Publ. I.R.M.A. Strasbourg, 1988, 358/S-18
Actes 16^e Séminaire Lotharingien, p. 77-86

FUNCTIONAL EQUATIONS FOR DATA STRUCTURES

BY

FRANÇOIS BERGERON, GILBERT LABELLE
AND PIERRE LEROUX

Abstract:

We show how tree-like data structures (B-trees, AVL trees, binary trees, etc. ...) can be characterized by functional equations in the context of the theory of **species of structures** which has been introduced as a conceptual framework for enumerative combinatorics. The generating functions associated to these abstract data structures are directly derived from the corresponding functional equations.

1. Introduction

Often the cost study of insertion, deletion and interrogation on tree-like data structures, involves the determination of the mean value of some parameters on these structures (height, number of branches or nodes, etc. ...), or an estimate of the number of structures that have fixed values for those parameters. Clearly, the two problems are closely related, and the object of this work is to show how modern combinatorial techniques can be applied to solve them.

The use of generating functions in enumerative problems has a long history, and many authors, such as Knuth, Flajolet, Françon, or Vuillemin (see [5],[6],[8],[10],[12], and [18]), have stressed the interest of these techniques in the cost study of algorithms. Let us briefly outline the steps involved in this approach. Recall that a generating function, $A(x)$ or $A(t,x)$, is associated to a given data structure in the following way:

$$A(x) = \sum_{n=0}^{\infty} a_n \frac{x^n}{n!} \qquad \text{or} \qquad A(t,x) = \sum_{n=0}^{\infty}\sum_{k=0}^{\infty} a_{k,n} \frac{t^k x^n}{n!}$$

where the a_n's , and the $a_{k,n}$'s respectively count the number of structures on a set of n data, and the number of such structures for which k is the value of some parameter. By a combinatorial argument we often obtain a functional equation for this generating function. Thus the study of the generating function, solution of this equation, becomes an algebraic or analytical problem.

With the support of the NSERC Canada (grants A9041* and A5660†‡), and FCAR Québec (grant EQ1608*†‡).

We shall show that in the framework of species of structures (see [9]), functional equations correspond exactly to the specifications of data structures, and the functional equations for the corresponding generating functions are built in these specifications. More precisely, there are combinatorial operations on species (of structures) corresponding to sum, product, derivative, integral and substitution of their generating functions. Moreover, we know (see [11] and [12]) that there is a unique solution to a functional equation (with the usual type of conditions) in the algebra of species. This is how a functional equation specifies a species. Further, we will show how this solution can be readily described for the so called "Bajraktarevic's equations" (see [1],[4], and [15]), of the form:

i) $A(X) = F(X, A(G(X)))$, or

ii) $A(X,Y) = F(X,Y,A(G(X,Y),H(X,Y)))$

We will also give examples of many popular data structures which are defined by such equations. These include binary trees, 2-3-trees (or more generally, a-b-trees), AVL and other balanced trees.

2. Elementary functional equations

In this section we would like to show how tree-like structures appear as solutions of functional equations as illustrated by the following example.

Example 1.

Let B denote the species of **binary trees**, where $B[S]$ is the set of all binary trees for which the set of nodes is S. Thus a B-structure is a binary tree. Then the well known generating function

$$B(x) = \frac{1 - \sqrt{1-4x}}{2x}$$

for binary trees, is deduced from the functional equation

$$B(x) = (1 + xB^2(x))$$

which corresponds (as we shall see) to the combinatorial specification of the species B of binary trees by the equation

$$B = 1 + XB^2. \qquad \Delta$$

More generally, Joyal and Labelle have formulated and proven combinatorial versions of the implicit function theorem:

Theorem

Let $F(X,Y)$ *be a species of two sorts, such that*

$$F[\emptyset,\emptyset] = \emptyset \quad \textit{and} \quad (\partial F/\partial Y)[\emptyset,\emptyset] = \emptyset,$$

then there exist a unique (up to isomorphism) species $A = A(X)$ *such that:*

$$A = F(X,A).$$

Proof (see [9] and [12]). Δ

Let us illustrate the corresponding combinatorial construction for the special case

$$A = X \cdot R(A), \tag{*}$$

with R a given species, such that $R[\emptyset] \neq \emptyset$. Equation (*) states that an A-structure on a set S is characterized by the selection of a point to which is attached an R-assembly of A-structures, such as represented in figure 5.

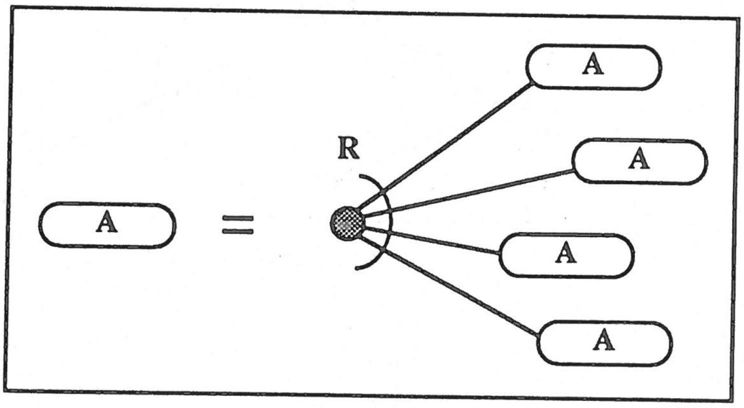

Figure 5

Unfolding this implicit recursion, we conclude that an A-structure is a tree-like structure, such that on each set of sons of the nodes of this tree, there is an R-structure, symbolically represented by an arc as in figure 6.

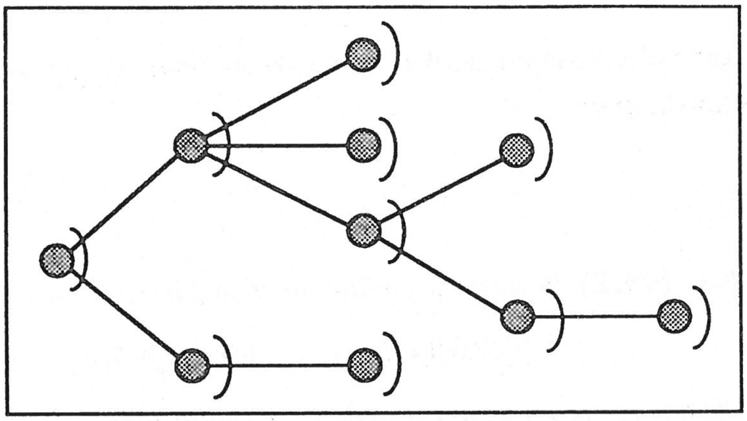

Figure 6

Another interesting special case of the implicit function theorem, is the equation

$$A = X + G(A), \qquad (**)$$

with G a given species such that $G[\emptyset] = \emptyset$, and $G'[\emptyset] = \emptyset$. Figure 7 represent a typical structure of the species A obtained as the unique solution of (**) by an approach similar to the one for equation (*).

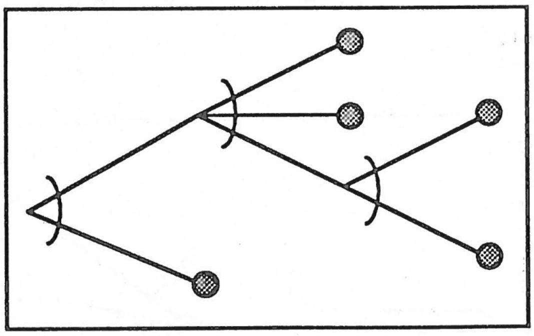

Figure 7

Example 2.

Let $G(Y) = fY^2+gY^3$, where f and g are formal parameters representing respectively a binary operation, and a ternary operation. Then the solution of (**) corresponds to the species whose structures on the underlying set $\{x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9\}$ are well formed expressions of the kind:

$$fgx_2fx_9x_1x_4fx_7gx_6x_5fx_8x_2.$$

3. Functional equations

Main lemma

For given species G(X) *and* F(X,Y) *on one and two sorts respectively, the general solution (if it exists) of the functional Bajraktarevic's equation:*

(*) $$A(X) = F(X, A \circ G(X))$$

is the species of tree-like structures such as represented in figure 8, where the arc on the sons of an internal node means that there is a structure of the species indicated on the sons of that node. Moreover, on the unique path going from the root to any leaf, the sequence of structures that are encountered is always of the form $F^{n+1}G^n$.

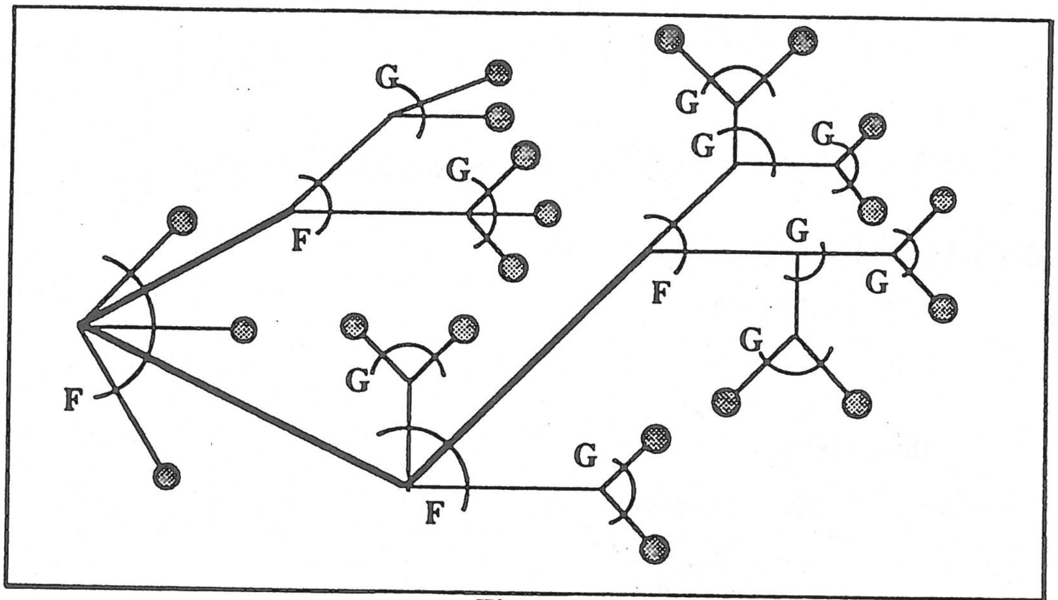

Figure 8

Δ

Example 3.

The case $F(X,Y) = X+Y$ and $G(X) = X^2+X^3$ gives rise to the 2-3 trees, and more generally, the a-b trees evidently correspond to the equation:

$$A(X) = X+X^2+ \dots +X^{a-1} + A(X^a+X^{a+1}+ \dots +X^b). \qquad \Delta$$

These are instances of an interesting specialization of equation (*), called the linear case:

(**) $$A(X) = Q(X) + P(X)\cdot A(G(X)),$$

where Q, P and G are given species. It is easy to deduce that:

Corollary

The general solution of $A(X) = Q(X) + P(X)\cdot A(G(X))$ *is of the form:*

$$A(X) = \sum_{k=0}^{\infty} P_k(X)\, Q(G^{<k>}(X))$$

where $G^{<k>}(X)$ *and* $P_k(X)$ *are defined recursively by:*

$$G^{<k>}(X) = \begin{cases} G(G^{<k-1>}(X)) & \text{, if } k>0, \\ X & \text{, if } k=0. \end{cases}$$

$$P_k(X) = \begin{cases} P_{k-1}(X)\cdot P(G^{<k-1>}(X)) & \text{, if } k>0 \\ 1 & \text{, if } k=0. \end{cases}$$

Δ

There is a simple algorithm giving a fast approximation of this solution:

Algorithm L (L for linear)

Let $A_1 := Q(X)$
$B_1 := P(X)$
$G_1 := G(X)$
then repeat
$G_{k+1} := G_k \circ G_k$
$B_{k+1} := B_k \cdot (B_k \circ G_k)$
$A_{k+1} := A_k + B_k \cdot (A_k \circ G_k)$
until satisfied.

Δ

In the context of generating functions, algorithm L gives an approximation A_k of the function $A(x)$ for which the number of correct terms in the series doubles at each iteration. The costly operation in this case is substitution but it can be done in $O(n\log(n))^{3/2}$ (see [2]).

The introduction of a formal parameter "t" in equation (**) permits the measure of the **iterative depth** of structures:

$$A(X) = P(X) + t\cdot Q(X)\cdot A(G(X)).$$

This corresponds to the enumeration of structures with a weight. We will illustrate this with a simple case. Figure 9 gives a representation of $G^{<k>}$-structures.

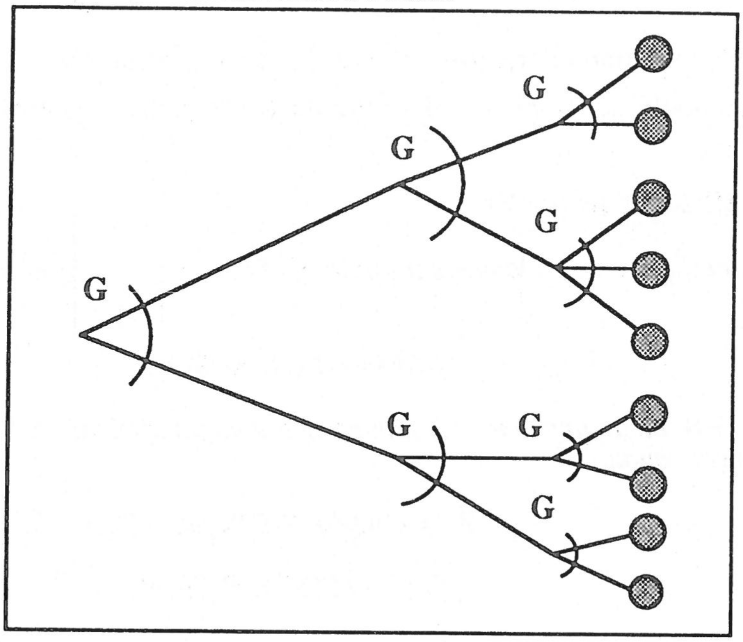

Figure 9

here k=3, let us say that this structure has weight t^3. The generating function associated with the solution of the equation:

$$A_t(X) = X + t \cdot A_t(G(X))$$

is the species A_t of all trees with G-structures on the sons of internal nodes, and with all leaves at the same depth . Each of those trees is given a weight t^d where d is the depth of the tree. The corresponding generating function is:

$$A(t,x) = \sum_{n=0}^{\infty}\sum_{k=0}^{\infty} a_{k,n} \frac{t^k x^n}{n!}$$

where $a_{k,n}$ is the number of such trees with n leafs and depth k.

Example 4.

It is well known that binary trees permit the coding of arithmetic expressions. The following equation (see [7]) characterizes binary trees with a parameter that measures the number of registers needed for the evaluation of such expressions:

$$B_t(x) = 1 + \frac{tx}{1-2x} B_t\left(\left(\frac{x}{1-2x}\right)^2\right)$$

Δ

We will now adapt Newton-Raphson's method for computing numerical approximations of the solution of an equation of the form y = f(x,y) (see [2] and [3]), to the species approximating the solution of Bajraktarevic's equation. Let us say that two species A and B have a contact of order k, if and only if, for all sets S of cardinality less than or equal to k, $A[S] \cong B[S]$.

Proposition R (R for reduction)

Let the species B *have a contact of order* k *with the solution* A(X) *of the equation:*

$$A(X) = F(X, A \circ G(X)),$$

and let ΔB *be an approximation having a contact of order* 2k+1 *with the solution of the linear equation:*

$$Z(X) = Q(X) + P(X) \cdot Z(G(X));$$

where

$$Q(X) = F(X, B \circ G(X)) - B,$$

and

$$P(X) = \frac{\partial F}{\partial Y}(X, B(G(X)))$$

then $B + \Delta B$ *has a contact of order* 2k+1 *with* A(X).

The proof is a simple combinatorial argument. Δ

Thus the problem is reduced to the approximation of the solution of a linear equation, which can be done by algorithm L.

4. Extensions

The extension to the many variables case is easily obtained by the introduction of different kinds of points in figure 8. An interesting examples comes from the study of **AVL-trees**, or **1-2 brother trees** (see [15]); the corresponding species is $A(X) = U(X,X)$, where $U(X,Y)$ is the species characterized by the functional equation:

$$U(X,Y) = X + U(X^2+2XY, X).$$

Other type of extensions involve systems of differential equations, which are more easily studied in the context of species on linear orders (see [13]) for which integration can be defined. It is then possible to characterize rooted trees with labels that are in increasing order when one goes from the roots to the leaves.Thus complete increasing binary trees are characterized by the differential equation $Y' = 1+Y^2$.

Bibliography

[1] M. Bajraktarevic; *Sur une équation fonctionnelle*, Glasnik Mat.-Fiz. I Astr. 12, 1957, pp 201-205.

[2] R.P. Brent and H.T. Kung; *Fast Algorithms for Manipulating Formal Power Series*, JACM 25 (4), 1978, pp 581-595.

[3] H. Décoste, G. Labelle, and P. Leroux; *Une approche combinatoire pour l'itération de Newton-Raphson*, Adv. in Appl. Math. 3, Acad. Press, 1982, pp 407-416.

[4] S. Dubuc; *Une équation fonctionnelle pour diverses constructions géométriques*, Annales des Sciences mathématiques du Québec, Vol. 9, No.2, 1985, pp 151-174.

[5] P. Flajolet; *Elements of a General Theory of Combinatorial Structures*, Proceedings FCT85,L. Budach, Ed., Lect. Notes in Comp. Science, Springer-Verlag, Vol. 199 (1985), pp. 112-127.

[6] J. Françon; *On the Analysis of Algorithms for Trees*, Theor. Comp. Science, Vol 4, 1977, pp 155-169.

[7] J.Françon; *Sur le nombre de registres nécessaires à l'évaluation d'une expression arithmétique*, RAIRO, Informatique théorique, 18 (1984), pp. 355-364.

[8] G.H. Gonnet; *Handbook of Algorithms and Data Structures*, International Computer Science Series, Addison-Wesley, 1984.

[9] A. Joyal; *Une théorie combinatoire des séries formelles*, Adv. in Math. 42, 1981, pp 1-82.

[10] D. Knuth; *The Art of Computer Programming*, vol.2, Addison-Wesley, 1981.

[11] G. Labelle; *Some New Computational Methods in the Theory of Species*, in Combinatoire énumérative, Proceedings, Montréal, Québec 1985, ed. G.Labelle and P.Leroux, Springer Lecture Notes in Math., No.1234, 1986, pp. 192-209.

[12] G. Labelle; *Une combinatoire sous-jacente au théorème des fonctions implicites*, J. Compin. Theory, Series A, 40, 1985, pp 377-393

[13] P. Leroux and G. Viennot; *Combinatorial Resolution of Systems of Differential Equations I, Ordinary Differential Equations*, in Combinatoire énumérative, Proceedings, Montréal, Québec 1985, ed. G.Labelle and P.Leroux, Springer Lecture Notes in Math., No.1234, 1986.

[14] K. Mehlhorn; *Data Structures and Algorithms, I : Sorting and Searching* , Springer - Verlag, 1985.

[15] T. Ottmann and D. Wood; *1-2 Brother Trees or AVL Trees Revisited* , Comp. Jour., Vol 23 (3), 1980, pp 248-255.

[16] A. Odlycko; Periodic Oscillations of coefficients of Power Series that satisfiy Functional equations, Adv. in Math., 44 (1982),pp. 180-205.

[17] A.H. Read. *The Solution of a Functional Equation*, Proc. Royal Soc. Edinburgh, A 63, 1951-1952, pp 336-345.

[18] J. Vuillemin; *A Unifying Look at Data Structures* , Comm. of ACM, Vol 23 (4), 1980, pp 229-239.

[19] A.C-C. Yao; *On Random 2-3 Trees* , Acta Informatica, Vol. 9, 1978, pp 159-170.

François BERGERON, Gilbert LABELLE
et Pierre LEROUX,
Département de mathématiques et d'informatique,
Université du Québec à Montréal,
Case Postale 8888, Succ. "A",
Montréal, H3C 3P8, Canada