Electronic Colloquium on Computational Complexity, Report No. 4 (1999)

Almost k-Wise Independence and Boolean Functions Hard for Read-Once Branching Programs

Valentine Kabanets
Department of Computer Science
University of Toronto
Toronto, Canada
kabanets@cs.toronto.edu

February 2, 1999

Abstract

Andreev et al. [ABCR97] give constructions of Boolean functions (computable by polynomial-size circuits) that require large read-once branching program (1-b.p.'s): a function in P that requires 1-b.p. of size at least $2^{n-\mathrm{polylog}(n)}$, a function in quasipolynomial time that requires 1-b.p. of size at least $2^{n-O(\log n)}$, and a function in LINSPACE that requires 1-b.p. of size $2^{n-\log n-O(1)}$. We point out alternative, much simpler constructions of such Boolean functions by applying the idea of almost k-wise independence more directly, without the use of discrepancy set generators for large affine subspaces. Our constructions are obtained by derandomizing the probabilistic proofs of existence of the corresponding combinatorial objects.

Keywords: almost k-wise independence, derandomization, exponential lower bounds for read-once branching programs, r-mixed Boolean functions.

1 Introduction

Andreev et al. [ABCR97] construct Boolean functions hard for read-once branching programs. Recall that a *branching program* is a directed acyclic graph with one source and with each node of out-degree at most 2. Each node of out-degree 2 (a branching node) is labeled by an index of an input bit, with one outgoing edge labeled by 0, and the other by 1; each node of out-degree 0 (a sink) is labeled by 0 or 1. The branching program accepts an input if there is a path from the source to a sink labeled by 1 such that, at each branching node of the path, the path contains the edge labeled by the input bit for the input index associated with that node. A branching program is *read-once (1-b.p.)* if, on every path from the source to a sink, no two branching nodes are labeled by the same input index. Finally, the *size* of a branching program is defined as the number of its nodes.

Read-once branching programs represent a model of computation for which we can prove strong lower bounds. One way of getting a lower bound is to apply a combinatorial theorem of Simon and Szegedy [SS93], a particular case of which states that any 1-b.p. computing an r-mixed Boolean function has size at least 2^r. Informally, an r-mixed function essentially depends on every set of r variables (see the next section for a precise definition).

Andreev et al. [ABCR97] show how to construct a Boolean function $f_n(x_1, \ldots, x_n)$ in LINSPACE$\cap$ P/poly that is r-mixed for $r = n - \lceil \log n \rceil - 2$ for almost all n. By the theorem of Simon and

ISSN 1433-8092

Szegedy mentioned above, this yields the lower bound $\Omega(2^n/n)$ for 1-b.p.'s, which is tight. A Boolean function in DTIME$(2^{\log^2 n}) \cap$ P/poly that requires a 1-b.p. of size at least $2^{n-O(\log n)}$ is constructed by derandomizing the probabilistic construction of a certain Boolean function given in [SZ96]. The derandomized algorithm needs only $O(\log^2)$ advice bits to determine a polynomial-time computable function with the 1-b.p. lower bound of at least $2^{n-O(\log n)}$; by making these bits a part of the input, one gets a function in P that requires a 1-b.p. of size at least $2^{n-O(\log^2 n)}$.

Both constructions in [ABCR97] use the idea of ϵ-biased sample spaces introduced by Naor and Naor [NN93], who also gave an algorithm for generating small sample spaces; three simpler constructions of such spaces were later given by Alon et al. [AGHP92]. Andreev et al. define certain ϵ-discrepancy sets for systems of linear equations over GF(2), and relate these discrepancy sets to the biased sample spaces of Naor and Naor through a reduction lemma. Using a particular construction of a biased sample space (the powering construction from [AGHP92]), Andreev et al. give an algorithm for generating ϵ-discrepancy sets, which is then used to derandomize both a probabilistic construction of an r-mixed Boolean function for $r = n - \lceil \log n \rceil - 2$ and the construction in [SZ96] mentioned above.

We will show that the known algorithms for generating small ϵ-biased sample spaces can be applied *directly* to get the r-mixed Boolean function as above, and to derandomize the construction in [SZ96]. The idea of our first construction is very simple: treat the elements (bit strings) of an ϵ-biased sample space as the truth tables of Boolean functions. This will induce a probability distribution on Boolean functions such that, on any subset A of k inputs, the restriction to A of a Boolean function chosen according to this distribution will look almost as if it were a uniformly chosen random function defined on the set A. By an easy probabilistic argument, we will show that such a space of functions will contain the desired r-mixed function, for a suitable choice of parameters ϵ and k. We mention several possible constructions of an r-mixed Boolean function with $r = n - \lceil \log n \rceil - 2$.

In our second construction, we derandomize a probabilistic existence proof in [SZ96]. We proceed along the usual path of derandomizing probabilistic algorithms whose analysis depends only on almost k-wise independence rather than full independence of random bits [NN93]. Observing that the construction in [SZ96] is one of such algorithms, we reduce its randomness complexity to $O(\log^3)$ bits (again treating strings of an appropriate sample space as truth tables). This gives us a DTIME$(2^{O(\log^3 n)})$-computable Boolean function of quasilinear circuit-size with the lower bound for 1-b.p.'s slightly better than that for the corresponding quasipolynomial-time computable function in [ABCR97], and a Boolean function in quasilinear time, QL, with the lower bound for 1-b.p.'s at least $2^{n-O(\log^3 n)}$, which is only slightly worse than the lower bound for the corresponding polynomial-time function in [ABCR97]. In the analysis of our construction, we employ a combinatorial lemma due to Razborov [Raz88], which bounds from above the probability that none of n events occur, given that these events are almost k-wise independent.

The remainder of the paper. In the following section, we state the necessary definitions and some auxiliary lemmas. In Section 3, we show how to construct an r-mixed function that has the same lower bound for 1-b.p. as that in [ABCR97]. In Section 4, we give a simple derandomization procedure for a construction in [SZ96], obtaining two more Boolean functions (computable in polynomial time and quasipolynomial time, respectively) that are hard with respect to 1-b.p.'s.

2 Preliminaries

Below we give the standard definitions of k-wise independence and (ϵ, k)-independence. We consider probability distributions that are uniform over some set $S \subseteq \{0,1\}^n$; such a set is denoted by S_n and called a *sample space*.

Let S_n be a sample space, and let $X = x_1 \dots x_n$ be a string chosen uniformly from S_n. Then S_n is *k-wise independent* if, for any k indices $i_1 < i_2 < \cdots < i_k$ and any k-bit string α,

$$\mathbf{Pr}[x_{i_1} x_{i_2} \dots x_{i_k} = \alpha] = 2^{-k}.$$

Similarly, for S_n and X as above, S_n is *(ϵ, k)-independent* if

$$|\mathbf{Pr}[x_{i_1} x_{i_2} \dots x_{i_k} = \alpha] - 2^{-k}| \leqslant \epsilon$$

for any k indices $i_1 < i_2 < \cdots < i_k$ and any k-bit string α.

Naor and Naor [NN93] present an efficient construction of small (ϵ, k)-independent sample spaces; three simpler constructions are given in [AGHP92]. Here we recall just one construction from [AGHP92], the powering construction, although any of their three constructions could be used for our purposes.

Consider the Galois field $\mathrm{GF}(2^m)$ and the associated m-dimensional vector space over $\mathrm{GF}(2)$. For every element u of $\mathrm{GF}(2^m)$, let $\mathrm{bin}(u)$ denote the corresponding binary vector in the associated vector space. The sample space Pow_N^{2m} is defined as a set of N-bit strings such that each string ω is determined as follows. Two elements $x, y \in \mathrm{GF}(2^m)$ are chosen uniformly at random. For each $1 \leqslant i \leqslant N$, the ith bit ω_i is defined as $\langle \mathrm{bin}(x^i), \mathrm{bin}(y) \rangle$, where $\langle a, b \rangle$ denotes the inner product over $\mathrm{GF}(2)$ of binary vectors a and b.

Lemma 1 ([AGHP92]) *For every $k \leqslant N$, the sample space Pow_N^{2m} is $(\frac{N}{2^m}, k)$-independent.*

The proof of the above lemma follows from the results in [AGHP92] (Proposition 3 and Corollary 1).

As we have mentioned in the introduction, we shall view the strings of the sample space Pow_N^{2m} as the truth tables of Boolean functions of $\log N$ variables. It will be convenient to assume that N is a power of 2, i.e., $N = 2^n$. Thus, the uniform distribution over the sample space $\mathrm{Pow}_{2^n}^{2m}$ induces a distribution $\mathbf{F}_{n,m}$ on Boolean functions of n variables that satisfies the following lemma.

Lemma 2 *Let A be any set of k tuples from $\{0,1\}^n$, for any $k \leqslant 2^n$. Let ϕ be any Boolean function defined on A. For a Boolean function f chosen according to the distribution $\mathbf{F}_{n,m}$ defined above, we have*

$$|\mathbf{Pr}[f|_A = \phi] - 2^{-k}| \leqslant 2^{-(m-n)},$$

where $f|_A$ denotes the restriction of f to the set A.

Proof: The k tuples in A determine k indices $i_1, \dots, i_k$ in the truth table of f. The function ϕ is determined by its truth table, a binary string α of length k. Now the claim follows immediately from Lemma 1 and the definition of (ϵ, k)-independence. ■

We say that a Boolean function $f_n(x_1, \dots, x_n)$ is *r-mixed* for some $r \leqslant n$ if, for every subset X of r input variables $\{x_{i_1}, \dots, x_{i_r}\}$, no two distinct assignments to X yield the same subfunction of f in the remaining $n - r$ variables. We shall see in the following section that an r-mixed function for $r = n - \lceil \log n \rceil - 2$ has a nonzero probability in a distribution $\mathbf{F}_{n,m}$, where $m \in O(n)$.

Following Savický and Žák [SZ96], we call a function $\phi : \{0,1\}^n \to \{1, 2, \dots, n\}$ *(s, n, q)-complete*, for some integers s, n, and q, if for every set $I \subseteq \{1, \dots, n\}$ of size $n - s$ we have

1. for every 0-1 assignment to the variables x_i, $i \in I$, the range of the resulting subfunction of ϕ is equal to $\{1, 2, \ldots, n\}$, and
2. there are at most q different 0-1 assignments to x_i, $i \in I$, that result in different subfunctions of ϕ.

It is proved in [SZ96] that a Boolean function $f(\vec{x}) = x_{\phi(\vec{x})}$ requires 1-b.p.'s of size at least $2^{n-s}/q$, provided that ϕ is an (s, n, q)-complete function. The following lemma can be used to construct an (s, n, q)-complete function.

Lemma 3 ([SZ96]) *Let A be a $t \times n$ matrix over* GF(2) *with every $t \times s$ submatrix of rank at least r. Let $\psi : \{0,1\}^t \to \{1, 2, \ldots, n\}$ be a mapping such that its restriction to every affine subset of $\{0,1\}^t$ of dimension at least r has the range $\{1, 2, \ldots, n\}$. Then the function $\phi(\vec{x}) = \psi(A\vec{x})$ is $(s, n, 2^t)$-complete.*

A probabilistic argument shows that a $t \times n$ matrix A and a function $\psi : \{0,1\}^t \to \{1, 2, \ldots, n\}$ exist that satisfy the assumptions of Lemma 3 for the choice of parameters $s, t, r \in O(\log n)$, thereby yielding a Boolean function that requires 1-b.p.'s of size at least $2^{n - O(\log n)}$. Below we will show that the argument uses only limited independence of random bits, and hence it can be derandomized using the known constructions of (ϵ, k)-independent spaces. Our proof will utilize the following lemma of Razborov.

Lemma 4 ([Raz88]) *Let $l > 2k$ be any natural numbers, let $0 < \theta, \epsilon < 1$, and let $\mathcal{E}_1, \ldots, \mathcal{E}_l$ be events such that, for every subset $I \subseteq \{1, \ldots, l\}$ of size at most k,*

$$|\mathbf{Pr}[\wedge_{i \in I} \mathcal{E}_i] - \theta^{|I|}| \leqslant \epsilon.$$

Then we have

$$\mathbf{Pr}[\wedge_{i=1}^{l} \bar{\mathcal{E}}_i] \leqslant e^{-\theta l} + \binom{l}{k+1}(\epsilon k + \theta^k)$$

For convenience of the reader, we give the proof of Lemma 4 in the appendix.

3 Constructing r-Mixed Boolean Functions

First, we give a simple probabilistic argument showing that r-mixed functions exist for $r = n - \lceil \log n \rceil - 2$. Let f be a Boolean function on n variables that is chosen uniformly at random from the set of all Boolean n-variable functions. For any fixed set of indices $\{i_1, \ldots, i_r\} \subseteq \{1, \ldots, n\}$ and any two fixed binary strings $\alpha = \alpha_1, \ldots, \alpha_r$ and $\beta = \beta_1, \ldots, \beta_r$, the probability that fixing $x_{i_1}, \ldots, x_{i_r}$ to α and then to β will give the same subfunction of f in the remaining $n - r$ variables is 2^{-k}, where $k = 2^{n-r}$. Thus, the probability that f is not r-mixed is at most

$$\binom{n}{r} 2^{2r} 2^{-k},$$

which tends to 0 as n grows.

We observe that the above argument only used the fact that f is random on any set of $2k$ inputs: those obtained after the r variables $x_{i_1}, \ldots, x_{i_r}$ are fixed to α, the set of which will be denoted as A_α, plus those obtained after the same variables are fixed to β, the set of which will be denoted as A_β. This leads us to the following theorem.

Theorem 5 *There is an $m \in O(n)$ for which the probability that a Boolean n-variable function f chosen according to the distribution $\mathbf{F}_{n,m}$ is r-mixed, for $r = n - \lceil \log n \rceil - 2$, tends to 1 as n grows.*

Proof: By Lemma 2, the distribution $\mathbf{F}_{n,m}$ yields a function f which is equal to any fixed Boolean function ϕ defined on a set $A_\alpha \cup B_\beta$ of $2k$ inputs with probability at most $2^{-2k} + 2^{-(m-n)}$. The number of functions ϕ that assume the same values on the corresponding pairs of elements $a \in A_\alpha$ and $b \in A_\beta$ is 2^k. Thus, the probability that f is not r-mixed is at most

$$\binom{n}{r} 2^{2r} (2^{-k} + 2^{-(m-n-k)}).$$

Setting $m = (7+\delta)n$ for any $\delta > 0$ makes this probability tend to 0 as n grows. ■

By definition, each function from $\mathbf{F}_{n,m}$ can be computed by a Boolean circuit of size poly(n,m). It must be also clear that checking whether a function from $\mathbf{F}_{n,m}$, given by a $2m$-bit string, is r-mixed can be done in LINSPACE. It follows from Theorem 5 that we can find an r-mixed function, for $r = n - \lceil \log n \rceil - 2$, in LINSPACE by picking the lexicographically first string of $2m$ bits that determines such a function.

Remark 6 Any of the three constructions of small (ϵ, k)-independent spaces in [AGHP92] could be used in the same manner as described above to obtain an r-mixed Boolean function computable in LINSPACE $\cap$ P/poly, for $r = n - \lceil \log n \rceil - 2$.

4 Constructing (s, n, q)-Complete Functions

Let us take a look at the probabilistic proof (as presented in [SZ96]) of the existence of a matrix A and a function ψ with the properties assumed in Lemma 3. Suppose that a $t \times n$ matrix A over GF(2) and a function $\psi : \{0,1\}^t \to \{1, 2, \ldots, n\}$ are chosen uniformly at random. For a fixed $t \times s$ submatrix B of A, if rank$(B) < r$, then there is a set of at most $r-1$ columns in B whose linear span contains each of the remaining $s-r+1$ columns of B. For a fixed set R of such $r-1$ columns in B, the probability that each of the $s-r+1$ vectors chosen uniformly at random will be in the linear span of R is at most $(2^{r-1}/2^t)^{s-r+1}$. Thus, the probability that the matrix A is "bad" is at most

$$\binom{n}{s} \binom{s}{r-1} 2^{-(t-r+1)(s-r+1)}. \tag{1}$$

For a fixed affine subspace H of $\{0,1\}^t$ of dimension r and a fixed $1 \leqslant i \leqslant n$, the probability that the range of ψ restricted to H does not contain i is at most $(1 - 1/n)^{2^r}$. The number of different affine subspaces of $\{0,1\}^t$ of dimension r is at most $2^{(r+1)t}$; the number of different i's is n. Hence the probability that ψ is "bad" is at most

$$2^{(r+1)t} n \left(1 - \frac{1}{n}\right)^{2^r} \leqslant 2^{(r+1)t} n e^{-2^r/n}. \tag{2}$$

An easy calculation shows that setting $s = \lceil (2+\delta) \log n \rceil$, $t = \lceil (3+\delta) \log n \rceil$, and $r = \lceil \log n + 2 \log\log n + b \rceil$, for any $\delta > 0$ and sufficiently large b (say, $b = 3$ and $\delta = 0.01$), makes expressions (1) and (2) tend to 0 as n grows.

Theorem 7 *There exist constants $d_1, d_2, d_3 \in \mathbb{N}$ such that every $(2^{-d_1 \log^3 n}, d_2 \log^2 n)$-independent sample space over n^{d_3}-bit strings contains both matrix A and function ψ with the properties as in Lemma 3, for $s, r, t \in O(\log n)$.*

Proof: We observe that both probabilistic arguments used only partial independence of random bits. For A, we need a tn-bit string coming from an (ϵ, k)-independent sample space with $k = ts$ and $\epsilon = 2^{-c_1 \log^2 n}$, for a sufficiently large constant c_1. Indeed, for a fixed $t \times s$ submatrix B of A and a fixed set R of $r-1$ columns in B, the number of "bad" $t \times s$-bit strings α filling B so that the column vectors in R contain in their linear span all the remaining $s-r+1$ column vectors of B is at most $2^{(r-1)t}2^{(r-1)(s-r+1)} = 2^{(r-1)(s+t-r+1)}$. If A is chosen from the (ϵ, k)-independent sample space with ϵ and k as above, then the probability that some fixed "bad" string α is chosen is at most $2^{-ts} + \epsilon$. Thus, in this case, the probability that A is "bad" is at most

$$\binom{n}{s}\binom{s}{r-1}(2^{-(t-r+1)(s-r+1)} + \epsilon 2^{(r-1)(s+t-r+1)}).$$

Choosing the same s, t, and r as in the case of fully independent probability distribution, one can make this probability tend to 0 as n grows, by choosing sufficiently large c_1.

Similarly, for the function ψ, we need a $2^t\lceil \log n\rceil$-bit string from an (ϵ, k)-independent sample space with $k = c_2 \log^2 n$ and $\epsilon = 2^{-c_3 \log^3 n}$, for sufficiently large constants c_2 and c_3. Here we view the truth table of ψ as a concatenation of 2^t $\lceil \log n\rceil$-bit strings, where each $\lceil \log n\rceil$-bit string encodes a number from $\{1, \ldots, n\}$. The proof, however, is slightly more involved in this case, and depends on Lemma 4.

Let s, r, and t be the same as before. For a fixed affine subspace $H \subseteq \{0,1\}^t$ of dimension r, such that $H = \{a_1, \ldots, a_l\}$ for $l = 2^r$, and for a fixed $1 \leqslant i \leqslant n$, let $\mathcal{E}_j$, $1 \leqslant j \leqslant l$, be the event that $\psi(a_j) = i$ when ψ is chosen from the (ϵ, k)-independent sample space defined above. Then Lemma 4 applies with $\theta = 2^{-\lceil \log n\rceil}$, yielding that the probability that ψ misses the value i on the subspace H is

$$\mathbf{Pr}[\wedge_{j=1}^{l}\bar{\mathcal{E}}_j] \leqslant e^{-2^{r-\lceil \log n\rceil}} + \binom{2^r}{k+1}(\epsilon k + 2^{-k\lceil \log n\rceil}). \tag{3}$$

It is easy to see that the first term on the right-hand side of (3) is at most $e^{-4\log^2 n}$ (when $b = 3$ in r). We need to bound from above the remaining two terms: $\binom{2^r}{k+1}2^{-k\lceil \log n\rceil}$ and $\binom{2^r}{k+1}\epsilon k$. Using Stirling's formula, one can show that the first of these two terms can be made at most $2^{-4\log^2 n}$, by choosing c_2 sufficiently large. Having fixed c_2, we can also make the second of the terms at most $2^{-4\log^2 n}$, by choosing $c_3 > c_2$ sufficiently large. It is then straightforward to verify that the probability that ψ misses at least one value i, $1 \leqslant i \leqslant n$, on at least one affine subspace of dimension r tends to 0 as n grows. ■

Using any efficient construction of almost independent sample spaces, for example, Pow_N^{2m} with $N = tn \in O(n \log n)$ and $m \in O(\log^2 n)$, we can find a matrix A with the required properties in $\mathrm{DTIME}(2^{O(\log^2 n)})$ by searching through all elements of the sample space and checking whether any of them yields a desired matrix. Analogously, we can find the required function ψ in $\mathrm{DTIME}(2^{O(\log^3 n)})$, by considering, e.g., $\mathrm{Pow}_{N'}^{2m'}$ with $N' = 2^t\lceil \log n\rceil$ and $m' \in O(\log^3 n)$. Thus, constructing both A and ψ can be carried out in quasipolynomial time.

Given the corresponding advice strings of $O(\log^3 n)$ bits, ψ is computable in time polylog(n) and all elements of A can be computed in time $n$polylog(n). So, in this case, the function $\phi(\vec{x}) = \psi(A\vec{x})$ is computable in QL. Hence, by "hard-wiring" good advice strings, we get the function $f(\vec{x}) = x_{\phi(\vec{x})}$ computable by quasilinear-size circuits, while requiring 1-b.p.'s of size at least $2^{n-(5+\epsilon)\log n}$, for any $\epsilon > 0$ and sufficiently large n; these parameters appear to be better than those in [ABCR97]. By making the advice strings a part of the input, we obtain a function in QL that requires 1-b.p.'s of size at least $2^{n-O(\log^3 n)}$.

Acknowledgements. I am grateful to Alexander Razborov for referring me to [Raz88]. I would like to thank Petr Savický for his comments on a preliminary version of this paper, and Dieter van Melkebeek for helpful discussions. Finally, I want to thank my supervisor, Stephen Cook, for his encouragement and support.

References

[ABCR97] A.E. Andreev, J.L. Baskakov, A.E.F. Clementi, and J.D.P. Rolim. Small pseudo-random sets yield hard functions: New tight explicit lower bounds for branching programs. *Electronic Colloquium on Computational Complexity*, TR97-053, 1997.

[AGHP92] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost $k-$wise independent random variables. *Journal of Random Structures and Algorithms*, 3(3):289–304, 1992. (preliminary version in the 31st FOCS).

[NN93] J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993. (preliminary version in the 22nd STOC).

[Raz88] A.A. Razborov. Bounded-depth formulae over $\{\&, \oplus\}$ and some combinatorial problems. In S. I. Adyan, editor, *Problems of Cybernetics. Complexity Theory and Applied Mathematical Logic*, pages 149–166. VINITI, Moscow, 1988. (in Russian).

[SS93] J. Simon and M. Szegedy. A new lower bound theorem for read-only-once branching programs and its applications. In *Advances in Computational Complexity (J. Cai, editor)*, pages 183–193. AMS-DIMACS Series, 1993.

[SZ96] P. Savický and S. Zák. A large lower bound for 1-branching programs. *Electronic Colloquium on Computational Complexity*, TR96-036, 1996.

A Proof of Lemma 4

We first consider the case where k is even. Let $\mathcal{C}_1, \ldots, \mathcal{C}_l$ be independent events, each having the success probability θ. Applying the Boole-Bonferroni inequality to $\mathbf{Pr}[\vee_{i=1}^{l}\mathcal{E}_i]$ and $\mathbf{Pr}[\vee_{i=1}^{l}\mathcal{C}_i]$, we obtain that

$$\mathbf{Pr}[\vee_{i=1}^{l}\mathcal{E}_i] \geqslant \sum_{\nu=1}^{k}(-1)^{\nu+1}\sum_{|I|=\nu}\mathbf{Pr}[\wedge_{i\in I}\mathcal{E}_i] \tag{4}$$

and

$$\mathbf{Pr}[\vee_{i=1}^{l}\mathcal{C}_i] \leqslant \sum_{\nu=1}^{k}(-1)^{\nu+1}\sum_{|I|=\nu}\theta^{|I|} + \sum_{|I|=k+1}\theta^{k+1}. \tag{5}$$

The assumption of the lemma that $\mathcal{E}_1, \ldots, \mathcal{E}_l$ are almost k-wise independent implies that the right-hand side in (4) is at least

$$\sum_{\nu=1}^{k}(-1)^{\nu+1}\sum_{|I|=\nu}\theta^{|I|} - \epsilon k\binom{l}{k}. \tag{6}$$

On the other hand, the independence of $\mathcal{C}_1, \dots, \mathcal{C}_l$ implies that

$$\mathbf{Pr}[\vee_{i=1}^{l} \mathcal{C}_i] = 1 - (1-\theta)^l \geqslant 1 - e^{-\theta l}. \tag{7}$$

Combining (4), (6), (5), and (7) yields (for even k) that

$$\begin{aligned} \mathbf{Pr}[\vee_{i=1}^{l} \mathcal{E}_i] &\geqslant 1 - e^{-\theta l} - \epsilon k \binom{l}{k} - \theta^{k+1} \binom{l}{k+1} \\ &\geqslant 1 - e^{-\theta l} - \binom{l}{k+1} (\epsilon k + \theta^{k+1}). \end{aligned}$$

In the case where k is odd, we use the above argument with $k-1$ substituted for k. This completes the proof.