Combinatorics, Probability and Computing (2004) 13, 475–497. © 2004 Cambridge University Press
DOI: 10.1017/S0963548304006273 Printed in the United Kingdom

And/Or Trees Revisited

B. CHAUVIN[1], P. FLAJOLET[2],
D. GARDY[3] and B. GITTENBERGER[4†]

[1] LAMA, CNRS UMR 8100, Université de Versailles Saint-Quentin,
78035 Versailles Cedex, France
(e-mail: chauvin@math.uvsq.fr)

[2] INRIA Rocquencourt, Projet Algorithmes, Domaine de Voluceau,
78153 Le Chesnay, France
(e-mail: Philippe.Flajolet@inria.fr)

[3] PRISM, CNRS UMR 8144, Université de Versailles Saint-Quentin,
78035 Versailles Cedex, France
(e-mail: Daniele.Gardy@prism.uvsq.fr)

[4] Department of Geometry, Technische Universität Wien,
Wiedner Hauptstraße 8-10/113,
A-1040 Wien, Austria
(e-mail: Bernhard.Gittenberger@tuwien.ac.at)

Received 11 December 2002; revised 31 December 2003

We consider Boolean functions over n variables. Any such function can be represented (and computed) by a complete binary tree with and or or in the internal nodes and a literal in the external nodes, and many different trees can represent the same function, so that a fundamental question is related to the so-called *complexity* of a Boolean function: $L(f) :=$ minimal size of a tree computing f.

The existence of a limiting probability distribution $P(\cdot)$ on the set of and/or trees was shown by Lefmann and Savický [8]. We give here an alternative proof, which leads to effective computation in simple cases. We also consider the relationship between the probability $P(f)$ and the complexity $L(f)$ of a Boolean function f. A detailed analysis of the functions enumerating some sub-families of trees, and of their radius of convergence, allows us to improve on the upper bound of $P(f)$, established by Lefmann and Savický.

† This author's work has been supported by the Austrian Science Foundation FWF, grant P16053-N05.

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

1. Introduction

Random And/Or Boolean formulas and functions play an important rôle in the literature of theoretical computer science, and one of the fundamental questions they raise is that of their representation by a data structure such as a tree or a circuit. Some properties of a representation, such as its size, are actually properties of the associated Boolean function, and estimating the usefulness of a given representation, *e.g.*, its average size, quickly leads to investigating probability distributions on some space of Boolean functions.

Several recent works have attempted to define probability distributions for suitable models of such functions. For instance, Friedman [6] investigated distributions involved in the random k-SAT problem, where iterated conjunctions of small disjunctions appear. A sequence of probability distributions defined on formulas of the same size and with the same tree structure appear in his model. Paris, Vencovská and Wilmers [11] show that the proportion $pr_{n,k}$ of Boolean formulas f on n variables which take the value true for k affectations of the variables (and the value false for the remaining $2^{2^n} - k$ affectations) has a well-defined limit, when the number n of Boolean variables and the formula size both tend to infinity; this amounts to defining the limiting probability distribution of the $pr_{n,k}$. Closely related to the present work, and indeed at its origin, are some papers by Savický and others. Lefmann and Savický [8] obtain a relation between the probability of a Boolean function and its complexity. Further papers (Savický [15], Savický and Woods [16]) establish relationships between functions of given limited complexity and some probability distributions or some enumeration results. By contrast, earlier results of Savický [13, 14] present a way of iteratively building Boolean formulas, which leads to a limiting uniform distribution on the set of Boolean functions on a fixed number of variables, and is closest in spirit to the work of Friedman [6]. Woods [17] presents, among other results relative to logical sentences, a model of Boolean formulas that is closely related to our approach, both in the tools used (generating functions) and in the final result (existence of a limiting distribution on the set of Boolean functions).

We consider in the present paper the model for Boolean functions presented for example by Lefmann and Savický in [8], where the probability of a Boolean function is proportional to the number of Boolean formulas of a given type which compute it. The Boolean functions are defined on n variables $x_1, x_2, \ldots, x_n$. Since the literals are then $x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_n, \bar{x}_n$, there are 2^{2^n} such Boolean functions. All along the paper, n is fixed, and we consider some special values in Sections 2.2 to 2.4.

In such a context, formulas of size m with n variables are represented by labelled rooted binary trees where the m internal nodes are labelled by and and or, and the $m+1$ external nodes by a literal, *i.e.*, a variable or its negation. Each of the m inner nodes is labelled by and or or with equal probability $1/2$ and independently of the other nodes; each leaf is labelled by a literal, chosen according to the uniform distribution on the $2n$ literals and independently of the labelling of all the other nodes. Many different trees can compute the same function, so that a fundamental question is to evaluate the so-called *complexity* of a Boolean function, which we define as

$$L(f) := \text{minimal size of a tree computing } f.$$

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

In this paper, we define the *size* of a binary tree as the number of its internal nodes.[1] We should also mention that several different complexity measures for Boolean functions have been proposed in the literature; see *e.g.*, [2] for a recent survey.

The aim of this paper is dual: a better understanding of the limiting probability distribution on the space of Boolean functions, and a study of the relationship between the probability of a given Boolean function and its complexity. For the first topic, we need to make precise what we mean by *the probability of a given function.* As suggested by Woods [17] and further argued by Lefmann and Savický [8], a natural definition of the limiting distribution is as the limit of the uniform distribution on and/or trees of finite size approaching infinity. More precisely, for any fixed m there is a uniform distribution P_m on the set of and/or trees with n variables and m internal nodes. Let f be some Boolean function and define $t(f,m)$ as the number of trees of size m that compute f. Then the probability of this function is by definition

$$P_m(f) = \frac{t(f,m)}{T_m},$$

where T_m is the total number of and/or trees with size m (we give later on a simple formula (2.1) for T_m). For simplicity's sake, we use the same notation P_m for the distributions on trees and on functions, although the last one is actually the image probability of P_m by the canonical application which associates to a tree the Boolean function it computes.

Lefmann and Savický proved [8, Theorem 2.3] that these distributions P_m have a limit P when m goes to $+\infty$, which they describe as a biased tree distribution. Section 2 of this paper is devoted to an alternative description of this limiting distribution P and to some explicit computations for the cases $n = 1, 2$ or 3 using generating functions. The case of general n relies on results for systems of algebraic equations due to Drmota [3], Lalley [7] or Woods [17].

The combined approach by generating functions and branching processes also allows us to define a second probability distribution on Boolean functions: starting from a critical branching process, we label at random its internal and external nodes to obtain a random and/or tree, *i.e.*, a random Boolean function. We present this approach in Section 2.6, together with some numerical computations and comparisons with our first probability distribution P.

The second topic, namely the study of the relationships between $P(f)$ and $L(f)$, improves on the relation proved by Lefmann and Savický:

$$\frac{1}{4} \cdot \left(\frac{1}{8n}\right)^{L(f)+2} \leqslant P(f) \leqslant (1 + O(1/n)) \exp\left(-c\frac{L(f)}{n^3}\right). \tag{1.1}$$

The lower bound seems to be tight, but the upper bound can be improved, to yield an order n^{-2} instead of n^{-3}. Following Lefmann and Savický, our method is to start from Markov inequality: for any function f and for $\varepsilon > 0$, the definition of the complexity gives

[1] We can also choose the number of external nodes or the total number of nodes. In Lefmann and Savický's paper [8], the size is the number of external nodes, so there is a +1 shift when comparing our results to theirs.

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

(again using the same notation for both distributions P on trees and on functions, as we did for P_m)

$$P(f) = P(\text{the tree } \tau \text{ computes } f) \leqslant P((1+\varepsilon)^{\|\tau\|} \geqslant (1+\varepsilon)^{L(f)}) \leqslant \frac{E[(1+\varepsilon)^{\|\tau\|}]}{(1+\varepsilon)^{L(f)}}.$$

The upper bound in (1.1) can be improved as ε becomes larger. That means a better control of $E[(1+\varepsilon)^{\|\tau\|}]$, in other words a fine evaluation of the radius of convergence of the generating function of the size of a tree. This is achieved in Section 3 where the following theorem is proved.

Theorem 1.1. *Almost surely,*

$$\frac{1}{4}\cdot\left(\frac{1}{8n}\right)^{L(f)+2} \leqslant P(f) \leqslant (1+O(1/n))\exp\left(-c\frac{L(f)}{n^2}\right). \tag{1.2}$$

Finally we discuss our results, both on the improvement on Lefmann and Savický's bound for the complexity and on the probability distributions, in Section 4, where we also consider possible extensions to other models of Boolean formulas, which would take into account the commutativity and associativity of the Boolean operators.

2. Enumerating functions and limit distribution for and/or trees

We recall that the generating function for binary trees, counted by the number of (internal and external) nodes, satisfies the equation

$$b(z) = 1 + z\,b(z)^2,$$

which gives

$$b(z) = \frac{1-\sqrt{1-4z}}{2z}.$$

Now define the set $\mathscr{T}$ of and/or trees, assuming that the number of variables is n:

$$\mathscr{T} = \oplus_{1\leqslant i\leqslant n}(\{x_i\}+\{\bar{x}_i\}) \oplus (\wedge, \mathscr{T}, \mathscr{T}) \oplus (\vee, \mathscr{T}, \mathscr{T});$$

hence the equation on the generating function for these trees enumerated by number of *internal* nodes:

$$T(z) = 2n + 2z\,T(z)^2,$$

which gives

$$T(z) = \frac{1-\sqrt{1-16nz}}{4z}.$$

This gives readily the number of and/or trees with m internal nodes (a formula already given in former papers [8, 11]):

$$T_m := [z^m]T(z) = 2^m(2n)^{m+1}C_m, \tag{2.1}$$

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

with C_m the Catalan number: $C_m = (2m)!/m!(m+1)!$.

2.1. From the distribution on trees to the distribution on Boolean functions

We assume in this section that the probability distribution over and/or trees on n variables and of size m (*i.e.*, number of internal nodes, or number of leaves minus 1) is uniform. Let us stress again that this distribution depends on m, and that throughout this section n is a fixed parameter. This induces a probability distribution P_m over the Boolean functions on n variables: Let $t(f,m)$ be the number of trees of size m that compute a given Boolean function f;[2] then the probability of this function is

$$P_m(f) = t(f,m)/T_m.$$

Assume that we know the generating function $t_f(z) = \sum_m t(f,m)z^m$ enumerating the trees that compute the function f; then the probability $P_m(f)$ is simply

$$P_m(f) = \frac{[z^m]t_f(z)}{[z^m]T(z)}.$$

Hence knowing the asymptotic behaviour, as m tends to infinity, of the coefficients of the functions t_f (and of T) will give us the existence of a limiting distribution on the Boolean functions and possibly a way of computing it. Now Lefmann and Savický [8, Theorem 2.3] simply assert the existence of this limiting distribution:

$$P(f) = \lim_{m\to+\infty} P_m(f).$$

Hereafter, we first examine how we can explicitly compute the limiting distribution for $n = 1,\ldots,3$, before turning to the case of general n.

Remark. One might wish to study a different probability distribution $P_{\leqslant m}(f)$, defined as the ratio of the number of trees of size *smaller than or equal to m* that compute the Boolean function f, over the total number of trees of size smaller than or equal to m. Such an approach has the advantage that the supports of the probability distributions $P_{\leqslant m}$, for increasing m, are increasing subsets of the set of all (finite or infinite) binary trees labelled by $\wedge$, $\vee$ and literals. By the Kolmogorov existence theorem [1, Section 36], we know the existence of a limiting probability distribution on the set of *finite* and/or trees. Using generating functions, we can write

$$P_{\leqslant m}(f) = \frac{[z^m]\{t_f(z)/(1-z)\}}{[z^m]\{T(z)/(1-z)\}}.$$

Assume that the function $t_f(z)$ has a radius of convergence $1/16n$ (see below); as the function $T(z)$ has the same radius of convergence $1/16n$, we see that dividing by $1-z$ introduces, in both cases, a singularity at 1, larger than the radius of convergence; hence the asymptotic behaviour is determined by the singularities at $1/16n$ and the asymptotic limit of $P_{\leqslant m}(f)$ is exactly the limit of $P_m(f)$ (although the values for finite m differ).

[2] Both the numbers of trees T_m and $t(f,m)$ and the distribution $P_m(f)$ depend on the parameter n, which we do not mention explicitly unless necessary.

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

2.2. Case of a single variable

In this part, we consider what happens when there is a single variable x, and two literals x and $\bar{x}$. There are four functions:

$$f_1 = \text{False}, \qquad f_2 = \bar{x}, \qquad f_3 = x, \qquad f_4 = \text{True}.$$

Let us denote by A_f the set of trees computing the Boolean function f. We have that

$$\begin{aligned} A_{\text{True}} = (\wedge, A_{\text{True}}, A_{\text{True}}) \oplus (\vee, A_x, A_{\bar{x}}) \oplus (\vee, A_{\bar{x}}, A_x) \\ \oplus (\vee, A_{\text{True}}, \mathscr{T}) \oplus (\vee, \mathscr{T}, A_{\text{True}}) \setminus (\vee, A_{\text{True}}, A_{\text{True}}). \end{aligned}$$

The subtraction of the last term comes from the fact that the two preceding terms both contain the trees $(\vee, A_{\text{True}}, A_{\text{True}})$, which we must count only once. We get an equation on the generating functions, where t_f is the enumerating generating function for the trees that compute the Boolean function f, and where T is, as above, the enumerating function for all trees:

$$t_{\text{True}}(z) = 2z t_x(z) t_{\bar{x}}(z) + 2z t_{\text{True}}(z) T(z). \tag{2.2}$$

By symmetry (exchange $\vee$ with $\wedge$ and True with False, in the equation defining A_{True}, to get the equation on A_{False}),

$$t_{\text{False}}(z) = 2z t_x(z) t_{\bar{x}}(z) + 2z t_{\text{False}}(z) T(z). \tag{2.3}$$

Now consider the set of trees that compute the function $f_3 = x$:

$$\begin{aligned} A_x = \{x\} \oplus (\wedge, A_x, A_x) \oplus (\wedge, A_x, A_{\text{True}}) \oplus (\wedge, A_{\text{True}}, A_x) \\ \oplus (\vee, A_x, A_{\text{False}}) \oplus (\vee, A_{\text{False}}, A_x) \oplus (\vee, A_x, A_x) \end{aligned}$$

Hence the equation on the generating functions:

$$t_x(z) = 1 + 2z t_x(z)^2 + 2z t_x(z) t_{\text{True}}(z) + 2z t_x(z) t_{\text{False}}(z). \tag{2.4}$$

By symmetry, we get a similar equation for the function $f_2 = \bar{x}$:

$$t_{\bar{x}}(z) = 1 + 2z t_{\bar{x}}(z)^2 + 2z t_{\bar{x}}(z) t_{\text{True}}(z) + 2z t_{\bar{x}}(z) t_{\text{False}}(z). \tag{2.5}$$

Now we solve this system of four equations in four variables, t_{True}, t_{False}, t_x and $t_{\bar{x}}$, to get first (obviously) that $t_{\text{True}} = t_{\text{False}}$ and $t_x = t_{\bar{x}}$, then that

$$t_{\text{True}}(z) = \frac{2z t_x^2(z)}{1 - 2z T(z)},$$

and finally a polynomial equation on the function $t_x(z)$,

$$1 + 2z y^2 - y + 8 \frac{z^2 y^3}{1 - 2z T(z)} = 0,$$

with $T(z) = (1 - \sqrt{1 - 16z})/4z$. Solving, we get three solutions for this equation:

$$\frac{2}{1 - \sqrt{1 - 16z}}, \qquad \frac{-1}{8z}\left(1 + \sqrt{1 - 16z} \pm \sqrt{2 + 16z + 2\sqrt{1 - 16z}}\right).$$

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

The solution $t_x(z)$ is the (unique) function such that its value at $z = 0$ exists and is equal to 1; hence

$$t_x(z) = \frac{-1}{8z}\Big(1 + \sqrt{1-16z} - \sqrt{2+16z+2\sqrt{1-16z}}\Big).$$

Expanding this expression around the singularity $z_0 = 1/16$, we get

$$t_x(z) = 2(\sqrt{3}-1) + 2\left(\frac{1}{\sqrt{3}} - 1\right)\sqrt{1-16z} + 2\left(\frac{7\sqrt{3}}{9} - 1\right)(1-16z) + O\big((1-16z)^{3/2}\big),$$

and a transfer lemma [4] readily gives

$$t(x,m) = [z^m]t_x(z) \sim 2^{2m+2}\frac{\sqrt{3}-1}{\sqrt{3}}\, C_{m-1}.$$

We finally obtain the asymptotic probability of the function x by dividing the number $t(x,m)$ of trees computing this function by the total number of trees T_m:

$$P_m(x) \sim \frac{\sqrt{3}-1}{\sqrt{3}} \cdot \frac{m+1}{2m-1} \to \frac{\sqrt{3}-1}{2\sqrt{3}} = 0.21132486\ldots.$$

Now the function enumerating the trees that compute the function True is

$$t_{\mathrm{True}}(z) = \frac{1}{8z}\,\Big(2 - \sqrt{2+16z+2\sqrt{1-16z}}\Big),$$

which gives for $P_m(\mathrm{True})$ an asymptotic probability equal to $1/2\sqrt{3} = 0.28867513,\ldots$.

In view of future generalization, let us look again at the initial system of four equations. We can rewrite each equation in a standard form,

$$t_f = 1_{f\ \mathrm{literal}} + z^t F(A_{f,\vee} + B_{f,\wedge})F,$$

where F is the vector $(t_{\mathrm{False}}, t_x, t_{\bar{x}}, t_{\mathrm{True}})$, and the matrices $A_{f,\vee}$ and $B_{f,\wedge}$ are obtained by a process to be described in Section 2.5, and are given below for the functions True and x (the other cases are symmetrical):

$$B_{\mathrm{True},\wedge} = \begin{pmatrix} 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&1 \end{pmatrix}, \qquad A_{\mathrm{True},\vee} = \begin{pmatrix} 0&0&0&1\\ 0&0&1&1\\ 0&1&0&1\\ 1&1&1&1 \end{pmatrix},$$

$$B_{x,\wedge} = \begin{pmatrix} 0&0&0&0\\ 0&1&0&1\\ 0&0&0&0\\ 0&1&0&0 \end{pmatrix}, \qquad A_{x,\vee} = \begin{pmatrix} 0&1&0&0\\ 1&1&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{pmatrix}.$$

2.3. Functions on two variables

We now have two variables x_1 and x_2, four literals and sixteen Boolean functions. Writing down the recurrence relations on the sets A_f and translating them into generating functions gives us a system of sixteen algebraic equations, each of degree two. Symmetries

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

(the generating function for a Boolean function f is equal to the generating function for $\neg f$; the variables x_1 and x_2 can be exchanged; the generating functions for the Boolean functions $l_1 \wedge l_2$ and $l_1 \vee l_2$, where the l_i are the literals on x_1 and x_2, are the same) reduce it to a system of order 4, where a, b, c and d are the generating functions, respectively, for the Boolean functions True, x_1, $x_1 \wedge x_2$ and $x_1 \,\mathsf{xor}\, x_2$:[3]

$$\begin{cases} a = 2zaT + 4zb^2 + 20zc^2 + 2zd^2 + 16zbc + 8zcd, \\ b = 1 + 2zb^2 + 4zc^2 + 4zab + 8zbc, \\ c = 2zb^2 + 8zc^2 + 4zac + 8zbc + 4zbd + 4zcd, \\ d = 4zc^2 + 2zd^2 + 4zad + 8zcd. \end{cases}$$

Furthermore, we have that

$$T = 2a + 4b + 8c + 2d. \tag{2.6}$$

The solution of this system will give us closed-form expressions of the functions, which can be readily expanded around the singularity $z = 1/32$, and finally exact and approximate expressions for the probabilities. We leave the details to the reader and give for example the function $d(z)$:

$$d(z) = \frac{1}{8z}\left(-1 - \tau_0 + 2\tau_1 - \sqrt{\tau_2 + \sqrt{8\tau_3}}\right),$$

with $\tau_0 = \sqrt{1 - 32z}$, $\tau_1 = \sqrt{2 + 2\tau_0}$ and

$$\begin{aligned} \tau_2 &= 2(1 + \tau_0)(2 - \tau_1) - 32z, \\ \tau_3 &= (1 + \tau_0)(5 - 2\tau_1) + 16z(-3 + 2\tau_0 + \tau_1 - \tau_0\,\tau_1) - 128z^2. \end{aligned}$$

Now, if we have an expansion $t_f = \alpha_f - \beta_f\sqrt{1 - z/\rho} + O(z - \rho)$ with $\beta_f > 0$ at the singularity $\rho = 1/32$, then the probability of the function f is (the details are given in Section 2.5)

$$P(f) = \frac{\beta_f}{T(\rho)} = \beta_f/8.$$

Define

$$\begin{aligned} \alpha &= (2\sqrt{2} - 1)^2 = 9 - 4\sqrt{2}, \\ \beta &= -129 + 90\sqrt{2} + 61\sqrt{3} - 38\sqrt{6}, \\ \gamma^2 &= (\sqrt{3} - 1)(2\sqrt{2} + \sqrt{3}) = 3 - 2\sqrt{2} - \sqrt{3} + 2\sqrt{6}, \\ \delta &= \sqrt{6 - 2\sqrt{2} + \sqrt{3} - 2\sqrt{6}} = \frac{1}{\sqrt{2}}(2\sqrt{2} - 1 - \sqrt{3}), \\ \nu &= 153 - 117\sqrt{2} + 61\sqrt{3} - 38\sqrt{6}. \end{aligned}$$

[3] We let $x_1 \,\mathsf{xor}\, x_2$ denote the Boolean function $(x_1 \wedge \bar{x_2}) \vee (\bar{x_1} \wedge x_2)$. This is simply the $+$ operator on $\{0, 1\}$.

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

We obtain the following asymptotic probabilities:

$$P(\text{True}) = \frac{\beta}{6\alpha\gamma\sqrt{2}},$$
$$P(x_1) = 1 - \frac{3}{2\sqrt{2}} + P(\text{True}) - \frac{v}{6\alpha\delta} = P(\text{True}) - \frac{(\sqrt{2}-1)^2}{2\sqrt{2}} - \frac{v}{6\alpha\delta},$$
$$P(x_1 \wedge x_2) = \frac{\sqrt{2}-1}{2} - P(\text{True}) + \frac{v}{12\alpha\delta} = \frac{v}{12\alpha\delta} - P(x_1 \,\mathsf{xor}\, x_2),$$
$$P(x_1 \,\mathsf{xor}\, x_2) = P(\text{True}) - \frac{\sqrt{2}-1}{2}.$$

Floating values are easy to compute:

$$P(\text{True}) = 0.20940201\ldots, \qquad P(x_1) = 0.06717345\ldots,$$
$$P(x_1 \wedge x_2) = 0.03848896\ldots, \qquad P(x_1 \,\mathsf{xor}\, x_2) = 0.00229522\ldots.$$

In other terms, a random Boolean function is one of the constant functions (True or False) almost 42% of the time, a literal 27% of the time, a function of the kind $l_1 \wedge l_2$ in 30% of the cases, and either $x_1 \,\mathsf{xor}\, x_2$ or its negation less than 0.5% of the times.

The average complexity of a random Boolean function under this probability distribution is $2/\sqrt{3} - \sqrt{2} + 1 = 0.740486,\ldots$.

2.4. Functions on three variables

We consider here the case where $n = 3$. There are fourteen different classes of Boolean functions; in each class the same generating function enumerates the binary trees associated to the Boolean functions. Let us denote these fourteen generating functions by the column vector $\mathbf{t}(z) = {}^t(t_1(z), \ldots, t_{14}(z))$. Then $\mathbf{t}(z)$ satisfies a functional equation of the form

$$\mathbf{t}(z) = \mathbf{Q}(\mathbf{t}(z)), \tag{2.7}$$

where each component of the vector-valued function $\mathbf{Q}$ is quadratic in each of the t_i (for details see Section 2.5). By the Drmota–Lalley–Woods theorem (see again Section 2.5) we know that each of the functions t_i admits a representation of the form

$$t_i(z) \sim \alpha_i - \beta_i\sqrt{1 - 48z}, \quad z \to \frac{1}{48}.$$

Thus for $z = 1/48$ the system (2.7) has a unique solution $(\alpha_1, \ldots, \alpha_{14})$. Hence the fixed point can be obtained by iteration, starting from a vector whose coordinates are all equal to zero.

In order to compute the values β_i, $i = 1, \ldots, 14$, observe that Drmota [3] showed that the vector $(\beta_i)_{i=1,\ldots,14}$ is an eigenvector with eigenvalue 1 of the Jacobian

$$\frac{\partial \mathbf{Q}}{\partial \mathbf{t}} = \left(\frac{\partial Q_i}{\partial t_j}\right)_{i,j=1,\ldots,14}$$

evaluated at $z = 1/48$. Since 1 is an eigenvalue of multiplicity 1 at $z = 1/48$, we can easily compute the eigenvector and normalize it to obtain the results presented below.

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

Table 1.

Boolean function	Card.	Compl.	Indiv. prob.	Cumul. prob.
True	2	1	0.165	0.330
l_1	6	0	0.0314	0.188
$l_1 \wedge l_2$	24	1	0.00995	0.239
$l_1 \wedge l_2 \wedge l_3$	16	2	0.00768	0.123
$(l_1 \wedge l_2) \vee l_3$	48	2	0.00211	0.101
$(l_1 \wedge l_3) \vee (\bar{l}_1 \wedge l_2)$	24	3	0.28710^{-3}	0.00689
l_1 xor l_2	6	3	0.19210^{-3}	0.00115
$(l_1$ xor $l_2) \vee l_3$	24	4	0.15710^{-3}	0.00377
$(l_1 \wedge (l_2 \vee l_3)) \vee (l_2 \wedge l_3)$	8	4	0.14910^{-3}	0.00119
$(l_1 \wedge l_2 \wedge l_3) \vee (\bar{l}_1 \wedge \bar{l}_2)$	48	4	0.96210^{-4}	0.00462
$(l_1 \wedge l_2 \wedge l_3) \vee (\bar{l}_1 \wedge \bar{l}_2 \wedge \bar{l}_3)$	8	5	0.56010^{-4}	0.44810^{-3}
$(l_1 \wedge (l_2 \vee l_3)) \vee (\bar{l}_1 \wedge \bar{l}_2 \wedge \bar{l}_3)$	24	5	0.21710^{-4}	0.52110^{-3}
$(l_1 \wedge (l_2$ xor $\bar{l}_3)) \vee (\bar{l}_1 \wedge (l_2 \vee \bar{l}_3))$	16	7	0.27910^{-5}	0.44610^{-4}
$(l_1$ xor $l_2)$ xor l_3	2	9	0.81410^{-7}	0.16310^{-6}

We give, for each class, the generic form of Boolean functions belonging to it ($l_i \in \{x_i, \bar{x}_i\}$ for $i = 1..3$; other functions of the class are obtained by permuting literals or exchanging $\vee$ and $\wedge$), its cardinality (number of Boolean functions), the complexity and probability common to all the functions of the class, and finally the cumulated probability of the class. The classes are in decreasing order of individual probability. The values given in Table 1 were obtained with 30.000 iteration steps, and rounded to three digits. Again, we notice that this model gives a predominant place to constants and to very simple functions: a function has a probability 0.330 to be constant and 0.757 to have complexity 0 or 1; the functions of complexity 2 have a global 0.224 probability; the probability that the complexity is equal to 3 drops to 0.008, and the cumulated probability of functions with complexity 4 or larger is 0.0106. The average complexity of a random Boolean function under this model is equal to 1.08.

2.5. General case

For each Boolean function f on n variables we can write an equation on the set A_f of trees computing it:

$$A_f = 1_{\{f \text{ literal}\}} \oplus \sum_{g,h:f=g\vee h} (\vee, A_g, A_h) \oplus \sum_{g,h:f=g\wedge h} (\wedge, A_g, A_h).$$

This equation on sets of trees translates into an equation on the generating functions enumerating these sets. We obtain

$$t_f(z) = 1_{\{f \text{ literal}\}} + z \sum_{g,h:g\vee h=f} t_g(z)\, t_h(z) + z \sum_{g,h:g\wedge h=f} t_g(z)\, t_h(z). \tag{2.8}$$

The 2^{2^n} Boolean functions on n variables can be defined by their truth table: we associate to each function f a word $(f[1], \ldots, f[p])$ of length $p = 2^n$ over the alphabet $\{0, 1\}$, representing its value for each of the p assignments of values 0 or 1 to the n variables x_i, *i.e.*, a column of the truth table. The alphabetical order of the words also gives an ordering

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

on the functions of the set $\mathscr{F}$ of Boolean functions, which we then denote by $f_1, \ldots, f_{2^p}$:

$$f_1 = (0, \ldots, 0) = \text{False}, \quad f_2 = (0, \ldots, 0, 1) = \bar{x_1} \wedge \bar{x_2} \wedge \cdots \wedge \bar{x_n}, \quad \text{and} \quad \neg f_i = f_{2^p - i}.$$

Now the relations $f = g \vee h$ and $f = g \wedge h$ translate into relations on the corresponding words: for each j, we have that $f[j] = g[j] + h[j]$ (for $\vee$, with $1 + 1 = 1$) or $f[j] = g[j].h[j]$ (for $\wedge$). For example, for $n = 3$, a Boolean function is defined by a word of $\{0,1\}^8$; the function $f = x_1 \vee x_3$ can equivalently be defined by the word $(0,1,0,1,1,1,1,1)$; let $g = (0,0,0,1,0,0,0,1)$ ($g = x_2 \wedge x_3$); then the functions h such that $f = g \vee h$ are all the functions $(0,1,0,*,1,1,1,*)$, where $*$ stands for 0 or 1, which gives us four possible functions.

Define the vector $\mathbf{f} = (t_{f_1}, \ldots, t_{f_{2^p}})$ of the generating functions; furthermore define, for each Boolean function f_k, two matrices

$$A_{f_k} = (a_{i,j}(f_k)), \qquad B_{f_k} = (b_{i,j}(f_k)),$$

with $a_{i,j}(f_k) = 1$ if $f_k = f_i \vee f_j$ and 0 otherwise, and with $b_{i,j}(f_k) = 1$ if $f_k = f_i \wedge f_j$ and 0 otherwise. Then we can write the equation (2.8) as

$$t_f(z) = 1_{\{f \text{ literal}\}} + z^t\mathbf{f}.(A_f + B_f).\mathbf{f} \tag{2.9}$$

$$= 1_{\{f \text{ literal}\}} + z \sum_{1 \leqslant i,j \leqslant 2^p} (a_{i,j} + b_{i,j})\, t_{f_i}(z)\, t_{f_j}(z), \tag{2.10}$$

where ${}^t\mathbf{f}$ is the vector obtained from $\mathbf{f}$ by transposition. Now we have such an equation for each of the 2^p functions f, which gives a system of 2^p algebraic equations on 2^p unknown functions f_i.

Theorem 2.1. *The limiting probability distribution of $P_m(f), m \to +\infty$, exists and can be computed.*

Proof. If we instantiate equation (2.9) for each of the f_i, we obtain a nonlinear polynomial system $\vec{y} = \Phi(\vec{y})$, where each component has nonnegative coefficients, and such that the dependency graph is connected and that the system is a-proper (*i.e.*, a certain Lipschitz condition is satisfied). Then, by results on systems of algebraic equations (see Drmota [3], Lalley [7] or Woods [17]), all component solutions are algebraic with a common singularity, and we can expand the functions around their singularity to get the asymptotic behaviour of the probabilities $P_m(f)$ for large m. We give below the version due to Flajolet and Sedgewick [5, Theorem 8.13, p. 71].

Positive polynomial systems. *Consider a nonlinear polynomial system $\vec{y} = \Phi(\vec{y})$ that is a-proper, a-positive and a-irreducible. In that case, all component solutions y_j have the same radius of convergence $\rho < \infty$. Then there exist functions h_j analytic at the origin such that*

$$y_j = h_j\left(\sqrt{1 - z/\rho}\right) \qquad (z \to \rho^-) \qquad (1 \leqslant j \leqslant 2^p).$$

In addition, all other dominant singularities are of the form $\rho\omega$, with ω a root of unity. If furthermore the system is a-aperiodic, all y_j have ρ as unique dominant singularity. In that

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

case, the coefficients admit a complete asymptotic expansion of the form

$$[z^n]y_j(z) \sim \rho^{-n}\left(\sum_{k\geqslant 1} d_k n^{-1-k/2}\right).$$

In our case, it is easy to check that the system is nonlinear, a-proper, a-positive, a-irreducible and a-aperiodic (the definitions come again from [5, Theorem 8.13, p. 71]). We can then apply the theorem. There exists a solution $(t_{f_1},\ldots,t_{f_{2^p}})$ to the algebraic system. The t_f have a common, strictly positive, radius of convergence ρ and a unique dominant algebraic singularity at $\rho < +\infty$, with an expansion around ρ

$$t_f(z) = \alpha_f - \beta_f\sqrt{1-z/\rho} + O(1-z/\rho), \tag{2.11}$$

which gives by a transfer lemma [4]

$$t(f,m) = [z^m]t_f(z) = -\beta_f[z^m]\sqrt{1-z/\rho}\ \ (1+O(1/m)).$$

The radius of convergence is also the radius for the function $T(z) = \sum_f t_f(z)$; hence $\rho = 1/16n$ and

$$t(f,m) = \beta_f C_{m-1}\rho^{-m}2^{1-2m}(1+O(1/m)).$$

Now we have that $P_m(f) = t(f,m)/T_m$; plugging in the value of T_m from equation (2.1), we get

$$P_m(f) = \frac{\beta_f}{4n}\,(1+O(1/m)),$$

which provides another proof of the existence of the asymptotic distribution P. □

Numerical computation of the asymptotic probabilities? The functions t_f are defined at ρ; plugging the values (2.11) for each f into equations (2.8) and identifying in each equation the coefficient of $\sqrt{1-z/\rho}$, we get a system of size 2^{p+1} on the α_f and β_f, whose generic equations are

$$\begin{cases} \alpha_f = 1_{\{f \text{ literal}\}} + \rho\sum_{g\vee h=f}\alpha_g\,\alpha_h + \rho\sum_{g\wedge h=f}\alpha_g\alpha_h, \\ \beta_f = \rho\sum_{g\vee h=f}(\alpha_g\,\beta_h+\alpha_h\,\beta_g) + \rho\sum_{g\wedge h=f}(\alpha_g\,\beta_h+\alpha_h\,\beta_g). \end{cases}$$

Now this system can be solved numerically with the help of a computer algebra system such as Maple; see the indications given in Section 2.4 for the special case $n = 3$.

2.6. Yet another probability distribution

The terms α_f that appear in the expansion (2.11) can be interpreted to give rise to a different probability distribution π on the space of Boolean functions.

Proposition 2.2. *The probability of computing the function f, when we start from a simple critical branching process (the probabilities that a node has 0 or 2 sons are both equal to 1/2) and label the nodes randomly and independently, to obtain a random* and/or *tree, is*

$$\pi(f) = \frac{t_f(\rho)}{T(\rho)} = \frac{\alpha_f}{4n}.$$

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

Proof. Let τ be a random and/or tree obtained by considering a simple critical Galton–Watson process and labelling at random the nodes. We label the internal nodes by $\wedge$ and $\vee$ with equal probability, and the leaves by the $2n$ literals, again with equal probability. Let $\tilde{\tau}$ be the underlying unlabelled tree; almost surely $\tilde{\tau}$ is finite. Denote by $|\tilde{\tau}|$ or $|\tau|$ its size (total number of nodes), and by $\|\tau\|$ the number of its internal nodes: $|\tau| = 2\|\tau\| + 1$. In this critical Galton–Watson process, the probability that the process stops at one node and the probability that the node has two sons are both equal to $1/2$; hence the probability that we obtain an unlabelled tree $\tilde{\tau}$ of size $|\tilde{\tau}|$ is

$$\mathbb{P}(\tilde{\tau}) = \frac{1}{2^{|\tilde{\tau}|}}.$$

Now the probability of a given labelled tree τ is obtained by multiplying the probability of $\tilde{\tau}$ by the probability of the labelling, which is itself equal to the probability $2^{-\|\tau\|}$ of labelling the internal nodes as in τ, times the probability $(1/2n)^{\|\tau\|+1}$ of labelling the leaves:

$$\mathbb{P}(\tau) = \frac{1}{2^{|\tau|}} \cdot \frac{1}{2^{\|\tau\|}} \cdot \frac{1}{(2n)^{\|\tau\|+1}}.$$

Define $\pi(f) := \sum_{\tau \text{ computes } f} \mathbb{P}(\tau)$; we have that

$$\begin{aligned}
\pi(f) &= \frac{1}{4n} \sum_{\tau \text{ computes } f} \left(\frac{1}{16n}\right)^{\|\tau\|} \\
&= \frac{1}{4n} \sum_{m} t(f,m) \left(\frac{1}{16n}\right)^{m} \\
&= \frac{t_f(1/16n)}{T(1/16n)} = \frac{\alpha_f}{4n}.
\end{aligned}$$

It is interesting to compute the distribution $\{\pi(f)\}$ for small n and to compare it to the distribution $\{P(f)\}$. Numerical data suggest that the distribution $\pi(f)$ is even more strongly biased towards functions of low complexity, mostly literals, than the distribution P, and that the average complexity of a random Boolean function under $\pi(f)$ is less than half its average complexity under P. For example, $n = 1$ leads to

$$\begin{aligned}
\pi(\text{True}) &= \frac{2-\sqrt{3}}{2} = 0.1339745960\ldots, \\
\pi(x) &= \frac{\sqrt{3}-1}{2} = 0.3660254040\ldots.
\end{aligned}$$

The average complexity under this distribution is $2-\sqrt{3} = 0.268$, versus $1/\sqrt{3} = 0.577$ under the distribution P.

For $n = 2$, we obtain (as in Section 2.3, we use $\gamma = \sqrt{3 - 2\sqrt{2} + 2\sqrt{6} - \sqrt{3}}$ and $\delta = (2\sqrt{2} - 1 - \sqrt{3})/\sqrt{2}$):

$$\begin{aligned}
\pi(\text{True}) &= 1 - \frac{\gamma}{2} &&= 0.0864216570\ldots, \\
\pi(x_1) &= \frac{3}{\sqrt{2}} - 1 - \frac{\gamma}{2} - \frac{\delta}{\sqrt{2}} &&= 0.1595538420\ldots,
\end{aligned}$$

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

Table 2.

Boolean function	$P(f)$	$\pi(f)$	Cumul. prob. for $\pi(f)$
True	0.165	0.0642	0.128
l_1	0.0314	0.0994	0.596
$l_1 \wedge l_2$	0.00995	0.00776	0.186
$l_1 \wedge l_2 \wedge l_3$	0.00768	0.00282	0.0451
$(l_1 \wedge l_2) \vee l_3$	0.00211	0.81710^{-3}	0.0392
$(l_1 \wedge l_2) \vee (\bar{l}_1 \wedge l_3)$	0.28710^{-3}	0.88010^{-4}	0.00211
$l_1 \,\mathsf{xor}\, l_2$	0.19210^{-3}	0.67310^{-4}	0.40410^{-3}
$(l_1 \,\mathsf{xor}\, l_2) \vee l_3$	0.15710^{-3}	0.31410^{-4}	0.75410^{-3}
$(l_1 \wedge (l_2 \vee l_3)) \vee (l_2 \wedge l_3)$	0.14910^{-3}	0.32110^{-4}	0.25710^{-3}
$(l_1 \wedge l_2 \wedge l_3) \vee (\bar{l}_1 \wedge \bar{l}_2)$	0.96210^{-4}	0.22010^{-4}	0.00106
$(l_1 \wedge l_2 \wedge l_3) \vee (\bar{l}_1 \wedge \bar{l}_2 \wedge \bar{l}_3)$	0.56010^{-4}	0.99910^{-5}	0.79910^{-4}
$(l_1 \wedge (l_2 \vee l_3)) \vee (\bar{l}_1 \wedge \bar{l}_2 \wedge \bar{l}_3)$	0.21710^{-4}	0.37010^{-5}	0.88810^{-4}
$(l_1 \wedge (l_2 \,\mathsf{xor}\, \bar{l}_3)) \vee (\bar{l}_1 \wedge (l_2 \vee \bar{l}_3))$	0.27910^{-5}	0.35410^{-6}	0.56610^{-5}
$(l_1 \,\mathsf{xor}\, l_2) \,\mathsf{xor}\, l_3$	0.81410^{-7}	0.76710^{-7}	0.15310^{-6}

$$\pi(x_1 \wedge x_2) = \frac{1+\gamma}{2} - \sqrt{2} + \frac{\delta}{2\sqrt{2}} = 0.0234588600\ldots,$$
$$\pi(x_1 \,\mathsf{xor}\, x_2) = \sqrt{2} - \frac{1+\gamma}{2} \qquad = 0.000635219\ldots.$$

The average complexity is now equal to $1 + 2(\sqrt{2} + \sqrt{3}) = 0.364$, to be contrasted with 0.740, the average complexity under $P(\cdot)$.

Finally, the case $n = 3$ gives the results shown in Table 2, where for comparison purposes we give both distributions $P(f)$ and $\pi(f)$, and where the weighted probabilities (for π) are relative to the cumulative values of Boolean functions in the fourteen classes. We see that the distribution $\pi(f)$ leads to literals almost 60% of the time, to functions of complexity 1 in more than 31% cases, of complexity 2 less than 9%, and that functions of complexity 3 or larger are less than 0.5%. Accordingly, the average complexity under this distribution is 0.4998, again less than half the average complexity 1.086 under the distribution $P(\cdot)$.

Comparing the two sets of values $P(f)$ and $\pi(f)$ for $n = 1, \ldots, 3$, we see that, apart from the swapping of literals and constants, the relative order of Boolean functions, sorted in decreasing probability, is the same for the two distributions. Moreover, with the exception of literals, we always have that $\pi(f) < P(f)$. It would be desirable to give a rigorous proof of this fact for all n, which is equivalent to $\alpha_f < \beta_f$, with the exception of literals where $\alpha_l > \beta_l$. □

3. Improving the bounds of Lefmann and Savický

3.1. Lefmann and Savický's model

In [8, Theorem 2.3], Lefmann and Savický obtain a limiting distribution P, which is both the limiting distribution of the uniform probability on finite trees of given size P_m, when m goes to infinity, and the limiting distribution when k goes to infinity, of some probabilities p_k constructed by the machinery of segments. This second representation leads to the following description of the limiting distribution P, as a pruned infinite binary biased tree

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

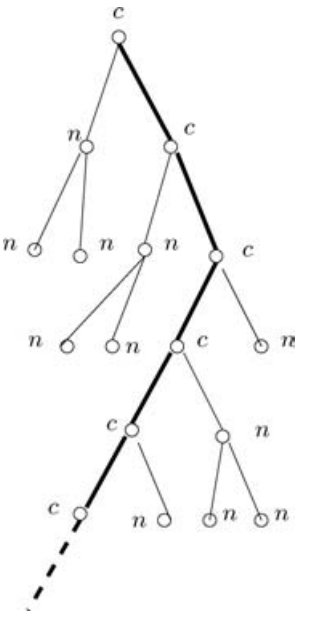


Figure 1.

(the terminology comes from the branching literature; see [9] for instance). Start from a binary biased tree, in which there are two types of nodes: c-nodes and n-nodes. All the nodes on the spine of the biased tree (including the root) are c-nodes which reproduce always with two children, a c-node and a n-node. The nodes that do not belong to the spine are n-nodes, and they split having no descendant with probability $1/2$ and having two n-node children with probability $1/2$ (note that these critical branching subtrees are a.s. finite and are those considered in Section 2.6). See Figure 1.

We then label internal nodes of the tree by $\wedge$ (resp. $\vee$) with probability $1/2$ (resp. $1/2$) and external nodes with literals (each literal occurs with probability $1/2n$). Finally we prune the tree: the idea is to propagate sets of conditions along the branches of the tree, from the root to the leaves; when the set associated to a node becomes inconsistent, the subtree rooted at this node does not need to be evaluated at all.

We then define the pruning as follows. For a given and/or tree, let us mark the internal nodes by conditions, with the following inductive procedure.

- If node v is the root, it is marked by a set ρ_0 of conditions; ρ_0 can be the empty set $\emptyset$, as does occur.
- If node v, marked by a set of conditions ρ, has two internal nodes as children, then ρ is inherited by the children (see Figure 2).
- If node v, marked by a set of conditions ρ, has for children an internal node and an external node containing a literal l, then (see Figure 3):
  - if node v is labelled by an 'and', then the internal child node is marked by $\rho \cup \{l = \text{True}\}$;

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

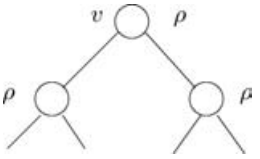


Figure 2.

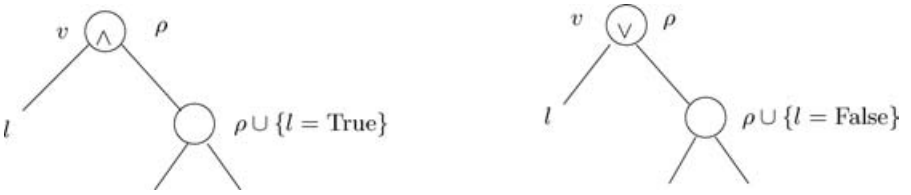


Figure 3.

– if node v is labelled by an 'or', then the internal child node is marked by $\rho \cup \{l = \text{False}\}$.

When applying this procedure to the whole tree, some sets of conditions associated to some nodes may no longer be consistent. In this case, the subtree beginning at such a node does not influence the Boolean function and thus it can be replaced by any constant. By pruning the tree, such a subtree is deleted. It is easy to see that the pruned tree is almost surely finite. See Figure 4.

After pruning, standard simplification rules (*e.g.*, $f \wedge f = f$) can also be applied to get a smaller marked tree. Finally, a tree τ gives a pruned tree $\widehat{\tau}$, then a simplified tree $\widetilde{\tau}$, and these three trees compute the same function $f \in \mathscr{F}$, so that there is an upper bound for the complexity of f:

$$L(f) \leqslant \|\widetilde{\tau}\| \leqslant \|\widehat{\tau}\|.$$

In the following, for simplicity, we denote by τ (instead of $\widetilde{\tau}$) a pruned, simplified tree and let $\|\tau\|$ be the number of internal nodes of the tree τ associated with a consistent set of conditions. The Markov inequality then gives a way of estimating the complexity: for any function f,

$$\begin{aligned} P(f) = P(\text{tree } \tau \text{ computes } f) &\leqslant P((1+\varepsilon)^{\|\tau\|} \geqslant (1+\varepsilon)^{L(f)}) \\ &\leqslant \frac{E[(1+\varepsilon)^{\|\tau\|}]}{(1+\varepsilon)^{L(f)}}. \end{aligned}$$

This inequality yields a nontrivial bound as soon as $1+\varepsilon$ is less than the radius of convergence of the generating function of the size of a tree. That is why, in the following, we are looking for a good evaluation of this radius. We get it by successive approximations, both a truncation at height d, and a set of k conditions at the root.

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

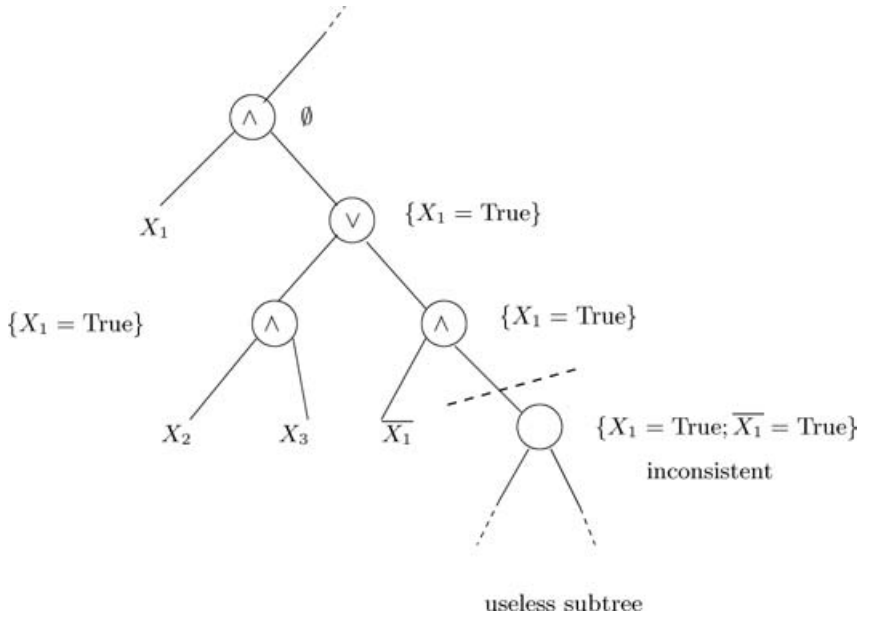


Figure 4.

3.2. Basic relations

For a tree τ and an integer d, let $\|\tau\|_{d,k}$ be the number of internal nodes in tree τ at height at most d, when τ is a tree obtained by pruning according to a set of k conditions at the root. The exact values of the conditions do not matter, only the cardinality of the set of conditions does. Notice that $0 \leqslant k \leqslant n$.

Define the two generating functions

$$F_{d,k}(z) := \mathbb{E}(z^{\|\tau\|_{d,k}} / \text{ the root of } \tau \text{ is an internal } n\text{-node}),$$

which means that the root has two successors, and

$$H_{d,k}(z) := \mathbb{E}(z^{\|\tau\|_{d,k}} / \text{ the root of } \tau \text{ is a } c\text{-node}).$$

We need to investigate these generating functions and their recurrence relations, especially the function H for $k = 0$ and for $d \to +\infty$ to get finally $\mathbb{E}(z^{\|\tau\|})$. We then take $z = 1 + \varepsilon$ for ε as large as possible. From now on, we assume that z is a positive real number.

The following relations are almost obvious, they are summarized in the following lemmas.

Lemma 3.1. *Starting from more conditions at the root gives a smaller pruned tree, so that*

$$\|\tau\|_{d,k+1} \leqslant \|\tau\|_{d,k},$$

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

and consequently, the following inequalities on the generating functions $F_{d,k}$ and $H_{d,k}$ hold for $z \geqslant 1$:

$$0 \leqslant F_{d,k+1}(z) \leqslant F_{d,k}(z), \tag{3.1}$$
$$0 \leqslant H_{d,k+1}(z) \leqslant H_{d,k}(z). \tag{3.2}$$

Lemma 3.2. *Cutting the tree at height $d+1$ gives a larger tree than cutting at height d, that is,*

$$\|\tau\|_{d,k} \leqslant \|\tau\|_{d+1,k}$$

so obviously, for $z \geqslant 1$,

$$F_{d,k}(z) \leqslant F_{d+1,k}(z),$$
$$H_{d,k}(z) \leqslant H_{d+1,k}(z).$$

The following recursions come from the structure of pruning the infinite tree. They are given by Lefmann and Savický [8] in their Lemmas 3.3 and 3.4.

Lemma 3.3.

$$F_{0,k}(z) = H_{0,k}(z) = z,$$
$$F_{d+1,k}(z) = \frac{z}{4}\left(F_{d,k}^2(z) + \frac{k}{n}F_{d,k}(z) + 2\left(1-\frac{k}{n}\right)F_{d,k+1}(z) + \frac{k}{n} + 1\right),$$
$$H_{d+1,k}(z) = \frac{z}{4}\left(2F_{d,k}(z)H_{d,k}(z) + \frac{k}{n}H_{d,k}(z) + 2\left(1-\frac{k}{n}\right)H_{d,k+1}(z) + \frac{k}{n}\right).$$

3.3. Comparing with a branching Galton–Watson process

Let us gather up the threads of the proof of Theorem 1.1: recall that the aim is to get a fine evaluation of the radius of convergence of the generating function $\mathbb{E}(z^{\|\tau\|})$ of the size of a tree τ. For a positive real argument, this generating function is the increasing limit, when $d \to +\infty$, of the generating functions $H_{d,k}$ for trees truncated at height d with $k = 0$ conditions. We are going to control the radius of convergence of $H_{d,k}$ with a uniform (in d) upper bound of the functions $H_{d,k}$ (Lemma 3.5). Because of the intricate relations between $F_{d,k}$ and $H_{d,k}$ (which appear in Lemma 3.3), we need the same kind of uniform (in d) upper bound of the functions $F_{d,k}$ (Lemma 3.4). Conveniently, it comes from comparing the generating function $F_{d,k}$ to the generating function of a Galton–Watson process (depending on k and not on d). Finally the study of the special case $k = 0$ is achieved in Lemma 3.6.

We begin with Lemma 3.3 together with inequality (3.1), which give, for every fixed k,

$$F_{d+1,k}(z) \leqslant \frac{z}{4}\left(F_{d,k}^2(z) + \left(2-\frac{k}{n}\right)F_{d,k}(z) + \frac{k}{n} + 1\right),$$

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

so that, defining

$$\tilde{\varphi}_k(u) := \frac{1}{4}u^2 + \left(\frac{1}{2} - \frac{k}{4n}\right)u + \frac{1}{4} + \frac{k}{4n},$$

it reads

$$F_{d+1,k}(z) \leqslant z\tilde{\varphi}_k(F_{d,k}(z)). \tag{3.3}$$

This is the key equation for comparing our generating function to the generating function of a Galton–Watson one: indeed, $\tilde{\varphi}_k$ is the reproduction function associated to a Galton–Watson process, with generating function G_k satisfying

$$G_k(z) = z\tilde{\varphi}_k(G_k(z)). \tag{3.4}$$

In this branching process,

$$\begin{aligned} p_2 &:= \mathbb{P}(2 \text{ children}) = \frac{1}{4}, \\ p_1 &:= \mathbb{P}(1 \text{ child}) = \frac{1}{2} - \frac{k}{4n}, \\ p_0 &:= \mathbb{P}(0 \text{ child}) = \frac{1}{4} + \frac{k}{4n}. \end{aligned}$$

Let $\alpha := k/n$ and notice that $\alpha \in [0, 1]$. Equation (3.4) is quadratic and can be explicitly solved, giving

$$G_k(z) = \frac{1}{2z}\left[4 - z(2-\alpha) - \sqrt{16 - 8z(2-\alpha) - (8-\alpha)\alpha z^2}\right]. \tag{3.5}$$

This expression allows us to compute explicitly the radius of convergence $\rho(G_k)$ of the generating function G_k (directly or by a derivation standard in branching processes),

$$\rho(G_k) = \frac{-2 + \alpha + 2\sqrt{1+\alpha}}{2\alpha(1 - \alpha/8)},$$

and its expansion around $\alpha = 0$ is

$$\rho(G_k) = 1 + \frac{\alpha^2}{16} + O(\alpha^3) = 1 + \frac{k^2}{16n^2} + O\left(\frac{k^3}{n^3}\right),$$

so that, for any constant $C < 1/16$, for $k > 0$ and n sufficiently large,

$$\rho(G_k) \geqslant 1 + C\frac{k^2}{n^2}.$$

Now, the key equation (3.3) provides a uniform (in d) upper bound for the generating functions $F_{d,k}$, and the following lemma is straightforward.

Lemma 3.4. *For any fixed $k > 0$ and for sufficiently large n, the sequence $(F_{d,k})_d$ increases to a limit F_k when $d \to +\infty$. Moreover, at the limit,*

$$\text{for all } z, \quad 1 \leqslant z \leqslant \rho(G_k), \qquad F_k(z) \leqslant G_k(z).$$

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

This implies that the radius of convergence $\rho(F_k)$ of the generating function F_k, is greater than $\rho(G_k)$, so that for any constant $C < 1/16$,

$$\rho(F_k) \geqslant 1 + C\frac{k^2}{n^2}.$$

Now we use the same kind of comparison for the generating functions $H_{d,k}$. For the same reasons as $F_{d,k}$, Lemma 3.3 with inequality (3.2) gives, for every fixed k,

$$H_{d+1,k} \leqslant \frac{z}{4}\left[2F_{d,k}H_{d,k} + \left(2 - \frac{k}{n}\right)H_{d,k} + \frac{k}{n}\right],$$

and by the previous study of the $F_{d,k}$,

$$H_{d+1,k} \leqslant \frac{z}{4}\left[\left(2G_k + 2 - \frac{k}{n}\right)H_{d,k} + \frac{k}{n}\right].$$

Define a candidate upper bound function H_k by the equation

$$H_k = \frac{z}{4}\left[\left(2G_k + 2 - \frac{k}{n}\right)H_k + \frac{k}{n}\right];$$

the explicit expression (3.5) of $G_k(z)$ gives the following form for H_k,

$$H_k(z) = \frac{\alpha z}{\sqrt{16 - 8(2-\alpha)z - \alpha(8-\alpha)z^2}}, \tag{3.6}$$

and it appears that H_k has the same radius of convergence as G_k:

$$\rho(H_k) = \rho(G_k) \geqslant 1 + C\frac{k^2}{n^2},$$

for any constant $C < 1/16$. An obvious recurrence shows that for every d and k, the functions $H_{d,k}$ are upper-bounded by H_k. By dominated convergence (recall that the sequence $(H_{d,k})$ is increasing in d), the sequence $(H_{d,k})$ converges to a function h_k, when d goes to infinity and $\rho(h_k) \geqslant \rho(H_k)$. We have proved the following.

Lemma 3.5. *For any fixed $k > 0$, and for sufficiently large n, the sequence $(H_{d,k})_d$ increases to a limit h_k when $d \to +\infty$. There exists a function H_k, given by* (3.6)*, which dominates h_k. This implies that the radius of convergence $\rho(h_k)$ of h_k, is greater than $\rho(H_k)$, so that for any constant $C < \frac{1}{16}$, and sufficiently large n,*

$$\rho(h_k) \geqslant 1 + C\frac{k^2}{n^2}.$$

Study for $k = 0$. The final step now comes from the *direct study of $F_{d,0}$ and $H_{d,0}$* whose evolution is given by Lemma 3.3:

$$F_{d+1,0}(z) = \frac{z}{4}\left[F_{d,0}(z)^2 + 2F_{d,1}(z) + 1\right],$$
$$H_{d+1,0}(z) = \frac{z}{4}\left[2H_{d,0}(z)F_{d,0}(z) + 2H_{d,1}(z)\right].$$

As before, we begin with the study of $F_{d,0}$ to deduce the radius of $H_{d,0}$. By Lemma 3.4,

$$F_{d+1,0} \leqslant \frac{z}{4}\left[F_{d,0}(z)^2 + 2G_1(z) + 1\right]$$

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

and we define a fixed point F by the equation

$$F = \frac{z}{4}\left[F^2 + 2G_1 + 1\right]. \tag{3.7}$$

This gives

$$F(z) = \frac{2 - \sqrt{4 - z^2(2G_1(z) + 1)}}{z}.$$

Recall that by (3.5),

$$G_1(z) = \frac{1}{2z}\left[4 - z\left(2 - \frac{1}{n}\right) - \sqrt{16 - 8z\left(2 - \frac{1}{n}\right) - \left(8 - \frac{1}{n}\right)\frac{1}{n}z^2}\right],$$

and

$$\rho(G_1) = 1 + \frac{1}{16n^2} + O\left(\frac{1}{n^3}\right). \tag{3.8}$$

so that, for $1 \leqslant z \leqslant \rho(G_1)$,

$$\sqrt{4 - z^2(2G_1(z) + 1)} = 1 + \frac{1}{n} + O(1 - z),$$

which shows that the radical does not cancel and thus does not introduce a singularity smaller than $\rho(G_1)$.

Consequently, $\rho(F) \geqslant \rho(G_1)$ and by (3.8), $\rho(F) \geqslant 1 + C/n^2$ for any constant $C < 1/16$ and sufficiently large n.

Using the fact that $G_1(z) \geqslant 1$ for $z \geqslant 1$ and equation (3.7), one can easily verify $F_{0,0}(z) = z \leqslant F(z)$ and obtain by the usual recursion that for every $d \geqslant 0$, $F_{d,0} \leqslant F$. Then

$$\rho(F_{d,0}) \geqslant \rho(F) \geqslant 1 + \frac{C}{n^2},$$

for any constant $C < 1/16$.

From the previous uniform upper bound on $F_{d,0}$ and from Lemma 3.5, the functions $H_{d,0}$ satisfy

$$H_{d+1,0} \leqslant \frac{z}{4}\left[2H_{d,0}F + 2H_1\right] = \frac{z}{2}\left[H_{d,0}F + H_1\right],$$

and H_1 is given by formula (3.6). Define a fixed point H by

$$H = \frac{z}{2}\left[HF + H_1\right],$$

so that $\rho(H) \geqslant \rho(H_1) \geqslant 1 + C/n^2$. For every $1 \leqslant z \leqslant \rho(G_1) = \rho(H_1)$, we have $H_1(z) \geqslant 1$ and $zF(z) \geqslant 1$. It follows that $H_{0,0}(z) = z \leqslant H(z)$ and by recursion $H_{d,0} \leqslant H$, for every $d \geqslant 0$. We have proved the following.

Lemma 3.6. *For every $d \geqslant 0$,*

$$\rho(H_{d,0}) \geqslant \rho(H) \geqslant 1 + C\frac{1}{n^2}.$$

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

When d goes to infinity, by Lemma 3.5, we finally obtain that $H_{d,0}$ increases to the generating function of the size of a pruned, simplified and/or tree, and the radius of convergence is as least $1 + C/n^2$, thus giving the theorem.

4. Concluding remarks

We begin by commenting on the quality of the bounds in Theorem 1.1. The lower bound is tight, but the upper bound is not, and can possibly be improved. A natural approach would be to get a better lower bound of the radius of convergence, possibly using results by Nguyên Thê [10]. However, it is possible that Markov's inequality is not strong enough to give a really tight upper bound, and that a different approach may have to be sought.

Numerical computations suggest a point worth mentioning (accordingly for a small number n of variables, but the situation is probably even more marked for larger n). Assuming all trees equally likely gives a very high probability $P(f)$ to functions of complexity 0 or 1 (mainly the constants), and functions of higher complexity quickly become negligible. The alternative distribution $\pi(f)$ behaves in a similar way, and is even more biased towards literals.

We should investigate further the relationship between the probability distributions $P(\cdot)$ (defined by equiprobable and/or trees) and $\pi(f)$ (defined by labelling critical branching processes). We conjecture that, except for literals, we always have that $\pi(f) < P(f)$ for binary planar trees.

A desirable extension of the model for Boolean formulas takes into account the commutativity or associativity of the Boolean operators when representing a function by a tree (*i.e.*, by a Boolean formula). Preliminary investigations show that the natural model becomes that of nonplanar (for commutativity) general (for associativity) trees, for which we can write generating functions using Polya's theory of tree enumeration [12]. Such tree models are also related to those of Woods [17], who proved a general theorem on the existence of a limiting distribution, although no explicit computations were given. This should allow us to prove, in a manner similar to that of Section 2, the existence of a limiting probability distribution, and to compute numerical distributions for small values of n.

It is then natural to try and compare these different distributions, which are all defined on the same set of Boolean functions but stem from different underlying tree models. We might extend the distribution $\pi(f)$ to nonplanar general trees, and examine the conjecture $\pi(f) < P(f)$ in this new context. The relationship between complexity and probability also deserves further investigations. Does a modification of Theorem 1.1 still hold if we substitute different probability distributions for P? What about other complexity measures? We hope to study all these points in a forthcoming paper.

Acknowledgements

We are indebted to the referee for careful reading and accurate suggestions.

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press

References

[1] Billingsley, P. (1995) *Probability and Measure*, 3rd edn, Wiley.
[2] Buhrman, H. and de Wolf, R. (2002) Complexity measures and decision tree complexity: A survey. *Theoret. Comput. Sci.* **288** 21–43.
[3] Drmota, M. (1997) Systems of functional equations. *Random Struct. Alg.* **10** 103–124.
[4] Flajolet, P. and Odlyzko, A. M. (1990) Singularity analysis of generating functions. *SIAM J. Discrete Math.* **3** 216–240.
[5] Flajolet, P. and Sedgewick, R. (2001) Analytic combinatorics: Functional equations, rational and algebraic functions. Technical Report 4103, INRIA.
[6] Friedman, J. (1992) Probabilistic spaces of boolean functions of a given complexity: Generalities and random k-sat coefficients. Technical Report CS-TR-387-92, Princeton University, Princeton, NJ.
[7] Lalley, S. P. (1993) Finite range random walk on free groups and homogeneous trees. *Ann. Probab.* **21** 2087–2130.
[8] Lefmann, H. and Savický, P. (1997) Some typical properties of large and/or boolean formulas. *Random Struct. Alg.* **10** 337–351.
[9] Lyons, R., Pemantle, R. and Peres, Y. (1995) Conceptual proofs of $l \log l$ criteria for mean behavior of branching processes. *Ann. Probab.* **23** 1125–1138.
[10] Nguyên Thê, M. (2002) Distribution of the size of simplified or reduced trees. In *Mathematics and Computer Science II*, Birkhäuser.
[11] Paris, J. B., Vencovská, A. and Wilmers, G. M. (1994) A natural prior probability distribution derived from the propositional calculus. *Ann. Pure Appl. Logic* **70** 243–285.
[12] Pólya, G. and Read, R. C. (1987) *Combinatorial enumeration of Groups, Graphs and Chemical Compounds*, Springer, New York.
[13] Savický, P. (1990) Random boolean formulas representing any boolean function with asymptotically equal probability. *Discrete Math.* **83** 95–103.
[14] Savický, P. (1995) Bent functions and random boolean formulas. *Discrete Math.* **147** 211–234.
[15] Savický, P. (1998) Complexity and probability of some boolean formulas. *Combin. Probab. Comput.* **7** 451–463.
[16] Savický, P. and Woods, A. (1998) The number of boolean functions computed by formulas of a given size. *Random Struct. Alg.* **13** 349–382.
[17] Woods, A. (1997) Coloring rules for finite trees, and probabilities of monadic second order sentences. *Random Struct. Alg.* **10** 453–485.

https://doi.org/10.1017/S0963548304006273 Published online by Cambridge University Press