Klazar Trees and Perfect Matchings

DAVID CALLAN

Department of Statistics
University of Wisconsin-Madison
Medical Science Center
1300 University Ave
Madison, WI 53706-1532

callan@stat.wisc.edu

October 27, 2008

Abstract

Martin Klazar computed the total weight of ordered trees under 12 different notions of weight. The last and perhaps most interesting of these weights, w_{12}, led to a recurrence relation and an identity for which he requested combinatorial explanations. Here we provide such explanations. To do so, we introduce the notion of a "Klazar violator" vertex in an increasing ordered tree and observe that w_{12} counts what we call Klazar trees—increasing ordered trees with no Klazar violators. A highlight of the paper is a bijection from n-edge increasing ordered trees to perfect matchings of $[2n] = \{1, 2, \ldots, 2n\}$ that sends Klazar violators to even numbers matched to a larger odd number. We find the distribution of the latter matches and, in particular, establish the one-summation explicit formula $\sum_{k=1}^{\lfloor n/2 \rfloor}(2k-1)!!^2 \left\{ {n+1 \atop 2k+1} \right\}$ for the number of perfect matchings of $[2n]$ with no even-to-larger-odd matches. The proofs are mostly bijective.

1 Introduction

Martin Klazar [1, 2] defined a *drawing* of an n-edge ordered tree T to be a sequence of trees $(T_1, T_2, \ldots, T_n)$ such that $T_n = T$ and T_i arises from T_{i-1} by deleting a leaf of T_i. He defined the weight w_{12} of T (the last of 12 weights he considered) to be the number of different drawings of T and defined $w_{12}(n) = \sum_{T\in\mathcal{T}_n} w_{12}(T)$ where $\mathcal{T}_n$ is denotes the set of n-edge ordered trees, counted by the Catalan numbers, sequence (A000108) in OEIS [3]. Using generating functions, he found the generating function $\sum_{n\geq 0} w_{12}(n)x^n/n! = \sqrt{e^x/(2-e^x)}$. He also established the identity

$$2^n \sum_{T\in\mathcal{T}_n} w_{12}(T)\left(\tfrac{1}{2}\right)^{\ell(T)} = (2n-1)!! \tag{1}$$

where $\ell(T)$ is the number of leaves in T, and the recurrence relation

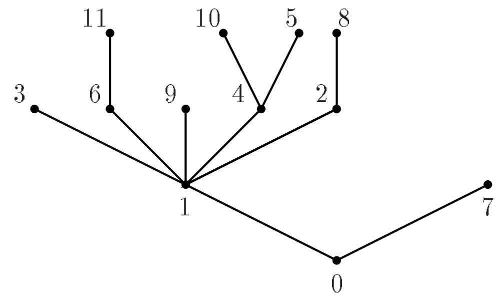
$$w_{12}(n) = w_{12}(n-1) + \sum_{i=1}^{n-1} w_{12}(i)\binom{n-1}{i-1}, \tag{2}$$

and wrote "It would ... be interesting to give direct combinatorial proofs and interpretations" of (1) and (2). Here we will do so.

The outline of the paper is as follows. Section 2 introduces the notion of Klazar violator and an explicit class of trees, Klazar trees, counted by w_{12}. Section 3 gives a combinatorial proof of identity (1) and Section 4 illuminates it. Section 5 finds the generating function for Klazar violators. Section 6 presents an easily described class of perfect matchings counted by w_{12}, which serves in Section 7 to give a combinatorial interpretation of recurrence (2). Section 8 discusses codes for trees and matchings that are useful as an intermediate construct for the bijection in the next section. Section 9 describes a bijection, both recursively and explicitly, between increasing ordered trees and perfect matchings represented as dot diagrams. This bijection, interesting in its own right, translates the result of Section 7 to provide a direct combinatorial proof of recurrence (2) in context. Section 10 presents some consequences of this bijection, including a bivariate generating function for perfect matchings on $[2n]$ counting instances of an even number matched to a larger odd number and some results for the "trapezoidal words" considered by Riordan.

2 The Weight w_{12} Counts Klazar Trees

Let $\mathcal{I}_n$ denote the set of n-edge increasing ordered trees, that is, n-edge ordered trees with vertices labeled $0, 1, \ldots, n$ so that the label of each child vertex exceeds that of its parent. It is well known, and indeed a nice proof is included in Klazar's paper [1], that $|\,\mathcal{I}_n\,| = (2n-1)!!$, the odd double factorial. Now $w_{12}(n)$ counts trees in $\mathcal{I}_n$ satisfying a technical condition whose description is facilitated by introducing some terminology for an increasing ordered tree. A *descent* is a pair of adjacent sibling vertices in which the first exceeds the second. The first is a *descent initiator*, the second a *descent terminator*. The (left) *cohort* $C(v)$ of a vertex v is the list of all siblings of v lying strictly to its left. The *big cohort* $B(v)$ of v is the maximal terminal sublist of the left cohort of v all of whose entries exceed v. Thus the big cohort of v is nonempty iff v is a descent terminator. The *associate* $A(v)$ of a descent terminator v is the smallest entry in the big cohort of v. Figure 1 illustrates these notions.



An increasing ordered tree

The cohort of 1 is empty, the cohort of 4 is $C(4) = (3, 6, 9)$;

the big cohort of 4 is $B(4) = (6, 9)$, the big cohort of 2 is $(3, 6, 9, 4)$;

the descent terminators are 4,2,5 and their associates are $A(4) = 6$, $A(2) = 3$, $A(5) = 10$.

Figure 1

Definition A *Klazar violator* (KV for short) in an increasing ordered tree is a descent terminator whose left associate is smaller than every child of v.

In particular, a descent terminator with no children is a Klazar violator because it satisfies this condition vacuously. For example, in the tree in Figure 1 above, the descent terminators are 4,2,5, and the Klazar violators are 2 and 5.

Remark If we follow the standard convention that the minimum of an empty set is ∞ and extend the notion of associate to all vertices, then we can say that a vertex v is a Klazar complier (KC), that is, not a Klazar violator, iff the associate of v is $\geq$ the minimum of the children of v. (The weak inequality $\geq$ is used to allow for $\infty \geq \infty$.)

Definition A *Klazar tree* is an increasing ordered tree with no Klazar violators.

An increasing (i.e. child > parent) labeling of an ordered tree T determines a drawing of T—delete the vertices in decreasing order—but the resulting drawings are not all distinct. The problem is that in labeling the vertices as the tree is built up from a drawing to produce an increasing ordered tree, adding an edge among a cluster of sibling leaf edges with a common parent gives the same tree regardless of where in the cluster the new edge is placed. It is straightforward to verify, however, that if in this situation, the new edge is always placed so that it is the rightmost edge of the cluster the resulting labeled tree will be a Klazar tree and otherwise at least one Klazar violator will be present. Thus $w_{12}(T)$ is the number of Klazar trees whose underlying ordered tree is T and, letting $\mathcal{K}_n$ denote the set of all Klazar trees with n edges, $w_{12}(n) = |\mathcal{K}_n|$.

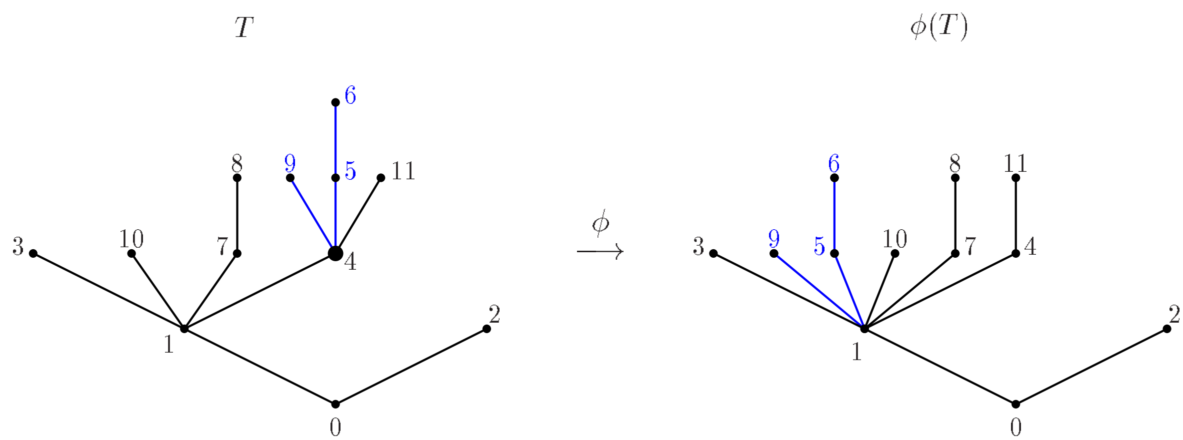
3 A Combinatorial Proof of Klazar's Identity

It is convenient, following Deutsch [4], to define a *node* in a rooted tree to be a vertex that is neither the root nor a leaf. Now Klazar's identity (1) can be written as

$$\sum_{K \in \mathcal{K}_n} 2^{\nu(K)} = (2n-1)!! \tag{3}$$

where $\nu(K)$ is the number of nodes in K. Since $2^{\nu(K)}$ is the number of subsets of the nodes of K, let us define a *node-marked* Klazar tree to be one in which some (all, or none) of its nodes are marked and let $\mathcal{NK}_n$ denote the set of node-marked Klazar trees on n edges. Thus (3) asserts that $|\mathcal{NK}_n| = (2n-1)!!$. To prove this assertion we exhibit a simple bijection ϕ from $\mathcal{NK}_n$ to $\mathcal{I}_n$, the set of n-edge increasing ordered trees.

Given a node-marked Klazar tree T, turn each marked node u into a Klazar violator by the following cut-and-paste procedure. Let v be the smallest child of u. Take v and its cohort and transfer all these vertices along with their subtrees and parent edges so that they become siblings of u situated immediately to the left of the big cohort of u, and then remove the mark from u as illustrated. (The $*$ superscript refers to the image tree. Thus $A^*(w), B^*(w), C^*(w)$ refer respectively to the associate, big cohort, and cohort of w in $\phi(T)$.)

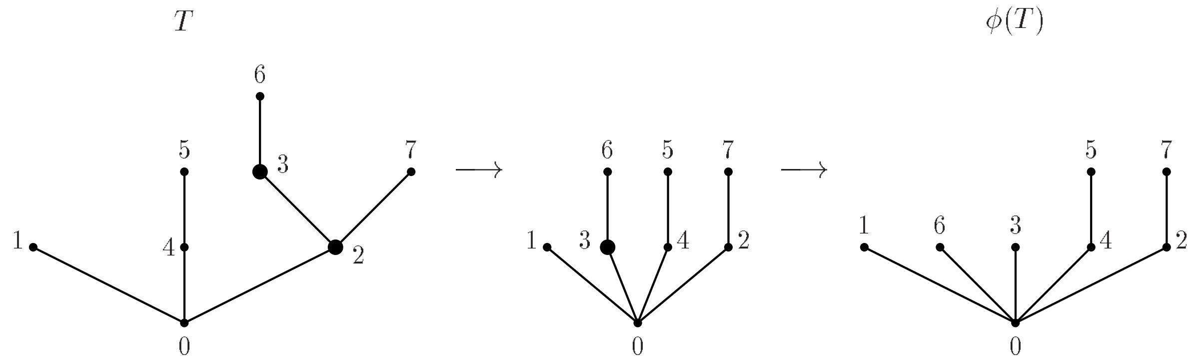


$u = 4$ is a marked node in T,
$v = 5$ is smallest child of u,
(9) is the cohort of v.

$u = 4$ is a Klazar violator in $\phi(T)$,
$B^*(u) = (\underbrace{9}_{C(v)}, \underbrace{5}_{v}, \underbrace{10, 7}_{B(u)})$,
$B^*(v) = C(v) = (9)$.

In $\phi(T)$, the marked node u is recaptured as a Klazar violator, the vertex v is recaptured as $A^*(u)$ because v is the smallest entry in $B^*(u)$, and $C(v)$ is recaptured as $B^*(v)$.

In this example there is just one marked node; if there is more than one, the marked nodes are processed one after the other (the order of processing is immaterial) as in the following example.



As each marked node is processed, it is turned into a Klazar violator without affecting the complier/violator status of any other vertex. Thus the originally marked nodes can be recovered as the Klazar violators in $\phi(T)$, and the entire process is reversible.

4 An Interpretation of Klazar's Identity

From the preceding section we have a combinatorial proof of (1) but not yet a satisfactory combinatorial interpretation: what is still missing is a nice characterization of the image in $\mathcal{I}_n$, under ϕ, of the node-marked Klazar trees with ℓ leaves. Such a characterization is far from obvious, so how to find one? We will tackle this problem (with gratifying success) but first let us review the genesis of (1). Klazar, looking for the generating function for $w_{12}(n)$, found a recurrence for a more refined count by number of leaves. This led him to to an expression for the bivariate generating function

$$F^*(x,y) := \sum_{n\geq 0} \sum_{T\in\mathcal{T}_n} w_{12}(T)\frac{x^n}{n!}y^{\ell(T)}, \tag{4}$$

where $\ell(T)$ is the number of leaves of T, to wit,

$$F^*(x,y) = \sqrt{\frac{2y-1}{2ye^{x(1-2y)}-1}}. \tag{5}$$

Of course, $F^*(x,1) = \sqrt{e^x/(2-e^x)}$ is the desired exponential generating function (egf) for $w_{12}(n)$. But Klazar also noted that the tweaked function $F^{**}(x,y) := F^*(2x, y/2)$ has the curious property that $F^{**}(x,1) = 1/\sqrt{1-2x}$, the exponential generating function for the odd double factorials. Since (4) says

$$F^{**}(x,y) = \sum_{n\geq 0} 2^n \sum_{T\in\mathcal{T}_n} w_{12}(T)\left(\frac{1}{2}\right)^{\ell(T)} \frac{x^n}{n!} y^{\ell(T)}, \tag{6}$$

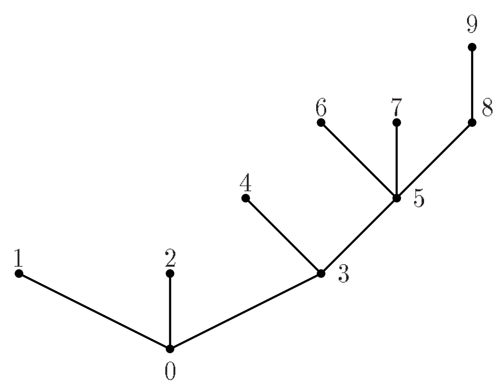
Klazar obtained (1) by setting $y = 1$ in (6) and equating coefficients of $\frac{x^n}{n!}$.

Now let $a(n,\ell)$ denote the coefficient of $\frac{x^n}{n!}y^\ell$ in $F^{**}(x,y) = \sqrt{\frac{y-1}{ye^{2x(1-y)}-1}} = 1 + \sum_{n,\ell\geq 1} a(n,\ell)\frac{x^n}{n!}y^\ell$. The first few values of $a(n,\ell)$ are given in the following table.

$n \setminus \ell$	1	2	3	4	5	6	7
1	1						
2	2	1					
3	4	10	1				
4	8	60	36	1			
5	16	296	516	116	1		
6	32	128	5158	3508	358	1	
7	64	5664	42960	64240	21120	1086	1

Table of values of $a_{n,\ell}$

The first order of business is to try to find a statistic on $\mathcal{I}_n$ whose distribution is given by the array $\big(a(n,\ell)\big)$. The first column $(a(n,1))_{n\geq 1}$ appears to be $(2^{n-1})_{n\geq 1}$ and 2^{n-1} is the number of compositions of n. A composition $n = n_1 + n_2 + \ldots + n_r$ $(n_i \geq 1)$ suggests an increasing ordered tree in a simple way: split $[n]$ into blocks of consecutive integers of lengths $n_1, n_2, \ldots, n_r$, and use the blocks as sibling lists, each having as common parent the last entry of the previous list (or the root, in the case of the first list) as illustrated in Figure 2.



The increasing ordered tree corresponding to the composition $(3,2,3,1)$ of 9

Figure 2

These trees, counted by 2^{n-1}, are clearly increasing, and they are characterized by the further properties:

- no sibling descents
- only the rightmost child of a vertex can have children.

This motivates us to define a *bad* vertex in an increasing ordered tree to be a vertex that (i) initiates a sibling descent or (ii) initiates a sibling ascent and has children. Thus the n-edge increasing ordered trees counted by $a(n, 1) = 2^{n-1}$ are those with no bad vertices. Could it be that $a(n, \ell)$ is the number with $\ell - 1$ bad vertices? Computer calculations suggest that indeed it is, and so we are (strongly) motivated to check if ϕ sends node-marked Klazar trees with ℓ leaves to increasing ordered trees with $\ell - 1$ bad vertices. It doesn't, but it does send them to increasing ordered trees with $\ell - 1$ *reverse-bad* vertices (a vertex is reverse-bad if it is bad viewing the tree from right to left, that is, if it is bad in the tree obtained by flipping the original tree over a vertical line). This follows from the following two key observations.

Proposition 1. *(i) In a Klazar tree, the number of reverse-bad vertices is one less than the number of leaves, and (ii) the bijection $\phi : \mathcal{NK}_n \to \mathcal{I}_n$ presented above preserves the number of reverse-bad vertices.*

Proof (i) Let T be a Klazar tree. Given a leaf u in T, consider the path from u to the root. Let $\pi(u)$ be the first vertex on this path (possibly u itself) that has a left sibling. The map π is defined for all leaves except the that terminates the leftmost path from the root. We claim it is a bijection to the reverse-bad vertices of T and the result follows. To see the claim, observe that if $\pi(u) = u$ then u has closest left sibling v and $v < u$ for otherwise u, being a leaf, would be a Klazar violator. Hence u is reverse-bad. If $\pi(u) \neq u$ then $\pi(u)$ has both a left sibling and a child and so is certainly reverse-bad. Thus π sends all but the exceptional leaf to reverse-bad vertices. Conversely, given a reverse-bad vertex v, map it to the leaf terminating the leftmost path from v away from the root. This map is the inverse of π.

(ii) It suffices to verify the assertion for a single application of the "mark to violator" process and this involves a routine check of various cases, which we leave to the reader. □

Of course, the distribution of reverse-bad vertices is the same as the distribution of bad vertices. The preceding discussion shows how I both stumbled upon and proved the following combinatorial interpretation for the array $\big(a(n, \ell)\big)$ implicit in (1).

Theorem 2. *$a(n, \ell)$ is the number of n-edge increasing ordered trees with ℓ bad vertices where a vertex is bad iff it either initiates a sibling descent or initiates a sibling ascent and has children.* □

5 The generating function for Klazar violators

The following result sheds further light on (1).

Theorem 3. *Let $F(x,y,z)$ denote the trivariate generating function $\sum_{n,i,j\geq 0} a_{n,i,j}\frac{x^n}{n!}y^i z^j$ where $a_{n,i,j}$ is the number of increasing ordered trees on $[n]$ with i Klazar violators and j leaves that do not terminate a sibling descent. Then*

$$F(x,y,z) = \left(\frac{1+y-2z}{1+y-2ze^{x(1+y-2z)}}\right)^{\frac{1}{2}}.$$

We defer the proof to list some simple corollaries.

Corollary 4. *The generating function for increasing ordered trees by number of Klazar violators is*

$$\left(\frac{1-y}{2e^{x(y-1)}-1-y}\right)^{\frac{1}{2}}.$$

Proof. Put $z=1$ in $F(x,y,z)$.

Corollary 5. *The statistics "# non-descent-terminator leaves" and "# reverse-bad vertices" are equidistributed on increasing ordered trees.*

Proof. The generating function for the first of these statistics is $F(x,1,z) = \sqrt{\frac{z-1}{ze^{2x(1-z)}-1}}$ and this agrees with F^{**} above.

We can also recover Klazar's bivariate generating function (5) by setting $y=0$ in $F(x,y,z)$: *every* leaf in a Klazar tree is a non-descent-terminator leaf since a descent-terminator leaf would be a Klazar violator.

Proof of Theorem 3 Consider the effect of adding a leaf n to an increasing ordered tree of size $n-1$. On the one hand, the number of Klazar violators increases by 1 if the new leaf is the immediate left sibling of what was originally a non-descent-terminator leaf; otherwise it stays the same. On the other hand, the number of non-descent-terminator leaves stays the same if the new leaf is either the rightmost child or the immediate left sibling of what was originally a non-descent-terminator leaf; otherwise it increases by 1. These observations lead to the recurrence relation

$$a_{n,i,j} = ja_{n-1,i,j} + ja_{n-1,i-1,j} + (2n-2j+1)a_{n-1,i,j-1}$$

for $n \geq 1$, $i \geq 0$, $j \geq 1$ and (n,i,j) neither $(1,0,2)$ nor $(1,1,1)$, with initial conditions $a_{0,0,1}=1$, $a_{1,0,2}=0$, $a_{1,1,1}=0$, $a_{0,i,j}=0$ for $(i,j)\neq(0,1)$ and $a_{n,i,j}=0$ if $i<0$ or $j<1$.

This recurrence translates to the first-order partial differential equation

$$(2xz-1)F_x + (z+yz-2z^2)F_z + zF = 0$$

with solution (à la [1, p. 207]) as asserted in the Theorem. □

By similar considerations it is also possible to obtain a recurrence relation for $a_{n,i,j,k}$, the number of increasing ordered trees on $[n]$ with i Klazar violators, j leaves that do not terminate a sibling descent, and k leaves altogether:

$a_{0,0,1,1} = a_{1,0,1,1} = a_{2,0,2,2} = a_{2,0,1,1} = a_{2,1,1,2} = 1$ and for other $n \leq 2$, $a_{n,i,j,k} = 0$,

and for $n \geq 3$, $a_{n,i,j,k} =$

$$ja_{n-1,i,j,k} + ja_{n-1,i-1,j,k-1} + (k-(j-1))a_{n-1,i,j-1,k} + (2n-k-j+1)a_{n-1,i,j-1,k-1}.$$

The presence of the statistic "total number of leaves", however, precludes finding an "elementary" generating function because the known (marginal) distribution of this statistic is not elementary; see comment on sequence A008517 in [3].

6 A Class of Perfect Matchings

A perfect matching (always on the support set $[2n] = \{1,2,\ldots,2n\}$) is a partition of $[2n]$ into 2-element subsets or *matches*. The *size* of the matching is n and we write all matches with the smaller entry first so that, for example, an even-to-odd match is an instance of an even number matched to a larger odd number. Thus the perfect matching $1\,5\,/\,2\,7\,/\,3\,4\,/\,6\,8$ has one even-to-odd match, namely $2\,7$. We will show that the number $a(n)$ of perfect matchings of size n with no even-to-odd matches has the same generating function as $w_{12}(n)$—$\sum_{n\geq 0} a(n)x^n/n! = \sqrt{e^x/(2-x)}$—by finding an explicit formula for $a(n)$.

Proposition 6. *The number of perfect matchings on $\{1,2,\ldots,2n\}$ with no even-to-odd matches and k even-to-even matches is $(2k-1)!!^2\left\{{n+1 \atop 2k+1}\right\}$.*

Here $\left\{{n \atop k}\right\}$ is the Stirling partition number: the number of partitions of $[n] := \{1,2,\ldots,n\}$ into k nonempty disjoint sets (blocks). We defer the proof of Prop. 6 to deduce the generating function. By convention, $(-1)!! = 1$.

Proposition 7.

$$\sum_{n\ge 0}\left(\sum_{k\ge 0}(2k-1)!!^2\left\{ {n+1 \atop 2k+1} \right\}\right)\frac{x^n}{n!}=\sqrt{\frac{e^x}{2-e^x}}.$$

Proof Reversing the order of summation, the left hand side is

$$\begin{aligned}
&D_x\left(\sum_{k\ge 0}(2k-1)!!^2\sum_{n\ge 0}\left\{ {n+1 \atop 2k+1} \right\}\frac{x^{n+1}}{(n+1)!}\right)\\
&=D_x\left(\sum_{k\ge 0}(2k-1)!!^2\sum_{n\ge 1}\left\{ {n \atop 2k+1} \right\}\frac{x^{n}}{n!}\right)\\
&=D_x\left(\sum_{k\ge 0}(2k-1)!!^2\frac{(e^x-1)^{2k+1}}{(2k+1)!}\right)\\
&=e^x\sum_{k\ge 0}(2k-1)!!^2\frac{(e^x-1)^{2k}}{(2k)!}\\
&=e^x\sum_{k\ge 0}\binom{2k}{k}\left(\frac{e^x-1}{2}\right)^{2k}\\
&=e^x\left(1-(e^x-1)^2\right)^{-1/2}\\
&=\sqrt{\frac{e^x}{2-e^x}},
\end{aligned}$$

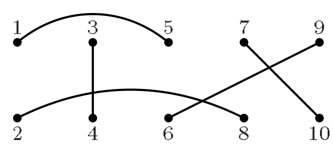
where a standard generating function for Stirling numbers [5, Eq. (7.49), p. 351] is used at the second equality. □

Remark A similar calculation gives a bivariate generating function. If there are k even-to-even matches, then there are also k odd-to-odd matches and so $n-2k$ opposite-parity matches. If $a(n,j)$ denotes the number of no-even-to-odd matchings with j odd-to-even matches, then the mixed generating function for $a(n,j)$ is given by

$$\sum_{n,j\ge 0}a(n,j)\frac{x^n}{n!}y^j=\frac{ye^{xy}}{\sqrt{y^2-1+e^{xy}(2-e^{xy})}}$$

□

It will be convenient to represent a perfect matching as a dot diagram with vertices arranged in two rows as illustrated.



a perfect-matching (PM) dot diagram

It is also convenient to distinguish an *arc* joining two dots in the same row (same parity matches) and a *line* joining dots in different rows (opposite parity matches). An even-to-odd match shows up in the dot diagram as an *upline* (line of positive slope) and an odd-to-even match as a *weak downline* (a vertical line or line of negative slope). Thus 6-9 is an upline, 1-5 is an arc, and 3-4 is a weak downline. The labels are not necessary in a PM dot diagram and we will often use i bot to refer to the ith dot in the bottom row and analogously for i top.

To establish Prop. 6 bijectively, we will actually give a more refined count of the no-even-to-odd matchings and then invoke the identity

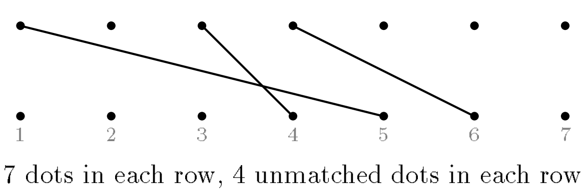
$$\left\{ {n+1 \atop 2k+1} \right\} = \sum_{j\ge 0} \left\{ {j \atop 2k} \right\} (2k+1)^{n-j},$$

which is easily proved bijectively [6, p. 106, Identity 201]: the right hand side counts the partitions of $[n+1]$ into $2k+1$ blocks by smallest entry of the last block where, as throughout this note, the blocks of a partition are arranged in a standard order so that the smallest entries are increasing left to right.

Theorem 8. *The number of perfect matchings on the set* $[2n]$ *in which no even number is matched to a larger odd number, with* k *even-to-even matches and* $2j$ *the largest number occurring among the even-to-even matches is* $(2k-1)!!^2 \left\{ {j \atop 2k} \right\} (2k+1)^{n-j}$.

Before proceeding with the proof we establish simple combinatorial interpretations of the Stirling partition numbers and the perfect powers in terms of matchings in dot diagrams.

Definition An (n, k) *Stirling matching* is a $2 \times n$ array of dots with $n-k$ disjoint edges, each connecting a dot in the top row to a dot lying strictly to its right in the bottom row.



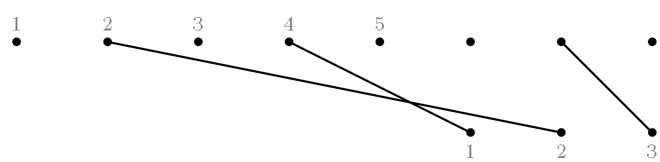
A (7,4) Stirling matching

Proposition 9. *The number of* (n, k) *Stirling matchings is the Stirling partition number* $\left\{ {n \atop k} \right\}$.

Proof Let $\mathcal{S}(n,k)$ denote the set of (n,k) Stirling matchings and set $S(n,k) = |\,\mathcal{S}(n,k)\,|$. A matching in $\mathcal{S}(n,k)$ can be obtained either (i) from a matching in $\mathcal{S}(n-1,k-1)$ by adding an unmatched dot at the end of each row—$S(n-1,k-1)$ choices—or (ii) from one in $\mathcal{S}(n-1,k)$ by similarly adding two dots and connecting the bottom one to any of the k unmatched dots in the original top row—$kS(n-1,k)$ choices. Every matching in $\mathcal{S}(n,k)$ arises uniquely in one of these two ways, and so $S(n,k)$ satisfies the basic recurrence for the Stirling partition numbers: $S(n,k) = S(n-1,k-1) + kS(n-1,k)$. Since the boundary conditions $S(1,1) = \left\{ {1 \atop 1} \right\} = 1$ and $S(n,k) = \left\{ {n \atop k} \right\} = 0$ for $k > n$ or $k < 1$ also hold, we conclude that $S(n,k) = \left\{ {n \atop k} \right\}$. $\square$

Remark It is possible to derive from this recurrence a bijection from $\mathcal{S}(n,k)$ to the partitions of $[n]$ into k blocks (arranged in standard order: smallest entries increasing left to right). First label the dots in each row $1, 2, \ldots, n$ left to right. Then for each $i \in [n]$, place i in block j $(1 \le j \le k)$ as follows. If dot i in the bottom row is unmatched, then $j = 1+$ # unmatched dots in the bottom row with label $< i$. If dot i in the bottom row is matched, say to dot k in the top row, then $j = k-$ # edges that connect a dot $< k$ in the top row to a dot $< i$ in the bottom row. For example, the Stirling partition illustrated above corresponds to the partition 1 5/2 6/3 4/7. $\square$

Definition A (k,n) perfect-power matching is a 2-row array consisting of $k+n$ dots in the top row and n dots flush right in the bottom row and a matching of all the lower dots to upper dots such that the bottom dot of each edge lies strictly to the right of its top dot.



A (5,3) perfect-power matching

Proposition 10. *The number of (k,n) perfect-power matchings is k^n.*

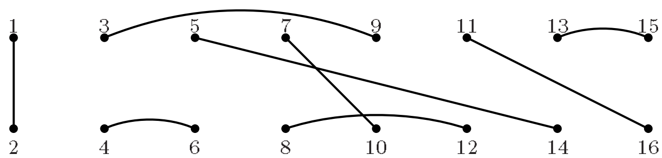
Proof Every (k,n) matching comes from a $(k,n-1)$ matching by appending a dot to each row and connecting the lower dot to one of the k unmatched dots in the original top row. This gives a multiplying factor of k each time n is incremented, and the result follows. $\square$

Proof of Theorem 8 Let us take, as a working example, the matching

1 2 / 3 15 / 4 8 / 5 14 / 6 12 / 7 10 / 9 13 / 11 16

with $n = 8$, $k = 2$ and $j = 6$.

First, we take care of the $(2k-1)!!^2$ factor (this step is easy). Pick out the k pairs consisting of two even integers, here $4\,8$ and $6\,12$ and let A denote their support, here $\{4, 6, 8, 12\}$. These pairs form a perfect matching on A—$(2k-1)!!$ possibilities—and so we may extract a $(2k-1)!!$ factor and assume the pairs in question form a standard matching on A: smallest entry of A matched to next smallest, third smallest to fourth smallest and so on. Likewise for the odd-to–odd pairs (also necessarily k in number) we may extract another $(2k-1)!!$ factor and assume a standard matching on their support, B. Standardizing the matchings on A and B for our working example, we get a PM dot diagram:



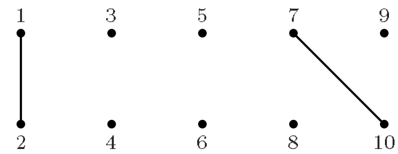
The defining characteristics of the PM dot diagrams in question are then

- same number of dots in each row
- all dots are matched
- no uplines
- the *arcs* in each row are standard:  . . . (no crossings or nestings)

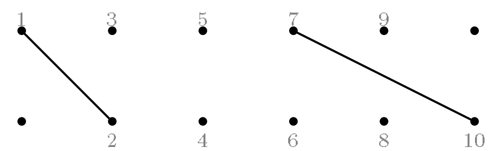
The parameters n, k, j appear respectively as number of dots in each row, number of arcs in each row, and the position in the bottom row of its last dot incident with an arc. As noted above, the labels are not necessary and serve only for identification.

Now we will give a bijection from these dot diagrams to the Cartesian product of $\mathcal{S}(j, 2k)$, the Stirling matchings defined above, and $\mathcal{P}(2k+1, n-j)$, the perfect-power matchings defined above. Since, by Props. 9 and 10, $\mathcal{S}(j, 2k)$ and $\mathcal{P}(2k+1, n-j)$ are counted by $\left\{ {j \atop 2k} \right\}$ and $(2k+1)^{n-j}$ respectively, the Theorem will follow.

To get the Stirling matching, take the first $j-1$ dots in each row and the *lines* connecting them.

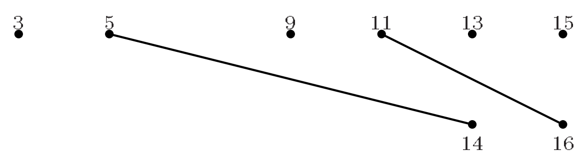


Notice that there may be vertical lines but now eliminate them by making the technical adjustment of shifting the bottom row one unit to the right and adding a dot to each row:

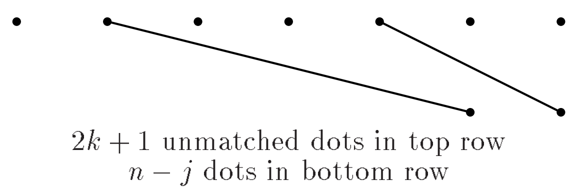


This is the $(j, 2k)$ Stirling matching.

To get the perfect-power matching, delete the first j dots in the bottom row and the dots in the top row connected to them, and then delete all arcs in the top row while leaving their endpoints intact:



Make the same technical adjustment of shifting the bottom row and "prettify" the diagram:



This is the $(2k+1, n-j)$ perfect-power matching.

It is easy to check that these maps define a bijection from no-upline PM dot diagrams to $\mathcal{S}(j, 2k) \times \mathcal{P}(2k+1, n-j)$ and, as noted, the Theorem follows. $\square$

7 An Interpretation of Klazar's Recurrence

In the preceding section we showed that the number $a(n)$ of no-upline PM dot diagrams of size n has the same generating function as $w_{12}(n)$. In this section we show directly that $a(n)$ satisfies recurrence (2). Write (2) in the equivalent form

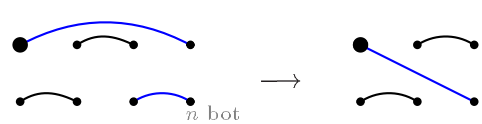
$$a(n) = \underbrace{a(n-1)}_{(1)} + \underbrace{(n-1)a(n-1)}_{(2)} + \underbrace{\sum_{k=0}^{n-3} \binom{n-1}{k+2} a(n-2-k)}_{(3)} \tag{7}$$

and split the no-upline PM dot diagrams of size n into 3 classes according to the partners (matched entries) of the last dots in each row, n top and n bot, as follows. Recall that the partner of n top, denoted $p(n$ top), is necessarily in the top row unless n top is matched to n bot. Class (1) consists of the dot diagrams in which n top is matched to n bot, that is, the last dots in each row are joined by a vertical line. Class (2) consists of the dot diagrams in which $p(n$ bot) is either (i) in the top row or (ii) in the bottom row and $p(n$ top) $<$ $p(n$ bot). Class (3) consists of the dot diagrams in which $p(n$ top) $>$ $p(n$ bot) and $p(n$ bot) is in the bottom row. (See the illustrations below.) Now let us count the dot diagrams in each class.

For a dot diagram in Class (1), delete the last dot in each row. This gives a dot diagram of size $n-1$—$a(n-1)$ possibilities—and the original dot diagram can of course be uniquely recovered from it.

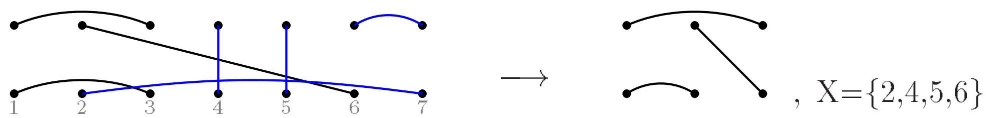
For a dot diagram in Class (2), highlight the partner of n top with a heavy dot, then delete the last dot in each row (and the arcs/lines therefrom) and join up their partners. The result is a dot diagram of size $n-1$ with one highlighted dot in the top row—$(n-1)a(n-1)$ possibilities—which uniquely determines the original.

partner of n bot is in the top row



partner of n bot is in the bottom row

For a dot diagram in Class (3), first record the locations of the partners of the last dots and of all $k \geq 0$ *vertical* lines lying *between* these partners. This gives a $(k+2)$-element subset X of $[n-1]$. Then delete the last dots, their partners, all these vertical lines, and "prettify" the diagram. The result is a dot diagram of size $n-k-2$ which, together with the set X, determines the original.



Notice that in this case the vertical lines have to be deleted, for otherwise the resulting dot diagram would contain lines of positive slope, which are forbidden.

Thus the three classes are counted by the three terms in (7). $\square$

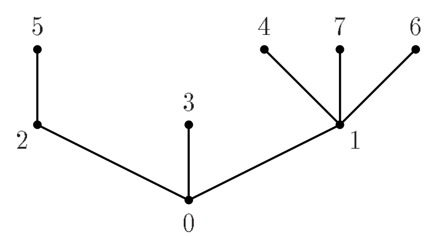
8 Codes for Trees and Matchings

An increasing ordered tree of size n can be built up from the root 0 by successively adding vertices $1, 2, \ldots, n$. Vertex 1 is necessarily a (rightmost) child of the root and for $2 \le i \le n$, vertex i is either the rightmost child of a vertex $v \in [0, i-1]$—coded as (R, v)—or the (immediate) left neighbor of a vertex $v \in [1, i-1]$—coded as (L, v). Thus an increasing ordered tree of size n corresponds naturally to a *build-tree code* $\big((X_k, i_k)\big)_{1\le k\le n}$ where $(X_1, i_1) = (R, 0)$ and for $2 \le k \le n$, $X_k = R$ and $i_k \in [0, k-1]$ or $X_k = L$ and $i_k \in [1, k-1]$. For example, with X_v short for (X, v),

1	2	3	4	5	6	7
R_0	L_1	L_1	R_1	R_2	R_1	L_6

build-tree code

$\longrightarrow$



increasing ordered tree

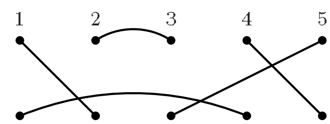
natural correspondence between build-tree codes and increasing ordered trees

To reverse this correspondence, if n is a rightmost child, record $(X_n, i_n) = (R, i)$ with i the parent of n, otherwise $(X_n, i_n) = (L, i)$ with i the right neighbor of n. Delete n and proceed similarly, starting with $n-1$, to obtain (X_k, i_k) for $k = n, n-1, \ldots, 1$ in turn.

Similarly, a PM dot diagram can be built up from the empty diagram by successively adding a rightmost dot to each row, connecting the new top dot to one of the dots i in the bottom row—coded as (B, i)—or to one of the old dots i in the top row—coded as (T, i). Thus a diagram of size n corresponds naturally to a *build-matching code* $\big((Y_k, i_k)\big)_{1 \le k \le n}$ where $(Y_1, i_1) = (B, 1)$ and for $2 \le k \le n$, $Y_k = B$ and $i_k \in [1, k]$ or $Y_k = T$ and $i_k \in [1, k-1]$.

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ B_1 & B_1 & T_2 & B_3 & B_3 \end{array} \quad \longrightarrow$$

build-matching code



PM dot diagram

natural correspondence between build-matching codes and PM dot diagrams

Since there are $2k - 1$ possibilities for (X_k, i_k) in a build-tree code and for (Y_k, i_k) in a build-matching code, both are counted by $(2n-1)!!$, and we see once again that there are $(2n-1)!!$ increasing ordered trees of size n.

We need the correspondence between the two codes obtained by identifying $R \leftrightarrow B$ and $L \leftrightarrow T$ and, when (X_k, i_k) in a build-tree code has $X_k = R$ and $i_k = 0$, replacing it by $(Y_k, i_k) = (B, k)$.

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ R_0 & R_1 & L_2 & R_0 & L_3 \end{array} \quad \longrightarrow \quad \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ B_1 & B_1 & T_2 & B_4 & T_3 \end{array}$$

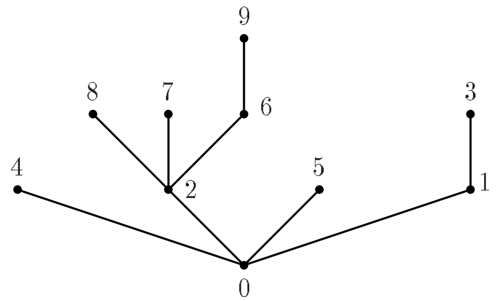
build-tree code   build-matching code

correspondence between build-tree codes and build-matching codes

9 A Combinatorial Proof of Klazar's Recurrence

9.1 Preliminaries

To complete the combinatorial proof of recurrence (2) in context we need a bijection from Klazar trees to perfect matchings with no even-to-odd matches, equivalently to PM dot diagrams with no uplines. In fact, we will give a bijection Φ from increasing ordered trees to PM dot diagrams that sends Klazar violators to uplines. More precisely, define the *partner* of a Klazar violator to be its rightmost child or closest left sibling, whichever is larger if both are present (for a Klazar violator at least one must be present). For example, the tree below has 4 Klazar violators with partners as shown.



Klazar violator	partner
1	5
2	6
6	9
7	8

Then the bijection Φ sends each Klazar violator i together with its partner j to an upline from the ith dot in the bottom row to the jth dot in the top row.

Proposition 11. *In an increasing ordered tree, the map Klazar violator $\mapsto$ partner is one-to-one.*

Proof. A vertex in an increasing ordered tree is the partner of at most one Klazar violator because a partner is either a rightmost child or a left neighbor and a vertex cannot simultaneously be both a rightmost child and a left neighbor. □

A *child* vertex in an increasing ordered tree is a vertex that is a child of some other vertex, that is, a non-root vertex. Partition the child vertices in an increasing ordered tree into two classes: those that are the partner of some Klazar violator ("partners") and those that are not ("non-partners").

Proposition 12. *In an increasing ordered tree, there is a bijection H from Klazar compliant child vertices to non-partners.*

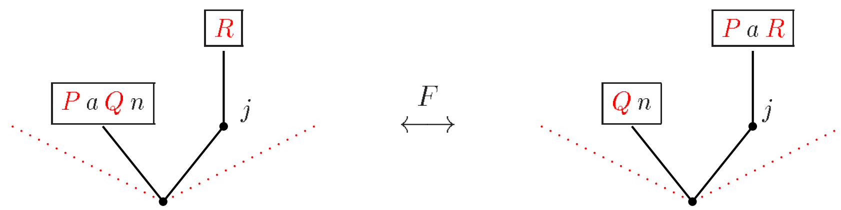
Proof. Suppose v is a Klazar compliant child vertex. If v is a non-partner, set $H(v) = v$. Otherwise, $v_1 := v$ is the partner of a (unique) Klazar violator v_2 and clearly $v_2 < v_1$. If v_2 is a non-partner, set $H(v) = v_2$. Otherwise proceed similarly to obtain vertices $v_2 > v_3 > \ldots > v_k$ stopping at the first v_k that is a non-partner and set $H(v) = v_k$. For example, the tree above has 5 Klazar compliant child vertices and 5 non-partners as in the table:

Klazar compliant child v	3	4	5	8	9
non-partner $H(v)$	3	4	1	7	2

□

Proposition 13. *There is an involution F on increasing ordered trees of size n that is the identity on trees in which n does not have a right neighbor and otherwise flips the Klazar violator/complier status of the right neighbor of n while not disturbing any other Klazar violators or their partners.*

Proof. In case the right neighbor j of n is a Klazar violator transfer the associate of j and its big cohort so that they become the leftmost segment of the children of j; in case the right neighbor j of n is a Klazar complier transfer the smallest child of j and its cohort to the immediate left of the big cohort of j, as illustrated.



n is left neighbor of Klazar violator j,
$a \neq n$ is the associate of j,
$PaQn$ is the big cohort of j,
R denotes the children of j

n is left neighbor of
Klazar compliant vertex j,
Qn is the big cohort of j,
a is the smallest child of j,

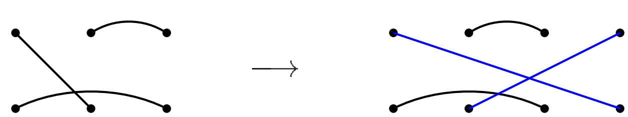
items in red may not be present

The involution F

$\square$

To prune an increasing ordered tree T of size n means to delete n (necessarily a leaf) and its incident edge. We use $P(T)$ to denote the pruned tree. Also, for a Klazar violator v, $p(v)$ denotes its partner.

A PM dot diagram of size $n-1$ can be enlarged to one of size n in $2n-1$ ways: add a dot at the end of each row and then either join these two dots together leaving the rest of the original perfect matching intact, or join the new dot in the top row to any one of the $2n-2$ original dots i, delete the edge from i to its original partner j and replace it with an edge from j to the new dot in the bottom row.



enlarge using "2 bot" (second dot in bottom row)

9.2 Recursive Φ

Now we can give a recursive definition of Φ. First, Φ sends the unique increasing ordered tree of one edge to the unique PM dot diagram of size 1. For an increasing ordered tree T of size $n \geq 2$, define $\Phi(T)$ according to the 6 cases in the following table. Recall that P (for prune), F (an involution), p (for partner), and the bijection H have all been defined in the preceding subsection and KV, KC mean Klazar violator, complier respectively.

#	Case	Status of j	Enlarge:	using:
1	n rightmost child of root	-	$\Phi(P(T))$	n bot
2	n rightmost child of KV j	j is KV in $P(T)$	$\Phi(P(T))$	j bot
3	n is left neighbor of KV j and n is associate of j	j is KC leaf in $P(T)$	$\Phi(P(T))$	j bot
4	n rightmost child of KC j	j is KC and $H(j)$ is non-partner in $P(T)$	$\Phi(P(T))$	$H(j)$ top
5	n is left neighbor of KV j and n is not associate of j	j is KC non-leaf in $P(F(T))$	$\Phi(P(F(T))$	j bot
6	n is left neighbor of KC j	j is KV in $P(F(T))$	$\Phi(P(F(T))$	$p(j)$ top

The bijection Φ

Theorem 14. Φ *is a size-preserving bijection from increasing ordered trees to perfect matchings that sends (Klazar violator, partner) pairs* (i,j) *to uplines* $i \nearrow j$*.*

Proof To show that Φ is a bijection it suffices, by induction, to show that for a given increasing ordered tree T of size n, the specifications in the table above will enlarge $\Phi(P(T))$ using a full complement of dots: i top, $1 \leq i \leq n-1$ and i bot, $1 \leq i \leq n$. In the first four cases $\Phi(P(T))$ gets enlarged from T itself and in the last two cases from $F(T)$ (recall F is an involution).

Partition the child (non-root) vertices j of $P(T)$ into three classes: (1) Klazar violators, (2) Klazar compliant leaves, (3) Klazar compliant non-leaves. Cases $2,3,5$ hit j bot for j in classes (1), (2), (3) respectively. Now partition the child vertices of $P(T)$ in another way into two classes: ($1'$) partners (of some Klazar violator), ($2'$) non-partners. Case 6 hits j top for all j in class ($1'$), and, using Prop. 12, case 4 hits j top for all j in class ($2'$). Since n bot is hit by case 1, this proves that Φ is a bijection.

Finally, to verify that Klazar violator/partner pairs (i, j) are transformed into uplines from i bot to j top is a matter of checking cases. For example, in case 6, by induction and Prop. 13, $\Phi(P(F(T)))$ has an upline from each Klazar violator to its partner in $P(T)$ (and hence in T) but also one from j to $p(j)$. The latter upline, however, is destroyed when $p(j)$ top is used to enlarge $\Phi(P(F(T)))$. Other cases are left to the reader. $\square$

9.3 Explicit Φ

Here we give an explicit description of Φ as a composition of three bijections: (1) a tweaked version σ of the natural bijection from increasing ordered trees to build-tree codes, (2) the correspondence to build-matching codes, and (3) a tweaking τ of the natural bijection from build-matching codes to PM dot diagrams.

Definition of σ: Given an increasing ordered tree, obtain a build-tree code just as in the natural correspondence except that the involution F is applied to the original tree before recording (X_n, i_n) and to each succeeding pruned tree before recording (X_k, i_k). Conversely, σ^{-1} builds up the tree in the natural way except that F is applied to each intermediate tree before the next one is constructed. For example, the last tree in Fig. 3 corresponds under σ to the code $R_1, R_1, L_1, L_2, L_1, R_2$ as illustrated (F is not shown when it is the identity).

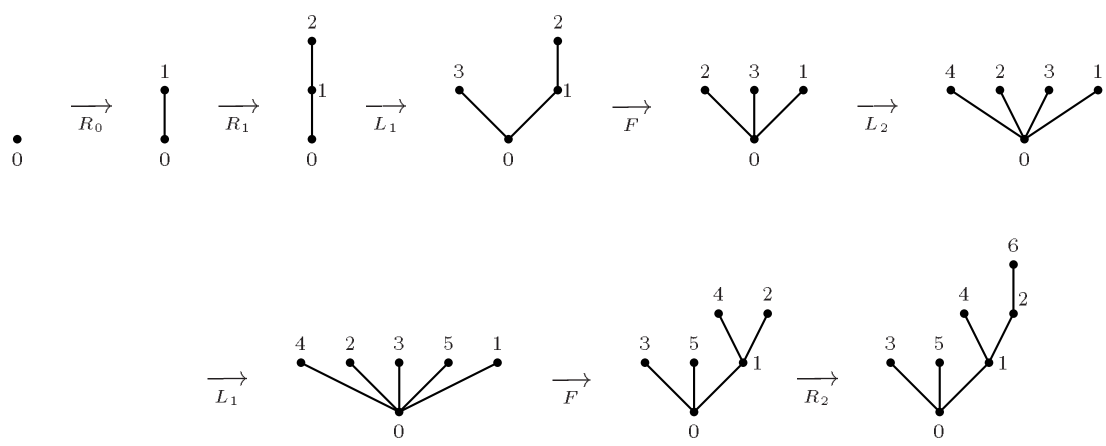
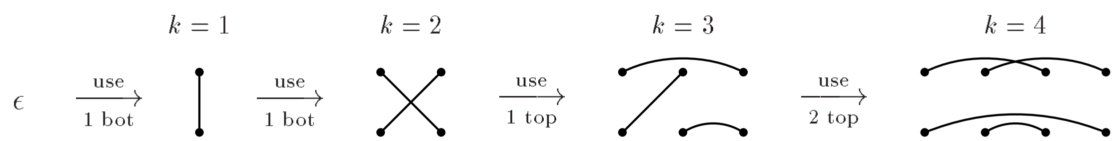


Figure 3

Proposition 15. *Under σ^{-1}, the pair (i, j) is a Klazar violator/partner pair iff (L, i) occurs an odd number of times in the build-tree code and j is the position in the code of the last occurrence of an (X, i) ($X = L$ or R).*

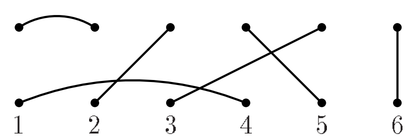
Proof First, adding a rightmost child from an $(R, -)$ entry in the code never introduces a Klazar violator in the increasing ordered tree. On the other hand, the first occurrence of an (L, i), say as the jth entry in the code, makes i a Klazar violator with partner j. Subsequent occurrences of (L, i) flip the violator/compliant status of i while occurrences of either (L, i) or (R, i) update the partner of i to the current position in the code whenever i is currently a violator. The result follows. □

Definition of τ: Given a build-matching code of length n, for $k = 1, 2, \dots, n$ enlarge (9.1) the current PM dot diagram (initially empty) using the vertex specified as follows. First, suppose $Y_k = B$. If $i_k = k$, use k bot; if $i := i_k < k$ and there is an upline from i in the current PM dot diagram, use i bot, otherwise use j_1 top where $j_1 \nearrow j_2 \nearrow \dots \nearrow i$ is the maximal run of uplines terminating at i with $j_1 = i$ if there is no upline terminating at i. Second, suppose $Y_k = T$. If there is an upline $i_k \nearrow j$ in the current PM dot diagram, use j top, otherwise use i_k bot. For example, the code B_1, T_1, B_2, T_1 successively yields (ϵ denotes the empty PM dot diagram)



Now we describe the inverse of τ. To prune a PM dot diagram means to delete the last dot in each row along with its incident edge and then join up their now-isolated partners unless the last dots were originally connected to each other in which case there is nothing to join up.

We will need the "shift" map S that takes a dot that starts an upline in a PM dot diagram to one that does not: if $i = i_1$ starts an upline then the upline ends at position $i_2 > i_1$. If i_2 starts another upline, proceed similarly to get $i_3 > i_2$ and so on, stopping at the first i_k that does not start an upline, and set $S(i) = i_k$.



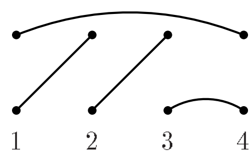
There is an upline from 2 to 3 and another from 3 to 5
but 5 does not start an upline. So $S(2) = 5$.

At the step where a PM dot diagram of size k is pruned to one of size $k-1$ record the pair (Y_k, i_k) of the build-matching code according as uplines are created and/or destroyed in the pruning process. First, if the last dots in each row are joined to each other, then $(Y_k, i_k) = (B, k)$. Otherwise, let i (resp. j) denote the position in its row of the partner of the last dot in the top (resp. bottom) row and consider four cases as in the Table.

case	row of i	row of j	restriction	upline created?	upline destroyed?	(Y_k, i_k)
1a	top	top	-	no	no	$(B, S(i))$
1b	top	bot	$i \le j$			
2	top	bot	$i > j$	yes	no	(T, j)
3a	bot	bot	-	no	yes	(T, i)
3b	bot	top	$i \ge j$			
4	bot	top	$i < j$	yes	yes	(B, i)

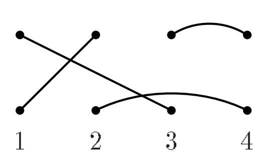
An example of each case is shown.

Case 1



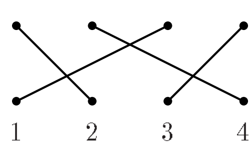
$i = 1$ [top], $j = 3$ [bot], $S(1) = 3$,
Record $(B, 3)$

Case 2



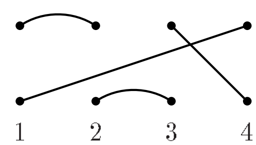
$i = 3$ [top], $j = 2$ [bot]
Record $(T, 2)$

Case 3



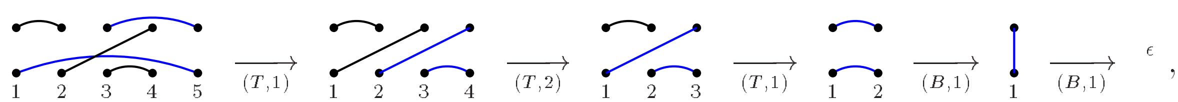
$i = 3$ [bot], $j = 2$ [top]
Record $(T, 3)$

Case 4



$i = 1$ [bot], $j = 3$ [top]
Record $(B, 1)$

The entire pruning process for the perfect matching 1 3 / 2 10 / 4 7 / 5 9 / 6 8 is shown:



yielding the build-matching code $\big((B,1),(B,1),(T,1),(T,2),(T,1)\big)$.

Proposition 16. *Under τ, the pair $i \nearrow j$ is an upline iff (T,i) occurs an odd number of times in the build-matching code and j is the position in the code of the last occurrence of a (Y,i) ($Y = T$ or B).*

Proof The first occurrence of T_i introduces an upline from i that remains undisturbed by each later Y_j with $j \neq i$, is switched to an upline from i to the new top end dot by B_i, and is killed by a second occurrence of T_i. A third occurrence of T_i reintroduces an upline from i. Thus an upline from i is present in the resulting PM dot diagram iff T_i occurs an odd number of times in the code and in this case the upline is $i \nearrow j$ where j is the position of the last Y_i in the code (Y may be B or T). □

That Φ sends Klazar violator/partner pairs (i,j) to uplines $i \nearrow j$ follows from Props. 15 and 16.

10 Trapezoidal Words

Riordan [7, p. 9] considered the Cartesian product $[1] \times [3] \times [5] \times \ldots \times [2n-1]$. He called its entries, $(a_k)_{1 \le k \le n}$, *trapezoidal words* and observed that the statistic "number of distinct entries" on trapezoidal words is distributed as what are now called the second-order Eulerian numbers, A008517. There is an obvious bijection from build-tree codes to trapezoidal words: $\big((X_k,i_k)\big)_{1 \le k \le n} \mapsto (a_k)_{1 \le k \le n}$ with $a_k = 2i_k$ if $X_k = L$ and $a_k = 2i_k + 1$ if $X_k = R$. Props. 15 and 16 now yield the following corollary.

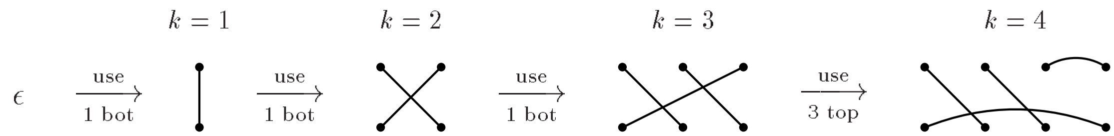
Corollary 17. *The following three statistics are equidistributed and all have the generating function*

$$\left(\frac{1-y}{2e^{x(y-1)} - 1 - y}\right)^{\frac{1}{2}}$$

of Cor. 4.

1. "# Klazar violators" on increasing ordered trees
2. "# uplines" on PM dot diagrams
3. "# even entries that occur an odd number of times" on trapezoidal words.

A variation of the bijection τ from build-matching codes to PM dot diagrams yields a generalization of the equidistribution of items 2 and 3 in Cor. 17. Given a build-matching code, this time build up a PM dot diagram by successively enlarging the current PM dot diagram as follows. First, suppose $Y_k = B$. If there is a weak downline from i use i top, otherwise use the partner of i (which may be top or bottom). Now suppose $Y_k = T$. If there is an upline $i \nearrow j$ from i use j top, otherwise use i bot. For example, the code B_1, T_1, B_2, T_1 successively yields



Corollary 18. *The joint distribution of the statistics "# even-to-odd matches" and "# odd-to-even matches" on perfect matchings of size n is the same as that of the statistics "# even entries that occur an odd number of times" and "# odd entries that occur an odd number of times" on trapezoidal words of length n.*

The generating function for "# uplines" in PM dot diagrams is given in Cor. 17 and it is not hard to find the analogous generating function for "# vertical lines":

$$\frac{1}{e^{x(1-y)}\sqrt{1-2x}}.$$

Is there a nice generating function for the joint distribution of the three statistics "# uplines", "# downlines", and "# vertical lines" in PM dot diagrams?

References

[1] Martin Klazar, Twelve countings with rooted plane trees, *European Journal of Combinatorics*, **18**, Issue 2, 1997, 195–210.

[2] Martin Klazar, Addendum to Twelve countings with rooted plane trees European Journal of Combinatorics Vol. 18, Issue 6, 1997, 739–740.

[3] The On-Line Encyclopedia of Integer Sequences, founded and maintained by Neil J. Sloane.

[4] Emeric Deutsch, Ordered trees with prescribed root degrees, node degrees, and branch lengths, *Discrete Math.* **282** (2004), 89-94.

[5] Ronald L. Graham, Donald E. Knuth, Oren Patashnik, *Concrete Mathematics* (2nd edition), Addison-Wesley, 1994.

[6] Arthur T. Benjamin and Jennifer J. Quinn, *Proofs That Really Count: The Art of Combinatorial Proof*, Dolciani Mathematical Expositions, Vol. 27, Mathematical Association of America, 2003.

[7] John Riordan, The blossoming of Schröders fourth problem, *Acta Math.* **137**, 1976, 1–16.