An Algebra for Proper Generating Trees

D. Merlini, R. Sprugnoli, M. C. Verri
Dipartimento di Sistemi e Informatica
via Lombroso 6/17, 50134, Firenze, Italia
[merlini,sprugnoli,verri]@dsi.unifi.it

Abstract

We find an algebraic structure for a subclass of generating trees by introducing the concept of *marked generating trees*. In these kind of trees, labels can be marked or non marked and the count relative to a certain label at a certain level is given by the difference between the number of non marked and marked labels. The algebraic structure corresponds to a non commutative group with respect to a *product* operation between two generating trees. Hence we define *the identity* generating tree and the *inverse* of a given generating tree.

1 Introduction

The concept of a *proper Riordan Array* (pRA, for short) is very useful in Combinatorics. The infinite triangles of Pascal, Catalan, Motzkin and Schröder are important and meaningful examples of pRA's, but many applications have been proposed and developed, thus proving the effectiveness of a Riordan Array approach to many combinatorial problems (see, e.g. [3, 4, 5, 9, 10]). Without doubt, the relevance of pRA's derives from the fact that a pRA D is completely and univocally described by a couple of *formal power series* (fps, for short): $D = (d(t), h(t))$, which can be seen as generating functions of sequences related to the array. This implies that a Riordan Array approach to a problem can take advantage from the concept of generating function and from the well-understood apparatus of formal power series. For example, this may concern the extraction of coefficients and the computation of asymptotic values, when fps are considered as analytic functions.

The algebraic structure of pRA's was one of the first properties pointed out (see, e.g., [8, 9]). In fact, they constitute a group by the usual row-by-column product, and this product can be easily defined through the fps describing the pRA's involved in the operation. Besides, a pRA D and its inverse $\bar{D}$ are related by the Lagrange Inversion Formula, which therefore becomes a fundamental tool in the theory and the practice of pRA's.

Recently, Merlini and Verri [5] pointed out an important connection between pRA's and *generating trees*. Generating trees are becoming more and more important in Combinatorics (see e.g., [1, 11, 12]), as a device to represent the development of many classes of combinatorial objects, which can then be counted by counting the different labels in the various levels of the tree. The proved relation between pRA's and generating trees allows to combine the counting capabilities of both approaches and thus improve our understanding of the problem under consideration. Therefore, we feel that a further, more deep insight to both pRA's and generating trees is worth studying, and this is the aim of the present paper.

There are two problems to which we dedicate our attention:

1. The definition of a pRA does not impose conditions on the fps involved, except that they should be invertible. On the other hand, it is clear that a fps with real coefficients (where *real* is intended as opposed to *integer*) cannot be seen as a counting generating function. We have important combinatorial examples of pRA's with rational coefficients, as the triangles related to Stirling numbers of both kinds (see e.g., [9, 10]). However, in general, only pRA's with integer coefficients can have a direct combinatorial interpretation. So, we will restrict our attention to fps $f(t) = \sum_{k=0}^{\infty} f_k t^k$ having $f_0 = 1$ and $f_k \in \mathbf{Z}$, for every $k \in \mathbf{N}$. These fps will be called *monic, integer fps*, and we will prove a number of their properties. Consequently, we define *monic, integer pRA's* as pRA's whose elements are in $\mathbf{Z}$ and those in the main diagonal are 1. These arrays are the main object of our investigations.

2. Several different combinatorial interpretations can be found for monic, integer pRA's, if we can assign a combinatorial meaning to negative values. As a matter of fact, we show that a combinatorial interpretation is possible through a suitable variant of generating trees. To this purpose, we introduce the concept of a *marked label* to be considered together with usual labels: generating trees can be extended to deal with negative values if we consider a node labelled k as opposed to a node labelled $\bar{k}$ (i.e., labelled with a *marked* k) and imagine that they annihilate each other. As far as we know, the concept of marked labels has been implicitly used for the first time in [5]; the same concept has been used in [2] in relation with the definition of the *ECO-systemes signés*. A quite different approach can be found in [6] where the authors deal with what they call *coloured rules*; these correspond to generating trees in which nodes with the same label can have various colours; what makes this approach different is the fact that these labels don't annihilate each other and the count relative to a level in the tree is exactly the number of nodes at that level.

   By introducing marked labels, the correspondence between the extended generating trees and monic, integer pRA's becomes effective. We wish to point out that, as a consequence of this correspondence, we are in a position to define the *product* of two generating trees, the *inverse* of a given generating tree and the *identity* generating tree, three concepts that can be rather surprising.

The structure of this paper is straight-forward. In Section 2. we develop the concept of a monic, integer fps and that of a monic, integer pRA. Then we show some general properties of pRA's to relate them to generating trees: these new properties mainly concern the A- and Z-sequences of a pRA, are valid in the general case, but will then be used in the monic, integer case. In Section 3. we give our main theorem, relating extended generating trees to monic, integer pRA's; finally, we show a number of examples, illustrating some applications of the main theorem and the two new concepts of the inverse and the product of generating trees. In order to safe space we omit all Theorem proofs in this paper.

2 Riordan Arrays

Let $\mathcal{F} = \mathbf{C}[\![t]\!]$ be the integral domain of formal power series over the complex field. Usually, in Combinatorics we refer to this domain because its algebraic structure is well-known. However, only a proper subset of $\mathcal{F}$ has a direct interest: when we use fps as counting generating functions, we implicitly assume that their coefficients are (positive) integer numbers or, at most, are rational numbers in the case of exponential generating functions. Obviously, during some computations, more general situations can arise; the reader can think of signed binomial coefficients, signed Stirling numbers and Bernoulli numbers. However, the set of fps with integer coefficients has so a direct combinatorial relevance that it is interesting to isolate and try to characterise them in some algebraic way. In particular, we wish to study fps with integer coefficients when the t^0 coefficient is 1.

Let $\mathcal{Z} = \mathbf{Z}[\![t]\!]$ be the set of fps $f(t) = \sum_{k=0}^{\infty} f_k t^k$ such that $f_0 = 1$ and $f_k \in \mathbf{Z}$, for every $k > 0$. As we are now going to see, this set has a rather rich algebraic structure and constitutes the basis of a number of interesting developments concerning proper Riordan Arrays and generating trees.

The following lemma describes the algebraic properties of the Cauchy product.

Lemma 2.1 *Let $\cdot$ (or the simple juxtaposition) denote the Cauchy product; then $(\mathcal{Z}, \cdot)$ is a commutative group.* ■

Generating functions are a device to deal with sequences of numbers, in the sense that we study a fps $f(t) = \sum_{k=0}^{\infty} f_k t^k$ instead of studying the sequence $\{f_0, f_1, f_2, \ldots\}$. When we pass from one dimensional sequences to two dimensional ones $\{d_{n,k}\}_{n,k \in N}$, a relevant concept is given by proper Riordan Arrays. They are defined by a couple of fps $(d(t), h(t))$, having $d(t), h(t) \in \mathcal{F}$, and the generic element of the two dimensional sequence induced by the pRA is $d_{n,k} = [t^n]d(t)(th(t))^k$. Many combinatorial triangles are examples of pRA's, like the Pascal, Catalan, Motzkin and Schröder triangles. In all these cases the elements are integer and those in the main diagonal are 1. Therefore, it seems appropriate to study monic, integer pRA's, and we can prove that these arrays are strictly related to fps in $\mathcal{Z}$. Actually, we have:

Theorem 2.2 *Let $D = (d(t), h(t))$ be a pRA: D is a monic, integer pRA if and only if $d(t)$ and $h(t)$ are monic, integer fps.* ■

If $\mathcal{R}$ denotes the set of pRA's, it is well-known that $(\mathcal{R}, \star)$, where $\star$ is the usual row-by-column product, is a non-commutative group, called the *Riordan group*. Actually, the operation $\star$ can also be given in terms of the fps involved in the definition of the pRA's. More specifically, let $D = (d(t), h(t))$ and $E = (f(t), g(t))$ be two pRA's; the generic element in $F = D \star E$ is:

$$F_{n,k} = \sum_j D_{n,j} E_{j,k} = \sum_j [t^n] d(t)(th(t))^j [t^j] f(t)(tg(t))^k =$$

$$= [t^n] d(t) \sum_j (th(t))^j [t^j] f(t)(tg(t))^k = [t^n] d(t) f(th(t))(th(t)g(th(t)))^k.$$

This is just the generic element of a pRA:

$$(d(t), h(t)) \star (f(t), g(t)) = (d(t)f(th(t)), h(t)g(th(t))).$$

This shows that $\mathcal{R}$ is closed under $\star$ and by setting $(d(t), h(t)) \star (\bar{d}(t), \bar{h}(t)) = (1, 1)$, the neutral element with respect to $\star$, or $d(t)\bar{d}(th(t)) = 1$ and $h(t)\bar{h}(th(t)) = 1$ we find:

$$\bar{d}(y) = \left[\frac{1}{d(t)} \middle| y = th(t) \right] \qquad \bar{h}(y) = \left[\frac{1}{h(t)} \middle| y = th(t) \right].$$

This notation means that we should first solve the functional equation $y = th(t)$ in t in order to find a function $t = t(y)$, and exactly the solution for which $t(0) = 0$, and then substitute this fps into $1/d(t)$ in order to find $\bar{d}(y) = 1/d(t(y))$: this fps is the d function in the inverse pRA, i.e. in the pRA $\bar{D}$ such that $D * \bar{D} = (1, 1) = I$. An analogous procedure should be followed to find $\bar{h}(t)$. The existence and uniqueness of $\bar{d}$ and $\bar{h}$ is garanted by the fact that the functional equation $y = th(t)$ has a unique solution $t = t(y)$ with $t(0) = 0$ if and only if $h(0) \neq 0$. An important result is:

Theorem 2.3 *If D is a monic integer pRA, then also its inverse $\bar{D}$ is such.* ■

The way to represent a pRA by means of a couple of fps: $D = (d(t), h(t))$ is not the only method to deal with this sort of infinite triangles. In 1978 Rogers [7] proved that pRA have a remarkable property: every element $d_{n+1,k+1}$ not belonging to column 0 or row 0, can be expressed as a linear combination of the elements in the previous row, starting with the element in column k:

$$d_{n+1,k+1} = a_0 d_{n,k} + a_1 d_{n,k+1} + a_2 d_{n,k+2} + \ldots = \sum_{j=0}^{\infty} a_j d_{n,k+j}.$$

The sum is actually finite and the sequence $A = \{a_0, a_1, a_2, \ldots\}$ is independent of n and k, that is, it characterizes the whole triangle, except for column 0 and row 0; A is called the $A-$sequence of the pRA. Actually, column 0 is given by the function $d(t)$, and row 0 is all composed of 0, except for the element $d_{0,0} = d_0$, which is determined by $d(t)$. In this way, the pRA D can be represented by the couple $(d(t), A(t))$, if $A(t)$ is the generating function of the A-sequence. It can be shown that $A(t)$ is univocally determined by the function $h(t)$, and we have:

$$h(t) = A(th(t)) \qquad \text{or} \qquad A(y) = [h(t) \,|\, y = th(t)] \tag{2.1}$$

The elements of column 0 can also be expressed in terms of all the elements of the previous row, but obviously the dependence should start with the same column 0. In this case, we can show that a new sequence exists: $Z = \{z_0, z_1, z_2, \ldots\}$, called the Z-sequence of the pRA, such that:

$$d_{n+1,0} = z_0 d_{n,0} + z_1 d_{n,1} + z_2 d_{n,2} + \ldots = \sum_{j=0}^{\infty} z_j d_{n,j}.$$

The sum is obviously finite, the Z-sequence can have no relation with the A-sequence, and, in fact, it depends on both the $d(t)$ and $h(t)$ functions: for $Z(t)$, the generating function of the Z-sequence, we have:

$$d(t) = \frac{d_{0,0}}{1 - tZ(th(t))} \qquad \text{or} \qquad Z(y) = \left[\frac{d(t) - d_0}{td(t)} \middle| \, y = th(t) \right] \tag{2.2}$$

where $d_{0,0} = d_0$. At this point, we have a third characterisation of pRA: a pRA D is uniquely determined by the triple $(d_{0,0}, A(t), Z(t))$, where $A(t)$ and $Z(t)$ are the generating functions of the A- and Z-sequences of D. By the previous lemma, it is now immediate to observe that a monic integer pRA corresponds to a monic-integer A-sequence, and vice versa if also $d(t) \in \mathbf{Z}[\![t]\!]$. Instead, the Z-sequence is integer, but it can be non-monic; in fact, from the second formula in (2.2) we see that z_0 is related to d_1 and not to d_0. It can be interesting to see how the A- and Z-sequences are changed when we perform operations on pRA, essentially when we invert one triangle or when we multiply two triangles. Let us begin with the A-sequence, whose inversion is particularly easy:

Theorem 2.4 *Let $D = (d(t), h(t))$ be a pRA and $\bar{D} = (\bar{d}(t), \bar{h}(t))$ its inverse. Then $\bar{A}(t) = 1/h(t)$.* ■

More interesting is the case of row-by-column product:

Theorem 2.5 *Let $D = (d(t), h(t))$, $E = (e(t), k(t))$ be two pRA and $F = (f(t), g(t))$ their product. If $A_D(t)$,$A_E(t)$ and $A_F(t)$ are the generating functions of the corresponding A-sequences, we have:*

$$A_F(t) = A_E(t) \left[A_D(y) \middle| \, t = yk(y) \right].$$

■

An example is in order here. Let D be the Pascal triangle and E the Catalan triangle, so that we have

$$D = \left(\frac{1}{1-t}, \frac{1}{1-t} \right) \qquad E = \left(\frac{1 - \sqrt{1-4t}}{2t}, \frac{1 - \sqrt{1-4t}}{2t} \right)$$

and $A_D(t) = 1 + t$, $A_E(t) = 1/(1-t)$. Numerically we have:

$$\begin{pmatrix} 1 & & & & \\ 1 & 1 & & & \\ 1 & 2 & 1 & & \\ 1 & 3 & 3 & 1 & \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix} \star \begin{pmatrix} 1 & & & & \\ 1 & 1 & & & \\ 2 & 2 & 1 & & \\ 5 & 5 & 3 & 1 & \\ 14 & 14 & 9 & -4 & 1 \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ 2 & 1 & & & \\ 5 & 4 & 1 & & \\ 15 & 14 & 6 & 1 & \\ 51 & 50 & 27 & 8 & 1 \end{pmatrix}$$

In this case, the equation $t = yk(y)$ is $t = \left(1 - \sqrt{1-4y}\right)/2$ and its solution is $y = t - t^2$, so that $A_D(y) = 1 + y = 1 + t - t^2$ and

$$A_F(t) = \frac{1 + t - t^2}{1 - t} = 1 + 2t + t^2 + t^3 + t^4 + \dots .$$

This is immediately checked against the numerical values just obtained.

For what concerns the Z-sequence, we can prove:

Theorem 2.6 *Let $D = (d(t), h(t))$ and $\bar{D} = (\bar{d}(t), \bar{h}(t))$ its inverse pRA; then if $Z(t)$, $\bar{Z}(t)$ are the generating functions of the corresponding Z-sequences, we have*

$$\bar{Z}(t) = \frac{d_0 - d(t)}{d_0 th(t)} = \frac{-Z(th(t))}{h(t)(1 - tz(th(t))}.$$

∎

For the row-by-column product we have:

Theorem 2.7 *Let $D = (d(t), h(t))$, $E = (e(t), k(t))$ be two pRA with Z-sequences $Z_D(t)$ and $Z_E(t)$. Then if $F = D \star E$, the Z-sequence of this product is*

$$Z_F(t) = \left[Z_D(y) + d_0 \frac{\bar{d}(y)}{\bar{h}(y)} Z_E(t) \middle| \, t = yk(y) \right].$$

∎

3 Generating trees

In the paper by Merlini and Verri [5], it is shown that a particular subset of generating trees has a correspondence with some pRA's. In fact, they define an infinite matrix $\{d_{n,k}\}_{n,k \in \mathbf{N}}$ to be *associated to a generating tree* with root (c) (AGT matrix for short) if $d_{n,k}$ is the number of nodes at level n with label $k + c$ and show that, under suitable conditions, this matrix corresponds to a pRA, and vice versa. This result allows us to use the Riordan Array approach to obtain a variety of counting results on problems which can be described by such generating trees. In the present paper we extend the correspondence between generating trees and pRA's to the whole group of monic integer pRA's. The primary difficulty in doing so is the fact that coefficients in pRA's can be negative, and this requires a particular interpretation of the labels in a generating tree. The idea is rather simple, but effective. Since, as we are going to show, the construction of the generating tree related to a given monic integer pRA requires the concept of the $A-$ and $Z-$sequences, the results obtained in the previous section will be important in applying the theory.

In order to specify a generating tree we have to specify a label for the root and a set of rules explaining how to derive from the label of a parent the labels of all of its children. For example, Figure 3.1 illustrates the upper part of the generating tree which corresponds to the following specification:

$$\begin{cases} root: & (2) \\ rule: & (k) \quad \to (k)(k+1) \end{cases} \tag{3.3}$$

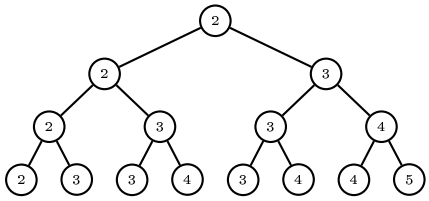

Figure 3.1: The Pascal generating tree: specification (3.3)

The first triangle in Table 3.1 illustrates the AGT matrix associated to the generating tree specification (3.3). Our main result is given by Theorem 3.3,

n/k	0	1	2	3	4
0	1				
1	1	1			
2	1	2	1		
3	1	3	3	1	
4	1	4	6	4	1

n/k	0	1	2	3	4
0	1				
1	-1	1			
2	1	-2	1		
3	-1	3	-3	1	
4	1	-4	6	-4	1

Table 3.1: The Pascal triangle and its inverse

which extends Theorem 3.9 in [5]. The concept of a generating tree is extended to deal with *marked* labels: a label is any positive integer, generated according to the generating tree specification; a *marked* label is any positive integer, marked by a bar, for which appropriate rules are given in the specification:

Definition 3.1 *A marked generating tree is a rooted labelled tree (the labels can be marked or non-marked) with the property that if v_1 and v_2 are any two nodes with the same label then, for each label l, v_1 and v_2 have exactly the same number of children with label l. To specify a generating tree it therefore suffices to specify:*

1) the label of the root;

2) a set of rules explaining how to derive from the label of a parent the labels of all of its children.

From here on we will use the term "generating tree" to denote a marked generating tree.

A simple example is given by the following generating tree specification:

$$\left\{ \begin{array}{lll} root: & (2) & \\ rule: & (k) & \rightarrow (\overline{k})(k+1) \\ & (\overline{k}) & \rightarrow (k)(\overline{k+1}) \end{array} \right. \tag{3.4}$$

The first 4 levels of the corresponding generating tree are shown in Figure 3.2.

The idea is that marked labels kill or annihilate the non-marked labels with the same number, i.e. the count relative to an integer j is the difference between the number of non-marked and marked labels j at a given level. This gives a negative

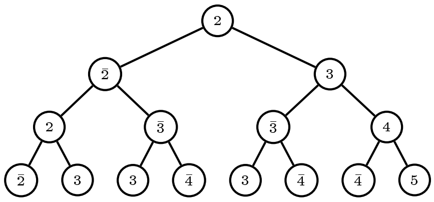

Figure 3.2: The *inverse* of Pascal generating tree: specification (3.4)

count if marked labels are more numerous than non-marked ones. Thus we can extend the concept of an AGT matrix in the following way:

Definition 3.2 *An infinite matrix $\{d_{n,k}\}_{n,k\in\mathbf{N}}$ is said to be "associated" to a marked generating tree with root (c) (AGT matrix for short) if $d_{n,k}$ is the difference between the number of nodes at level n with label $k+c$ and the number of nodes with label $\overline{k+c}$. By convention, the level of the root is 0.*

The second triangle in Table 3.1 corresponds to the AGT matrix associated to the specification (3.4).

Before stating our main result we introduce the following notations for generating tree specifications.

$(x) = (\bar{\bar{x}});$
$(x)^p = \underbrace{(x)\cdots(x)}_{p},\ p \geq 0$
$(x)^p = \underbrace{(\bar{x})\cdots(\bar{x})}_{-p},\ p < 0$
$\overline{(x)^p} = (\overline{x})^p,\ p > 0$
$\overline{(x)^p} = (x)^{-p},\ p < 0$
$\prod_{j=0}^{i}(k-j)^{\alpha_j} = (k)^{\alpha_0}(k-1)^{\alpha_1}\cdots(k-i)^{\alpha_i}$

We note that $(x)^0$ is the empty sequence. We can finally state the following important theorem which relates monic integer pRA's with marked generating trees:

Theorem 3.3 *Let $c \in \mathbf{N}$, $a_j, b_k \in \mathbf{Z}$, $\forall j \geq 0$ and $k \geq c$, $a_0 = 1$, and let*

$$\left\{ \begin{array}{lll} root: & (c) & \\ rule: & (k) & \to (c)^{b_k} \prod_{j=0}^{k+1-c}(k+1-j)^{a_j} \\ & (\overline{k}) & \to \overline{(c)^{b_k}}\, \prod_{j=0}^{k+1-c} \overline{(k+1-j)^{a_j}} \end{array} \right. \tag{3.5}$$

be a marked generating tree specification. Then, the AGT matrix associated to (3.5) is a monic integer pRA D defined by the triple (d_0, A, Z), such that

$$d_0 = 1, \quad A = (a_0, a_1, a_2, \ldots), \quad Z = (b_c + a_1, b_{c+1} + a_2, b_{c+2} + a_3, \ldots).$$

Vice versa, if D is a monic integer pRA defined by the triple $(1, A, Z)$ with $a_j, z_j \in \mathbf{Z}$, $\forall j \geq 0$ and $a_0 = 1$, then D is the AGT matrix associated to the generating tree specification (3.5) with $b_{c+j} = z_j - a_{j+1}$, $\forall j \geq 0$. ■

A generating tree corresponding to the specification (3.5) will be called a *proper generating tree.*

Since monic integer pRA's constitute a non-commutative group with respect to the usual row by colum product, as we have seen in Section 2, an important consequence of Theorem 3.3 is that we can define the *product* of two proper generating trees and the *inverse* of a proper generating tree:

Definition 3.4 *Given two generating tree specifications t_1 and t_2 of type (3.5) and the corresponding AGT matrices T_1 and T_2, we define the generating tree specification product of t_1 and t_2 as the specification t_3 having $T_3 = T_1 \star T_2$ as AGT matrix.*

Definition 3.5 *Given a generating tree specification t_1 of type (3.5) and the corresponding AGT matrix T_1, we define the generating tree specification inverse of t_1 as the specification t_2 having $T_2 = T_1^{-1}$ as AGT matrix.*

To complete the definition of the algebraic structure of generating trees we only need to define an *identity*:

Definition 3.6 *The identity generating tree specification t_I is the one having the identity matrix I as AGT matrix. The specification and the corresponding generating trees are shown in (3.6) and Figure 3.3.*

$$\begin{cases} root: & (c) \\ rule: & (k) \quad \rightarrow (k+1) \end{cases} \tag{3.6}$$

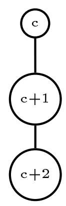

Figure 3.3: The *identity* generating tree

For example, specification (3.4) is the inverse of (3.3), as can be easily verified by using formulas in Section 2. In fact, for the Pascal triangle we have $d_0 = 1$, $A = \{1, 1, 0, 0, \ldots\}$ and $Z = \{1, 0, 0, \ldots\}$ and for its inverse $\bar{d}_0 = 1$, $\bar{A} = \{1, -1, 0, 0, \ldots\}$ and $\bar{Z} = \{-1, 0, 0, \ldots\}$.

In what follows, we examine some other generating tree specifications, very well known in literature, by finding for each of them the corresponding *inverse*. We leave to the reader the necessary algebraic computations, which can be easily performed by hand or by some symbolic system.

The first specification is related to Motzkin numbers $M_j = \{1, 1, 2, 4, 9, \ldots\} = [t^j](1 - t - \sqrt{1 - 2t - 3t^2})/(2t)$:

$$\left\{ \begin{array}{lll} root: & (1) & \\ rule: & (k) & \to (1) \ldots (k-1)(k+1) \end{array} \right. \tag{3.7}$$

(we observe that $(1) \to (2)$) and a partial generating tree is illustrated in Figure 3.4. The specification inverse of (3.7) is the following:

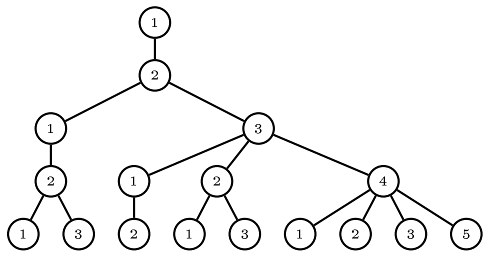

Figure 3.4: The Motzkin generating tree: specification (3.7)

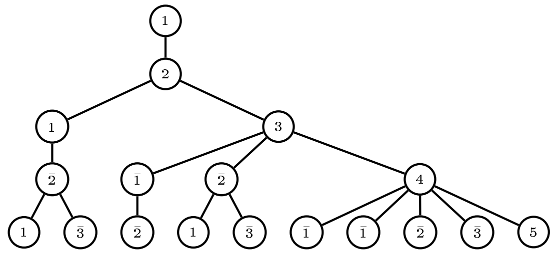

Figure 3.5: The *inverse* of Motzkin generating tree: specification (3.8)

$$\left\{ \begin{array}{lll} root: & (1) & \\ rule: & (k) & \to (k+1) \prod_{j=2}^{k} (\overline{k+1-j})^{M_{j-2}} \\ & (\overline{k}) & \to (\overline{k+1}) \prod_{j=2}^{k} (k+1-j)^{M_{j-2}} \end{array} \right. \tag{3.8}$$

and is illustrated in Figure 3.5. Finally, Table 3.2 shows the AGT matrices associated to the specifications (3.7) and (3.8).

n/k	0	1	2	3	4
0	1				
1	0	1			
2	1	0	1		
3	1	2	0	1	
4	3	2	3	0	1

n/k	0	1	2	3	4
0	1				
1	0	1			
2	-1	0	1		
3	-1	-2	0	1	
4	0	-2	-3	0	1

Table 3.2: The Motzkin triangle and its inverse

The following example is related to the Catalan numbers $C_j = \{1, 1, 2, 5, 14, \ldots\} = \frac{1}{j+1}\binom{2j}{j}$. In fact, specification (3.9), illustrated in Figure 3.6, has the Catalan triangle as AGT matrix (see the first triangle in Table 3.3).

$$\begin{cases} root: & (2) \\ rule: & (k) \quad \rightarrow (2) \cdots (k)(k+1) \end{cases} \tag{3.9}$$

We point out that specification (3.9) represents the *product* between the speci-

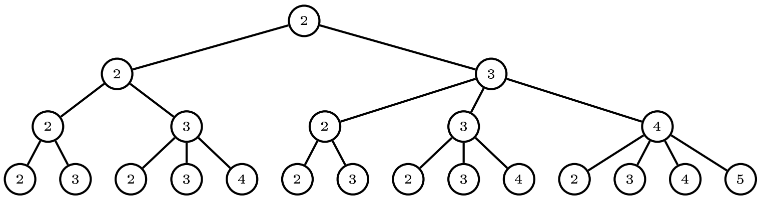

Figure 3.6: The Catalan generating tree: this is the *product* between the Pascal and the Motzkin trees: specification (3.9)

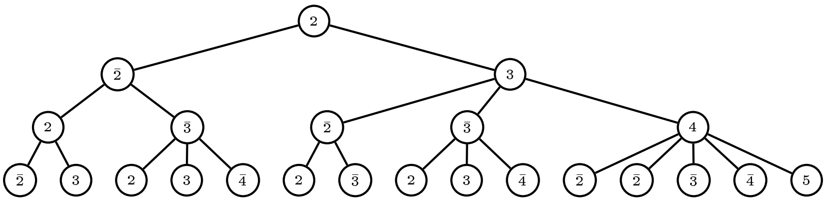

Figure 3.7: The *inverse* of Catalan generating tree: specification (3.10)

fication (3.3) and the specification (3.7). Figure 3.7 gives the inverse of Catalan generating tree; the AGT matrix is shown in Table 3.3.

$$\begin{cases} root: & (2) \\ rule: & (k) \quad \rightarrow (k+1)\prod_{j=1}^{k-1}(\overline{k+1-j})^{C_{j-1}} \\ & (\bar{k}) \quad \rightarrow (\overline{k+1})\prod_{j=1}^{k-1}(k+1-j)^{C_{j-1}} \end{cases} \tag{3.10}$$

n/k	0	1	2	3	4
0	1				
1	1	1			
2	2	2	1		
3	5	5	3	1	
4	14	14	9	4	1

n/k	0	1	2	3	4
0	1				
1	-1	1			
2	0	-2	1		
3	0	1	-3	1	
4	0	0	3	-4	1

Table 3.3: The Catalan triangle and its inverse

The last example is related to the Fibonacci numbers $F_j = \{1, 1, 2, 3, 5, 8, \ldots\} = [t^j]1/(1-t-t^2)$. We give the two specifications (3.11) and (3.12), illustrated in

Figures 3.8 and 3.9 respectively, and the corresponding AGT matrices in Table 3.4.

$$\begin{cases} root: & (2) \\ rule: & (k) \quad \rightarrow (2)^{k-1}(k+1) \end{cases} \tag{3.11}$$

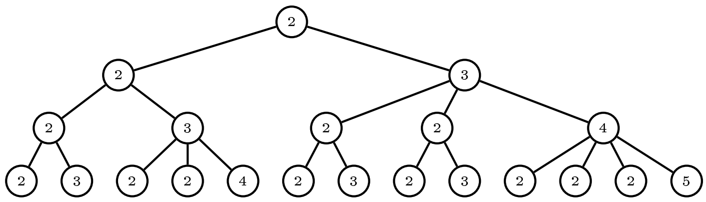

Figure 3.8: The odd Fibonacci generating tree: specification (3.11)

$$\begin{cases} root: & (2) \\ rule: & (k) \quad \rightarrow (\overline{2})^{F_{2k-3}}(k+1) \\ & (\bar{k}) \quad \rightarrow (2)^{F_{2k-3}}(\overline{k+1}) \end{cases} \tag{3.12}$$

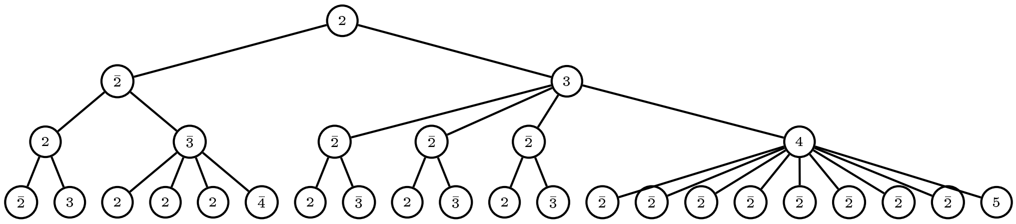

Figure 3.9: The *inverse* of odd Fibonacci generating tree: specification (3.12)

n/k	0	1	2	3	4
0	1				
1	1	1			
2	3	1	1		
3	8	3	1	1	
4	21	8	3	1	1

n/k	0	1	2	3	4
0	1				
1	-1	1			
2	-2	-1	1		
3	-3	-2	-1	1	
4	-4	-3	-2	-1	1

Table 3.4: The odd Fibonacci triangle and its inverse

References

[1] C. Banderier, M. Bousquet-Mélou, A. Denise, P. Flajolet, D. Gardy, and D. Gouyou-Beauchamps. On generating functions of generating trees. *INRIA Research Report RR-3661. In Proceedings of Formal Power Series and Algebraic Combinatorics (FPSAC'99), Barcelona*, 1999.

[2] S. Corteel. Séries génératrices exponentielles pour les eco-systèmes signés. *Proceedings of the 12-th International Conference on Formal Power Series and Algebraic Combinatorics, Moscow*, 2000.

[3] D. Merlini, D. G. Rogers, R. Sprugnoli, and M. C. Verri. On some alternative characterizations of Riordan arrays. *Canadian Journal of Mathematics*, 49(2):301–320, 1997.

[4] D. Merlini, R. Sprugnoli, and M. C. Verri. Algebraic and combinatorial properties of simple, coloured walks. In *Proceedings of CAAP'94*, volume 787 of *Lecture Notes in Computer Science*, pages 218–233, 1994.

[5] D. Merlini and M. C. Verri. Generating trees and proper Riordan Arrays. *Discrete Mathematics*, 218:167–183, 2000.

[6] E. Pergola, R. Pinzani, and S. Rinaldi. Towards an algebra of succession rules. *Proceedings of MATHINFO2000*.

[7] D. G. Rogers. Pascal triangles, Catalan numbers and renewal arrays. *Discrete Mathematics*, 22:301–310, 1978.

[8] L. W. Shapiro, S. Getu, W.-J. Woan, and L. Woodson. The Riordan group. *Discrete Applied Mathematics*, 34:229–239, 1991.

[9] R. Sprugnoli. Riordan arrays and combinatorial sums. *Discrete Mathematics*, 132:267–290, 1994.

[10] R. Sprugnoli. Riordan arrays and the Abel-Gould identity. *Discrete Mathematics*, 142:213–233, 1995.

[11] J. West. Generating trees and the Catalan and Schröder numbers. *Discrete Mathematics*, 146:247–262, 1995.

[12] J. West. Generating trees and forbidden subsequences. *Discrete Mathematics*, 157:363–374, 1996.