Random Boolean expressions

Danièle Gardy[†]

PRiSM, Univ. Versailles-Saint Quentin and CNRS UMR 8144. Daniele.Gardy@prism.uvsq.fr

We examine how we can define several probability distributions on the set of Boolean functions on a fixed number of variables, starting from a representation of Boolean expressions by trees. Analytic tools give us a systematic way to prove the existence of probability distributions, the main challenge being the actual computation of the distributions. We finally consider the relations between the probability of a Boolean function and its complexity.

Keywords: Boolean function, complexity, probability distribution for Boolean functions, probability of tautologies

Contents

[†] Part of this work was supported by the contract ACI-NIM No. 20003-63 ACPA.

1 Introduction

Boolean functions are fundamental objects, both in Mathematics and Logic, and in Computer Science, where they appear in various areas: conception of circuits, satisfiability problems with threshold phenomena, constraint resolution, etc.

Mathematically, one of the simplest ways of representing and manipulating Boolean functions may well be Boolean expressions, which are equivalent to tree representations for a suitable set of trees. It is well known that many expressions can represent a single Boolean function, and a natural question is to find an expression of smallest size. ¿From a Computer Science point of view, a major issue is the understanding of the different representations for Boolean functions: binary decision diagrams, circuits, tree representations, and of their properties, one of the most important being their size. The *complexity* of a Boolean function is here relative to a representation, the usual notion of complexity being defined for a circuit (directed graph) representation. The interested reader may refer for example to the book of Wegener [38] for a general introduction to this subject.

In many studies on Boolean functions on a given number of Boolean variables, one assumes that all these Boolean functions have the same probability; see for example Shannon's result [31] about their average complexity. However, consider for example the four functions $True$, $False$, x and $\bar{x}$ on a single Boolean variable x, and the associated Boolean expressions built on this variable, its negation, and the restricted set of connectors $\{\wedge, \vee\}$ (And/Or trees) : A few experiments will suffice to notice that drawing an expression -even quite large- at random will not give all four Boolean functions with equal probability. This suggests that it might be worthwhile to study *the probability distributions induced by expressions (or tree representations) on the set of Boolean functions,* and the relation between the probability of a Boolean function and its complexity. One of the first studies on this subject is a paper by Paris et al. [28], which aims at defining a "natural" probability distribution on Boolean expressions; then the existence of a probability distribution on And/Or trees was established by Lefman and Savicky [20]; further results were by Savicky and Woods [33, 34, 39]; recent works are [2, 3].

The appearance of Boolean functions in Logic, more precisely in propositional calculus, gives a special role to the constant function *True:* Any proof of a formula from specific axioms can be rewritten as a formula that is always true, i.e. a tautology. In other words, the probability of *True* is directly related to the proportion of sentences that are provable in a specific logical system. This approach has been recently illustrated in a series of papers by Zaionc et al., who have begun a systematic study of the density of truth in different propositional systems [17, 18, 26, 24, 41, 42, 43].

In other words, the question: *What is the probability that a random Boolean expression defines a tautology (a Boolean function that is always true)?* has lead us once again to ponder what is a random Boolean expression, and to consider how this definition of randomness translates on the set of Boolean functions

The plan of the paper is as follows. After recalling some basic definitions for Boolean functions and expressions in Section 2, and considering the associated tree representations, we consider in Section 3 the special case of And/Or trees, or equivalently Boolean expressions in which the negation is restricted to Boolean variables and the only other operators, $\wedge$ and $\vee$, are non-commutative and binary. We show on this example how the technics from Analysis of Algorithms can be used to prove the existence of a probability distribution on the set of Boolean functions. We next recall the tools which are basic to our approach (analytic combinatorics, mostly generating functions for the enumeration of various trees, asymptotics, and the Drmota-Lalley-Woods theorem) in Section 4, before making precise how we can

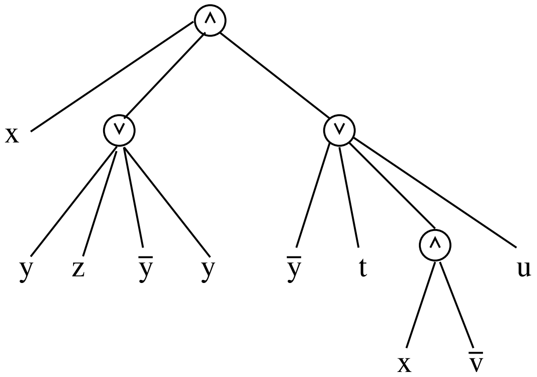


Fig. 1: Planar tree representation of the Boolean expression $E = x \wedge (y \vee z \vee \bar{y} \vee y) \wedge (\bar{y} \vee t \vee (x \wedge \bar{v}) \vee u)$

define probability distributions on the set of Boolean functions in Section 5. We consider commutative or associative operators in Section 6, before turning to tautologies in Section 7. We take a closer look at the relationship between the probability of a Boolean function and its complexity in Section 8, while Section 9 presents some open questions and possible extensions of the results presented in this paper.

2 Boolean expressions and trees representations

2.1 Boolean functions and expressions

We shall consider *functions* on a fixed number n of Boolean variables x_i

$$f : \{0,1\}^n \to \{0,1\} \qquad (or \quad \{True, False\}^n \to \{True, False\})$$

Let $\mathcal{B}_n$ be the set of Boolean functions on n variables; its cardinal is $Card(\mathcal{B}_n) = 2^{2^n}$.

A *Boolean expression* is defined from literals (the x_i or their negations $\bar{x_i}$) and a set of logical operators, for example $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\oplus$, ..., together with rules for building logical expressions : $x_1 \wedge x_2 \wedge \bar{x_3}$, $x_2 \oplus (\bar{x_2} \vee x_3)$, $(\neg(x_1 \vee x_2)) \leftrightarrow (x_1 \wedge x_3)$,...

2.2 Trees for Boolean expressions

We consider first, as an example, a function f defined by the following Boolean expression E on $n = 5$ variables and several formulae, or equivalently tree representations, for it.

$$E = x \wedge (y \vee z \vee \bar{y} \vee y) \wedge (\bar{y} \vee t \vee (x \wedge \bar{v}) \vee u)$$

A representation of f by a plane tree of size (number of leaves) 10 is given in Fig. 1. Assuming that the operators $\wedge$ and $\vee$ are binary gives another representation of expression E (Fig. 2), by an And/Or tree

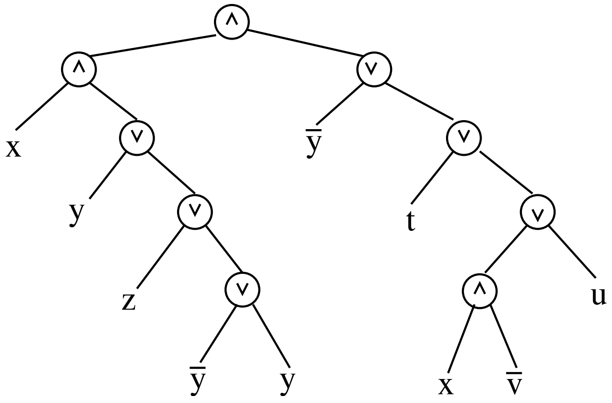


Fig. 2: Binary (And/Or) tree representation of E

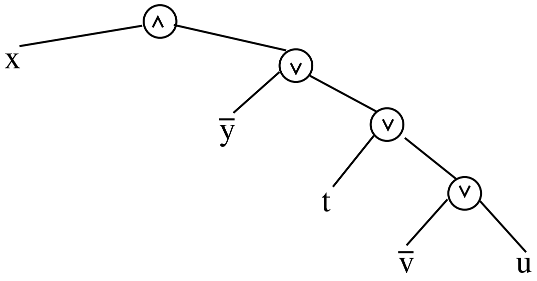


Fig. 3: Minimal tree for E (complexity = 5)

with the same size (again, number of leaves). Finally, an expression equivalent to E and of minimal size 5 is $x \wedge (\bar{y} \vee t \vee \bar{v} \vee u)$, corresponding to the tree of Fig. 3.

The translation from a Boolean expression to a tree follows some simple rules :

- Internal nodes are labelled by logical operators.
- The arity of an internal node is the arity of the logical operator that labels it.
- External nodes (leaves) are labelled by the Boolean variables, or by their negations.
- An associative logical operator labels a node of any arity at least equal to 2; the operators labelling its sons differ from the operator of the father.
- The commutativity of an operator translates on the tree : in such a case, the sons are unordered.

Different sets of rules give different kinds of trees: If all operators are commutative then the tree becomes non-plane; if all operators are binary then the tree is also binary; if the operators are associative then the nodes of the tree can have any arity, except 1.

A Boolean expression represents a single Boolean function but a Boolean function can be represented by an infinite number of Boolean expressions, or equivalently of trees. Hence the definition of the *complexity* of a Boolean function f, as the minimal size $L(f)$ of an expression (a tree) representing f.

When the operators are non-commutative, i.e. when the trees are plane, the set of trees associated to a given class of Boolean expressions is usually called a *simple variety of trees*. Such sets of plane trees were studied for example by Meir and Moon [25]; see also Flajolet and Sedgewick [8, Ch. III.6] for a unifying study.

3 And/Or trees

In this section, we assume that the logical operators are restricted to the non-commutative and non-associative binary operators $\wedge$ and $\vee$, and to the negation on Boolean variables. The underlying tree model is called And/Or trees; see for example Lefman and Savicky [20]. We shall see how this model defines a natural probability distributions on Boolean functions, and how computing this distribution amounts to enumerating sets of trees. We refer the reader to [2] for detailed computations.

Consider Boolean expressions built on the operators $\wedge$, $\vee$ and $\neg$, the Boolean variables and their negations, with the restrictions mentioned above. The associated trees are complete (no single node) binary plane (Catalan) trees; the internal nodes are labelled with $\wedge$ and $\vee$, and the leaves by the literals.

The *size* of a tree is the number of its *leaves*, which is also 1 plus the number of internal nodes, for a complete binary tree.[i]

[i] We might choose another definition for the size, namely the number of internal nodes, or the total number of nodes. However, such definitions of size depend on the specific tree representation (binary or with general arity) for the random Boolean expression E, whereas the number of leaves is the same for a binary tree representation of E or for a tree of unbounded arity. See the example in Section 2.

3.1 A natural probability distribution

Consider the set $\mathcal{T}$ of And/Or trees on n variables: We can partition it in 2^{2^n} subsets, each subset gathering all the trees that compute a specific Boolean function. This partition in turn induces a natural probability distribution on the set $\mathcal{B}_n$ of Boolean functions: the probability of a function f is proportional to the number of trees that compute it.

Well... almost! The set $\mathcal{T}$ of And/Or trees is infinite, and so is the subset $\mathcal{T}_f$ of trees associated to a specific Boolean function f. However, the set $\mathcal{T}_m$ of trees of *fixed* size m is finite, and so is the subset $\mathcal{T}_{f,m}$ associated to f; hence we can, quite naturally, define a probability $P_m(f)$ that a tree of fixed size m computes the Boolean function f, as the ratio of the cardinalities:

$$P_m(f) = \frac{Card(\mathcal{T}_{f,m})}{Card(\mathcal{T}_m)}.$$

The next step is to let m tend to infinity, i.e. to consider the limit $P(f)$ (if it exists) of the family of distributions $P_m(f)$:

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

The first proof of the existence of the limiting distribution P was established by Lefman and Savicky [20], who also gave bounds relating the probability $P(f)$ of a Boolean function and the complexity $L(f)$ of this function. Their technics involved growing biased infinite trees, which were then pruned, and establishing a Chernov bound:

$$\frac{1}{4}\left(\frac{1}{8n}\right)^{L(f)+1} \leq P(f) \leq e^{-cL(f)/n^3}\,(1+o(1)) \tag{1}$$

The upper bound was later improved to $e^{-cL(f)/n^2}$ by Chauvin et al. [2], who presented in the same paper an alternative proof of the existence of P, based on the enumeration of suitable sets of trees. This alternative proof has the advantage of easily generalizing to other types of trees, i.e. to other kinds of Boolean formulas.

It should be obvious that, in defining the distributions P_m, and consequently P, a major step will be the computation of various cardinalities, starting with the enumeration of And/Or trees of fixed size. We shall consider first the enumeration of $\mathcal{T}$ for general n, then turn to explicit enumeration of the subsets $\mathcal{T}_f$ and computation of the probability distribution P on $\mathcal{B}_n$, in the simple case of one Boolean variable ($n = 1$), and finally give indications for the extension to larger n.

3.2 Enumerating And/Or trees

Here we first recall the enumeration of (unlabelled) Catalan trees of size m, then consider all possible labellings to get the exact number of trees and its asymptotic equivalent.

3.2.1 Enumerating binary (Catalan) trees

A binary plane tree [ii] is either a single node, or a root with a left and a right subtree [8, pp. 15-16]:

$$\mathcal{C} = \bullet \oplus (\bullet, \mathcal{C}, \mathcal{C}). \tag{2}$$

[ii] We consider here *complete* binary trees, i.e. trees without simple nodes.

Let c_m be the number of binary trees of size m, and $C(z)$ their generating function:

$$C(z) = \sum_{m \geq 1} c_m z^m.$$

The equation (2) on the set of trees translates readily into an equation on the generating function $C(z)$:

$$C(z) = z + C(z)^2.$$

Solving this quadratic equation gives two solutions; we recall that $C(0)$ is well defined and equal to 0, and obtain

$$C(z) = \frac{1}{2}\left(1 - \sqrt{1-4z}\right).$$

¿From it, we obtain the exact and asymptotic values of the c_m, which are better known as Catalan numbers:

$$c_m = [z^m]C(z) = C_{m-1} = \frac{(2m-2)!}{m!(m-1)!} \sim \frac{4^{m-1}}{m\sqrt{\pi m}}.$$

¿From the algebraic (square-root) singularity of the generating function $C(z)$ at $z = 1/4$, we can also obtain directly the asymptotic expansion of c_m. This approach relies on contour integration of analytic functions and is developed in the *transfert lemmas* of Flajolet and Odlyzko [7]. A general presentation, together with many examples of application in the analysis of algorithms, is given in [8, Ch. VI]; useful pointers can also be found in [10, 13].

3.2.2 Labelling Catalan trees

To obtain an And/Or tree of size m, we choose a random Catalan tree of adequate size, then label its $m-1$ internal nodes by $\wedge$ or $\vee$ and its m leaves by any of the $2n$ literals x_i and $\bar{x}_i$. Each Catalan tree of size m gives thus $2^{m-1}\,(2n)^m = (4n)^m/2$ different And/Or trees, and each such tree can be obtained from a unique Catalan tree. Hence the number of And/Or trees of size m is

$$\frac{1}{2}\,(4n)^m\,C_{m-1}.$$

We can also obtain this result by writing down the generating function $T(z) = \sum_n t_n z^n$ enumerating And/Or trees: such a tree is either one of the $2n$ literals, or a tree starting with a root labelled by $\wedge$ and two subtrees, or a tree with a root labelled by $\vee$ and two subtrees:

$$\mathcal{T} = \{x_1\} \oplus ... \oplus \{x_n\} \oplus \{\bar{x_1}\} \oplus ... \oplus \{\bar{x_n}\} \oplus (\vee, \mathcal{T}, \mathcal{T}) \oplus (\wedge, \mathcal{T}, \mathcal{T}).$$

Hence $T(z)$ satisfies a quadratic equation:

$$T(z) = 2nz + 2T(z)^2.$$

Solving this equation gives

$$T(z) = \frac{1}{4}\left(1 - \sqrt{1-16nz}\right) = \frac{1}{2}\,C(2nz).$$

Taking coefficients, we obtain anew

$$t_n = [z^n]T(z) = \frac{1}{2}\,(4n)^m\,C_{m-1}.$$

3.3 A simple case: n=1

Let $\mathcal{T}$ be the set of And/Or trees on a single Boolean variable; its generating function is

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

and the number of And/Or trees of size m is

$$T_m = 2^{2m+1} C_m. \tag{3}$$

The function $T(z)$ has a unique algebraic singularity at $\rho = 1/16$.

3.3.1 Computing the enumerating series $T_f(z)$

Let $\mathcal{T}_f$ be the set of trees that compute the Boolean function f, for f one of the four Boolean functions on one variable: *True*, *False*, x and $\bar{x}$. A tree of size 1 has a single leaf, which contains either x or $\bar{x}$. A tree of size at least 2 is built from one root, and two subtrees, each of which computes itself a Boolean function. Considering all the possible subtrees and the Boolean functions they compute gives us the following relations:

$$\begin{aligned} \mathcal{T}_{True} &= (\wedge, \mathcal{T}_{True}, \mathcal{T}_{True}) \oplus (\vee, \mathcal{T}_x, \mathcal{T}_{\bar{x}}) \oplus (\vee, \mathcal{T}_{\bar{x}}, \mathcal{T}_x) \\ &\quad \oplus (\vee, \mathcal{T}_{True}, \mathcal{T}) \oplus (\vee, \mathcal{T}, \mathcal{T}_{True}) \setminus (\vee, \mathcal{T}_{True}, \mathcal{T}_{True}); \end{aligned}$$

$$\begin{aligned} \mathcal{T}_x &= \{x\} \oplus (\wedge, \mathcal{T}_x, \mathcal{T}_x) \oplus (\wedge, \mathcal{T}_x, \mathcal{T}_{True}) \oplus (\wedge, \mathcal{T}_{True}, \mathcal{T}_x) \\ &\quad \oplus (\vee, \mathcal{T}_x, \mathcal{T}_{False}) \oplus (\vee, \mathcal{T}_{False}, \mathcal{T}_x) \oplus (\vee, \mathcal{T}_x, \mathcal{T}_x). \end{aligned}$$

By symmetry, similar equations hold for $\mathcal{T}_{False}$ and $\mathcal{T}_{\bar{x}}$, and we check that $T_{\bar{x}} = T_x$ and $T_{False} = T_{True}$. The recurrence equations on the sets of trees translate at once into equations on the generating functions, which satisfy an algebraic system:

$$\begin{cases} T_{True}(z) = 2T_x(z)^2 + 2T_{True}(z)T(z); \\ T_x(z) = z + 2T_x(z)^2 + 4T_x(z)T_{True}(z). \end{cases}$$

Solving, we obtain

$$\begin{aligned} T_{True}(z) &= \frac{1}{8}\left(2 - \sqrt{2 + 16z + 2\sqrt{1-16z}}\right); \\ T_x(z) &= \frac{-1}{8}\left(1 + \sqrt{1-16z} - \sqrt{2 + 16z + 2\sqrt{1-16z}}\right). \end{aligned}$$

3.3.2 Computing the distribution P

The probability that an And/Or tree of size m computes the Boolean function f is

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

where $[z^m]T(z)$ is the number of And/Or trees of size m, and $[z^m]T_f(z)$ is the number of those trees that represent the function f.
We check easily that the functions $T_x(z)$ and $T_{True}(z)$ have the same algebraic singularity $\rho = 1/16$, which is also the singularity of $T(z)$. By a transfert lemma [7]:

$$[z^m]T_x(z) \sim 2^{2m+2}\frac{\sqrt{3}-1}{\sqrt{3}}C_{m-1}.$$

Using the expression of $[z^m]T(z)$ given by (3), we obtain

$$P_m(x) \sim \frac{\sqrt{3}-1}{\sqrt{3}}.\frac{m+1}{2m-1},$$

and it is easy to check that its limit exists for $m \to +\infty$:

$$P_m(x) \to \frac{\sqrt{3}-1}{2\sqrt{3}} = P(x) = 0.2113...$$

A similar computation gives

$$P_m(True) \to \frac{1}{2\sqrt{3}} = P(True) = 0.2886...$$

Recall that the complexity $L(f)$ of the Boolean function f is here the number of leaves in an And/Or tree representation, and that $L(x) = 1$, $L(True) = 2$. The *average complexity* of a random Boolean function, for $n = 1$ and under the distribution P, is

$$\begin{aligned} \mathbb{E}_P[L] &= 2L(x)P(x) + 2L(True)P(True) \\ &= 1 + \frac{1}{\sqrt{3}} = 1.577 \end{aligned}$$

For comparison purposes, we give below the average complexity under the uniform distribution on Boolean functions: $\mathbb{E}_{Unif}[L] = (2L(x) + 2L(True))/4 = 1.5$.

3.3.3 Two definitions of randomness

In the definition of the distribution P, we have considered the set $\mathcal{T}_m$ of trees of fixed size m, then assumed that this size was going to infinity. Working with the uniform distribution on $\mathcal{T}_{\leq m}$, the set of trees of size smaller than or equal to m, gives the same limiting distribution P, although the intermediate distributions P_m and $P_{\leq m}$ do differ.[iii] But what happens if we build the tree by some branching process before labelling its nodes, i.e. if the unlabelled plane binary tree is no longer drawn at random from the set of Catalan trees of fixed size, but if its size itself is random? What if the tree is generated by a branching process?

A critical Galton-Watson branching process and the distribution π induced on $\mathcal{B}_n$.
We refer the reader to [23, 29] for introductory courses and background on branching processes, and

[iii] Technically, working with sizes at most equal to m amounts to multiplying the generating functions by $1/(1-z)$; this introduces a polar singularity at $z = 1$ larger than the dominant singularity; hence the asymptotics are not modified.

consider next how a tree generated in this way might represent a Boolean expression drawn at random. We generate a random binary plane labelled tree, i.e. a random And/Or tree, as follows:

1. We start from the root; at each node the process either stops with probability $1/2$, or gives two sons with probability $1/2$. The tree we obtain is a.s. finite.
2. We label independently each internal node by $\wedge$ and $\vee$ (with uniform probability) and each leaf by one of the $2n$ literals (again with uniform probability).

This gives a new probability distribution $Pr(.)$ on the set $\mathcal{T}$ of *finite* And/Or trees; now the size of a random tree is itself a random variable on $\mathbb{N}$. This distribution in turn induces *another* probability distribution $\pi(.)$ on $\mathcal{B}_n$, which differs from $P(.)$, as can be checked on numerical examples (see below and Section 3.4):

$$\pi(f) = \sum_{\tau \in \mathcal{T};\ \tau\ represents\ f} Pr(\tau).$$

Computation of π

Assuming that the generating functions $T_f(z)$ and $T(z)$ have a common algebraic singularity ρ, the probability $\pi(f)$ of the Boolean function f can be computed from the values of these functions at ρ (see [2] for the proof):

$$\pi(f) = \frac{T_f(\rho)}{T(\rho)}.$$

In our case ($n = 1$), the expressions obtained for the generating functions $T_x(z)$ and $T_{True}(z)$ give readily

$$\begin{aligned} \pi(True) &= \tfrac{T_{True}(1/16)}{T(1/16)} &= \frac{2-\sqrt{3}}{2} = 0.1339... \\ \pi(x) &= \tfrac{T_x(1/16)}{T(1/16)} &= \frac{\sqrt{3}-1}{2} = 0.3660... \end{aligned}$$

The average complexity of a Boolean function under the distribution π is now

$$\begin{aligned} \mathbb{E}_\pi[L] &= 2L(x)\pi(x) + 2L(True)\pi(True) \\ &= 3 - \sqrt{3} = 1.268... \end{aligned}$$

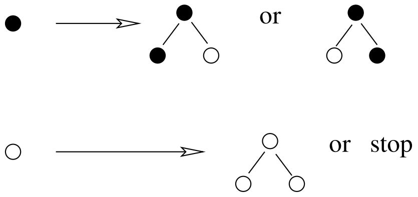
We recall that $\mathbb{E}_P[L] = 1.577$; hence, on $\mathcal{B}_1$: $\mathbb{E}_\pi[L] < \mathbb{E}_{Unif}[L] < \mathbb{E}_P[L]$.

3.3.4 Conditioning on the size, and pruning an infinite tree

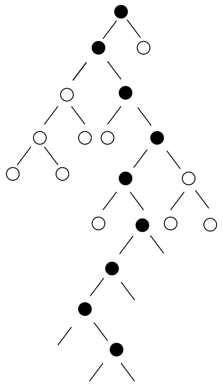
We now have the probability distributions Pr on And/Or trees, the uniform distribution U_m on And/Or trees of fixed size m (which we have not yet used explicitly, but which was implicit in the definition of P_m), and the distributions P_m, P and π on the set $\mathcal{B}_n$ of Boolean functions. How do these distributions relate to each other, and to the distribution on biased trees used by Lefman and Savicky?

The probability distribution Pr on $\mathcal{T}$ induces the image distribution π on the set of Boolean functions. The distribution Pr also induces the *uniform* distribution U_m on the set of And/Or trees of size m, by conditioning on the size of the tree [23]. This uniform distribution on And/Or trees gives the image distribution P_m on Boolean functions, and the sequence P_m has for limit P when $m \to +\infty$.

A biased tree can be viewed as a branching process, with two types of nodes, let's call them black and white, which follow the following branching rules [29]:



The root is always a black node, and the tree becomes infinite, with a single infinite branch of black nodes, on which are grafted subtrees of white nodes; those subtrees follow the distribution Pr on $\mathcal{T}$ defined in Section 3.3.3.



The probability distribution Pr on $\mathcal{T}$ induces a distribution $\hat{Pr}$ on the set of *biased* trees. Lefman and Savicky [20] define *pruning* of an infinite biased tree by the following rules:

- A set of conditions $\mathcal{L}_t$ on the Boolean variables $x_1, ..., x_n$ is associated with each internal node t of the tree.
- The root is assigned the empty set: $\mathcal{L}_{root} = \emptyset$.
- Let t be an internal node labelled with a set of conditions $\mathcal{L}_t$: $t = (operator, l, r)$, If both sons l and r of t are internal nodes, then they are labelled by the same set of conditions as t: $\mathcal{L}_l = \mathcal{L}_r = \mathcal{L}_t$.
- If t is an internal node $t = (\wedge, l, r)$ where l is a leaf labelled by a litteral λ and r is an internal node, then $\mathcal{L}_r = \mathcal{L}_t \cup \{\lambda = True\}$. A symmetric rule holds for an internal node whose left son is an internal node and the right son is a leaf.

- If t is an internal node $t = (\vee, l, r)$ where l is a leaf labelled by a litteral λ and r is an internal node, then $\mathcal{L}_r = \mathcal{L}_t \cup \{\lambda = False\}$; the symmetric rule holds when the only leaf is the right son.
- When an internal node is assigned a contradictory set of rules, it is deleted.

A pruned tree is a.s. finite, see the proof in [20]. Let P_{red} be the probability distribution induced on these pruned trees by $\hat{Pr}$. The main result of Lefman and Savicky is that the image distribution of P_{red} on the set of Boolean functions is precisely the distribution P which was defined as the limit of the distributions P_m induced by the uniform distribution on trees of fixed size m.

The next drawing summarizes the relations between the different distributions on the various sets of trees: Pr on the set of all (a.s. finite) trees, $\hat{Pr}$ on biased trees, P_{red} on (a.s. finite) pruned trees, the uniform distribution U_m on trees of fixed size m, and the distributions π, P_m and P on the set $\mathcal{B}_n$ of Boolean functions [3].

$$\begin{tikzcd}
P_{red} \arrow[ddrr, bend right, "image"'] & \widehat{Pr} \arrow[l, "pruning"'] & Pr \arrow[l, "bias"'] \arrow[r, "size"] \arrow[d, "image"'] & U_m \arrow[d, "image"'] & \text{Trees} \\
 & & \pi & P_m \arrow[dl, bend left, "m \longrightarrow \text{infinity}"] & \\
 & & P & & \text{Boolean functions}
\end{tikzcd}$$

3.4 Extention to larger n

3.4.1 Equations on the generating functions

Enumerating the subsets $\mathcal{T}_f$ of trees associated to each Boolean function f can be done in a systematic way, and allows us first to write down equations on the subsets T_f, then to translate them into algebraic equations on the generating functions $T_f(z)$, for each $f \in \mathcal{B}_n$:

$$\mathcal{T}_f = 1_{\{f\ literal\}} \oplus \sum_{g,h: f=g\vee h} (\vee, \mathcal{T}_g, \mathcal{T}_h) \oplus \sum_{g,h: f=g\wedge h} (\wedge, \mathcal{T}_g, A_h).$$

This gives an equation on the g.f. $T_f(z)$:

$$T_f(z) = z\, 1_{\{f\ literal\}} + \sum_{g,h: g\vee h=f} T_g(z)\, T_h(z) + \sum_{g,h: g\wedge h=f} T_g(z)\, T_h(z).$$

Writing an equation for each of the 2^{2^n} Boolean functions gives a quadratic system of size 2^{2^n} on their generating functions.

3.4.2 The case $n = 2$

For $n = 2$, the algebraic system is of size 16, which is already large for exact solving. Luckily, using symmetry arguments we can reduce the system to an equivalent one of size 4. Taking into account the following operations:

- permutation of Boolean variables,
- negation of a Boolean variable,
- negation of the whole Boolean expression,

allows us to define classes of Boolean functions which share the same generating functions. For $n = 2$, there are only 4 different classes, which correspond to the Boolean functions $True$, x, $x \wedge y$, $x \oplus y$, and we obtain a system of 4 quadratic equations with 4 variables:

$$\begin{cases} T_{True} = 2T_{True}T + 4T_x^2 + 20T_{x\wedge y}^2 + 2T_{x\oplus y}^2 + 16T_xT_{x\wedge y} + 8T_{x\wedge y}T_{x\oplus y}; \\ T_x = z + 2T_x^2 + 4T_{x\wedge y}^2 + 4T_{True}T_x + 8T_xT_{x\wedge y}; \\ T_{x\wedge y} = 2T_x^2 + 8T_{x\wedge y}^2 + 4T_{True}T_{x\wedge y} + 8T_xT_{x\wedge y} + 4T_xT_{x\oplus y} + 4T_{x\wedge y}T_{x\oplus y}; \\ T_{x\oplus y} = 4T_{x\wedge y}^2 + 2T_{x\oplus y}^2 + 4T_{True}d + T_{x\wedge y}T_{x\oplus y}. \end{cases}$$

Recall that $T(z)$ is the global generating function for all trees of $\mathcal{T}$; hence, taking into account symmetries:

$$T(z) = 2T_{True} + 4T_x + 8T_{x\wedge y} + 2T_{x\oplus y}.$$

The common singularity of the generating functions is $\rho = 1/32$. The algebraic system can be solved explicitly; for example the generating function for $True$ is:

$$T_{True}(z) = \frac{1}{8}\left(-2 + \sqrt{\tau_1^2(2+\tau_1) - 32z + \sqrt{\tau_2}}\right),$$

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

$$\tau_2 \quad = \quad 10 - 96z - 256z^2 - 2\tau_1^3 + 32z(\tau_1 + \tau_0(2-\tau_1)).$$

¿From the generating functions, we obtain numerical values (which are the ratio of two numbers defined by algebraic equations, i.e. of two algebraic numbers) for the probabilites of the Boolean functions; for example:

$$P(True) = \frac{-129 + 90\sqrt{2} + 61\sqrt{3} - 38\sqrt{6}}{6\sqrt{2}\,(9-4\sqrt{2})\,\sqrt{\sqrt{3}-1}\,\sqrt{2\sqrt{2}+\sqrt{3}}}.$$

As before, we can also define the probability distribution $\pi(.)$, and compute its exact values (again, we obtain algebraic numbers) from the generating functions. Approximate values are given below for both distributions:

$$\begin{aligned} P(True) &= .209; & \pi(True) &= 0.0864; \\ P(x) &= .0672; & \pi(x) &= 0.159; \\ P(x\wedge y) &= .0385; & \pi(x\wedge y) &= 0.0234; \\ P(x \ \oplus \ y) &= .00229; & \pi(x\oplus y) &= 0.000635. \end{aligned}$$

The average complexities for a Boolean function on 2 variables are

$$\mathbb{E}_{Unif} = 2; \qquad \mathbb{E}_P = 2 + 2/\sqrt{3} - \sqrt{2} = 1.740; \qquad \mathbb{E}_\pi = 2 + 2\sqrt{3} + 2\sqrt{2} = 1.364.$$

3.4.3 $n = 3$ *and beyond*

For $n = 3$, the algebraic system has size 256, and symmetries once again reduce its size: If we consider the symmetries due to permutation of variables, negation of litterals and negation of the whole expression, we obtain 14 classes of Boolean functions, the functions of each class sharing the same generating function. Explicit resolution of the quadratic system (of size 14) has proved beyond our abilities, even with the help of a Computer Algebra System, but the combination of a general theorem, due independently to Drmota, to Lalley and to Woods (see Section 4.2), and of an iteration method allows us to obtain approximate values for the probabilities. We refer the reader to [2] for details.

Boolean Function	$P(f)$	$\pi(f)$
True	0.165	0.0642
l_1	0.0314	0.0994
$l_1 \wedge l_2$	0.00995	0.00776
$l_1 \wedge l_2 \wedge l_3$	0.00768	0.00282
$(l_1 \wedge l_2) \vee l_3$	0.00211	$0.817\ 10^{-3}$
$(l_1 \wedge l_2) \vee (\bar{l}_1 \wedge l_3)$	$0.287\ 10^{-3}$	$0.880\ 10^{-4}$
$l_1\, \mathsf{xor}\, l_2$	$0.192\ 10^{-3}$	$0.673\ 10^{-4}$
$(l_1\, \mathsf{xor}\, l_2) \vee l_3$	$0.157\ 10^{-3}$	$0.314\ 10^{-4}$
$(l_1 \wedge (l_2 \vee l_3)) \vee (l_2 \wedge l_3)$	$0.149\ 10^{-3}$	$0.321\ 10^{-4}$
$(l_1 \wedge l_2 \wedge l_3) \vee (\bar{l}_1 \wedge \bar{l}_2)$	$0.962\ 10^{-4}$	$0.220\ 10^{-4}$
$(l_1 \wedge l_2 \wedge l_3) \vee (\bar{l}_1 \wedge \bar{l}_2 \wedge \bar{l}_3)$	$0.560\ 10^{-4}$	$0.999\ 10^{-5}$
$(l_1 \wedge (l_2 \vee l_3)) \vee (\bar{l}_1 \wedge \bar{l}_2 \wedge \bar{l}_3)$	$0.217\ 10^{-4}$	$0.370\ 10^{-5}$
$(l_1 \wedge (l_2\, \mathsf{xor}\, \bar{l}_3)) \vee (\bar{l}_1 \wedge (l_2 \vee \bar{l}_3))$	$0.279\ 10^{-5}$	$0.354\ 10^{-6}$
$(l_1\, \mathsf{xor}\, l_2)\, \mathsf{xor}\, l_3$	$0.814\ 10^{-7}$	$0.767\ 10^{-7}$

The average complexities of a Boolean function on three variables are

$$\mathbb{E}_{Unif} = 4.336; \qquad \mathbb{E}_P = 2.086; \qquad \mathbb{E}_\pi = 1.499.$$

For $n = 4$, the algebraic system has size $2^{2^4} = 65536$. How many classes of Boolean functions are they? Can we still hope to solve (at least approximately) the system? What about $n \geq 5$?

Results dating back from Harrison [12, 11] give some indications on these questions; e.g. the number of equivalence classes of Boolean functions is given for the first few values of n in the table below.

n	1	2	3	4	5	6
$\lvert\mathcal{B}_n\rvert$	2	16	256	65 536	4.2 10^9	1.8 10^{19}
Nb. classes	2	4	14	222	616 126	2 10^{14}

These results suggest that, although we might still (with substantial work) hope for approximate values when $n = 4$, the extension to larger n is definitely out of our reach and we should turn to other tactics, such as establishing bounds (see for example the bound (1) in Section 3.1 that comes from the initial paper of Lefman and Savicky, and more generally Section 8), to get further information on the probability distributions on $\mathcal{B}_n$.

4 Tools of our trade

Our approach relies on the following idea: Boolean expressions, or equivalently their tree representations, define a language, for which we can write down generating functions; we then have at our disposal all the tools of Analysis of Algorithms [35, 8], and we can (at least in theory) obtain asymptotic expressions for the number of expressions of a given (large) size, then deduce from these asymptotic expressions the limiting probability distributions P and π on $\mathcal{B}_n$.

The main framework of our approach is that of context-free languages: the expressions associated with plane trees define a context-free language; the corresponding enumerating functions are algebraic and are defined by an algebraic system of equations. The existence of a solution for this type of system results from a general theorem, due to Drmota [4], Lalley [19] and Woods [39], which also provides the asymptotic evaluation of the coefficients. This asymptotic evaluation is the next-to final step in the proof in the existence of the desired probability distribution.

In the case of non-plane trees, the functions are no longer algebraic (and the languages are no more context-free), but Polya's theory [30, 27] allows us to write down implicit equations on the generating functions, and an adaptation of the theorem for algebraic systems helps us to prove again the existence of a probability distribution.

4.1 Generating functions for enumeration

We consider here how to obtain generating functions for the enumeration of different classes of Boolean expressions, or in other terms different families of trees. The framework here is symbolic combinatorics, as exposed in the works of Flajolet and Sedgewick, for example in their forthcoming book *Analytic Combinatorics* [8]. It provides a way of systematically translating the recursive definitions of trees into equations on generating functions.

Let $f(z)$ be the generating function for a sequence f_n, where for example f_n is the number of trees of size $|t| = n$ in a specific family $\mathcal{T}$ of trees : $f(z) = \sum_{n\geq 0} f_n z^n = \sum_{t\in\mathcal{T}} z^{|t|}$. A recursive definition of $\mathcal{T}$ translates into an equation on $f(z)$, which can sometimes be solved explicitly, as for binary plane trees. We refer the reader to [8, Ch. I] for a systematic exposition and many examples, including several tree structures.

Once we have an expression for the generating function, we obtain the term $f_n = [z^n]f(z)$ either directly from a catalogue of coefficients, or by way of a *transfert lemma* [7]. See [8, p. 375] for a table of commonly used functions and of their coefficients; for example the binomial formula for $\alpha \in \mathbb{R}$ is

$$[z^m](1+z)^\alpha = \frac{1}{m!}\alpha(\alpha-1)...(\alpha-m+1).$$

The square-root cases $\alpha = \pm\, 1/2$ are specially useful for tree generating functions :

$$[z^m]\sqrt{1-z} = -\frac{C_{m-1}}{2^{2m-1}}; \qquad [z^m]\frac{1}{\sqrt{1-z}} = \frac{m+1}{4^m}\,C_m.$$

A transfert lemma basically asserts that, under suitable regularity conditions, if the function $f(z)$ can be written as $O((1-z)^\alpha)$ near its dominant singularity (wl.o.g. we assume this singularity is at 1), a similar condition holds on the coefficients: $[z^m]f(z) = O([z^m](1-z)^\alpha)$. Hence, if we have an expansion $f(z) = f_1(z) + O((1-z)^\alpha)$ near $z = 1$, and if f_1 belongs to a class of functions for which we know the coefficients, then $[z^m]f(z) = [z^m]f_1(z) + O([z^m](1-z)^\alpha)$. See the book of Flajolet and Sedgewick [8, Ch. VI-VII] for detailed results and many examples.

4.2 The Drmota-Lalley-Woods theorem

In the mid-90s, the study of some problems involving the enumeration of families of plane trees or context-free languages lead independently Drmota [4], Lalley [19] and Woods [39] to establish closely-related results on the asymptotic behaviour of solutions of positive algebraic systems. Our presentation of these results follows the unifying version given by Flajolet and Sedgewick in [8, pp. 446-451].

Theorem
Consider a non-linear polynomial system, defined by a set of equations

$$\{y_j = \Phi_j(z, y_1, ..., y_m)\}, \qquad 1 \leq j \leq m$$

and satisfying the following properties:

1. *a-positivity:* All the terms of the series $\Phi_j(\overrightarrow{y})$ are ≥ 0.
2. *a-proper:* Φ is a contraction, i.e. satisfies a Lipschitz condition ($K < 1$)

$$d(\Phi(y_1, ..., y_m), \Phi(y_1^{'}, ..., y_m^{'})) < \; K \; d((y_1, ..., y_m), (y_1^{'}, ..., y_m^{'}))$$

3. *a-irreductibility:* The *dependency graph* of the algebraic system is built on m vertices 1, 2, ... , m; there is an edge from a vertex k to a vertex j if y_j appears in ϕ_k. The algebraic system is a-irreducible if its dependency graph is strongly connected.
4. *a-aperiodicity:* z (not z^2 or z^3 or...) is the "right" variable
For each ϕ_j, there exist three monomials z^a, z^b and z^c s.t. $b - a$ and $c - a$ are relatively prime

Then

1. All the coordinates y_j of the solution have the same radius of convergence $\rho < \infty$.
2. There exist functions h_j, analytical around 0, s.t. ($1 \leq j \leq m$)

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

3. All the other dominant singularities are of the type $\rho \; \omega$ with ω a root of 1
4. If the system is a-aperiodic, then the y_j have a single dominant singularity ρ, and coefficients have full asymptotic expansion

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

5 Back to the probability distributions on $\mathcal{B}_n$

5.1 Randomness on trees

What we have established for And/Or trees in Section 3 can be mimicked for other kinds of Boolean expressions and trees. We have seen in Section 2 that various sets of operators define various families of trees; most notably, non-commutative operators define simple varieties of trees. Each definition of Boolean expressions, or equivalently each family $\mathcal{F}$ of trees, leads to two distributions on the set $\mathcal{B}_n$: $\pi_{\mathcal{F}}$ and $P_{\mathcal{F}}$.(iv) For $P_{\mathcal{F}}(f)$, we consider trees built at random by drawing uniformly a tree of specified size m, then compute the limiting ratio of the number of those trees that represent f to the total number of trees, letting the size go to infinity. Concurrently, $\pi_{\mathcal{F}}(f)$ is defined from trees built by a random Boltzmann generation process [5, 6], where the size of the tree is itself a random variable. The precise definitions of these two probability distributions are as follows:

1. Denote by $\mathcal{F}_m$ the subset of $\mathcal{F}$, whose trees have the same size m, and by $\mathcal{F}(f)$ (resp. $\mathcal{F}_m(f)$) the set of all trees (resp. trees of size m) that compute the Boolean function f. The probability distribution $P_{\mathcal{F}}$ is the limit of the probability distribution induced on $\mathcal{B}_n$ by the uniform distribution on $\mathcal{F}_m$:
$$P_{\mathcal{F}}(f) = \lim_{m \to +\infty} \frac{Card(\mathcal{F}_m(f))}{Card(\mathcal{F}_m)}.$$

2. The probability distribution $\pi_{\mathcal{F}}$ on $\mathcal{B}_n$ is induced by the distribution obtained on $\mathcal{F}$ by building a tree according to some branching process, for plane trees, or more generally by Boltzmann generation process,(v) then labelling its nodes according to the set of Boolean variables and the set of operators :
$$\pi_{\mathcal{F}}(f) = \sum_{\tau \in \mathcal{F}(f)} Pr_{\mathcal{F}}(\tau),$$
where $Pr_{\mathcal{F}}$ is the distribution on the (infinite) set of trees $\mathcal{F}$ that comes from drawing the trees according to a Boltzmann process.

The extension of the definitions and results of Section 3.3.4 should be straightforward for *plane trees.* As for And/Or trees, the probability distribution Pr on the set $\mathcal{T}$ of trees induces on one hand the image distribution π on the set of Boolean functions, on the other hand the uniform distribution U_m on the set of trees of size m by conditioning on the size of the tree [23]; U_m itself gives the image distribution P_m on Boolean functions, which converges towards the distribution P when $m \to +\infty$. Then a branching process can be used to define biased trees, according to the rules defining the Boolean expressions and the operators under consideration; such biased trees can be seen as an infinite spine on which are grafted finite trees according to a suitable rule, and the probability distribution Pr on (a.s.) finite trees accordingly gives a distribution $\hat{Pr}$ on biased trees. To mimic the approach on And/Or trees, one would then need to define pruning on the biased trees, according to the set of operators under consideration, and to prove that the image distribution of the distribution on biased trees is indeed P. The relations between the

(iv) We use the notations $\pi_{\mathcal{F}}$ and $P_{\mathcal{F}}$ when considering different families of trees, or equivalently different sets of rules for building Boolean expressions; when the context makes clear the family $\mathcal{F}$, we use the simpler notations π and P.

(v) Branching processes cannot be used to define plane trees.

different probability distributions are summarized below; an open question is to complete this diagram as in Section 3.3.4.

$$\begin{tikzcd}
 & \widehat{\mathrm{Pr}} & \mathrm{Pr} \arrow[l, "\text{bias}"'] \arrow[r, "\text{size}"] \arrow[d, "\text{image}"'] & \mathrm{U}_{\mathrm{m}} \arrow[d, "\text{image}"'] \\
\text{Trees} & & \pi & \mathrm{P}_{\mathrm{m}} \arrow[d, "\mathrm{m} \longrightarrow \text{infinity}"'] \\
\text{Boolean functions} & & & \mathrm{P}
\end{tikzcd}$$

For *non-plane trees,* the situation is somewhat different: the definitions and relations for the probability distributions Pr, U_m, P_m P and π are relatively straightforward, but biased trees cannot be defined by branching processes, and pruning of such trees remains an open question.

$$\begin{tikzcd}
 & \mathrm{Pr} \arrow[r, "\text{size}"] \arrow[d, "\text{image}"'] & \mathrm{U}_{\mathrm{m}} \arrow[d, "\text{image}"'] \\
\text{Trees} & \pi & \mathrm{P}_{\mathrm{m}} \arrow[d, "\mathrm{m} \longrightarrow \text{infinity}"'] \\
\text{Boolean functions} & & \mathrm{P}
\end{tikzcd}$$

5.2 Computing the probability distributions P and π

As for And/Or trees, the key to the proof of existence of the distributions $\pi_{\mathcal{F}}$ and $P_{\mathcal{F}}$ in the general case is the existence of a limit for the ratio of trees that compute $\mathcal{F}$, relatively to the total number of trees. The existence of these limits and their computation requires either the exact computation of the values at the dominant singularity, or the asymptotic evaluation of coefficients, of the generating functions

$T_{\mathcal{F}}(z)$, enumerating the family $\mathcal{F}$ by size, and $T_{\mathcal{F}}(f)(z)$, enumerating those trees of $\mathcal{F}$ that compute the Boolean function f. The global enumerating function $T_{\mathcal{F}}(f)$ can usually be computed explicitly, and the functions $T_{\mathcal{F}}(f)$ are defined by an algebraic (or implicit) system; the conditions of the Drmota-Lalley-Woods theorem (for algebraic functions) or of extensions thereof (for implicit functions) are generally easy to check, and one can conclude that there exists a solution $(T_{f_1}, ..., T_{f_{2^p}})$ to the system of equations, and that the T_f have the same common single dominant singularity ρ as the global g.f. $T(z)$.

To sum up, applying analytic combinatorics and singularity analysis to the proof of existence and computation of probabilities for Boolean functions usually leads to the following steps:

- Write down the equations defining the sets of Boolean expressions, or equivalently the set $\mathcal{T}$ of trees under consideration, and the sets $\mathcal{T}_f$ of the trees computing the Boolean functions.

- Translate these equations into (algebraic or implicit) equations on the generating functions $T(z)$ enumerating all trees, and $T_f(z)$ enumerating those trees that compute f.

- Solve the system of equations if possible; if not, apply the Drmota-Lalley-Woods theorem or its extensions, to prove that the functions $T(z)$ and $T_f(z)$ have a common smallest algebraic singularity ρ (square root for most families of trees).

- Compute $T(\rho)$ and the $T_f(\rho)$ and obtain for each Boolean function f

$$\pi(f) = \frac{T_f(\rho)}{T(\rho)}.$$

- Find the expansion of $T(z)$ and of the $T_f(z)$ around ρ:

$$\begin{aligned} T(z) &= T(\rho) - \beta\sqrt{1 - z/\rho} + o(1 - z/\rho); \\ T_f(z) &= T_f(\rho) - \beta_f\sqrt{1 - z/\rho} + o(1 - z/\rho). \end{aligned}$$

- Apply a transfert lemma to get the number of trees of size m:

$$[z^m]T(z) = -\beta\,\rho^{-m}[z^m]\sqrt{1-z}\,(1 + o(1)),$$

  and the number of trees of size m that compute a Boolean function f:

$$[z^m]T_f(z) = -\beta_f\,\rho^{-m}[z^m]\sqrt{1-z}\,(1 + o(1)).$$

- Write down the probability $P(f)$ of the Boolean function f, as the ratio of the number of expressions that compute this function, to the total number of expressions, and conclude to the existence of the probability distribution P:

$$P(f) = \lim_{m \to +\infty} \frac{[z^m]T(z)}{[z^m]T_f(z)} = \frac{\beta_f}{\beta}.$$

6 Commutative or associative operators

We consider in this section the effect of the commutativity and associativity of logical operators on the probability distributions of Boolean functions. As we shall see, this leads to variations on the underlying tree model. The approach by enumeration of trees and analytic combinatorics allows us to treat these variations in a systematic way, although the generating functions become quite involved and the challenge of obtaining numerical results becomes even harder than for And/Or trees. The existence of the limiting probability distribution $P(f)$ was established by Woods in [39] for planar trees of general arity and nodes labelled by $\wedge$ and $\vee$. In the following subsections, we show how we can write down the equations on the generating functions, then try and solve them explicitly for $n = 1$. We refer the reader to [3] for detailed results and proofs.

6.1 Commutativity, and non-plane trees

Non-plane trees appear when we consider *commutative* operators:

- We can define the distribution P exactly in the same way as for And/Or trees, as the limit of the ratio of trees of "large" size that represent a specific Boolean function;
- The distribution π can also be extended, by considering Boltzmann generation of random non-plane trees [6].

We can write down equations on the generating functions T_f associated to each Boolean function f. These equations are no longer algebraic, but can still be solved, or at least give sufficient information on the generating functions for an asymptotic study which allows us to define the probability distributions P and π, and to compute at least a few numerical values.
We examine what happens in the commutative case for binary operators $\wedge$ and $\vee$ in the following subsections, beginning with the case $n = 1$, then considering how we can extend our approach for general n.

6.1.1 Binary non-plane trees

We assume in this part that the logical binary operators $\vee$ and $\wedge$ are now commutative: the underlying binary trees are no longer plane. The generating function $A(z)$ for such trees satisfies a functional equation :

$$A(z) = z + \frac{1}{2}\left(A(z^2) + A(z)^2\right). \tag{4}$$

This equation dates back to Polya [30]; see also the presentation of [8, p. 66-68]. To solve it, we consider it as a quadratic equation on $A(z)$, with a "perturbation" $A(z^2)$. This gives

$$A(z) = 1 - \sqrt{1 - 2z - A(z^2)}. \tag{5}$$

Now substitute $1 - \sqrt{1 - 2z^2 - A(z^4)}$ to $A(z^2)$, and iterate: We get

$$A(z) \quad = \quad 1 - \sqrt{-2z + \sqrt{-2z^2 + ... + \sqrt{1 - 2z^{2^p} - A(z^{2^{p+1}}})}.$$

Equation (5) shows that the function $A(z)$ has an algebraic (square-root) singularity at its radius of convergence ρ, which is defined by the equation $1 - 2z - A(z^2) = 0$. With the equation (4) this simplifies into $A(\rho) = 1$, an equation that can be solved numerically to obtain $\rho = 0.4026975037....$

The asymptotic expansion of the coefficients $[z^n]A(z)$ is obtained by considering the expansion of $A(z)$ for z "close" to the dominant singularity ρ. A transfer lemma gives [8, p. 433]

$$A_m \sim \frac{\lambda}{m\sqrt{m}} \left(\frac{1}{\rho}\right)^m,$$

where the constant can be computed : $\lambda = 0.3187766259...$

6.1.2 Enumerating labelled non-plane trees

The generating function for all trees on n variables satisfies the implicit equation

$$F(z) = 2nz + F(z)^2 + F(z^2).$$

Again, considering it as a perturbed quadratic equation and iterating gives an expression for $F(z)$:

$$\begin{aligned} F(z) &= \frac{1}{2}\left(1 - \sqrt{1 - 8nz - 4F(z^2)}\right) \\ &= \frac{1}{2}\left(1 - \sqrt{-1 - 8nz + 2\sqrt{... + 2\sqrt{1 - 8nz^{2^p} - 4F(z^{2^{p+1}})}}}\right). \end{aligned}$$

6.1.3 The case $n = 1$

In this part, we give explicit formulae for the generating functions and numerical values for the probabilities. The generating function enumerating all trees satisfies the equation

$$F(z) = 2z + F(z)^2 + F(z^2).$$

Solving gives

$$\begin{aligned} F(z) &= \frac{1}{2}\left(1 - \sqrt{-1 - 8z + 2\sqrt{... + 2\sqrt{-1 - 8z^{2^p} - 4F(z^{2^{p+1}})}}}\right) \\ &= 2z + 6z^2 + 24z^3 + 138z^4 + 840z^5 + 5616z^6 + 39168z^7 + 283566z^8 + 2105688z^9 + O(z^{10}). \end{aligned}$$

The singularity is defined by the equation $1 - 8\tau - 4F(\tau^2) = 0$, which can be simplified into $F(\tau) = 1/2$, and gives $\tau = 0.1119665176...$

The equations on the generating functions for True and x, simplified by symmetries, become

$$\begin{aligned} A_{True}(z) &= A_{True}(z)F(z) + A_x(z)^2 + A_{True}(z^2); \\ A_x(z) &= z + (F(z) - A_x(z))A_x(z) + A_x(z^2). \end{aligned}$$

Solving for the generating function A_x gives an expression in terms of the global generating function $F(z)$:

$$A_x(z) = \frac{1}{2}\left(-1 + F(z) + \sqrt{-1 - F^2(z) + 2\sqrt{.... + 2\sqrt{-1 - F^2(z^{2^p}) + 2\sqrt{v(z^{2^{p+1}})}}}}\right).$$

The real positive singularities of $A_x(z)$ are τ and the terms $\tau^{1/2^p}$; the dominant singularity is algebraic at τ. $A(z)$ has an asymptotic expansion near ρ of the form

$$A_x(z) = \alpha_x - \beta_x\sqrt{1-z/\tau} + O(z-\tau),$$

which gives an asymptotic expansion for its coefficients

$$[z^m]A_x(z) \sim \frac{\beta_x}{2m\sqrt{\pi m}}\left(\frac{1}{\tau}\right)^m .$$

The numerical probabilities can be computed explicitly, e.g.

$$P_{non-planar}(x) \quad = \quad \frac{1}{2} - \frac{1}{\sqrt{3-16A_{True}(\tau^2)}}.$$

Numerically:

$$P_{non-planar}(True) = 0.2888; \quad P_{non-planar}(x) = 0.2112;$$
$$\pi_{non-planar}(True) = 0.1344; \quad \pi_{non-planar}(x) = 0.3656.$$

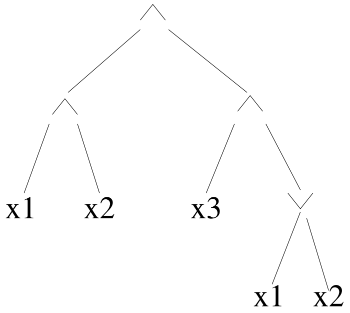
Commutativity has no effect on the number of leaves; hence the average complexities of a random Boolean function are $\mathbb{E}_{Unif}[L] = 1.5$, $\mathbb{E}_P[L] = 1.557$ and $\mathbb{E}_\pi[L] = 1.268$.

6.1.4 Commutativity for general n

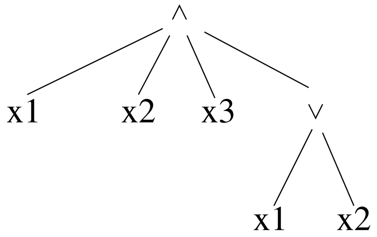
The approach we have used for the special case $n = 1$ can be generalized to any number n of Boolean variables. We first write down a system of 2^{2^n} *functional* equations. To prove the existence of a solution, we have to consider an extension of the Drmota-Lalley-Woods theorem to a non-algebraic system of functional equations [4, 39]. Then we can prove that all the generating functions have the same singularity, and that the distributions $\pi_{non-planar}$ and $P_{non-planar}$ exist.

6.2 Associativity, and trees of general arity

Let's consider the Boolean function $x_1 \wedge x_2 \wedge x_3 \wedge (x_1 \vee x_2)$, represented by the Boolean expression $(x_1 \wedge x_2) \wedge (x_3 \wedge (x_1 \vee x_2))$: The associated And/Or tree is



Assume now that the operator $\wedge$ is associative : The tree representing the expression is no longer binary; it becomes



6.2.1 Tree model for associative operators

We are again dealing with plane trees; here each internal node has at least two sons, but there is no upper bound on the number of sons. These trees are labelled at the root by $\wedge$ or $\vee$, then at successive levels by alternating $\wedge$ and $\vee$ for internal nodes, and at the leaves by literals.
As usual, two probability distributions can be defined on $\mathcal{B}_n$ from this set of trees:

- Assume all trees of size m are equally likely, then let $m \to +\infty$. Let P_{assoc} be the limiting distribution.
- Consider a critical branching process (we are dealing with plane trees) where every node can have any number of sons, except maybe 1, then label the (a.s. finite) trees we obtain. The distribution of the number of sons follows a geometric distribution; conditioning by the size of the tree gives the uniform distribution on the set of trees of fixed size n [23]. Let π_{assoc} be the corresponding distribution.

6.2.2 The case $n = 1$

The set $\mathcal{S}$ of plane trees such that all internal nodes have arity at least 2, satisfies the recurrence equation

$$\mathcal{S} = \bullet \oplus (\bullet, \mathcal{S}, \mathcal{S}) \oplus (\bullet, \mathcal{S}, \mathcal{S}, \mathcal{S}) \oplus ...$$

which gives on the generating function for all (unlabelled) trees

$$S(z) = z + \sum_{l \geq 2} S^l(z) = z + \frac{S^2(z)}{1 - S(z)};$$

hence

$$S(z) \quad = \quad \frac{1}{2}\left(1 + z - \sqrt{1 - 6z + z^2}\right).$$

This is the generating function for Schröder numbers, with a singularity at $\rho = 3 - 2\sqrt{2} = 0.171572876....$ We have [8, p. 76, p. 430]

$$S_m = [z^m]S(z) = \frac{1}{n}\sum_{k=0}^{n-2}\binom{n-2}{k}\binom{2n-k-2}{n-1} \sim \frac{1}{4n\sqrt{\pi n}}\,(3 - 2\sqrt{2})^{-n+1/2}.$$

Consider now these trees, where the root is labelled by $\wedge$ or $\vee$, and the leaves are labelled by x or $\bar{x}$. Their generating function is

$$T(z) = 2S(2z) - 2z = \frac{1}{2}\left(1 - 2z - \sqrt{1 - 12z + 4z^2}\right).$$

Hence the number of trees of size m representing Boolean expressions with associative operators $\wedge$ and $\vee$ is

$$[z^m]T(z) = 2[z^m]S(2z) = 2^{m+1}S_m.$$

To write down the equations on the trees that represent a Boolean function f, we have to consider the operator at the root of the tree. This leads us to consider *two* generating functions: $T_f^{\wedge}$ and $T_f^{\vee}$, enumerating the trees that compute f according to the label of their root.
Symmetry again allows us to consider only the generating functions for $True$ and x:

$$\begin{aligned} T_{True} &= T_{True}^{\wedge} + T_{True}^{\vee}; \\ T_x &= T_x^{\wedge} + T_x^{\vee}. \end{aligned}$$

The functions $T_x^{\wedge}$, $T_x^{\vee}$, $T_{True}^{\wedge}$, $T_{True}^{\vee}$ satisfy

$$\begin{aligned} T_x^{\wedge} &= \frac{1}{1-(z+T_x^{\vee}+T_{True}^{\vee})} - \frac{1}{1-T_{True}^{\vee}} - (z+T_x^{\vee}); \\ T_x^{\vee} &= \frac{1}{1-(z+T_x^{\wedge}+T_{True}^{\vee})} - \frac{1}{1-T_{True}^{\vee}} - (z+T_x^{\wedge}). \\ T_{True}^{\wedge} &= \frac{(T_{True}^{\vee})^2}{1-T_{True}^{\vee}}; \\ T_{True}^{\vee} &= \frac{1}{1-(2z+T^{\wedge})} - T_{True}^{\wedge} - \frac{1}{1-(z+T_x^{\wedge}+T_{True}^{\vee})} \\ &\quad - \frac{1}{1-(z+T_x^{\vee}+T_{True}^{\vee})} + \frac{1}{1-T_{True}^{\vee}}. \end{aligned}$$

This system is algebraic, and can be solved exactly. For example

$$f_x^{\wedge} = \frac{-5-39z+14z^2+(-7+z)\lambda+\lambda\sqrt{Q1+Q_2\lambda}}{16(3-z)},$$

with $\lambda = \sqrt{1-12z+4z^2}$ and

$$\begin{aligned} Q_1 &= 514-2744z-3048z^2+8768z^3-4000z^4+128z^5+128z^6; \\ Q_2 &= 510-340z-2096z^2+1696z^3-160z^4-64z^5. \end{aligned}$$

The common algebraic singularity for all the generating functions appears at $\rho = (3-2\sqrt{2})/2$. Asymptotic expansion around ρ gives the probabilities for $True$ and x:

$$\begin{aligned} P_{assoc}(True) &= 4\sqrt{2} - \frac{23}{4} + \frac{555-733\sqrt{2}}{904}\sqrt{22-4\sqrt{2}} = .235; \\ P_{assoc}(x) &= \frac{25}{4} - 4\sqrt{2} + \frac{733\sqrt{2}-555}{904}\sqrt{22-4\sqrt{2}} = .265; \\ \pi_{assoc}(True) &= \frac{3}{2} - \frac{\sqrt{2}}{4} - \frac{1}{4}\sqrt{22-4\sqrt{2}} = .136; \\ \pi_{assoc}(x) &= \frac{\sqrt{2}}{4} - 1 + \frac{1}{4}\sqrt{22-4\sqrt{2}} = .364. \end{aligned}$$

The number of leaves is the same for a binary tree and for the associated tree of general arity: associativity does not modify the complexity of a Boolean function. The average complexities are here $\mathbb{E}_{Unif}[L] = 1.5$, $\mathbb{E}_P[L] = 1.470$ and $\mathbb{E}_\pi[L] = 1.272$.

6.2.3 Associativity for general n

We can write down a system of 2^{2^n+1} algebraic equations: two equations for each Boolean function, according to the label at the root of the tree. We obtain an algebraic system of size 2^{2^n+1}. The next step is to apply the Drmota-Lalley-Woods theorem, and deduce from it, once again, first the existence of a common algebraic singularity ρ for all the generating functions, then the existence of the distributions π_{assoc} and P_{assoc}. As the equivalence classes for Boolean functions are still valid, we have hope to solve the algebraic system and to obtain numerical results for $n = 2, 3$.

6.3 Some challenging questions

We first summarize the numerical results we were able to obtain for $n = 1$.

	Binary tree	General arity (associativity)
Planar tree	$P_{planar}(x)$ = 0.21132... $P_{planar}(True)$ = 0.2886...	$P_{assoc}(x)$ = 0.2649... $P_{assoc}(True)$ = 0.2350...
	$\pi_{planar}(x)$ = 0.3660... $\pi_{planar}(True)$ = 0.1339...	$\pi_{assoc}(x)$ = 0.3642... $\pi_{assoc}(True)$ = 0.1358...
Non planar tree (commutativity)	$P_{non-planar}(x)$ = 0.21119... $P_{non-planar}(True)$ = 0.2888...	
	$\pi_{non-planar}(x)$ = 0.3656... $\pi_{non-planar}(True)$ = 0.1343...	

These results ought to be completed at least for $n = 1$ or $n = 2$; see [3] for further numerical results. In their present state, they suggest the following questions and extensions :

- Commutativity seems to have a negligible effect on the probability distributions P and π. Can we prove it?
- Associativity seems to have a greater effect on the P distributions than on the π distributions. Can we quantify it? Can we prove that, on the opposite, the effect of commutativity on the π distributions is minimal?
- We should be able to write down the general system for operators that are both commutative *and* associative (corresponding to *reduced* expressions [34]), and to compute the numerical values of $\pi_{comm-assoc}$ and $P_{comm-assoc}$ for $n \leq 3$.

7 The probability of tautologies

Among Boolean functions, tautologies have a special status. One of the first papers to try and define a “natural” probability distribution on the set of Boolean functions was written in 1994 by Paris et al. [28], and centered on the probability of a tautology. Then Zaionc et al. began a systematic study of different logical systems, again giving a special place to tautologies. We present below, first the logical systems studied up to now and exact results, which unfortunately seem to be available for only a small number n of

Boolean variables, then another approach, which aims at obtaining bounds in a systematic way when exact results are unattainable. These bounds have been established only for And/Or trees, but our approach can be extended to other families of trees and expressions, and should allow us to obtain further results.

7.1 Propositional logic: the different systems

A few years ago, a group of polish researchers in logic began a systematic study on the probability of a tautology (or in their terms the density of truth) in several logical systems, with the aim of comparing versions of propositional logic together and with intuitionistic logic.

The simplest logical system is the propositional calculus with a single connector of implication. The results are presented in the papers [26, 41] by Moczurad, Tyszkiewicz and Zaionc for the version without negation, and in the paper [42] of Zaionc for the extension to expressions built with the connectors of implication and negation. In our framework, the logics with a single connector (either implication or equivalence; see the results of Matecki [24] below) correspond to the classical Catalan trees, the logic of implication and negation corresponds to unary-binary trees,

The probability that an expression, built on a single Boolean variable ($n = 1$) and the connector $\rightarrow$ (implication) but without negation, is a tautology is $P(True) = 1/2 + \sqrt{5}/10 \sim 0.7236$; for n variables $P(True)$ belongs to the interval $[(4n+1)/(2n+1)^2, (3n+1)/(n+1)^2]$. Explicit results are also given for expressions built on the negation and implication connectors on a single Boolean variable: $P(True) = 0,4232...$, $P(False) = 0,1632...$, $P(x) = 0,2154...$ and $P(\bar{x}) = 0,1980....$ It is worth mentioning that, in such a system, the probability of a Boolean function f differs from that of its negation $\bar{f}$.

Other papers of Zaionc, again for the logic of implication, consider a subset of tautologies: *simple* tautologies [40, 43]. When the single connector is the implication $\rightarrow$, Boolean expressions can be written in an equivalent form as $(\tau_1 \wedge \tau_2 \wedge .... \wedge \tau_p) \rightarrow \alpha$: the τ_i are called the premises, and the number of premises is simply the length of the right-most path in the underlying binary tree. Expressions for which $\exists i : \tau_i = \alpha$ are obvious tautologies; they are called *simple* tautologies. Of course, $\{$Simple Tautologies$\} \subset \{$Tautologies$\}$, and Zaionc proves that

$$P(Simple\ \ tautologies) = \frac{4n+1}{(2n+1)^2} < P(True).$$

The paper also examines what happens when one takes into consideration the number p of premises, and finally investigates the ratio of simple tautologies, when considering expressions with p premises (and again for fixed n): when p grows to infinity, almost all the expressions with p premises become tautologies.

The case of the single connector $\leftrightarrow$ (equivalence), without negations either on the expressions or on the Boolean variables, is studied by Matecki in [24]. Constraints due to the equivalence operator require that the size of an expression representing a tautology is always even, and the probability $P(True)$ that a Boolean expression of even size is a tautology, is asymptotically equal to $1/2^{n-1}$. Again the underlying structure is that of Catalan trees.

Kostrzycka and Zaionc [17, 18] extend the results on propositional logic to intuitionistic logic, in the case of a single Boolean variable ($n = 1$) and two operators $\rightarrow$ and $\neg$ ((implication and negation). The focus here is no longer strictly on the probability of $True$, but rather on the relative probability of provable sentences in the intuitionistic logic versus tautologies of classical logic. Therefore the papers [17, 18] answered, for a single variable, the question: How big is an intutionistic fragment of the classical logic?

Kostrzycka investigates also other non classical logics, including modal and Grzegorczyck logic, in order to compute and compare the asymptotic density of provable and true sentences for $n = 1$ [14, 15, 16].

7.2 Lower bounds for And/Or *trees*

Consider the probabilities $P_{\mathsf{And/Or}}$ and $\pi_{\mathsf{And/Or}}$ of a tautology for And/Or trees. For small n, we have explicit results (cf. Section 3) :

- For $n = 1$, $P_{\mathsf{And/Or}}(True) = 0.288$ and $\pi_{\mathsf{And/Or}}(True) = 0.134$;
- For $n = 2$, $P_{\mathsf{And/Or}}(True) = 0.209$ and $\pi_{\mathsf{And/Or}}(True) = 0.086$;
- For $n = 3$, $P_{\mathsf{And/Or}}(True) = 0.165$ and $\pi_{\mathsf{And/Or}}(True) = 0.064$.

For general n, we have seen that we cannot compute the generating functions and the probabilities associated to *any* Boolean function f. But can we obtain some results for a *specific* function, namely *True*? The results presented below come from [9], where one can find detailed proofs and extensions.

The set of And/Or trees $\mathcal{T}_{True}$ that represent tautologies can be written as

$$\begin{aligned}
\mathcal{T}_{True} \quad = \quad & \oplus_{f \neq True, False}\{(\vee, \mathcal{T}_{True}, \mathcal{T}_f) \oplus (\vee, \mathcal{T}_f, \mathcal{T}_{True})\} \\
& \oplus_{f,g \neq True, False; f \vee g = True}(\vee, \mathcal{T}_f, \mathcal{T}_g) \\
& \oplus(\vee, \mathcal{T}_{False}, \mathcal{T}_{True)} \oplus (\vee, \mathcal{T}_{True}, \mathcal{T}_{False)} \\
& \oplus(\vee, \mathcal{T}_{True}, \mathcal{T}_{True}) \oplus (\wedge, \mathcal{T}_{True}, \mathcal{T}_{True})
\end{aligned}$$

This translates on the generating functions as

$$T_{True} = \sum_{f,g \neq True, False; f \vee g = True} T_f T_g + 2T \, . \, T_{True},$$

where $T(z) = (1 - \sqrt{1 - 16nz})/4$ is the global generating function for And/Or trees. Unfortunately, this equation cannot be solved exactly. So let's try a simpler equation!

To this effect, we shall try to find a subset of the trees that represent *True*, simple enough that the (in)equation on generating functions can be solved explicitly and give numerical results, but large enough that this numerical results still provide meaningful information.

Define a *subset* $\mathcal{E}_{True} \subset \mathcal{T}_{True}$ by

$$\begin{aligned}
\mathcal{E}_{True} \quad = \quad & \oplus_{l\ literal}(\vee, l, \bar{l}) \\
& \oplus_{f \neq True}\{\oplus(\vee, \mathcal{E}_{True}, \mathcal{T}_f) \oplus (\vee, \mathcal{T}_f, \mathcal{E}_{True})\} \\
& \oplus(\vee, \mathcal{E}_{True}, \mathcal{E}_{True}) \oplus (\wedge, \mathcal{E}_{True}, \mathcal{E}_{True}).
\end{aligned}$$

Let ϕ_{True} be the generating function for $\mathcal{E}_{True}$:

$$\begin{aligned}
\phi_{True}(z) \quad &= \quad \sum_{l\ \ literal} (g.f.\ for\ l)^2 + 2T(z)\ \phi_{True}(z) \\
&= \quad 2nz^2 + 2T(z)\ .\phi_{True}(z).
\end{aligned}$$

Hence $\phi_{True}(z) = 2nz^2/(1 - 2T) = zT(z)$. We have $\mathcal{E}_{True} \subset \mathcal{T}_{True}$; then

- $[z^m]T_{True}(z) \geq [z^m]\phi_{True}(z)$ and, asymptotically:

$$P(f) = \lim_{m \to \infty} \frac{[z^m]T_{True}(z)}{[z^m]T(z)} \geq \lim_{m \to \infty} \frac{[z^m]\phi_{True}(z)}{[z^m]T(z)}.$$

- $T_{True}(x) \geq \phi_{True}(x)$ for any real positive x, including the algebraic singularity ρ, hence

$$\pi(f) = \frac{T_{True}(\rho)}{T(\rho)} \geq \frac{\phi_{True}(\rho)}{T(\rho)};$$

We get readily a common lower bound $1/16n$ for both $P(True)$ and $\pi(True)$.

7.3 Probability distributions for tautologies

We summarize below the few results already obtained for a number n of Boolean variables and the probability of $True$:

- For the uniform distribution on $\mathcal{B}_n$ [31]:

$$Prob(True) = 1/2^{2^n}.$$

- With a single equivalence connector [24]:

$$P(True) \sim \frac{1}{2^n}.$$

- With a single implication connector and variables without negations [26, 43]:

$$\frac{1}{n}(1 + o(1)) \leq P(True) \leq \frac{3}{n}(1 + o(1))$$

- With the connectors $\wedge$ and $\vee$ and negation on the variables [9]:

$$P(True) \geq \frac{1}{16n}; \qquad \pi(True) \geq \frac{1}{16n}.$$

Thus, tautologies have widely varying probabilities. The smallest known probability (up to now) appears when all the Boolean functions have the same probability, and the probability of tautologies is at least exponentially larger, or even doubly exponentially larger, when we consider the probability distributions defined from various Boolean expressions. It would certainly be worthwhile to characterize the logical systems and sets of operators according to the probability of tautologies they induce on the set of Boolean functions.

8 Bounds, complexity and probabilities

We now turn to closer examination of the relations between the complexity of a Boolean function and its probability. We have seen that computing the probability of a function often amounts to knowing the number of trees that represent this function; a dual approach was investigated in [34] by Savicky and Woods, who consider the number of Boolean functions of given (low) complexity under the model of And/Or trees. They show that the number of distinct Boolean functions, of complexity at most l, is polynomial in the number of Boolean variables: it is equal to $b(n,l)^l$, with $b(n,l) \sim cn$ for large n, with c an explicit constant. This results holds as soon as $l \leq 2^n/n^{\beta(n)}$, where $\beta(n) \to +\infty$ for large n, i.e. far from the maximal complexity (which is of order $2^n/\log n$); it is valid, among others, for functions of polynomial complexity.

We present in the following subsections, first an extension of the approach that allowed us to obtain bounds on the probability of a tautology, then a comparison of the two probability distributions P and π for the specific case of And/Or trees, and finally some simulation results that suggest further extensions.

8.1 And/Or *trees, and lower bounds*

We have seen in Section 3.4 that numerical values, even for quite small values of n, are hard to obtain exactly with our tools. Can we get lower bounds for *any* Boolean function f, as we did in Section 7 for tautologies?

First results were obtained by Lefmann and Savicky [20], who proved by a probabilistic technic that, for And/Or trees:

$$P(f) \geq \frac{1}{4}\left(\frac{1}{8n}\right)^{L(f)+1}.$$

Let us sketch here an alternative approach, using our favorite tools, i.e. generating functions [9]. Mimicking what we did for tautologies, we shall define a subset $\mathcal{E}_f$ of the set $\mathcal{T}_f$ of the And/Or trees that represent f.

Assume we know $\mathcal{M}_f$, the set of *minimal* trees that represent f, $L(f)$ (the size of those minimal trees) and $Card(\mathcal{M}_f) = k(f)$. Define $\mathcal{E}_f \subset \mathcal{T}_f$ by

$$\begin{aligned} \mathcal{E}_f &= \mathcal{M}_f \oplus (\wedge, \mathcal{E}_f, \mathcal{E}_f) \oplus (\vee, \mathcal{E}_f, \mathcal{E}_f) \oplus (\vee, \mathcal{E}_f, \mathcal{E}_{False}) \\ &\quad \oplus (\vee, \mathcal{E}_{False}, \mathcal{E}_f) \oplus (\wedge, \mathcal{E}_{True}, \mathcal{E}_f) \oplus (\wedge, \mathcal{E}_f, \mathcal{E}_{True}). \end{aligned}$$

The generating function ϕ_f of $\mathcal{E}_f$ satisfies the equation

$$\phi_f = k(f) z^{L(f)} + 2\phi_f^2 + 4\phi_f \phi_{True}.$$

This equation can be solved explicitly:

$$\phi_f(z) = \frac{1}{4}\left(1 - 4\phi_{True}(z) - \sqrt{1 - 4\phi_{True}(z) - 8k(f)\, z^{L(f)}}\right).$$

The generating function ϕ_f has a square-root singularity ρ:

$$\phi_f(z) = \alpha_f - \beta_f \sqrt{1 - z/\rho} + O(1 - z/\rho).$$

Hence

$$\pi(f) \geq 4\alpha_f; \qquad P(f) \geq 4\beta_f.$$

The constants α_f and β_f can be written as (simple) functions of $k(f)$ and $L(f)$. Asymptotically:

$$\pi(f) \geq \frac{4k(f)}{(16n)^{L(f)}}; \qquad P(f) \geq \frac{4k(f)}{(16n)^{L(f)+1}}.$$

How close to the exact values are these lower bounds? We give below numerical results for tautologies and for literals, in some cases for which we also know the exact values of the probabilities, i.e. for $n \leq 3$.

For tautologies, the bound of Lefman and Savicky is $P(f) \geq 1/(2048n^3)$, and the bound we have obtained in Section 7, and which is valid both for P and for π, is $1/(16n)$. Numerically,

	$\pi(True)$	Lower bound	$P(True)$
$n = 1$	0.134	0.0625	0.288
$n = 2$	0.086	0.0312	0.209
$n = 3$	0.064	0.0156	0.165

On these few results, the bound on $\pi(True)$ seems closer to the actual value than the bound on $P(True)$.

For literals, Lefmann and Savický's bound gives $P(x) \geq 1/(256n^2)$; our approach with $k(x) = L(x) = 1$ gives again an improvment: $P(x) \geq 1/(64n^2)$; $\pi(x) \geq 1/(4n)$.

	$\pi(x)$	Lower bound	$P(x)$	Lower bound
$n = 1$	0.366	0.322	0.211	0.0327
$n = 2$	0.159	0.139	0.067	0.0052
$n = 3$	0.099	0.089	0.031	0.0021

Again, the bound on $\pi(x)$ is not too bad, but the bound on $P(x)$ is far away from the actual value.

8.2 Comparing the distributions $P_{\mathcal{F}}$ et $\pi_{\mathcal{F}}$

The numerical results obtained for And/Or trees in Section 3 show that, at least for small n, "usually" $\pi_{\mathsf{And/Or}}(f) < P_{\mathsf{And/Or}}(f)$, *except* for literals. But the situation becomes more complex when n grows large: literals are no more the only functions which have a larger probability under π than under P. It is possible to prove that, for a read-once function f, i.e. a function that can be represented by an expression where no variable appears twice, $P(f) < \pi(f)$ when n is "large enough" [9, Th. 8]. This is to be contrasted to the result for tautologies: for all n, $P(True) > \pi(True)$ [9, Th. 7]. At this point, we know of no further results relative to the comparison of $P_{\mathsf{And/Or}}(f)$ and $\pi_{\mathsf{And/Or}}(f)$, for functions which are neither constants nor read-once functions.

An intuitive feeling of these results may come from the following ideas: The probability distribution $\pi_{\mathsf{And/Or}}$ takes into account trees of any finite size, and should be biased towards those Boolean functions that can be represented with the "smallest" trees, i.e. which have smallest complexity, such as read-once functions, while the probability distribution $P_{\mathsf{And/Or}}$ is less biased, being defined from the distribution of "very large" trees. If this heuristic holds, then we should expect similar results for other families $\mathcal{F}$ of trees. Anyway, it should be definitely worthwhile to study different families of trees, as well as subsets of Boolean functions larger than read-once functions.

8.3 Simulation results

When do the first k levels of the tree determine the Boolean function? Simulation results [37] indicate that experimentally, the first levels of tree give a good indication on the Boolean function it computes. For example, we considered two "heads" (shape and labelling of the first levels) $H_1 = (t_1 \vee t_2) \wedge (t_3 \vee t_4)$ and $H_2 = (t_1 \wedge t_2) \wedge (t_3 \wedge t_4)$, and computed experimentally the conditional probabilities $P(f/H_i)$ obtained by simulation on trees with either of these heads, for two Boolean variables: $n = 2$. We give below the probabilities obtained by simulation; for comparison purposes we also recall the probabilities computed in the unbiased case (cf. Section 3). In the table below, the probabilities are cumulated on classes of functions with equal probability, and the sum of the probabilities left in blank for the last line is less than 0.0002.

Boolean function	True	False	l	$l_1 \vee l_2$	$l_1 \wedge l_2$	$l_1 \oplus l_2$
Unbiased prob. $P(f)$	0.209	0.209	0.269	0.154	0.154	0.005
Cond. prob. $P(f/H_1)$	0.115	0.04	0.37	0.27	0.17	0.035
Cond. prob. $P(f/H_2)$		0.78	< 0.03		0.19	

We see that the probabilities vary widely, according to the head. Conditionning by H_1 multiplies roughly by 7 the probability of the exclusive or (the function $l_1 \oplus l_2$) and decreases the probabilities of the constants, while the probabilities of literals and of conjunctions or disjunctions, although modified, stay roughly in the same range of values. With the head H_2, things become quite contrasted: the probability of the constant function *False* dominates all other probabilities; the probability that the Boolean function is a conjunction of two literals (or a single literal) is still significant, but the probabilities of disjunctions or of exclusive or become negligible. Such results suggest the following questions on the probability distributions, when conditioned by the head of the tree:

- For a specific Boolean function f, can we define heads such that the trees with these heads compute f with "high" probability?
- If we consider all trees with a fixed head H, what of the *conditional* probability distributions $\pi_{(}./H)$ and $P(./H)$?

9 Extensions and open questions

The maximum complexity for a Boolean function on n variables is 2^n. Under the uniform distribution on the set $\mathcal{B}_n$ of Boolean functions on n variables, "almost all" functions have complexity $2^n/\log n$, close to the maximal complexity; see the classical results of [31, 21, 22] and the simple counting argument for trees in [8, p. 74]. Of course, the uniform distribution is not the only one that can be defined on $\mathcal{B}_n$; in fact we have shown in this paper that it is possible to give a precise definition for two distributions $P_{\mathcal{F}}$ and $\pi_{\mathcal{F}}$, for *any* family $\mathcal{F}$ of trees that correspond to a valid set of rules for Boolean expressions.

At this point, we should mention here another approach, aiming in a similar vein at the definition of a limiting probability distributions on $\mathcal{B}_n$ but growing trees according to a specific set of rules, that leads to widely different distributions. This approach, initiated by Valiant [36], then by Savicky [32, 33], and generalized recently by Brosky and Pippenger [1], considers trees built recursively from trees of smaller height chosen, not at random among all possible trees, but among all trees of equal height. Starting from leaves, such a regularity condition ensures that the trees are well balanced, with all the leaves at the same level, and (for binary trees) all the internal nodes present. A probability distribution is defined for each family of trees of given height h, and it is possible to show that this sequence of distributions has a limit for $h \to +\infty$. The technics involve a discrete Fourier transform and the fixed point method. Up to now, the limiting distributions obtained from such trees are quite different from ours: they are either uniform (or oscillate between two uniform distributions on complementary subsets of $\mathcal{B}_n$), or a Dirac distribution that gives the weight 1 to a threshold function, or a uniform distribution on slice functions.[vi]

Pondering the properties of specific probability distributions and the relationships between different distributions, a wide range of questions appears:

- What if the logical operators (or the Boolean variables) have non-uniform probabilities? Our framework gives us the existence of the limiting distributions, but can we compute them numerically? compare them with those for uniform operators?

- What of the conditional probability distribution defined by any specific head?

- What is the influence of the family $\mathcal{F}$ of trees, i.e. of the set of rules for building Boolean expressions? Can we compare the probabilities $P_{\mathcal{F}}$ (resp. $\pi_{\mathcal{F}}$) for different families $\mathcal{F}$?

- What of the relation between complexity and probability? Can we obtain bounds such as the one established by Lefmann and Savicky, or the lower bounds presented in Section 8, for any probability distribution? Can we improve on these bounds, which are obviously not tight on the numerical examples we computed?

- What is the probability of a function with low complexity, such as a read-once function?

- We know the probability of tautologies for a few type of trees; what of other types of Boolean expressions? Can we classify the different sets of rules for Boolean expressions, according to the probability of a tautology (doubly exponential, exponential, of order $1/n$,...)?

- What becomes the average complexity of a random Boolean function under a specific probability distribution when the number of variables becomes large? Does it remain equal to $2^n/\log n$, as for the uniform distribution, is it still of exponential order α^n for some $\alpha < 2$, or can it even become sub-exponential?

Acknowledgements

Thanks to B. Chauvin, P. Flajolet, B. Gittenberger and A. Woods for their various collaborations in establishing most of the results presented in this paper, to R. David, P. Lescanne and M. Zaionc for suggesting an invited talk at the Chambéry colloquium *Computational Logic and Applications* and the stimulating discussions that followed, to D. Villa-Monteiro and F. Quessette for fruitful simulations.

[vi] A threshold function takes the value 1 for all arguments of weight greater than or equal to some m, and 0 for all arguments of weight smaller than m. A slice function takes the value 1 for all arguments of weight greater than m, 0 for all arguments of weight smaller than m, and may take either value for arguments of weight equal to m.

References

[1] A. Brodsky and N. Pippenger. The Boolean functions computed by random Boolean formulas, or, how to grow the right function. *Random Structures and Algorithms*, 27(4):490–519, 2005.

[2] B. Chauvin, P. Flajolet, D. Gardy, and B. Gittenberger. And/Or trees revisited. *Combinatorics, Probability and Computing*, 13(4-5):475–497, July-September 2004.

[3] B. Chauvin, D. Gardy, and A. Woods. Commutative and associative tree representations for Boolean functions. Technical report, U.V.S.Q., 2006. To appear.

[4] M. Drmota. Systems of functional equations. *Random Structures and Algorithms*, 10:103–124, 1997.

[5] P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Combinatorics, Probability, and Computing*, 13(4-5):577–625, 2004. Special issue on the Analysis of Algorithms.

[6] P. Flajolet, E. Fusy, and C. Pivoteau. Boltzmann sampling of unlabelled structures. Technical report, INRIA, 2006. Submitted for publication. Available on the web at the site http://algo.inria.fr/flajolet/Publications/.

[7] P. Flajolet and A. M. Odlyzko. Singularity analysis of generating functions. *SIAM J. on Discrete Math.*, 3(2):216–240, 1990.

[8] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. 2006. Available on the web at the site http://algo.inria.fr/flajolet/Publications/books.html.

[9] D. Gardy and A. Woods. Lower bounds on probabilities for Boolean functions. In C. Martinez, editor, *First International Conference on the Analysis of Algorithms*, pages 139–146, Barcelona (Spain), June 2005. DMTCS Proceedings AD, Discrete Mathematics and Theoretical Computer Science.

[10] D.H. Greene and D.E. Knuth. *Mathematics for the analysis of algorithms*. Birkhäuser, Boston, 1981.

[11] M. A. Harrison. The number of classes of Boolean functions under groups containing negation. *I.E.E.E. Trans. Elect. Comput.*, 12:559–561, 1963.

[12] M. A. Harrison. Counting theorems and their applications to classification of switching functions. In Amar Mukhopadhyay, editor, *Recent developments in switching theory*, pages 85–120. Academic Press, 1971.

[13] P. Henrici. *Applied and computational analysis, Vol. 2.* John Wiley, New York, 1974.

[14] Z. Kostrzycka. On the density of truth of implicational parts of intuitionistic and classical logics. *J. of Applied Non-Classical Logics*, 13(2), 2003.

[15] Z. Kostrzycka. On the density of truth in Grzegorczyck's modal logic. *Bulletin of the Section of Logic*, 33(2):107–120, June 2004.

[16] Z. Kostrzycka. On the density of truth in modal logics. In *Mathematics and Computer Science*, Nancy (France), September 2006. Proceedings to appear in DMTCS.

[17] Z. Kostrzycka and M. Zaionc. On the density of truth in Dummett's logic. *Bulletin of the section of logic*, 32/1:43–55, 2003.

[18] Z. Kostrzycka and M. Zaionc. Statistics of intuitionnistic versus classical logic. *Studia Logica*, 76(3):307–328, 2004.

[19] S.P. Lalley. Finite range random walk on free groups and homogeneous trees. *Ann. Probab.*, 21(4):2087–2130, 1993.

[20] H. Lefmann and P. Savický. Some typical properties of large And/Or Boolean formulas. *Random Structures and Algorithms*, 10:337–351, 1997.

[21] O. B. Lupanov. Complexity of formula realization of functions of logical algebra. *Problemy Kibernetiki, 3:61-80, 1960. English translation: Problems of Cybernetics, Pergamon Press*, 3:782–811, 1962.

[22] O. B. Lupanov. On the realization of functions of logical algebra by formulae of finite classes (formulae of limited depth) in the basis $\wedge$, $\vee$, $\neg$. *Problemy Kibernetiki, 6:5-14, 1961. English translation: Problems of Cybernetics, Pergamon Press*, 6:1–14, 1965.

[23] J.-F. Marckert. Mouvement brownien et arbre continu. Notes for a graduate course. Available on the web at the site http://www.labri.fr/perso/marckert/papers.html, 2003.

[24] G. Matecki. Asymptotic density for equivalence. *Electronic Notes in Theoretical Computer Science*, 140:81–91, 2005.

[25] A. Meir and J. W. Moon. On the altitude of nodes in random trees. *Canadian Journal of Mathematics*, 30:997–1015, 1978.

[26] M. Moczurad, J. Tyszkiewicz, and M. Zaionc. Statistical properties of simple types. *Mathematical Structures in Computer Science*, 10(5):575–594, 2000.

[27] R. Otter. The number of trees. *Annals of Mathematics*, 49(3):583–599, 1948.

[28] J. B. Paris, A. Vencovská, and G. M. Wilmers. A natural prior probability distribution derived from the propositional calculus. *Annals of Pure and Applied Logic*, 70:243–285, 1994.

[29] Y. Peres. *Probability on trees: an introductory climb.* Springer Lecture Notes in Mathematics 1717, pp. 193-280, 1999. Notes from the St-Flour (France) Summer School 1997.

[30] G. Pólya and R.C. Read. *Combinatorial enumeration of Groups, Graphs and Chemical Compounds*. Springer Verlag, New York, 1987.

[31] J. Riordan and C. E. Shannon. The number of two terminal series-parallel networks. *Journal of Math. and Physics*, 21:83–93, 1942.

[32] P. Savickỳ. Random Boolean formulas representing any Boolean function with asymptotically equal probability. *Discrete Mathematics*, 83:95–103, 1990.

[33] P. Savický. Complexity and probability of some Boolean formulas. *Combinatorics, Probability and Computing*, 7:451–463, 1998.

[34] P. Savický and A. Woods. The number of Boolean functions computed by formulas of a given size. *Random Structures and Algorithms*, 13:349–382, 1998.

[35] R. Sedgewick and P. Flajolet. *An introduction to the Analysis of Algorithms*. Addison Wesley, 1996.

[36] L. Valiant. Short monotone formulae for the majority function. *Journal of Algorithms*, 5:363–366, 1984.

[37] D. Villa-Monteiro. Expressions Booléennes aléatoires. Technical report, Master Thesis, University of Versailles-St Quentin, 2005.

[38] I. Wegener. *The complexity of Boolean functions*. Wiley and sons, 1987.

[39] A. Woods. Coloring rules for finite trees, and probabilities of monadic second order sentences. *Random Structures and Algorithms*, 10:453–485, 1997.

[40] M. Zaionc. On the density of truth in logic of implication and negation. Technical report, 2002.

[41] M. Zaionc. Statistics of implicational logic. *Electronic Notes in Theoretical Computer Science*, 84, 2003.

[42] M. Zaionc. On the asymptotic density of tautologies in logic of implication and negation. *Reports on Mathematical Logic*, 39:67–87, 2005.

[43] M. Zaionc. Probability distribution for simple tautologies. *Theoretical Computer Science*, 355(2):243–260, 2006.