Contents

Two New One-to-one Correspondences on Trees

Robert Muth

March 27, 1997

Abstract. The classical one-to-one correspondence between ordered trees with $n+1$ nodes and binary trees with n nodes is generalized in two ways:

1. monotonically labeled ordered trees $\leftrightarrow$ multi-dimensional binary trees
2. monotonically labeled binary trees $\leftrightarrow$ multi-dimensional ordered trees

Simple generating functions for these tree classes are presented from which combinatorial information can be computed.

1 Introduction

In this paper we present two new extensions of the classical correspondence between binary trees with n nodes and ordered trees with $n+1$ nodes.

After reviewing some basic concepts in section 2 we shall devote section 3 to formalizing the classical correspondence by describing graphical operations that transform ordered trees into binary trees and vice versa.

The generalized families of ordered and binary trees belong to the class of "simply generated" trees — thus their generating functions satisfy certain (simple) types of functional equations. Given these functional equations combinatorial information can be computed by inversion. Before we describe the new correspondences we shall extend the class of simply generated tree families to the classes of:

- Multidimensional simply generated tree families (section 4).
- Monotonically labeled simply generated tree families (section 5).

The new correspondences involve familes from both classes.

In section 6 we present the corresponence between monotonically labeled ordered trees and multidimensional binary trees. In section 7 we present the corresponence between monotonically labeled binary trees and multidimensional ordered trees. Both correspondences reduce to the classical correspondence in the case of 1-dimensional/1-labeled trees. Both correspondences will be stated as graphical operations that show how trees from one family can be transformed into trees from the corresponding family.

In section 8 we shall exploit the new correspondences to obtain results about the average label distribution in monotonically labeled ordered and binary trees with n nodes.

2 Fundamentals

Definition 2.1 (Tree) *A rooted directed graph $t = (V, E)$ with root $r(t) \in V$ is called an (unordered,oriented) tree, if $indeg(r(t)) = 0$ and $indeg(v) = 1$ for all other nodes (we will use the term node instead of vertex throughout this paper).*

If $(v_1, v_2) \in E$, v_1 is called the father of v_2 and v_2 is called son of v_1.

Two nodes v_1 and v_2 are called brothers, if they are sons of the same father.

If there is a path from node v_1 to node v_2, we call v_1 a predecessor of v_2 and v_2 a successor of v_1.

The tree $t' = (V', E')$ with root $r(t')$, is called a subtree of $t = (V, E)$ if $V' \subset V$ and $E' = E \cap (V' \times V')$ where V' consists of $r(t')$ and all its successors in t.

The size of a tree $t = (V, E)$ which is denoted by $|t|$ is the number of its vertices ($|V|$).

In figures the root will be the topmost node. Edges are directed downwards, i.e. leading away from the root. The empty tree, i.e. the tree with no nodes and edges, will be depicted by a box ($\square$).

Definition 2.2 (Generating Function) *Let F be a family of trees and let $e : F \to \mathbb{N}_0$ be a mapping into the natural numbers. The generating function $F(z)$ of F (for e) is then given by the following (formal) power series:*

$$F(z) = \sum_{n \in \mathbb{N}_0} |e^{-1}(n)|\, z^n$$

Conversely, if $F(z)$ is a given generating function , $[z^n]F(z)$ denotes the coefficient of z^n in the formal taylor expansion around the origin. By the inversion theorem, this coefficient will be $|e^{-1}(n)|$.

If e is a mapping $F \to \mathbb{N}_0^d$, we obtain in an analogous manner a generating function in d variables — a multivariate generating function :

$$\widetilde{F}(z_1, \ldots, z_d) = \sum_{(n_1,\ldots,n_d) \in \mathbb{N}_0^d} |\, e^{-1}(n_1, \ldots, n_d)|\, z_1^{n_1} \cdot \ldots \cdot z_d^{n_d}$$

and analogously:

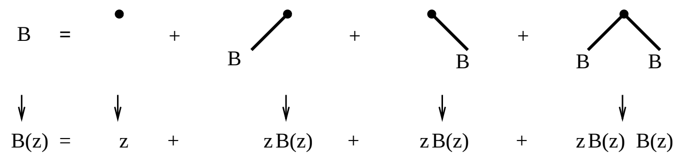
$$[z_1^{n_1} \cdot \ldots \cdot z_d^{n_d}]\widetilde{F}(z_1, \ldots, z_d) = |\, e^{-1}(n_1, \ldots, n_d)|$$

Unless otherwise indicated, we shall choose the function e, used in forming our generating functions to be the size of the tree, i.e. $e(t) = |t|$.

In [Fla88] Flajolet describes an intuitive way to translate recursively defined combinatorial objects into functional equations defining their generating functions . The method is successful for many “size” functions e.

Giving a detailed discussion of the translation rules is outside of the scope of this paper. We will instead state some examples that should enable the reader to apply the method.

Definition 2.3 (Binary Trees) *The following symbolic equation defines the family of binary trees recursively and translates the definition into a functional equation of its generating function.*

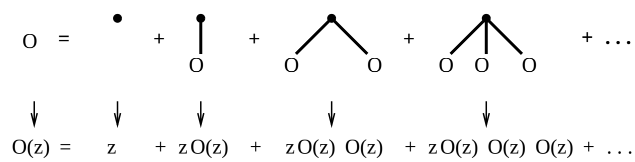


This can be interpreted as follows: A binary tree consists of a root to which up to two binary (sub-) trees can be attached. The left (right) son of a node v is also denoted by $LSON(v)$ $(RSON(v))$.

(By excluding the empty tree the definition above differs slightly from the definition in [Knu73].)
For the generating function $B(z)$ of ordered trees we obtain:

$$B(z) = z + zB(z) + zB(z) + zB(z)B(z) = z\ (1 + 2B(z) + B(z)^2) \tag{1}$$

Definition 2.4 (Ordered Tree) *The following symbolic equation defines the family of ordered trees recursively and translates the definition into a functional equation of its generating function.*



This can be interpreted as follows: An ordered tree consists of a root to which arbitrarily many ordered (sub-) trees can be attached.

For the generating function $O(z)$ of ordered trees we obtain:

$$O(z) = z + zO(z) + zO(z)O(z) + zO(z)O(z)O(z) + \ldots = z\ (1 - O(z))^{-1} \tag{2}$$

The functional equations for $B(z)$ and $O(z)$ are similar in structure. This motivates the introduction of a more general concept, the simply generated trees ([MM78]). Most of the frequently used tree families are simply generated (an important exception being unordered trees).

Definition 2.5 (Simply Generated Trees) *A family of trees F is said to be simply generated, if its generating function $F(z)$ satisfies an implicit equation: $F(z) = z\,\Theta(\,F(z)\,)$, where $\Theta(t)$ is an analytic function of the form:*

$$\Theta(t) = 1 + \sum_{n \geq 1} c_n t^n$$

defined in some disk $|t| < R < \infty$; with $c_n \geq 0$ and $c_n > 0$ for at least one $n \geq 1$.

By this definition the families O and B of ordered and binary trees are simply generated. The corresponding Θ-functions are:

$$\Theta_o(t) \;=\; 1 + t + t^2 + t^3 + \ldots = (1-t)^{-1}\text{and} \tag{3}$$
$$\Theta_b(t) \;=\; 1 + 2t + t^2\text{respectively.} \tag{4}$$

3 The Classical One-to-one Correspondence: $O \leftrightarrow B$

Solving the quadratic equations for $B(z)$ and $O(z)$ yields the closed forms:

$$B(z) = (1-2z-\sqrt{1-4z})/(2z) \quad (5)$$

$$O(z) = (1-\sqrt{1-4z})/2 \quad (6)$$

It is well known that the coefficents of $O(z)$ and $B(z)$ are just the Catalan numbers (cf. [Knu73]):

$$[z^n]\, B(z) = \frac{1}{n+1}\binom{2n}{n} = [z^{n+1}]\, O(z) \quad (n \geq 1)$$

Which leads us to the classical enumeration result:

Theorem 3.1 *There are as many ordered trees of size $n+1$ as there are binary trees of size n. More formally:* $|\{t \in O \mid |t| = n+1\}| = |\{t \in B \mid |t| = n\}|$.

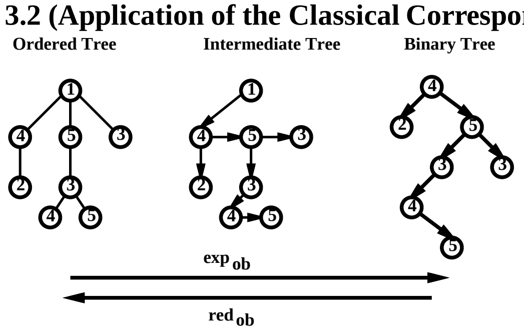
This can be written more compactly as an equation of generating functions .

$$O(z) = z\,(B(z)+1) \quad (7)$$

What is more intuitive is the structural correspondence between members of O and B. The following recipe shows how to bijectively transform ordered trees into the corresponding binary trees (c.f.[Knu73, pages 332ff]). We will already make intuitive use of labeled trees even though they are defined later.

Given a (labeled) ordered tree, remove all edges except for those connecting a father with its leftmost son. Then, add edges between brothers by connecting brother i with brother $i+1$ yielding an intermediate tree. Finally rotate this intermediate tree by $-\pi/4$ and remove the root.

Example 3.2 (Application of the Classical Correspondence)



When the root is removed in the last step of the transformation its labeling information is lost. To overcome this problem we require that the root be always labeled 1.

We are now concerned with formalizing the steps of the "recipe".

We shall call the transformation from ordered trees to binary trees $\exp_{ob}$, and the inverse transformation red_{ob} . The graphical operations are given in Figures 1 and 2 (m_i denotes the label of the corresponding node).

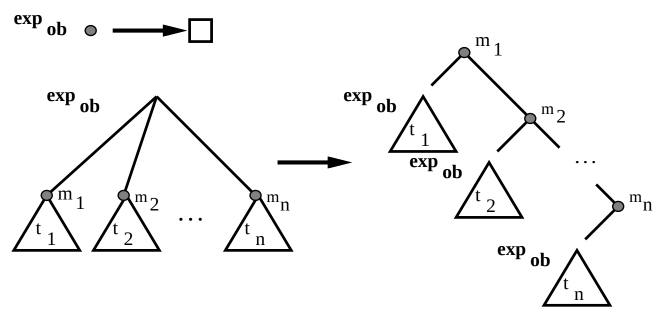


Figure 1: Transformation $\exp_{\mathrm{ob}}$

The definition of $\mathrm{red}_{\mathrm{ob}}$ is somewhat different from $\exp_{\mathrm{ob}}$ because we have to “generate” some extra information rather than throwing information away. Therefore, $\mathrm{red}_{\mathrm{ob}}$ has an upper index x, whose domain is the set of labels. This index will be the labeling of the root in the transformed tree. By the convention above we have $\mathrm{red}_{\mathrm{ob}}\,(t) := \mathrm{red}_{\mathrm{ob}}{}^{1}(t)$.

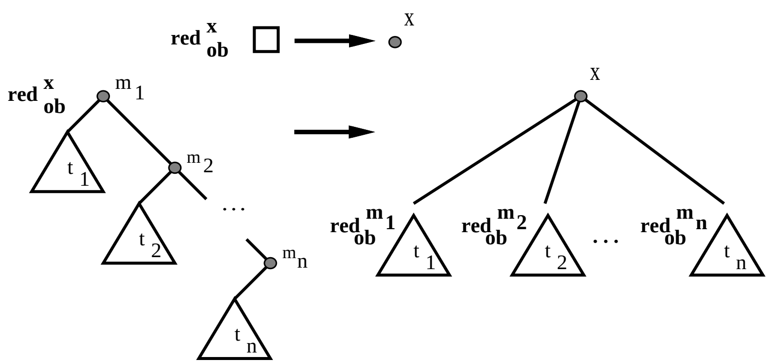


Figure 2: Transformation $\mathrm{red}_{\mathrm{ob}}{}^{x}$

Theorem 3.3 *The transformation* $\exp_{\mathrm{ob}}$ *is a bijection between the d-labeled ordered trees with* $n+1$ *nodes and a root labeled 1, and and the d-labeled binary trees with n nodes.*

Proof:

Obviously, $\exp_{\mathrm{ob}}$ reduces the number of nodes in a tree by one. An easy induction on the number of nodes shows that $\exp_{\mathrm{ob}}$ is injective.

Bijectivity follows from the fact that $\exp_{\mathrm{ob}}$ is a mapping between two sets of the same cardinality.

$\square$

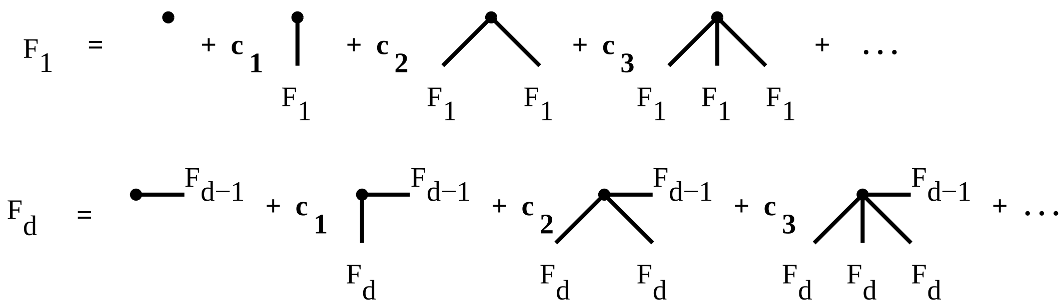
An analogous result can be derived for $\mathrm{red}_{\mathrm{ob}}$ in a similar manner.

4 Multi-dimensional Simply Generated Trees

4.1 Multi-dimensional Simply Generated Trees

In this section we describe multi-dimensional simply generated families of trees which are an extension of simply generated trees. Such trees were first investigated by Kemp in [Kem89, Kem93b, Kem95] who examined many different parameters and different probability models.

Definition 4.1 (Tree Family: F_d) *Given an arbitrary simply generated family of trees F with $\Theta(t) = 1 + \sum_{n\geq 1} c_n t^n$ the corresponding family of d-dimensional simply generated trees is recursively defined by the following symbolic equations, where $F_1 = F$:*



This can be interpreted as follows: A d-dimensional simply generated tree consists of a root to which a $(d-1)$-dimensional simply generated trees is attached. If $c_n \neq 0$, an additional n d-dimensional simply generated trees may be attached to this root as subtree.

In a d-dimensional tree, the nodes and edges that do not belong to the $(d-1)$-dimensional subtrees are said to be in layer 1. Similarly, the nodes and edges in the $(d-1)$-dimensional trees that do not belong to $(d-2)$-dimensional trees are said t be in layer 2, etc..

The tree restricted to layer 1 nodes is called the header tree.

To make the description of node numbers in the different layers more convenient, we introduce the following abbreviations:

Definition 4.2 *Let $t = (V,E) \in F_d$. We define:*

$$\begin{aligned} s_i(t) &:= |\{v \in V \mid v \text{ belongs to layer } i\}| \text{ ,and} \\ \vec{s}(t) &:= (s_1(t),\ldots,s_d(t)) \text{ with } \sum_{1\leq i\leq d} s_i(t) = |t| \end{aligned}$$

Generating Functions

The generating function $F_d(z)$ defined for d-dimensional trees:

$$[z^n]F_d(z) = |\{t \in F_d \mid s_d(t) = n\}|$$

satisfies the following implicit equations which can be derived from the above definitions by using the translation rules from [Fla88].

$$\begin{aligned} F_1(z) &= z\,\Theta(\,F_1(z)\,) && (8)\\ F_d(z) &= F_{d-1}(z)\,\Theta(\,F_d(z)\,) && (9) \end{aligned}$$

If $F(z)$ is the generating function of the simply generated family of trees used to define F_d then $F(z)$ must be a solution of (8) and thus:

$$\begin{aligned} F_1(z) &= F(z) && (10)\\ F_d(z) &= F(\,F_{d-1}(z)\,) && (11) \end{aligned}$$

Sometimes it is desirable to take account of the numbers $\vec{s}(t)$ of nodes in the different layers of a multi-dimensional tree t. This can be achieved by defining a multivariate generating function $\widetilde{F}_d(z_1,\ldots,z_d)$ with

$$[z_1^{n_1}\cdot\ldots\cdot z_d^{n_d}]\widetilde{F}_d(z_1,\ldots,z_d) \quad = \quad |\{t \in F_d \mid (n_1,\ldots,n_d) = \vec{s}(t)\}| \tag{12}$$

We have the functional equations:

$$\begin{aligned} \widetilde{F}_1(z_1) &= z_1\,\Theta(\,\widetilde{F}_1(z_1)\,) && (13)\\ \widetilde{F}_d(z_1,\ldots,z_d) &= z_1\widetilde{F}_{d-1}(z_2,\ldots,z_d)\,\Theta(\widetilde{F}_d(z_1,\ldots,z_d)) && (14) \end{aligned}$$

In a manner similar to the univariate case, we conclude:

$$\begin{aligned} \widetilde{F}_1(z_1) &= F(z_1) && (15)\\ \widetilde{F}_d(z_1,\ldots,z_d) &= F(\,z_1\,\widetilde{F}_{d-1}(z_2,\ldots,z_d)\,) && (16) \end{aligned}$$

The relationship of the univariate and multivariate generating functions is given by:

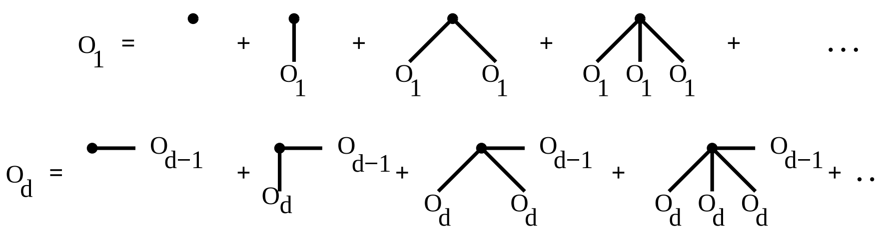
$$\widetilde{F}_d(1,\ldots,1,z) = F_d(z) \tag{17}$$

Equations (11) and (16) can also be obtained by applying the translation rule “substitution” from [Fla88], since a tree in F_d can be regarded as tree in F whose nodes have be substituted by a single node plus a tree in F_{d-1}.

4.2 Special Case: Multi-dimensional Ordered Trees

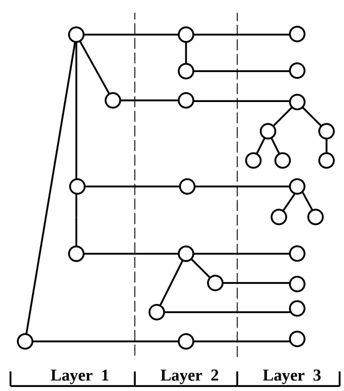
Multi-dimensional ordered trees are a special case of multi-dimensional simply generated trees, in which the Θ-function is given by: $\Theta_o(t) = \sum_{n\geq 0} t^n = (1-t)^{-1}$.

Definition 4.3 (Tree Family O_d) *The family of d-dimensional ordered trees is recursively defined by the following symbolic equations:*



This can be interpreted as follows: A d-dimensional ordered tree consists of a root to which at least one $(d-1)$-dimensional ordered trees is attached (as a tree in the next layer). In addition, arbitrarily many d-dimensional ordered trees can be attached to this root (as subtrees).

Example 4.4 (3-dimensional ordered tree with $\vec{s}(t) = (5, 8, 15)$)



Generating Functions

Substituting $\Theta_o(t)$ into the equations for the general case we obtain for the univariate and multivariate generating functions $O_d(z)$ and $\widetilde{O}_d(\vec{z})$:

$$O_1(z) = O(z) \tag{18}$$
$$O_d(z) = O(O_{d-1}(z)) \tag{19}$$

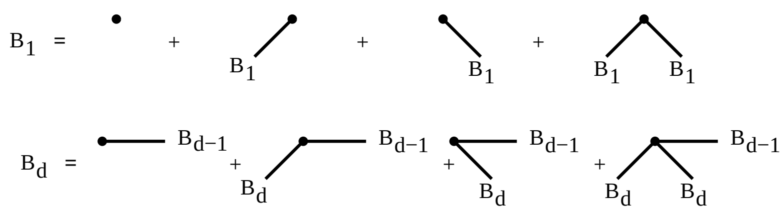
and

$$\widetilde{O}_1(z_1) = O(z_1) \tag{20}$$
$$\widetilde{O}_d(z_1, \ldots, z_d) = O(\, z_1 \widetilde{O}_{d-1}(z_2, \ldots, z_d)\,) \tag{21}$$

4.3 Special Case: Multi-dimensional Binary Trees

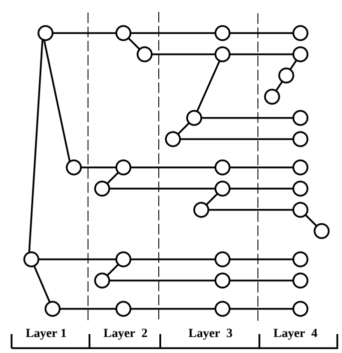
Multi-dimensional binary trees are a special case of multi-dimensional simply generated trees, in which the Θ-function is given by: $\Theta_b(t) = 1 + 2t + t^2$.

Definition 4.5 (Tree Family B_d) *The family of d-dimensional binary trees is recursively defined by the following symbolic equations:*



This can be interpreted as follows: A d-dimensional binary tree consists of a root to which at least one $(d-1)$-dimensional binary tree is attached (as a tree in the next layer). In addition, up to two d-dimensional binary trees can be attached to this root (as subtrees). If there is only one subtree we distinguish between a right and a left subtree.

Example 4.6 (4-dimensional binary tree with $\vec{s}(t) = (4, 7, 10, 13)$)



Generating Functions

Substituting $\Theta_b(t)$ in the equations for the general case we obtain for the univariate and multivariate generating functions $B_d(z)$ and $\widetilde{B}_d(\vec{z})$:

$$B_1(z) = B(z) \tag{22}$$

$$B_d(z) = B(B_{d-1}(z)) \tag{23}$$

and

$$\widetilde{B}_1(z_1) = B(z_1) \tag{24}$$

$$\widetilde{B}_d(z_1, \ldots, z_d) = B(z_1 \widetilde{B}_{d-1}(z_2, \ldots, z_d)) \tag{25}$$

5 Monotonically Labeled Simply Generated Trees

5.1 Monotonically Labeled Simply Generated Trees

In this section we describe monotonically d-labeled simply generated families of trees. Monotone labelings of tree families were investigated first in [PU83].

Definition 5.1 (Labeled Tree) *A tree $t = (V, E)$ together with a set of labels M and a (total) labeling function $f : V \to M$ is called a (totally) labeled tree.*

If $M = \{1, \ldots, d\}$ we call t d-labeled.

If f is partial, the tree is said to be partially labeled.

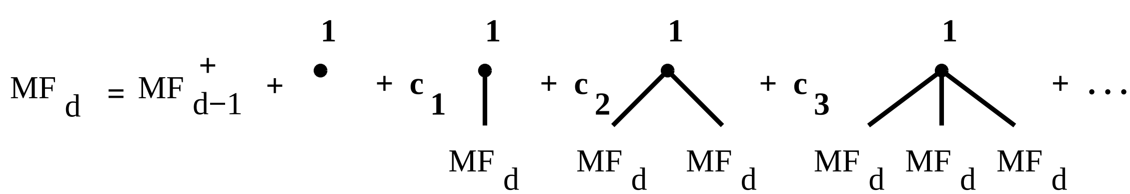
In order to describe the number of occurrences of certain labels more easily we define the following abbreviations for d-labeled trees.

$$\begin{aligned} \vec{m}(t) &:= (m_1(t), \ldots, m_d(t)) \quad \text{,with} \\ m_i(t) &:= |f^{-1}(i)| \end{aligned}$$

If f is partial, we define: $m_0(t) := |V - f^{-1}(M)|$. Note, that $|t| = \sum_{0 \le i \le d} m_i(t)$ always holds.

Definition 5.2 (Monotonically d-labeled Tree) *A d-labeled tree t with a labeling function f is called monotonically (increasing) d-labeled, if $f(v_j) \ge f(v_i)$ whenever 'v_j is a son of v_i'.*

Definition 5.3 (Tree Family MF_d) *The family of monotonically d-labeled simply generated trees is recursively defined by the following symbolic equations where $MO_0 = \emptyset$ and $\Theta(t) = 1 + \sum_{n \ge 1} c_n t^n$ stems from a simply generated tree family F:*



This can be interpreted as follows, corresponding to the first term and the remaining terms.

If a monotonically d-labeled tree has a root labeled differently from 1, then it corresponds to a $(d-1)$-labeled simply generated tree whose labels have each been increased by 1 (symbolized by MF^+_{d-1}). If it has a root labeled 1, n monotonically d-labeled simply generated trees can be attached (as subtrees) if $c_n \neq 0$.

On consequence of the distinction is important enough to put into a formula:

$$|MF_d| - |MF_{d-1}| = |\{t \in MF_d \mid \text{the root of } t \text{ is labeled } 1\}| \tag{26}$$

Subsequently, we will use the letter "E" to prefix subfamilies of trees where the root is labeled 1.

Generating Functions

The generating function $MF_d(z)$ with:

$$[z^n]MF_d(z) = |\{t \in MF_d \mid |t| = n\}|$$

satisfies the following implicit equations which can be derived by using the translation rules from [Fla88].

$$\begin{aligned} MF_0(z) &= 0 && (27)\\ MF_d(z) &= MF_{d-1}(z) + z\Theta(\,MF_d(z)\,) && (28) \end{aligned}$$

Sometimes one wants to count the numbers $\vec{m}(t)$ of occurrences of the different labels in a tree t. This can be achieved by defining the multivariate generating function $\widetilde{MF}_d(z_1,\ldots,z_d)$ with

$$[{z_1}^{n_1} \cdot \ldots \cdot {z_d}^{n_d}]\widetilde{MF}_d(z_1,\ldots,z_d) = |\{t \in MF_d \mid \vec{m}(t) = (n_1,\ldots,n_d)\}| \quad (29)$$

We have the functional equations:

$$\begin{aligned} \widetilde{MF}_0() &= 0 && (30)\\ \widetilde{MF}_d(z_1,\ldots,z_d) &= \widetilde{MF}_{d-1}(z_2,\ldots,z_d) + z_1\,\Theta(\,\widetilde{MF}_d(z_1,\ldots,z_d)\,) && (31) \end{aligned}$$

The relationship of the univariate and multivariate generating functions is given by:

$$\widetilde{MF}_d(z,\ldots,z) = MF_d(z)$$

For the subfamily of trees with roots labeled 1 we have by the remark after Definition 5.3:

$$\begin{aligned} EMF_d(z) &= \Theta(\,MF_d(z)\,) && (32)\\ \text{and} \qquad \widetilde{EMF}_d(z_1,\ldots,z_d) &= z_1\,\Theta(\,\widetilde{MF}_d(z_1,\ldots,z_d)\,) && (33) \end{aligned}$$

Moreover, since unlabeled trees are isomorphic to trees labeled with just one label:

$$F(z) = MF_1(z) = EMF_1(z) \quad (34)$$

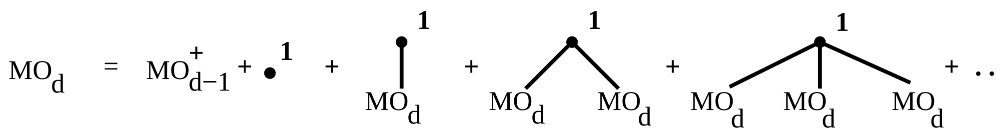
Little is known about the coefficients of these generating functions in general. In [PU83] several special families of trees were investigated including unordered trees (which are not simply generated). The investigations were limited to counting the number of trees.

For the special case of monotonically d-labeled ordered trees which is described below a new result was published in [Kem93a]. We rediscover this result later and also derive a similar one for the special case of monotonically d-labeled binary trees.

5.2 Special Case: Monotonically Labeled Ordered Trees

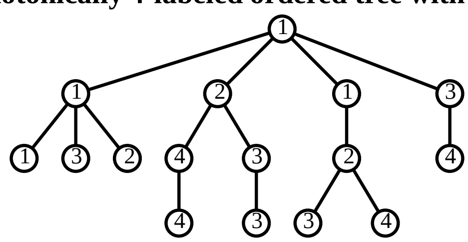
Monotonically labeled ordered trees are a special case of monotonically labeled simply generated trees in which the Θ-function is given by: $\Theta_o(t) = \sum_{n\geq 0} t^n = (1-t)^{-1}$.

Definition 5.4 (Tree family MO_d) *The family of monotonically d-labeled ordered trees is recursively defined by the following symbolic equation ($MO_0 = \emptyset$):*



This can be interpreted as follows: If a monotonically d-labeled ordered tree has a root labeled differently from 1, then it corresponds to a $(d-1)$-labeled ordered tree whose labels have each been increased by 1. (Symbolized by MO_{d-1}^{+}.) If it has a root labeled 1, arbitrarily many monotonically d-labeled ordered trees can be attached (as subtrees).

Example 5.5 (Monotonically 4-labeled ordered tree with $\vec{m}(t) = (4,3,5,4)$)



Generating Functions

Substituting $\Theta_o(t)$ in the equations for the general case we obtain for the univariate and multivariate generating functions $MO_d(z)$, $EMO_d(z)$ and $\widetilde{MO}_d(\vec{z})$,$\widetilde{EMO}_d(\vec{z})$:

$$\begin{aligned} MO_0(z) &= EMO_0(z) = 0 && (35)\\ MO_d(z) &= MO_{d-1}(z) + z\,(1 - MO_d(z))^{-1} && (36)\\ EMO_d(z) &= z\,(1 - MO_d(z))^{-1} && (37) \end{aligned}$$

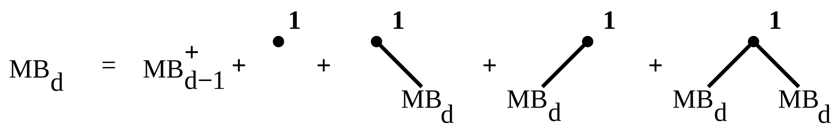
and

$$\begin{aligned} \widetilde{MO}_0() &= \widetilde{EMO}_0() = 0 && (38)\\ \widetilde{MO}_d(z_1,\ldots,z_d) &= \widetilde{MO}_{d-1}(z_2,\ldots,z_d) + z_1\,(1 - \widetilde{MO}_d(z_1,\ldots,z_d))^{-1} && (39)\\ \widetilde{EMO}_d(z_1,\ldots,z_d) &= z_1\,(1 - \widetilde{MO}_d(\vec{z}))^{-1} && (40) \end{aligned}$$

5.3 Special Case: Monotonically Labeled Binary Trees

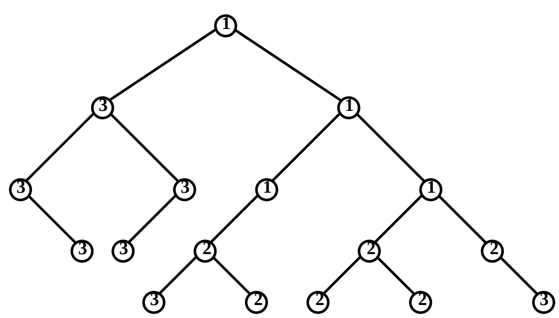
Monotonically labeled binary trees are a special case of monotonically labeled simply generated trees in which the Θ-function is given by : $\Theta_b(t) = 1 + 2t + t^2$.

Definition 5.6 (Tree Family MB_d) *The family of monotonically d-labeled binary trees is recursively defined by the following symbolic equations ($MB_0 = \emptyset$):*



This can be interpreted as follows: If a monotonically d-labeled binary tree has a root labeled differently from 1, then it corresponds to a $(d-1)$-labeled binary tree whose labels have each been increased by 1. (Symbolized by MB^+_{d-1}.) If it has a root labeled 1, up to 2 monotonically d-labeled binary trees can be attached (as subtrees). If there is only one subtree we distinguish between a right and a left subtree.

Example 5.7 (Monotonically 3-labeled binary tree with $\vec{m}(t) = (4, 6, 7)$)



Generating Functions

Substituting $\Theta_b(t)$ into the equations for the general case we obtain for the univariate and multivariate generating function $MB_d(z)$,$EMB_d(z)$, and $\widetilde{MB}_d(\vec{z})$,$\widetilde{EMB}_d(\vec{z})$:

$$
\begin{aligned}
MB_0(z) &= EMB_0(z) = 0 && (41)\\
MB_d(z) &= MB_{d-1}(z) + z\,(1 + MB_d(z))^2 && (42)\\
EMB_d(z) &= z\,(1 + MB_d(z))^2 && (43)
\end{aligned}
$$

and

$$
\begin{aligned}
\widetilde{MB}_0() &= \widetilde{EMB}_0() = 0 && (44)\\
\widetilde{MB}_d(z_1, \ldots, z_d) &= \widetilde{MB}_{d-1}(z_2, \ldots, z_d) + z_1(1 + \widetilde{MB}_d(z_1, \ldots, z_d))^2 && (45)\\
\widetilde{EMB}_d(z_1, \ldots, z_d) &= z_1(1 + \widetilde{MB}_d(z_1, \ldots, z_d))^2 && (46)
\end{aligned}
$$

6 A One-to-one Correspondence: $EMO_d^2 \leftrightarrow B_d$

6.1 Identity 1

Theorem 6.1

$$z^{-1}EMO_d(z)^2 \quad = \quad B(z^{-1}EMO_{d-1}(z)^2)$$

Proof: Let $Y_d := z^{-1}EMO_d(z)^2$, then $Y_{d-1} = (2 + Y_d + Y_d^{-1})^{-1}$ by (36) and (37). The identity follows immediately after substituting Y_{d-1} into (5). □

Corollary 6.2 (Identity 1,[Kem93a])

$$z^{-1}EMO_d(z)^2 \quad = \quad B_d(z)$$

Proof: The proof is an easy induction on d using $z^{-1}EMO_1(z)^2 = z^{-1}(O(z))^2 = B(z) = B_1(z)$ for the basis and (23) and Theorem 6.1 for the induction step. □

Passing to the coefficients of the generating functions we obtain the following enumeration result.

Corollary 6.3 *There are as many tuples of d-labeled ordered trees with the root labeled 1 and a total of $n+1$ nodes as there are d-dimensional binary trees with n nodes in the last (d-th) layer. More formally:* $|\{(t_1,t_2) \in EMO_d \times EMO_d \mid |t_1|+|t_2| = n+1\}| \; = \; |\{t \in B_d \mid s_d(t) = n\}|$.

The rest of this section is structured as follows: First we shall construct auxiliary tree families whose generating functions satisfy the left and right hand side of Theorem 6.1 (EX_d and $B[EX_d]$). Then we will give a structural correspondence between members of EX_d and $B[EX_d]$ in form of the transformations $\exp_{\text{ex}-\text{bex}}$ and $\text{red}_{\text{ex}-\text{bex}}$.

By iterating these transformations and employing two others ($\exp_{\text{emo}-\text{ex}}$ and $\text{red}_{\text{emo}-\text{ex}}$) we finally obtain a transformation associated with Corollary 6.2.

6.2 Auxiliary Tree Families

Definition 6.4 (Tree Families X_d and EX_d) *The tree family X_d and its subfamily EX_d are defined as follows:* $X_d = \{ \exp_{\text{ob}}(t) \mid t \in EMO_d\}$ *and* $EX_d = \{t \in X_d \mid t \text{ has a root labeled } 1\}$.

This definition is not very handy. Theorem 6.7 will provide a better characterization of the tree family.

Definition 6.5 (Rightmost Branch, RA) *Let $t = (V,E)$ be a binary tree and $v \in V$. $RA_v \subset V$ is the smallest set containing v such that $w \in RA_v \to RSON(w) \in RA_v$ holds, i.e. RA_v are all the nodes on a rightmost branch starting with node v.*

Definition 6.6 (Property M) *Let $t = (V,E)$ be a labeled binary tree with labeling function f. A node $v \in V$ is said to satisfy Property M, if it either has no left son or for all $w \in RA_{LSON(v)}$ $f(w) \geq f(v)$ is true. This definition is clarified in Figure 3.*

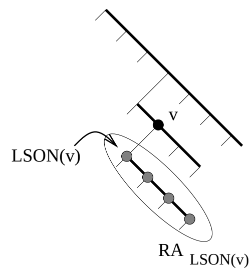


Figure 3: Definition 6.6

Theorem 6.7 (Alternative Characterization of X_d) *X_d consists of all those d-labeled binary tree whose nodes satisfy property M.*

Proof: Let g be an injective labeling function of an ordered tree $t = (V,E)$. We are now able to distinguish the individual nodes of the tree. Let $t' = (V',E') = \exp_{\text{ob}}(t)$. We say two nodes from t and t' correspond if they have the same label. From the definition of $\exp_{\text{ob}}$ it should be clear that if $v' \in V'$ corresponds to $v \in V$, then each of the nodes in $RA_{LSON(v')}$ must correspond to exactly one nodes in the set S, which is contains all sons of v.

Now all the nodes in trees from X_d satisfy property M because the original trees were monotonically labeled.

Conversely, if all nodes in a labeled binary tree t satisfy property M, $\text{red}_{\text{ob}}(t)$ generates a tree $t' \in EMO_d$ with $\exp_{\text{ob}}(t') = t$ implying $t \in X_d$. □

Corollary 6.8 (Subtrees) *If t is in X_d then any subtree of t is also in X_d.*

Proof: Because all the nodes of t satisfy property M, this is also true for the nodes in any subtree. □

Corollary 6.9 (Concatenation of Trees) *If $t_l, t_r \in X_d$ then the tree t constructed by attaching t_l, t_r as the left and right subtree to a new root node with label 1 is also in X_d (even in EX_d):*

Proof: Because $t_l, t_r \in X_d$ all nodes of t except for the root satisfy property M. But since 1 is smallest label the root satisfies property M, too. □

Corollary 6.10 (Left Subtree) *Let $t \in X_d$ with root $r(t)$ and let $t_l = (V_l, E_l)$ be the left subtree of t then $f(v) \geq f(r)$ for all $v \in V_l$.*

Proof: By induction on $|V_l|$. For $|V_l| = 0$ the claim is trivially true. For $|V_l| > 0$ the subtree t_l must be of the form depicted in Figure 4. Because of Theorem 6.7 we have: $\forall_{1 \leq i \leq n} f(v_i) \geq f(r)$. Additionally, the induction hypothesis applies to all v_i and their left subtrees. □

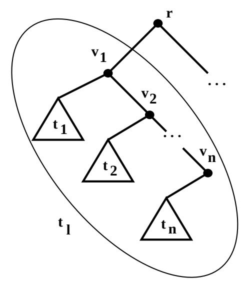


Figure 4: Corollary 6.10

Generating Function

For the generating functions $X_d(z)$ and $\widetilde{X}_d(\vec{z})$ defined by:

$$[z^n]X_d(z) = |\{t \in X_d \mid |t| = n\}|$$

$$[z_1^{n_1} \cdot \ldots \cdot z_d^{n_d}]\widetilde{X}_d(z_1,\ldots,z_d) \quad = \quad |\{t \in X_d \mid \vec{m}(t) = (n_1,\ldots,n_d)\}|$$

We have by definition and the fact that $\exp_{\mathrm{ob}}$ is a bijection.

$$\begin{aligned} X_d(z) &= EMO_d(z)/z && (47)\\ \widetilde{X}_d(z_1,\ldots,z_d) &= \widetilde{EMO}_d(z_1,\ldots,z_d)/z_1 && (48)\\ \widetilde{X}_d(z,\ldots,z) &= X_d(z) && (49) \end{aligned}$$

For the generating functions $EX_d(z)$ and $\widetilde{EX}_d(\vec{z})$ defined by:

$$[z^n]EX_d(z) = |\{t \in EX_d \mid |t| = n\}|$$

$$[z_1^{n_1} \cdot \ldots \cdot z_d^{n_d}]\widetilde{EX}_d(z_1,\ldots,z_d) \quad = \quad |\{t \in EX_d \mid \vec{m}(t) = (n_1,\ldots,n_d)\}|$$

We have:

$$\begin{aligned} EX_d(z) &= zX_d(z)^2 = EMO_d(z)^2/z && (50)\\ \widetilde{EX}_d(z_1,\ldots,z_d) &= \widetilde{EMO}_d(z_1,\ldots,z_d)^2/z_1 && (51)\\ \widetilde{EX}_d(z,\ldots,z) &= EX_d(z) && (52) \end{aligned}$$

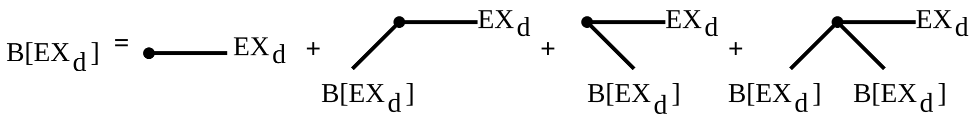
This follows from Corollaries 6.9 and 6.8, which establish a bijection between tuples of trees from X_d and trees from EX_d. Using Corollary 6.2 we get:

$$EX_d(z) = B_d(z) \tag{53}$$

Passing to the coefficients of the generating functions we obtain the following enumeration result.

Theorem 6.11 *There are as many trees in EX_d with n nodes as there are trees in B_d with n nodes in the last layer. More formally:* $|\{t \in EX_d \mid |t| = n\}| \ = \ |\{t \in B_d \mid s_d(t) = n\}|$.

Definition 6.12 (Tree Family $B[EX_d]$**)** *The tree family* $B[EX_d]$ *is recursively defined by the following symbolic equation:*



This can be interpreted as follows: A tree $t \in B[EX_d]$ is a binary tree where trees from EX_d have been attached to every node.

Because every node of t is associated with a tree from EX_d, we can think of the nodes as being substituted by the trees.

t is a partially labeled tree.

Generating Functions

For the generating functions $B[EX_d](z)$ and $B[\widetilde{EX_d}](y,\vec{z})$ defined by:

$$[z^n]\, B[EX_d](z) = |\{t \in B[EX_d]|\sum_{1\le i\le d} m_i(t) = n\}|$$

$$[y^{n_0} z_1^{n_1} \cdot \ldots \cdot z_d^{n_d}] B[\widetilde{EX_d}](y, z_1, \ldots, z_d) = |\{t \in B[EX_d] \mid m_0(t) = n_0 \wedge \vec{m}(t) = (n_1, \ldots, n_d)\}|$$

we have (using the translation rule “substitution” from [Fla88]):

$$B[EX_d](z) \quad = \quad B(\,EX_d(z)\,) \tag{54}$$

$$B[EX_d](y,\vec{z}) \quad = \quad B(\,y\widetilde{EX}_d(\vec{z})\,) \tag{55}$$

$$B[EX_d](1,z,\ldots,z) \quad = \quad B[EX_d](z) \tag{56}$$

Theorem 6.1 implies:

$$B[EX_d](z) = EX_{d+1}(z) \tag{57}$$

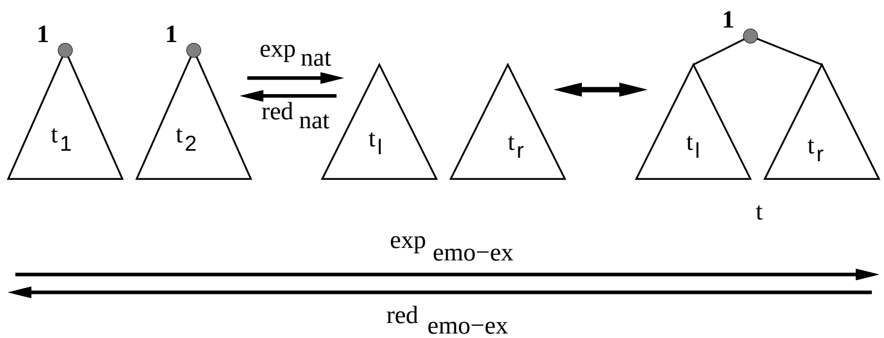
Passing to the coefficients of the generating functions we obtain the following enumeration result.

Theorem 6.13 *There are as many trees in* $B[EX_d]$ *with a total of* n *labeled nodes as there are trees in* EX_{d+1} *with* n *nodes. More formally:* $|\{t \in B[EX_{d+1}] \mid \sum_{1\le i\le d} m_i(t) = n\}| \; = \; |\{t \in EX_d \mid |t| = n\}|$.

6.3 Auxiliary Transformations

Definition 6.14 (Transformations $\mathrm{red}_{\mathrm{emo-ex}}$ **and** $\mathrm{exp}_{\mathrm{emo-ex}}$ **)**

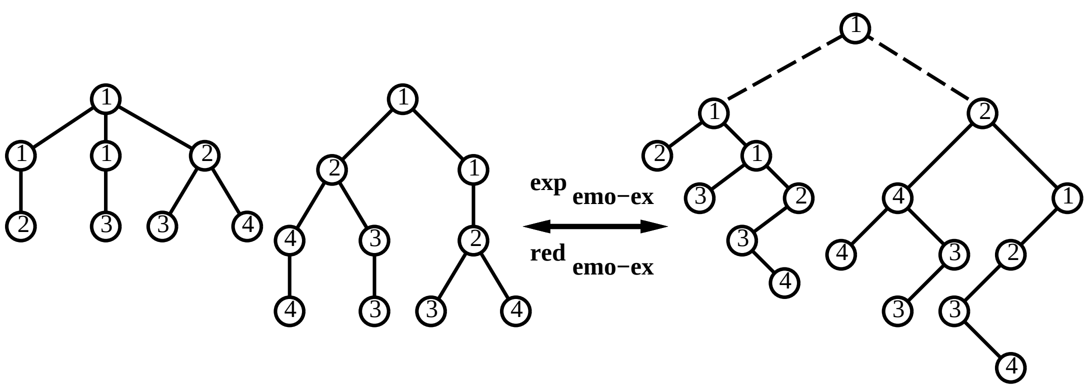
The transformations $\mathrm{red}_{\mathrm{emo-ex}} : EX_d \mapsto EMO_d^2$ *and* $\mathrm{exp}_{\mathrm{emo-ex}} : EMO_d^2 \mapsto EX_d$ *are defined below:*



The transformation $\mathrm{exp}_{\mathrm{emo-ex}}$ *consists of 2 steps. Step 1 transforms* $t_1, t_2 \in EMO_d$ *into* $t_l, t_r \in X_d$ *by applying transformation* $\mathrm{exp}_{\mathrm{ob}}$ *to each tree. The next step merges these 2 trees into a single tree* $t \in EX_d$*, by adding a root node that is labeled 1.* $\mathrm{red}_{\mathrm{emo-ex}}$ *is the inverse transformation.* $\mathrm{exp}_{\mathrm{emo-ex}}$ *is bijective because* $\mathrm{exp}_{\mathrm{ob}}$ *is bijective.*

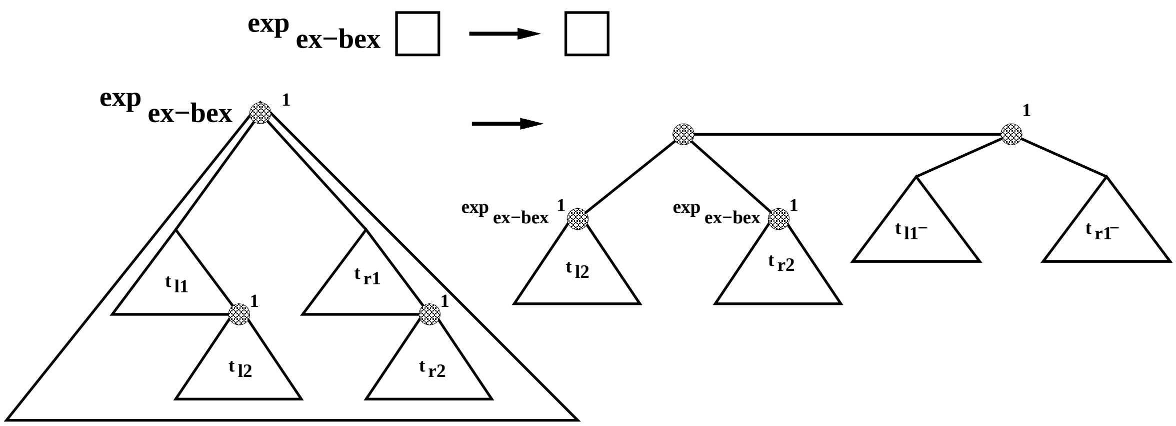
The transformations we have just presented correspond to the enumeration results of (50).

Example 6.15 (Application example of the transformations $\mathrm{red}_{\mathrm{emo-ex}}$ **and** $\mathrm{exp}_{\mathrm{emo-ex}}$ **)**



Definition 6.16 (Transformation $\exp_{\text{ex}-\text{bex}}$)

The transformation $\exp_{\text{ex}-\text{bex}} : EX_d \mapsto B[EX_{d-1}]$ is defined below:



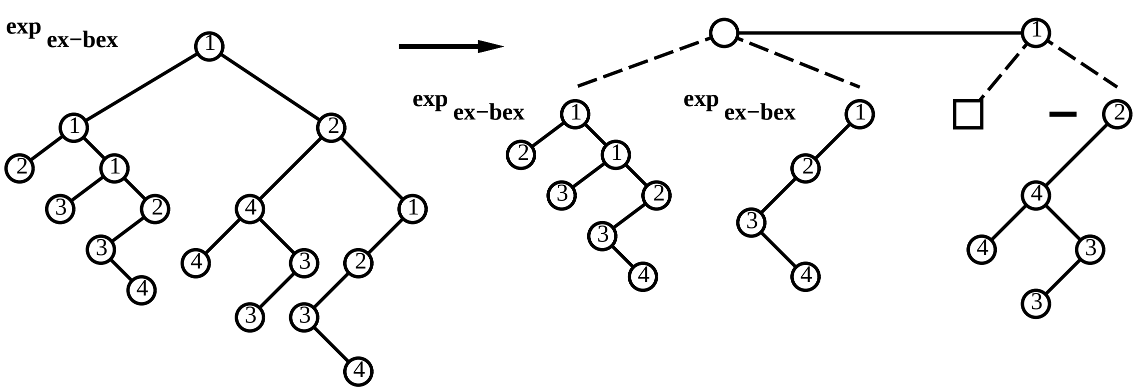
The roots of the trees t_{l2} and t_{r2} are chosen as follows: they have to be in $RA_{LSON(r)}$ and $RA_{RSON(r)}$ respectively; they must be labeled with 1 and they must be as close as possible to the root of the whole tree.

If there is no node labeled 1 in $RA_{r(t_l)}$ ($RA_{r(t_r)}$) then t_{l2} (t_{r2}) is the empty tree.

The trees $t_{l1}-$ and $t_{r1}-$ are obtained from t_{l1} and t_{r1} by decreasing all labels by 1.

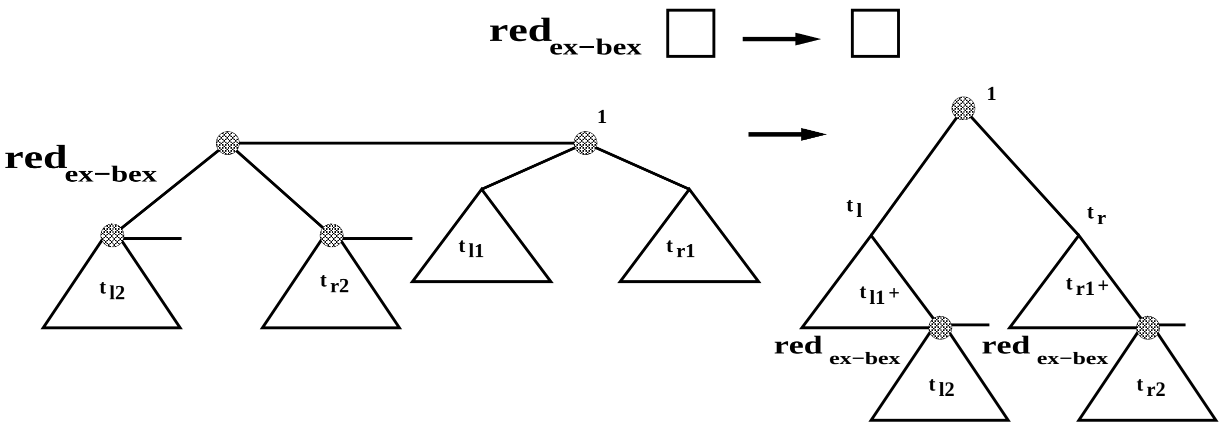
We postpone the proof of the transformations being well-defined and bijective until the next subsection.

Example 6.17 (Application example of transformation $\exp_{\text{ex}-\text{bex}}$)



Definition 6.18 (Transformation $\mathrm{red}_{\mathrm{ex-bex}}$)

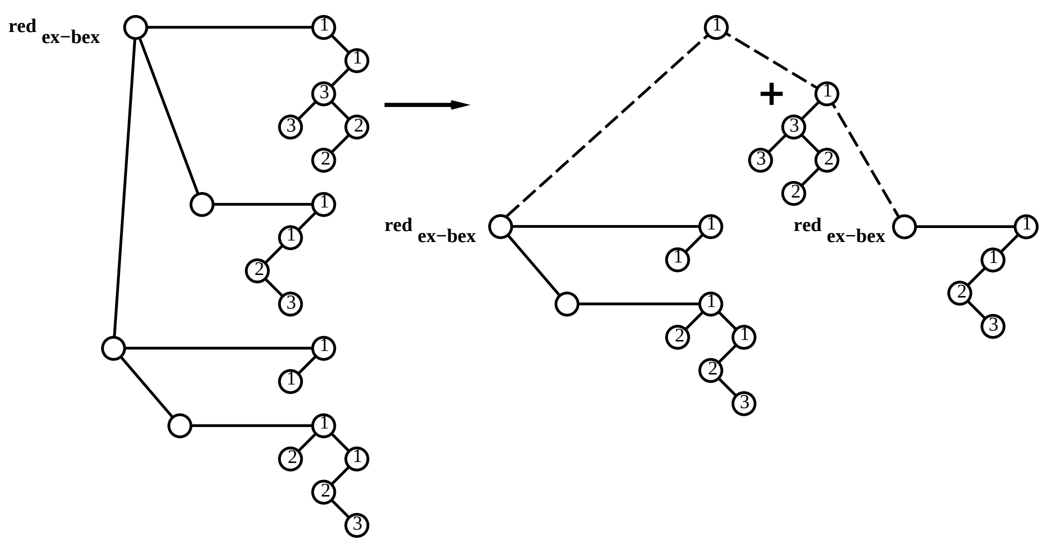
The transformation $\mathrm{red}_{\mathrm{ex-bex}}$ *:* $B[EX_{d-1}] \mapsto EX_d$ *is defined below:*



The trees $t_{l1}+$ and $t_{r1}+$ are obtained from t_{l1} and t_{r1} by increasing all labels by 1.

The trees $\mathrm{red}_{\mathrm{ex-bex}}$ (t_{l1}) *and* $\mathrm{red}_{\mathrm{ex-bex}}$ (t_{r1}) *are attached to the rightmost nodes in $t_{l1}+$ and $t_{r1}+$ as right subtrees.*

Example 6.19 (Application example of transformation $\mathrm{red}_{\mathrm{ex-bex}}$)



Theorem 6.20 *The transformation* $\exp_{\text{ex-bex}}$ *is a bijection between the tree families* EX_d *and* $B[EX_{d-1}]$.

Proof:

$\exp_{\text{ex-bex}}$ *is well-defined:*

We have to show that the tree on the left of Definition 6.16 has a unique representation as an argument of $\exp_{\text{ex-bex}}$. We also have to show that t_{l2} and t_{r2} are in the domain of $\exp_{\text{ex-bex}}$ (i.e. $EX_d \times EX_d$) and that the resulting tree is in the range of $\exp_{\text{ex-bex}}$ (i.e. $B[EX_{d-1}]$).

Without loss of generality we only consider the left subtree which consists of t_l1 and t_l2. As depicted in Figure 5, let v_i be the root of t_{l2}. Hence, all v_j with $j < i$ have labels greater than 1.

Because of Corollary 6.10 there are no nodes with label 1 in the left subtree of any such v_j.

Thus, $t_{l1}-$ is well-defined and similarly $t_{r1}-$. If no such node v_i exists, no node in the entire tree t_l is labeled with 1.

The trees t_{l2}, t_{r2} are subtrees of a tree in X_d. Because of Corollary 6.8 they are themselves in X_d, and since their roots are labeled with 1 they are also in EX_d.

Now the tree consisting of $t_{l1}-, t_{r1}-$ is in EX_{d-1} by Corollary 6.9.

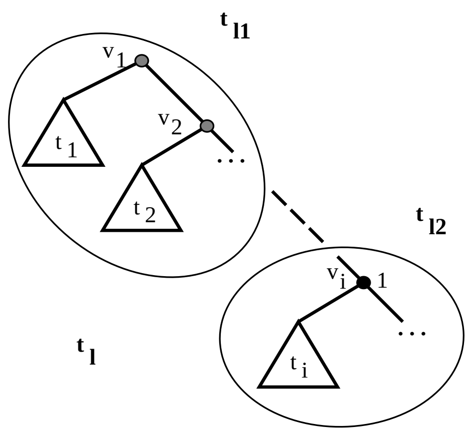


Figure 5: Left subtree of a tree in EX_d

Remark on the node numbers:

Let t be tree in EX_d and $\vec{m}(t) = (m_1, \ldots, m_d)$. Obviously, $\vec{m}(\exp_{\text{ex-bex}}(t)) = (m_1 + m_2, m_3, \ldots, m_d)$. That is, the trees from EX_{d-1} generated by the transformation of t consist of as many nodes as the original tree. Moreover, the (unlabeled) binary header tree to which the EX_{d-1} trees are attached consists of m_1 nodes.

$\exp_{\text{ex-bex}}$ *is injective:*

Let $t^1 \neq t^2$ be two trees in EX_d. We will show that $\exp_{\text{ex-bex}}(t^1) \neq \exp_{\text{ex-bex}}(t^2)$.

Because of the remark on the node numbers we can assume: $|t^1| = |t^2|$.

The proof is an induction on the number of nodes $n = |t^1| = |t^2|$.

Basis $n = 1$: Since there is just one tree with one node which in addition must be labeled with 1, nothing remains to be shown.

Now, suppose that the induction hypothesis is valid for trees with less than n nodes and let $t^1 \neq t^2$ be two trees with n nodes. These trees must consist of subtrees as depicted in Definition 6.16.

We distinguish four different cases according to the subtree(s) in which t^1 and t^2 differ:

1. $t^1_{l2} \neq t^2_{l2}$:
   Using the induction hypothesis it follows that $\exp_{\text{ex}-\text{bex}}(t^1_{l2}) \neq \exp_{\text{ex}-\text{bex}}(t^2_{l2})$ and hence $\exp_{\text{ex}-\text{bex}}(t^1) \neq \exp_{\text{ex}-\text{bex}}(t^2)$
2. $t^1_{r2} \neq t^2_{r2}$:
   Similar to 1.
3. $t^1_{l1} \neq t^2_{l1}$:
   We have $t^1_{l1}- \neq t^2_{l1}-$ and therefore $\exp_{\text{ex}-\text{bex}}(t^1) \neq \exp_{\text{ex}-\text{bex}}(t^2)$
4. $t^1_{r1} \neq t^2_{r1}$:
   Similar to 3.

$\exp_{\text{ex}-\text{bex}}$ *is bijective:*

This follows immediately from the fact that $\exp_{\text{ex}-\text{bex}}$ is a mapping between two sets of the same cardinality (Theorem 6.13).

$\square$

By the above remark on the nodes numbers we get the following refinement of Theorem 6.13:

Corollary 6.21 (Refined Theorem 6.13)

$$\begin{aligned}
&|\{t \in EX_d \mid \vec{m}(t) = (n_1, \ldots, n_d)\}| \quad = \\
&|\{t \in B[EX_{d-1}] \mid \vec{m}(t) = (n_1 + n_2, n_3, \ldots, n_d) \wedge m_0(t) = n_1\}|
\end{aligned}$$

This can be written more compactly as an equation of generating functions:

$$\widetilde{EX}_d(z_1, \ldots, z_d) = B[\widetilde{EX}_{d-1}](z_1/z_2, z_2, z_3, \ldots, z_d)$$

An analogous result can be derived for $\text{red}_{\text{ex}-\text{bex}}$ in a similar manner.

6.4 A One-to-one Correspondence: $B_d \leftrightarrow EMO_d \times EMO_d$

Let t^1 be a tree in EX_d with $\vec{m}(t^1) = (m_1, \ldots, m_d)$. Applying the transformation $\mathrm{exp}_{\mathrm{ex-bex}}$ to t^1 a new tree $t^2 \in B[EX_{d-1}]$ with $\vec{m}(t^2) = (m_1 + m_2, m_3, \ldots, m_d)$ is generated.

t^2 includes m_1 trees from EX_{d-1}.

Using the transformation again these m_1 trees can themselves be replaced by trees from $B[EX_{d-2}]$ yielding a tree t^3 with $\vec{m}(t^3) = (m_1 + m_2 + m_3, m_4, \ldots, m_d)$.

Iterating this $d-1$ times we get $\sum_{1 \le i \le d-1} m_i$ trees from EX_1 that consist of total of $\sum_{1 \le i \le d} m_i$ nodes all of which are labeled 1. If we remove the labels we obtain trees from B_1. Altogether, we have generated a tree in B_d.

The iterated transformation we have just presented corresponds to the enumeration result of Theorem 6.11. From the effects of the transformations on node numbers we are now able to refine this theorem.

Theorem 6.22 (Refined Theorem 6.11)

$$|\{t \in B_d \mid \vec{s}(t) = (n_1, \ldots, n_d)\}| \quad = \quad |\{t \in EX_d \mid \vec{m}(t) = (n_1, n_2 - n_1, \ldots, n_d - n_{d-1})\}|$$

This can be written more compactly as an equation of generating functions:

$$\widetilde{EX}_d(z_1, \ldots, z_d) = \widetilde{B}_d(z_1/z_2, \ldots, z_{d-1}/z_d, z_d)$$

An example of the iterated application of the transformation is given in Example 6.24.

Combine the iterated transformation with the transformations $\mathrm{exp}_{\mathrm{emo-ex}}$ and $\mathrm{red}_{\mathrm{emo-ex}}$ yields the desired one-to-one correspondence for Identity 1, namely $\mathrm{exp}_{\mathrm{emo-b}} : EMO_d \times EMO_d \to B_d$ and $\mathrm{red}_{\mathrm{emo-b}} : B_d \to EMO_d \times EMO_d$ which can be described as:

$$\mathrm{exp}_{\mathrm{emo-b}} \quad = \quad \mathrm{unmark1} \circ \mathrm{exp}_{\mathrm{ex-bex}}{}^{d-1} \circ \mathrm{exp}_{\mathrm{emo-ex}} \tag{58}$$

$$\mathrm{red}_{\mathrm{emo-b}} \quad = \quad \mathrm{red}_{\mathrm{emo-ex}} \circ \mathrm{red}_{\mathrm{ex-bex}}{}^{d-1} \circ \mathrm{mark1} \tag{59}$$

unmark1 stands for the removal of labels from the EX_1 trees and mark1 for labeling all nodes of B_1 with 1. For $d = 1$ this essentially simplifies to the classical correspondence.

Proceeding similarly as above we get a refinement of Corollary 6.2:

Theorem 6.23 (Refined Identity 1)

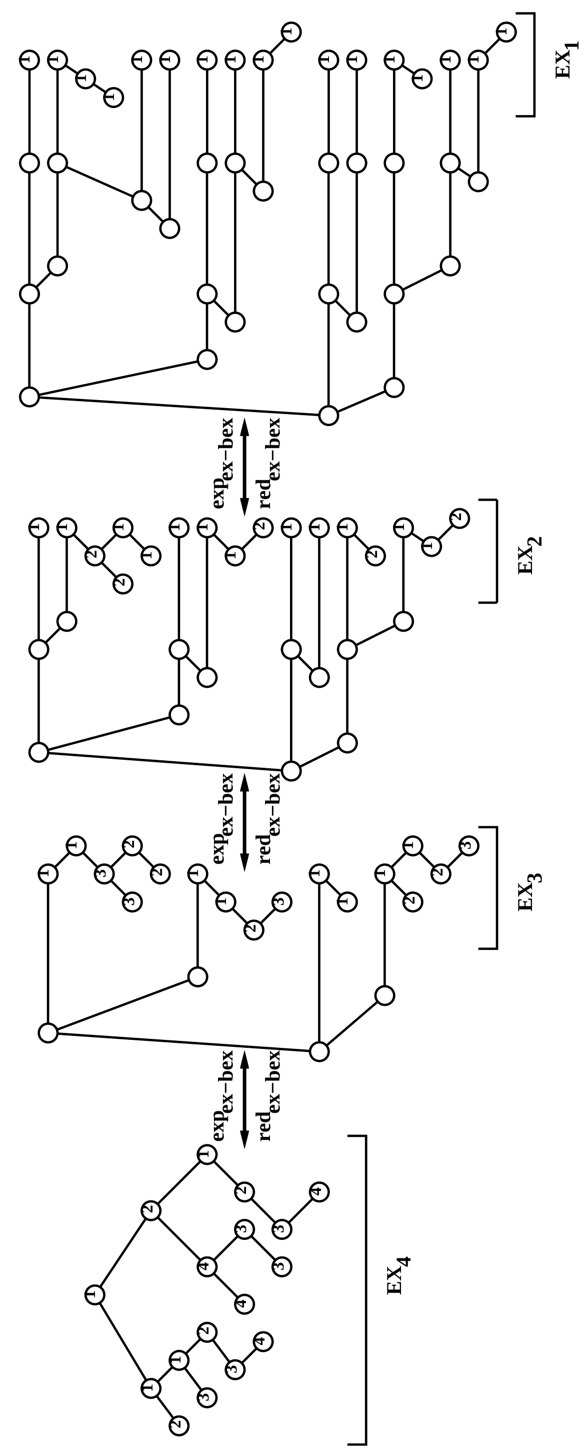
$$\begin{aligned} &|\{t \in B_d \mid \vec{s}(t) = (n_1, \ldots, n_2)\}| \quad = \\ &|\{(t_1, t_2) \in EMO_d \times EMO_d \mid \vec{m}(t_1) + \vec{m}(t_2) = (n_1 + 1, n_2 - n_1, \ldots, n_d - n_{d-1})\}| \end{aligned}$$

This can be written more compactly as an equation of generating functions:

$$\widetilde{B}_d(z_1/z_2, \ldots, z_{d-1}/z_d, z_d) = z_1^{-1} \widetilde{EMO}_d(z_1, \ldots, z_d)^2$$

Example 6.15 combined with Example 6.24 give an example for $\mathrm{exp}_{\mathrm{emo-b}}$ and $\mathrm{red}_{\mathrm{emo-b}}$. (Note, that the labels in the last layer have not been removed in order to keep the figure at a reasonable size.)

Example 6.24 (Example of the iterated application of $\mathrm{exp}_{\mathrm{ex-bex}}$ **and** $\mathrm{red}_{\mathrm{ex-bex}}$ **)**



7 A One-to-one Correspondence: $O_d \leftrightarrow MB_d$

7.1 Identity 2

Theorem 7.1

$$z(1+MB_d(z)) \quad = \quad O(\,z(1+MB_{d-1}(z))\,)$$

Proof: Let $Z_d := z(1+MB_d(z))$, then $Z_{d-1} = Z_d + Z_d^2$ by (42). The identity follows immediately after substituting Z_{d-1} into (6). □

This identity regarding monotonically labeled binary trees is a new result.

Corollary 7.2 (Identity 2)

$$z(1+MB_d(z)) \quad = \quad O_d(z)$$

Proof: The proof is an induction on d using $z(1+MB_1(z)) = z(1+B(z)) = O(z) = O_1(z)$ for the basis and (19) and Theorem 7.1 for the induction step. □

Passing to the coefficients of the generating functions we obtain the following enumeration result:

Corollary 7.3 *There are as many monotonically d-labeled binary trees with n nodes as there are d-dimensional ordered trees with $n+1$ nodes in the last (d-th) layer. More formally:* $|\{t \in MB_d \mid |t| = n\}| \;=\; |\{t \in O_d \mid s_d(t) = n+1\}|$.

The rest of this section is structured as follows: First we shall construct auxiliary tree families whose generating functions satisfy the left and right hand side of Theorem 7.1 ($MB_d^\square$ and $O[MB_d^\square]$). Then we will give a structural correspondence between members of $MB_d^\square$ and $O[MB_d^\square]$ in form of the transformations $\exp_{\mathrm{omb-mb}}$ and $\mathrm{red}_{\mathrm{omb-mb}}$.
By iterating these transformations we finally obtain a transformation associated with Corollary 7.2.

7.2 Auxiliary Tree Families

Definition 7.4 (Tree Family $MB_d^\square$) *The tree family $MB_d^\square$ is obtained from MB_d by adding the empty tree $\square$, i.e.* $MB_d^\square = MB_d \cup \{\square\}$.

Generating Functions

This implies for the generating functions:

$$MB_d^\square(z) \quad = \quad MB_d(z)+1 \tag{60}$$
$$\widetilde{MB}_d^\square(\vec{z}) \quad = \quad \widetilde{MB}_d(\vec{z})+1 \tag{61}$$

With Corollary 7.2 we have:

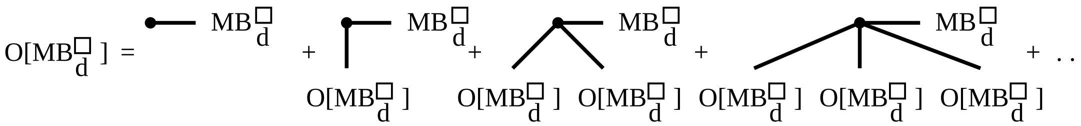
$$zMB_d^\square(z) = O_d(z) \tag{62}$$

Passing to the coefficients of the generating functions we obtain the following enumeration result.

Theorem 7.5 *There are as many tree in $MB_d^\square$ with n nodes, as there are trees in O_d with $n+1$ nodes in the last layer. More formally:* $|\{t \in MB_d^\square \mid |t| = n\}| \;=\; |\{t \in O_d \mid s_d(t) = n+1\}|$.

Note, that this enumeration result is also valid for MB_d instead of $MB_d^\square$ — except for the pathological case $n = 0$.

Definition 7.6 (Tree Family $O[MB_d^{\square}]$) *The tree family $O[MB_d^{\square}]$ is defined as follows:*



This can be interpreted as follows: A tree $t \in O[MB_d^{\square}]$ is an ordered tree. But an additional tree from $MB_d^{\square}$ is attached to every node of t.

Because every node of t is associated with a tree from $MB_d^{\square}$, we can think of the nodes as being substituted by the trees.

t is a partially labeled tree.

Generating Functions

For the generating functions $O[MB_d^{\square}](z)$ and $O[\widetilde{MB}_d^{\square}](y,\vec{z})$ defined by:

$$[z^n]O[MB_d^{\square}](z) = |\{t \in B[EX_d] \mid |t| = n\}|$$

$$\begin{aligned}&[y^{n_0} z_1^{n_1} \cdot \ldots \cdot z_d^{n_d}]O[\widetilde{MB}_d^{\square}](y, z_1, \ldots, z_d) = \\ &|\{t \in O[MB_d^{\square}] \mid m_0(t) = n_0 \wedge \vec{m}(t) = (n_1, \ldots, n_d)\}|\end{aligned}$$

we get (using the construct “substitution” from [Fla88]):

$$O[MB_d^{\square}](z) = O(\, z MB_d^{\square}(z)\,) = O(\, z\,(1 + MB_d(z))\,) \tag{63}$$

$$O[\widetilde{MB}_d^{\square}](y,\vec{z}) = O(\, y \widetilde{MB}_d^{\square}(\vec{z})\,) = O(\, y\,(1 + MB_d(\vec{z}))\,) \tag{64}$$

$$O[\widetilde{MB}_d^{\square}](z, z, \ldots, z) = O[MB_d^{\square}](z) \tag{65}$$

Theorem 7.1 implies:

$$O[MB_d^{\square}](z) = z MB_{d+1}^{\square}(z) \tag{66}$$

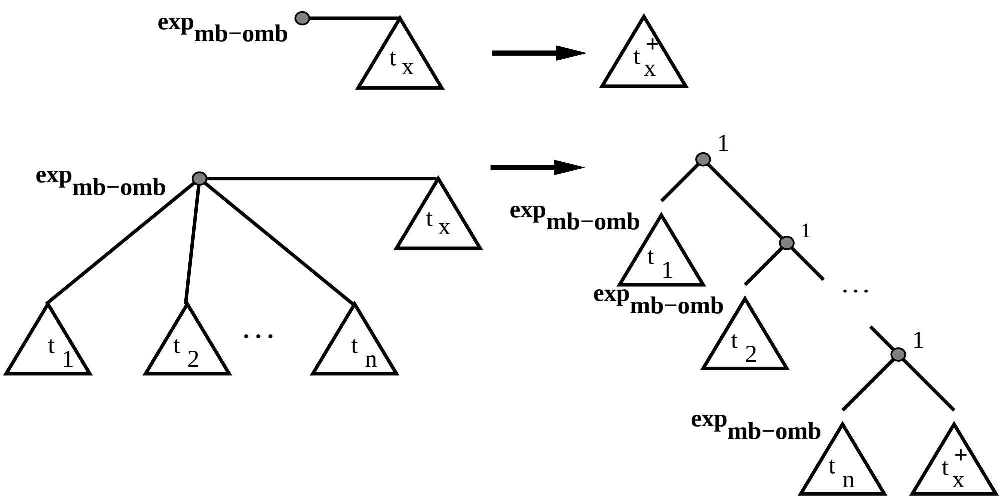
Passing to the coefficients of the generating functions we obtain the following enumeration result.

Theorem 7.7 *There are as many trees in $O[MB_d^{\square}]$ with a total of $n+1$ nodes, as there are trees in $MB_{d+1}^{\square}$ with n nodes. More formally: $|\{t \in O[MB_{d+1}^{\square}] \mid |t| = n+1\}| \;=\; |\{t \in MB_d^{\square} \mid |t| = n\}|$.*

7.3 Auxiliary Transformations

Definition 7.8 (Transformation $\exp_{\mathrm{omb-mb}}$ **)**

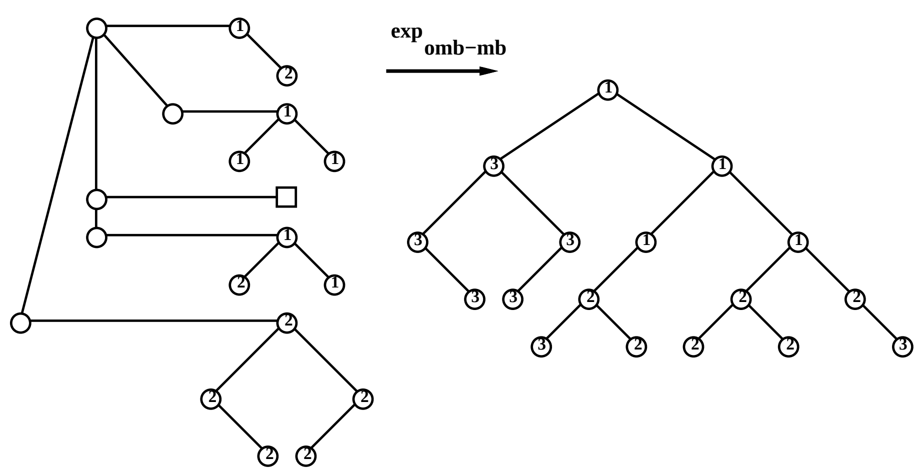
The transformation $\exp_{\mathrm{omb-om}}$ *:* $O[MB^{\Box}_{d-1}] \mapsto MB^{\Box}_{d}$ *is defined as follows:*



The tree $t+$ *are obtained from* t *by increasing all labels by 1.*

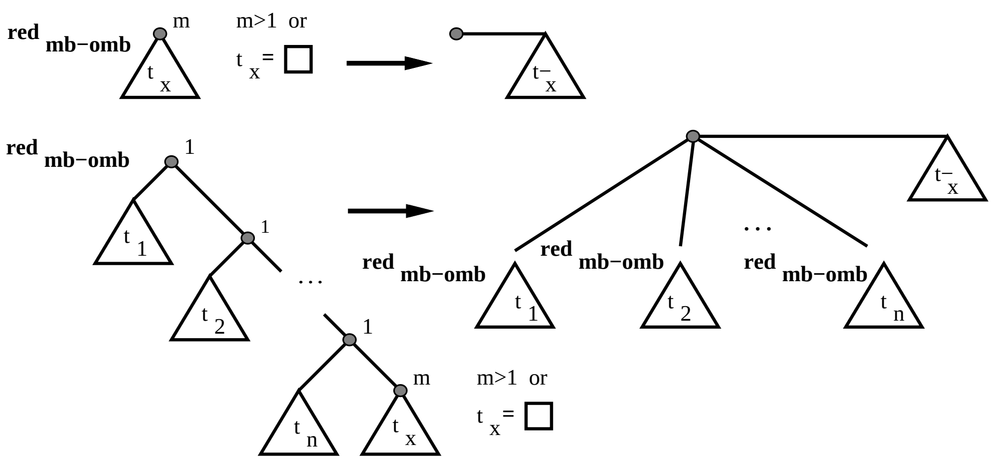
Note that this transformation is very similar to $\exp_{\mathrm{ob}}$.

Example 7.9 (Application Example of Transformation $\exp_{\mathrm{omb-mb}}$ **)**



Definition 7.10 (Transformation $\mathrm{exp}_{\mathrm{omb-om}}$ **)**

The transformation $\mathrm{red}_{\mathrm{omb-mb}} : MB_d^{\square} \mapsto O[MB_{d-1}^{\square}]$ *is defined as follows:*

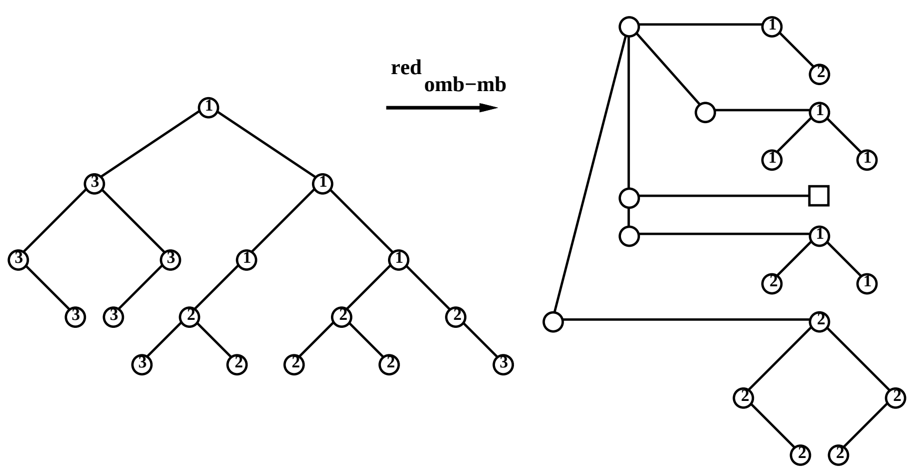


The tree $t-$ are obtained from t by decreasing all labels by 1.

Note that this transformation is very similar to $\mathrm{red}_{\mathrm{ob}}$.

“m” is the node on the rightmost branch that is labeled 1 and as close to the root as possible.

Example 7.11 (Application Example of Transformation $\mathrm{red}_{\mathrm{omb-mb}}$ **)**



Theorem 7.12 *The transformation* $\exp_{\mathrm{omb-mb}}$ *is a bijection between the tree families* $O[MB^{\square}_{d-1}]$ *and* $MB^{\square}_{d}$.

Proof:

Remark on the node numbers:

Let t be in $O[MB^{\square}_{d}]$ with $\vec{m}(t) = (m_1, \ldots, m_{d-1})$ and $m_0(t) = m_0)$. Obviously $\vec{m}(\exp_{\mathrm{omb-mb}}(t)) = (m_0 - 1, m_1, \ldots, m_{d-1})$ holds.

$\exp_{\mathrm{omb-mb}}$ *is injective:*

Let $t^1 \neq t^2$ be two trees in $O[MB^{\square}_{d}]$. We will show that $\exp_{\mathrm{omb-mb}}(t^1) \neq \exp_{\mathrm{omb-mb}}(t^2)$. Because of the remark on the node numbers we can assume $m_0(t^1) = m_0(t^2)$.

The proof is an induction on the number of unmarked nodes $n = m_0(t^1) = m_0(t^2)$, which is equivalent to the number of nodes in the ordered header tree.

Basis $n = 1$: Because there is only one tree with a single node, the trees from $MB^{\square}_{d-1}$ which are attached to them must be different, but then the transformed trees are clearly distinct.

Now, suppose the induction hypothesis be valid for node numbers less than n and let t^1 and t^2 be 2 trees with n nodes in the header tree. These trees must consist of subtrees as depicted in Definiton 7.8 We distinguish two different cases according to the subtree(s) in which t^1 and t^2 differ:

1. $t^1_i \neq t^2_i$

   Using the induction hypothesis it follows that: $\exp_{\mathrm{omb-bm}}(t^1_i) \neq \exp_{\mathrm{omb-mb}}(t^2_i)$ and thereby $\exp_{\mathrm{omb-mb}}(t^1) \neq \exp_{\mathrm{omb-mb}}(t^2)$

2. $t^1_x \neq t^2_x$:

   We have $t^1_x+ \neq t^2_x+$ and thereby $\exp_{\mathrm{omb-mb}}(t^1) \neq \exp_{\mathrm{omb-mb}}(t^2)$

$\exp_{\mathrm{omb-mb}}$ *is bijective:*

This follows immediately from the fact that $\exp_{\mathrm{omb-mb}}$ is a mapping between two sets of the same cardinality. (Theorem 7.7).

$\square$

By the above remark on the nodes numbers we get the following refinement of Theorem 7.7:

Corollary 7.13 (Refined Theorem 7.7)

$$|\{t \in MB^{\square}_{d} \mid \vec{m}(t) = (n_1, \ldots, n_d)\}| = \tag{67}$$

$$|\{t \in O[MB^{\square}_{d-1}] \mid \vec{m}(t) = (n_2, \ldots, n_d) \wedge m_0(t) = n_1 + 1\}| \tag{68}$$

This can be written more compactly as an equation of generating functions:

$$z_1 \widetilde{MB}^{\square}_{d}(z_1, \ldots, z_d) = O[\widetilde{MB}^{\square}_{d-1}](z_1, z_2, \ldots, z_d) \tag{69}$$

An analogous result can be derived for $\mathrm{red}_{\mathrm{omb-mbx}}$ in a similar manner.

7.4 A One-to-to Correspondence: $O_d \leftrightarrow MB_d^\square$

Let t^1 be a tree in $MB_d^\square$ with $\vec{m}(t^1) = (m_1, \ldots, m_d)$.

The transformation $\mathrm{red}_{\mathrm{omb-mb}}$ generates a tree $t^2 \in O[MB_{d-1}^\square]$ with $\vec{m}(t^2) = (m_2, m_3, \ldots, m_d)$ which includes $1 + m_1$ trees from $MB_{d-1}^\square$.

These $1 + m_1$ trees can themselves be transformed into trees from $MB_{d-2}^\square$ yielding a tree t^3 with $\vec{m}(t^3) = (m_3, m_4, \ldots, m_d)$ which includes $1 + m_1 + m_2$ trees from $MB_{d-2}^\square$.

Iterating this d times, we get $1 \sum_{1 \le i \le d} m_i$ empty trees.

Removing these empty trees we finally obtain a tree in O_d.

Hence, we have found the desired one-to-one correspondence for Identity 2, namely $\mathrm{exp}_{\mathrm{o-mb}} : O_d \to MB_d^\square$ and $\mathrm{red}_{\mathrm{o-mb}} : MB_d^\square \to O_d$ which can be described as:

$$\mathrm{exp}_{\mathrm{o-mb}} = {\mathrm{exp}_{\mathrm{omb-mb}}}^d \circ \mathrm{add}\square \tag{70}$$

$$\mathrm{red}_{\mathrm{o-mb}} = \mathrm{rem}\square \circ {\mathrm{red}_{\mathrm{omb-mb}}}^d \tag{71}$$

$\mathrm{rem}\square$ stands for the removal of the empty trees from $O[MB_0^\square]$ as described above and $\mathrm{add}\square$ for the attachment of empty trees to every node of a tree in O. For $d = 1$ this essentially simplifies to the classical correspondence.

Looking at the effects of the transformations on node numbers we can state the following refinement of Corollary 7.2:

Theorem 7.14 (Refined Identity 2)

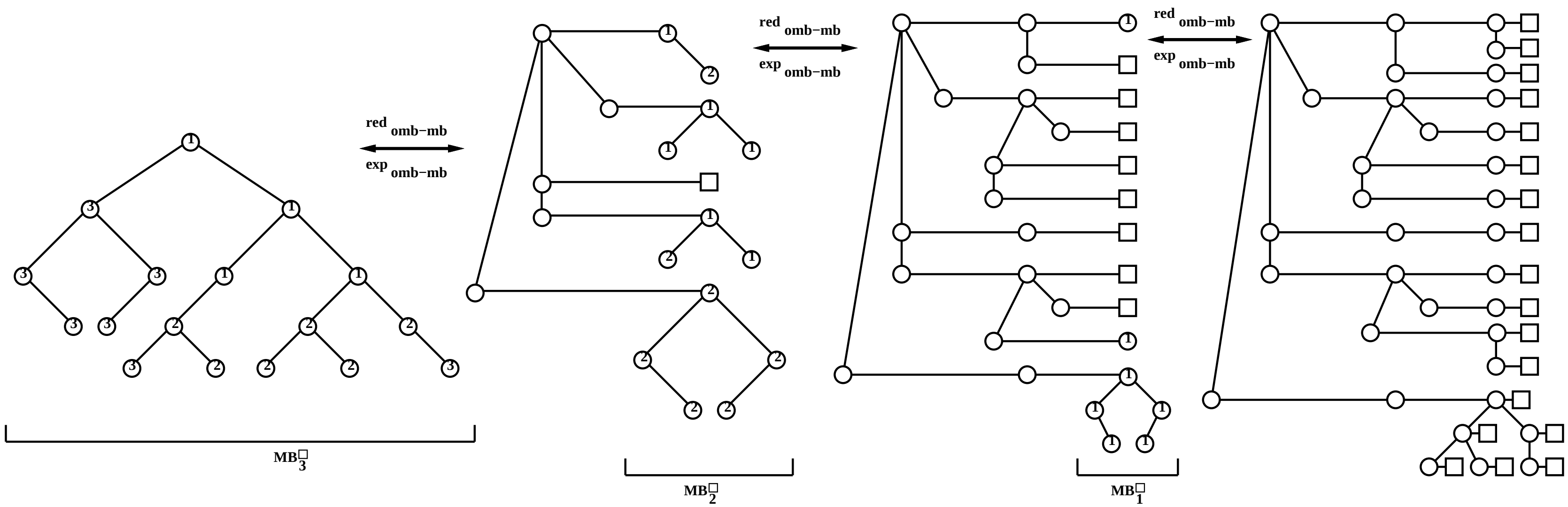
$$|\{t \in O_d \mid \vec{s}(t) = (n_1, \ldots, n_d)\}| = |\{t \in MB_d^\square \mid \vec{m}(t) = (n_1 - 1, n_2 - n_1, \ldots, n_d - n_{d-1}\}| \tag{72}$$

This can be written more compact as an equation of generating functions.

$$z_1 \widetilde{MB}_d^\square(z_1, \ldots, z_d) = \widetilde{O}_d(z_1/z_2, \ldots, z_{d-1}/z_d, z_d) \tag{73}$$

An example of the iterated application of the transformation is given in Example 7.15. (Note, that the empty trees in the last layer have not been removed in order to keep the figure at a reasonable size.)

Example 7.15 (An application example of $\mathrm{exp}_{\mathrm{o-mb}}$ **and** $\mathrm{red}_{\mathrm{o-mb}}$ **)**



8 Application: Analysis of the Label Distribution in EMO_d and MB_d

According to [Kem95, Theorem 7b] the asymptotic expected number of nodes in layer l of a d-dimensional simply generated tree with n nodes in the last layer $I(d,l,n)$ (assuming equidistribution of the trees) is given by:

$$I(d,l,n) \sim \xi(d,l)\, n \qquad (n \to \infty) \tag{74}$$

$$\text{with:} \quad \xi(d,l) = \prod_{l \le i \le d-1} \frac{\Theta(u_i) - u_i \Theta'(u_i)}{\Theta(u_i)} \tag{75}$$

Where the u_i are recursively defined by:

$$u_0 = \text{smallest positive solution of } x\Theta'(x) = \Theta(x) \tag{76}$$

$$u_{i+1} = u_i / \Theta(u_i) \ \text{ for } \ i > 0 \tag{77}$$

Note, that not all values for n might actually occur as valid node numbers in the last layer.

For the special case of multi-dimensional ordered trees we have:

$$\xi_o(d,l) = \prod_{s \le i \le d-1} (1-2u_i)/(1-u_i) \quad \text{ and } \quad u_1 = 1/4, \quad u_{i+1} = u_i(1-u_i) \ \text{ for } \ i > 0$$

From Theorem 6.23 and a symmetry argument we can infer that — assuming equidistribution of all trees from EMO_d with size n — the asymptotic expected number of occurrences of label m, denoted $M_o(d,m,n)$, is given by:

$$M_o(d,m,n) \sim I_o(d,m,n) - I_o(d,m-1,n) \qquad (n \to \infty)$$

We have — in a slightly different shape — rediscovered a result from [Kem93a] where a one-to-one correspondence between monotonically ordered trees and multi-dimensional **extended** binary trees was presented.

For the special case of multi-dimensional binary trees we have:

$$\xi_b(d,l) = \prod_{l \le i \le d-1} (1-u_i^2)/(1+2u_i+u_i^2) \quad \text{ and } \quad u_0 = 1, \quad u_{i+1} = u_i/(1+2u_i+u_i^2) \ \text{ for } \ i > 0$$

From Theorem 7.14 we can infer that — assuming equidistribution of all trees from MB_d with size n — the expected asymptotic number of occurrences of label m, denoted $M_b(d,m,n)$, is given by:

$$M_b(d,m,n) \sim I_b(d,m,n) - I_b(d,m-1,n) \qquad (n \to \infty)$$

The following tables state the asymptotic expected label distributions for small values of d:

Label Distribution of MB_d in Percent

d \ m	1	2	3	4	5	6
1	100.0					
2	66.7	33.3				
3	51.2	25.6	23.1			
4	42.1	21.0	18.9	18.0		
5	35.8	17.9	16.1	15.3	14.8	
6	31.3	15.6	14.1	13.4	12.9	12.7

Label Distribution of EMO_d in Percent

d \ m	1	2	3	4	5	6
1	100.0					
2	60.0	40.0				
3	43.4	29.0	27.6			
4	34.2	22.8	21.7	21.3		
5	28.3	18.9	18.0	17.6	17.3	
6	24.1	16.1	15.3	15.0	14.8	14.6

9 Conclusions

We have presented two new correspondences between families of monotonically labeled and multi-dimensional simply generated trees generalizing the classical correspondence between binary and ordered trees.

The search for the two correspondences was motivated by enumeration results like Corollary 6.3 which challange one to establish systematic correspondences between the classes of objects involved.

The first correspondece solves an open problem from [Kem93a] namely to find a combinatorial interpretation of the functional relation $z^{-1}EMO_d(z)^2 = B_d(z)$ (Corollary 6.2). Kemp's paper also describes another correspondence between monotonically labeled ordered and multi-dimensional extended binary trees.

The question arises naturally, whether there are more such correspondences. Or, why such correspondences are limited to binary and ordered trees only. No systematic approach to answer this question has been undertaken so far.

The actual coefficients of the generating functions defined in equations 8/9 and 27/28 may be recovered by computational inversion. Mathematica routines for computing these coefficients can be obtained by email from `muth@cs.arizona.edu`.

References

[Fla88] Philippe Flajolet. Mathematical methods in the analysis of algorithms and data structures. In Egon Börger, editor, *Trends in Theoretical Computer Science*, pages 225–304. Computer Science Press, 1988.

[Kem89] Rainer Kemp. Binary search trees for d-dimensional keys. *J. Inform. Process. Cybernet. EIK*, 25(10):513–527, 1989.

[Kem93a] Rainer Kemp. Monotonically labelled ordered trees and multidimensional binary trees. In Zoltán Ésik editor, *Fundamentals of Computation Theory*, pages 329–341. Springer, 1993. Lecture Notes in Computer Science Vol. 710.

[Kem93b] Rainer Kemp. Random multidimensional binary trees. *J. Inform. Process. Cybernet. EIK*, 29(1):9–36, 1993.

[Kem95] Rainer Kemp. On the inner structure of multidimensional simply generated trees. *Random Structures and Algorithms*, 6:121–146, 1995.

[Knu73] Donald E. Knuth. *Fundamental Algorithms*, volume 1 of *The Art of Computer Programming*. Addison-Wesley, second edition, 1973.

[MM78] A. Meir and John W. Moon. On the altitude of nodes in random trees. *Can. J. Math.*, 30(5):997–1015, 1978.

[PU83] Helmut Prodinger and Friedrich J. Urbanek. On monotone functions of tree structures. *Disc. Appl. Math.*, 5:223–239, 1983.