DOCCOURSE Project 2006

Derandomizing hitting sets for read-once CNF and DNF formulae

Massimo Lauria *
lauria@di.uniroma1.it

March 21, 2006

Abstract

It is well known that for any constant $\epsilon > 0$, exists a polynomial hitting set for any exponential size class of boolean functions whose fraction of satisfying assignment (the weight) is at least ϵ. The derandomization process consists in finding a deterministic polynomial time construction of such set, whose existence is proved by simple probabilistic arguments. We show how to do this for some specific classes of boolean functions: CNF and DNF formulae where each variable appears in at most one clause. We also assume that the weight is greater and bounded away from $\frac{1}{e}$ (for CNFs) and from 0 (for DNFs). We show that an almost $O(\log n)$-wise independent sample space is a hitting set for these classes. Finally we present some deterministic polynomial time constructions for such space.

Introduction

We are interested in derandomizing the construction of a polynomial-size *hitting set* for classes of boolean functions. A boolean function f from $\{0,1\}^n$ to $\{0,1\}$ can be viewed as a subset of the space of the assignment to the n variables. For $x = x_1 \cdots x_n \in \{0,1\}^n$ we say that $x \in f$ if and only if $f(x) = 1$. The problem addressed in this work that of is finding a *hitting set* of assignments for a given class of functions. Such set must "hit" (i.e. satisfy with at least one element) every function in the given class.

Some observations about the problem (see Section 1 for a proper definition): first of all, not every class has a hitting set. Any class that contains a the function 0 has no hitting set. Also it is difficult to decide if hitting sets exist given a general class: deciding this for all classes with just one boolean formula is essentially the SAT problem. Furthermore even if such a set exists maybe its size is at least exponential. The class of boolean formulae $l_1 \wedge l_2 \wedge \cdots \wedge l_n$ where l_i is either x_i or $\neg x_i$ has clearly just one hitting set of exponential size: the full space of assignments. For some other cases the choice is trivial: any polynomial class of non-zero functions has a polynomial hitting set, even if can be hard to compute; the class of monotone functions has the easy hitting set $\{0^n, 1^n\}$. The existence of hitting sets for interesting/nontrivial classes of functions is not easy to prove directly. Instead proofs are easier if they rely on probabilistic tools and randomized constructions.

Randomness is used by algorithms to search quickly in a space of configurations, trying to pick the desired one. In mathematical proofs mostly it is used mostly to prove the existence of structures with desired properties. Usually the technique is to prove that the probability of having the property on a random structure is greater that zero. This proves the existence of a

*University of Rome "La Sapienza". Computer Science Department.

structure with such property. A classic about such method is [AlSp92]. Derandomizing consists in removing randomness from an algorithm, a combinatorial construction or a mathematical proof. In the first case we refer to the fact that an algorithm is allowed to use random coin tosses. A derandomized version of this algorithm accomplishes the same task, but it is completely deterministic. In the latter cases the existential proof is substituted with a constructive one that shows how to obtain structures with the desired property.
Why derandomizing a mathematical proof? Because an explicit construction can lead to a characterization of the objects with good properties, and to a better understanding of these properties. An explicit construction can also be used in deterministic algorithms. Then why derandomizing an algorithm? One reason is in some sense practical: random algorithms usually do not guarantee success. Some of them can be repeated to obtain a more reliable output, but some others cannot. But there is a more deeper reason that involves the physical world: no random bit exists, and good approximations of it are expensive. Extractors exist, but they rely on entropy properties of the physical source.
Usual approaches to this problem are very different. One could just list all the elements in the sample space (i.e. the set of elements from which the random object is picked) when it is finite. This could be feasible in computation but not in mathematical proofs. Most proofs are in fact *schemes* of proofs. They are parametric and the parameter space is not finite. For example a proof about Ramsey graphs usually involves the size of the graph as parameter. And even if listing was possible, such a proof would not be interesting nor insightful.
We have just said that derandomization in computation is as easy as listing all configuration and verifying them. But of course this cannot be done in reasonable time because usually the sample space is exponentially large in the size of the problem. Computer scientists refer to derandomization of a (probabilistic) algorithm when the time complexity of the deterministic version blows up at most polynomially with respect to the complexity of the probabilistic one. Nontrivial techniques are required to find a deterministic version of a probabilistic algorithm. Usually a general strategy is to focus on the original sample space (usually of exponential size in the input) and to reduce it to a polynomial-size one. Then a simple listing is performed on the smaller sample space.
Derandomizing is also important for theoretical reasons. We want to know if random coin tosses are really necessary for accomplish computational tasks. Maybe coins give just a polynomial speed up in solving problems, in change of the safety of results. Or maybe exist computationally hard tasks that can be solved easily only using the power of randomness. Derandomization has also practical application when a failure of an algorithm has catastrophic consequences.
Nowadays a big open question is of course the so-called "P vs BPP" problem[Bpp]. As much as important as "P versus NP" is from the theoretical point of view, the collapse of BPP in P would have a lot of positive implications such as derandomization of most probabilistic proofs. Of course this is a far more general derandomization process. We are interested in removing randomness from probabilistic algorithms or proofs without pretending to solve big open problems. Luckily in some constructions randomness is redundant and can be safely removed without losing to much in efficiency.
Here we prove probabilistically that a class containing an exponential number of "large" (i.e. of weight[1] bounded away from zero) functions has a polynomial size-hitting set. The proof is a classic use of probabilistic methods, and shows that hitting sets exist for very general classes of functions. Derandomizing such a proof is far from being easy. We just concentrate on a limited class of boolean formulae with some structural constraints: the read-once formulae, in particular DNF and CNF.

[1]The weight of a boolean function f on $\{0,1\}^n$ is the fraction of satisfying assignments $\frac{\#\{x:f(x)=1\}}{2^n}$.

A formula is a boolean function written as composition of $\wedge$, $\vee$ and $\neg$ function over the n variables. All functions can be written by formulae, some of them requiring an exponential size formula but none of them requiring more. The formulae considered here are *read-once*: each variable appears at most once in the whole formula.
This restriction allows us to manage each component of the formula independently from the other ones. An assignment of the variables in a sub-formula does not affect the rest. It is a nice property that is used heavily in the construction shown here.
We show that the known constructions give hitting sets for the class of all read-once CNF[2] with weight greater than $\frac{1}{e}$ and for the class of all read-once DNF[3] with constant positive weight.
These proofs are an example of derandomization of a mathematical proof. For these limited cases we derandomize the existential proof given in Section 1 with ones given in Section 3 and 4.
The hitting set used is an almost $O(\log n)$-wise independent set. The read-once property allows us to manage each clause of the CNF and each monomial of the DNF independently. A fully independent assignment set requires of course 2^n elements. We look for a set that at least locally (in clauses and monomial) resembles the full assignment. A k-wise independent set is a sample space contained in $\{0,1\}^n$ such that every subset of at most k variables is independent (i.e. the probability of a k-bit configuration is 2^{-k}). In [ABI86] a k-wise independent set construction is presented. The size of this set is $n^{k/2}$ that is too much because for $k = O(\log n)$ the sample space is superpolynomial. Unluckily [Chor85] shows that this is the best that can be achieved. To overcome this problem Naor and Naor [NaNa90] introduce the notion of an *almost* independent set. The variables are not independent but the probability of a fixed configuration is near to the uniform one. They compose the construction of an almost independent space with a k-wise independent space construction. In this way they obtain an almost k-wise independent space that is "essentially" as good as the k-wise independent set, but it is smaller. Here we use the same composition of [NaNa90] but using the k-wise independent set in [ABI86] and one of the simple algebraic constructions in [AGHP02] for the almost independent set.
The full construction gives an almost $O(\log n)$-wise independent with polynomially small error. This is sufficient for hitting the two classes of function in Sections 3 and 4.
In Section 1 we prove that any exponential-size class of sufficiently "large" boolean function has a polynomial hitting set. The class of large read-once formulae is one such class. Then we present a definition of almost k-wise independent sets and we discuss the use of such sets as hitting sets. In Section 3 we prove that an almost $2\log n$-wise independent set with n^{-2} bias is a hitting set for the read-once CNF with weight bigger than $\frac{1}{e}$. In Section 4 we prove the same for DNF with weight bigger than any positive constant.
The Last Section is devoted to show the construction of such hitting sets: first we give an outline of the construction scheme, and we prove that the composition of an almost independent set with the k-wise independent set construction leads to an almost k-wise independent set. Then we show the almost independence of the sample set obtained with the algebraic constructions in [AGHP02].

1 Existence of polynomial hitting sets

We start by giving a formal definition of hitting sets.

[2]Conjunctive normal form: the formula is a conjunction of disjunctions. And each disjunction is between positive and negated variable.

[3]Disjunctive normal form: the formula is a disjunction of conjunctions of variables, positive or negated.

Definition 1. *Given a class $\mathcal{C}$ of functions from $\{0,1\}^n$ to $\{0,1\}$, a set $A \subseteq \{0,1\}^n$ is a* hitting set *for $\mathcal{C}$ if for every $f \in \mathcal{C}$ there exists an $x \in A$ such that $f(x) = 1$.*

A polynomial hitting set always exists for a $2^{\text{Poly}(n)}$ size class $\mathcal{C}$ of boolean functions on $\{0,1\}^n$ with weight bigger than a constant (say ϵ). This can be shown by a simple application of the probabilistic method: the average number of functions satisfied by an assignment is at least $|\mathcal{C}|\epsilon$, and so exists one assignment that satisfies at least the average number. Let $\mathcal{C}'$ be the set of remaining functions, then $|\mathcal{C}'| \leq (1-\epsilon)|\mathcal{C}|$. We repeat the argument on C' and so on. Doing this s times leads to s assignments that hit all but *at most* an $(1-\epsilon)^s$ fraction of $\mathcal{C}$. So with $s \geq \frac{\log |\mathcal{C}|}{\log \frac{1}{1-\epsilon}}$ assignments we can satisfy all functions in $\mathcal{C}$. If $\mathcal{C}$ has $2^{\text{Poly(n)}}$ size, then s is polynomial in n.
An example of a class of size at most $2^{\text{Poly}(n)}$ is the class of read-once formulae. Read-once are formulae where any variable appears just once. A read-once formula can be seen as a tree with n leaves (literals with two configuration: positive or negative) and $n-1$ internal nodes (they can be $\wedge$ or $\vee$). A rough counting of all possible read-once formulae can be done observing that binary trees with $n-1$ internal nodes are at most 4^n. There are 2^{n-1} ways to put operators on internal nodes, 2^n ways to put negations over literals and $n!$ ways to put the variables on the leaves. This gives less than $(16n)^n$ possible configurations, that is $2^{4n+n\log n}$. Furthermore notice that for any read-once formula without negation, all formulae obtained adding some negations on top of literals lead to different boolean functions. So read-once formulae describe at least an exponential number of functions. By the probabilistic construction, a set of size $O(4n + n\log n)$ hits all read-once formulae with a given constant minimum weight.

2 Almost k-wise independent sets as hitting sets

An assignment for a boolean formula is a sequence of boolean values associated to the variables. We consider this assignment as a random variable, or better as sequence of random variables $x = (x_1, x_2, \ldots, x_n)$. The distribution of these variables is the uniform on the space $\{0,1\}$. Clearly we cannot use all these assignments as a hitting set for a class. What we are looking for is a small set of assignments (a sample space of the random variables) such that the probability of being satisfied is more than zero for any function in the class.
For this purpose we use a sample space that locally resembles the complete one: the set should be *indistinguishable* from the complete one for any observed set of k variables. More formally we define the *bias* of the k-wise independence of a sample space A as the maximum value of

$$\left| \Pr_{x \in A}\{(x_{i_1}, \ldots, x_{i_k}) = \alpha\} - \frac{1}{2^k} \right|$$

over all $i_1 < i_2 < \ldots < i_k \in [n]$, and $\alpha \in \{0,1\}^k$.

Definition 2. *A set $A \subseteq \{0,1\}^n$ is k-wise independent if its bias is 0.*

Example 1. *The set $\{000, 011, 101, 110\}$ is pairwise independent, but it's not 3-wise independent.*

It is easy to see that the only n-wise independent set is $\{0,1\}^n$. Known deterministic constructions of k-wise independent sets require $n^{k/2}$ elements in the sample space [ABI86], so for any nonconstant k they are too big for our purpose. There is no hope for better because the result is tight [Chor85].
Constructing smaller sample spaces requires the notion of *almost* independence. We say that a sample space is *almost k-wise independent* if the bias tends to zero as n goes to infinity. Such

sets can be constructed in polynomial time and size when $k = O(\log n)$, so we can use them as hitting sets.

3 Derandomizing hitting sets for large read-once CNFs

Derandomizing hitting set constructions for boolean formulae is a difficult task. As we said in Section 1 there exists a quasi-linear hitting set for all large read-once formulae. Unluckily a derandomized construction for such a general set is not known; to find one we need strong hypothesis both on the structure and on the weight of the formula. This Section is devoted to prove that an almost $2\log n$-wise independent space is a hitting set for all read-once CNFs with a large weight. Such sets can be constructed in polynomial time by a deterministic algorithm, so this solves the problem at least for this class of formulae. The following theorem gives a precise statement.

Theorem 1. *Let $A \in \{0,1\}^n$ be an almost $2\log n$-wise independent sample space with $\frac{1}{n^2}$ bias. Then A is a hitting set for the class of read-once CNFs on n variables with weight at least $\frac{\gamma}{e}$ (for any $\gamma > 1$).*

Proof. **(V. Kabanets [Kab06])** Given a CNF formula, we split it in two CNFs: L (long) composed by all clauses with more than $2\log n$ literals, and S (short) with the other ones. Now we estimate the probability that an assignment picked randomly in A satisfies the two formulae.
Long clauses. Even considering only $2\log n$ literals from a long clause, the probability that it is not satisfied by a random assignment is less than $\frac{1}{2^{2\log n}} = \frac{1}{n^2}$. A random assignment from A fails with probability less than $\frac{2}{n^2}$ even considering the bias. The probability of failing in at least one of the long clauses is at most $\frac{2}{n}$ because of the union bound on at most n clauses.
Short clauses. Let be b_i the number of literals in the i^{th} short clause. It is unsatisfied by at most a 2^{-b_i} fraction of all assignments. Clauses are independent because the formula is read-once, so the weight of S is equal to $\prod_i(1 - \frac{1}{2^{-b_i}})$. By the almost independence of A, a $2^{-b_i} + \frac{1}{n^2}$ fraction of the set does not satisfy the i^{th} clause. The union over of unsatisfying assignments of whole clauses is smaller than A, as shown by the following inequalities starting with the union bound:

$$\begin{aligned}
\Pr_{x\in A}\{S \text{ is not satisfied}\} &\leq \textstyle\sum_i \left(\frac{1}{n^2} + 2^{-b_i}\right) \\
&\leq \tfrac{1}{n} + \textstyle\sum_i 2^{-b_i} && \text{Clauses are at most } n \\
&\leq \tfrac{1}{n} - \textstyle\sum_i \ln(1 - 2^{-b_i}) && x \geq -\ln(1-x) \\
&\leq \tfrac{1}{n} - \ln \textstyle\prod_i \left(1 - \frac{1}{2^{-b_i}}\right) \\
&\leq \tfrac{1}{n} - \ln \tfrac{\gamma}{e} = \tfrac{1}{n} + 1 - \ln\gamma
\end{aligned}$$

So $S \wedge L$ is not satisfied by at most a $1 - \ln\gamma + \frac{3}{n}$ fraction of A. γ is a positive constant, so for n big enough we have the statement. □

4 Derandomizing hitting sets for (not so) large read-once DNFs

Read-once DNFs on n variables are large enough for derandomization if the weight is bounded away from 0. The goal of this Section is to show that such class of functions is hitten by a polynomial time constructable set. First of all we notice than any $\log n$-wise independent set of assignments with $\frac{1}{n^2}$ bias solve any AND-term of less or equal than $\log n$ elements with probability at least $\frac{1}{n} - \frac{1}{n^2}$ that is greater than 0. So this set solves all the DNFs with such small AND-terms. The proof of the following claim completes the goal.

Claim 1. *Any read-once DNF formula over n variables with weight bounded away from 0 has an AND-term with less than $\log n$ literals for n large enough.*

Proof. Given a read-once DNF formula, let be b_i the number of literals in the i^{th} AND-term. The weight of the formula is $1 - \prod_i \left(1 - \frac{1}{2^{b_i}}\right)$. If for every i, $b_i > \log n$, then the weight is bounded from above by

$$1 - \prod_i \left(1 - \frac{1}{2^{\log n}}\right) \leq 1 - \left(1 - \frac{1}{n}\right)^{n/\log n} \leq 1 - e^{-\frac{1}{\log n}}$$

The last term of the inequalities goes to 0 as n goes to infinity. So the weight of the formulae is not bounded away from zero for n large enough. □

A hitting set for read-once DNFs on n variables with weight greater than a constant ϵ, can be produced in the following way:

- Let be n_ϵ the smallest number of variables such Claim 1 holds
- If $n > n_\epsilon$ use an almost $\log n$-wise independent set with $\frac{1}{n^2}$ bias
- If $n \leq n_\epsilon$ produce the full set of assignments over $\{0,1\}^{n_\epsilon}$.

When ϵ is a constant, also n_ϵ is constant and the full set $\{0,1\}^{n_\epsilon}$ has constant size.

5 Almost k-wise independent sets contructions

In this Section we show three constructions of an almost k-wise independent sample space with bias ϵ, such that if $k = O(\log n)$ and $\epsilon^{-1} = \text{Poly}(n)$ then the space has polynomial size in n and is built by a polynomial time algorithm. Logarithmic independence and polynomially small bias are sufficient for the sample space to be a hitting set for both classes of formulae studied in the previous sections.
Such a sample set is built in two steps like in [NaNa90], but for the first step we use one of the constructions in [AGHP02] instead of the original one. In each step there is one element of a small sample space that is mapped to a bigger space by a deterministic polynomial algorithm. Such algorithm is nothing else than a function from boolean strings to larger boolean strings. So we have three sample spaces: $\{0,1\}^m$, $\{0,1\}^M$, $\{0,1\}^n$ with $m < M < n$. The first construction maps two m-bit strings to M-bit strings, the second one maps M-bit strings to n-bit strings.
In the first step we build a set that is almost M-independent with bias ϵ. To do this we use one of the simple algebraic constructions in [AGHP02]. In the second step we enlarge to n the number of variables, obtaining almost k-wise independence with the same bias ϵ.
Almost k-wise independence is somehow difficult to verify directly. So we will look for a stronger property that implies the independence but it is simpler to manage. In [Vaz86] Vazirani observes that *if linear combinations of the variables almost resemble a random coin, then the variables are almost independent.* In the following we will look for sample sets with this stronger property.
Notation (Variables and Constants). n is the number of variables in the final sample space. k is the size of almost independent subsets of variables in the final space. ϵ is the bias of the constructions. m is the size of the seeds picked from the truly random sample space. M is the number of variables in the intermediate sample space. d is the number of variables when the sample space is used in general statements. x, y denote either elements of a field, binary strings or polynomials. Polynomials are also written over the variable t (as $x(t)$). p denotes a

prime number. The relation $x =_\gamma y$ means that x and y are equal modulo γ, which can be an integer or a polynomial.

Notation (Algebraic structures). We will heavily use the standard mapping between elements in $\{0,1\}^d$ and d-degree polynomials over GF(2): any polynomial $x(t) = t^d + \sum_{i=0}^{d-1} x_i t^i$ of degree d is related to the string $x = x_0 x_1 x_2 \cdots x_{d-1}$. In the same way an element in GF(2^d) is related with d binary strings using the standard representation. For x, y (polynomials, field elements or binary strings) we denote $\langle x, y\rangle = \sum_i x_i y_i \pmod 2$. We also denote the *quadratic character* [4] of x with respect to p as $\chi_p(x)$.

5.1 Small bias w.r.t linear tests implies small biased independence

In [Vaz86] Vazirani observes that when any linear combination of the variables in the sample space is near to a random coin toss, then the general sample space is near to be independent.

Definition 3. *If α is a binary string over $\{0,1\}^d$ or a subset of $[d]$ then we say that α is a* k-linear test *if $0 < |\alpha| \leq k$.*

We refer briefly to d-linear tests as *linear tests*.

Definition 4. *We say that a sample space $A \subseteq \{0,1\}^d$ is ϵ-biased with respect to linear tests if for every linear test $\alpha \neq 0 \in \{0,1\}^d$*

$$|\Pr_{x\in A}[\langle x, \alpha\rangle = 0] - \Pr_{x\in A}[\langle x, \alpha\rangle = 1]| \leq \epsilon$$

If we restrict to $|\alpha| \leq k$ we say that A is ϵ-biased w.r.t. k-linear tests. The following lemma that shows how linear tests are related with independence.

Lemma 1. *If $A \subseteq \{0,1\}^d$ is ϵ-biased w.r.t. linear tests then A is an ϵ-biased almost independent sample space.*

Proof. Define $f(z) := \Pr_{x\in A}[x_0 x_1 \ldots x_{d-1} = z]$, and consider the Fourier coefficients

$$\hat{f}(y) = \frac{1}{2^d} \sum_{z\in\{0,1\}^d} f(z)(-1)^{\langle y,z\rangle} = \sum_{z:\langle y,z\rangle=0} \Pr_{x\in A}[x = z] - \sum_{z:\langle y,z\rangle=1} \Pr_{x\in A}[x = z]$$

So $\hat{f}(y)$ is the bias of the linear test y. We have $\hat{f}(0) = 1$, and $|\hat{f}(y)| \leq \epsilon$ for $y \neq 0$ by hypothesis. The bias of the independence of the variables is

$$\left| f(z) - \frac{1}{2^d} \right| = \frac{1}{2^d} \left| \sum_{y\neq 0} \hat{f}(y)(-1)^{\langle y,z\rangle} \right| \leq \epsilon$$

□

Applying the same argument to every subset of k-variables we obtain that

Corollary 1. *If $A \subseteq \{0,1\}^d$ is ϵ-biased w.r.t. k-linear tests (linear tests of size at most k) then A is an ϵ-biased almost k-wise independent sample space.*

[4] The quadratic character of $x \neq_p 0$ with respect to p is 1 if $y^2 =_p x$ has a solution, -1 otherwise. If $x =_p 0$ the character is defined to be 0.

5.2 The construction

First Step. In [AGHP02] the authors show three quite simple constructions for sample spaces with small biased w.r.t linear tests. In all of them we select a pair of small (m bits) strings at random, and we use them as input of a deterministic boolean function from the small space to the M-bit space. The output string is denoted $b_0 b_1 \cdots b_M$.

Linear feedback shift register

SAMPLE. An irreducible polynomial f of degree m over GF(2) (the feedback rule).
A string s in $\{0,1\}^m$ (the seed).

RULE. $b_0 \cdots b_{m-1} := s_0 \cdots s_{m-1}$
$b_{i+1} := \langle b_{i+1-m} b_{i+2-m} \cdots b_i, x \rangle$ for $m \leq i < M$.

Quadratic character

SAMPLE. A prime $p > 2$ and $x \in \mathrm{GF}(p)$ (in this case $m = O(\log p)$).

RULE. If $x + i \neq_p 0$ then $b_i := \frac{1-\chi_p(x+i)}{2}$ (mapping from {1,-1} to {0,1}).
If $x + i =_p 0$ then $b_i := 1$.

Powering

SAMPLE. x and y in $GF(2^m)$.

RULE. $b_i := \langle x^i, y \rangle$

In subsections 5.3, 5.4 and 5.5 we show that proper tuning of $m = O(\log M + \log \frac{1}{\epsilon})$ leads to an output sample space on M variables that is ϵ-biased w.r.t. linear tests.

Second step. Now the following construction gives the final sample space ϵ-biased w.r.t. k-linear tests.

Claim 2. *Let be $\{v_1, v_2, \ldots, v_n\}$ a set of k-wise linear independent vectors over $\{0,1\}^M$. Let r be an ϵ-biased random vector on the same space. The random variables $\{y_i = r^T v_i\}$ are ϵ-biased with respect to k-linear tests.*

Proof. Fix S an nonempty subset of $[n]$, with at most k elements.

$$\sum_{i \in S} y_i = \sum_{i \in S} \langle r, v_i \rangle = \langle r, \sum_{i \in S} v_i \rangle$$

The sum of at most k vectors is not zero by the linear independence, so $\sum_{i \in S} v_i$ is a linear test for r. The bias of the last term is ϵ. □

For the previous construction the linear transformation from $\{0,1\}^M$ to $\{0,1\}^n$ can be any k-wise linear independent set of n vectors. In [ABI86] a construction with $\frac{k}{2} \log n$ size vectors is shown.

First and second step are composed setting $M = \frac{k}{2} \log n$ and $m = O(\log k + \log \log n + \log \frac{1}{\epsilon})$. Applying the full construction to all possible inputs gives a subset of $\{0,1\}^n$ which is almost k-wise independent with ϵ-bias.

Now we prove that the algebraic constructions pass linear tests with small bias. In each proof we have to show that the random variable $\langle \alpha, x \rangle$ (over x in the output space) is a small bias random coin toss for every $\alpha \neq 0$.

5.3 Linear Feedback Shift Register construction

We prove that the Linear feedback construction passes linear tests with sufficiently small bias.

Claim 3. *The Linear feedback construction passes linear tests with bias* $\frac{M-1}{2^m}$.

Proof. Fix the test $\alpha \neq 0$ and the feedback polynomial $f(t) = t^m + \sum_{i=0}^{m-1} f_i t^i$. Any bit in b is a fixed linear combination of the random seed $\{s_0, s_1, \ldots, s_{m-1}\}$. Now observe that the same linear combination over $\{t^0, t^1, \ldots, t^{m-1}\}$ represents t^j modulo f.

$(j < m)$ Trivially true.

$(j = m)$ We have $t^m =_{f(t)} \sum_{i=0}^{m-1} f_i t^i$.

$(j > m)$ We have $t^j = t^{j-m} \cdot t^m =_{f(t)} \sum_{i=0}^{m-1} f_i t^{j-m+i}$

The recursion is the same, so the linear combinations match. Using this correspondence we can say that $\langle \alpha, b \rangle$ is identically zero if and only if $\sum_{i=0}^{M-1} \alpha_i x^i$ is a multiple of f. If it is not zero, then varying the seed s the value of $\langle \alpha, b \rangle$ is an unbiased random coin. If it is zero then $\langle \alpha, b \rangle = 0$ always . So the bias is the probability that, given $\alpha \neq 0$, f divides α (as polynomials over t of degree respectively m and M).
This probability is obtained observing that an M-degree polynomial can be divided by at most $\frac{M-1}{m}$ irreducible polynomials of proper degree m. Such polynomials are at least $\frac{2^m}{m}$ [Gol82, p. 39]. So the bias is less than $\frac{M-1}{2^m}$. □

5.4 Quadratic Character construction

We prove that the quadratic character vector of a fixed prime, starting from a random point, resembles a random sequence. We will use the following theorem by Weil [Sch76]:

Theorem 2. *Let $p > 2$ a prime. Let $f(t)$ a polynomial over* GF(p) *with M distinct roots and which is not a square of any other polynomial. Then*

$$|\sum_{x \in \mathrm{GF}(p)} \chi_p(f(x))| \leq (M-1)\sqrt{(p)}$$

This result is useful to bound the bias of the construction.

Claim 4. *The quadratic character construction passes linear tests with* $\frac{M-1}{\sqrt{(p)}} + \frac{M}{p}$ *bias.*

Proof. Fix the test $\alpha \neq 0$. We say x is good if $x + i \neq_p 0$ for every i, otherwise we say x is bad. For good x's we have $(-1)^{b_i} = \chi_p(x+i)$. Bad x's are at most M because for each i at most one x can have $x + i =_p 0$.
The bias of $\langle \alpha, b \rangle$ is equal to the expected value of $(-1)^{\langle \alpha, b \rangle}$ over x and it is

$$\frac{1}{p}\left|\sum_{x \in \mathrm{GF}(p)} (-1)^{\sum_{i=0}^{M-1} \alpha_i b_i}\right| = \frac{1}{p}\left|\sum_{x \in \mathrm{GF}(p)} \prod_{i=0}^{M-1} \chi_p^{\alpha_i}(x+i)\right| + \frac{M}{p} = \frac{1}{p}\left|\sum_{x \in \mathrm{GF}(p)} \chi_p(\prod_{i=0}^{M-1} (x+i)^{\alpha_i})\right| + \frac{M}{p}$$

because of the multiplicative property of χ_p. The polynomial in χ_p in the rightmost expression has M roots and is square free, so Weil's Theorem concludes the proof. □

5.5 Powering construction

We prove that the Powering construction passes linear tests with small bias.

Claim 5. *The Powering construction passes linear tests with* $\frac{M-1}{2^m}$ *bias.*

Proof. Fix $\alpha \neq 0$. Note that $b_i = \langle x^i, y\rangle$ by definition. So by linearity we have:

$$\langle \alpha, b\rangle = \sum_{i=0}^{M-1} \alpha_i \langle x^i, y\rangle = \langle \sum_{i=0}^{M-1} \alpha_i x^i, y\rangle$$

Let be $p_\alpha(t) = \sum_{i=0}^{M-1} \alpha_i t^i$. If x is a root of p_α then $\langle \alpha, b\rangle = \langle p_\alpha(x), y\rangle$ is 0 for every choice of y. If x is not, then the outcome is fully random. This leads to a linear test biased by a factor of $\frac{\#\{\text{roots of } p_\alpha\}}{2^m} \leq \frac{M-1}{2^m}$. □

6 Conclusion

We have exhibited hitting sets for quite simple classes of functions. We suspect that an almost $O(\log n)$-wise independent sample space can be a hitting set for larger classes of functions. Furthermore we should investigate on the possible use of almost n-wise independent spaces to overcome the per-clause analysis shown here.

Acknowledgements. I wish to thank P. Pudlák for valuable hints during the early work for this project, and Anna de Mier for helpful discussions. Finally most of my gratitude goes to the members of the Department of Applied Mathematics at Charles University in Prague for providing a stimulating working environment and the opportunities of learning a lot.

References

[ABI86] N. Alon, L. Babai and A. Itai, "A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem", *Journal of Algorithms*, Vol 7 (1986), pp. 567-583.

[AGHP02] N. Alon, O. Goldreich, J. Håstad and R. Peralta,"Simple Constructions of Almost k-wise Independent Random Variables", *Random Structures and Algorithms*, Vol 3 (1992), pp. 289-304.

[AlSp92] N. Alon and J. Spencer, *The Probabilistic Method*, John Wiley and Sons Inc., New York, 1992.

[Bpp] Michael Sipser, *Introduction to the Theory of Computation*, PWS Publishing, (1997). Section 10.2.1: The class BPP, pp.336-339.

[Chor85] B. Chor, J. Friedmann, O. Goldreich, J. Hastad, S. Rudish, and R. Smolensky, "The bit extraction problem and t-resilient functions", *26th FOCS*, 1985, pp. 396-407.

[Gol82] S.W. Golomb, *Shift Register Sequences*, Aegean Park Press, Revised edition, 1982.

[Kab06] V. Kabanets, *private communications.*

[NaNa90] J.Naor and M.Naor, "Small-bias Probability Spaces: Efficient Constructions and Applications", *22th STOC*, 1990, pp. 213-223.

[Sch76] W. M. Schmidt, "Equations over Finite Fields, an elementary approach", *Lecture Notes in Mathematics*, Vol. 536, Springer Verlag, 1976.

[Vaz86] U.V. Vazirani, *Randomness Adversaries and Computation*, Ph.D. Thesis, EECS, UC Berkeley, 1986.