Electronic Colloquium on Computational Complexity, Report No. 87 (2010)

A lower bound for dynamic approximate membership data structures

Shachar Lovett *
The Weizmann Institute of Science
shachar.lovett@weizmann.ac.il

Ely Porat †
Bar-Ilan University
porately@cs.biu.ac.il

May 17, 2010

Abstract

An approximate membership data structure is a randomized data structure for representing a set which supports membership queries. It allows for a small false positive error rate but has no false negative errors. Such data structures were first introduced by Bloom in the 1970's, and have since had numerous applications, mainly in distributed systems, database systems, and networks.

The algorithm of Bloom is quite effective: it can store a set S of size n by using only $\approx 1.44n \log_2(1/\epsilon)$ bits while having false positive error ϵ. This is within a constant factor of the entropy lower bound of $n \log_2(1/\epsilon)$ for storing such sets. Closing this gap is an important open problem, as Bloom filters are widely used is situations were storage is at a premium.

Bloom filters have another property: they are dynamic. That is, they support the iterative insertions of up to n elements. In fact, if one removes this requirement, there exist static data structures which receive the entire set at once and can almost achieve the entropy lower bound; they require only $n \log_2(1/\epsilon)(1 + o(1))$ bits.

Our main result is a new lower bound for the memory requirements of any dynamic approximate membership data structure. We show that for any constant $\epsilon > 0$, any such data structure which achieves false positive error rate of ϵ must use at least $C(\epsilon) \cdot n \log_2(1/\epsilon)$ memory bits, where $C(\epsilon) > 1$ depends only on ϵ. This shows that the entropy lower bound cannot be achieved by dynamic data structures for any constant error rate.

1 Introduction

Suppose we want to build a data structure, that given a set of elements $S = \{x_1, \ldots, x_n\}$ and an additional element y, will be able to distinguish whether $y \in S$ or not. The *approximate membership problem* consists of storing a data structure that supports membership queries in the following manner: For a query on $y \in S$ it is always reported that $y \in S$. For a query on $y \notin S$ it is reported with probability at least $1 - \epsilon$ that $y \notin S$, and with probability at most ϵ that $y \in S$. That is, an approximate membership data structure has no false negative errors, and allows false positive errors with probability at most ϵ.

The approximate membership problem has attracted significant interest in recent years, since it is a common building block for various applications, mainly in distributed systems, database systems and networks (see [BM03] for a survey). Approximate membership data structures are

*Research supported by the Israel Science Foundation grant 1300/05 and by the ERC starting grant 239985.
†Research supported by the Israel Science Foundation and the United States-Israel Binational Science Foundation.

ISSN 1433-8092

often used in practice when storage is at a premium, while a small probability for false positive errors can be tolerated. The false positive error rate which can be tolerated is often relatively large, say, in the range $1\% - 10\%$.

The study of approximate membership was initiated by Bloom [Blo70] who described the *Bloom filter* data structure, which provides a simple and elegant solution for the problem which is near-optimal. Bloom showed that a space usage of $n \log_2(1/\epsilon) \log_2 e$ bits suffices for a false positive error probability of ϵ. This is quite close to the entropy lower bound. Carter et. al [CFG+78] showed that $n \log_2(1/\epsilon)$ bits are required when the universe set $\mathbf{U}$ is large $|\mathbf{U}| \gg n$ (see also [DP08] for details). Thus Bloom filters have a space usage within a factor $\log_2 e \approx 1.44$ of the lower bound. As Bloom filters are widely used in practice, mainly in situations when storage is scarce, this factor of 1.44 is not negligible. The main object of study of this paper is whether this factor can be eliminated. I.e., we study whether there exist data structures for approximate membership which achieve the entropy lower bound.

An important feature of Bloom filters is that they are *dynamic*. That is, the elements $x_1, \ldots, x_n$ can be inserted one at a time, while maintaining the succinct representation of the data structure. If, on the other hand, one limits itself to *static* data structures, which are given the entire set $S = \{x_1, \ldots, x_n\}$ at once, and are allowed to preprocess it before creating the succinct data structure, then the entropy lower bound can be nearly achieved. Dietzfelbinger and Pagh [DP08] and Porat [Por09] gave data structures for the static approximate membership problem using only $n \log_2(1/\epsilon)(1 + o(1))$ bits.

The main result of this paper is that dynamic data structures for approximate membership *cannot* achieve the entropy lower bound.

Theorem 1. *Let $|\mathbf{U}|$ be a universe set. Consider any randomized data structure which allows for the dynamic insertion of up to n elements (where $n \ll |\mathbf{U}|$), has false positive error at most ϵ (where $\epsilon > 0$ is a constant), and which allows no false negative errors. Then for large enough n, any such data structure must use at least $C(\epsilon) n \log_2(1/\epsilon)$ memory bits, where $C(\epsilon) > 1$ is a constant depending only on ϵ. In particular, for $\epsilon = 1/2$ we get $C(1/2) \geq 1.1$.*

We note that the requirement that the false negative error is constant cannot be eliminated. In fact, for every $\epsilon = o(1)$ there is a simple dynamic approximate membership data structure which requires only $n \log_2(1/\epsilon)(1 + o(1))$ bits: pick a (good enough) hash function $h : \mathbf{U} \to [n/\epsilon]$, and at each step maintain the set $\{h(x_1), h(x_2), \ldots, h(x_n)\}$. The space requirements of this algorithm are $\log_2 \binom{n/\epsilon}{n} = n \log_2(1/\epsilon) + O(n)$, which is $n \log_2(1/\epsilon)(1 + o(1))$ for any $\epsilon = o(1)$. The data structure we just described is not efficient; efficient versions are achieved implicitly in the work of Matias and Porat [MP07], and explicitly in works of Pagh, Pagh and Rao [PPR05] (which is based on a work of Rajeev and Rao [RR03]) and in a work of Arbitman, Naor and Segev [ANS10].

1.1 Proof overview

The proof of the lower bound is conducted in two steps: we first transform the problem to a graph-theoretic problem, and then we prove results on this graph-theoretic problem.

The graph-theoretic problem Assume there exists a dynamic approximate membership data structure, which allows insertion of up to n elements from a universe set $\mathbf{U}$, has false positive error of at most ϵ, and which requires M memory bits. Consider first for simplicity a *deterministic* data structure. We model such a data structure by a labelled layered graph, which captures all possible insertions of up to n elements.

The graph G has $n+1$ layers $V_0 \cup V_1 \cup \ldots \cup V_n$, where each vertex in V_i corresponds to a possible state of the data structure after insertions of i elements. In particular $V_0 = \{v_0\}$ and $|V_1|, \ldots, |V_n| \leq 2^M$. The edges connect vertices in adjacent layers, and are labelled by elements $x \in \mathbf{U}$. Given a vertex $v \in V_i$ and an element $x \in \mathbf{U}$, there is an outgoing edge $v \to u$ which is labelled with x, where $u \in V_{i+1}$ corresponds to the state reached after inserting x when the state of the data structure was v. Thus, any sequence $w = x_1, \ldots, x_i \in \mathbf{U}^i$ defines a path from $v_0 \in V_0$ to some vertex $v(w) \in V_i$.

For a vertex v define $L(v)$ to be the set of all labels in paths between v_0 and v. For a sequence $w \in \mathbf{U}^i$ define $L(w) = L(v(w))$ to be all labels in paths reaching $v(w)$. We prove that since G is based on an approximate membership data structure with error ϵ, then for many vertices $v \in V_n$, if we consider all labels in all paths reaching v, we only cover approximately an ϵ-fraction of $\mathbf{U}$. Formally, we show that if $w \in \mathbf{U}^n$ is chosen uniformly at random, then $\mathbb{E}[|L(w)|] \leq \epsilon|\mathbf{U}|(1+o(1))$. We will then use this property to infer a lower bound on the number of vertices 2^M in the layers of G, which will give a lower bound on the memory requirements of the data structures.

In the case of *randomized* data structures, we prove such a graph still exists for some fixing of the internal randomness of the data structure, hence giving the same lower bounds also for randomized data structures.

Lower bound on the layers sizes Let G be the labelled layered graph we constructed. Let $1 \leq k \leq n$ be some intermediate layer. We will in fact prove the following lower bound

$$\max(|V_k|, |V_n|) \geq (1/\epsilon)^{C(\epsilon)n}.$$

Pick $w = x_1, \ldots, x_n \in \mathbf{U}^n$ uniformly at random, and partition w to the first k elements $w' = x_1, \ldots, x_k$ and the last $n-k$ elements $w'' = x_{k+1}, \ldots, x_n$. Consider inserting the elements in w as a two-step process: first insert w', reaching an intermediate vertex $v(w') \in V_k$, and then insert w'', reaching a final vertex $v(w'w'') \in V_n$. We get that with good probability we the following two events occur simultaneously:

$$|L(w'w'')| \leq \alpha|\mathbf{U}| \tag{1}$$

$$|L(w')| \geq \beta|\mathbf{U}| \tag{2}$$

where $\alpha \approx \epsilon$ and $\beta \approx 2^{-M/k}$.

We will prove the lower bound by a covering argument, based on the above properties. We first sketch a simple covering argument, which fails at giving a lower bound better than the entropy lower bound. We then show a more complex covering argument which give a non-trivial lower bound on M.

We first consider the simple covering argument. Fix some $v' = v(w')$ and $v'' \in V_n$. If w'' is such that $v'' = v(w'w'')$ then we must have all elements in w'' appear in $L(w'w'')$. However, since $|L(w'w'')| \approx \epsilon|\mathbf{U}|$, the number of possibilities for w'' is at most $(\epsilon|\mathbf{U}|)^{n-k}$. Thus, since the total number of $w'' \in \mathbf{U}^{n-k}$ is $|\mathbf{U}|^{n-k}$, there must be at least $(1/\epsilon)^{n-k}$ different vertices in V_n which can be reached from v'. This yields the bound $|V_n| \geq (1/\epsilon)^{n-k}$, which is optimized by taking $k = 0$ and gives $M \geq n \log_2(1/\epsilon)$.

We now show how to obtain an improved covering argument. Say a sequence $w'' \in \mathbf{U}^{n-k}$ is *good* for w' if w'' intersects $L(w')$ in about the right number of times, that is

$$|w'' \cap L(w')| \approx (n-k)\frac{L(w')}{|\mathbf{U}|}. \tag{3}$$

Assume w'' is good for w' such that $v'' = v(w'w'')$. Let $\beta(w') = \frac{L(w')}{|\mathbf{U}|}$. The number of such w'' is bounded by

$$\approx \binom{n-k}{\beta(w')(n-k)} |L(w')|^{\beta(w')(n-k)} |L(v'') \setminus L(w')|^{(1-\beta(w'))(n-k)}.$$

We show that events (1), (2) and (3) all occur simultaneously with a relative large probability. We infer that for a large fraction of w', there must be many distinct $v(w'w'')$ where w'' is good for w',

$$|\{v(w'w'') : w'' \text{ is good for } w'\}| \geq \left(\frac{1-\beta(w')}{\alpha-\beta(w')}\right)^{(1-\beta(w'))(n-k)}.$$

Combining this with the simple bound, that the number of $w' \in \mathbf{U}^k$ which can reach some vertex in a path to $v'' \in V_n$, is bounded by $|L(v'')|^k \approx (\alpha|\mathbf{U}|)^k$, we deduce the following inequality. Set $c = \frac{k}{n}$ and $\eta = \frac{M}{n \log_2(1/\epsilon)}$. We get

$$(1/\epsilon)^\eta \epsilon^c \geq \left(\frac{1-\epsilon^{\eta/c}}{\epsilon-\epsilon^{\eta/c}}\right)^{(1-\epsilon^{\eta/c})(1-c)}. \tag{4}$$

This is a non-trivial inequality relating the different parameters ϵ, c and η. Note it should hold for any value of $0 < c < 1$. In the final step we study inequality (4), and prove that for every constant $\epsilon > 0$ we can choose some value for c such that we must have $\eta > C(\epsilon) > 1$ for the inequality to hold.

Paper organization We formally define approximate membership data structures in Section 2. We prove Theorem 1 in Section 3.

2 Preliminaries

Let $\mathbf{U}$ be a universe set. An *approximate membership* data structure is a space-efficient randomized data structure that represents a subset $S \subset \mathbf{U}$ of size $|S| \leq n$ and supports queries whether $x \in S$ for elements $x \in \mathbf{U}$, with the following guarantees:

- No false negatives: if $x \in S$, the query will always return *true*.
- Few false positives: if $x \notin S$, the query will return *false* with probability at least $1 - \epsilon$, and will return *true* with probability at most ϵ (probabilities are over the internal randomness of the data structure).

The main goal of this paper is to study the tradeoff between the maximal set size n, the false positive error parameter ϵ and the memory requirements of the data structure. We will assume throughout the paper that the subset S is a small fraction of the universe, i.e. that $n \ll |\mathbf{U}|$.

We now define *dynamic* vs. *static* approximate membership data structures.

Definition 1 (Dynamic approximate membership data structure)**.** *A dynamic approximate membership data structure is composed of two algorithms: an insertion algorithm and a query algorithm.*

- *The insertion algorithm $\mathcal{I}$ is a randomized algorithm, which allows for the insertion of up to n elements sequentially. The algorithm maintains a succinct representation R of the set of elements inserted so far, and for each new element $x \in \mathbf{U}$ updates $R \leftarrow \mathcal{I}(R, x)$.*

- *The query algorithm $\mathcal{Q}$ receives as inputs the succinct representation R of S and an element $x \in \mathbf{U}$, and outputs an estimate $\mathcal{Q}(R, x) \in \{true, false\}$ whether $x \in S$.*

The memory requirements of a dynamic approximate membership data structure is the maximal number of bits required to represent R throughout the insertion phase. We denote by $M_D(n, \epsilon)$ the minimal memory required by a dynamic approximate membership data structure which stores up to n elements and has false positive errors with probability at most ϵ.

Definition 2 (Static approximate membership data structure)**.** *A static approximate membership data structure is composed of two algorithms: a preprocessing algorithm and a query algorithm.*

- *The preprocessing algorithm $\mathcal{P}$ is a randomized algorithm, which receives as input a subset $S \subset \mathbf{U}$ of size at most n, and outputs a succinct representation $R = \mathcal{P}(S)$ of S.*
- *The query algorithm $\mathcal{Q}$ receives as inputs the succinct representation R of S and an element $x \in \mathbf{U}$, and outputs an estimate $\mathcal{Q}(R, x) \in \{true, false\}$ whether $x \in S$.*

The memory requirements of a static approximate membership data structure is the number of bits required to represent $P(S)$. We denote by $M_S(n, \epsilon)$ the minimal memory required by a static Bloom filter which stores up to n elements and has false positive error with probability at most ϵ.

For the convenience of the reader we recap the known properties of the memory requirements of dynamic and static approximate membership data structure. These include the entropy lower bound of Carter et. al [CFG+78]; Bloom filters [Blo70]; and efficient static data structures of Dietzfelbinger and Pagh [DP08] and of Porat [Por09].

Fact 2. *For any constant $\epsilon > 0$ we have*

- $M_S(n, \epsilon) = (1 + o(1)) \cdot n \log_2(1/\epsilon)$.
- $(1 - o(1)) \cdot n \log_2(1/\epsilon) \leq M_D(n, \epsilon) \leq \log_2 e \cdot n \log_2(1/\epsilon) \approx 1.44 \cdot n \log_2(1/\epsilon)$.

Our main result is an improved lower bound on $M_D(n, \epsilon)$,

$$M_D(n, \epsilon) \geq C(\epsilon) \cdot n \log_2(1/\epsilon),$$

where $C(\epsilon) > 1$ is a constant depending only on ϵ.

3 Proof of the lower bound

We prove Theorem 1 in this section.

3.1 The graph-theoretic problem

Let $(\mathcal{I}, \mathcal{Q})$ be the insertion and query randomized algorithms in an optimal dynamic approximate membership data structure for sets of size n with false positive error of ϵ, which uses $M = M_D(n, \epsilon)$ memory bits. Let r denote the internal randomness used by the algorithms. We denote by $\mathcal{I}^r, \mathcal{Q}^r$ the algorithms given an explicit value r for the internal randomness.

It will be convenient for us to model a dynamic approximate membership data structure by a labeled layered graph. For any fixing of r, define a labeled layered graph G^r as follows. The graph will have $n+1$ layers $V_0 \cup V_1 \cup \ldots \cup V_n$. Each vertex in V_i corresponds to a possible state of the data

structure after insertions of i elements. In particular, $|V_0| = 1$ and $|V_1|, \ldots, |V_n| \leq 2^M$. The edges connect vertices in adjacent layers, and are labeled by elements $x \in \mathbf{U}$. Given a vertex $v \in V_i$ and an element $x \in \mathbf{U}$, there is an outgoing edge $v \to u$ which is labeled with x, where $u = \mathcal{I}^r(v, x)$. Thus, the graph G^r describes all possible iterative insertions of n elements (given the fixing r of the internal randomness), and the collection of graphs $\{G^r\}$ is a complete description of the insertion algorithm.

For ease of notation, we extend the definition of $\mathcal{I}^r$ for sequences of elements. Let $w = x_1, \ldots, x_i \in \mathbf{U}^i$ be a sequence of i elements, and let $v \in V_j$ where $i+j \leq n$. We define $\mathcal{I}^r(v, w) \in V_{i+j}$ to be the vertex reached from v after insertion of $x_1, \ldots, x_i$, i.e.

$$\mathcal{I}^r(v, w) = \mathcal{I}^r(\ldots \mathcal{I}^r(\mathcal{I}^r(v, x_1), x_2) \ldots, x_i).$$

We also shorthand $\mathcal{I}^r(w) = \mathcal{I}^r(v_0, w)$ where $v_0 \in V_0$ is the initial state of the data structure.

For a sequence $w = x_1, \ldots, x_n \in \mathbf{U}^n$, denote by $A^r(w)$ the set of all elements $x \in \mathbf{U}$ which are accepted by $\mathcal{Q}^r$ given the succinct representation $v = \mathcal{I}^r(w)$, i.e.

$$A^r(w) = \{x \in \mathbf{U} : \mathcal{Q}^r(\mathcal{I}^r(w), x) = true\}.$$

We can summarize the properties that $(\mathcal{I}, \mathcal{Q})$, being a dynamic approximate membership data structure, has no false negative erros and has false positive errors with probability at most ϵ by the following claim.

Claim 3. *Let $w = x_1, \ldots, x_n \in \mathbf{U}^n$. Then:*

- *For any setting of r, we have $\{x_1, \ldots, x_n\} \subset A^r(w)$.*
- *Let $y \notin \{x_1, \ldots, x_n\}$. Then $\Pr_r[y \in A^r(w)] \leq \epsilon$.*

Proof. The first claim follows from the assumption that $(\mathcal{I}, \mathcal{Q})$ have no false negative errors. Thus, for any x_i $(i = 1, \ldots, n)$ since $x_i \in \{x_1, \ldots, x_n\}$ we must have that $\Pr_r[\mathcal{Q}^r(\mathcal{I}^r(w), x_i) = true] = 1$. The second claim follows from the assumption that $(\mathcal{I}, \mathcal{Q})$ have false positive errors with probability at most ϵ. Thus, for a random choice of r, $\Pr_r[\mathcal{Q}^r(\mathcal{I}^r(w), y) = true] \leq \epsilon$. □

As a corollary we get that the size of $A^r(w)$ must be small for average r.

Claim 4. *Let $w = x_1, \ldots, x_n \in \mathbf{U}^n$. Then $\mathbb{E}_r[|A^r(w)|] \leq \epsilon|\mathbf{U}| + n$.*

Proof. The proof follows immediately from Claim 3. Let $S = \{x_1, \ldots, x_n\}$. Then

$$\mathbb{E}_r[|A^r(w)|] = \sum_{y \in \mathbf{U}} \Pr_r[y \in A^r(w)] \leq |S| + \sum_{y \in \mathbf{U} \setminus S} \Pr_r[y \in A^r(w)] \leq n + \epsilon|\mathbf{U}|.$$

□

We now fix the randomness for the algorithms. Let $w = x_1, \ldots, x_n \in \mathbf{U}^n$ be uniformly chosen. By Claim 4 we have in particular that

$$\mathbb{E}_r \mathbb{E}_{w \in \mathbf{U}^n}[|A^r(w)|] \leq \epsilon|\mathbf{U}| + n.$$

Thus, there must exist some fixing $r = r^*$ such that

$$\mathbb{E}_{w \in \mathbf{U}^n}[|A^{r^*}(w)|] \leq \epsilon|\mathbf{U}| + n.$$

From now on we fix the internal randomness to r^*, and for ease of notation omit the superscript r^* from $G, A, \mathcal{I}, \mathcal{Q}$. Hence we have

Claim 5. $\mathbb{E}_{w \in \mathbf{U}^n}[|A(w)|] \leq \epsilon|\mathbf{U}| + n$.

3.2 Properties of the graph

We will prove some properties of the layered graph we obtained. These properties will later be used to prove the lower bound.

Let $0 < \delta \ll 1$ be a small parameter to be determined later. We first show that for a relatively large fraction of $w \in \mathbf{U}^n$, the set $A(w)$ is not much larger than the average size of these sets.

Claim 6. *Let $w \in \mathbf{U}^n$ be chosen uniformly. Set $\alpha = \epsilon(1 + \frac{n}{|\mathbf{U}|})(1 + 6\delta) = \epsilon(1 + o(1))$. Then*

$$\Pr_{w \in \mathbf{U}^n}[|A(w)| \leq \alpha|\mathbf{U}|] \geq 3\delta.$$

Proof. By Markov's inequality,

$$\Pr_{w \in \mathbf{U}^n}[|A(w)| \geq \alpha|\mathbf{U}|] \leq \frac{\mathbb{E}_{w \in \mathbf{U}^n}[|A(w)|]}{\alpha|\mathbf{U}|} = \frac{1}{1 + 6\delta} \leq 1 - 3\delta$$

for any $\delta < 1/6$. □

We now make an important definition. Let $w = x_1, \ldots, x_i \in \mathbf{U}^i$ and let $v = \mathcal{I}(w)$. We define $L(w)$ to be the set of labels on any path which reaches v. That is,

$$L(w) = \{y \in \mathbf{U} : \exists w' = x'_1, \ldots, x'_i \in \mathbf{U}^i \text{ such that } \mathcal{I}(w') = \mathcal{I}(w) \text{ and } y \in \{x'_1, \ldots, x'_i\}\}.$$

We now prove two useful properties of the sets $L(w)$.

Claim 7.

1. *Let $w = x_1, \ldots, x_i \in \mathbf{U}^i$ and $w' = x_{i+1}, \ldots, x_j \in \mathbf{U}^{i-j}$ for $i < j$. Let $ww' \in \mathbf{U}^j$ be the concatenation of w and w'. Then $L(w) \subseteq L(ww')$.*
2. *Let $w = x_1, \ldots, x_n \in \mathbf{U}^n$. Then $L(w) \subseteq A(w)$.*

Proof. The first claim follows immediately from the definition of L. If $y \in L(w)$ then there exists $\widetilde{w} = \widetilde{x}_1, \ldots, \widetilde{x}_i \in \mathbf{U}^i$ such that $\mathcal{I}(\widetilde{w}) = \mathcal{I}(w)$ and $y \in \{\widetilde{x}_1, \ldots, \widetilde{x}_i\}$. But then $\mathcal{I}(\widetilde{w}w') = \mathcal{I}(ww')$, hence also $y \in L(ww')$.

The second claim follows since a dynamic approximate membership data structure has no false negative errors. Let $y \in L(w)$, and let $\widetilde{w} = \widetilde{x}_1, \ldots, \widetilde{x}_n \in \mathbf{U}^n$ such that $\mathcal{I}(\widetilde{w}) = \mathcal{I}(w)$ and $y \in \{\widetilde{x}_1, \ldots, \widetilde{x}_n\}$. By Claim 3 we know that $\{\widetilde{x}_1, \ldots, \widetilde{x}_n\} \subset A(w)$. Hence also $y \in A(w)$. □

Let $1 \leq k \leq n$ be a parameter to be fixed later. We show that most sets $L(w)$ for $w \in \mathbf{U}^k$ cannot be too small.

Claim 8. *Let $w = x_1, \ldots, x_k \in \mathbf{U}^k$ be chosen uniformly. Then*

$$\Pr_{w \in \mathbf{U}^k}[|L(w)| \leq \beta|\mathbf{U}|] \leq \delta$$

where $\beta = \delta^{1/k} 2^{-M/k}$.

Proof. The proof is by a simple counting argument. Let $\mathcal{L} = \{L(w) : w \in \mathbf{U}^k,\ |L(w)| \leq \beta|\mathbf{U}|\}$ be the set of all possible $L(w)$ of size at most $\beta|\mathbf{U}|$. The size of $\mathcal{L}$ is at most 2^M as distinct sets in $\mathcal{L}$ match distinct vertices in V_k. For any set $\widetilde{L} \in \mathcal{L}$, we can have $L(w) = \widetilde{L}$ for $w = x_1, \ldots, x_k \in \mathbf{U}^k$ only if $\{x_1, \ldots, x_k\} \subset \widetilde{L}$. Thus, for any fixed $\widetilde{L}$, the number of such sequences is bounded by $(\beta|\mathbf{U}|)^k$. Hence,

$$\Pr_{w \in \mathbf{U}^k}[|L(w)| \leq \beta|\mathbf{U}|] \leq \frac{(\beta|\mathbf{U}|)^k 2^M}{|\mathbf{U}|^k} \leq \delta.$$

□

Let $w' = x_1, \ldots, x_k \in \mathbf{U}^k$ and $w'' = x_{k+1}, \ldots, x_n \in \mathbf{U}^{n-k}$. We denote by $C(w', w'')$ the number of elements in w'' which are in $L(w')$, i.e.

$$C(w', w'') = |\{x_i : k+1 \leq i \leq n,\ x_i \in L(w')\}|.$$

The next claim shows that w.h.p we can assume that $C(w', w'') \approx \frac{|L(w')|}{|\mathbf{U}|}(n-k)$.

Claim 9. *Fix $w' = x_1, \ldots, x_k \in \mathbf{U}^k$. Let $w'' = x_{k+1}, \ldots, x_n \in \mathbf{U}^{n-k}$ be distributed uniformly at random. Then*

$$\Pr_{w'' \in \mathbf{U}^{n-k}}\left[\left|C(w', w'') - \frac{|L(w')|}{|\mathbf{U}|}(n-k)\right| \geq \gamma(n-k)\right] \leq \delta$$

where $\gamma = \sqrt{3\ln(2/\delta)/(n-k)}$.

In order to prove Claim 9 we will apply the Chernoff-Hoeffding bound which we recall below.

Lemma 10 (Chernoff-Hoeffding bound)**.** *Let $X_1, \ldots, X_m \in \{0, 1\}$ be independent random variables such that $\mathbb{E}[X_i] = p$. Then for any $\gamma > 0$*

$$\Pr\left[\left|\frac{1}{m}\sum X_i - p\right| \geq \gamma\right] \leq 2e^{-\frac{\gamma^2}{3}m}.$$

Proof of Claim 9. Set $m = n-k$ and define $X_i = \mathbf{1}_{x_{k+i} \in L(w')}$ for $i = 1, \ldots, n-k$. Then $C(w', w'') = \sum_{i=1}^{n-k} X_i$, we have $\mathbb{E}_{w''}[X_1] = \ldots, = \mathbb{E}_{w''}[X_{n-k}] = \frac{|L(w')|}{|\mathbf{U}|}$ and the Chernoff-Hoeffding bound gives

$$\Pr_{w'' \in \mathbf{U}^{n-k}}\left[\left|C(w', w'') - \frac{|L(w')|}{|\mathbf{U}|}(n-k)\right| \geq \gamma(n-k)\right] \leq 2e^{-\frac{\gamma^2}{3}(n-k)} \leq \delta.$$

□

We conclude Claims 6, 8 and 9 by the following claim, showing that there is a relatively large subset $W \subset \mathbf{U}^n$ for which all three claims hold simultaneously.

Claim 11. *Let $W \subset \mathbf{U}^n$ be defined as follows. For $w \in \mathbf{U}^n$ write $w = w'w''$ where $w' \in \mathbf{U}^k$ and $w'' \in \mathbf{U}^{n-k}$. An element $w \in \mathbf{U}^n$ is in W if all the following conditions hold:*

(i) $|A(w'w'')| \leq \alpha|\mathbf{U}|$.

(ii) $|L(w')| \geq \beta|\mathbf{U}|$.

(iii) $\left|C(w', w'') - \frac{|L(w')|}{|\mathbf{U}|}(n-k)\right| \leq \gamma(n-k)$.

Then $|W| \geq \delta|\mathbf{U}|^n$.

Proof. The proof is an immediate corollary of Claims 6, 8 and 9. For uniformly chosen $w \in \mathbf{U}^n$, condition (i) holds with probability at least 3δ, and conditions (ii) and (iii) each hold with probability at least $1-\delta$. Hence by the union bound all three hold simultaneously with probability at least δ. Hence $|W| \geq \delta|\mathbf{U}|^n$. □

3.3 Inequalities on paths in the graph

We will prove a certain family on inequalities on the graph which relate to paths in the graph. Define X to be the set

$$X = \{(w', A(w'w'')) : w'w'' \in W\}.$$

We will prove lower and upper bounds on $|X|$ which will imply lower bounds on the memory requirement M. We start with a simple upper bound.

Claim 12. $|X| \leq (\alpha|\mathbf{U}|)^k 2^M$.

Proof. Any accepting set $\widetilde{A} \in \{A(w) : w \in W\}$ must have size at most $\alpha|\mathbf{U}|$ by condition (i) of Claim 11. Thus, since all elements of w' must be contained in $\widetilde{A}$, the number of $w' \in \mathbf{U}^k$ such that $(w', \widetilde{A}) \in X$ is at most $|\widetilde{A}|^k \leq (\alpha|\mathbf{U}|)^k$. The number of distinct sets $\widetilde{A}$ is bounded by the number of vertices in V_n, which is at most 2^M. Hence we conclude that $|X| \leq (\alpha|\mathbf{U}|)^k 2^M$. □

For $w' \in \mathbf{U}^k$ define $W(w') \subset \mathbf{U}^{n-k}$ to be the set of continuations of w' to elements in W, i.e.

$$W(w') = \{w'' \in \mathbf{U}^{n-k} : w'w'' \in W\}.$$

The following is an immediate corollary of Claim 11.

Corollary 13. $\mathbb{E}_{w' \in \mathbf{U}^k}[|W(w')|] \geq \delta|\mathbf{U}|^{n-k}$.

For $w' \in \mathbf{U}^k$ define $N(w')$ to be the set of accepting sets

$$N(w') = \{A(w'w'') : w'' \in W(w')\}.$$

Note that $|X| = \sum_{w' \in \mathbf{U}^k} |N(w')|$. We now turn to prove lower bounds for the size of $N(w')$. These will then be used to prove lower bounds on $|X|$.

Lemma 14. *Fix $w' \in \mathbf{U}^k$, and assume that $W(w') = \delta'|\mathbf{U}|^{n-k}$. Then*

$$|N(w')| \geq \delta' \left(\frac{1-\beta}{\alpha-\beta}\right)^{(1-\beta)(n-k)\left(1-\frac{\gamma}{1-\alpha}\right)}.$$

Proof. Denote $|L(w')| = \beta'|\mathbf{U}|$ where $\beta' \geq \beta$ by condition (ii). Let $\widetilde{A} \in N(w')$ be some set. By condition (i) we know that $|\widetilde{A}| \leq \alpha|\mathbf{U}|$. Observe that if $A(w'w'') = \widetilde{A}$ for $w'' = x_{k+1}, \ldots, x_n \in W''$, then we must have $x_{k+1}, \ldots, x_n \in \widetilde{A}$. Moreover, by condition (iii) we must have that the number of elements of w'' which intersect $L(w')$ must be $\approx \beta'(n-k)$. Let m denote a possible number of elements of w'' which occur in $L(w')$. The number of sequences $w'' \in \mathbf{U}^{n-k}$ which contain exactly m elements in $L(w')$ and $n-k-m$ elements in $\widetilde{A} \setminus L(w')$ is given by

$$\binom{n-k}{m}|L(w')|^m(|\widetilde{A}| - |L(w')|)^{n-k-m} \leq \binom{n-k}{m}(\beta')^m(\alpha-\beta')^{n-k-m}|\mathbf{U}|^{n-k}.$$

Thus, the total number of $w'' \in W(w')$ for which $A(w'w'') = \widetilde{A}$ is bounded by

$$|\{w'' \in W(w') : A(w'w'') = \widetilde{A}\}| \leq \sum_{m=(\beta'-\gamma)(n-k)}^{(\beta'+\gamma)(n-k)} \binom{n-k}{m}(\beta')^m(\alpha-\beta')^{n-k-m}|\mathbf{U}|^{n-k}. \quad (5)$$

On the other hand, we have that

$$|W(w')| = \delta'|\mathbf{U}|^{n-k} \geq \delta' \sum_{m=(\beta'-\gamma)(n-k)}^{(\beta'+\gamma)(n-k)} \binom{n-k}{m} (\beta')^m (1-\beta')^{n-k-m} |\mathbf{U}|^{n-k}. \tag{6}$$

Thus, the number of distinct sets $\widetilde{A} \in N(w')$ can be lower bounded by

$$\begin{aligned} |N(w')| &\geq \frac{|W(w')|}{\max_{\widetilde{A} \in N(w')} |\{w'' \in N(W') : A(w'w'') = \widetilde{A}\}|} \\ &\geq \delta' \frac{\sum_{m=(\beta'-\gamma)(n-k)}^{(\beta'+\gamma)(n-k)} \binom{n-k}{m} (\beta')^m (1-\beta')^{n-k-m}}{\sum_{m=(\beta'-\gamma)(n-k)}^{(\beta'+\gamma)(n-k)} \binom{n-k}{m} (\beta')^m (\alpha-\beta')^{n-k-m}}. \end{aligned}$$

As always for any numbers $a_1, \ldots, a_t, b_1, \ldots, b_t > 0$ we have the bound

$$\frac{a_1 + \ldots + a_t}{b_1 + \ldots + b_t} \geq \min_i \frac{a_i}{b_i}$$

we get the bound

$$|N(w')| \geq \delta' \min_{(\beta'-\gamma)(n-k) \leq m \leq (\beta'+\gamma)(n-k)} \left(\frac{1-\beta'}{\alpha-\beta'}\right)^{n-k-m} = \delta' \left(\frac{1-\beta'}{\alpha-\beta'}\right)^{(1-\beta'-\gamma)(n-k)}. \tag{7}$$

We will use the following technical claim.

Claim 15. *Let $0 < \alpha < 1$ and define $f : [0, \alpha) \to \mathbb{R}$ by $f(x) = \left(\frac{1-x}{\alpha-x}\right)^{1-x}$. Then f is monotone increasing.*

We prove Claim 15 in Appendix A. Applying Claim 15 we get that since $\beta' \geq \beta$ we have

$$\left(\frac{1-\beta'}{\alpha-\beta'}\right)^{1-\beta'} \geq \left(\frac{1-\beta}{\alpha-\beta}\right)^{1-\beta}$$

hence

$$|N(w')| \geq \delta' \left(\frac{1-\beta'}{\alpha-\beta'}\right)^{(1-\beta')\left(\frac{1-\beta'-\gamma}{1-\beta'}\right)(n-k)} \tag{8}$$

$$\geq \delta' \left(\frac{1-\beta}{\alpha-\beta}\right)^{(1-\beta)(n-k)\left(1-\frac{\gamma}{1-\beta'}\right)} \tag{9}$$

$$\geq \delta' \left(\frac{1-\beta}{\alpha-\beta}\right)^{(1-\beta)(n-k)\left(1-\frac{\gamma}{1-\alpha}\right)} \tag{10}$$

□

We obtain as a corollary a lower bound on $|X|$.

Claim 16. $|X| \geq \delta|\mathbf{U}|^k \left(\frac{1-\beta}{\alpha-\beta}\right)^{(1-\beta)(n-k)\left(1-\frac{\gamma}{1-\alpha}\right)}$.

Proof. By Corollary 13 and Lemma 14 we have

$$\begin{aligned}|X| &= \sum_{w' \in \mathbf{U}^k} |N(w')| \\ &\geq \sum_{w' \in \mathbf{U}^k} \frac{|W(w')|}{|\mathbf{U}|^{n-k}} \left(\frac{1-\beta}{\alpha-\beta}\right)^{(1-\beta)(n-k)\left(1-\frac{\gamma}{1-\alpha}\right)} \\ &\geq \delta |\mathbf{U}|^{n-k} \left(\frac{1-\beta}{\alpha-\beta}\right)^{(1-\beta)(n-k)\left(1-\frac{\gamma}{1-\alpha}\right)}\end{aligned}$$

□

Combining Claims 12 and 16 we deduce the inequality

$$2^M \alpha^k \geq \delta \left(\frac{1-\beta}{\alpha-\beta}\right)^{(1-\beta)(n-k)\left(1-\frac{\gamma}{1-\alpha}\right)}. \tag{11}$$

We now fix parameters. Let $k = cn$ where $0 < c < 1$ is a fixed parameter. Denote $M = M_D(n, \epsilon) = \eta \cdot n \log_2(1/\epsilon)$ where apriory we know that $1 - o(1) \leq \eta \leq \log_2(e) \approx 1.44$. We will prove a lower bound on η.

We think of $n \to \infty$ where the parameters ϵ, c, η are fixed, and take $\delta = 1/n$. This gives the following quantities for α, β, γ:

$$\begin{aligned}\alpha &= \epsilon(1 + \frac{n}{|\mathbf{U}|})(1 + 6\delta) = \epsilon(1 + o(1)) \\ \beta &= \delta^{1/k} 2^{-M/k} = \epsilon^{\eta/c}(1 + o(1)) \\ \gamma &= \sqrt{3\ln(2/\delta)/(n-k)} = o(1).\end{aligned}$$

Substituting the parameters to inequality (11), and taking $n \to \infty$, gives the following simplified form

$$(1/\epsilon)^\eta \epsilon^c \geq \left(\frac{1-\epsilon^{\eta/c}}{\epsilon - \epsilon^{\eta/c}}\right)^{(1-\epsilon^{\eta/c})(1-c)}. \tag{12}$$

Note that for any given fixed value of ϵ, η, Equation 12 should hold for **any** value of $0 < c < 1$. Thus we are now left we a problem in analysis: for a given value of ϵ, what is the minimal value of η such that Equation (12) holds.

3.4 Obtaining the lower bound from Inequality (12)

We start by noting that Equation (12) is monotone in η, that is, if it holds for some η it holds for all $\eta' > \eta$. This can be verified since the LHS is increasing with η while the RHS is decreasing, as can be seen by Claim 15. We thus define

$$\eta^*(\epsilon) = \min\{\eta : \text{Equation (12) holds for } \epsilon, \eta \text{ for all } 0 < c < 1\}$$

We have the bound $M_D(n, \epsilon) \geq \eta^*(\epsilon) \cdot n \log_2(1/\epsilon)$. It is easy to verify that taking limits $c \to 0$ or $c \to 1$ gives the bound $\eta^*(\epsilon) \geq 1$, which we already knew from the entropy lower bound. Thus, in order to get non-trivial lower bounds, we need to consider intermediate values of c.

We start by giving a non-trivial lower bound for the common case of $\epsilon = 1/2$.

Claim 17. $\eta^*(1/2) \geq 1.1$.

Proof. It is straightforward to verify that inequality (12) is not satisfied for $\epsilon = 1/2, \eta = 1.1$ and $c = 0.7$. We empirically found that $\eta^*(1/2) = 1.10213...$ □

Claim 18. *For any* $0 < \epsilon < 1$ *we have* $\eta^*(\epsilon) > 1$.

Proof. Let $0 < c < 1$ be any fixed value. We will show any such value gives a non-trivial lower bound on $\eta^*(\epsilon)$. We know that $\eta = \log_2(e)$ satisfies inequality (12) for any value of $0 < c < 1$, since a Bloom filter [Blo70] gives a dynamic approximate membership data structure using $\log_2(e) \cdot n \log_2(1/\epsilon)$ memory bits. Thus, we can limit ourselves to considering $1 \leq \eta \leq \log_2(e) \approx 1.44$. Define $f : [0, \epsilon) \to \mathbb{R}$ by $f(x) = \left(\frac{1-x}{\epsilon-x}\right)^{1-x}$, and set $\tau = \epsilon^{\log_2(e)/c}$. We first note that By Claim 15 we have

$$\left(\frac{1-\epsilon^{\eta/c}}{\epsilon-\epsilon^{\eta/c}}\right)^{1-\epsilon^{\eta/c}} = f(\epsilon^{\eta/c}) \geq f(\tau).$$

Moreover, by another application of Claim 15 we have

$$f(\tau) > f(0) = 1/\epsilon.$$

Hence, we get that if $\eta \geq \eta^*(\epsilon)$, then by inequality (12) we must have that

$$(1/\epsilon)^{\eta-c} \geq f(\tau)^{1-c}.$$

Define ρ such that $f(\tau) = (1/\epsilon)^\rho$. We must have $\rho > 1$ since $f(\tau) > 1/\epsilon$. Hence we get that we must

$$\eta - c \geq \rho(1-c)$$

hence

$$\eta \geq \rho(1-c) + c > 1.$$

Thus we have the lower bound

$$\eta^*(\epsilon) \geq \rho(1-c) + c,$$

which is non-trivial for any $0 < c < 1$. □

3.5 Improved bounds via recursion

We note that one may use recursion of the argument we presented so far, in order to derive an improved bound on $M_D(n, \epsilon)$. The main claim which can be improved is Claim 8, which gives a bound on β in terms of a covering argument on the first k layers of the graph. We could use instead a recursive argument: first derive a lower bound on $M_D(k, \epsilon)$, and the use it to define β appropriately, i.e.

$$\beta = \delta^{1/k} 2^{-M_D(k,\epsilon)/k}.$$

This is a two-step recursive argument. A general r-step recursive argument entails choosing constants $0 < c_r < \ldots < c_1 < 1$ and performing the analysis for $\{k_i = c_i n\}$. It turns out that using a recursive argument improves the bounds we get using the non-recursive approach, but only slightly. We performed a computer search for $\epsilon = 1/2$ for a recursive sequence $c_1 > \ldots > c_r$ that will give the best result. We obtained the bound $\eta^*(1/2) \geq 1.13$, compared with $\eta^*(1/2) \geq 1.1$ which can be obtained by a non-recursive argument.

References

[ANS10] Yuriy Arbitman, Moni Naor, and Gil Segev. Backyard cuckoo hashing: Constant worst-case operations with a succinct representation, 2010. manuscript.

[Blo70] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

[BM03] A. Broder and M. Mitzenmacher. Network applications of bloom filters: a survey. *Internet Math.*, 1(4):485–509, 2003.

[CFG+78] Larry Carter, Robert Floyd, John Gill, George Markowsky, and Mark Wegman. Exact and approximate membership testers. In *STOC '78: Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 59–65, New York, NY, USA, 1978. ACM.

[DP08] Martin Dietzfelbinger and Rasmus Pagh. Succinct data structures for retrieval and approximate membership (extended abstract). In *ICALP '08: Proceedings of the 35th international colloquium on Automata, Languages and Programming, Part I*, pages 385–396, Berlin, Heidelberg, 2008. Springer-Verlag.

[MP07] Yossi Matias and Ely Porat. Efficient pebbling for list traversal synopses with application to program rollback. *Theor. Comput. Sci.*, 379(3):418–436, 2007.

[Por09] Ely Porat. An optimal bloom filter replacement based on matrix solving. In *CSR '09: Proceedings of the Fourth International Computer Science Symposium in Russia on Computer Science - Theory and Applications*, pages 263–273, Berlin, Heidelberg, 2009. Springer-Verlag.

[PPR05] Anna Pagh, Rasmus Pagh, and S. Srinivasa Rao. An optimal bloom filter replacement. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 823–829, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[RR03] Rajeev Raman and Satti Srinivasa Rao. Succinct dynamic dictionaries and trees. In *ICALP'03: Proceedings of the 30th international conference on Automata, languages and programming*, pages 357–368, Berlin, Heidelberg, 2003. Springer-Verlag.

A Proof of Claim 15

Let $0 < \alpha < 1$ and define $f : [0, \alpha) \to \mathbb{R}$ by $f(x) = \left(\frac{1-x}{\alpha-x}\right)^{1-x}$. We will prove f is monotone increasing. Let $g(x) = \ln(f(x)) = (1-x)(\ln(1-x) - \ln(\alpha-x))$. It is sufficient to prove g is monotone increasing. We have

$$\begin{aligned} g'(x) &= \ln(\alpha - x) - \ln(1-x) - 1 + \frac{1-x}{\alpha - x} \\ &= -\ln\left(\frac{1-x}{\alpha-x}\right) - 1 + \frac{1-x}{\alpha-x}. \end{aligned}$$

For any $z > 0$ we have $e^z > 1 + z$. Thus for any $y > 1$ we have $\ln(y) < y - 1$. Set $y = \frac{1-x}{\alpha-x} > 1$. We have

$$g'(x) = -\ln(y) - 1 + y > 0.$$

Hence g is monotone increasing, and so is f.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de