Complexity and Limiting Ratio of Boolean Functions over Implication*

Hervé Fournier[1], Danièle Gardy[1], Antoine Genitrini[1], and Bernhard Gittenberger[2]

[1] Laboratoire PRiSM
CNRS UMR 8144 and Université de Versailles St-Quentin en Yvelines
45 av. des États-Unis, 78035 Versailles, France
{herve.fournier, daniele.gardy, antoine.genitrini}@prism.uvsq.fr
[2] Technische Universität Wien
Wiedner Hauptstrasse 8-10/104, A-1040 Wien, Austria
gittenberger@dmg.tuwien.ac.at

Abstract. We consider the logical system of boolean expressions built on the single connector of implication and on positive literals. Assuming all expressions of a given size to be equally likely, we prove that we can define a probability distribution on the set of boolean functions expressible in this system. We then show how to approximate the probability of a function f when the number of variables grows to infinity, and that this asymptotic probability has a simple expression in terms of the complexity of f. We also prove that most expressions computing any given function in this system are "simple", in a sense that we make precise.

Keywords: Boolean functions; Implicational formulas; Complexity; Limiting ratio; Probability distribution; Analytic combinatorics

1 Introduction

Write at random a boolean expression on given sets of boolean variables and of connectors: we obtain a boolean function. How random is this boolean function? E.g., what is the probability that we obtain a tautology? A literal? Any specified function? Is the probability of obtaining a given function related to the complexity of the function? Does the Shannon effect, i.e. the fact that "almost all" functions have maximal complexity, still hold for this probability distribution? These and some others are questions that we would like to investigate for general logic systems.

We present here a first step in this direction, with an in-depth study of the simple system obtained from the single connector of implication and positive literals. Our interest for this system stems from its relation to intuitionistic logic [1,2] and from its relative simplicity, although not all boolean functions can be obtained in this system: The set of functions that can be obtained in this system is the Post class S_0, i.e. the set of functions that we can write as $x \vee g$ for suitable boolean variable x and function g.

Consider the ratio of the number of formulas of size n that compute a fixed boolean function f, among all formulas of size n, and let the size grow to infinity. It is possible to show that the limit of this ratio exists for a wide variety of logical systems [3], and that we can thus define a probability distribution on the set of boolean functions.

Some of us have shown in a former paper [4] that the tautologies in the implication system have the simple shape $(..., a, ...) \to a$ with high probability, and that, if the number k of boolean variables grows large enough, the probability of a tautology is asymptotically $1/k$. The next natural step is then to try and compute the probability that a random expression computes a literal, a

* This research was partially supported by the A.N.R. project *SADA* and by the P.H.C. Amadeus project *Probabilities and tree representations for boolean functions*. The last author's work has been supported by *ÖAD, project Amadée, grant 3/2006*, as well as by *FWF (Austrian Science Foundation), National Research Area S9600, grant S9604*.

function $x_i \vee x_j$, etc., and to check if the "average" expression computing, e.g., a literal, has a simple form. When studying the random expressions that compute a given boolean function f, one major parameter is the complexity $L(f)$, i.e. the size of the smallest expressions that represent f. We shall prove in the present paper that the probability of any given function f depends exponentially on its complexity; in passing we are also able to characterize the shape of a random expression computing f, and to show that these expressions are obtained quite simply from minimal trees.

The efforts to define non-uniform probability distributions, induced by random boolean expressions, or formulae, on the set of boolean functions, date back several years. The starting point is generally the description of formulae as *trees* of a suitable shape and suitably labelled. The first efforts in this direction were by Paris et al. [5] on And/Or trees (i.e. expressions built on the two connectors $\wedge$ and $\vee$); the underlying model was that of binary Catalan trees, suitably labelled. The study of these trees was further pursued by Lefman and Savický [6], who proved by a pruning argument the existence of a probability distribution induced by random expressions, and established important lower and upper bounds for the probability of any boolean function in terms of its complexity. At the same time, Woods [7] proved independently the existence of a limiting distribution for general formulae. Some of the authors of the present paper then gave an alternative construction of the probability distribution for And/Or trees, together with an improvement on the upper bound [8]. The survey paper [3] presents an overview of the probability distributions induced by random boolean expressions on boolean functions, and of the way we can obtain them using the tools of analytic combinatorics: enumeration of formulae/trees by generating functions, the Drmota-Lalley-Woods theorem for solving an algebraic system of equations and asymptotics.

We should also mention that several researchers have concentrated on the probability of tautologies, i.e. on the probability of the single constant function *True*. Let us mention the Polish school around Zaionc, who began a systematic investigation of the probability of a tautology in various logical frameworks [9,10,11,12]; see also [13,14] for the expressions built on the single equivalence connector. For And/Or trees, we refer the reader to Woods's result that the tautologies have asymptotic probability $3/4k$, and that almost all of them have the simple form $l \vee \bar{l} \vee ...$ [15], and to Kozik [16] for a different, later proof.

Significant results have also been established for a different family of formulae/trees, namely balanced trees obtained by iteration of a single connector. The first result in this area is due to Valiant [17], whose aim was to compute a boolean expression for the function *Majority* with high enough probability. Then Boppana [18] and Gupta-Mahajan [19] improved Valiant's result for majority; Boppana went on to prove that iteration by a single, well-chosen connector gives a uniform distribution on the set of threshold functions. Savický [20] showed that iterating a non-linear balanced connector leads to the uniform distribution on the set of all boolean functions. Finally, Brodsky and Pippenger [21] present a systematic study of different classes of connectors and of the distributions induced on boolean functions; these distributions are either uniform on subsets of boolean functions, or concentrated on a single function.

The present paper is organized as follows. We show in Section 2 how all the trees computing a specific boolean function can be derived from a finite set of minimal trees by a few simple operations. Our main results are also given in this section, namely the asymptotic expression of the probability of the boolean function in terms of its complexity, and the (relatively) simple form of a random expression computing a boolean function. The rest of the paper is devoted to the proof of these results. We first recall in Section 3 basic facts and former results on tautologies, i.e. on the trees that compute the simplest boolean function in our system: the constant *True*. Next we give technical results on expansions and on the inverse operation of pruning in Section 4, before considering irreducible trees in Section 5. Finally we present possible extensions in Section 6.

2 Results: limiting ratio of trees computing a given function

We begin by a brief presentation of the formulas we consider, then give a couple of definitions in order to state the main result concerning the limiting ratio of trees computing a given function.

Trees over implication. We consider in this paper formulas built with the single connector of implication (denoted by $\rightarrow$) and k positive literals $\{x_1, \ldots, x_k\}$. These formulas can be represented

by full binary trees whose internal nodes are all labelled by $\rightarrow$ and leaves by some literals. We denote by $\mathcal{F}_k$ this set of formulas. Each formula, or tree, is associated to a specific boolean function; we say that a tree is *computing* a specific boolean function. The boolean function computed by a tree A is denoted by $[A]$. Every tree A of $\mathcal{F}_k$ can be written in a unique way as

$$A = A_1 \rightarrow (A_2 \rightarrow (\ldots \rightarrow (A_p \rightarrow r(A))\ldots))$$

where $A_i \in \mathcal{F}_k$ and $r(A) \in \{x_1, \ldots, x_k\}$. We refer to this as the *canonical decomposition* of a

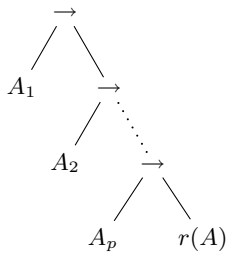

Fig. 1. The canonical decomposition of a tree.

tree A – see Figure 1. The subtrees $A_1, \ldots, A_p$ are called the *premises* of A, and the rightmost leaf $r(A)$ is called the *goal* of A. Analogously, premises and goal of any subtree of A is defined.

Limiting ratio. We define the size $|A|$ of a tree A as the number of its *leaves*. The *limiting ratio* of a subset $\mathcal{A}$ of trees is defined as

$$\mu_k(\mathcal{A}) = \lim_{n\to\infty} \frac{|\{A \in \mathcal{A}, |A| = n\}|}{|\{A \in \mathcal{F}_k, |A| = n\}|}$$

if this limit exists. We now define the *limiting ratio of a function* f as the limiting ratio of all trees computing f; that is, $\mu_k(f) = \mu_k(\{A \in \mathcal{F}_k \mid [A] = f\})$. Introducing the generating functions $\sum_n |\{A \in \mathcal{A}, |A| = n, [A] = f\}| z^n$, the results of Drmota [22], Lalley [23] and Woods [7] give us an easy way to prove that the limiting ratio of each boolean function is defined in the system $\mathcal{F}_k$ – i.e. for all boolean functions f, the limit defining $\mu_k(f)$ exists. These theorems are nicely described in Flajolet and Sedgewick [24,25].

Valid expansions of a tree. We now define three rules, called *expansion rules*, that allow, starting from a tree A, to obtain larger trees computing the same function as A. Let A be a tree and B one of its subtrees and let the root of B be denoted by ν.

The first expansion of A is called *valid expansion by a tautology*. We say that the tree A' obtained by replacing the subtree B with the subtree $C \rightarrow B$ in A, where C is a tautology, is a valid expansion of A by a tautology at node ν. Of course A' computes the same function as A since $[C \rightarrow B] = [B]$.

The second expansion of A is called *valid expansion by goal* α. If substituting B with $C \rightarrow B$ yields a tree A' computing the same function as A *for any tree* C *with goal* α, we say that any of these trees A' is obtained from A by a valid expansion of type "goal α" at node ν.

The third expansion of A is called *valid expansion by premise* α. If substituting B with $C \rightarrow B$ yields a tree A' computing the same function as A *for any tree* C *with a premise equal to* α, we say that any of these trees A' is obtained from A by a valid expansion of type "premise α" at node ν.

Figure 2 represents the shape of the tree obtained after a valid expansion at the root of B. Given a tree A, we define $E(A)$ to be the set of all trees obtained from A by a single valid expansion of any of the three types defined above. Note that all trees in $E(A)$ compute the same function as

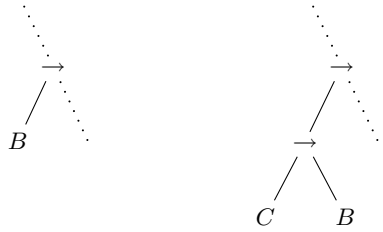

Fig. 2. Valid expansion with the subtree C in the root of B.

A. We naturally extend E to any set of trees $\mathcal{A} \subseteq \mathcal{F}_k$ by letting $E(\mathcal{A}) = \bigcup_{A \in \mathcal{A}} E(A)$. In the same way we define $E^0(\mathcal{A}) = \mathcal{A}$, $E^i(\mathcal{A}) = E(E^{i-1}(\mathcal{A}))$ and $E^*(\mathcal{A}) = \bigcup_{i \in \mathbb{N}} E^i(\mathcal{A})$.

Given a tree A, we define $\lambda(A)$ as the number of types of valid expansions of A; more precisely, this is the number of pairs (ν, α), where ν is a node of A (either an internal node or a leaf) and $\alpha \in \{x_1, \ldots, x_k\}$, such that an expansion of type "goal α" is valid in the node ν, plus the number of couples (ν, α) such that an expansion of type "premise α" is valid in the node ν, plus $2|A| - 1$ (this is counting the tautology expansions in every of the $2|A| - 1$ nodes of A).

For a boolean function f depending on a finite number of variables of $\{x_i \mid i > 0\}$, we define its *complexity* $L(f)$ to be the size of the smallest trees (over implication) computing f. Trees of size $L(f)$ computing f are called *minimal trees* of f; their set is denoted by $\mathcal{M}(f)$. Given a boolean function f, we define $\lambda(f)$ as the sum of all $\lambda(M)$ when M runs over all minimal trees computing f. It will be proved that $\lambda(f)$ does not depend on the number k of ambient variables. We can now state the main result of this paper.

Theorem 1. *Let f be a boolean function different from* True*. Almost all trees computing f are obtained by a single expansion of a minimal tree of f:*

$$\mu_k(f) \sim \mu_k(E(\mathcal{M}(f))).$$

As a consequence, the limiting ratio of f is asymptotically (as $k \to \infty$) equal to:

$$\mu_k(f) = \frac{\lambda(f)}{4^{L(f)}\ k^{L(f)+1}} + O\left(\frac{1}{k^{L(f)+2}}\right).$$

A proof of this theorem is given at the end of Section 5, where bounds on $\lambda(f)$ are also provided – see Proposition 1.

3 Limiting ratio and structure of tautologies

In this section, we recall some results from [4] on trees computing the constant function *True* (tautologies). Some of us proved there that the limiting ratio of all tautologies is equivalent to the limiting ratio of the family of *simple tautologies*: formulas such that one premise is equal to the goal of the formula. The limiting ratio of the set G_k of simple tautologies is equal to:

$$\mu_k(G_k) = \frac{4k+1}{(2k+1)^2} = \frac{1}{k} - \frac{3}{4k^2} + O\left(\frac{1}{k^3}\right).$$

Moreover, the following bounds on the limiting ratio of all tautologies (denoted by Cl_k) were given for k tending to infinity:

$$\frac{1}{k} - \frac{47}{64k^2} + O\left(\frac{1}{k^3}\right) \leqslant \mu_k(Cl_k) \leqslant \frac{1}{k} + \frac{17}{4k^2} + O\left(\frac{1}{k^3}\right).$$

Finally we recall two facts on the structure of tautologies. For a node ν of a tree A, we define the *left depth* of the node ν as the number of left branches needed to reach ν from the root of A.

We define in the same way the left depth of a subtree B of A as the left depth of its root. Let A be a tree and B one of its subtrees; B is called a *left subtree* of A if the root of B is the left son of its first ancestor. Let A be a tree which is a tautology: then the goal of A has a second occurrence at left depth 1 in A. Moreover, if A is a non-simple tautology, there exist in A either three occurrences of the same variable or two times two occurrences of two distinct variables among the leaves of left depth at most 3.

4 Expansion and pruning

We now study some of the properties of the expansion rules defined in Section 2. Given a tree A and a left subtree B of A, we denote by $A \setminus B$ the tree obtained by removing B from A. More precisely, since B is a left subtree of A, it is the left son of a tree of the form $B \to C$ in A; the tree $A \setminus B$ is obtained by substituting the subtree $B \to C$ by C in A – see Figure 3. The following three lemmas give (necessary and) sufficient conditions for a tree to be a single expansion of a certain type of a smaller tree.

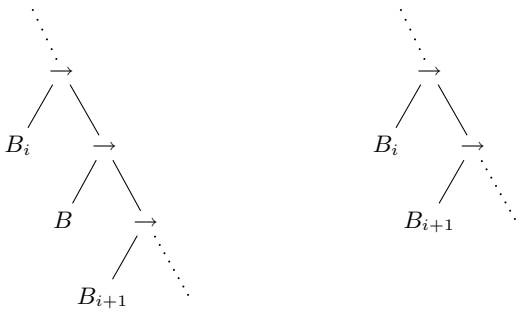

Fig. 3. Removing a left subtree B.

Lemma 1. *Let A be a tree and B be a left subtree of A. If B is a tautology, then A is obtained by a single valid expansion of type "tautology" of $A \setminus B$.*

Proof. This is obvious from the definition of expansions by tautologies. □

Lemma 2. *Let A be a tree and B be a left subtree of A. Let β be the goal of B. If substituting B by 1 or β in A yields a tree computing $[A]$ in both cases, then A is obtained by a single valid expansion of type "goal β" of $A \setminus B$.*

Proof. Let A_1 be the tree A where B is replaced with β, and A_2 be the tree A where B is replaced with 1. Let B' be any tree with goal β, and A' be the tree obtained from A by replacing B with B'. Of course $\beta \leqslant [B'] \leqslant 1$. Then by induction on the size of the formula, we obtain $[A] = [A_1] \leqslant [A'] \leqslant [A_2] = [A]$ or $[A] = [A_1] \geqslant [A'] \geqslant [A_2] = [A]$, depending whether the left depth of the root of B is even or odd. In any case, $[A'] = [A]$. Moreover, $[A \setminus B] = [A]$ since $[A \setminus B] = [A_2]$. □

Lemma 3. *Let A be a tree and B be a left subtree of A. Suppose that B has a premise of size one β. If substituting B with 1 or $\bar{\beta}$ in A gives a tree computing $[A]$ in both cases, then A is obtained by a single valid expansion of type "premise β" of $A \setminus B$.*

Proof. The proof is similar to the previous one. □

When going from A to $A \setminus B$ with one of the three lemmas above, we shall say $A \setminus B$ is obtained by *pruning* the left subtree B in A. Note the difference between pruning a subtree and cutting a subtree: if A is a tree and B one of its left subtrees, we will use the term "cutting the subtree B" when we remove B from A without any condition on B, and the term "pruning the subtree B" when we remove B from A while A is a valid expansion of $A \setminus B$. However, both final trees are denoted by $A \setminus B$.

A tree which cannot be pruned is called an *irreducible tree*. Of course all minimal trees computing a function f are irreducible. However, the converse is not true; indeed consider the function $f = x_1 \vee (\bar{x}_2 \wedge \bar{x}_3) \vee (\bar{x}_2 \wedge x_4)$. It can be checked that $(x_4 \to x_2) \to (((x_2 \to x_3) \to x_3) \to x_1)$ computes f, is irreducible, but not minimal since $((x_3 \to x_4) \to x_2) \to x_1$ is smaller and also computes f. We also remark that the system of pruning rules is not confluent.

We now define a new way of getting large trees from a smaller one. But this time, it does not preserve the function computed by the initial tree; its purpose is to establish some upper bounds on the limiting ratio of expansions. This new mapping X is called *extension* (it is different from expansion). The mapping X is defined recursively as follows: for a tree T consists of a single leaf α, $X(\alpha)$ is the set of all trees whose goal is labelled by α. If $T = L \to R$, we let

$$X(L \to R) = \left\{ A_1 \to \left(\cdots \to \left(A_p \to \left(\tilde{L} \to \tilde{R} \right) \right) \dots \right) \mid A_1, \dots, A_p \in \mathcal{F}_k,\ \tilde{L} \in X(L),\ \tilde{R} \in X(R) \right\}.$$

See Figure 4 for a graphical representation of the recursive definition of this mapping, and Figure 5

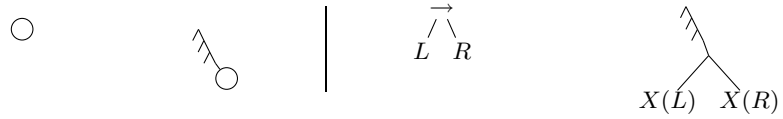

Fig. 4. The recursive definition of the *extension* mapping.

for the general shape of extensions of a given tree: the left sons of the combs are arbitrary trees (not represented not to overload the picture). We naturally extend X to a set of trees $\mathcal{A} \subseteq \mathcal{F}_k$

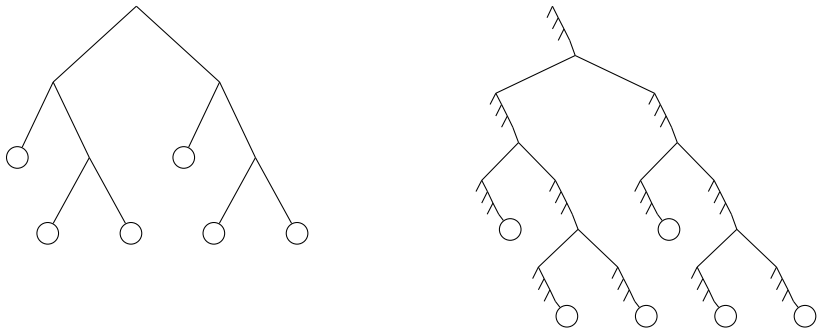

Fig. 5. A tree A on the left and the set $X(A)$ it defines, on the right.

by letting $X(\mathcal{A}) = \bigcup_{A \in \mathcal{A}} X(A)$. Notice that $X(X(\mathcal{A})) = X(\mathcal{A})$ for any $\mathcal{A} \subseteq \mathcal{F}_k$. The relationship between extensions and expansions is given below.

Lemma 4. *Let A be a tree. Then $E^*(A) \subseteq X(A)$.*

Proof. Let A be a tree. Since $X(X(A)) = X(A)$, all is needed is to prove that $E(A) \subseteq X(A)$. Recall that any tree $A' \in E(A)$ is obtained by substituting a subtree B of A with a tree of the form $C \to B$. It is clear from the definition of extensions that $C \to B \in X(B)$, and it follows that $A' \in X(A)$. □

Let $\mathcal{V}$ be a fixed finite subset of the variables $\{x_i \mid i > 0\}$, independent of the number of variables k we consider. Let p and q be two integers. Let $\mathcal{B}_q^p(\mathcal{V})$ the set of trees $B \subseteq \mathcal{F}_k$ such that $p \leqslant |B| \leqslant pq + 1$ and which contain at least p leaves labelled in $\mathcal{V}$. Note that $\mathcal{B}_q^p(\mathcal{V})$ implicitly depends on the number k of variables considered in our system.

Lemma 5. *The limiting ratio of $X(\mathcal{B}_q^p(\mathcal{V}))$ satisfies the next equation:*

$$\mu_k \left(X \left(\mathcal{B}_q^p(\mathcal{V}) \right) \right) = O\left(\frac{1}{k^p} \right).$$

Proof of this lemma is omitted in this short abstract. Let $\mathcal{A}_q^p(\mathcal{V})$ be the set of trees of $\mathcal{F}_k$ which contain p leaves labelled in $\mathcal{V}$, all of them being of left depth at most q. Notice that $\mathcal{A}_q^p(\mathcal{V})$ is infinite – as opposed to $\mathcal{B}_q^p(\mathcal{V})$.

Lemma 6. *$\mathcal{A}_q^p(\mathcal{V}) \subseteq X(\mathcal{B}_q^p(\mathcal{V}))$ and consequently $E^*(\mathcal{A}_q^p(\mathcal{V})) \subseteq X(\mathcal{B}_q^p(\mathcal{V}))$.*

Proof. Let $A \in \mathcal{A}_q^p(\mathcal{V})$. Let $\nu_1, \ldots, \nu_p$ be p leaves of A labelled with variables from $\mathcal{V}$, all with left depth at most q. Let $C_1, \ldots, C_r$ be the set of maximal (w.r.t. inclusion) left subtrees of A not containing any of the nodes ν_i. Let B be the tree obtained from A by removing all C_i, i.e. $B = A \setminus \{C_1, \ldots, C_r\}$. Of course $A \in X(B)$, and it can be checked that $B \in \mathcal{B}_q^p(\mathcal{V})$: indeed the largest tree B that can be obtained is when all nodes ν_i have a left depth q and belong to distinct premises of A, and $|B| = pq + 1$ in this case. Thus $A \in X(\mathcal{B}_q^p(\mathcal{V}))$. The second part of the lemma follows from Lemma 4 and the fact that $X(X(B_q^p(\mathcal{V}))) = X(\mathcal{B}_q^p(\mathcal{V}))$. □

Using Lemma 5 and 6 we obtain:

Corollary 1. *It holds that:*

$$\mu_k \left(E^* \left(\mathcal{A}_q^p(\mathcal{V}) \right) \right) = O\left(\frac{1}{k^p} \right).$$

5 Irreducible trees and their expansions

Let f be a boolean function different from *True*. The variable x is called an *essential variable* of f if the two functions obtained by evaluating x to 0 and to 1 are two distinct functions. Otherwise, x is called an *inessential variable* of f.

Let A be a tree and ν one of its nodes of positive left depth (either an internal node or a leaf). We define $\Delta(\nu)$ to be the smallest left subtree of A containing ν. In the same way, for a node ν of left depth at least 2, we define $\Delta^2(\nu)$ to be the smallest left subtree strictly containing $\Delta(\nu)$ – see Figure 6. We shall also write $\Delta(B)$ for a subtree B as a shortcut for $\Delta(\nu)$, where ν is the root of B (and in the same way $\Delta^2(B)$ for $\Delta^2(\nu)$).

Lemma 7. *Any tree A computing a function $f \neq True$ contains at least $L(f)$ occurrences of essential variables of f.*

Proof. Let A be a tree computing $f \neq True$. First remark that the goal of A is an essential variable; otherwise A would compute the constant *True* function. So $\Delta(\nu)$ is well defined for any leaf labelled with an inessential variable. Let $\{\Delta_1, \ldots, \Delta_p\}$ be the set of maximal $\Delta(\nu)$ (with respect to inclusion) when ν runs over all the leaves of A labelled with inessential variables. Of course all Δ_i are disjoint. If we assign the value 1 to all inessential variables, all Δ_i evaluate to 1, because their goals are inessential variables. Thus $A' = A \setminus \{\Delta_1, \ldots, \Delta_p\}$ computes f. Since A' only contains essential variables and $|A| \geqslant |A'| \geqslant L(f)$, it follows that there are at least $L(f)$ essential variables. □

The size of a tree A computing $f \neq True$ can be written $|A| = L(f) + e + i$, where $L(f) + e$ is the number of leaves labelled with essential variables and i is the number of leaves labelled with inessential variables; notice that $e \geqslant 0$ by Lemma 7. Given a function f different from *True*, we decompose the set of irreducible trees computing f into the following disjoint sets:

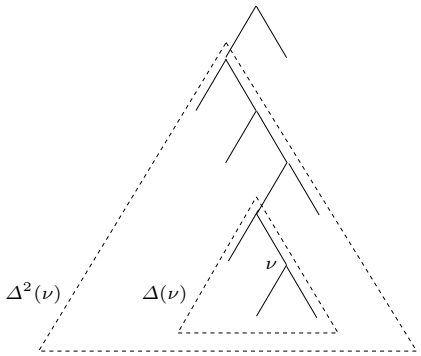

Fig. 6. The left subtrees $\Delta(\nu)$ and $\Delta^2(\nu)$ associated to a node ν of a tree.

- $\mathcal{M}(f)$ is the set of all minimal trees, i.e. trees of size $L(f)$ (case $e = i = 0$);
- $\mathcal{P}_1(f)$ is the set of irreducible trees of size greater than $L(f)$, with exactly $L(f)$ occurrences of essential variables and at least one occurrence of an inessential variable (case $e = 0$, $i > 0$);
- $\mathcal{P}_2(f)$ is the set of irreducible trees of size $L(f) + 1$, without any occurrence of inessential variables (case $e = 1$, $i = 0$);
- $\mathcal{P}_3(f)$ is the set of irreducible trees of size greater than $L(f) + 1$, with exactly $L(f) + 1$ occurrences of essential variables and $i > 0$ occurrences of *all distinct* inessential variables (case $e = 1$, $i > 0$, first part);
- $\mathcal{P}_4(f)$ is the set of irreducible trees of size greater than $L(f) + 2$, with exactly $L(f) + 1$ occurrences of essential variables and $i > 0$ occurrences of inessential variables such that at least one inessential variable is repeated (case $e = 1$, $i > 0$, second part);
- $\mathcal{P}_5(f)$ is the set of irreducible trees containing at least $L(f)+2$ occurrences of essential variables (case $e \geqslant 2$, $i \geqslant 0$).

Of course any tree computing f falls in an iterated expansion of an irreducible tree computing f (obtained by repeated pruning). Theorem 1 relies on evaluating the limiting ratios of $E^*(\mathcal{C})$ for each of the classes $\mathcal{C}$ defined above. We first prove that the two sets $\mathcal{P}_1(f)$ and $\mathcal{P}_3(f)$ are empty.

Lemma 8. *For any boolean function f different from* True*, the set $\mathcal{P}_1(f)$ is empty.*

Proof. Suppose that $\mathcal{P}_1(f)$ is not empty, and let $A \in \mathcal{P}_1(f)$. The size A is $L(f) + i$, with exactly $L(f)$ occurrences of essential variables and $i > 0$ occurrences of inessential ones. Let $\{\Delta_1, \ldots, \Delta_p\}$ be the set of maximal $\Delta(\nu)$ (with respect to inclusion) when ν runs over all the leaves of A labelled by an inessential variable. If we assign the value 1 to all inessential variables, all Δ_i evaluate to 1, because their goals are inessential variables. Moreover, since they are left subtrees, the tree $A' := A \setminus \{\Delta_1, \ldots, \Delta_p\}$ computes f. Since A contains $L(f)$ occurrences of essential variables, no Δ_i contains an essential variable.

Suppose now that there does not exist any assignment of the inessential variables such that Δ_1 evaluates to 0: then Δ_1 is a tautology and A is reducible, this is absurd. Hence there exists an assignement a of all inessential variables such that Δ_1 evaluates to 0 under a. Notice that Δ_1 cannot be a premise of A because $f \neq True$, so $\Delta_1^2 := \Delta^2(\Delta_1)$ is well defined and evaluates to 1 under a (because its premise Δ_1 evaluates to 0). Let $S = \{\Delta_i \mid [\Delta_{i|a}] = 1\} \cup \{\Delta_i^2 \mid [\Delta_{i|a}] = 0\}$ – where $[C_{|a}]$ denotes the function computed by the subtree C under the assignement a. The set S is composed of left subtrees, all evaluating to 1 under a. Moreover S contains Δ_1^2 which contains at least one essential variable (its goal) – otherwise Δ_1 would not be maximal. Thus $A'' := A \setminus S$ is of size at most $L(f) - 1$ and computes f, this is absurd. □

Using similar ideas, we obtain the following:

Lemma 9. *For any boolean function f different from* True*, the set $\mathcal{P}_3(f)$ is empty.*

Proof is not given in this short abstract. It is easy to check that both $\mathcal{P}_4(f)$ and $\mathcal{P}_5(f)$ are non empty for any function $f \neq True$. On the other hand, $\mathcal{P}_2(f)$ may be empty or not, depending on f: $\mathcal{P}_2(f)$ is empty for $f = x_1$ while $\mathcal{P}_2(g)$ is not empty for $g = x_1 \vee (\bar{x}_2 \wedge x_3 \wedge x_4) \vee (\bar{x}_2 \wedge \bar{x}_5)$. Indeed, it can be checked that $L(g) = 6$ and $(x_3 \to (x_4 \to x_2)) \to (((x_5 \to x_2) \to x_2) \to x_1)$ belongs to $\mathcal{P}_2(g)$. Our next step is to prove that iterated expansions of $\mathcal{P}_2(f)$ yields a family with small limiting ratio. Thus we can assume that f is such that $\mathcal{P}_2(f) \neq \emptyset$ – otherwise the limiting ratio of $E^*(\mathcal{P}_2(f))$ is obviously 0. The following lemma establishes some restrictions on the possible types of expansions in $\mathcal{M}(f)$ and $\mathcal{P}_2(f)$.

Lemma 10. *Let f be a boolean function different from* True*, and $A \in \mathcal{M}(f) \cup \mathcal{P}_2(f)$. No valid expansion of type* goal *or* premise *with respect to an inessential variable is possible in A.*

Proof. We develop here a proof by contradiction. Let $A \in \mathcal{M}(f) \cup \mathcal{P}_2(f)$. Let ν be one of its nodes where we are able to perform a valid expansion of type *goal* or *premise* with respect to an inessential variable α. Notice that the left depth of ν is at least 1; otherwise f would be equal to *True*. Thus $\Delta(\nu)$ is well defined. We shall first prove that $|\Delta(\nu)| = 1$.

Suppose the valid expansion in ν is of type *goal* α. Let A' be the tree obtained by expanding A in ν with the left subtree reduced to α. Then we have $[A'_{|\alpha=0}] = [A \setminus \Delta(\nu)] = f$ and we conclude that $|\Delta(\nu)| = 1$ and $|A| = L(f) + 1$ (otherwise we would have a tree smaller than $L(f)$ computing f). Suppose now the valid expansion in ν is of type *premise* α. Let x be the goal of A, and let A' be the tree obtained by expanding A in ν with the left subtree $\alpha \to x$. Then we have $[A'_{|\alpha=1,\ x=0}] = [A \setminus \Delta(\nu)_{|x=0}] = f_{|x=0}$ and of course $[A \setminus \Delta(\nu)_{|x=1}] = 1 = f_{|x=1}$. Again we conclude that $[A \setminus \Delta(\nu)] = f$; it follows that $|\Delta(\nu)| = 1$ and $|A| = L(f) + 1$ in this case too.

So no matter the type of the valid expansion, we know that $\Delta(\nu)$ is a leaf. Let y be the label of $\Delta(\nu)$. In the tree A, the left subtree $\Delta(\nu)$ computes y. Moreover, for both types of expansion, we have shown that $A \setminus \Delta(\nu)$ still computes f; to put it otherwise, substituting $\Delta(\nu)$ with 1 or its goal y in A does not change the computed function. It follows by Lemma 2 that A is reducible, which is absurd. □

Notice that Lemma 10 shows that $\lambda(f)$ defined in Section 2 does not depend on the number of variables k we consider. We are now ready to bound the limiting ratio of $E^*(\mathcal{P}_2(f))$.

Lemma 11. *For any boolean function f different from* True*,*

$$\mu_k(E^*(\mathcal{P}_2(f))) = O\left(\frac{1}{k^{L(f)+2}}\right).$$

Proof. Since $\mathcal{P}_2(f)$ is finite, $\mu_k(\mathcal{P}_2(f)) = 0$ and all we have to show is that $\mu_k(E^*(E(\mathcal{P}_2(f)))) = O(1/k^{L(f)+2})$. Let $A \in E(\mathcal{P}_2(f))$; we have $A \in E(I)$ for an irreducible tree $I \in \mathcal{P}_2(f)$. We will prove that A satisfies one of the following conditions:

- A contains at least $L(f)+2$ occurrences of essential variables with left depth at most $L(f)+2$;
- A contains $L(f)+1$ occurrences of essential variables and two occurrences of the same inessential variable, all with a left depth at most $L(f)+2$.

If A is obtained from I by an expansion of type *goal* or *premise*, then it must be with respect to an essential variable by Lemma 10. Now remark that I is of size $L(f)+1$, so all its node are of left depth at most $L(f)$ (there is at least a node per left depth). Since expansions preserve left depth of nodes present in the initial tree, we conclude that A satisfies the first condition above.

Suppose now that A is obtained from I by an expansion of type *tautology*. From the results recalled in Section 3, we know that this tautology contains two occurrences of some variable x in its nodes of left depth at most 1. If x is an essential variable of f, then A satisfies the first condition above. Otherwise, if x is an inessential variable of f, then A satisfies the second condition above.

Let us denote by Γ the set of essential variables of f. Let $\mathcal{N}_1 = \mathcal{A}^{L(f)+2}_{L(f)+2}(\Gamma)$ and

$$\mathcal{N}_2 = \bigcup_{\alpha \in \{x_1,\ldots,x_k\}} \mathcal{A}^{L(f)+3}_{L(f)+2}(\Gamma \cup \{\alpha\}).$$

We have just proved that $E(\mathcal{P}_2(f)) \subseteq \mathcal{N}_1 \cup \mathcal{N}_2$. It follows that $E^*(E(\mathcal{P}_2(f))) \subseteq E^*(\mathcal{N}_1) \cup E^*(\mathcal{N}_2)$. Corollary 1 yields

$$\mu_k(E^*(\mathcal{N}_1)) = O\left(\frac{1}{k^{L(f)+2}}\right).$$

Moreover, still with Corollary 1 we obtain that for any variable α:

$$\mu_k(E^*(\mathcal{A}^{L(f)+3}_{L(f)+2}(\Gamma \cup \{\alpha\}))) = O\left(\frac{1}{k^{L(f)+3}}\right).$$

It follows that

$$\mu_k(E^*(\mathcal{N}_2)) = O\left(\frac{1}{k^{L(f)+2}}\right).$$

Thus, we have proved that $\mu_k(E^*(\mathcal{P}_2(f))) = O(1/k^{L(f)+2})$. □

We now turn our attention towards $\mathcal{P}_4(f)$ and $\mathcal{P}_5(f)$. In the same way as for the limiting ratio of iterated expansions of $\mathcal{P}_2(f)$, we can show the following:

Lemma 12. *For any boolean function f different from* True*, it holds that:*

$$\mu_k(E^*(\mathcal{P}_4(f) \cup \mathcal{P}_5(f))) = O\left(\frac{1}{k^{L(f)+2}}\right).$$

Proof is omitted in this short abstract. The last step towards the proof of Theorem 1 is to study limiting ratio of expansions of the minimal trees computing a given function. We show the following:

Lemma 13. *Let f be a boolean function different from* True*. Using a single expansion of the minimal trees, we get:*

$$\mu_k(E(\mathcal{M}(f))) = \frac{\lambda(f)}{4^{L(f)}k^{L(f)+1}} + O\left(\frac{1}{k^{L(f)+2}}\right).$$

Moreover,

$$\mu_k\left(E^*\left(\mathcal{M}(f)\right) \setminus E(\mathcal{M}(f))\right) = O\left(\frac{1}{k^{L(f)+2}}\right).$$

The proof is not given in this short abstract. The first part relies on simple calculations of limiting ratio using generating functions and the results on tautologies recalled in Section 3; the second part, in the spirit of Lemma 11, is more technical and relies on a case study with regard to the location of the second expansion with respect to the first one. Theorem 1 is now obtained easily.

Proof of Theorem 1. Of course each tree computing f falls in a set obtained by an arbitrary number of expansions of an irreducible tree computing f. That is, the set of trees computing f is exactly $\mathcal{A}(f) = E^*(\mathcal{M}(f) \cup \mathcal{P}_1(f) \cup \mathcal{P}_2(f) \cup \mathcal{P}_3(f) \cup \mathcal{P}_4(f) \cup \mathcal{P}_5(f))$. Now of course

$$E(\mathcal{M}(f)) \subseteq \mathcal{A}(f) \subseteq E(\mathcal{M}(f)) \cup (E^*(\mathcal{M}(f)) \setminus E(\mathcal{M}(f))) \cup \bigcup_{i\in\{1,\ldots,5\}} E^*(\mathcal{P}_i(f)).$$

The result follows from Lemmas 8, 9, 11, 12 and 13. □

Let us now provide some bounds on the integer $\lambda(f)$.

Proposition 1. *Let f be a boolean function different from* True*, and let n be its number of essential variables. It holds that*

$$2 \cdot (2L(f) - 1) \cdot |\mathcal{M}(f)| \leqslant \lambda(f) \leqslant (1 + 2n) \cdot (2L(f) - 1) \cdot |\mathcal{M}(f)|.$$

Proof. Let M be a minimal tree computing f and let ν be a node of M. Since M is of size $L(f)$, the tree M has $2L(f)-1$ nodes in total. Thus all we have to show is that the number λ_ν of types of valid expansions in the node ν of M satisfies $2 \leqslant \lambda_\nu \leqslant 1+2n$.

The upper bound is simply obtained by counting all possible types of expansions: 1 for the tautology type, n of goal type, and n of premise type – recall from Lemma 10 that no expansion of type goal or premise with respect to an inessential variable is valid in M.

The lower bound is obtained by remarking that, besides the tautology type of expansion which is of course valid in ν, the expansion of type *premise* x is also valid in ν, where x is the goal of M. Indeed, let A be the tree obtained from M by replacing the subtree B rooted in ν by $C \to B$, where C is any tree with a premise equal to x. Of course $[A|_{x=0}] = [M|_{x=0}]$ because $[C|_{x=0}] = 1$. Moreover, $[A|_{x=1}] = [M|_{x=1}] = 1$ because x is the goal of M. Thus $[A] = [M]$ and we conclude that the expansion of type "goal x" is valid in ν. □

When building the tree underlying the formula, we assumed that it is chosen uniformly among the trees of a given (large) size. Assume now that it is obtained by a critical branching process, so that its size itself is random (see [8] for the definition of this process in the case of And/Or trees). This gives a different probability distribution on the set of boolean functions; let us denote it by π. Then we can obtain a similar result for this new distribution.

Proposition 2. *Let f be a boolean function different from* True*; then*

$$\pi(f) = \frac{|\mathcal{M}(f)|}{2^{2L(f)-1}\ k^{L(f)}} + O\left(\frac{1}{k^{L(f)+1}}\right).$$

6 Conclusion

When considering the limiting ratio of a boolean function, e.g., in the system of implication, it may not be enough to know that the limiting ratio exists, and one may naturally wish for some numerical information. For a fixed, (very) small number of boolean variables, explicit computation of the limiting ratios is feasible by writing, then solving, an algebraic system; see [3] for an overview of the mathematical technology involved and [8] for the application to And/Or trees. However, the fact that size of the system grows exponentially in k severely restricts hand-made evaluation. For a moderate number of variables, very recent results on explicit solving of algebraic systems [26] give us hope to extend the numerical computations a little bit farther. But exact computation will eventually fail, even for a "reasonable" number of boolean variables. Then we turn to asymptotic analysis; this is where our result comes in. Although it is likely that no general, easy-to-use expression of the constant factor $\lambda(f)$ holds for all boolean functions, we can still hope to obtain results for well-defined classes. Consider for example *read-once* functions, i.e., functions with $L(f)$ essential variables. An alternative definition is as functions whose minimal trees contain no repetition of variables. We can prove that the average number of expansions of read-once functions of complexity c is

$$\bar{\lambda}_{r.o.}(c) \sim \frac{\sqrt{\pi}}{2\sqrt{2}}\, c^{3/2} \left(\frac{4}{e}\right)^c.$$

We should also mention that Theorem 1 requires us to specify the boolean function, and does not hold uniformly over *all* boolean functions; hence we are still unable to compute the average complexity of a boolean function chosen according to this probability distribution. Further work is required before we can either verify or invalidate the Shannon effect for this non-uniform probability distribution.

Acknowledgements. We are grateful to Jakub Kozik for fruitful discussions about this problem.

References

1. Zaionc, M.: Statistics of implicational logic. Electronic Notes in Theoretical Computer Science **84** (2003)

2. Kostrzycka, Z., Zaionc, M.: Statistics of intuitionnistic versus classical logic. Studia Logica **76**(3) (2004) 307–328
3. Gardy, D.: Random boolean expressions. In: Colloquium on Computational Logic and Applications. (2006)
4. Fournier, H., Gardy, D., Genitrini, A., Zaionc, M.: Classical and intuitionistic logic are asymptotically identical. In: CSL. (2007) 177–193
5. Paris, J.B., Vencovská, A., Wilmers, G.M.: A natural prior probability distribution derived from the propositional calculus. Annals of Pure and Applied Logic **70** (1994) 243–285
6. Lefmann, H., Savický, P.: Some typical properties of large And/Or Boolean formulas. Random Structures and Algorithms **10** (1997) 337–351
7. Woods, A.R.: Coloring rules for finite trees, and probabilities of monadic second order sentences. Random Struct. Algorithms **10**(4) (1997) 453–485
8. Chauvin, B., Flajolet, P., Gardy, D., Gittenberger, B.: And/Or trees revisited. Combinatorics, Probability and Computing **13**(4-5) (July-September 2004) 475–497
9. Moczurad, M., Tyszkiewicz, J., Zaionc, M.: Statistical properties of simple types. Mathematical Structures in Computer Science **10**(5) (2000) 575–594
10. Kostrzycka, Z.: On the density of truth of implicational parts of intuitionistic and classical logics. J. of Applied Non-Classical Logics **13**(2) (2003)
11. Zaionc, M.: On the asymptotic density of tautologies in logic of implication and negation. Reports on Mathematical Logic **39** (2005) 67–87
12. Kostrzycka, Z.: On the density of truth in modal logics. In: Mathematics and Computer Science, Nancy (France), Proceedings to appear in DMTCS (September 2006)
13. Matecki, G.: Asymptotic density for equivalence. Electronic Notes in Theoretical Computer Science **140** (2005) 81–91
14. Kostrzycka, Z.: On asymptotic divergency in equivalential logics. Mathematical Structures in Computer Science **18** (2008) 311–324
15. Woods, A.: On the probability of absolute truth for and/or formulas. Bulletin of Symbolic Logic **12**(3) (2005)
16. Kozic, J.: Subcritical pattern languages for and/or trees. Technical report (2008)
17. Valiant, L.: Short monotone formulae for the majority function. Journal of Algorithms **5** (1984) 363–366
18. Boppana, R.B.: Amplification of probabilistic boolean formulas. Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science (1985) 20–29
19. Gupta, A., Mahajan, S.: Using amplification to compute majority with small majority gates. Comput. Complex. **6**(1) (1997) 46–63
20. Savický, P.: Random Boolean formulas representing any Boolean function with asymptotically equal probability. Discrete Mathematics **83** (1990) 95–103
21. Brodsky, A., Pippenger, N.: The boolean functions computed by random boolean formulas or how to grow the right function. Random Structures and Algorithms **27** (2005) 490–519
22. Drmota, M.: Systems of functional equations. Random Structures and Algorithms **10**(1-2) (1997) 103–124
23. Lalley, S.P.: Finite range random walk on free groups and homogeneous trees. The Annals of Probability **21** (1993)
24. Flajolet, P., Sedgewick, R.: Analytic combinatorics: Functional equations, rational and algebraic functions. Technical Report 4103, INRIA (January 2001)
25. Flajolet, P., Sedgewick, R.: Analytic Combinatorics. Cambridge University Press (2008) To appear. Available from the authors' web page.
26. Pivoteau, C., Salvy, B., Soria, M.: Combinatorial Newton iteration to compute Boltzmann oracle. Technical report (March 2008) Journées ALÉA; available at http://www-calfor.lip6.fr/pivoteau.