Set Membership with Non-Adaptive Bit Probes

Mohit Garg*[1] and Jaikumar Radhakrishnan[2]

1 Tokyo Institute of Technology, Tokyo, Japan
garg.m.aa@m.titech.ac.jp

2 Tata Institute of Fundamental Research, Mumbai, India
jaikumar@tifr.res.in

Abstract

We consider the non-adaptive bit-probe complexity of the set membership problem, where a set S of size at most n from a universe of size m is to be represented as a short bit vector in order to answer membership queries of the form "Is x in S?" by *non-adaptively* probing the bit vector at t places. Let $s_N(m,n,t)$ be the minimum number of bits of storage needed for such a scheme. Buhrman, Miltersen, Radhakrishnan, and Srinivasan [4] and Alon and Feige [1] investigated $s_N(m,n,t)$ for various ranges of the parameter t. We show the following.

General upper bound ($t \geq 5$ and odd): For odd $t \geq 5$, $s_N(m,n,t) = O(tm^{\frac{2}{t-1}} n^{1-\frac{2}{t-1}} \lg \frac{2m}{n})$. This improves on a result of Buhrman *et al.* that states for odd $t \geq 5$, $s_N(m,n,t) = O(m^{\frac{4}{t+1}} n)$. For small values of t (odd $t \geq 3$ and $t \leq \frac{1}{10} \lg \lg m$) and $n \leq m^{1-\epsilon}$ ($\epsilon > 0$), we obtain adaptive schemes that use a little less space: $O(\exp(e^{2t}) m^{\frac{2}{t+1}} n^{1-\frac{2}{t+1}} \lg m)$.

Three probes ($t = 3$) lower bound: We show that $s_N(m,n,3) = \Omega(\sqrt{mn})$ for $n \geq n_0$ for some constant n_0. This improves on a result of Alon and Feige that states that for $n \geq 16 \lg m$, $s_N(m,n,3) = \Omega(\sqrt{\frac{mn}{\lg m}})$. The complexity of the non-adaptive scheme might, in principle, depend on the function that is used to determine the answer based on the three bits read (one may assume that all queries use the same function). Let $s_N^f(m,n,3)$ be the minimum number of bits of storage required in a three-probe non-adaptive scheme where the function $f : \{0,1\}^3 \rightarrow \{0,1\}$ is used to answer the queries. We show that for large class of functions f (including the majority function on three bits), we in fact have $s_N(m,n,3) = \Omega(m^{1-\frac{1}{cn}})$ for $n \geq 4$ and some $c > 0$. In particular, three-probe non-adaptive schemes that use such query functions f do not give any asymptotic savings over the trivial characteristic vector when $n \geq \log m$.

1998 ACM Subject Classification E.1 Data Structures, E.4 Coding and Information Theory

Keywords and phrases Data Structures, Bit-probe model, Compression, Bloom filters, Expansion

Digital Object Identifier 10.4230/LIPIcs.STACS.2017.38

1 Introduction

The *set membership problem* is a fundamental problem in the area of data structures and information compression and retrieval. In its abstract form we are given a subset S of size at most n from a universe of size m and required to represent it as a bit string so that membership queries of the form "Is x in S?" can be answered using a small number of probes into the bit string. The characteristic function representation provides a solution to this problem: just one bit-probe is needed to answer queries, but all sets are represented using m-bit strings (which is very wasteful when n is promised to be small).

[*] A part of this work was done when the first author was at the Tata Institute of Fundamental Research.

34th Symposium on Theoretical Aspects of Computer Science (STACS 2017).
Editors: Heribert Vollmer and Brigitte Vallée; Article No. 38; pp. 38:1–38:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The trade-off between the number of bits in the representation and the number of probes is the subject of several previous works: it was studied by Minsky and Papert in their 1969 book *Perceptrons* [11]; more recently, Buhrman, Miltersen, Radhakrishnan and Venkatesh [4] showed the existence of randomized schemes that answer queries with just one bit probe and use near optimal space. In contrast, they showed that deterministic schemes that answer queries by making a constant number of probes cannot use optimal space. The deterministic worst-case trade-off for this problem was also considered in the same paper and in several subsequent works (e.g., Radhakrishnan, Raman and Rao [12], Alon and Feige [1], Radhakrishnan, Shah and Shannigrahi [14], Viola [15], Lewenstein, Munro, Nicholson and Raman [8], Garg and Radhakrishnan [5]). For sets where each element is included with probability p, Makhdoumi, Huang, Médard and Polyanskiy [10] showed, in particular, that no savings over the characteristic vector can be obtained in this case for non-adaptive schemes with t = 2.

In this work, we focus on deterministic schemes with *non-adaptive* probes, where the probes are made in parallel (or equivalently the location of probes do not depend on the value read in previous probes). Such schemes have been studied in several previous works. Let $s_N(m, n, t)$ be the minimum number of bits of storage required in order to answer membership queries with t non-adaptive probes.

▶ **Definition 1.** A non-adaptive (m, n, s, t)-scheme consists of a storage function and a query scheme. The storage function has the form $\phi : \binom{[m]}{\leq n} \to \{0,1\}^s$ that takes a set of size at most n and returns its s-bit representation. The query scheme associates with each element x the t probe locations $(i_1(x), \ldots, i_t(x)) \in [s]^t$ and a function $f_x : \{0,1\}^t \to \{0,1\}$. We require that for all $S \in \binom{[m]}{\leq n}$ and all $x \in [m]$: $x \in S$ iff $f_x(\phi(S)[i_1(x)], \phi(S)[i_2(x)], \ldots, \phi(S)[i_t(x)]) = 1$. Let $s_N(m, n, t)$ denote the minimum s such that there is an (m, n, s, t)-scheme.

In our discussion, we use $s(m, n, t)$ (without the subscript N) to denote the minimum space required for adaptive schemes. Using the above notation, we now describe our results and their relation to what was known before. All asymptotic claims below hold for large m.

1.1 General non-adaptive schemes

▶ **Theorem 2** (Result 1, non-adaptive schemes)**.** *For odd $t \geq 5$, we have*

$$s_N(m, n, t) = O(tm^{\frac{2}{t-1}} n^{1-\frac{2}{t-1}} \lg \frac{2m}{n}).$$

In comparison, for odd $t \geq 5$, Buhrman *et al.* showed that $s_N(m, n, t) = O(m^{\frac{4}{t+1}} n)$. The exponent of m in their upper bound result is roughly four times the exponent of m appearing in their lower bound result. Their schemes are non-adaptive and use the MAJORITY function to answer membership queries. We exhibit schemes that still use MAJORITY but need less space. Buhrman *et al.* also show a lower bound of $s(m, n, t) = \Omega(tm^{\frac{1}{t}} n^{1-\frac{1}{t}})$ valid (even for adaptive schemes) when $n \leq m^{1-\epsilon}$ (for $\epsilon > 0$ and $t \ll \lg m$). Note that the exponent of m in our result is twice the exponent of m appearing in the lower bound result. These schemes, as well as the non-adaptive scheme for $t = 4$ due to Alon and Feige [1], have implications for the problem studied by Makhdoumi *et al.* [10]; unlike in the case of $t = 2$, significant savings are possible if $t \geq 4$, even with non-adaptive schemes[1]. Using a similar proof idea, we obtain slightly better upper bound with adaptive schemes when t is small and n is at most $m^{1-\epsilon}$.

[1] We are grateful to Tom Courtade and Ashwin Pananjady for this observation.

▶ **Theorem 3** (Result 2, adaptive schemes). *For odd $t \geq 3$ and $t \leq \frac{1}{10} \lg \lg m$ and for $n \leq m^{1-\epsilon}$ ($\epsilon > 0$), we have $s(m, n, t) = O(\exp(e^{2t}) m^{\frac{2}{t+1}} n^{1-\frac{2}{t+1}} \lg m)$.*

Technique. To justify our claim, we need to describe the query scheme, that is, $(i_1(x), i_2(x), \ldots, i_t(x))$ for each $x \in [m]$ and the query function $f_x : \{0,1\}^t \to \{0,1\}$. For f_x we use the MAJORITY on t bits (t is odd). The locations to be probed for each element will be obtained using a probabilistic argument. Once a query scheme is fixed, we need to show how the assignment to the memory is obtained. For this, we describe a sequential algorithm. We show that the random assignment of locations ensures sufficient expansion allowing us to start with a greedy argument arrange that most queries are answered correctly, and then use Hall's bipartite graph matching theorem to find the required assignment for the remaining elements. Versions of this argument have been used in previous works [9, 4, 7, 1, 5].

1.2 Three non-adaptive probes

For one probe and $m \geq 2$, it is easy to show that no space can be saved over the characteristic vector representation. For two non-adaptive probes, only for the special case $n = 1$, some non-trivial savings over the characteristic vector representation are possible: $s_N(m, 1, 2) = \theta(\sqrt{m})$). For $n \geq 2$, Buhrman *et al.* [4] showed $s_N(m, n, 2) = m$. The smallest number of probes for which the complexity of problem with non-adaptive probes is not settled is three. Observe that any scheme with two adaptive probes can be converted to a scheme with three non-adaptive probes; the two probe decision tree has at most three nodes. Thus, using the two adaptive probes upper bound result of Garg and Radhakrishnan [5], we have $s_N(m, n, 3) \leq s(m, n, 2) = O(m^{1-\frac{1}{4n+1}})$. Thus, non-trivial savings in space over the characteristic vector representation is possible when $n = o(\lg m)$. Consequently, the question is whether more space can be saved or is this upper bound tight? We are not aware of any three-probe non-adaptive scheme that manages with $o(m)$ space for sets of size $\omega(\lg m)$. Alon and Feige [1] show the following lower bound: $s_N(m, n, 3) = \Omega(\sqrt{\frac{mn}{\lg m}})$ for $n \geq 16 \lg m$.

In order to obtain better lower bounds for three-probe non-adaptive schemes, we proceed as follows. In any three-probe non-adaptive scheme, the query scheme specifies, for each element, the three locations to probe and a three variable boolean function to be applied on three values read. In principle, for different elements, the query scheme can specify different boolean functions. But since there are only a finite number (256) of boolean functions on three variables, some set of at least $m/256$ elements of the universe use a common function. We may thus restrict attention to this part of the universe, and assume that the function being employed to answer queries is always the same. Furthermore, we may place functions obtained from one another by negating and permuting variables in a common equivalence class, and restrict our attention to one representative in each class. For three variable boolean functions, Pólya counting yields that there are twenty-two equivalence classes. This classification of the 256 functions into twenty-two classes is already available in the literature [16]. We show the following.

▶ **Theorem 4** (Result 3).

(a) *If the query function $f : \{0,1\}^3 \to \{0,1\}$ is not equivalent to $(x, y, z) \mapsto (x \wedge y) \oplus z$ or $(x, y, z) \mapsto 1$ iff $x + y + z = 1$, then $s_N(m, n, 3) = \Omega(m^{1-\frac{1}{cn}})$ for $n \geq 4$ and some $c > 0$.*

(b) *If the query function $f : \{0,1\}^3 \to \{0,1\}$ is equivalent to $(x, y, z) \mapsto (x \wedge y) \oplus z$ or $(x, y, z) \mapsto 1$ iff $x + y + z = 1$, then $s_N(m, n, 3) = \Omega(\sqrt{mn})$.*

(c) *If the query function $f : \{0,1\}^3 \to \{0,1\}$ is equivalent to $(x, y, z) \mapsto (x \wedge y) \oplus z$ and $\lg m \leq n \leq \frac{m}{\lg m}$, then $s_N(m, n, 3) = \Omega(\sqrt{mn \frac{\lg \frac{m}{n}}{\lg \lg m}})$.*

The best upper bounds for non-adaptive schemes with four or more probes use the MAJORITY function to answer membership queries. Our result implies that for three non-adaptive probes, when queries are answered by computing MAJORITY, the space required is at least $\Omega(m^{1-\frac{1}{cn}})$ for some constant c. In fact, similar lower bound holds if membership queries are answered using most boolean functions. Our results do not yield a similar lower bound for $(x, y, z) \mapsto (x \wedge y) \oplus z$ and $(x, y, z) \mapsto 1$ iff $x + y + z = 1$ types. For these two types of query functions, we get a slightly better lower bound than what is implied by [1]. Thus, further investigations on three probes non-adaptive schemes need to focus on just $(x, y, z) \mapsto 1$ iff $x + y + z = 1$ and $(x, y, z) \mapsto (x \wedge y) \oplus z$ as the query functions.

Technique. As mentioned above, there are twenty-two types of functions for which we need to prove a lower bound. Seven of the twenty-two classes contain functions that can be represented by a decision tree of height at most two. Thus, for these functions, the two probe adaptive lower bound in [5] implies the result. These functions are: constant 0, constant 1, the DICTATOR function $(x, y, z) \mapsto x$, the function $(x, y, z) \mapsto x \wedge y$, its complement $(x, y, z) \mapsto \bar{x} \vee \bar{y}$, $(x, y, z) \mapsto (x \wedge y) \vee (\bar{x} \wedge z)$, and $(x, y, z) \mapsto (x \wedge y) \vee (\bar{x} \wedge \bar{y})$.

After this, fifteen classes remain. Functions in some eleven of the remaining fifteen classes admit a density argument, similar in spirit to the adaptive two-probes lower-bound proof in [5]. To streamline the argument, we classify these eleven classes into two parts. The first part contains the MAJORITY function. The second part contains the AND function, the ALL-EQUAL function, the functions $(x, y, z) \mapsto (x \oplus y) \wedge z$, $(x, y, z) \mapsto (x \vee y) \wedge z$, $(x, y, z) \mapsto (x \wedge y \wedge z) \vee (\bar{y} \wedge \bar{z})$, and their complements. For functions in the second part we deal with two functions—a function and its complement—with a single proof. In these proofs, we produce sets S and T of size at most n such that storing S and not storing T leads to a contradiction. The proof for the complement function works with a small twist: storing T and not storing S leads to the contradiction. Thus, these eleven cases are handled by six proofs. In each of these proofs we roughly argue (sometimes probabilistically) that if the scheme is valid, it must conceal a certain dense graph that avoids small cycles. Standard graph theoretic results (the Moore bound) that relate density and girth then gives us the lower bound.

For the remaining four classes, we employ linear-algebraic arguments. Representatives chosen from these classes are PARITY, $(x, y, z) \mapsto 1$ iff $x + y + z \neq 1$, $(x, y, z) \mapsto (x \wedge y) \oplus z$, and $(x, y, z) \mapsto 1$ iff $x + y + z = 1$. For PARITY and $(x, y, z) \mapsto 1$ iff $x + y + z \neq 1$, we show using standard dimension argument, that if the space used is smaller than the universe size m, then there is some element $u \in [m]$ that is (linearly) dependent on the other elements. Not storing the other elements, leaves the scheme with no choice for u, thus leading to a contradiction. For $(x, y, z) \mapsto (x \wedge y) \oplus z$ and $(x, y, z) \mapsto 1$ iff $x + y + z = 1$ a modification of an algebraic argument of Radhakrishnan, Sen and Venkatesh [13] implies a lower bound of $\sqrt{mn}$. (Interestingly, we need to choose an appropriate characteristic of the field (2 or 3) based on which function we deal with.) For $(x, y, z) \mapsto (x \wedge y) \oplus z$, we further improve on this argument by employing random restrictions. These results together improve the previous best lower bound (due to Alon and Feige [1]) irrespective of the query function used.

2 General non-adaptive upper bound

In this section, we prove the general non-adaptive upper bound result: Theorem 2.

▶ **Definition 5.** A non-adaptive (m, s, t)-graph is a bipartite graph G with vertex sets $U = [m]$ and V ($|V| = ts$). V is partitioned into t disjoint sets: $V_1, \ldots, V_t$; each V_i has s vertices. Every $u \in U$ has a unique neighbour in each V_i. A non-adaptive (m, s, t)-graph naturally gives rise to a non-adaptive (m, ts, t)-query scheme $\mathcal{T}_G$ as follows. We view the memory (an array L of ts bits) to be indexed by vertices in V. On receiving the query "Is u in S?", we answer "Yes" iff the MAJORITY of the locations in the neighbourhood of u contain a 1. We say that the query scheme $\mathcal{T}_G$ is satisfiable for a set $S \subseteq [m]$, if there is an assignment to the memory locations $(L[v] : v \in V)$, such that $\mathcal{T}_G$ correctly answers all queries of the form "Is x in S?".

We now restrict attention to odd $t \geq 5$. First, we identify an appropriate property of the underlying non-adaptive (m, s, t)-graph G that guarantees that $\mathcal{T}_G$ is satisfiable for all sets S of size at most n. We then show that such a graph exists for some $s = O(m^{\frac{2}{t-1}} n^{1-\frac{2}{t-1}} \lg \frac{2m}{n})$.

▶ **Definition 6** (Non-adaptive admissible graph)**.** We say that a non-adaptive (m, s, t)-graph G is admissible for sets of size at most n if the following two properties hold:

(P1) $\forall R \subseteq [m]$ $(|R| \leq n + \lceil 2n \lg \frac{2m}{n} \rceil)$: $|\Gamma_G(R)| \geq \frac{t+1}{2}|R|$, where $\Gamma_G(R)$ is the set of neighbors of R in G.

(P2) $\forall S \subseteq [m]$ $(|S| = n)$: $|T_S| \leq \lceil 2n \lg \frac{2m}{n} \rceil$, where $T_S = \{y \in [m] \setminus S : |\Gamma_G(y) \cap \Gamma_G(S)| \geq \frac{t+1}{2}\}$.

Our theorem will follow from the following claims.

▶ **Lemma 7.** *If a non-adaptive (m, s, t)-graph G is admissible for sets of size at most n, then the non-adaptive (m, ts, t)-query scheme $\mathcal{T}_G$ is satisfiable for every set S of size at most n.*

▶ **Lemma 8.** *There is a non-adaptive (m, s, t)-graph, with $s = O(m^{\frac{2}{t-1}} n^{1-\frac{2}{t-1}} \lg \frac{2m}{n})$, that is admissible for every set $S \subseteq [m]$ of size at most n.*

Proof of Lemma 7. Fix an admissible graph G. Thus, G satisfies (P1) and (P2) above. Fix a set $S \subseteq [m]$ of size at most n. We will show that there is a 0-1 assignment to the memory such that all queries are answered correctly by $\mathcal{T}_G$.

Let $S' \subseteq [m]$ be such that $S \subseteq S'$ and $|S'| = n$. From (P2), we know $|T_{S'}| \leq \lceil 2n \lg \frac{2m}{n} \rceil$. Hence, $|S' \cup T_{S'}| \leq n + \lceil 2n \lg \frac{2m}{n} \rceil$. From (P1) and Hall's theorem, we may assign to each element $u \in S' \cup T_{S'}$ a set $A_u \subseteq V$ such that (i) $|A_u| = \frac{t+1}{2}$ and (ii) the A_u's are disjoint. For each $u \in S \subseteq S'$, we assign the value 1 to all locations in A_u. For each $u \in (S' \cup T_{S'}) \setminus S$, we assign the value 0 to all locations in A_u. Since $\frac{t+1}{2} > \frac{t}{2}$, all queries for $u \in S' \cup T_{S'}$ are answered correctly.

Assign 0 to all locations in $\Gamma_G([m] \setminus (S' \cup T_{S'}))$. For $y \in [m] \setminus (S' \cup T_{S'})$, $|\Gamma_G(y) \cap \Gamma_G(S)| \leq \frac{t-1}{2}$. As a result, queries for elements in $[m] \setminus (S' \cup T_{S'})$ are answered correctly, as the majority evaluates to 0 for each one of them. ◀

Proof of Lemma 8. In the following, set

$$s = \left\lceil 60 m^{\frac{2}{t-1}} n^{1-\frac{2}{t-1}} \lg \frac{2m}{n} \right\rceil.$$

We show that a suitable random non-adaptive (m, s, t)-graph G is admissible for sets of size at most n with positive probability. The graph G is constructed as follows. Recall that $V = \bigcup_i V_i$. For each $u \in U$, one neighbor is chosen uniformly and independently in each V_i.

(P1) holds. If (P1) fails, then for some non-empty $W \subseteq U$, $(|W| \leq n + \lceil 2n \lg \frac{2m}{n} \rceil)$, we have $|\Gamma_G(W)| \leq \frac{t+1}{2}|W| - 1$. Fix a set W of size $r \geq 1$ and $L \subseteq V$ of size $\frac{t+1}{2}r - 1$. Let L have ℓ_i elements in V_i; thus, $\sum_i \ell_i = \frac{t+1}{2}r - 1$. Then,

$$\Pr[\Gamma_G(W) \subseteq L] \leq \prod_{i=1}^{t} \left(\frac{\ell_i}{|V_i|}\right)^r \leq \left(\frac{(\frac{t+1}{2})r - 1}{ts}\right)^{tr},$$

where the last inequality is a consequence of GM $\leq$ AM. We conclude, using the union bound over choices of W and L, that (P1) fails with probability at most

$$\begin{aligned} & \sum_{r=1}^{n+\lceil 2n\lg\frac{2m}{n}\rceil} \binom{m}{r}\binom{ts}{\frac{t+1}{2}r-1}\left(\frac{\frac{t+1}{2}r-1}{ts}\right)^{tr} & (1) \\ \leq & \sum_{r=1}^{n+\lceil 2n\lg\frac{2m}{n}\rceil} \left(\frac{em}{r}\right)^{r}\left(\frac{tes}{\frac{t+1}{2}r-1}\right)^{\frac{t+1}{2}r-1}\left(\frac{\frac{t+1}{2}r-1}{ts}\right)^{tr} & \\ \leq & \sum_{r=1}^{n+\lceil 2n\lg\frac{2m}{n}\rceil} \left[\frac{(e^{\frac{t+3}{2}-\frac{1}{r}})mr^{\frac{t-1}{2}-1+\frac{1}{r}}}{(s^{\frac{1}{r}})s^{\frac{t-1}{2}}}\right]^{r} \leq \frac{1}{3}, & (2) \end{aligned}$$

where the last inequality holds because we have chosen s large enough.

(P2) holds. For (P2) to fail, there must exist a set $S \subseteq [m]$ of size n such that $|T_S| > \lceil 2n\lg\frac{2m}{n}\rceil$. Fix a set S of size n. Fix a $y \in [m]\setminus S$.

$$\Pr[y \in T_S] \leq \binom{t}{\frac{t+1}{2}}\left(\frac{n}{s}\right)^{\frac{t+1}{2}} \leq \frac{n}{10m},$$

where the last inequality holds because of choice of s and m is large. Thus, $\mathbb{E}[|T_S|] \leq \frac{n}{10}$. To conclude that $|T_S|$ is bounded with high probability, we will use the following version of Chernoff bound: if $X = \sum_{i=1}^{N} X_i$, where each random variable $X_i \in \{0,1\}$ independently, then if $\gamma > 2e\mathbb{E}[X]$, then $\Pr[X > \gamma] \leq 2^{-\gamma}$. Then, for all large m,

$$\Pr[|T_S| > 2n\lg\frac{2m}{n}] \leq 2^{-2n\lg\frac{2m}{n}}.$$

Using the union bound, we conclude that

$$\Pr[\text{(P2) fails}] \leq \left(\frac{em}{n}\right)^{n} 2^{-2n\lg\frac{2m}{n}} \leq \frac{1}{3}.$$

Thus, with probability at least $\frac{1}{3}$ the random graph G is admissible. ◀

3 Three non-adaptive probes lower bound

In this section, we prove the three probe lower bound result: Theorem 4.

▶ **Definition 9** (Equivalent). Two boolean functions are called equivalent if one can be obtained from the other by negating and permuting the variables.

▶ **Proposition 10.** *Let $f, g : \{0,1\}^t \to \{0,1\}$ be equivalent. If s_1 and s_2 are the minimum bits of space required for non-adaptive (m, n, s_1, t) and (m, n, s_2, t)-schemes with query functions f and g respectively, then $s_1 = s_2$.*

For three variable boolean functions, there are twenty-two equivalence classes (see [16]). To prove Theorem 4, we provide proofs for these twenty-two query functions, each from a different class. In many proofs below we assume that the memory consists of three arrays of size s each, and the three probes are made on different arrays. Given any scheme that uses space s, we can always modify it to meet our assumption, by expanding the space by factor 3.

3.1 Decision trees of height two

Seven of the twenty-two classes contain functions that can be represented by a decision tree of height at most two. Thus, for these functions, the two probe adaptive lower bound [5] implies the result. These functions are: constant 0, constant 1, the DICTATOR function $(x, y, z) \mapsto x$, the function $(x, y, z) \mapsto x \wedge y$, its complement $(x, y, z) \mapsto \bar{x} \vee \bar{y}$, $(x, y, z) \mapsto (x \wedge y) \vee (\bar{x} \wedge z)$, and $(x, y, z) \mapsto (x \wedge y) \vee (\bar{x} \wedge \bar{y})$.

3.2 MAJORITY

Let Φ be a non-adaptive $(m, n, s, 3)$-scheme with MAJORITY as the query function. The memory is a bit array $A[1, \cdots, s]$ of length s. For each element $u \in [m]$, $x(u), y(u), z(u) \in [s]$ are the three distinct locations in A that are probed to determine whether u is in the set or not. For each set $S \subseteq [m]$ of size at most n, the assignment $\sigma(S) \in \{0, 1\}^s$ to A is such that for all elements $u \in [m]$, $\mathsf{Maj}(A[x(u)], A[y(u)], A[z(u)])$ is 1 iff $u \in S$, where Maj is the MAJORITY of 3 bits.

▶ **Definition 11.** (model-graph for Φ, third vertex, meet) Let Φ be a $(m, n, s, 3)$-scheme with MAJORITY as the query function. Fix a graph G such that $V(G) = [s]$, $|E(G)| = m$ and edge labels: $\{\text{lab}(e)|e \in E(G)\} = [m]$ (there is a unique edge for each label in $[m]$). G is called a *model-graph* for Φ if for each $u \in [m]$ the edge labelled u has the set of endpoints in $\{\{x(u), y(u)\}, \{y(u), z(u)\}, \{z(u), x(u)\}\}$. For example, the graph $G = ([s], \{x(u) \xleftrightarrow{u} y(u)|u \in [m]\})$ is a model graph for Φ.

In a model-graph for Φ, let e be the set of endpoints of the edge with label u. The element in the singleton $(\{x(u), y(u), z(u)\} \setminus e)$ is defined to be the *third* vertex of u.

Two edge-disjoint cycles C_1 and C_2 are said to *meet* in a model-graph for Φ if there exist elements $u, v \in [m]$ such that the third vertices of u and v are the same vertex and the edges labelled u and v are in the different cycles C_1 and C_2 respectively.

▶ **Definition 12.** A model-graph G for an $(m, n, s, 3)$-scheme with MAJORITY as the query function is said to be *forced* if at least one of the following three conditions hold.

(P1) $\exists$ edge-disjoint odd cycles C_1, C_2 in G with lengths at most n each that intersect at a vertex.

(P2) $\exists$ edge disjoint even cycles C_1, C_2 in G with lengths at most n each and C_1 and C_2 meet.

(P3) $\exists$ an even cycle C of length at most n, such that some two edges in C, labelled e and f say, have an even number of edges between them (while traversing the edges of the cycle in order) and the third vertices of e and f are the same vertex.

▶ **Lemma 13.** *A model-graph for a scheme with MAJORITY as the query function cannot be forced.*

▶ **Lemma 14.** *Any* $\left(m, n, \left\lfloor \frac{1}{6} m^{1-\frac{1}{\lfloor \frac{n}{2} \rfloor + 1}} \right\rfloor, 3\right)$*-scheme with MAJORITY as the query function has a forced model-graph.*

From lemmas 13 and 14, it follows that when MAJORITY is used as the query function, $s_N(m, n, 3) > \frac{1}{6} m^{1-\frac{1}{\lfloor \frac{n}{2} \rfloor + 1}}$.

Proof of Lemma 13. Fix a $(m, n, s, 3)$-scheme Φ with MAJORITY as the query function. Fix a model-graph G for Φ. Assume G is forced, that is, it satisfies (P1) or (P2) or (P3) above.

Case: (P1) holds. (P1) implies that there are edge-disjoint cycles C_1 and C_2 in G such that,

$$C_1 : u_0 \xrightarrow{e_1} u_1 \xrightarrow{e_2} \cdots \xrightarrow{e_{2k+1}} u_{2k+1} = u_0,$$

$$C_2 : u_0 \xrightarrow{f_1} v_1 \xrightarrow{f_2} \cdots \xrightarrow{f_{2l+1}} v_{2l+1} = u_0,$$

and $2k+1, 2l+1 \leq n$. Let $S_0 = \{e_1, e_3, \cdots, e_{2k+1}\} \cup \{f_2, f_4, \cdots, f_{2l}\}$ and $S_1 = \{e_2, e_4, \cdots, e_{2k}\} \cup \{f_1, f_3, \cdots, f_{2l+1}\}$. Note that $|S_0| = |S_1| \leq n$. We claim that Φ cannot represent any set S such that

$$S_0 \subseteq S \subseteq \bar{S_1}.$$

In particular, Φ cannot represent the set S_0. Assume Φ represents such an S. We claim that u_0 cannot be assigned a 0. If u_0 is assigned a 0, then since $e_1 \in S$, u_1 must be assigned a 1. Otherwise, $\mathsf{Maj}(A[x(e_1)], A[y(e_1)], A[z(e_1)]) = \mathsf{Maj}(0, 0, b) = 0$, where b is the bit assigned to the location in $\{x(e_1), y(e_1), z(e_1)\} \setminus \{u_0, u_1\}$. Since, u_1 is assigned a 1 and $e_2 \notin S$, u_2 must be assigned a 0. Similarly, since $e_3 \in S$, u_3 must be assigned a 1 and so on. Finally, $u_{2k+1} = u_0$ must be assigned a 1. A contradiction. Hence u_0 cannot be assigned a 0.

Again, we claim u_0 cannot be assigned a 1. For if u_0 is assigned a 1, since $f_1 \notin S$, v_1 must be assigned a 0. Again, since $f_2 \in S$, v_2 must be assigned a 1 and so on. Finally, $v_{2k+1} = u_0$ must be assigned a 0. A contradiction.

Since u_0 can neither be assigned a 0 or a 1, we get a contradiction.

▶ Remark. In the proofs below, we will often encounter similar arguments, where we will have a cycle of dependencies: assigning a particular bit to a location will force the assignment to the next location along the cycle.

Case: (P2) holds. (P2) implies that there are edge-disjoint cycles C_1 and C_2 in G such that,

$$C_1 : u_0 \xrightarrow{e_1} u_1 \xrightarrow{e_2} \cdots \xrightarrow{e_{2k}} u_{2k} = u_0,$$

$$C_2 : v_0 \xrightarrow{f_1} v_1 \xrightarrow{f_2} \cdots \xrightarrow{f_{2l}} v_{2l} = v_0,$$

$2k, 2l \leq n$, and the third vertices of e_1 and f_1 are the same vertex w. Let $S_0 = \{e_1, e_3, \cdots, e_{2k-1}\} \cup \{f_2, f_4, \cdots, f_{2l}\}$ and $S_1 = \{e_2, e_4, \cdots, e_{2k}\} \cup \{f_1, f_3, \cdots, f_{2l-1}\}$. Note that $|S_0| = |S_1| \leq n$. We claim that Φ cannot represent any set S such that

$$S_0 \subseteq S \subseteq \bar{S_1}.$$

In particular, Φ cannot represent the set S_0. Assume Φ represents such an S. Since $e_1 \in S$, either the location u_0 or the location u_1 of the memory A must be assigned a 1, otherwise $\mathsf{Maj}(A[x(e_1)], A[y(e_1)], A[z(e_1)]) = \mathsf{Maj}(A[u_0], A[u_1], A[w]) = \mathsf{Maj}(0, 0, A[w]) = 0$. Assume u_1 is assigned a 1. Then, since e_2 is not in the set, using a similar argument, u_2 must be assigned a 0. Similarly, u_3 must be assigned a 1 and so on. Finally, $u_{2k} = u_0$ must be assigned a 0. Similarly, if u_0 was assigned a 1, then u_1 must be assigned a 0. Thus, $\mathsf{Maj}(x(e_1), y(e_1), z(e_1)) = \mathsf{Maj}(0, 1, A[w]) = A[w]$. Hence w must be assigned a 1.

Again, since f_1 is not in S, either v_0 or v_1 is assigned a 0. If v_1 is assigned 0, v_2 must be assigned a 1, v_3 a 0, and so on. Finally, $v_{2l} = v_0$ must be assigned a 1. Similarly, if v_0 is assigned 1, then v_1 is assigned a 0. Therefore, $\mathsf{Maj}(x(f_1), y(f_1), z(f_1)) = \mathsf{Maj}(A[v_0], A[v_1], A[w]) = \mathsf{Maj}(0, 1, A[w]) = A[w]$. Since $f_1 \notin S$, w must be assigned a 0. A contradiction.

Case: (P3) holds. (P3) implies that there is a cycle C:

$$v_0 \xrightarrow{e_1} v_1 \cdots \xrightarrow{e_{2k}} v_{2k} \cdots \xrightarrow{e_{2l}} v_{2l} = v_0,$$

$2k \leq 2l \leq n$ and the third vertices of e_1 and e_{2k} are the same vertex w. Let $S_0 = \{e_1, e_3, \cdots, e_{2l-1}\}$ and $S_1 = \{e_2, e_4, \cdots, e_{2k}, \cdots, e_{2l}\}$. Note that $|S_0| = |S_1| \leq n$. We claim that Φ cannot represent any set S such that

$$S_0 \subseteq S \subseteq \bar{S_1}.$$

In particular, Φ cannot represent the set S_0. Assume Φ represents such an S. Since $e_1 \in S$, either v_0 or v_1 must be assigned a 1. Assume that v_1 is assigned a 1. Then, since $e_2 \notin S$, v_2 must be assigned a 0. Again, since $e_3 \in S$, v_3 must be assigned a 1 and so on. All locations in $R := \{v_{2r} | 0 \leq r \leq l\}$ must be assigned a 0 and all locations in $Q := \{v_{2r+1} | 0 \leq r \leq l-1\}$ must be assigned a 1. Else if v_0 is assigned a 1, then all locations in R must be assigned a 1 and all locations in Q must be assigned a 0. Now, $\mathsf{Maj}(x(e_1), y(e_1), z(e_1)) = \mathsf{Maj}(A[v_0], A[v_1], A[w]) = \mathsf{Maj}(0, 1, A[w]) = A[w]$. Since $e_1 \in S$, w must be assigned a 0. Similarly, $\mathsf{Maj}(x(e_{2k}), y(e_{2k}), z(e_{2k})) = \mathsf{Maj}(A[v_{2k-1}], A[v_{2k}], A[w]) = \mathsf{Maj}(0, 1, A[w]) = A[w]$. Since $e_{2k} \notin S$, w must be assigned a 1. A contradiction. ◀

In order to prove Lemma 14 we will make use of the following proposition, which is a consequence of a theorem of Alon, Hoory and Linial [2] (see also Ajesh Babu and Radhakrishnan [3]).

▶ **Proposition 15.** *Fix a graph G such that the average degree $d \geq 2$. Then,*

$$(d-1)^k > |V(G)| \implies \exists \text{ a cycle } C \subseteq E(G), |C| \leq 2k.$$

Proof of Lemma 14. Fix an $\left(m, n, \left\lfloor \frac{1}{6} m^{1-\frac{1}{\lfloor \frac{n}{2} \rfloor + 1}} \right\rfloor, 3\right)$-scheme Φ that uses MAJORITY as the query function. Note that $s := \left\lfloor \frac{1}{6} m^{1-\frac{1}{\lfloor \frac{n}{2} \rfloor + 1}} \right\rfloor$ implies

$$m \geq s^{1+\frac{1}{\lfloor \frac{n}{2} \rfloor}} + 4s + 1. \tag{$\star$}$$

For Φ we will come up with a model-graph which is forced, that is, one of (P1), (P2) or (P3) holds. We will start with an initial model-graph G for Φ. We will observe that the average degree of G is high and invoke Proposition 15 to find a small cycle C. If $|C|$ is odd, we will bin it in ODD, delete C and repeat. If $|C|$ is even and all the third vertices of the labels of edges in C are distinct, we will bin C in EVEN, delete the edges of C and repeat; otherwise, we will either discover that property (P3) holds or we will modify our model-graph and find an odd cycle in it and bin it in ODD, delete it and repeat. The moment the sum of the lengths of the deleted cycles exceeds $2s$, we know either the sum of the lengths of odd or even cycles exceeds s and two odd cycles intersect or even cycles with distinct third vertices meet, which means either (P1) or (P2) holds. Formally, the procedure can be described as below. We will maintain the following invariant. EVEN will contain edge-disjoint cycles of length even and at most n each and the third vertices of the labels in such a cycle will be all distinct. ODD will contain edge-disjoint cycles of length odd and at most n. Furthermore $([s], E(G) \cup \text{EVEN} \cup ODD)$ will always be a model-graph for Φ.

Step 0: Initialization. EVEN $= \emptyset$. ODD $= \emptyset$. $G = ([s], \{x(u) \xleftrightarrow{u} y(u) | u \in [m]\})$. Observe G is a model-graph for Φ.

Step 1. If $\sum_{C \in \text{EVEN} \cup \text{ODD}} |C| > 2s$, END (this ensures that either (P1) or (P2) holds). Else, using Proposition 15 fix a cycle $C \subseteq E(G)$ such that $|C| \leq n$.

Step 2. If $|C|$ is odd, ODD $\leftarrow$ ODD $\cup \{C\}$ and $E(G) \leftarrow E(G) \setminus C$ and GOTO Step 1.

Step 3. If $|C|$ is even and all the third vertices of the labels of edges in C are distinct, EVEN $\leftarrow$ EVEN $\cup \{C\}$ and $E(G) \leftarrow E(G) \setminus C$ and GOTO Step 1.

Step 4. If $|C|$ is even and the third vertices of the labels of two edges in C which have an even number of edges between them while traversing the edges of C in order, then END (Note this means that (P3) holds).

Step 5. If $|C|$ is even and the third vertices of the labels of two edges in C have an odd number of edges between them (while traversing the edges of C in order), then represent C as

$$C : v_0 \xrightarrow{e_1} v_1 \cdots v_{2k} \xrightarrow{e_{2k+1}} v_{2k+1} \cdots \xrightarrow{e_{2l}} v_{2l} = v_0,$$

such that the third vertices of e_1 and e_{2k+1} are the same vertex w. We modify the model-graph G by changing the endpoints of the edges appearing with labels e_1, e_{2k+1} from $\{v_0, v_1\}$, $\{v_{2k}, v_{2k+1}\}$ to $\{v_1, w\}$, $\{v_{2k}, w\}$ respectively, thus obtaining a shorter odd cycle C' in G:

$$E(G) \leftarrow (E(G) \setminus \{v_0 \xleftrightarrow{e_1} v_1, v_{2k} \xleftrightarrow{e_{2k+1}} v_{2k+1}\}) \cup \{v_1 \xleftrightarrow{e_1} w, v_{2k} \xleftrightarrow{e_{2k+1}} w\}$$

(Observe: G with $E(G) \cup \{e | e \in \text{ODD} \cup \text{EVEN}\}$ continues to be a model-graph for Φ).

$$C' \subseteq E(G) : w \xrightarrow{e_1} v_1 \xrightarrow{e_2} v_2 \cdots v_{2k} \xrightarrow{e_{2k+1}} w$$

is an odd length cycle of length at most n in G.
ODD $\leftarrow$ ODD $\cup \{C'\}$. $E(G) \leftarrow E(G) \setminus C'$. GOTO Step 1.

In Step 1, if $|E(G)| \leq 2s$, then the average degree d is at least $\frac{m-2s}{s} > s^{\frac{1}{\lfloor \frac{n}{2} \rfloor}} + 2$ (from $\star$) and $(d-1)^{\lfloor \frac{n}{2} \rfloor} > s$ which implies from Proposition 15 that there is a cycle of length at most n.

We claim that the procedure terminates only by encountering an END statement in Step 1 or in Step 4. Observe that once the procedure finds a cycle in Step 1, then exactly one of the four if conditions in Steps 2-5 holds. If the procedure does not encounter an END statement in Step 4, then the procedure moves to Step 1 again as each of the Steps 2, 3 and 5 end in a 'GOTO Step 1' statement.

If the procedure encounters the END statement in Step 4, then (P3) holds. If the procedure encounters the END statement in Step 1, then from the pigeonhole principle, either $\sum_{C \in \text{ODD}} |C| > s$ or $\sum_{C \in \text{EVEN}} |C| > s$. In the first case, (P1) holds. In the second case, since each edge in a cycle in EVEN has a distinct third vertex, two cycles in EVEN meet.

Finally, we observe that the procedure terminates. If the procedure does not terminate in Step 4, then the procedure repeatedly finds edge disjoint cycles and deletes them. If the number of edges in the deleted cycle exceeds $2s$, then the procedure will terminate when it encounters the END statement in Step 1. ◀

3.3 Degree argument

In this section we provide lower bound proofs for the query functions $(x, y, z) \mapsto (x \wedge y) \oplus z$ and $(x, y, z) \mapsto 1$ iff $x + y + z = 1$.

3.3.1 $(x, y, z) \mapsto (x \wedge y) \oplus z$

Let Φ be a scheme with $(x, y, z) \mapsto (x \wedge y) \oplus z$ as the query function. The memory consists of three distinct bit arrays: $A[1, \cdots, s], B[1, \cdots, s]$ and $C[1, \cdots, s]$. For any element $u \in [m]$, the scheme probes three locations $x(u) \in A$, $y(u) \in B$ and $z(u) \in C$ to determine if u is in the set or not. We treat each location as a boolean variable. Given any set $S \subseteq [m]$ of size at most n the assignment $\sigma(S) \in \{0,1\}^{3s}$ to A, B and C is such that for all elements $u \in [m]$, $(x(u) \wedge y(u)) \oplus z(u)$ is 1 iff $u \in S$.

We first prove that $s = \Omega(\sqrt{mn})$ by specializing the lower bound proof in [13] to our case.

▶ **Definition 16** (Field $\mathbb{F}_2$, vector space V, polynomials P_S)**.** Let $\mathbb{F}_2$ denote the field $\{0,1\}$ with mod 2 arithmetic. The query function $(x, y, z) \mapsto (x \wedge y) \oplus z$ is same as $(x, y, z) \mapsto xy + z$ (over $\mathbb{F}_2$).

Let V be the vector space over the field $\mathbb{F}_2$ of all multilinear polynomials of total degree at most $2n$ in the $3s$ variables: $A[1], \cdots, A[s]$, $B[1], \cdots, B[s]$, $C[1], \cdots, C[s]$ with coefficients coming from $\mathbb{F}_2$.

For each set $S \subseteq [m]$, we define the polynomial P_S in $3s$ variables and coefficients coming from the field $\mathbb{F}_2$ as follows:

$$P_S = \prod_{u \in S} (x(u)y(u) + z(u)).$$

We make P_S multilinear by reducing the exponents of each variable using the identity $x^2 = x$ for each variable x. This identity holds since we will be considering only 0-1 assignment to the variables.

To prove the theorem for $(x, y, z) \mapsto (x \wedge y) \oplus z$, we use the following two lemmas.

▶ **Lemma 17.** *The set of $\binom{m}{n}$ multilinear polynomials $\{P_S : |S| = n\}$ is linearly independent in the vector space V.*

▶ **Lemma 18.** *V has a spanning set of size at most $\binom{3s+2n}{2n}$.*

Using these two lemmas, we first prove the theorem and provide the proofs of the lemmas later.

Proof. Now, since the size of a linearly independent set is at most the size of a spanning set, using Lemmas 17 and 18, we have

$$\begin{aligned} \binom{m}{n} &\leq \binom{3s+2n}{2n} \\ \implies \left(\frac{m}{n}\right)^n &\leq \left(\frac{e(3s+2n)}{2n}\right)^{2n} \\ \implies 3s &\geq \frac{2}{e}\sqrt{n}(\sqrt{m} - e\sqrt{n}) \\ \implies 3s &\geq \frac{18}{10e}\sqrt{mn} \quad (\text{when } n \leq \frac{m}{900}\{\implies e\sqrt{n} \leq \frac{1}{10}\sqrt{m}\}). \end{aligned}$$

When $n \geq \frac{m}{900}$, the fact that the assignments to the memory for storing different sets of size $\lceil \frac{m}{900} \rceil$ are different implies that the space required is at least $\lg \binom{m}{\lceil \frac{m}{900} \rceil} \geq \Omega(m) \geq \Omega(\sqrt{mn})$. ◀

Now, we prove the two lemmas.

Proof of Lemma 17. First observe that any S of size n, the polynomial P_S has n factors of degree 2 each. Hence, the degree of P_S is at most $2n$.

For sets $S, S' \subseteq [m]$ of size n each, the evaluation of the polynomial P_S on the assignment $\sigma(S')$ is

$$P_S(\sigma(S')) = \begin{cases} 0 & \text{if } S \neq S' \\ 1 & \text{if } S = S'. \end{cases}$$

Since $S \neq S'$ and $|S| = |S'| = n \geq 1$, there exists $u \in S$ such that $u \notin S'$ and thus under the assignment $\sigma(S')$, the factor $x(u)y(u) + z(u)$ in $P_S(\sigma(S'))$ evaluates to 0. While, when $S = S'$, for each $u \in S$ the factor $x(u)y(u) + z(u)$ in $P_S(\sigma(S'))$ evaluates to 1.

In particular, this proves that $\{P_S : |S| = n\}$ has size $\binom{m}{n}$. Further we use this observation below to prove the lemma.

Let $\sum_{S:|S|=n} \alpha_S P_S = 0$ where each $\alpha_S \in \mathbb{F}_2$. To show that the P_S's are linearly independent, we need to show that each α_S is 0. Consider an arbitrary set S' of size n, consider the assignment $\sigma(S')$ to the variables in the above identity.

$$\begin{aligned} 0 &= \sum_{S:|S|=n} \alpha_S P_S(\sigma(S')) \\ &= \alpha_{S'} P_{S'}(\sigma(S')) \quad + \sum_{S:S\neq S', |S|=n} \alpha_S P_S(\sigma(S')) \\ &= \alpha_{S'} P_{S'}(\sigma(S')) \quad (\text{since, } P_S(\sigma(S')) = 0 \text{ for each } S \neq S') \\ &= \alpha_{S'} \quad (\text{since, } P_{S'}(\sigma(S')) = 1). \end{aligned}$$ ◀

Proof of Lemma 18. The monomials of total degree at most $2n$ form a spanning set; each polynomial in V can be written as a linear combination of these monomials. Thus, the size of this spanning set is

$$\sum_{k=0}^{2n} \binom{3s}{k} \leq \binom{3s+2n}{2n},$$

where the last inequality follows from the fact that $T \mapsto T \cap [3s]$ is an onto map from $\binom{[3s+2n]}{2n}$ to $\binom{[3s]}{\leq 2n}$. ◀

3.3.2 $(x, y, z) \mapsto 1$ iff $x + y + z = 1$

The lower bound proof for $(x, y, z) \mapsto 1$ iff $x + y + z = 1$ is similar to the lower bound proof for $(x, y, z) \mapsto (x \wedge y) \oplus z$. The only difference here is that instead of looking at the query function over the field $\mathbb{F}_2$, we consider the query function over the field $\mathbb{F}_3$ (the set of three elements $\{0, 1, 2\}$ with mod 3 arithmetic). Over the field $\mathbb{F}_3$, the query function $(x, y, z) = 1$ iff $x + y + z = 1$ is same as $(x, y, z) \mapsto x + y + z + xy + yz + zx$ (a degree 2 polynomial). Accordingly, the multilinear polynomial corresponding to a set S of size n is defined to be

$$P_S = \prod_{u \in S} (x(u) + y(u) + z(u) + x(u)y(u) + y(u)z(u) + z(u)x(u)).$$

where we reduce the exponents using the identity $x^2 = x$ for each variable x (this identity holds as we consider only 0-1 assignments). Notice that P_S has degree at most $2n$ and the rest of the proof is same as before.

The proofs for functions from the remaining classes, the proofs of Theorems 4(c) and 3 are available in the full version of the paper [6].

References

1 Noga Alon and Uriel Feige. On the power of two, three and four probes. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*, pages 346–354, 2009. URL: `http://dl.acm.org/citation.cfm?id=1496770.1496809`.

2 Noga Alon, Shlomo Hoory, and Nathan Linial. The Moore bound for irregular graphs. *Graphs and Combinatorics*, 18(1):53–57, 2002. `doi:10.1007/s003730200002`.

3 Ajesh Babu and Jaikumar Radhakrishnan. An entropy based proof of the Moore bound for irregular graphs. *CoRR*, abs/1011.1058, 2010. URL: `http://arxiv.org/abs/1011.1058`.

4 Harry Buhrman, Peter Bro Miltersen, Jaikumar Radhakrishnan, and Srinivasan Venkatesh. Are bitvectors optimal? *SIAM J. Comput.*, 31(6):1723–1744, 2002. `doi:10.1137/S0097539702405292`.

5 Mohit Garg and Jaikumar Radhakrishnan. Set membership with a few bit probes. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 776–784, 2015. `doi:10.1137/1.9781611973730.53`.

6 Mohit Garg and Jaikumar Radhakrishnan. Set membership with non-adaptive bit probes. *CoRR*, abs/1612.09388, 2016. URL: `http://arxiv.org/abs/1612.09388`.

7 Michael T. Goodrich and Michael Mitzenmacher. Invertible Bloom lookup tables. In *49th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2011, Allerton Park & Retreat Center, Monticello, IL, USA, 28-30 September, 2011*, pages 792–799, 2011. `doi:10.1109/Allerton.2011.6120248`.

8 Moshe Lewenstein, J. Ian Munro, Patrick K. Nicholson, and Venkatesh Raman. Improved explicit data structures in the bitprobe model. In *Algorithms – ESA 2014 – 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, pages 630–641, 2014. `doi:10.1007/978-3-662-44777-2_52`.

9 Michael Luby, Michael Mitzenmacher, Mohammad Amin Shokrollahi, and Daniel A. Spielman. Efficient erasure correcting codes. *IEEE Trans. Information Theory*, 47(2):569–584, 2001. `doi:10.1109/18.910575`.

10 Ali Makhdoumi, Shao-Lun Huang, Muriel Médard, and Yury Polyanskiy. On locally decodable source coding. In *2015 IEEE International Conference on Communications, ICC 2015, London, United Kingdom, June 8-12, 2015*, pages 4394–4399, 2015. `doi:10.1109/ICC.2015.7249014`.

11 Marvin Minsky and Seymour Papert. Perceptrons. MIT press, Cambridge, MA, 1969.

12 Jaikumar Radhakrishnan, Venkatesh Raman, and S. Srinivasa Rao. Explicit deterministic constructions for membership in the bitprobe model. In *Algorithms – ESA 2001, 9th Annual European Symposium, Aarhus, Denmark, August 28-31, 2001, Proceedings*, pages 290–299, 2001. `doi:10.1007/3-540-44676-1_24`.

13 Jaikumar Radhakrishnan, Pranab Sen, and Srinivasan Venkatesh. The quantum complexity of set membership. *Algorithmica*, 34(4):462–479, 2002. `doi:10.1007/s00453-002-0979-0`.

14 Jaikumar Radhakrishnan, Smit Shah, and Saswata Shannigrahi. Data structures for storing small sets in the bitprobe model. In Mark de Berg and Ulrich Meyer, editors, *Algorithms – ESA 2010, 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part II*, volume 6347 of *Lecture Notes in Computer Science*, pages 159–170. Springer, 2010. `doi:10.1007/978-3-642-15781-3_14`.

15 Emanuele Viola. Bit-probe lower bounds for succinct data structures. *SIAM J. Comput.*, 41(6):1593–1604, 2012. `doi:10.1137/090766619`.

16 Wikiversity. The 22 becs, 3-ary boolean functions – wikiversity, 2016. [Online; accessed 7-August-2016]. URL: `https://en.wikiversity.org/w/index.php?title=3-ary_Boolean_functions&oldid=1587287`.