Algorithms Seminar 2002–2004,
F. Chyzak (ed.), INRIA, (2005), pp. 51–54.

Available online at the URL
http://algo.inria.fr/seminars/.

Patterns in Trees

Thomas Klausner
Technical Mathematics, Technische Universität Wien (Austria)

December 9, 2002

Summary by Marianne Durand and Julien Clément

Abstract

Given a tree, considered as a pattern, the question is how many times this pattern appears in a tree of size n. The average and variance of this parameter are obtained in this talk.

1. Introduction

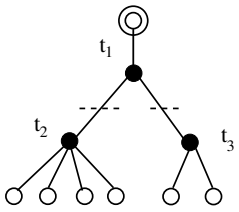
The problem of counting the number of occurrences of a pattern in a general tree is motivated for example by compression of arithmetical expressions. This talk presents first a simpler problem, that is counting planted patterns in planted trees, and reduces this problem to solving asymptotically a system of functional equations satisfied by related generating functions. The second part shows that the problem of general trees and general patterns is in fact very close to the planted problem, and can be reduced in a similar way to solving certain systems of functional equations. In the last section, asymptotic results on the number of occurrences of a given pattern in trees of size n are found from those systems, namely a normal distribution with explicit mean and variance.

2. Planted Trees and Planted Patterns

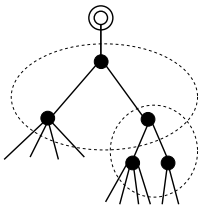
A planted tree is a rooted tree, where the degree of the root is equal to 1. To begin with, it is simpler to search planted patterns in planted trees.

2.1. **Combinatorial decomposition.** The search of a planted pattern in a planted tree is as follows. First, see if the pattern and the tree match when you match the two roots, this may give an occurrence, second build planted subtrees, and search recursively inside. Building planted subtrees consists in erasing the root (this gives a tree, as the root was of degree 1) and then split the new root into a root for each of his sons, to create a forest of planted trees.

FIGURE 1. Example of a pattern.



(a) A white circle stands for any tree.



(b) A tree in which the pattern occurs twice.

The pattern shown in Figure 1 is the example we use all along this summary. The pattern is first decomposed in planted subtrees (also named sub-patterns) which are named t_i (the ordering does not matter). Formally to obtain a planted subtree, one has to cut an internal edge in 2, and add a planted root on the cut side of the edge . To get all the subtrees, do it for all internal edges. The pattern is then fully known by the relation between its sub-patterns. For the example, we have the relations:

$$t_1 = o \times t_2 \times t_3$$
$$t_2 = o \times p \times p \times p \times p$$
$$t_3 = o \times p \times p$$

where o stands for the (planted) root, and p for any tree. In a planted tree the root is used to indicate where is the "top," so when a planted tree is seen as a subtree of a planted tree, this information is no longer necessary. This explains why the relation $t_1 = o \times t_2 \times t_3$ holds, with the subtrees t_2 and t_3 "losing" their planted root.

Now with this description, we are able to search recursively a planted pattern in a planted tree, but to count them, we have to take care of overlaps as shown in Figure 1(b). The reason appears during the writing of the generating function equation. Overlaps are possible, because of the non trivial intersection of the definition of t_1 and t_3. To avoid this, it is sufficient to rewrite the t_i differently to obtain a disjoint set of specifications (that is no tree satisfy two specifications). The t_i define sub-sets with overlaps of the set p of all trees, the symbols a_i are defined as standing for the underlying partition of the set, based on theoretic set operations (intersections, union, difference) involving the t_i's or the set of all trees p. Now all the a_i are disjoint (they are defined as a partition), and each t_i can be written as a union of a_i's.

The system of equations obtained on the t_i's is then easily translated into a system in the a_i's. In the example we obtain:

$$a_1 = t1 = \{o\} \times a_2 \times (a_1 \cup a_3)$$
$$a_2 = \{o\} \times p \times p \times p \times p$$
$$a_3 = (\{o\} \times (p \times p)) \setminus a_1$$
$$a_4 = p \setminus (a_1 \cup a_2 \cup a_3)$$

This is a system involving only the a_i's, as the relation $p = a_1 \cup a_2 \cup a_3 \cup a_4$ holds. In the a_i's basis, only a_1 represents a pattern, and so can be counted as a pattern, that is marked with a u symbol in the generating function, as explain in the next paragraph. Whereas in the t_i basis, t_3 may be a pattern, but without certainty, so that we do not know whether it should be counted or not.

2.2. **Generating functions.** The generating function of all trees is denoted by $p(z, u)$, where z codes the size of the tree, and u the number of patterns. So that $p(z, u) = \sum_{n,k} p_{n,k} \frac{z^n}{n!} u^k$, with $p_{n,k}$ the number of trees of size n (the number of nodes of the planted rooted tree) that contains k occurrences of the pattern. In what follows,if a letter a stands for a set of trees, then $a(z, u)$ stands for the corresponding generating function. As the set of all the trees p is decomposed as the disjoint union of the a_i's, the generating function $p(z, u)$ is the sum of the $a_i(z, u)$. For a presentation of the relation between combinatorial decomposition and generating functions, see [2]. Basically, the operations on sets are translated into operations on generating functions; unions of disjoint sets, exclusions and products translate into $+$, $-$ and $\times$. The system of equations between the a_i is translated into generating functions equations. The number of a_i's is denoted by L. All the relations but the last are written as equalities between a_j and the root times disjoint union or exclusion of a_i's, so that we have the relation $a_j(z, u) = zP_j(a_1(z, u), \ldots, a_L(z, u), u)$,

where P_j is a polynomial. The last variable of the polynomial, u, is used to mark the patterns. For the last relation, $a_L = p \setminus \cup a_i$, we get $a_L(z,u) = ze^{p(z,u)} - z\sum_{j=1}^{L-1} P_j(a_1(z,u),\ldots,a_L(z,u),1)$. To understand this equation, remember that the generating function of all trees, when there is no pattern to be counted, is $p(z) = ze^{p(z)}$. The rest is a basic translation, except for the last variable u. The P_j's have to be applied to 1 for their last variable, because in the set a_L, there is no pattern to be marked.

On the example, we have the system:

$$\begin{aligned}
a_1(z,u) &= uza_2(z,u)(a_1(z,u) + a_3(z,u)) \\
a_2(z,u) &= zp(z,u)^4 \\
a_3(z,u) &= z(p(z,u)^2 - a_1(z,u)) \\
a_4(z,u) &= ze^{p(z,u)} - z(a_2(z,u)(a_1(z,u) + a_3(z,u)) + p(z,u)^4 + p(z,u)^2 - a_1(z,u)).
\end{aligned}$$

The only pattern marked is in the first equation.

In this section, we have found how to obtain a system of equations satisfied by the generating function $p(z,u)$, that counts the number of occurrences of a given planted rooted pattern in planted rooted trees.

3. General Trees and General Patterns

Before searching general patterns in general trees, we consider the problem of searching two planted patterns in a planted tree. The number of patterns counted is the sum of the number of occurrences of the two patterns, eventually with overlaps. The idea is to make a "union" counting on the patterns. The technique is very similar to what is presented in section 2, so we just give the main lines.

The first pattern is decomposed into sub-patterns, named $t_1,\ldots,t_k$, the second pattern is also decomposed into the sub-patterns $t_{k+1},\ldots,t_j$. Then, all the t_i are grouped as if they came from the same pattern, the partition a_i is found from all the t_i's. The system of equation in the generating function $a_i(z,u)$ is built in a similar way, and both patterns are marked with a u. At the end we have a system of equation satisfied by the generating function $p(z,u)$ we are looking for.

Now that we know how to count (count is used in the sense of having an equation satisfied by the generating function) for two patterns, the next step is to count the occurrences of a not-planted pattern in a planted rooted tree. In order to do this, we build all possible ways of planting the patterns, and we consider the union of all these planted patterns as explained in the previous paragraph.

Finally to search a pattern (non-planted) in a tree (non-planted), we simply plant the tree, and then search the pattern inside the planted tree, as the planting of a tree does not change the number of occurrences of the pattern.

So the problem of counting the number of occurrences of a pattern in a tree is reduced to "solving" a system of equations of corresponding generating functions. Next section is devoted to finding asymptotic results on the coefficients of the generating function $p(z,u)$.

4. Asymptotic

To study the asymptotic behavior of the $p_{n,k}$, defined as coefficients of $p(z,u) = \sum p_{n,k}\frac{z^n}{n!}u^k$, we rely on a general theorem of Michael Drmota [1], that studies solutions of systems of functional

equations. Let X_n denotes the random variable of law defined by

$$\mathbf{P}(X_n = k) = \frac{p_{n,k}}{p_n} \qquad \text{with } p_n = \sum_k p_{n,k}.$$

This theorem, under some conditions, proves that X_n is asymptotically Gaussian, and provides explicit formulas for the mean and variance, that are here proportional to n.

To start with, we define some notations. Bold letters always stand for a vector. The letter $\mathbf{a}$ denotes the vector $(a_1(z,u),\dots,a_L(z,u))$. The symbol $\mathbf{F}$ stands for

$$\mathbf{F}(u,\mathbf{y},z) = (F_1(u,\mathbf{y},z),\dots,F_L(u,\mathbf{y},z))$$

where

$$F_i(u,\mathbf{y},z) = zP_i(\mathbf{y},u) \quad (\text{for } 1 \le i < L), \qquad \text{and} \qquad F_L(u,\mathbf{y},z) = ze^{\sum_{i=1}^{L} y_i} - z\sum_{i=1}^{L-1} P_i(\mathbf{y},1).$$

So that the system of functional equations can be rewritten as $\mathbf{a} = \mathbf{F}(u,\mathbf{a},z)$. The differentiation of a function f with respect to x is written f_x, and $\mathbf{F}_\mathbf{y}$ is the matrix with entries $\frac{\partial F_i}{\partial y_j}$.

We want the system to be strongly connected, which means that there is no sub-system that can be solved independently. This is verified when the pattern does not contain leaves, as each a_i but the last depends on the last (as each sub-pattern ends into at least one unspecified tree), and a_L depends on all a_i but itself. For the particular case where the pattern requires the presence of leaves, the branch leading to the leaf is considered as a black box, and plays no role in the decomposition. This particular case can thus be reduced to the general case without leaves.

A slightly modified version of the main theorem from [1] now says that under conditions that are satisfied in this context, X_n is asymptotically Gaussian, with mean and variance

$$\mathbf{E}[X_n] = \mu n + O(1) \qquad \text{and} \qquad \mathbf{Var}[X_n] = \sigma^2 n + O(1).$$

The value of the constants μ and σ comes from the solution of the system of functional equations

$$\begin{aligned} \mathbf{y} &= \mathbf{F}(u,\mathbf{y},z) \\ 0 &= \det(\mathbf{I} - \mathbf{F}_\mathbf{y}(u,\mathbf{y},z)) \end{aligned}$$

that admits a solution $\mathbf{y}(z)$ and $u(z)$. The constants μ and σ^2 are given by

$$\mu = -\frac{u_z(1)}{u(1)} \qquad \text{and} \qquad \sigma^2 = -\frac{u_{zz}(1)}{u(1)} + \mu^2 + \mu.$$

Bibliography

[1] Drmota (Michael). – Systems of functional equations. *Random Structures and Algorithms*, vol. 10, n° 1–2, 1997, pp. 103–124.

[2] Sedgewick (Robert) and Flajolet (Philippe). – Analytic combinatorics—Symbolic combinatorics. – 2005. `http://algo.inria.fr/flajolet/Publications/books.html`.