)} - 1) \\ &+ \sum_{\lambda^1 \cup \lambda^2 = \lambda} \binom{\lambda}{\lambda^1, \lambda^2} \cdot \prod_{i=2}^{\ell(\lambda^1)} (2(\lambda^1_i + \cdots + \lambda^1_{\ell(\lambda^1)}) - 1) \cdot \prod_{i=2}^{\ell(\lambda^2)} (2(\lambda^2_i + \cdots + \lambda^2_{\ell(\lambda^2)}) - 1), \end{aligned}$$

where $\binom{\lambda}{\lambda^1,\lambda^2} = \prod_i \binom{m_i(\lambda)}{m_i(\lambda^1)}$. This equality holds by Lemma 5 with $x_i = 2\lambda_i$. □

We conclude this section with the proof of Theorem 3.

Proof of Theorem 3. Let $\mathbf{T} = (T_1, T_2, \ldots, T_k)$ be an ordered list of binary trees in B_n. Define $C^{\mathbf{T}}$ to be the set of "multicosets" of $\mathfrak{S}_n$ with respect to $A(T_1) \times A(T_2) \times \cdots \times A(T_k)$. More concretely, given $(w_1, \ldots, w_{k-1}), (w'_1, \ldots, w'_{k-1}) \in (\mathfrak{S}_n)^{k-1}$, we say $(w_1, \ldots, w_{k-1}) \equiv_{\mathbf{T}} (w'_1, \ldots, w'_{k-1})$ provided there exist $t_i \in A(T_i)$ such that $w_i = t_i w'_i t_{i+1}$ for all $i = 1, \ldots, k-1$. Then, $C^{\mathbf{T}}$ is the set of equivalence classes modulo $\equiv_{\mathbf{T}}$. By definition, the number of tangled chains of length k and size n, denoted $t(k,n)$, is given by

$$t(k,n) = \sum |C^{\mathbf{T}}| \tag{9}$$

where the sum is over all ordered lists $\mathbf{T} = (T_1, T_2, \ldots, T_k)$ of trees $T_i \in B_n$.

Fix a particular list of trees $\mathbf{T} = (T_1, T_2, \ldots, T_k)$, and let $C^{\mathbf{T}}(w_1, \ldots, w_{k-1})$ be the multicoset in $C^{\mathbf{T}}$ containing $(w_1, \ldots, w_{k-1})$. Clearly,

$$|C^{\mathbf{T}}| = \sum_{w_1 \in \mathfrak{S}_n} \sum_{w_2 \in \mathfrak{S}_n} \cdots \sum_{w_{k-1} \in \mathfrak{S}_n} \frac{1}{|C^{\mathbf{T}}(w_1, \ldots, w_{k-1})|}.$$

We give a recurrence for $|C^{\mathbf{T}}(w_1, \ldots, w_{k-1})|$ in terms of the following subgroup. Let $A(C^{\mathbf{T}}(w_1, \ldots, w_{k-1}))$ be the subgroup of all $t_1 \in A(T_1)$ such that there exist $t_i \in A(T_i)$ for $2 \le i \le k$ satisfying $w_i = t_i w_i t_{i+1}$ for all $i = 1, \ldots, k-1$. In this case, $(t_1 w_1, w_2, \ldots, w_{k-1}) \equiv_{\mathbf{T}} (w_1, w_2, \ldots, w_{k-1})$ so we think of $A(C^{\mathbf{T}}(w_1, \ldots, w_{k-1}))$ as the "left automorphism group" of $C^{\mathbf{T}}(w_1, \ldots, w_{k-1})$. Observe that

$$\begin{aligned} &A(C^{\mathbf{T}}(w_1, \ldots, w_{k-1})) \\ &\quad = A(T_1) \cap w_1 A(T_2) w_1^{-1} \cap \cdots \cap w_1 w_2 \cdots w_{k-1} A(T_k) w_{k-1}^{-1} \cdots w_2^{-1} w_1^{-1}, \end{aligned}$$

so

$$
\begin{aligned}
&|A(C^{\mathbf{T}}(w_1,\ldots,w_{k-1}))| \\
&\quad = \sum_{i=1}^{k} \sum_{t_i \in A(T_i)} [\![t_1 = w_1 t_2 w_1^{-1}]\!] \cdot [\![t_2 = w_2 t_3 w_2^{-1}]\!] \cdots [\![t_{k-1} = w_{k-1} t_k w_{k-1}^{-1}]\!].
\end{aligned}
$$

Now let $\mathbf{T}' = (T_2, \ldots, T_k)$. For each $(v_2, \ldots, v_{k-1}) \in C^{\mathbf{T}'}(w_2, \ldots, w_{k-1})$, we can prepend a v_1 to create a distinct element $(v_1, v_2, \ldots, v_{k-1}) \in C^{\mathbf{T}}(w_1, \ldots, w_{k-1})$ exactly when v_1 is in $A(T_1)w_1A(C^{\mathbf{T}'}(w_2, \ldots, w_{k-1}))$ which is again a double coset of $\mathfrak{S}_n$. Thus, by the formula (2) for double cosets we have

$$
\begin{aligned}
|C^{\mathbf{T}}(w_1,\ldots,w_{k-1})| &= \frac{|A(T_1)| \cdot |A(C^{\mathbf{T}'}(w_2,\ldots,w_{k-1}))|}{|A(C^{\mathbf{T}}(w_1,\ldots,w_{k-1}))|} \cdot |C^{\mathbf{T}'}(w_2,\ldots,w_{k-1})| \\
&= \frac{|A(T_1)| \cdot |A(T_2)| \cdots |A(T_k)|}{|A(C^{\mathbf{T}}(w_1,\ldots,w_{k-1}))|}
\end{aligned}
$$

by induction on k. Therefore,

$$
|C^{\mathbf{T}}| = \sum_{w_1 \in \mathfrak{S}_n} \sum_{w_2 \in \mathfrak{S}_n} \cdots \sum_{w_{k-1} \in \mathfrak{S}_n} \frac{|A(C^{\mathbf{T}}(w_1,\ldots,w_{k-1}))|}{|A(T_1)| \cdot |A(T_2)| \cdots |A(T_k)|}, \tag{10}
$$

where the denominators do not depend on the w_i's.

Focusing on the sum in the numerator in (10), we have

$$
\begin{aligned}
&\sum_{(w_1,w_2,\ldots,w_{k-1})} |A(C^{\mathbf{T}}(w_1,\ldots,w_{k-1}))| \\
&\quad = \sum_{(w_1,w_2,\ldots,w_{k-1})} \sum_{t_1 \in A(T_1)} \cdots \sum_{t_k \in A(T_k)} [\![t_1 = w_1 t_2 w_1^{-1}]\!] \cdots [\![t_{k-1} = w_{k-1} t_k w_{k-1}^{-1}]\!] \\
&\quad = \sum_{t_1 \in A(T_1)} \cdots \sum_{t_k \in A(T_k)} \sum_{(w_1,w_2,\ldots,w_{k-1})} [\![t_1 = w_1 t_2 w_1^{-1}]\!] \cdots [\![t_{k-1} = w_{k-1} t_k w_{k-1}^{-1}]\!]
\end{aligned}
$$

and so with similar logic as before, noting that the summand will be nonzero exactly when $t_1, t_2, \ldots, t_k$ are all of the same conjugacy type λ,

$$
|C^{\mathbf{T}}| = \frac{\sum_\lambda |A(T_1)_\lambda| \cdot |A(T_2)_\lambda| \cdots |A(T_k)_\lambda| \cdot z_\lambda^{k-1}}{|A(T_1)| \cdot |A(T_2)| \cdots |A(T_k)|}. \tag{11}
$$

Plugging (11) into (9), we obtain

$$
\begin{aligned}
t(k,n) &= \sum_{(T_1,\ldots,T_k)} \frac{\sum_\lambda |A(T_1)_\lambda| \cdot |A(T_2)_\lambda| \cdots |A(T_k)_\lambda| \cdot z_\lambda^{k-1}}{|A(T_1)| \cdot |A(T_2)| \cdots |A(T_k)|} \\
&= \sum_\lambda z_\lambda^{k-1} \cdot \left(\sum_{T \in B_n} \frac{|A(T)_\lambda|}{|A(T)|} \right)^k,
\end{aligned}
$$

and Theorem 3 now follows from Proposition 4. □

4. Random generation of tanglegrams and inequivalent binary trees

In this section, we describe an algorithm in 3 stages to produce a random tanglegram in T_n. The stages are based on Equation (4) and the proof of Proposition 4. A similar algorithm is also described to choose a random binary tree with n leaves. In this section, "random" will mean uniformly at random unless specified otherwise.

Recall from Section 3 that if T is a tree with equivalent left and right subtrees, we denote by π the "tree flip permutation" between the subtrees. Also, for a partition λ, we defined

$$q_\lambda = \frac{\prod_{i=2}^{\ell(\lambda)} (2(\lambda_i + \cdots + \lambda_{\ell(\lambda)}) - 1)}{z_\lambda}.$$

The q_λ notation also makes sense if $\lambda_{\ell(\lambda)} = 1/2$, as in that case $q_\lambda = 0$.

We will frequently require an algorithm to generate a binary partition λ with probability q_λ. This is simply done by generating all binary partitions of n, computing q_λ for each, and then sampling from this categorical distribution using inverse transform sampling.

Algorithm 1 *(Random generation of $w \in A(T)$).*

Input: Binary tree $T \in B_n$.

Procedure: If T is the tree with one vertex, let w be the unique element of $A(T)$. Otherwise, the root of T has subtrees T_1 and T_2. Assume the leaves of T_1 are labeled $[1, k]$ and the leaves of T_2 are labeled $[k+1, n]$. Use the algorithm recursively to produce $w_i \in A(T_i)$, $i = 1, 2$ where $A(T_1)$ is a subset of the permutations of $[1, n]$ which fix $[k+1, n]$ and $A(T_2)$ is a subset of the permutations of $[1, n]$ which fix $[1, k]$. Construct w as follows.

- If $T_1 \neq T_2$, set $w = w_1 w_2$.
- If $T_1 = T_2$, choose either $w = w_1 w_2$ or $w = \pi w_1 w_2$ with equal probability.

Output: Permutation $w \in A(T)$.

Algorithm 2 *(Random generation of T with non-empty $A(T)_\lambda$ and $w \in A(T)_\lambda$).*

Input: Binary partition λ of n.

Procedure: If $n = 1$, let T be the tree with one vertex, and let w be the unique element of $A(T)$.

Otherwise, pick a subdivision (λ^1, λ^2) from $\{(\lambda^1, \lambda^2) \colon \lambda^1 \cup \lambda^2 = \lambda\} \cup \{(\lambda/2, \lambda/2)\}$, where (λ^1, λ^2) is chosen with probability proportional to $q_{\lambda^1} q_{\lambda^2}$ and $(\lambda/2, \lambda/2)$ with probability proportional to $q_{\lambda/2}$.

- If $\lambda^1, \lambda^2 \neq \lambda/2$, use the algorithm recursively to produce trees T_1, T_2 and permutations $w_1 \in A(T_1)_{\lambda^1}$, $w_2 \in A(T_2)_{\lambda^2}$. If necessary, switch $T_1 \leftrightarrow T_2$, $w_1 \leftrightarrow w_2$ so that $T_1 \geq T_2$. Let $T = (T_1, T_2)$, $w = w_1 w_2$.
- If $\lambda^1 = \lambda^2 = \lambda/2$, use the algorithm recursively to produce a tree T_1 and a permutation $w_2 \in A(T_1)_{\lambda/2}$, and use Algorithm 1 to produce a permutation $w_1 \in A(T_1)$. Let $T = (T_1, T_1)$ and $w = \pi w_1 \pi w_1^{-1} \pi w_2$.

Output: Binary tree T and permutation $w \in A(T)_\lambda$.

Algorithm 3 *(Random generation of tanglegrams).*

Input: Integer n.

Procedure: Pick a random binary partition λ of n with probability proportional to $z_\lambda q_\lambda^2$ where $t_n = \sum z_\lambda q_\lambda^2$. Use Algorithm 2 twice to produce random trees T and S and permutations $u \in A(T)_\lambda$, $v \in A(S)_\lambda$. Among the permutations w for which $u = wvw^{-1}$, pick one at random from the z_λ possibilities.

Output: Binary trees T and S and double coset $A(T)wA(S)$, or equivalently (T, w, S).

Algorithm 4 *(Random generation of $T \in B_n$).*

Input: Integer n.

Procedure: Pick a random binary partition λ of n with probability proportional to q_λ. Use Algorithm 2 to produce a random tree T (and a permutation $u \in A(T)_\lambda$).

Output: Binary tree T.

Algorithm 4 is not the first of its kind, see also [10].

Algorithm 5 *(Random generation of tangled chains).*

Input: Positive integers k and n.

Procedure: Pick a random binary partition λ of n with probability proportional to $z_\lambda^{k-1} q_\lambda^k$ where $t(k, n) = \sum z_\lambda^{k-1} q_\lambda^k$. Use Algorithm 2 k times to produce random trees T_i and permutations $u_i \in A(T_i)_\lambda$ for $i = 1, \ldots, k$. Among the permutations w_i for which $u_i = w_i u_{i+1} w_i^{-1}$, pick one uniformly at random for each $i = 1, \ldots, k-1$.

Output: $(T_1, \ldots, T_k)$ and $(w_1, \ldots, w_{k-1})$.

Theorem 6. *For any positive integer n, the following hold.*

- *Algorithm 1 produces every permutation $w \in A(T)$ with probability $\frac{1}{|A(T)|}$.*
- *Algorithm 2 produces every pair (T, w), where $w \in A(T)_\lambda$, with probability $\frac{1}{|A(T)| \cdot q_\lambda}$.*
- *Algorithm 3 produces every tanglegram with probability $\frac{1}{t_n}$.*
- *Algorithm 4 produces every inequivalent binary tree with probability $\frac{1}{b_n}$.*
- *Algorithm 5 produces every tangled chain of length k of trees in B_n with probability $\frac{1}{t(k,n)}$.*

Proof. The first two proofs are by induction, with the case $n = 1$ being obvious. The induction for Algorithm 1 is trivial.

For Algorithm 2, say that we are given a binary partition λ, a tree T with $n = |\lambda|$ leaves, and $w \in A(T)_\lambda$. We compute the probability that Algorithm 2 produces T and w. Assume first that $T_1 > T_2$ are the subtrees of T. In particular, that means that w can be written uniquely as $w_1 w_2$, where $w_1 \in A(T_1)$ and $w_2 \in A(T_2)$. Say that w_i is of type λ^i; we must have $\lambda = \lambda^1 \cup \lambda^2$. If $\lambda^1 \neq \lambda^2$, there are two ways in which Algorithm 2 can produce (T, w): either we partition λ into (λ^1, λ^2), and then the algorithm produces (T_1, w_1) and (T_2, w_2), or we partition λ into (λ^2, λ^1), then the algorithm produces (T_2, w_2) and (T_1, w_1), and finally switches $T_1 \leftrightarrow T_2$, $w_1 \leftrightarrow w_2$. Since T_1 and T_2 are chosen independently, we can apply (8) and induction to obtain the probability that (T, w) is chosen, namely

$$2 \cdot \frac{q_{\lambda^1} q_{\lambda^2}}{2q_\lambda} \cdot \frac{1}{|A(T_1)| \cdot q_{\lambda^1}} \cdot \frac{1}{|A(T_2)| \cdot q_{\lambda^2}} = \frac{1}{|A(T_1)| \cdot |A(T_2)| \cdot q_\lambda} = \frac{1}{|A(T)| \cdot q_\lambda}.$$

If $\lambda^1 = \lambda^2$, but $T_1 \neq T_2$, there are again two ways in which Algorithm 2 can produce (T, w): we must partition λ into (λ^1, λ^1), and then it can either produce (T_1, w_1) and (T_2, w_2) or (T_2, w_2) and (T_1, w_1); in the latter case it switches $T_1 \leftrightarrow T_2$, $w_1 \leftrightarrow w_2$. Similarly, the probability is $\frac{1}{|A(T)| \cdot q_\lambda}$.

Now assume that $T_1 = T_2$. Either w can be written as $w_1 w_2$, where $w_1 \in A(T_1)_{\lambda^1}$ and $w_2 \in A(T_2)_{\lambda^2}$, or as $\pi w_2 \pi w_2^{-1} \pi w_1$, where $w_1 \in A(T_1)_{\lambda/2}$ and $w_2 \in A(T_1)$. In the first case, (T, w) is produced with probability

$$\frac{q_{\lambda^1} q_{\lambda^2}}{2q_\lambda} \cdot \frac{1}{|A(T_1)| \cdot q_{\lambda^1}} \cdot \frac{1}{|A(T_1)| \cdot q_{\lambda^2}} = \frac{1}{2 \cdot |A(T_1)|^2 \cdot q_\lambda} = \frac{1}{|A(T)| \cdot q_\lambda}.$$

In the second case, it is produced with probability

$$\frac{q_{\lambda/2}}{2q_\lambda} \cdot \frac{1}{|A(T_1)| \cdot q_{\lambda/2}} \cdot \frac{1}{|A(T_1)|} = \frac{1}{2 \cdot |A(T_1)|^2 \cdot q_\lambda} = \frac{1}{|A(T)| \cdot q_\lambda}.$$

This finishes the case for Algorithm 2.

The proof of the statement for Algorithm 3 is essentially just a rewriting of the proof from Section 3; we include it for completeness. We are given n and a tanglegram (T, w, S) with T and S binary trees with n leaves, $C = A(T)wA(S)$ the double coset containing w with respect to $A(T)$ and $A(S)$, and we want to prove that $P(T, S, C)$, the probability that this triple is produced by Algorithm 3, is $1/t_n$.

We proved that $\sum z_\lambda q_\lambda^2 = t_n$, so the probability of choosing a binary partition λ is $z_\lambda q_\lambda^2 / t_n$. So we have

$$P(T, S, C) = \sum_\lambda \frac{z_\lambda q_\lambda^2}{t_n} P(T, S, C|\lambda),$$

where $P(T,S,C|\lambda)$ is the conditional probability that (T,S,C) is produced if λ is chosen. We can further condition the probability: $P(T,S,C|\lambda) = \sum P(T,S,C|u,v,T,S,\lambda) \cdot P(u,v,T,S|\lambda)$, where the sum is over $u \in A(T)_\lambda$, $v \in A(S)_\lambda$. Furthermore,

$$P(T,S,C|u,v,T,S,\lambda) = P(C|u,v) \text{ and } \; P(u,v,T,S|\lambda) = P(T,u|\lambda) \cdot P(S,v|\lambda),$$

and so

$$\begin{aligned} P(T,S,C) &= \sum_\lambda \frac{z_\lambda q_\lambda^2}{t_n} \sum_{u \in A(T)_\lambda} \sum_{v \in A(S)_\lambda} P(C|u,v) \cdot \frac{1}{|A(T)| \cdot q_\lambda} \cdot \frac{1}{|A(S)| \cdot q_\lambda} \\ &= \frac{1}{t_n} \cdot \sum_\lambda \frac{z_\lambda}{|A(T)| \cdot |A(S)|} \cdot \sum_{u \in A(T)_\lambda} \sum_{v \in A(S)_\lambda} \frac{|C \cap B^{u,v}|}{|B^{u,v}|}, \end{aligned}$$

where $B^{u,v} = \{w \in \mathfrak{S}_n \colon u = wvw^{-1}\}$. We know that $|B^{u,v}| = z_\lambda$, so

$$\begin{aligned} P(T,S,C) &= \frac{1}{t_n} \cdot \sum_\lambda \frac{1}{|A(T)| \cdot |A(S)|} \sum_{u \in A(T)_\lambda} \sum_{v \in A(S)_\lambda} \sum_{w \in C} [\![u = wvw^{-1}]\!] \\ &= \frac{1}{t_n} \cdot \sum_{w \in C} \sum_\lambda \frac{1}{|A(T)| \cdot |A(S)|} \sum_{u \in A(T)_\lambda} \sum_{v \in A(S)_\lambda} [\![u = wvw^{-1}]\!] \\ &= \frac{1}{t_n} \cdot \sum_{w \in C} \sum_\lambda \frac{|A(T)_\lambda \cap wA(S)_\lambda w^{-1}|}{|A(T)| \cdot |A(S)|} = \frac{1}{t_n} \cdot \sum_{w \in C} \frac{|A(T) \cap wA(S)w^{-1}|}{|A(T)| \cdot |A(S)|} \\ &= \frac{1}{t_n} \cdot \sum_{w \in C} \frac{1}{|C_w|} = \frac{1}{t_n}. \end{aligned}$$

Finally, let us prove the statement for Algorithm 4. We have

$$P(T) = \sum_\lambda P(T|\lambda) \cdot P(\lambda) = \sum_\lambda \frac{|A(T)_\lambda|}{|A(T)| \cdot q_\lambda} \cdot \frac{q_\lambda}{b_n} = \frac{1}{b_n} \cdot \frac{\sum_\lambda |A(T)_\lambda|}{|A(T)|} = \frac{1}{b_n},$$

which proves that Algorithm 4 produces every inequivalent binary tree with the same probability. The proof for Algorithm 5 is similar to Algorithms 3 and 4 so we omit the formal proof. □

5. Asymptotic expansion of t_n

In this section, we use Theorem 1 to obtain another formula for t_n and several formulas to approximate t_n for large n.

Corollary 7. *We have*

$$t_n = \frac{c_{n-1}^2 n!}{4^{n-1}} \sum_\mu \frac{n(n-1)\cdots(n-|\mu|+1)}{z_\mu \cdot \prod_{i=1}^{\ell(\mu)} \prod_{j=1}^{\mu_i - 1} (2n - 2(\mu_1 + \cdots + \mu_{i-1}) - 2j - 1)^2}, \tag{12}$$

where the sum is over binary partitions μ *with all parts equal to a positive power of* 2 *and* $|\mu| \leq n$ *including the empty partition in which case the summand is 1.*

Proof. Every binary partition λ of size n can be expressed as $\mu 1^{n-|\mu|}$, where all parts of μ are at least 2. We have $z_\lambda = z_\mu (n-|\mu|)!$ and

$$\begin{aligned}\prod_{i=2}^{\ell(\lambda)} \left(2(\lambda_i + \cdots + \lambda_{\ell(\lambda)}) - 1\right) &= \prod_{i=1}^{\ell(\lambda)-1} (2(n - \lambda_1 - \cdots - \lambda_i) - 1) \\ &= \prod_{i=1}^{\ell(\mu)-1} (2(n - \mu_1 - \cdots - \mu_i) - 1) \cdot (2n - 2|\mu| - 1)!! \\ &= \frac{(2n-3)!!}{\prod_{i=1}^{\ell(\mu)} \prod_{j=1}^{\mu_i - 1} (2n - 2(\mu_1 + \cdots + \mu_{i-1}) - 2j - 1)}.\end{aligned}$$

Since $(2n-3)!!/n! = c_{n-1}/2^{n-1}$, (12) is an equivalent way to express the number of tanglegrams. □

The first few terms of the sum corresponding to partitions $\emptyset$, (2), (4), (2, 2), (4, 2), (2, 2, 2), (8) are

$$\begin{aligned}1 &+ \frac{n(n-1)}{2(2n-3)^2} + \frac{n(n-1)(n-2)(n-3)}{4(2n-3)^2(2n-5)^2(2n-7)^2} + \frac{n(n-1)(n-2)(n-3)}{8(2n-3)^2(2n-7)^2} \\ &+ \frac{n(n-1)(n-2)(n-3)(n-4)(n-5)}{8(2n-3)^2(2n-5)^2(2n-7)^2(2n-11)^2} + \frac{n(n-1)(n-2)(n-3)(n-4)(n-5)}{48(2n-3)^2(2n-7)^2(2n-11)^2} \\ &+ \frac{n(n-1)(n-2)(n-3)(n-4)(n-5)(n-6)(n-7)}{8(2n-3)^2(2n-5)^2(2n-7)^2(2n-9)^2(2n-11)^2(2n-13)^2(2n-15)^2}.\end{aligned}$$

Corollary 8. *We have*

$$\frac{t_n}{n!} \sim \frac{e^{\frac{1}{8}} c_{n-1}^2}{4^{n-1}} \sim \frac{e^{\frac{1}{8}} 4^{n-1}}{\pi n^3} \qquad \text{and} \qquad t_n \sim \frac{2^{2n-\frac{3}{2}} \cdot n^{n-\frac{5}{2}}}{\sqrt{\pi} \cdot e^{n-\frac{1}{8}}}.$$

We can also compute approximations of higher degree. For example, we have

$$\begin{aligned}t_n &= \frac{e^{\frac{1}{8}} c_{n-1}^2 n!}{4^{n-1}} \cdot \left(1 + \frac{1}{4\,n} + \frac{137}{256\,n^2} + \frac{1285}{1024\,n^3} + \frac{456017}{131072\,n^4} + \frac{6140329}{524288\,n^5} + O\left(n^{-6}\right)\right) \\ &= \frac{2^{2n-\frac{3}{2}} \cdot n^{n-\frac{5}{2}}}{\sqrt{\pi} \cdot e^{n-\frac{1}{8}}} \cdot \left(1 + \frac{13}{12\,n} + \frac{3089}{2304\,n^2} + \frac{931423}{414720\,n^3} + \frac{826301423}{159252480\,n^4}\right. \\ &\qquad \left. + \frac{211060350013}{13377208320\,n^5} + O\left(n^{-6}\right)\right).\end{aligned}$$

Sketch of proof. The crucial observation is that

$$\frac{n(n-1)\cdots(n-|\mu|+1)}{z_\mu\cdot\prod_{i=1}^{\ell(\mu)}\prod_{j=1}^{\mu_i-1}(2n-2(\mu_1+\cdots+\mu_{i-1})-2j-1)^2}\sim\frac{n^{|\mu|}}{z_\mu\cdot(2n)^{2(|\mu|-\ell(\mu))}}$$
$$=\frac{1}{2^{2(|\mu|-\ell(\mu))}\cdot z_\mu\cdot n^{|\mu|-2\ell(\mu)}}.$$

So, to find an asymptotic approximation of order $O(n^{-2m})$ or $O(n^{-2m-1})$, we only have to consider partitions μ with $|\mu|-2\ell(\mu)\leq 2m$ in Equation (12). For $m=0$, we only consider partitions of the type $22\cdots 2$. The contribution of $\mu=2^k$ is $1/(2^{2k}2^k k!)$, and the sum converges to $\sum_k\frac{1}{2^{3k}k!}=e^{\frac{1}{8}}$.

Similarly, the coefficient of n^{-1} can be obtained by considering the coefficient of n^{-1} in each of these terms, and the higher terms by considering in turn partitions of type 42^k, 4^22^k, 4^32^k, 82^k, etc. The last expansion is obtained by considering the asymptotic expansions of c_{n-1} and $n!$. □

6. A recurrence for enumerating tanglegrams and tangled chains

In this section, we give a recurrence for computing t_n. Recall that for each nonempty binary partition λ, we can construct its *multiplicity vector* $m^\lambda=(m_0,m_1,m_2,m_3,\ldots)$ where m_i is the number of times 2^i occurs in λ. The map $\lambda\mapsto m^\lambda$ is a bijection from binary partitions to vectors of nonnegative integers with only finitely many nonzero entries. The quantity z_λ for a binary partition λ is easily expressed in terms of the multiplicities in m^λ as

$$z_\lambda=\prod_{h\geq 0}2^{h\cdot m_h}\ m_h!=\prod_{\substack{h\geq 0\\ m_h\neq 0}}\prod_{j=1}^{m_h}j\cdot 2^h$$

We will use the functions

$$f^2(s):=(2s-1)^2, \tag{13}$$

$$c(h,m,s):=\prod_{j=1}^{m}\frac{f^2(s+j\cdot 2^h)}{j\cdot 2^h}, \tag{14}$$

and

$$r(h,n,s):=\sum_{\substack{m=0\\(n-m)\text{ even}}}^{n}c(h,m,s)\ r\left(h+1,\frac{n-m}{2},s+m2^h\right) \tag{15}$$

with base cases

$$c(h,0,s)=r(h,0,s)=1. \tag{16}$$

Lemma 9. *For $n \geq 1$, the number of tanglegrams is*

$$t_n = \frac{r(0,n,0)}{f^2(n)},$$

which can be computed recursively using (15).

Proof. Let $\tilde{t}_n := (1-2n)^2 t_n$. By the main formula

$$\tilde{t}_n = \sum_{\lambda} \frac{\prod_{i=1}^{\ell(\lambda)} \left(2(\lambda_i + \cdots + \lambda_{\ell(\lambda)}) - 1\right)^2}{z_\lambda}, \tag{17}$$

where the sum is over binary partitions of n.

We will consider the contribution to (17) from the parts of the partition of size 2^h for each h separately. To do this we will need to keep track of the partial sums of parts smaller than 2^h. Let $s^\lambda = (s_0^\lambda, s_1^\lambda, \ldots)$ where $s_h^\lambda = \sum_{i=0}^{h-1} m_i 2^i$ and $s_0^\lambda = 0$. Then the contribution of the parts of size 2^h in λ to the corresponding term in (17) is the factor $c(h, m_h, s_h^\lambda)$. Using this notation, we have

$$\tilde{t}_n = \sum_{m^\lambda = (m_0, m_1, \ldots) \vdash n} c(0, m_0, 0) c(1, m_1, s_1^\lambda) c(2, m_2, s_2^\lambda) \cdots \tag{18}$$

where the sum is over binary partitions of n represented by their multiplicity vector.

Next consider the binary partitions with exactly j parts of size 1. Note $n-j$ must be even for this set to be nonempty. The binary partitions of n with exactly j parts equal to 1 are in bijection with the binary partitions of $\frac{n-j}{2}$, so

$$\tilde{t}_n = \sum_{\substack{m_0=0 \\ (n-m_0) \text{ even}}}^{n} c(0, m_0, 0) \sum_{(m_1, m_2, \ldots) \vdash \frac{n-m_0}{2}} c(1, m_1, m_0) c(2, m_2, m_0 + 2 \cdot m_1) \cdots . \tag{19}$$

Observe that the recurrence in (15) gives rise to the expansion

$$\begin{aligned} r(h,n,s) = \sum_{(m_h, m_{h+1}, \ldots) \vdash n} & c(h, m_h, s) c(h+1, m_{h+1}, s + m_h \cdot 2^h) \\ & \times c(h+2, m_{h+2}, s + m_h \cdot 2^h + m_{h+1} \cdot 2^{h+1}) \cdots \end{aligned}$$

where the sum is over binary partitions of n but the indexing is shifted so m_h is the number of parts of size 1. Thus,

$$\tilde{t}_n = \sum_{\substack{m=0 \\ (n-m) \text{ even}}}^{n} c(0, m, 0)\, r\left(1, \frac{n-m}{2}, m\right) = r(0, n, 0)$$

which completes the proof since $f^2(n) = (2n-1)^2$. □

We can extend the functions above to count tangled chains:

$$f^k(s) := (2s-1)^k, \tag{20}$$

$$c^k(h,m,s) := \prod_{j=1}^{m} \frac{f^k(s+j\cdot 2^h)}{j\cdot 2^h}, \tag{21}$$

and

$$r^k(h,n,s) := \sum_{\substack{m=0 \\ (n-m)\text{ even}}}^{n} c^k(h,m,s)\; r\left(h+1, \frac{n-m}{2}, s+m2^h\right) \tag{22}$$

with base cases

$$c^k(h,0,s) = r^k(h,0,s) = 1. \tag{23}$$

Then a proof very similar to the case $k=2$ also proves the following statement.

Corollary 10. *For $n \geq 1$, the number of tangled chains of length k is*

$$\frac{r^k(0,n,0)}{f^k(n)}$$

which can be computed recursively using (22).

7. Final remarks

7.1. Generating functions

It is known (and easy to prove) that the ordinary generating function for inequivalent trees satisfies the functional equation

$$B(x) = x + \frac{1}{2}\left(B(x)^2 + B(x^2)\right).$$

This is, of course, equivalent to a recurrence for the sequence b_n. Given that in this paper we prove both explicit formulas and recurrences for the numbers of tanglegrams and tangled chains, it makes sense to ask the following.

Question 1. *Does there exist a closed form or a functional equation for the generating function of tanglegrams or tangled chains?*

7.2. Number of cherries and other subtrees

Cherries play an important role in the literature on tanglegrams. For example, Charleston's analysis [3, pp. 325–326] suggests the following question.

Question 2. *What is the expected number of matched cherries in a random tanglegram?*

Computer experiments with random tanglegram generation suggest that the following is true.

Conjecture 1. *The expected number of cherries in the left tree in a random tanglegram converges to $n/4$.*

Conjecture 2. *The expected number of copies of the tree T in the left tree of a random tanglegram of size n is asymptotically equal to $2^{-(l+k-1)}n$, where l is the number of leaves of T and k is the number of symmetries of T, i.e. vertices with identical subtrees.*

Assuming the conjecture, for every tree T with l leaves and k symmetries, the number of copies of the tree with T as left and as right subtree in the left tree of a randomly chosen tanglegram asymptotically equals $2^{-(2l+(2k+1)-1)}n = 4^{-(l+k)}n$. So that would imply the following.

Conjecture 3. *Let $T' \in B_n$ be the left tree of a tanglegram chosen uniformly at random. The expected number of generators of $A(T')$ is asymptotically equal to*

$$\left(\sum_{T \in B_n} \frac{1}{4^{l(T)+k(T)}} \right) n.$$

It is not hard to see that the sum in the conjecture equals $f(\frac{1}{4})n$, where $f(x)$ is the function defined by $f(0) = 0$ and $f(x) = x + \frac{1}{2}f(x)^2 + (x - \frac{1}{2})f(x^2)$, or explicitly

$$f(x) = 1 - \sqrt{1 - 2x + (1 - 2x)\left(1 - \sqrt{1 - 2x^2 + (1 - 2x^2)\left(1 - \sqrt{1 - 2x^4 + \cdots}\right)}\right)}.$$

Note that the computation of $f(\frac{1}{4}) = 0.27104169360883278703...$ converges very rapidly: the number of correct digits roughly doubles after each step.

Remark. Based on a preprint version of this paper, all these conjectures were proved in [19] by Wagner and the second author. The main result in that paper is the more general statement that the two halves of a random tanglegram essentially look like two independently chosen random plane binary trees.

In a related development, Czabarka, Székely, and Wagner proved some results involving the crossing number of large random tanglegrams [6].

7.3. Connection with symmetric functions

The main theorems suggest that symmetric functions might be at play; note, for example, the similarity with the formula $h_n = \sum_\lambda z_\lambda^{-1} p_\lambda$, where h_n is the homogeneous symmetric function, p_λ the power sum symmetric function, and the sum is over all partitions of n.

Question 3. *Is there a connection between tanglegrams (or more generally tangled chains) and symmetric functions?*

Remark. Based on a preprint version of this paper, Ira Gessel pointed out that there is indeed a connection between symmetric functions and the enumeration of the ordered and unordered tanglegrams based on the theory of species. He has beautifully spelled out this connection. This approach leads to a simple formula for the number of unordered tanglegrams and a generating function for the number of unrooted tanglegrams along with several other variations on tanglegrams [13].

7.4. Variants on tanglegrams

Tanglegrams as described here fit in a set of more general setting of pairs of graphs with a bijection between certain subsets of the vertices (more completely described and motivated in [20]). One can also consider *unordered tanglegrams* by identifying (T, v, S) with (S, v^{-1}, T). For example, the 4th and 5th tanglegrams in Fig. 2 are equivalent as unordered tanglegrams, and so are the 8th and 10th. From this picture, the reader can verify that there are 10 unordered tanglegrams of size 4.

Because of reversibility assumptions for the continuous time Markov mutation models commonly used to reconstruct phylogenetic trees, unrooted trees are the most common output of phylogenetic inference algorithms. Thus another variant of tanglegrams involves using unrooted trees in place of rooted ones. The motivation for studying these variants comes from noting that many problems in computational phylogenetics such as distance calculation between trees [1] “factor” through a problem on tanglegrams.

Question 4. *Is there a nice formula for the number of*

- *unordered binary rooted tanglegrams,*
- *ordered binary unrooted tanglegrams, or*
- *unordered binary unrooted tanglegrams?*

These counts have been found up to 9 leaves (Table 1) by direct enumeration of double cosets [20]. Note that Gessel [13] has made significant progress on all of these questions using the theory of species, but explicit formulas are still not known.

The third, fourth and fifth column of Table 1 are sequences A259114, A259115 and A259116, respectively, from [21].

Table 1
The number of tanglegrams of various types up to 9 leaves.

Leaves	Rooted ord.	Rooted unord.	Unrooted ord.	Unrooted unord.
1	1	1	1	1
2	1	1	1	1
3	2	2	1	1
4	13	10	2	2
5	114	69	4	4
6	1509	807	31	22
7	25595	13048	243	145
8	535753	269221	3532	1875
9	13305590	6660455	62810	31929

7.5. Alternative proof of the main theorem

Ira Gessel [12] pointed out that up to Proposition 4, our main theorem can be proved using Pólya theory. Recently, Eric Fusy [11] gave a combinatorial proof of Proposition 4, which also yields a remarkable simplification of the random sampler for tangled chains.

Acknowledgments

We thank Ira Gessel, Arnold Kas, Jim Pitman, Xavier G. Viennot, Paul Viola, Bianca Viray, and Chris Whidden for helpful discussions.

References

[1] B.L. Allen, M. Steel, Subtree transfer operations and their induced metrics on evolutionary trees, Ann. Comb. 5 (2001) 1–15.
[2] K. Buchin, M. Buchin, J. Byrka, M. Nöllenburg, Y. Okamoto, R.I. Silveira, A. Wolff, Drawing (complete) binary tanglegrams: hardness, approximation, fixed-parameter tractability, Algorithmica 62 (2012) 309–332.
[3] M.A. Charleston, Recent results in cophylogeny mapping, in: T. Littlewood (Ed.), The Evolution of Parasitism-A Phylogenetic Perspective, in: Advances in Parasitology, vol. 54, Academic Press, 2003, pp. 303–330.
[4] F.R.K. Chung, R.L. Graham, V.E. Hoggatt Jr., M. Kleiman, The number of Baxter permutations, J. Combin. Theory Ser. A 24 (1978) 382–394.
[5] R. Cori, S. Dulucq, G. Viennot, Shuffle of parenthesis systems and Baxter permutations, J. Combin. Theory Ser. A 43 (1986) 1–22.
[6] É. Czabarka, L.A. Székely, S. Wagner, Inducibility in binary trees and crossings in random tanglegrams, 2016, arXiv e-prints.
[7] P.W. Diaconis, S.P. Holmes, Matchings and phylogenetic trees, Proc. Natl. Acad. Sci. USA 95 (1998) 14600–14602.
[8] S. Dulucq, O. Guibert, Permutations de Baxter, Sém. Lothar. Combin. 33 (1994) B33c, Tagung des Lotharingischen Kombinatorikseminars (Freiberg, 1994).
[9] S. Dulucq, O. Guibert, Stack words, standard tableaux and Baxter permutations, in: Proceedings of the 6th Conference on Formal Power Series and Algebraic Combinatorics, New Brunswick, NJ, 1994, vol. 157, 1996, pp. 91–106.
[10] G.W. Furnas, The generation of random, binary unordered trees, J. Classification 1 (1984) 187–233.
[11] É. Fusy, On symmetries in phylogenetic trees, Electron. J. Combin. 23 (2016) 25 (electronic).
[12] I. Gessel, The Konvalinka–Amdeberhan conjecture and plethystic inverses, in preparation.
[13] I.M. Gessel, Counting tanglegrams with species, arXiv e-prints, 2015.
[14] M.S. Hafner, S.A. Nadler, Phylogenetic trees support the coevolution of parasites and their hosts, Nature 332 (1988) 258–259.

[15] I.N. Herstein, Topics in Algebra, second ed., Xerox College Publishing, Lexington, Mass.–Toronto, Ont., 1975.

[16] C. Jordan, Sur les assemblages de lignes, J. Reine Angew. Math. (1869) 185–190.

[17] D.E. Knuth, Correction: "An almost linear recurrence", Fibonacci Quart. 4 (1966) 354.

[18] M. Konvalinka, I. Pak, Cayley compositions, partitions, polytopes, and geometric bijections, J. Combin. Theory Ser. A 123 (2014) 86–91.

[19] M. Konvalinka, S. Wagner, The shape of random tanglegrams, Adv. in Appl. Math. 78 (2016) 76–93.

[20] F.A. Matsen IV, S.C. Billey, A. Kas, M. Konvalinka, Tanglegrams: a reduction tool for mathematical phylogenetics, IEEE/ACM Trans. Comput. Biol. Bioinform. (2016), to appear, arXiv preprint available.

[21] OEIS Foundation Inc., The on-line encyclopedia of integer sequences, http://oeis.org, 2015.

[22] R.D. Page, in: Roderic D.M. Page (Ed.), Tangled Trees: Phylogeny, Cospeciation, and Coevolution, The University of Chicago Press, Chicago, 2003.

[23] P. Pevzner, R. Shamir (Eds.), Bioinformatics for Biologists, Cambridge University Press, 2011, Cambridge Books Online.

[24] C. Scornavacca, F. Zickmann, D.H. Huson, Tanglegrams for rooted phylogenetic trees and networks, Bioinformatics 27 (2011) i248–i256.

[25] N.J.A. Sloane, J.A. Sellers, On non-squashing partitions, Discrete Math. 294 (2005) 259–274.

[26] R.P. Stanley, Enumerative Combinatorics, vol. 1, Cambridge Studies in Advanced Mathematics, vol. 49, Cambridge University Press, Cambridge, 1997.

[27] G. Viennot, A bijective proof for the number of Baxter permutations, Sém. Lothar. Combin. (1981).

[28] C. Whidden, N. Zeh, R.G. Beiko, Supertrees based on the subtree prune-and-regraft distance, Syst. Biol. 63 (2014) 566–581.