On Deterministic Approximation of DNF

Michael Luby[*] Boban Veličković[†]

TR-93-009

March 1993

The best throw of the die is to throw the die away
Chinese fortune cookie

Abstract

We develop efficient deterministic algorithms for approximating the fraction of truth assignments that satisfy a disjunctive normal form formula. Although the algorithms themselves are deterministic, their analysis is probabilistic and uses the notion of limited independence between random variables.

[*]International Computer Science Institute, 1947 Center Street, Berkeley, California 94704 and Computer Science Department, UC Berkeley, research partially supported by NSF operating grant CCR-9016468 and by grant No. 89-00312 from the United States-Israel Binational Science Foundation (BSF), Jerusalem, Israel.

[†]Department of Mathematics, U.C. Berkeley, research partially supported by NSF, research partially done while visiting the International Computer Science Institute

1 Introduction

Throughout this paper, let F denote a formula in disjunctive normal form (DNF) on n variables with m clauses of length at most t, and let $\Pr[F]$ denote the probability that a random, independent and uniformly chosen truth assignment to the variables satisfies F. We use nm throughout as the size of the description of F. The problem of computing $\Pr[F]$ exactly is known to be #P-complete [18, Valiant]. On the other hand, for many applications a good estimate of $\Pr[F]$ is all that is needed.

Let us say that Y is an *ϵ-approximation* of $\Pr[F]$ if $|Y - \Pr[F]| \leq \epsilon$, and that Y is an *ϵ-relative approximation* of $\Pr[F]$ if $|Y - \Pr[F]| \leq \epsilon \Pr[F]$. There is a trivial probabilistic algorithm which on input F, $\epsilon > 0$, and $\delta > 0$, computes an ϵ-approximation Y of $\Pr[F]$ with probability at least $1 - \delta$. The algorithm simply chooses $N = O(\ln(1/\delta)/\epsilon^2)$ random truth assignments and sets Y to the fraction of these assignments that satisfy F. The drawback of this algorithm is that it requires a source of truly random bits, which seems hard to come by in practice. Even if there were such a source, there is still a chance that the output of the algorithm has error greater than ϵ, and there seems to be no way of checking when this occurs. These difficulties motivate the goal of trying to find a deterministic polynomial time algorithm for approximating $\Pr[F]$.

Towards this goal, [21, Yao] shows that if there are pseudo-random generators then any problem in RP can be approximated in subexponential time, and [5, Boppana, Hirschfeld] extends this result to show that any problem in BPP can be approximated in subexponential time. All other previous work on deterministic DNF approximation that does not rely on any unproven conjectures is based on lower bounds on the computational power of constant depth unbounded fan-in boolean circuits (the complexity class AC^0) for computing parity (see [8, Furst Saxe Sipser], [2, Ajtai], [21, Yao], [9, Håstad]). A DNF formula is the special case when the depth of the circuit is two. [3, Ajtai Wigderson] describe a deterministic algorithm which given as input a constant depth circuit of size n and $\epsilon > 0$ runs in time $2^{n^{o(1)}}$ and produces an ϵ-approximation of the probability that the output of the circuit is 1 when the input is randomly chosen. They also provide a polynomial time approximation algorithm for the DNF problem when the length of each clause is bounded by a constant. [16, Nisan Wigderson] improve this by giving an algorithm that has running time $2^{\log^c(n/\epsilon)}$, where c is a constant that depends on the depth, and for the DNF problem c is approximately 12. An observation of [14, Luby Veličković Wigderson] improves their analysis to yield $c = 4$ for the DNF problem.

[3, Ajtai Wigderson] describe a polynomial time approximation algorithm for the DNF problem when t is a constant. On the other hand, for an ϵ-approximation we may assume without loss of generality that t is at most $\log(m/\epsilon)$, since by discarding from F all longer clauses we obtain a new formula G such that $|\Pr[F] - \Pr[G]| \leq \epsilon$.

This paper introduces several ideas which can be combined with known results to obtain approximation algorithms for the DNF problem. The first idea is that of sunflower reduction in a way similar to the way that [17, Razborov] uses it to prove exponential lower bounds for the size of the smallest monotone circuit for finding the biggest clique in a graph. Given a DNF formula F we look for a large collection of clauses which form a sunflower, i.e., the intersection of any two distinct clauses in this collection is the same. Then, we replace all the clauses in this collection by their common pairwise intersection, thus obtaining another formula which has probability of being satisfied close to that of F. We then repeat this procedure until no large sunflowers can be found, obtaining a new formula F'. A theorem of Erdös-Rado [6] implies that at the end of this procedure there are not too many clauses in F'. Because F' is so small, it turns out to be easier to approximate $\Pr[F']$, and because of the properties of the reduction this approximation is also a good approximation of $\Pr[F]$. This reduction can be combined with the algorithm from [16, Nisan Wigderson] (using the improvement based on an observation from [14, Luby Veličković Wigderson]) to produce a polynomial time ϵ-approximation for $\Pr[F]$ when $t = O(\log^{1/8}(nm))$ and ϵ is not too small.

The second approach relies on special constructions of small probability distributions. Given a distribution $\mathcal{D}$ let $\Pr_{\mathcal{D}}[F]$ denote the probability that F is satisfied by a truth assignment chosen according to $\mathcal{D}$. Our goal is to find a distribution $\mathcal{D}$ with small sample space such that $\Pr_{\mathcal{D}}[F]$ is close to $\Pr[F]$. Then, $\Pr_{\mathcal{D}}[F]$ can be calculated by exhaustive consideration of the points in the space. An easy counting argument shows that such a distribution exists. Our results can be viewed as progress towards finding a polynomial time construction of such a distribution.

Recall that a probability distribution $\mathcal{D}$ on the truth assignments is k-wise independent if any variable is equally likely to be 0 or 1 independently of the value of any other $k-1$ variables. We show that if t is bounded by a constant and $k = c\log(1/\epsilon)$ for some constant c then for any probability distribution $\mathcal{D}$ which is k-wise independent $\Pr_{\mathcal{D}}[F]$ is a good approximation of $\Pr[F]$. Our proof works if $\mathcal{D}$ is only k-wise almost independent in a certain technical sense. This combined with the results of [15, Naor Naor] and [1, Alon Goldreich Håstad Peralta], which produce such distributions with small sample spaces, yields a polynomial time approximation algorithm for $\Pr[F]$ if t is bounded by a constant, a result previously obtained by [3, Ajtai Wigderson] by different means.

The main contribution of this paper consists of the coloring algorithm. A *proper k-coloring* of F is a coloring of the variables of F using at most k colors such that no clause of F contains two variables of the same color. For all $i \in \{1, \ldots, k\}$, let $\mathcal{D}_i$ be a probability distribution on the variables in color class i such that there is l-wise independence between the variables, and let $\mathcal{D} = \times_{i\in\{1,\ldots,k\}}\mathcal{D}_i$ be the product distribution defining a distribution on all variables of F. We prove that $|\Pr_{\mathcal{D}}[F] - \Pr[F]| \leq k/2^l$. This suggests the following approximation algorithm for $\Pr[F]$. Given a proper coloring of F, for all $i \in \{1, \ldots, k\}$ we can explicitly construct sample space S_i for distribution $\mathcal{D}_i$ and let the sample space for $\mathcal{D}$ be $S = \times_{i\in\{1,\ldots,k\}}S_i$. Given S, we can compute $\Pr_{\mathcal{D}}[F]$ exactly. This yields an algorithm with running time dominated by $\times_{i\in\{1,\ldots,k\}}|S_i|$ for approximating $\Pr[F]$. The running time of the algorithm depends in an exponential way on the value of k. In general we may not be able to find a proper k-coloring for F where k is small. Instead, we find a proper coloring for a subformula G of F such that $\Pr[G]$ is close to $\Pr[F]$. This general approach is used to design deterministic algorithm that on input F and ϵ produces an ϵ-relative approximation of $\Pr[F]$ in polynomial time for fixed ϵ and $t = \log^{1-o(1)}(nm)$. For fixed ϵ and unrestricted clause length the running time the algorithm produces an ϵ-approximation in time polynomial in nm and $2^{\log^{1+o(1)}(nm)}$. For a fixed ϵ this algorithm is faster than the algorithm in [16, Nisan Wigderson], but, unlike [16, Nisan Wigderson], the running time grows exponentially with $1/\epsilon$.

The paper is organized as follows. In Section 2 we present a reduction procedure implicit in [10, Karp Luby] which converts an absolute error approximation algorithm to a relative error approximation algorithm. This is important since in applications one is mainly interested in obtaining a relativized approximation of the probability that a formula is satisfied and this is typically very small. In Section 3 we review some constructions of small probability distributions with limited independence between the random variables which will be used in subsequent sections. Section 4 contains a review of the algorithm of [16, Nisan Wigderson] and then presents the sunflower reduction algorithm. These two algorithms are then combined to obtain the algorithm described above. In Section 5 we develop a simple version of the coloring algorithm that illustrates the basic ideas used in the more refined (and faster) algorithm presented in Section 6. In Section 7 we generalize the coloring algorithm to the case when the variables are independently but not necessarily uniformly distributed. In Section 8 we make some conclusions and pose some open problems.

2 Absolute versus Relative Error

For practical reasons, it is sometimes important to obtain more than an absolute error approximation. For example, the failure probability of some networks can be expressed as $\Pr[F]$ and typically $\Pr[F]$ is very small. Thus, an estimate Y is more useful if Y is an ϵ-relative error approximation of $\Pr[F]$, i.e., if $|Y - \Pr[F]| \leq \epsilon \Pr[F]$. This raises the general question of whether or not an efficient approximation algorithm can be converted into an efficient relative error approximation algorithm. In general this seems unlikely. For example, in terms of absolute error there is no difference between the DNF problem and the CNF problem, in the sense that an approximation algorithm for one can easily be changed to work for the other. On the other hand, there is no efficient relative error approximation algorithm for the CNF problem that is deterministic unless P = NP or that is probabilistic unless RP = NP.

[10, Karp Luby], [11, Karp Luby Madras] provide probabilistic algorithms which, given a DNF formula F, $\epsilon > 0$ and $\delta > 0$, compute Y that is an ϵ-relative error approximation of $\Pr[F]$ with probability at least $1 - \delta$. The running time of these algorithms is polynomial in nm, $1/\epsilon$ and $\log(1/\delta)$. As described below, their work can be used to reduce the problem of finding an ϵ-relative error approximation of $\Pr[F]$ to the problem of finding an ϵ/m-approximation of $\Pr[F]$.

Given a deterministic approximation algorithm A we define a deterministic relative error approximation algorithm R as follows. On input F and $\epsilon > 0$, R first derives DNF formulas $F_1, \ldots, F_m$, where F_i is the formula obtained by setting all the literals in C_i to true and taking the disjunction of the first $i-1$ original clauses reduced according to this partial truth assignment. For each $i \in \{1, \ldots, m\}$, R makes a call to the algorithm A with input F_i and ϵ/m, and A returns Y_i. R then outputs

$$Y = \sum_{i \in \{1,\ldots,m\}} \Pr[C_i](1 - Y_i).$$

Y_i is within ϵ/m of the conditional probability that at least one of the first $i-1$ clauses is true given that C_i is true, and thus $\Pr[C_i](1-Y_i)$ is within $\epsilon \Pr[C_i]/m$ of the probability that C_i is the first clause to be satisfied. If, for each i, Y_i contained no error, then Y would be equal to $\Pr[F]$. Hence,

$$|Y - \Pr[F]| \leq \frac{\epsilon}{m} \cdot \sum_{i \in \{1,\ldots,m\}} \Pr[C_i].$$

But, $\sum_{i \in \{1,\ldots,m\}} \Pr[C_i]/\Pr[F] \leq m$ and thus Y is within $\epsilon \Pr[F]$ of $\Pr[F]$, as desired.

3 Constructions of Small Probability Spaces

Let $\mathcal{D}$ be a probability distribution. We say that P-valued random variables

$$\{x_1, \ldots, x_n\}$$

over $\mathcal{D}$ are *k-wise ϵ-dependent* (see [19, Vazirani], [15, Naor Naor]) provided for any subset I of $\{1, \ldots, n\}$ of size at most k and for every assignment $\sigma : I \to P$,

$$|\Pr_{\mathcal{D}}[\bigwedge_{i \in I}(x_i = \sigma(i))] - \prod_{i \in I} \Pr_{\mathcal{D}}[x_i = \sigma(i)]| \leq \epsilon.$$

If $\epsilon = 0$ then the random variables are *k-wise independent.*

We now review some known constructions of probability distributions with the above properties with small sample spaces. We start with multi-valued variables that are k-wise independent and uniformly distributed. Suppose l and k are integers and let $GF[2^l]$ be the field with 2^l elements. We identify elements of $GF[2^l]$ with l-bit strings. Let the sample space be the set of all polynomials over $GF[2^l]$ of degree at most $k-1$. We define random variables $\{x_1, \ldots, x_{2^l}\}$ as follows. Let x_i be $p(i)$ where p is a uniformly chosen polynomial from the sample space.

Lemma 1 $\{x_1, \ldots, x_{2^l}\}$ *as described above are k-wise independent and uniformly distributed over $GF[2^l]$. The size of the sample space is 2^{kl}. The sample space can be constructed in time polynomial in its size.*

We can modify this construction to obtain $\{0,1\}$-valued random variables

$$\{y_1, \ldots, y_{l2^l}\}$$

as follows. If $i = ul + b$, where $0 \leq u \leq 2^l - 1$ and $1 \leq b \leq l$ let y_i be the b-th bit of x_u.

Lemma 2 $\{y_1, \ldots, y_{l2^l}\}$ *as described above are k-wise independent and uniformly distributed over $\{0,1\}$. The size of the sample space is 2^{kl}. The sample space can be constructed in time polynomial in its size.*

Given values $v = \{v_1, \ldots, v_{2^l}\}$, where $v_i \in \{0, \ldots, 2^l\}$, one can use the same sample space to define a distribution on $\{0,1\}$-valued random variables $\{z_1, \ldots, z_{2^l}\}$ as $z_i = 1$ if $x_i \leq v_i$ and 0 otherwise.

Lemma 3 $\{z_1, \ldots, z_{2^l}\}$ *as described above are k-wise independent and non-uniformly distributed as follows:* $\Pr[z_i = 1] = v_i/2^l$. *The size of the sample space is 2^{kl}. The sample space can be constructed in time polynomial in its size.*

The idea of relaxing the requirement from perfect k-wise independent to approximately k-wise independent distributions and the initial work on explicit constructions of small sample spaces that meet these relaxed requirements is due to [15, Naor Naor]. If $\{0,1\}$-valued random variables are uniformly distributed we say that they are k-wise ϵ-*biased* if for any subset I of $\{1, \ldots, n\}$ of size at most k,

$$\Pr_{\mathcal{D}}[\oplus_{i \in I} x_i = 0] - 1/2| \leq \epsilon.$$

[15, Naor Naor] present, for given n and ϵ, an explicit construction of a probability distribution $\mathcal{D}$ on $\{0,1\}$-valued random variables $\{x_1, \ldots, x_n\}$ which are n-wise ϵ-biased where the size of the sample space of $\mathcal{D}$ is polynomial in n/ϵ. They combine this with the construction for k-wise independent $\{0,1\}$-valued uniformly distributed random variables described above, yielding a probability distribution on $\{0,1\}$-valued random variables $\{x_1, \ldots, x_n\}$ which are k-wise ϵ-biased, and the size of the sample space is polynomial in $\log(n)$, k and $1/\epsilon$. Notice that if a distribution is k-wise ϵ-biased then it is k-wise $\epsilon 2^k$ dependent. This immediately yields the following lemma.

Lemma 4 *There is an explicit construction of a sample space where random variables $\{x_1, \ldots, x_n\}$ are k-wise ϵ-dependent $\{0,1\}$-valued uniformly distributed. The size of the sample space is polynomial in* $\log(n)$, 2^k *and* $1/\epsilon$. *The sample space can be constructed in time polynomial in its size.*

Simpler constructions have subsequently been found by [1, Alon Goldreich Håstad Peralta]. We review one of their constructions, which is based on a result of [20, Weil] on character sums over finite fields. Suppose n is a positive integer and let $p > n$ be a prime. Let $\mathcal{D}$ be the uniform distribution on $\mathbf{Z}_p$. Let $\left(\frac{a}{b}\right)$ denote the Jacobi symbol. Given a random element $x \in \mathbf{Z}_p$, define the random variables $\{x_1, \ldots, x_n\}$, by $x_i = \left(\frac{x+i}{p}\right)$ if $x + i < p$ and $x_i = \left(\frac{x+i+1}{p}\right)$ if $x + i \geq p$. Then, x_i has equal probability of being 1 or -1. Mapping -1 to 1 and 1 to 0, we obtain $\{0,1\}$-valued random variables which are ϵ-biased for $\epsilon = n/\sqrt{p}$.

The following lemma, due to [7, Even Goldreich Luby Nisan Veličković], generalizes this to non-uniformly distributed random variables.

Lemma 5 *Given values $\{v_1, \ldots, v_n\}$, where $v_i \in [0,1]$, there is an explicit construction of a sample space where $\{0,1\}$-valued random variables $\{x_1, \ldots, x_n\}$ are k-wise ϵ-dependent and non-uniformly distributed as follows: $|\Pr[x_i = 1] - v_i| \leq \epsilon$. The size of the sample space is polynomial in $\log(n)$, 2^k and $1/\epsilon$. The sample space can be constructed in time polynomial in its size.*

4 Sunflower Reduction Algorithm

If we want to approximate $\Pr[F]$ to within an additive error ϵ, we can start by discarding all clauses of size greater than $\log(2m/\epsilon)$ and the probability of the resulting formula being satisfied is within $\epsilon/2$ of $\Pr[F]$. We now describe a slight modification of the algorithm from [16, Nisan Wigderson] due to [14, Luby Veličković Wigderson]. The algorithm takes as input a DNF formula F with m clauses on n variables and $\epsilon > 0$ and outputs an ϵ-approximation of $\Pr[F]$. By the above remark we may assume that all the clauses in F have length at most $\log(m/\epsilon)$. Set $l = \log(2(nm)/\epsilon)$, $r = c_1 l^2$, and $t = c_2 l^4$ for some constants c_1 and c_2 and, using Lemma 1 from Section 3, find a probability distribution $\mathcal{D}$ with sample space of size $(nr)^{2l}$ and nr random variables which are $2l$-wise independent and uniformly distributed over $\{1, \ldots, t\}$. Break the random variables into n blocks with r variables each and denote the variables in the i-th block by $\{y_1^i, \ldots, y_r^i\}$. Let $\mathcal{E}$ be the product of $\mathcal{D}$ and the uniform distribution on $\{0,1\}$-strings of length t. Define n random variables on $\mathcal{E}$ as follows. Given a random string σ of length t and a setting of the random variables y_j^i, for $1 \leq i \leq n$ and $1 \leq j \leq r$ the i-th random variable is defined by:

$$x_i = \oplus_{j=1}^{j=r} \sigma(y_j^i).$$

By modifying the original analysis of [16, Nisan Wigderson] it is shown in [14, Luby Veličković Wigderson] that $\Pr_{\mathcal{E}}[F]$ is within ϵ of $\Pr[F]$. The size of the sample space of $\mathcal{E}$ is $2^{O(\log^4((nm)/\epsilon))}$. Computing the average over all sample points of $\mathcal{E}$ we obtain the desired approximation.

We now show that the above algorithm can be transformed to a polynomial time ϵ-approximation algorithm for a DNF formula F whose clauses are not too long. To achieve this we first describe a reduction algorithm which transforms F to a DNF formula G such that $\Pr[G]$ is close to $\Pr[F]$ and such that G has substantially fewer clauses than F. The idea of the reduction algorithms is to repeatedly find large "sunflowers" of clauses and to discard all but their common center until no more large "sunflowers" can be found.

A family of sets $\mathcal{S}$ is called a *sunflower* if there is a set C, called the *center*, such that for every $A, B \in \mathcal{S}$ if $A \neq B$ then $A \cap B = C$. The following theorem was proved by [6, Erdös Rado]. We reproduce the proof for completeness and future reference.

Theorem 1 *Let t and m be positive integers, and let $\mathcal{F}$ be a family of sets such that every element of $\mathcal{F}$ has size at most t and $|\mathcal{F}| > t!(m-1)^t$. Then, $\mathcal{F}$ contains a sunflower of size m.*

PROOF: The statement is proved by induction on t. For $t = 1$ it is trivial. For the inductive step, given $\mathcal{F}$ let $\mathcal{S}$ be a maximal disjoint subfamily of $\mathcal{F}$. If $\mathcal{S}$ has size at least m we are done. Otherwise, the union of $\mathcal{S}$ has size at most $(m-1)t$ and intersects every member of $\mathcal{F}$. By the pigeon hole principle there is an $i \in \bigcup \mathcal{S}$ such that the family $\mathcal{G} = \{F \setminus \{i\} : i \in F \in \mathcal{F}\}$ has size bigger than $(t-1)!(m-1)^{t-1}$. Since every member of $\mathcal{G}$ has size at most $t-1$, by the inductive hypothesis $\mathcal{G}$ contains a sunflower $\mathcal{S}$ of size at least m with center S. Then, the family $\{F \cup \{i\} : F \in \mathcal{S}\}$ is a sunflower with center $S \cup \{i\}$. ■

It is clear that the above argument produces an efficient algorithm for finding the required sunflower.

Theorem 2 *There is an algorithm which given a DNF formula F with m clauses of size at most t, and $\epsilon > 0$ produces another DNF formula G with at most $t!\lceil 2^t \ln(m/\epsilon)\rceil^t$ clauses such that $|\Pr[G] - \Pr[F]| \leq \epsilon$. In addition, the clauses of G also have size at most t. The running time of the algorithm is polynomial in nm, and $1/\epsilon$.*

PROOF: For a clause C let the domain of C be the set of variables such that either they or their negation appear in C. The algorithm finds a collection $\mathcal{S}$ of clauses of F of size $\lceil 2^t \ln(m/\epsilon)\rceil$ such that their domains form a sunflower with center S. If $|S| = i$ then it finds a clause C on the variables in S such that the set $\mathcal{R}$ of clauses in $\mathcal{S}$ which extend C has size at least $\lceil 2^{t-i} \ln(m/\epsilon)\rceil$. Then, in F, it replaces all clauses in $\mathcal{R}$ by a single clause C, thus obtaining a new formula F_1. Note that, given that C is satisfied, the probability that none of the clauses in $\mathcal{R}$ is satisfied is at most $(1-2^{i-t})^{2^{t-i}\ln(m/\epsilon)} \leq \epsilon/m$. Moreover, if a clause in $\mathcal{R}$ is satisfied then so is C. Thus, $|\Pr[F_1] - \Pr[F]| \leq \epsilon/m$. The algorithm then repeats this procedure until no sufficiently large sunflowers can be found. Since there are m clauses to start with this can happen at most m times. Thus, if the last formula is F_l then $|\Pr[F_l] - \Pr[F]| \leq \epsilon$. The algorithm then returns $G = F_l$. ■

This algorithm together with the algorithm from [16, Nisan Wigderson] described above can be combined to give the following.

Theorem 3 *There is an algorithm which on input a DNF formula F on n variables with m clauses each of length at most $t = O(\log^{1/8}(nm))$ and an error parameter ϵ such that $1/\epsilon = 2^{O(\log^{1/4}(nm))}$ produces an ϵ-approximation of* $\Pr[F]$ *in time polynomial in nm.*

PROOF: First apply the sunflower reduction algorithm to F with error parameter $\epsilon/2$ to obtain a formula G such that $|\Pr[G] - \Pr[F]| \leq \epsilon/2$ and such that the the number of clauses in G is $t![2^t \ln(m/\epsilon)]^t = 2^{O(t^2)}$, each of length at most t. Next apply the algorithm described above to produce an $\epsilon/2$-approximation of $\Pr[G]$ and this is an ϵ-approximation of $\Pr[F]$. Since the total number of variables mentioned in G is at most $l = 2^{O(t^2)}$, and since the algorithm described above runs in time $2^{\log^4(l/\epsilon)}$, the running time is as claimed. ■

One might think that the above algorithm can be combined with the reduction from a relative to absolute error approximation described in Section 2 to yield an approximation algorithm with relative error at most ϵ that under the same conditions as stated in Theorem 3 runs in time polynomial in nm. However, in that reduction, to achieve a final relative error of ϵ, we needed to achieve an absolute error of ϵ/m in each of the m subproblems. For this error absolute error value, each subproblem takes time at least $2^{\log^4(m/\epsilon)}$ to solve, and this is not polynomial in nm even for constant ϵ.

5 The Simple Coloring Algorithm

We now develop a simple approximation algorithm for the DNF problem. Central to this development is the notion of a proper coloring of a DNF formula. A *k-coloring* of a set X is a function $f : X \rightarrow \{1, \ldots, k\}$. Given a subset A of X we say that f colors A *properly* if f restricted to A is 1-1. Suppose $X = \{x_1, \ldots, x_n\}$ is a set of n variables and G is a DNF formula on X with m clauses. We say f is a *proper* k-coloring of G if the domain of every clause of G is colored properly.

Suppose we have a proper k-coloring of a formula G. Let i be one of the colors and let H be a formula in the variables with color i obtained by fixing a truth assignment to all variables with color different than i. The main point is that each clause of H is of length one. We first show that if $\mathcal{D}_i$ is a probability distribution on the variables colored i with enough independence between the variables then the probability that H is satisfied with respect to $\mathcal{D}_i$ is very close to the probability that H is satisfied with respect to choosing the assignments to the variables colored i randomly and completely independently. We use this to show that $\Pr_{\mathcal{D}}[G]$ is approximately equal to $\Pr[G]$, where $\mathcal{D}$ is the product over each color class i of $\mathcal{D}_i$. This gives us a method for approximating $\Pr[G]$: enumerate all the sample points in $\mathcal{D}$ to compute $\Pr_{\mathcal{D}}[G]$. Since the size of $\mathcal{D}$ depends exponentially on the number of colors used, it is crucial to find a proper k-coloring of G where k is as small as possible. Given any DNF formula F, we show how to construct a subformula G of F together with a proper k-coloring of G such that k is small and such that $\Pr[G]$ is approximately $\Pr[F]$. Then, $\Pr_{\mathcal{D}}[G]$ is a good approximation of $\Pr[F]$, and we can calculate exactly $\Pr_{\mathcal{D}}[G]$ by enumerating all sample points in the small distribution $\mathcal{D}$.

5.1 All clauses of length one

Lemma 6 *Let F be a DNF formula on $\{0,1\}$-valued variables $\{x_1, \ldots, x_n\}$ where each clause is a single literal. Let $\mathcal{D}$ be any probability distribution where the variables are k-wise δ-dependent and uniformly distributed. Then,*

$$|\Pr_{\mathcal{D}}[F] - \Pr[F]| \leq 2^{-k} + \delta.$$

Furthermore,

$$|\Pr_{\mathcal{D}}[F] - \Pr[F]| \leq 2(2^{-k} + \delta)\Pr[F].$$

PROOF: If F is the empty formula then $\Pr[F] = \Pr_{\mathcal{D}}[F] = 0$. If F contains both a literal and its negation among its clauses then $\Pr_{\mathcal{D}}[F] = \Pr[F] = 1$. Suppose neither of these cases holds and that the number of literals in F is $m \geq 1$. If $m \leq k$ then, by the properties of $\mathcal{D}$, $|\Pr_{\mathcal{D}}[F] - \Pr[F]| \leq \delta$. On the other hand, if $m > k$ then $1 \geq \Pr[F] \geq 1 - 2^{-k}$ and, by the properties of $\mathcal{D}$, $1 \geq \Pr_{\mathcal{D}}[F] \geq 1 - 2^{-k} - \delta$. Thus, $|\Pr[F] - \Pr_{\mathcal{D}}[F]| \leq 2^{-k} + \delta$. Since $\Pr[F] = 1 - 2^{-m} \geq 1/2$ it also follows that $|\Pr_{\mathcal{D}}[F] - \Pr[F]| \leq 2(2^{-k} + \delta)\Pr[F]$. ■

5.2 The coloring lemma

Let f be a k-coloring of a DNF formula G on X, and let X_i be the set of variables colored i. Let $\mathcal{D}_i$ be a probability distribution on the variables in X_i and define

$$\epsilon_i = \max_z\{|\Pr[G(z)] - \Pr_{\mathcal{D}_i}[G(z)]|\},$$

where the maximum is over all truth assignments z to variables in $X \setminus X_i$ and where $G(z)$ is the DNF formula on X_i obtained by reducing G according to z. Let

$$\mathcal{D} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_k.$$

Then, $\mathcal{D}$ is a probability distribution on the set of all truth assignments on X. The following lemma is key to our approximation algorithm.

Lemma 7 $|\Pr_{\mathcal{D}}[G] - \Pr[G]| \leq \sum_{i \in \{1,\ldots,k\}} \epsilon_i$.

PROOF: Let $\mathcal{U}_i$ be the uniform distribution on the truth assignments to the variables in X_i, and let $\mathcal{V}_i = \mathcal{D}_1 \times \cdots \times \mathcal{D}_{i-1} \times \mathcal{U}_i \times \cdots \times \mathcal{U}_k$, for $i \in \{1, \ldots, k+1\}$. Then, $\Pr_{\mathcal{V}_1}[G] = \Pr[G]$ and $\Pr_{\mathcal{V}_{k+1}}[G] = \Pr_{\mathcal{D}}[G]$. Thus, to establish the lemma it suffices to prove, for $i \in \{1, \ldots, k\}$,

$$|\Pr_{\mathcal{V}_i}[G] - \Pr_{\mathcal{V}_{i+1}}[G]| \leq \epsilon_i. \quad (1)$$

Let z be any assignments to the variables in $X \setminus X_i$ and let $G(z)$ be the DNF formula on X_i obtained by reducing G according to z. By definition,

$$|\Pr_{\mathcal{D}_i}[G(z)] - \Pr_{\mathcal{U}_i}[G(z)]| \leq \epsilon_i.$$

To prove (1), let $\mathcal{W}_i = \mathcal{D}_1 \times \cdots \times \mathcal{D}_{i-1} \times \mathcal{U}_{i+1} \times \cdots \times \mathcal{U}_k$. $\mathcal{W}_i$ is a probability distribution on the set of all truth assignments to the variables in $X \setminus X_i$. Then, $\Pr_{\mathcal{V}_i}[G] = \mathrm{E}_{z \in \mathcal{W}_i}[\Pr_{\mathcal{U}_i}[G(z)]]$ and $\Pr_{\mathcal{V}_{i+1}}[G] = \mathrm{E}_{z \in \mathcal{W}_i}[\Pr_{\mathcal{D}_i}[G(z)]]$. Hence, $|\Pr_{\mathcal{V}_i}[G] - \Pr_{\mathcal{V}_{i+1}}[G]|$ is at most

$$\mathrm{E}_{z \in \mathcal{W}_i}[|\Pr_{\mathcal{U}_i}[G(z)] - \Pr_{\mathcal{D}_i}[G(z)]|] \leq \epsilon_i.$$

This finishes the proof of (1) and the lemma. ■

5.3 Finding proper colorings

Lemma 7 can be used to approximate $\Pr[G]$ given a proper k-coloring of G, for a small value of k. We can use one of the probability spaces described in Section 3 for each $\mathcal{D}_i$ and then enumerate all sample points in the resulting product probability distribution $\mathcal{D}$ to calculate $\Pr_{\mathcal{D}}[G]$. Lemma 7 shows that $\Pr_{\mathcal{D}}[G]$ is a good approximation of $\Pr[G]$.

In this subsection we discuss how to find proper k-colorings of a given DNF formula F for small values of k. There is an inherent problem with this, because even if we consider coloring only the class of DNF formula with clauses of length two then the resulting coloring problem is the same as the coloring problem for graphs, and thus the smallest value of k possible can be linear in the number of variables, e.g., a proper coloring of the complete graph requires a distinct color for each vertex. Let t be the length of the longest clause in F. Let $\delta > 0$ be an error parameter and let k be t^2/δ rounded up to the nearest power of two. Instead, we construct a small set $\mathcal{C}$ of k-colorings of F such that each clause of F is properly colored for a $1 - \delta$ fraction of the colorings in $\mathcal{C}$. Let G_f be the DNF formula which contains exactly the clauses of F colored properly with respect to $f \in \mathcal{C}$. We show that $\mathrm{E}_{f \in \mathcal{C}}[\Pr[G_f]] \geq (1 - \delta)\Pr[F]$. From this the algorithm follows: For each $f \in \mathcal{C}$, compute $\Pr_{\mathcal{D}}[G_f]$ and then output the maximum value of this quantity.

The intuition for this choice of k is that for each particular clause, in a randomly chosen k-coloring of the variables the clause is properly colored with probability at least $1 - \delta$.

Lemma 8 *If f is a randomly chosen k-coloring of the variables where the choices of the colors for variables is pairwise independent, then the probability any particular clause is properly colored is at least* $1-\delta$.

PROOF: The probability that a particular pair of variables receive the same color is $1/k \leq 2\delta/t^2$. Since there are at most $\binom{t}{2}$ pairs of variables in the clause, the probability that some pair receives the same color is at most δ. ■

We say a set of colorings $\mathcal{C}$ is *δ-good* if each particular clause is properly colored in at least a $1-\delta$ fraction of the colorings. By Lemma 8 and using Lemma 1 given in Section 3, we can construct a δ-good set of colorings of size $O(\max\{n, t/(2\delta)\}^2)$.

We now present a different way of obtaining in polynomial time a set of size $s = \lceil \log(m)/\delta \rceil$ of k-colorings which is 5δ-good. We use the algorithm described in the following lemma as a subroutine in our overall algorithm to achieve this goal.

Lemma 9 *There is a polynomial time algorithm which given a positive weight w_j for each clause C_j, produces a k-coloring f such that the sum of the weights of clauses colored properly by f is at least* $(1-\delta)\sum_{j\in\{1,\ldots,m\}} w_j$.

PROOF: Consider a greedy algorithm which colors the variables consecutively in the following way. At stage i the variable x_i is colored in such a way that the total weight z_i of the clauses that become colored improperly at stage i is the least possible. If u_i is the total weight of the clauses that contain x_i then z_i is at most $u_i t/k$. Let z be the total weight of the clauses that are colored improperly. Then:

$$z = \sum_{i\in\{1,\ldots,n\}} z_i \leq \frac{t}{k}\cdot \sum_{i\in\{1,\ldots,n\}} u_i \leq \frac{t^2}{k}\cdot \sum_{j\in\{1,\ldots,m\}} w_j \leq \delta \cdot \sum_{j\in\{1,\ldots,m\}} w_j.$$

■

Lemma 10 *Let $\delta \leq 1/4$. There is an algorithm that on input a DNF formula F produces a set of 5δ-good k-colorings of size s.*

PROOF: The overall algorithm is obtained as follows. Initially all the clause weights are set to $1/m$, and then we apply the algorithm described in Lemma 9 repeatedly s times to produce the set of k-colorings. After each call the weight of all the clauses not colored properly at that stage is multiplied by 2 and the weight of each clause colored properly is multiplied by $(1-2\delta)/(1-\delta)$. Let α_i be the sum of the weights of the clauses after i k-colorings have been produced. By Lemma 9 and the way the weights are adjusted after each coloring is produced,

$$\alpha_{i+1} \leq 2\delta\alpha_i + \left(\frac{1-2\delta}{1-\delta}\right)(1-\delta)\alpha_i = \alpha_i.$$

Thus, for all i, $\alpha_i \leq \alpha_0 = 1$. Note that because $\delta < 1/2$ all clause weights remain positive. Thus, at any point in time, the weight of a particular clause is at most 1. If a clause is colored properly by l of the colorings and colored improperly by the remaining $s - l$ colorings then the weight of the clause at the termination of the algorithm is at least

$$\frac{1}{m} \cdot 2^l \left(\frac{1-2\delta}{1-\delta} \right)^s .$$

Since this quantity is at most 1, it follows that $l \leq s \log((1-\delta)/(1-2\delta)) + \log(m)$. Using the fact that $\ln(z) \leq z - 1$ for all positive z, it follows that $l \leq s \log(e)\delta/(1-2\delta) + \log(m)$ and, since $\log(e) \leq 2$, and since $\delta \leq 1/4$ implies that $1/(1-2\delta) \leq 2$, it follows that this is at most $4\delta s + \log(m)$. Because $(4\delta s + \log(m))/s \leq 5\delta$, the set of k-colorings is 5δ-good. ■

Let $\mathcal{C}$ be a family of δ-good colorings. Given $f \in \mathcal{C}$ let G_f be the DNF formula obtained by taking the disjunction of the clauses in F which are properly colored by f. Then, $\Pr[G_f]$ is at most $\Pr[F]$. We shall need the following.

Lemma 11 $\mathrm{E}_{f\in\mathcal{C}}[\Pr[G_f]] \geq (1-\delta)\Pr[F]$.

PROOF: Let $\mathcal{T}$ be the set of all truth assignments which satisfy F. Then, $\mathrm{E}_{f\in\mathcal{C}}[\Pr[G_f]]$ is equal to $\sum_{\tau\in\mathcal{T}} \Pr[\tau]\mathrm{E}_{f\in\mathcal{C}}[\tau \text{ satisfies } G_f]$ which is at least $\Pr[F](1-\delta)$. ■

5.4 The algorithm

Theorem 4 *There is a deterministic algorithm which on input a DNF formula F on n variables with m clauses, and $\epsilon > 0$, $\delta > 0$ produces an estimate Y such that*

$$(1-\delta)\Pr[F] - \epsilon \leq Y \leq \Pr[F] + \epsilon.$$

The running time of the algorithm is polynomial in nm and

$$\left(\frac{\log(nm)}{\delta\epsilon} \right)^{\log^2(m/\epsilon)/\delta} .$$

PROOF: Let $t = \lceil \log(2m/\epsilon) \rceil$, $k = \lceil 2t^2/\delta \rceil$, $\rho = \frac{\epsilon}{4k}$ and $l = \lceil \log(1/\rho) \rceil$. Let X be the set of variables. Let G be the subformula of F obtained by taking the disjunction of all clauses in F of length at most t. Then, $|\Pr[G] - \Pr[F]| \leq \epsilon/2$. Let $\mathcal{C}$ be a δ-good set of k-colorings for G constructed as described just after Lemma 8. If $f \in \mathcal{C}$ then $\Pr[G_f] \leq \Pr[G] \leq \Pr[F]$, and Lemma 11 guarantees that there is an f in $\mathcal{C}$ such that

$$\Pr[G_f] \geq (1-\delta)\Pr[G] \geq (1-\delta)\Pr[F] - \epsilon/2.$$

The algorithm proceeds as follows. Given f in $\mathcal{C}$ let X_i^f be the set of variables in X which are colored i by f. Let $\mathcal{D}_i^f$ be an l-wise ρ-dependent probability distribution on the truth assignments to the variables in X_i^f as described in Lemma 4 of Section 3 which has size s which is polynomial in 2^l, $1/\rho$ and $\log(n)$. Let $\mathcal{D}^f = \prod_{i\in\{1,\ldots,k\}} \mathcal{D}_i^f$ be the probability distribution on total truth assignments with sample space size s^k. Lemma 6 and Lemma 7 together imply that

$$|\Pr_{\mathcal{D}^f}[G_f] - \Pr[G_f]| \leq k(2^{-l} + \rho) \leq \epsilon/2.$$

Thus, the algorithm simply queries each sample point of $\mathcal{D}^f$ and computes Y_f as the fraction of sample points which satisfy G_f. To compute the estimate Y the algorithm repeats this procedure for each $f \in \mathcal{C}$ and outputs $Y = \max\{Y_f : f \in \mathcal{C}\}$.

The total running time of the algorithm, which includes constructing the sample space and computing the fraction of truth assignments that satisfy the formula, is as claimed in the statement of the theorem. ■

For a fixed value of δ, the running time is polynomial in nm and

$$\left(\frac{\log(nm)}{\epsilon}\right)^{\log^2(m/\epsilon)},$$

and for fixed values for both ϵ and δ the running time is polynomial in nm and

$$\log(nm)^{\log^2(m)}.$$

We now present a relative error version of Lemma 7. For this presentation, we assume the notation established just before the statement of Lemma 7.

Lemma 12 $\Pr[G](\prod_{i\in\{1,\ldots,k\}}(1-2\epsilon_i)) \leq \Pr_{\mathcal{D}}[G] \leq \Pr[G](\prod_{i\in\{1,\ldots,k\}}(1+2\epsilon_i))$.

PROOF: We retain the notation established in the proof of Lemma 7. Fix $z \in \mathcal{W}_i$ and consider the DNF formula $G(z)$ with all clauses of length 1. Similar to the proof of Lemma 6, if $G(z)$ is the empty formula or $G(z)$ is identically true then $\Pr_{\mathcal{U}_i}[G(z)] = \Pr_{\mathcal{D}_i}[G(z)]$. Otherwise, $G(z)$ contains $m \geq 1$ literals, and thus $\Pr_{\mathcal{U}_i}[G(z)] \geq 1/2$. This, together with $|\Pr_{\mathcal{U}_i}[G(z)] - \Pr_{\mathcal{D}_i}[G(z)]| \leq \epsilon_i$ implies that

$$\Pr_{\mathcal{U}_i}[G(z)](1-2\epsilon_i) \leq \Pr_{\mathcal{D}_i}[G(z)] \leq \Pr_{\mathcal{U}_i}[G(z)](1+2\epsilon_i).$$

This in turn implies that

$$\Pr_{\mathcal{V}_i}[G](1-2\epsilon_i) \leq \Pr_{\mathcal{V}_{i+1}}[G] \leq \Pr_{\mathcal{V}_i}[G](1+2\epsilon_i).$$

Multiplying together the k inequalities yields the result. ■

As the following theorem shows, Lemma 12 can be used to obtain a good relative estimate of $\Pr[F]$ when the maximum clause length t in F is not too long.

Theorem 5 *There is a deterministic algorithm which on input DNF formula F with maximum clause length t and $\beta > 0$ produces an estimate Y such that*

$$(1-\beta)\Pr[F] \leq Y \leq (1+\beta)\Pr[F].$$

The running time of the algorithm is polynomial in nm and

$$\left(\frac{\max\{t, \log(n)\}}{\beta}\right)^{t^2/\beta}.$$

PROOF: The proof is analogous to the proof of Theorem 4. The differences are that we omit the first step of that algorithm that throws away clauses that are too long, and we use Lemma 12 in place of Lemma 7 in the analysis. To achieve total relative error β, we set $\epsilon = \delta = \beta/3$ in the proof of Theorem 4. The settings of the parameters are $k = \lceil 6t^2/\beta \rceil$ (this ensures that $\delta = \beta/3$), $\rho = \frac{\beta}{12k}$ and $l = \lceil \log(1/\rho) \rceil$ (this ensures that $1+\epsilon = (1+2(2^{-l}+\rho))^k \leq 1+\beta/3$). ■

The following theorem is a corollary of Theorem 5. This theorem is better than Theorem 3 because the maximum clause length allowed is larger and because the algorithm produces a constant relative error approximation as opposed to just an absolute error approximation.

Theorem 6 *There is a deterministic algorithm which on input DNF formula F with maximum clause length*

$$t = O\left(\left(\frac{\log(nm)}{\log\log(nm)}\right)^{1/2}\right)$$

and a constant $\beta > 0$ produces an estimate Y such that

$$(1-\beta)\Pr[F] \leq Y \leq (1+\beta)\Pr[F].$$

The running time of the algorithm is polynomial in nm.

6 The General Coloring Algorithm

Based on the ideas developed in the previous section, we develop a more efficient algorithm for the DNF problem. The major conceptual difference between the more efficient algorithm and the algorithm already introduced is a liberalized notion of a proper coloring. Based on this, we introduce generalizations of Lemma 6 and Lemma 8. We put together these lemmas analogously to what we did previously, where Lemma 7 is the ingredient used to glue them together.

A (k, c)-coloring for a DNF formula G is a k-coloring of the variables such that no clause in G contains more than c variables of the same color. Suppose we have a (k, c)-coloring of a formula G. Let i be one of the colors and let H be a formula in the variables with color i obtained from G by fixing a truth assignment to all variables with color different than i. The main point is that each clause of H is of length at most c. We first show that if $\mathcal{D}_i$ is a probability distribution on the variables colored i with enough independence between the variables then the probability that H is satisfied with respect to $\mathcal{D}_i$ is very close to the probability that H is satisfied with respect to choosing the assignments to the variables colored i randomly and completely independently. We apply Lemma 7 to show that $\Pr_{\mathcal{D}}[G]$ is approximately equal to $\Pr[G]$, where $\mathcal{D}$ is the product over each color class i of $\mathcal{D}_i$. This gives us a method for approximating $\Pr[G]$: enumerate all the sample points in $\mathcal{D}$ to compute $\Pr_{\mathcal{D}}[G]$. Since the size of $\mathcal{D}$ depends exponentially on k, and since the size of the sample space of $\mathcal{D}_i$ depends exponentially on c, it is crucial to choose the values of k and c carefully so as to minimize the total size of the final sample space. Given any DNF formula F, we show how to construct a subformula G of F together with a (k, c)-coloring of G such that $\Pr[G]$ is approximately $\Pr[F]$. Then, $\Pr_{\mathcal{D}}[G]$ is a good approximation of $\Pr[F]$, and we can calculate exactly $\Pr_{\mathcal{D}}[G]$ by enumerating all sample points in the small distribution $\mathcal{D}$.

We first derive an approximation algorithm for DNF formulas in the case that the maximum length of each clause is small. [3, Ajtai Wigderson] show that when the maximum clause length is constant there is an ϵ-approximation algorithm with running time polynomial in nm and $1/\epsilon$. We present a version of their algorithm which uses distributions with limited amount of independence. A similar analysis was found independently by Noam Nisan.

Theorem 7 *Let F be a DNF formula with clauses of length at most t. Let $\mathcal{D}$ be an l-wise δ-dependent distribution on $\{0,1\}$-valued uniformly distributed random variables $\{x_1, \dots, x_n\}$. Then,*

$$|\Pr_{\mathcal{D}}[F] - \Pr[F]| \leq e^{-\frac{l}{t2^t}} + 2^l \delta.$$

Also,

$$|\Pr_{\mathcal{D}}[F] - \Pr[F]| \leq 2^t (e^{-\frac{l}{t2^t}} + 2^l \delta) \Pr[F].$$

PROOF: We assign to each node σ of the full binary tree $\{0,1\}^{<l}$ a literal y_σ. If σ is in $\{0,1\}^{\leq l}$ let z_σ be the partial assignment to the variables appearing on the path from the root to σ such that if $i < |\sigma|$ then $y_{\sigma\restriction i}$ is made true by z_σ iff $\sigma(i) = 1$. The assignment is done recursively as follows. Suppose $\sigma \in \{0,1\}^{<l}$ is a node which has not yet been assigned a literal but all of its ancestors have been assigned a literal. If σ is the root or the last bit of σ is 0 and if there is a clause C which is not made false by z_σ, then set $y_\sigma = a_0$, $y_{\sigma\hat{}1} = a_1, \dots, y_{\sigma\hat{}1^{k-1}} = a_{k-1}$, where $a_0, \dots, a_{k-1}$ are the literals in C which do not appear along the path to σ and where 1^j is the string of j 1's. If no such clause exists, then z_σ falsifies all clauses in F, in which case the assignment of literals to σ and all of its descendants is made arbitrarily, consistent with the rule that no path contains two literals

associated with the same variable. Similarly, if the last bit of σ is 1, then the assignment to σ is made arbitrarily subject to the rule that no path contains two literals associated with the same variable.

For $\sigma \in \{0,1\}^l$ let S_σ be the set of all total assignments which extend z_σ. Then, the S_σ, for $\sigma \in \{0,1\}^l$, form a partition of the set of all total truth assignments and each set has size 2^{n-l}. We label σ TRUE if all assignments in S_σ satisfy F, FALSE if all the assignments on S_σ do not satisfy F, and $*$ otherwise. Let A_0 be the set of all $\sigma \in \{0,1\}^l$ labeled TRUE and let A_1 be the set of all $\sigma \in \{0,1\}^l$ labeled either TRUE or $*$. For $i \in \{0,1\}$ let $T_i = \cup\{S_\sigma : \sigma \in A_i\}$. Then, every assignment in T_0 satisfies F, and every assignment that satisfies F is in T_1, and thus $\Pr[T_0] \leq \Pr[F] \leq \Pr[T_1]$. Note that if a string $\sigma \in \{0,1\}^l$ contains at least t consecutive 1's then it cannot be labeled $*$. The percentage of strings that do not contain t consecutive 1's is at most $(1-1/2^t)^{l/t} \leq e^{-\frac{l}{t2^t}}$. Thus, $\Pr[T_1] - \Pr[T_0] \leq e^{-\frac{l}{t2^t}}$.

Let $\Pr_{\mathcal{D}}[S_\sigma]$ denote the probability that a random assignment chosen according to $\mathcal{D}$ is in S_σ. Then, by the l-wise δ-dependence of $\mathcal{D}$, $|\Pr_{\mathcal{D}}[S_\sigma] - 1/2^l| \leq \delta$, for $\sigma \in \{0,1\}^l$. On the other hand, for $i \in \{0,1\}$, $|\Pr_{\mathcal{D}}[T_i] - \Pr[T_i]|$ is equal to

$$|\sum_{\sigma \in A_i} \Pr_{\mathcal{D}}[S_\sigma] - \sum_{\sigma \in A_i} \Pr[S_\sigma]| \leq \delta |A_i| \leq \delta 2^l.$$

Clearly, $\Pr_{\mathcal{D}}[T_0] \leq \Pr_{\mathcal{D}}[F] \leq \Pr_{\mathcal{D}}[T_1]$. From this, we can conclude that

$$\begin{aligned} |\Pr_{\mathcal{D}}[F] - \Pr[F]| &\leq \max\{\Pr[T_1] - \Pr_{\mathcal{D}}[T_0], \Pr_{\mathcal{D}}[T_1] - \Pr[T_0]\} \\ &\leq \Pr[T_1] - \Pr[T_0] + \max_{i \in \{0,1\}} |\Pr[T_i] - \Pr_{\mathcal{D}}[T_i]| \\ &\leq e^{-\frac{l}{t2^t}} + \delta 2^l \end{aligned}$$

as desired. Because $\Pr[F] \geq 2^{-t}$, we can also conclude that

$$|\Pr_{\mathcal{D}}[F] - \Pr[F]| \leq 2^t(e^{-\frac{l}{t2^t}} + 2^l\delta)\Pr[F].$$

■

The following corollary is due to [3, Ajtai Wigderson]. We prove it using a different algorithm.

Corollary 13 *Let t be a fixed constant. There is a deterministic algorithm which given a DNF formula F on n variables with clauses of length at most t, and $\epsilon > 0$, produces an ϵ-relative approximation of* $\Pr[F]$ *in time polynomial in nm and $1/\epsilon$.*

PROOF: We first show that there is an algorithm which produces an ϵ-approximation with the claimed running time, and then make the parameter adjustments to produce an ϵ-relative approximation. Let $l = \lceil t2^t \log(2/\epsilon) \rceil$ and let $\delta = \epsilon 2^{-(l+1)}$. By Lemma 4 described in Section 3, there is a probability distribution $\mathcal{D}$ on the set of truth assignments such that the random variables $\{x_1, \ldots, x_n\}$ are l-wise δ-dependent and whose sample space has size polynomial in $\log(n)$, 2^l and $1/\delta$. Thus, the sample space is of size polynomial in n and $1/\epsilon$. The algorithm then simply queries all the assignments in the sample space and outputs the ratio of the assignments satisfying F. By Theorem 7 this ratio is within ϵ of $\Pr[F]$. The total running time is polynomial in nm and $1/\epsilon$. To obtain the ϵ-relative approximation, note that if F contains at least one clause then $\Pr[F]$ is at least 2^{-t}. Thus, if we increase l by a multiplicative factor of t and decrease δ by a multiplicative factor of 2^{-t} the approximation has relative error at most ϵ. ■

Theorem 8 *There is a deterministic algorithm which given a DNF formula F on n variables with m clauses and $\epsilon > 0$ outputs an estimate Y such that $|\Pr[F] - Y| \leq \epsilon$. For a fixed ϵ the running time of the algorithm is $(m \log(n))^{\exp(O(\sqrt{\log\log(m)}))}$.*

PROOF: The algorithm is very similar to the one described in the proof of Theorem 4. We only indicate the differences. As described in the introduction to this section we shall use the concept of a (k, c)-coloring of a DNF formula.

Let F be a given DNF formula on n variables and m clauses, and suppose we are given $\epsilon > 0$. We may assume that $\epsilon < 1/4$. Let X denote the set of variables of F. Set $t = \lceil \log(\frac{2m}{\epsilon}) \rceil$ and let G be the formula obtained from F by deleting the clauses of length at least t. Then $|\Pr[G] - \Pr[F]| \leq \frac{\epsilon}{2}$. We shall in fact produce an $\frac{\epsilon}{2}$-approximation to $\Pr[G]$.

Let $c = \left\lceil \sqrt{\log(t)} \right\rceil$ and let $k = \lceil t 2^{c+1}/\epsilon \rceil$. Consider now a randomly chosen k-coloring f of the variables of G where each color is chosen equally likely and the choice of colors is c-wise independent. For any given clause of G the probability that there are c variables from this clause which receive the same color is at most

$$\binom{t}{c} \frac{1}{k^{c-1}} \leq \frac{t^c}{k^{c-1}} \leq 2\epsilon^{c-1}.$$

For sufficiently large t this quantity is less than $\frac{\epsilon}{4}$. Let us say that f *c-properly colors* a clause if there are no c variables appearing in this clause which receive the same color. As in Section 5.3 we produce an efficiently constructible set $\mathcal{C}$ of k-colorings of size $O(n^c)$ such that every clause of G is c-properly colored by at least $1 - \epsilon/4$ fraction of members of $\mathcal{C}$. For $f \in \mathcal{C}$ let G_f denote the disjunction of the clauses of G which are c-properly colored by f. As before, using Lemma 11 we can show that there is an f in $\mathcal{C}$ such that

$$\Pr[G_f] \geq (1 - \epsilon/4) \Pr[G] - \epsilon/2 \geq \Pr[F] - 3\epsilon/4.$$

Let $l = \lceil \log(\frac{8k}{\epsilon}) \rceil c 2^c$ and let $\delta = \epsilon(8k2^l)^{-1}$. Given a coloring $f \in \mathcal{C}$ let X_i^f be the set of variables which are colored i by f. Let $\mathcal{D}_i^f$ be an l-wise δ-dependent probability distribution on the truth assignments to the variables in X_i^f as constructed in Lemma 4. The size of the sample space of $\mathcal{D}_i^f$ is polynomial in $\log(n)$, 2^l, and $1/\delta$. For a truth assignment z to the variables in $X \setminus X_i^f$ let $G_f(z)$ be the formula obtained by reducing G_f according to z. Then the clauses of $G_f(z)$ have size at most c. Therefore by Theorem 7 for every such z

$$|\Pr_{\mathcal{D}_i^f}[G_f(z)] - \Pr[G(z)]| \leq e^{-\frac{l}{c2^c}} + 2^l\delta \leq \frac{\epsilon}{4k}.$$

Let now $\mathcal{D}^f = \times_{i\in\{1,\ldots,k\}}\mathcal{D}_i^f$. Then by Lemma 7 we have that $|\Pr_{\mathcal{D}}^f[G_f] - \Pr[G_f]| \leq \epsilon/4$. The size of the sample space of $\mathcal{D}$ is $(\frac{2^l \log(n)}{\delta})^{O(k)}$. For a constant ϵ this is $(m\log(n))^{\exp(O(\sqrt{\log\log(m)}))}$.

The approximation algorithm is now the following. For each $f \in \mathcal{C}$ the algorithm computes Y_f as the fraction of sample points of $\mathcal{D}^f$ which satisfy G_f. The algorithm then outputs $Y = \max\{Y_f : f \in \mathcal{C}\}$. It follows then that $|\Pr[F] - Y| \leq \epsilon$. The running time of the algorithm is proportional to the sum of the sizes of the $\mathcal{D}^f$ and is therefore as claimed in the statement of the theorem. ■

Note that this argument yields, for any constant $\epsilon > 0$, a deterministic polynomial time algorithm for finding an ϵ-relative approximation to $\Pr[F]$ for DNF formulas F which have clauses of length at most

$$\frac{\log(mn)}{\exp\left(O\left(\sqrt{\log\log(mn)}\right)\right)} = \log^{1-o(1)}(mn).$$

7 General Distributions

In this section we consider the case when we want to approximate the probability that a DNF F formula is satisfied when the distribution on the input variables are not necessarily uniformly distributed. By considering the k-th term of the inclusion-exclusion formula, [7, Even Goldreich Luby Nisan Veličković] proved the following generalization of Lemma 6.

Lemma 14 *Let F be a DNF formula on $\{0,1\}$-valued variables $\{x_1,\ldots,x_n\}$ where each clause is a single literal. Let $\mathcal{V}$ be a probability distribution such that the random variables are completely independent but not necessarily uniformly distributed and let $\mathcal{D}$ be k-wise independent distribution such that for each i, $\Pr_{\mathcal{V}}[x_i = 1] = \Pr_{\mathcal{D}}[x_i = 1]$. Then, $|\Pr_{\mathcal{D}}[F] - \Pr_{\mathcal{V}}[F]| \leq 2^{-\Omega(k)}$. Even stronger, $|\Pr_{\mathcal{D}}[F] - \Pr_{\mathcal{V}}[F]| \leq 2^{-\Omega(k)}\Pr_{\mathcal{V}}[F]$.*

Consider the general DNF approximation problem when probabilities $p_1,\ldots,p_n$ are specified for the variables $x_1,\ldots,x_n$ along with F as part of the input to the problem. The problem is to approximate $\Pr[F]$, where $\Pr[F]$ is the probability that F is satisfied by a random independently chosen setting of the variables, where $\Pr[x_i = 1] = p_i$.

An interesting version of the problem with respect to applications is when, for each i, p_i is not too large, i.e., $p_i \leq \frac{1}{2}$, and F is monotone, i.e., no variable appears negated in any clause. A typical example is when the variables correspond to individual components of a complex system, a variable taking on the value 1 indicates that the corresponding component fails, clauses correspond to minimal groups of components whose collective failure makes the entire system fail and thus truth assignments that satisfy the DNF formula correspond to failure states of the system (see e.g., [11, Karp Luby Madras]). Usually, the individual components are fairly reliable, in which case the failure probability p_i of component i is small.

For this version of the problem, Lemma 14 can be used in place of Lemma 6 (Theorem 7, respectively) to derive a deterministic approximation algorithm using the methods developed in Section 5 (Section 6, respectively) with basically the same (slightly worse) running times as for the previously developed algorithms.

Another interesting version of the problem is when the maximum length of any clause in F is $O(\log(mn))$; for this version essentially the same results as described in the previous paragraph can be obtained.

8 Conclusions and Open Problems

One obvious open problem is to develop a polynomial time approximation algorithm for DNF with no restrictions on the length of the clauses. One possible approach to this goal, which was one of the motivations for the work presented in this paper, was the following conjecture. There is a constant c such that if $k = c\log(m/\epsilon)$ and $\mathcal{D}$ is k-wise independent then $|\Pr_{\mathcal{D}}[F] - \Pr[F]| \leq \epsilon$. Conjectures similar to this have also been made in [12, Linial Nisan]. If this conjecture were true in a slightly weaker form, i.e., if it were true with respect to distributions that are "almost" independent in the sense of [15, Naor Naor], this would immediately imply yield a desired algorithm: construct $\mathcal{D}$ and compute $\Pr_{\mathcal{D}}[F]$ in polynomial time, and this is provably a good approximation of $\Pr[F]$.

Subsequently to our work Yishay Mansour found the following counterexample to this conjecture. Consider the following function defined on n boolean valued random variables $\{x_1, \ldots, x_n\}$. Let $l = \log(n)$ and $k = \log(1/\epsilon)$. Partition the first lk random variables into k blocks of l variables each, and let the function F be the disjunction over all blocks $i \in \{1, \ldots, k\}$ of the parity of the bits in the i^{th} block, i.e., F takes on value 1 if there is at least one block whose parity is 1. Note that F can be written as a DNF formula with $m = n\log(1/\epsilon)$ clauses. Furthermore, $\Pr[F] = 1 - (1/2)^k = 1 - \epsilon$. Let $\mathcal{D}$ be the uniform distribution on all $\{0,1\}$-assignments to random variables $\{x_1, \ldots, x_n\}$ for which the parity of the first lk variables is equal to 1. It is easy to see that $\{x_1, \ldots, x_n\}$ are $(lk-1)$-wise independent in $\mathcal{D}$. On the other hand, since each setting under this restriction makes the parity of at least one of the blocks equal to 1, it follows that $\Pr_{\mathcal{D}}[F] = 1$, and thus $|\Pr[F] - \Pr_{\mathcal{D}}[F]| = \epsilon$. Thus, there is a distribution $\mathcal{D}$ where the error is ϵ even when the random variables are $\Omega(\log(1/\epsilon)\log(m))$-independent, contradicting the conjecture. This still leaves open the existence of other polynomial time constructible distributions which approximate well the DNF problem.

We point out that the coloring algorithm presented above can also be used as a probabilistic procedure for estimating $\Pr[F]$ using a total number of random bits which is substantially less than n.

9 Acknowledgments

We thank Oded Goldreich and Noam Nisan for their contributions to this work. We thank Yishay Mansour for contributing the counterexample presented in Section 8 and other useful remarks.

References

[1] Alon, N., Goldreich, O., Håstad, J., Peralta, R., "Simple constructions of almost k-wise independent random variables", *FOCS 90.*

[2] Ajtai, M., "$\sum_1^1$-Formulae on Finite Structures", *Annals of Pure and Applied Logic*, 24, 1983, pp. 1-48.

[3] Ajtai, M., Wigderson, A., "Deterministic Simulation of Probabilistic constant depth circuits", *FOCS 85.*

[4] Alon, N., L. Babai, A. Itai, "A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem", *Journal of Algorithms*, 7, pp. 567–583, 1986.

[5] Boppana, R., Hirschfeld, R., "Pseudo-random generators and complexity classes", S. Micali, ed., Advances in Computer Research, vol. 5, pp. 1-26, JAI Press, 1989.

[6] Erdös, P., Rado, R, "Intersection theorems for systems of sets", Journal of the London Math. Society, vol.35, pp.85-90.

[7] Even, G., Goldreich, O., Luby, M., Nisan, N., Veličković, B., "Approximations of General Independent Distributions", 24th *STOC*, 1992.

[8] Furst, M., Saxe, J., Sipser, M., "Parity, Circuits and the Polynomial Time Hierarchy", *FOCS 81.*

[9] Håstad, J., "Computational limitations for small depth circuits", Ph.D. thesis, M.I.T. press, 1986.

[10] Karp, R., Luby, M., "Monte-Carlo algorithms for enumeration and reliability problems", *STOC 83*.

[11] Karp., R., Luby, M., Madras, N., "Monte-Carlo Approximation Algorithms for Enumeration Problems," *J. of Algorithms*, Vol. 10, No. 3, Sept. 1989, pp. 429-448.

[12] Linial, N., Nisan, N., "Approximate Inclusion-Exclusion", *STOC 90*.

[13] Luby, M., "A Simple Parallel Algorithm for the Maximal Independent Set Problem", *STOC 85*, pp. 1-10, *SIAM J. Computing*, Vol. 15, No. 4, November 1986, pp. 1036-1053.

[14] Luby, M., Veličković, B., Wigderson, A., "On Deterministic Approximate Counting of Solutions to Polynomials", paper in preparation.

[15] Naor, M., Naor, S., "Small Bias Probability Spaces: Efficient Constructions and Applications", *STOC 90*.

[16] Nisan, N., Wigderson, A., "Hardness vs. Randomness", *FOCS 88*, pp. 2-11.

[17] Razborov, A., "Lower bounds on the monotone complexity of some Boolean functions," *Doklady Akademii Nauk SSSR* 281:4, 1985, pp. 798–801, (in Russian). English translation in *Soviet Mathematics Doklady* 31, 354–357.

[18] Valiant, L. G., "The complexity of computing the permanent", *Theoretical Computer Science*, 1979, No. 8, pp 189-201.

[19] Vazirani, U., "Randomness, adversaries and computation", Ph.D. thesis, UC Berkeley, 1986.

[20] Weil, A., "Sur les courbes algébriques et les variétés qui s'en déduisent", *Actualités Sci. Ind.* No. 1041, 1948.

[21] Yao, A., "Separating the Polynomial-Time Hierarchy by Oracles", *FOCS 85*, pp. 1-10.