Electronic Colloquium on Computational Complexity, Report No. 9 (2024)

Unambiguous parity-query complexity*

Dmytro Gavinsky†‡

January 20, 2024

Abstract

We give a lower bound of $\Omega\left(\sqrt{n}\right)$ on the *unambiguous randomised parity-query complexity* of the *approximate majority* problem – that is, on the lowest randomised parity-query complexity of any *function* over $\{0,1\}^n$ whose value is "0" if the Hamming weight of the input is at most $n/3$, is "1" if the weight is at least $2n/3$, and may be arbitrary otherwise.

1 Introduction

The computational model of *parity queries* is a well-known natural strengthening of the (more common) bit-wise query model: in both models the input is $x \in \{0,1\}^n$, in the bit-query model a protocol can get the value of a single bit of x at each step, while in the parity-query model it can, for any $s \subseteq \{1,\dots,n\}$, obtain the value of $\bigoplus_{i\in s} x_i$ via a single query. Alternatively, one may view a parity-query protocol as receiving its input x from the binary linear space $\mathcal{GF}_2^n$ and being able to make an arbitrary linear query to the coordinates of x at each step.

From the combinatorial standpoint, in the standard bit-query model a deterministic protocol partitions its input space $\{0,1\}^n$ into same-answer *sub-cubes*, while a deterministic parity-query protocol partitions its input space $\mathcal{GF}_2^n \simeq \{0,1\}^n$ into same-answer *affine subspaces*, and so, the parity-query model is a linear closure of the bit-query one. The corresponding randomised models are the convex closures of their deterministic counterparts. The parity-query model is, obviously, at least as strong as the bit-query one, and it is not hard to see that it can be much stronger (e.g., both deterministic and randomised bit-query complexity of computing $x_1 \oplus x_2 \oplus \ldots \oplus x_n$ is n).

Let g be a *relational computational problem* – that is, one that admits non-unique correct answers for some input values. Let $\mathcal{C}$ be a complexity measure that is applicable to g. We will denote by $\bar{\mathcal{C}}$ the *unambiguous-$\mathcal{C}$* – that is, let $\bar{\mathcal{C}}(g)$ be the minimum of $\mathcal{C}(f_g)$ over all *functions* f_g that have the same domain as g and values that are consistent with g (that is, $f_g(x)$ is a correct answer to $g(x)$ for every x in the domain).

Our notion of *unambiguous algorithms* (or protocols) was studied in other computational regimes under the names of *Bellagio algorithms* and *pseudo-deterministic algorithms*. Italophile himself, the author would happily accept the former term, but unfortunately it is not commonly used and the connotation might not be obvious. As for the latter, it perfectly fits models like bit-wise query complexity, where randomised algorithms cannot outperform qualitatively the deterministic ones

*This is a preliminary version...

†In 2022 the author changed his first name, as explained on his Internet page.

‡Institute of Mathematics of the Czech Academy of Sciences, Žitna 25, Praha 1, Czech Republic.

Partially funded by the grant 19-27871X of GA ČR and by RVO: 67985840.

ISSN 1433-8092

in computing functional problems: in that case forcing unambiguous answers – that is, turning the original relational problem into a functional one – means depriving the computing algorithms of its "randomised privileges". On the other hand, for structurally richer settings like parity queries or communication complexity, the requirement of unambiguity is unrelated to the protocol's being deterministic, as the qualitative superiority of randomness in those models can be manifested in solving functional problems as well. We will use the term *unambiguity* with respect to all computational settings considered in this work.

The concept of unambiguity highlights some alluring gaps in the current understanding of randomised computation. For instance, consider the *approximate majority* relation – one of the simplest examples that demonstrate the "power of randomness" – here the answer must be "0" if the Hamming weight of the input $x \in \{0,1\}^n$ is at most $n/3$, "1" if the weight is at least $2n/3$ and may be arbitrary otherwise.

- In the standard bit-query model both the deterministic ($\Omega(n)$) and the randomised ($O(1)$) complexities of approximate majority can be established via nearly trivial arguments;
- the unambiguous randomised query complexity ($\Omega(n)$) demands a somewhat more involved proof;
- if we strengthen the model slightly by admitting parity queries, then bounding the unambiguous randomised complexity becomes even more challenging (as witnessed by this work), while both the deterministic and the randomised cases remain trivial;
- similar situation can be expected in other "rich enough" models, like communication complexity.[1]

The intricacy of analysing the unambiguous complexity can be reflected by the "added universal quantifier" in the corresponding formal statement. Say, a usual claim of intractability of a relational task g can be interpreted as

$$\text{for } \textbf{any} \textit{ efficient protocol } \Pi \textit{ there } \underline{\textit{exists}} \textit{ input } x\textit{, such that } \Pi(x) \textit{ disagrees with } g(x), \tag{1}$$

while a claim of its *unambiguous* intractability would stand for

$$\begin{aligned} &\textit{for } \textbf{\textit{any}} \textit{ function } f_g \textit{ that is consistent with } g \\ &\text{and } \textbf{\textit{any}} \textit{ efficient protocol } \Pi \textit{ there } \underline{\textit{exists}} \textit{ input } x\textit{, such that } \Pi(x) \neq f_g(x). \end{aligned} \tag{2}$$

The former statement implies the latter, which is not surprising: if g admits no efficient randomised protocol, then it admits no efficient unambiguous randomised protocol either. If, on the other hand, a relational problem does have an efficient randomised protocol – which is the case, in particular, for approximate majority in the parity-query model – then arguing its unambiguous randomised intractability requires *distinguishing* between statements (1) and (2) in the analysis, which may require rather fine tuning of the argument.

The state of affairs when one seems to understand well *why approximate majority is hard for deterministic protocols*, as well as *why the problem is easy for randomised protocols*, but not *why any function that computes approximate majority is hard for randomised protocols* seems to imply that one does not understand well the role of randomness in computing approximate majority – one of the "most canonical" problems demonstrating the utility of randomness in query protocols.

[1] To the best of our knowledge, no non-trivial lower bound on the unambiguous complexity is known in communication complexity and this paper presents the first such example in the parity-query model.

1.1 Technical challenges

And so, for this work we picked the setting of *parity queries* – a model where determining *the unambiguous randomised complexity of approximate majority* looks challenging, but not hopeless. This choice can be further justified via observing that three approaches that may look very natural at first glance are inherently incapable of providing a non-trivial lower bound on the quantity that we are interested in. The first two of them are "tactical" approaches that work in the case of the standard bit-query model.

The *most straightforward way* of arguing the hardness of unambiguity in the *bit-query model* is to apply the result of [Nis91] stating that the randomised and the deterministic complexities of a function in that model are qualitatively the same (they can differ at most polynomially). As the deterministic complexity of approximate majority is high, every function consistent with this relation is hard for deterministic bit-query protocols, and therefore for the randomised ones also. Obviously, this approach is unsuitable for parity queries, as here the gap between the randomised and the deterministic complexity of a function can be as wide as $O(1)$ vs. $\Omega(n)$ (cf. Claim 1).

The *second approach* that works well for the same case of the standard query model is even "more ad hoc", but also quantitatively tighter than the first one (it leads to the optimal bound of $\Omega(n)$, while the previous argument can only give $\Omega\left(n^{1/3}\right)$). Let f be a function consistent with $\widetilde{Maj}$ and computed by a randomised bit-query protocol, and let $x \in \{0,1\}^n$ be a point of (globally) maximal Hamming weight for which $f(x) = 0$. Such $x \in \{0,1\}^n$ must have Hamming weight less than $2n/3$, that is, $s \stackrel{\text{def}}{=} \left\{i \middle| x_i = 0\right\}$ contains at least $n/3$ elements. By the assumption, if x' differs from x only at some non-empty subset of bits whose indices are in s, then $f(x') = 1$ – that is, a protocol for f makes enough queries, at least in some cases, in order to solve the *or* function on $|s|$ bits, and the randomised bit-query complexity of that task is, trivially, in $\Omega(|s|) = \Omega(n)$. This argument cannot handle the case of parity-queries, as here the randomised complexity of the *or* function is in $O(1)$ (again, cf. Claim 1).

The *third approach* is one of the most efficient and widely applicable known strategies for analysing randomised complexity and, as such, is entitled to a section of its own.

1.2 The minimax principle

To prove a lower bound on the *randomised* complexity of certain task, one can start by guessing a "hard" input distribution μ and then prove that any *deterministic* protocol errs very often with respect to μ. On the one hand, as long as the family of randomised solutions in the considered computational model is the convex closure of the family of deterministic ones (which is the case not for all, but for most of natural computational models, in particular, for all those of interest to us in this work), Von Neumann's minimax principle [vN28] implies that such hard distribution necessarily exists, so this approach can be viewed as universal.[2] On the other hand, the simpler structure of deterministic protocols usually makes their computational potential much easier to understand and to analyse than that of their "convex generalisation": the randomised protocols.

The regime of unambiguous complexity, on the other hand, is certainly not a "mere convex combinations" of deterministic protocols. Recall that a relation g has an efficient unambiguous solution if *there exists* some function f that both agrees with g and has an efficient solution. Suppose that g is hard, then *any f that agrees with g is hard* too, that is – here applies the minimax principle principle in

[2] There are interesting models of computation where the randomised regime is not closed with respect to convex combinations of protocols: e.g., such is the case for the communication-complexity model of *simultaneous message passing with private randomness*, where the minimax principle in the above form doesn't hold (*equality* with constant error is hard in the worst-case regime but easy deterministically with respect to any fixed distribution).

the above form – *for any f that agrees with g there is a hard distribution μ_f*. These μ_f, however, *may* be different for different functions f, not corresponding to any "universally hard" distribution for the relation g itself.

What is more, these μ_f not only *can* but actually *must* disagree with one another, allowing for no single hard distribution for the relation g – at least, in all the qualitatively interesting cases, that is, as long as the gap between the deterministic and the randomised complexities of g is significant.

Consider, for instance, a "structurally meaningful" case where $R^{\oplus}(g)$ is at most poly-logarithmic but both $D^{\oplus}(g)$ and $\overline{R^{\oplus}}(g)$ are beyond that. Then for any $\varepsilon > 0$ and distribution μ there exists a *deterministic* protocol $\Pi_{\mu,\varepsilon}$ of complexity in $O(R^{\oplus}(g)/\log\varepsilon)$ that solves g with error at most ε with respect to μ – denote by $f_{\mu,\varepsilon}$ the actual function computed by $\Pi_{\mu,\varepsilon}$ (it is well-defined, as $\Pi_{\mu,\varepsilon}$ is deterministic). This $f_{\mu,\varepsilon}$ disagrees with the relation g with probability at most ε with respect to μ. Let $f'_{\mu,\varepsilon}(\cdot)$ be an arbitrary function with value $f_{\mu,\varepsilon}(x)$ when $(x,\ f_{\mu,\varepsilon}(x)) \in g$ and some value from $\{a \,|\, (x, a) \in g\}$ otherwise. Note that $f'_{\mu,\varepsilon}$ perfectly agrees with g and $D^{\oplus}_{\mu,\varepsilon}(f'_{\mu,\varepsilon}) \in O(R^{\oplus}(g)/\log\varepsilon)$, as witnessed by the protocol $\Pi_{\mu,\varepsilon}$ (here "$D^{\oplus}_{\mu,\varepsilon}$" stands for the deterministic complexity of solving the task with error at most ε with respect to μ).

In other words, for every μ and $\varepsilon > 0$, there *always is* a function $f'_{\mu,\varepsilon}$ that perfectly agrees with g and for which there exists an efficient deterministic protocol with error at most ε with respect to μ – in spite of the assumed intractability of the relation g itself for *unambiguous* randomised protocols.[3] Therefore, the concept of unambiguous complexity can be viewed not only as "computational randomness *without* the minimax principle" (which by itself would likely be very interesting, although not unique), but as "computational randomness *against* the minimax principle" since the setting guarantees that either the considered case is structurally trivial or for every distribution μ the problem is easy.

This work

To illustrate the structural richness of unambiguity, we present a lower bound of $\Omega(\sqrt{n})$ on the unambiguous randomised parity-query complexity of the approximate majority problem. It follows that with respect to parity queries there exist:

- a problem that is intractable deterministically but easy for protocols with randomness, even under the requirement of unambiguity;
- a problem that is intractable deterministically, easy with randomness but becomes intractable for randomised protocols if the requirement of unambiguity is imposed.

Prior work

The concept of unambiguous complexity was introduced under the name of *Bellagio algorithms* by Gat and Goldwasser [GG11] and first studied in the context of the bit-wise query model under the name of *pseudo-determinism* by Goldreich, Goldwasser and Ron [GGR13].

2 Preliminaries and definitions

By default the logarithms are base-2. We will write $[n]$ to denote the set $\{1, \ldots, n\} \subset \mathbb{N}$ for $n \in \mathbb{N} \cup \{0\}$ and let $[a] \stackrel{\text{def}}{=} [\min\{0, \lfloor a \rfloor\}]$ for $a \in \mathbb{R}$. Let $(a,\ b)$, $[a,\ b]$, $[a,\ b)$ and $(a,\ b]$ denote the corresponding

[3] The above argument readily adapts to virtually every reasonable model of computation with natural notions of determinism, randomness and unambiguity.

open, closed and half-open intervals in $\mathbb{R}$. Towards readability, we will allow both $\{\cdot|\cdot\}$ and $\{\cdot:\cdot\}$ to denote sets with conditions (preferring the former). Let $\bot$ and $\top$ denote, respectively, the false and the true values.

For a linear space S, we will write "$\leq S$", "$< S$", "$\preceq S$" and "$\prec S$" to denote, respectively, its subspaces, proper subspaces, affine subspaces and proper affine subspaces:

$$A + b \preceq S$$

for any $A \leq S$ and $b \in S$ (and similarly for "$\prec$" and "$<$"). The zero element will be denoted by $\bar{0}$.

For any set S, we will denote by $\mathsf{pow}(S)$ the family of all its subsets and by $\binom{S}{t}$ the family of its size-t subsets. We will write $\mathcal{U}_S$ to denote the uniform distribution over the elements of S and use the notation "$X \sim \mathcal{U}_S$" and "$X \circledS S$" interchangeably. For a finite $S \subset \mathbb{N}$, we will write $S(i)$ to address the i'th element of S in natural ordering.

For $x \in \{0,1\}^n$, we let $|x|$ denote its Hamming weight. For $i \in [n]$, we will write either x_i or $x(i)$ to address the i'th bit of x (preferring "x_i" unless it causes ambiguity) and for any $s \subseteq [n]$ both x_s and $x(s)$ will stand for $x_{s(1)} \dots x_{s(|s|)} \in \{0,1\}^{|s|}$.

At times we will assume the following trivial isomorphism:

- between the bit-strings $x \in \{0,1\}^n$ and the subsets $\{i | x_i = 1\} \subseteq [n]$ (in particular, the notation $\binom{[n]}{k}$ will stand for $\{x \in \{0,1\}^n \,||x| = k\}$, and $x \cap y$ will address the set $\{i \in [n] | x_i = y_i = 1\}$);
- between the n-bit strings and the elements of $\mathcal{GF}_2^n$ (in particular, the answer to the *linear* query represented by $s \subseteq [n]$ will be the parity $\bigoplus_{i \in s} x_i$ for $x \in \{0,1\}^n$).

For $i \in [n]$, we will denote by e_i both the unit vector in $\mathcal{GF}_2^n$ and the weight-1 bit string in $\{0,1\}^n$ that correspond to $\{i\}$.

We will use the following notation for affine subspaces of $\mathcal{GF}_2^n$: If $C \preceq \mathcal{GF}_2^n$, then $C' \leq \mathcal{GF}_2^n$ is the "supporting" subspace for C, defined as

$$C' \stackrel{\text{def}}{=} C + C = C + c_0$$

for any $c_0 \in C$ and $\overline{C} \leq \mathcal{GF}_2^n$ is the *dual* subspace (sometimes called the *annihilator*) of either C or C', defined as

$$\begin{aligned} \overline{C} = \overline{C'} &\stackrel{\text{def}}{=} \left\{ x \in \mathcal{GF}_2^n \middle| \forall\, y \in C' :\ \langle x\,,\, y \rangle = 0 \right\} \\ &= \left\{ x \in \mathcal{GF}_2^n \middle| \exists\, c_x \in \mathcal{GF}_2 :\ \forall\, y \in C :\ \langle x\,,\, y \rangle = c_x \right\}, \end{aligned}$$

where "$\langle \cdot\,,\, \cdot \rangle$" stands for the inner product in $\mathcal{GF}_2^n$.[4] The *co-dimension* of C equals $n - \log|C| = \dim\left(\overline{C}\right)$.

2.1 Query complexity and unambiguity

The standard model of *query complexity* is among the simplest and the most natural settings for analysing the computational complexity of a Boolean function.

Definition 1 (*R^q and D^q, (standard) query complexity*)**.** *Let $x \in \{0,1\}^n$ and Π be a deterministic protocol that queries individual bits of x and outputs a value denoted by $\Pi(x)$. The* complexity *of Π is the maximum possible number of queries that it makes.*

A randomised *query protocol is a convex combination of deterministic protocols: $(\Pi_i, \alpha_i)_i$ with $\sum_i \alpha_i = 1$ outputs $\Pi_i(x)$ with probability α_i for every i and x. The complexity of such protocol is the maximum complexity of an individual Π_i.*

[4] As $\mathcal{GF}_2^n$ is not an inner product space for $n \geq 2$, the intersection of a linear subspace with its own dual can be a non-trivial linear subspace. On the other hand, $\overline{\overline{A}} = A$ and $\dim(A) + \dim\left(\overline{A}\right) = n$ for any linear subspace $A \subseteq \mathcal{GF}_2^n$.

The deterministic (randomised) query complexity *of a function f, denoted by $D^q(f)$ ($R^q(f)$), is at most the complexity of a deterministic (randomised) query protocol that outputs $f(x)$ (with probability at least $2/3$) for every input value x.*

A query protocol is called efficient *if its complexity is at most* poly-log$(())n$.

Obviously, every D^q-protocol of complexity k partitions the input space $\{0,1\}^n$ into at most 2^k monochromatic (with respect to the computed function) *sub-cubes* of co-dimension at most k.

The setting of *parity query complexity* is a natural strengthening of the standard model.

Definition 2 (*$R^\oplus$ and $D^\oplus$, parity query complexity*)**.** *Let $x \in \mathcal{GF}_2^n$ and Π be a deterministic protocol that makes linear queries (or parity queries) to the bits of x, that is, for the query represented by $s \subseteq [n]$ the protocol receives the response $\bigoplus_{i \in s} x_i$. Denote the output of the protocol by $\Pi(x)$. The* complexity *of Π is the maximum possible number of linear queries that it makes.*

A randomised *parity-query protocol is a convex combination of deterministic protocols (cf. Definition 1).*

The deterministic (randomised) parity-query complexity *of a function f, denoted by $D^\oplus(f)$ ($R^\oplus(f)$), is at most the complexity of a deterministic (randomised) parity-query protocol that outputs $f(x)$ (with probability at least $2/3$) for every input value x.*

A parity-query protocol is called efficient *if its complexity is at most* poly-log$(())n$.

We will see in Section 3 that every $D^\oplus$-protocol of complexity k partitions the input space $\mathcal{GF}_2^n$ into at most 2^k monochromatic (with respect to the computed function) *affine subspaces* of co-dimension at most k.

The primary context of this work is *structural complexity* and we will ask whether one computational setting can "qualitatively outperform" the other, that is, whether there is a computational problem that has an efficient solution in the model $\mathcal{M}_1$, thought not in $\mathcal{M}_2$. In other words, computational problems are tools for separating computational models, and there is a class of problems that generalises the class of functions and, in some cases, give rise to model separations that wouldn't be possible via functions alone.

Definition 3 (*relational problems*)**.** *Let X be the domain, that is, the set of possible input values to a computational problem, and let A be the range, that is, the set of possible answers. Then a relation $g \subseteq X \times A$ defines the following computational problem: "a" is a* correct *answer with respect to the input value $x \in X$ if $(x,a) \in g$ and it is* wrong *otherwise.*

A relation is called partial *if for some input values there is no correct answer, that is, $\exists\, x \in X : \forall\, a \in A :\ (x,a) \notin g$; otherwise the relation is* total.

A function $f : X \to A$ is said to be consistent *with g if all its answers are consistent with those of the relation, that is, $\forall\, x \in X :\ (x, f(x)) \in g$.*

In other words, relations admit ambiguous answers or no answer at all for some input values, while the functional problems are a special case with exactly one correct answer being assigned to every $x \in X$. All relations considered in this work are *total*.[5]

[5] Partial relations are usually interpreted as "guarantees" that only the input values for which there is a correct answer are to be expected: otherwise a protocol is allowed to answer "anything". This is useful in the context of *partial functions* (or *promise functions*): these are relational problems with *at most* one correct answer corresponding to every $x \in X$ (as a class of computational problems, it is intermediate between functions and relations). On the other hand, in the case of relational problems one may consider, instead of a partial relation g, the total one $g' \stackrel{\text{def}}{=} g \cup \left\{x \middle| \forall\, a \in A :\ (x,a) \notin g\right\} \times A$, as g and g' are describing the same computational problem.

In Definitions 1 and 2 we have addressed the complexity of functional problems only. There are at least two natural ways to generalise it for a relation $g \subseteq X \times A$:

- the complexity of a g can be defined as the smallest complexity of a protocol that produces answers that are *correct* with respect to g;
- alternatively, it can be defined as the smallest complexity of a function f that has the same domain as g and "agrees" with it answer-wise, that is, $\forall\, x \in X :\ (x, f(x)) \in g$.

The corresponding pair of definitions are equivalent for the deterministic models D^q and $D^\oplus$; on the other hand, for both R^q and $R^\oplus$ the first version is the standard notion of relational complexity, while the second one is the unambiguous complexity.

Definition 4 (*deterministic complexity of relations*)**.** *Let $g \subseteq X \times A$ be a relational problem and $\mathcal{C}$ be either D^q or $D^\oplus$.*

The $\mathcal{C}$-complexity *of g, denoted by $\mathcal{C}(g)$, is at most $\mathcal{C}(f)$ for any $f :\ X \to A$ such that $\forall\, x \in X :\ (x, f(x)) \in g$.*

Definition 5 (*randomised complexity of relations; unambiguity*)**.** *Let $g \subseteq X \times A$ be a relational problem and $\mathcal{C}$ be either R^q or $R^\oplus$.*

The $\mathcal{C}$-complexity *of g, denoted by $\mathcal{C}(g)$, is at most the $\mathcal{C}$-complexity of a randomised protocol from the corresponding query model that outputs with probability at least $2/3$ a value from $\left\{a \in A \middle| (x,a) \in g\right\}$ for every $x \in X$.*

The unambiguous $\mathcal{C}$-complexity*, denoted by $\overline{\mathcal{C}}(g)$, is at most $\mathcal{C}(f)$ for any $f :\ X \to A$ such that $\forall\, x \in X :\ (x, f(x)) \in g$.*

2.2 Tasks to consider

It follows readily from the definitions that

$$\begin{aligned} &D^q(g) \geq D^\oplus(g), R^q(g) \geq R^\oplus(g) \ \text{ and } \ \overline{R^q}(g) \geq \overline{R^\oplus}(g); \\ &R^q(g) \leq \overline{R^q}(g) \leq D^q(g) \ \text{ and } \ R^\oplus(g) \leq \overline{R^\oplus}(g) \leq D^\oplus(g) \end{aligned} \tag{3}$$

for any relation g. The inequalities in the first line of (3) can correspond to separations of $O(1)$ vs. $\Omega(n)$: this is witnessed, in particular, by the parity function $\bigoplus_{i\in[n]} x_i$.

The first inequality chain of the second line of (3) can correspond to at most polynomial gaps: it is known [Nis91] that $D^q(g) \in O\big((R^q(g))^3\big)$, and therefore from the standpoint of structural complexity all the considered regimes of (standard) query complexity are equivalent. This leaves us with the second chain, namely $R^\oplus(g) \leq \overline{R^\oplus}(g) \leq D^\oplus(g)$. As we are going to study the effect of the unambiguity upon the *efficient computability* of tasks, we shall only consider those relations g for which $R^\oplus(g) \in \text{poly-log}(()n)$ and $D^\oplus(g)$ is much higher, preferably in $n^{\Omega(1)}$.

Our example of g_1 such that $R^\oplus(g_1) = \overline{R^\oplus}(g_1) \ll D^\oplus(g_1)$ is fairly simple: it is the *or* function.

Claim 1. *Let $X = \{0,1\}^n$ and $g_1 :\ X \to \{0,1\}$ be the* or *function:*

$$g_1(x) = \vee(x) \stackrel{def}{=} \bigvee_{i\in[n]} x_i.$$

Then

$$R^\oplus(g_1) = \overline{R^\oplus}(g_1) \in O(1) \quad \textit{and} \quad D^\oplus(g_1) = n.$$

Proof. As g_1 is a function, $R^{\oplus}(g_1) = \overline{R^{\oplus}}(g_1)$.

To compute $\vee(x)$, a protocol must check whether $x = 0^n$ and output "0" if that is the case and "1" otherwise.

If $x \neq 0^n$, then more than half of non-empty $s \subseteq [n]$ satisfy $\bigoplus_{i \in s} x_i = 1$ (let $x_{i_0} = 1$, then $\bigoplus_{i \in s'} x_i \neq \bigoplus_{i \in s' \cup \{i_0\}} x_i$ for each $s' \subseteq [n] \setminus \{i_0\}$ and $\bigoplus_{i \in \emptyset} x_i = 0$). Accordingly, a single parity-query with uniformly random non-empty $s \subseteq [n]$ will return "1" with probability greater than $1/2$ if $x \neq 0^n$ and "0" with certainty if $x = 0^n$. A constant number of independent random parity-queries allows solving $\vee(x)$ with arbitrarily small constant error probability.

On the other hand, if a *deterministic* protocol that has asked at most $n-1$ parity-queries answers "0", then, obviously, there is some $x \neq 0^n$ that is consistent with all the answers received protocol; accordingly, this protocol does not compute $\vee(x)$.

■*Claim 1*

A very similar behaviour with respect to unambiguity is demonstrated by the *search* relation, which can be viewed as a natural relational version of the function $\vee(x)$.

Corollary 1. *Let $X = \{0,1\}^n$ and $Search \subseteq X \times [n]$ be the* search *relation:*

$$g_1' = Search \stackrel{def}{=} \left\{ (x,\ i) \middle| x = 0^n \ \vee\ x_i = 1 \right\}.$$

Then

$$R^{\oplus}(g_1'),\ \overline{R^{\oplus}}(g_1') \in O(\log n \cdot \log\log n) \quad \textit{and} \quad D^{\oplus}(g_1') = n.$$

Proof. Solving $Search$ is at least as hard as solving the *or* function and $D^{\oplus}(Search) = n$ follows from Claim 1.

To solve $Search$ in the model of $\overline{R^{\oplus}}$ (as well as in $R^{\oplus}$), we perform a binary search for the smallest non-zero coordinate of x and output it if $x \neq 0^n$. This requires solving $\log n$ instances of the $\vee$ function with error probability $O(1/\log n)$, which can be achieved via solving $O(\log n \cdot \log\log n)$ instances with error $O(1)$ and the statement follows, again, from Claim 1.

■*Corollary 1*

Next we define the relational problem whose analysis will be the primary technical concern of this work – the *approximate majority* relation, which is our example of g_2 such that $R^{\oplus}(g_2) \ll \overline{R^{\oplus}}(g_2)$.

Definition 6 ($\widetilde{Maj}$, *approximate majority problem*)**.** *Let $X = \{0,1\}^n$. The following relational problem is called* approximate majority*:*

$$\begin{aligned}\widetilde{Maj} \stackrel{def}{=} & \left\{ x \in X \middle| |x| \leq \frac{n}{3} \right\} \times \{0\} \\ & \cup \left\{ x \in X \middle| \frac{n}{3} < |x| < \frac{2n}{3} \right\} \times \{0,1\} \\ & \cup \left\{ x \in X \middle| \frac{2n}{3} \leq |x| \right\} \times \{1\}.\end{aligned}$$

3 Some properties of affine subspaces in $\mathcal{GF}_2^n$

Parity-query protocols are naturally viewed as *partitions* of $\mathcal{GF}_2^n$ into *affine subspaces*.

Recall Definition 2 and assume that Π is a $D^{\oplus}$-protocol of complexity k. It has a natural representation as a binary tree T_Π of depth k: execution starts from the root, every non-leave vertex is

marked with a parity query, the two edges leaving a node are marked by the complementary possible answers to the node's query and the leaves are marked with the answers that the protocol returns when reaching it.

The state of Π at every moment – that is, at each vertex of T_Π (either a leaf or not) – corresponds to an *affine subspace of* $\mathcal{GF}_2^n$: Let $C \subseteq \{0,1\}^n$ be the set of input values corresponding to a vertex in T_Π at depth $d \leq k$, that is, for some

$$(s_1, a_1), \ldots, (s_d, a_d) \in \mathsf{pow}\big([n]\big) \times \{0,1\}$$

it holds that

$$C = \left\{ x \in \{0,1\}^n \middle| \forall\, j \in [d] :\ \oplus_{i \in s_j} x_i = a_j \right\} = \left\{ x \in \mathcal{GF}_2^n \middle| \forall\, j \in [d] :\ \sum_{i \in s_j} x_i = a_j \right\}.$$

As a valid protocol state, C must be non-empty.

Assume that the sequence $(s_1, a_1), \ldots, (s_d, a_d)$ is a shortest among those that define our C in the above sense. Then $s_1, \ldots, s_d \in \mathcal{GF}_2^n$ must be linearly independent, as if, say, $s_1 + \cdots + s_{t-1} = s_t$ for some $t \leq d$, then $\left[\oplus_{i \in s_1} x_i = a_1, \ldots, \oplus_{i \in s_{t-1}} x_i = a_{t-1}\right]$ implies $[\oplus_{i \in s_t} x_i = a_t]$ or its negation (or both), which contradicts at least one of the above assumptions. Accordingly, $C' \stackrel{\text{def}}{=} \left\{ x \in \mathcal{GF}_2^n \middle| \sum_{i \in s_1} x_i = 0, \ldots, \sum_{i \in s_d} x_d = 0 \right\}$ is a subspace of $\mathcal{GF}_2^n$ of co-dimension d and $C = C' + x_0$ for any $x_0 \in C$, i.e., $C \preceq \mathcal{GF}_2^n$.

The leaves of the protocol tree T_Π correspond to disjoint monochromatic (with respect to the computed function) affine subspaces of $\mathcal{GF}_2^n$ of co-dimension at most k, together forming a monochromatic partition of $\mathcal{GF}_2^n$ of size at most 2^k (attained if and only if T_Π is complete).

In this work we will investigate some properties of affine subspaces in $\mathcal{GF}_2^n$.

Lemma 1 (*Likely unfixed coordinates*)**.** *Let $n \geq 14$ and $C \preceq \mathcal{GF}_2^n$, $\dim(C) \geq 2n/3$. Then there exists $J \subset [n]$, $|J| = \lfloor n/3 \rfloor$, such that for every $j \in J$, X_j is unbiased when $X \sim \mathcal{U}_C$ and*

$$\forall\, D \preceq \mathcal{GF}_2^n :\ \underset{j \in J}{\boldsymbol{Pr}}\big[x_j \text{ is fixed for all } x \in C \cap D\big] \leq 7 \cdot \frac{\dim(C) - \dim(C \cap D)}{n}.$$

The claim is non-trivial owing to the universal quantifier in front of "D": that is, J can depend only on C and must be universal with respect to D.[6] The claim is useful, as it will be applied later in the argument. Finally, the claim is interesting, as it highlights a structural property that is special to affine spaces: if C were allowed to be any large subset of $\mathcal{GF}_2^n$, then the analogous statement would be false (even if we drop the requirement for every X_j to be unbiased under $X \sim \mathcal{U}_C$).[7]

Let us have a closer look at the linear structure underlying the lemma.

[6] If D were known, then $C \cap D$ would be known too, in which case letting J contain as many as possible indices of non-fixed coordinates for $x \in C \cap D$ and the rest of indices non-fixed for $x \in C$ would satisfy the lemma promise.

[7] For sufficiently large $m \in \Theta(n)$ with $\log m \in \mathbb{N}$, pick $s_1, \ldots, s_m \in \binom{[n-\log m]}{\lfloor n/4 \rfloor}$ uniformly at random and let

$$C \stackrel{\text{def}}{=} \bigcup_{i=1}^{m} \left\{ x \in \mathcal{GF}_2^n \middle| x_{s_i} = \bar{0},\ x_{\{n-\log m+1,\ldots,n\}} = bin(i-1) \right\},$$

where $bin(i-1) \in \{0,1\}^{\log m}$ is the binary representation of $i-1$, then $|C| \approx 2^{3n/4}$. For large enough n with high probability it will be the case that for every $J \in \binom{[n]}{\lfloor n/3 \rfloor}$ there is some $i_J \in [m]$, such that $|J \cap s_{i_J}| \geq |J|/5$. Let $D_J \stackrel{\text{def}}{=} \left\{ x \in \mathcal{GF}_2^n \middle| x_{\{n-\log m+1,\ldots,n\}} = bin(i_J - 1) \right\}$ – clearly, this is an affine space of co-dimension $\log m$ and $\boldsymbol{Pr}_{j \in J}[x_j \text{ is fixed for all } x \in C \cap D] \geq 1/5$.

Think about affine spaces in $\mathcal{GF}_2^n$ as being "parametrised" (and therefore represented) by the corresponding dual spaces.[8] On the one hand,

$$\overline{C \cap D} = \mathsf{span}\big(\overline{C} \cup \overline{D}\big) = \overline{C} + \overline{D},$$

where "+" stands for element-wise addition (and the equality holds as we are working in $\mathcal{GF}_2^n$, where a linear combination is determined by the set of summands with non-zero coefficients). On the other hand, x_j is fixed for all $x \in C \cap D$ if and only if $e_j \in \overline{C \cap D}$. That is, the probability in the formulation of Lemma 1 can be rewritten as

$$\underset{j \in\!\!\circlearrowright J}{\boldsymbol{Pr}}\big[e_j \in \overline{C} + \overline{D}\big] \tag{4}$$

and the lemma claims a rather strong upper bound on it (again, for any D).

Note that even though the set $J \subset [n]$ in the statement can be made a function of the affine space C (the corresponding statement would be logically equivalent to our lemma), constructing J explicitly may be somewhat challenging: e.g., choosing any J consisting of such j that $e_j \notin \overline{C}$ (that is, X_j is unbiased in $X \circlearrowright C$) is unsuitable, as shown by letting $\overline{C}$ be the linear span of $\{e_1 + e_2, e_1 + e_3, \ldots, e_1 + e_{n/3+1}\}$ and considering $J \stackrel{\text{def}}{=} \{e_2, e_3, \ldots, e_{n/3+1}\}$ and any D with $\overline{D}$ containing e_1: in this case (4) equals 1 as $e_j \in \overline{C} + \overline{D}$ for every $j \in J$.[9]

Proof of Lemma 1. Towards contradiction, assume the opposite, that is, let $\forall J \in \binom{[n]}{\lfloor n/3 \rfloor} : \exists\, D_J \preceq \mathcal{GF}_2^n$, such that

$$\underset{j \in\!\!\circlearrowright J}{\boldsymbol{Pr}}\big[x_j \text{ is fixed for all } x \in C \cap D_J\big] > 7 \cdot \frac{\dim(C) - \dim(C \cap D)}{n}.$$

Based on (4) and the surrounding discussion, the above implies that for any such J there exists $\overline{D_J} \leq \mathcal{GF}_2^n$ such that

$$\frac{\Big|\big\{e_j \big| j \in J\big\} \cap \big(\overline{C} + \overline{D_J}\big)\Big|}{|J|} > 7 \cdot \frac{\dim(C) - \dim(C \cap D_J)}{n} = 7 \cdot \frac{\dim\big(\overline{C} + \overline{D_J}\big) - \dim\big(\overline{C}\big)}{n},$$

that is,

$$\left|\left\{e_j \middle| j \in J\right\} \cap \big(\overline{C} + \overline{D_J}\big)\right| > 2 \cdot \Big(\dim\big(\overline{C} + \overline{D_J}\big) - \dim\big(\overline{C}\big)\Big),$$

as $|J| = \lfloor n/3 \rfloor$ and $n \geq 14$. As we can take such $\overline{D_J} = \mathsf{span}\Big(w_1, \ldots, w_{\dim\left(\overline{D_J}\right)}\Big)$ that $\overline{C} \cap \overline{D_J} = \bar{0}$,

$$\forall J \in \binom{[n]}{\lfloor n/3 \rfloor} : \exists\, w_1, \ldots, w_k \in \mathcal{GF}_2^n : \left|\left\{e_j \middle| j \in J\right\} \cap \big(\overline{C} + \mathsf{span}(w_1, \ldots, w_k)\big)\right| > 2 \cdot k. \tag{5}$$

Consider the following recursion, indexed by $i \geq 0$. Let

$$C_0 \stackrel{\text{def}}{=} \overline{C},$$

[8] The dual linear space consists, precisely, of the "known parities" for the full set of elements of the corresponding primary affine space – that is, in the context of a parity-query protocol Π, the dual space of (the affine subspace corresponding to) a vertex in T_Π is the linear span of the queries marking the path from the root of the tree to that vertex.

[9] This example can be easily generalised: e.g., allowing non-unique distances from $\overline{C}$ to a very large set of weight-1 vectors that are nevertheless "cancellable" by some D of small co-dimension. Although in this work we do not need an explicit construction of J in the statement of Lemma 1, obtaining it might be interesting for its own sake.

then $\dim(C_0) \le n/3$ by the lemma assumption. As long as $\dim(C_i) \le 2n/3$, let $J_i \in \binom{[n]}{\lfloor n/3 \rfloor}$ be arbitrary, subject to

$$\left\{e_j \middle| j \in J_i\right\} \cap C_i = \emptyset :$$

such J_i necessarily exists, as otherwise $\dim(C_i) \ge \dim\left(\mathsf{span}\left(\left\{e_j \middle| j \in C_i\right\}\right)\right) > 2n/3$. Apply (5) with J_i taking place of J, then let

$$C_{i+1} \stackrel{\text{def}}{=} C_i + \mathsf{span}\left(w_1, \ldots, w_{k_i}\right),$$

where $w_1, \ldots, w_{k_i}$ are those guaranteed by (5). Then by the trivial induction,

$$\dim(C_{i+1}) \le \dim(C_i) + k_i \le \frac{n}{3} + \sum_{j=0}^{i} k_j. \tag{6}$$

Let $\Gamma_i \stackrel{\text{def}}{=} \left\{e_j \middle| j \in J_i\right\} \cap C_{i+1}$ and $\Delta_i \stackrel{\text{def}}{=} \mathsf{span}(w_1, \ldots, w_{k_i})$ (in the earlier notation this is $\overline{D_{J_i}}$). Then (5) guarantees that

$$|\Gamma_i| = \left|\left\{e_j \middle| j \in J_i\right\} \cap (C_i + \Delta_i)\right| \ge \left|\left\{e_j \middle| j \in J_i\right\} \cap \left(\overline{C} + \Delta_i\right)\right| > 2 \cdot k_i,$$

where the first inequality reflects the relation $\overline{C} = C_0 \subseteq C_1 \subseteq \ldots \subseteq C_i$. Trivially,

$$\Gamma_0, \ldots, \Gamma_{i-1} \subseteq C_i$$

but

$$\Gamma_i \cap C_i = \emptyset,$$

as $\Gamma_i \subseteq \left\{e_j \middle| j \in J_i\right\}$ while $\left\{e_j \middle| j \in J_i\right\} \cap C_i = \emptyset$. Accordingly, Γ_i-s are pairwise disjoint and

$$\dim(C_i) \ge \dim\left(\mathsf{span}(\Gamma_0 \cup \cdots \cup \Gamma_{i-1})\right) = \sum_{j=0}^{i-1} \left|\Gamma_j\right| > 2 \cdot \sum_{j=0}^{i-1} k_j, \tag{7}$$

as Γ_i-s consist of unit vectors from $\mathcal{GF}_2^n$.

Evidently, (6) and (7) disagree. Indeed, let i_0 be the index of the last round in the above recursion, then it follows from the halting condition $[\dim(C_{i_0}) > 2n/3]$ and (6):

$$\frac{2n}{3} < \dim\left(C_{i_0}\right) \le \frac{n}{3} + \sum_{j=0}^{i_0-1} k_j,$$

that is,

$$\dim\left(C_{i_0}\right) < 2 \cdot \sum_{j=0}^{i_0-1} k_j,$$

which is in contradiction with (7), as required.

Finally, if $J \subset [n]$ is such that for one or more indices $j \in J$ the coordinate X_j is not unbiased when $X \sim \mathcal{U}_C$ but the set J satisfies the rest of lemma guarantees, then every such j can be replaced by a coordinate that is "free" in C without breaking the guarantees (note that a coordinate X_i can only be either unbiased or fixed when $X \sim \mathcal{U}_C$ for $C \preceq \mathcal{GF}_2^n$).

■*Lemma 1*

We will often consider the behaviour of *deterministic* parity-query protocols (that is, of $D^{\oplus}$-protocols) with respect to random inputs coming from a known distribution of the following form.

Definition 7 (*Distribution $\mu_C^{(t)}$*). *Assume that $C \preceq \mathcal{GF}_2^n$ and $0 \leq t \leq \dim(C) - 2n/3$. Denote by $\mu_C^{(t)}$ the following distribution of $X \in C$.*

- *Choose i_1 uniformly at random from the set J guaranteed by Lemma 1 with respect to C. Let $C_1 \stackrel{def}{=} \{x \in C | x_{i_1} = 1\}$.*
- *For each $2 \leq j \leq t$ consecutively choose i_j uniformly at random from the set J guaranteed by Lemma 1 with respect to C_{j-1} taking place of C and let $C_j \stackrel{def}{=} \{x \in C_{j-1} | x_{i_j} = 1\}$.*
- *Output $X \sim \mathcal{U}_{C_t}$.*

In other words, $X \sim \mu_C^{(t)}$ corresponds choosing $X \in C$ at uniform, subject to consecutive fixing t random coordinates of the elements in C to "1", each time selecting a "likely unfixed" coordinates according to Lemma 1.

We will be arguing that a parity-query protocol Π cannot distinguish well between the input distributions $\mathcal{U}_C$ and $\mu_C^{(t)}$, as long as t is not too large with respect to the complexity of Π. To that end we will use the following facts concerning a pair of affine spaces.

Lemma 2. *Let $C, D \preceq \mathcal{GF}_2^n$, $\dim(C) \geq 3n/4$ and $t \leq \frac{n}{14 \dim(\overline{D})}$. Then*

$$\mu_C^{(t)}(D) \geq \frac{1}{2} \cdot \mathcal{U}_C(D).$$

The claim is non-trivial as the distribution $\mu_C^{(t)}$ doesn't depend on D.

Proof of Lemma 2. Assume that $C \cap D \neq \emptyset$ and $\dim(\overline{D}) \geq 1$ (otherwise the statement holds trivially), then $t \leq n/14$. Also assume $n \geq 14$ (otherwise $t = 0$ and the statement holds trivially). Note that for any C' it holds that

$$\dim(C') - \dim(C' \cap D) = \dim(\overline{C}' + \overline{D}) - \dim(\overline{C}') \leq \dim(\overline{D}).$$

Let us look at the procedure for generating $X \sim \mu_C^{(t)}$, as given by Definition 7. Denote $C_0 \stackrel{\text{def}}{=} C$, then the requirements of Lemma 1 are satisfied with respect to C_0 taking place of C and it holds with respect to the choice of i_1 (from Definition 7) that

$$\Pr_{i_1}\left[x_{i_1} \text{ is fixed for all } x \in C_0 \cap D\right] \leq 7 \cdot \frac{\dim(\overline{D})}{n},$$

where we view i_1 as a random variable. For $2 \leq j \leq t$, let $C_j \stackrel{\text{def}}{=} \{x \in C_{j-1} | x_{i_j} = 1\}$, then the requirements of Lemma 1 are satisfied with respect to C_j, and so,

$$\Pr_{i_j}\left[x_{i_j} \text{ is fixed for all } x \in C_{j-1} \cap D\right] \leq 7 \cdot \frac{\dim(\overline{D})}{n}.$$

By the union bound,

$$\Pr_{i_1,\ldots,i_t}\left[\text{any of } x_{i_j} \text{ is fixed for } x \in C_{j-1} \cap D, 1 \leq j \leq t\right] \leq 7t \cdot \frac{\dim(\overline{D})}{n} \leq \frac{1}{2}, \tag{8}$$

viewing $i_1, \ldots, i_t$ as random variables that are distributed according to the procedure for generating $X \sim \mu_C^{(t)}$ from Definition 7.

Now let us compare $\mathcal{U}_C(D)$ to $\mu_C^{(t)}(D)$. The former is the probability that $X \in D$ when $X \sim \mathcal{U}_C$, which equals

$$2^{\dim(C \cap D) - \dim(C)}.$$

The latter is the probability that $X \in D$ when $X \sim \mathcal{U}_{C_t}$ with respect to C_t constructed in Definition 7, which equals

$$2^{\dim(C_t \cap D) - \dim(C_t)}, \quad [10]$$

where $[\dim(C_t) = \dim(C) - t]$ always (by the construction). Accordingly, for a tuple $i'_1, \ldots, i'_t \in [n]^t$ it can be the case that

$$\underset{X \sim \mu_C^{(t)}}{\boldsymbol{Pr}} \left[X \in D \middle| i_1 = i'_1, \ldots, i_t = i'_t \right] < \mathcal{U}_C(D)$$

only if at least one of the (consecutive) choices "$i_j = i'_j$" fixes the corresponding coordinate in $x \in C_{j-1} \cap D$. The probability of the latter is, according to (8), at most $1/2$, and therefore

$$\mu_C^{(t)}(D) \geq \frac{1}{2} \cdot \mathcal{U}_C(D),$$

as required.

■*Lemma 2*

4 Unambiguous parity-query complexity of approximate majority

Theorem 1 (*Unambiguous parity-query complexity of* $\widetilde{Maj}$)**.**

$$\overline{R^{\oplus}}(\widetilde{Maj}) \in \Omega\left(\sqrt{n}\right).$$

Obviously, $R^{\oplus}(\widetilde{Maj}) \in O(1)$, so the statement is interesting: As we discussed earlier, the unambiguous regime in the model of parity queries defies some of the "most intuitive" lower-bound approaches, so the proof of Theorem 1 will implement its own ad hoc strategy.

The analysis must distinguish between the cases of relational and functional problems, so we will investigate the behaviour of our f in the region where $\widetilde{Maj}$ would allow uncertainty: that is, on the input values of Hamming weight between $n/3$ and $2n/3$. Intuitively, the transition from the "0"-region to the "1"-region of $\widetilde{Maj}$ is hard to handle for an unambiguous protocol – as opposed to an arbitrary randomised one. An unambiguous protocol has to "adhere to" its own choices of the answers in the uncertainty region, while usual randomised protocols may answer inconsistently there.

As $R^{\oplus}$-protocols of complexity k are convex combinations of $D^{\oplus}$-protocols, which, in turn, partition the input space $\mathcal{GF}_2^n$ into at most 2^k affine subspaces whose elements receive the same answer, our proof will be based technically on bounding the ability of large affine subspaces to discriminate input values, based on their Hamming weight.

Proof of Theorem 1. Let $k(n) \in O\left(\overline{R^{\oplus}}(\widetilde{Maj})\right)$ be the parity-query complexity of unambiguously solving $\widetilde{Maj}$ with error at most $1/20$ and denote by $f : \{0,1\}^n \to \{0,1\}$ some $\widetilde{Maj}$-consistent function of that complexity – that is, f takes value "0" on the inputs of Hamming weight at most $n/3$ and value "1" on the inputs of Hamming weight at least $2n/3$. Assume without loss of generality that $k(n)$ is monotone non-decreasing and

$$\underset{\mathcal{U}_{\{0,1\}^n}}{\boldsymbol{Pr}} \left[f(X) = 0 \right] \geq \frac{1}{2}$$

[10] Let $\dim(\emptyset) \stackrel{\text{def}}{=} -\infty$ (note that affine spaces do not need any "special element", like $\bar{0}$ for linear spaces, so empty affine spaces may be consistently allowed, thus making the whole concept closed under intersection).

(otherwise replace $f(x)$ by $1 - f(\neg x)$, where "$\neg$" stands for the bit-wise negation).

Let $n_0 \stackrel{\text{def}}{=} n$, $C_0 \stackrel{\text{def}}{=} \mathcal{GF}_2^{n_0}$, $f_0 \stackrel{\text{def}}{=} f$ and consider the following recursion, indexed by $j \geq 0$.

We will make sure that for every j throughout the recursion: $n_j \leq n_{j-1}$, $C_j \preceq \mathcal{GF}_2^{n_j}$, $f_j : C_j \to \{0,1\}$ is a sub-function of f (namely, a restriction of f to certain affine subspace of dimension n_j) and $\mathbf{Pr}_{\mathcal{U}_{C_j}}[f_j(X) = 0] \geq 1/2$. Let $t_j \stackrel{\text{def}}{=} \left\lfloor \frac{n_j}{14 \cdot k(n_j)} \right\rfloor$, μ_j be the distribution $\mu_{C_j}^{(t_j)}$ and $\nu_j \stackrel{\text{def}}{=} \frac{\mathcal{U}_{C_j} + \mu_j}{2}$.

As long as $\dim(C_j) \geq 3n_j/4$ and $\sum_{\ell<j} t_\ell < 2n/3$, let Π_j be a parity-query deterministic protocol of complexity at most $k(n_j)$ that computes f_j with error at most $1/20$ with respect to the distribution ν_j. Then it has error at most $1/10$ with respect to both $\mathcal{U}_{C_j}$ and μ_j. As $\mathbf{Pr}_{\mathcal{U}_{C_j}}[f_j(X) = 0] \geq 1/2$, the protocol answers "0" with probability at least $1/2 - 1/10 = 2/5$ with respect to $X \sim \mathcal{U}_{C_j}$.

Denote by $\mathcal{D}_j$ the family of affine subspaces of $\mathcal{GF}_2^{n_j}$ corresponding to the "0"-marked leaves of Π_j ($D \in \mathcal{D}_j$ are pairwise disjoint, each of co-dimension at most $k(n_j)$). Then by Lemma 2:

$$\sum_{D \in \mathcal{D}_j} \mu_j(D) \geq \frac{1}{2} \cdot \sum_{D \in \mathcal{D}_j} \mathcal{U}_{C_j}(D) \geq \frac{1}{5}.$$

Accordingly,

$$\mathop{\mathbf{Pr}}_{X \sim \mu_j}\left[f_j(X) = 1 \middle| X \in \cup_{D \in \mathcal{D}_j} D\right] = \frac{\mathbf{Pr}_{\mu_j}\left[\Pi_j \text{ errs and } X \in \cup_{D \in \mathcal{D}_j} D\right]}{\sum_{D \in \mathcal{D}_j} \mu_j(D)} \leq 5 \cdot \mathop{\mathbf{Pr}}_{\mu_j}\left[\Pi_j \text{ errs}\right] \leq \frac{1}{2}$$

and there must exist some $D_j \in \mathcal{D}_j$, such that $C_j \cap D_j \neq \emptyset$ and

$$\mathop{\mathbf{Pr}}_{X \sim \mu_j}\left[f_j(X) = 1 \middle| X \in D_j\right] \leq \frac{1}{2}. \tag{9}$$

Here again (like in the proof of Lemma 2), let us view $i_1, \ldots, i_{t_j}$ as random variables that accompany the generation of $X \sim \mu_j = \mu_{C_j}^{(t_j)}$ according to Definition 7. Then

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{X \sim \mu_j}\left[f_j(X) = 1 \middle| X \in D_j\right] \\
&\quad = \sum_{i'_1, \ldots, i'_{t_j}} \mathbf{Pr}\left[i_1 = i'_1, \ldots, i_{t_j} = i'_{t_j}\right] \cdot \mathop{\mathbf{Pr}}_{X \sim \mathcal{U}_{C_j}}\left[f_j(X) = 1 \middle| X \in D_j,\, X_{i'_1} = \cdots = X_{i'_{t_j}} = 1\right],
\end{aligned}$$

and therefore for some values $i'_1, \ldots, i'_{t_j} \in [n_j]$ it holds that $\mathbf{Pr}\left[i_1 = i'_1, \ldots, i_{t_j} = i'_{t_j}\right] > 0$ and

$$\mathop{\mathbf{Pr}}_{X \sim \mathcal{U}_{C_j}}\left[f_j(X) = 1 \middle| X \in D_j,\, X_{i'_1} = \cdots = X_{i'_{t_j}} = 1\right] \leq \mathop{\mathbf{Pr}}_{X \sim \mu_j}\left[f_j(X) = 1 \middle| X \in D_j\right] \leq \frac{1}{2}, \tag{10}$$

where the last inequality is (9). These $i'_1, \ldots, i'_{t_j}$ are distinct and every $X_{i'_\ell}$ is unbiased under $X \sim \mathcal{U}_{C_j}$ by the construction of μ_j (Definition 7) and the guarantees of Lemma 1.

Let

$$\begin{aligned}
&L_j \stackrel{\text{def}}{=} \left\{i'_1, \ldots, i'_{t_j}\right\}; \\
&\overline{L_j} \stackrel{\text{def}}{=} [n_j] \setminus L_j; \\
&\widehat{C}_j \stackrel{\text{def}}{=} \left\{x \in C_j \cap D_j \middle| x_{i'_1} = \cdots = x_{i'_{t_j}} = 1\right\}; \\
&n_{j+1} \stackrel{\text{def}}{=} n_j - t_j; \\
&C_{j+1} \stackrel{\text{def}}{=} \left\{x_{\overline{L_j}} \middle| x \in \widehat{C}_j\right\} \text{ (the element-wise projection of } \widehat{C}_j \text{ on } \overline{L_j});
\end{aligned}$$

$$\forall\, x \in \widehat{C}_j :\ f_{j+1}(x_{\overline{L_j}}) \stackrel{\text{def}}{=} f_j(x)\ \ \text{(this defines } f_{j+1} : C_{j+1} \to \{0,1\}).$$

As $C_j \preceq \mathcal{GF}_2^{n_j}$ by the induction hypothesis, $\widehat{C}_j \preceq \mathcal{GF}_2^{n_j}$ too and $C_{j+1} \preceq \mathcal{GF}_2^{n_{j+1}}$. Obviously, f_{j+1} is a sub-function of f_j (and therefore of f) and

$$\underset{X\sim\mathcal{U}_{C_{j+1}}}{\boldsymbol{Pr}}\left[f_{j+1}(X)=0\right] = 1 - \underset{X\sim\mathcal{U}_{C_j}}{\boldsymbol{Pr}}\left[f_j(X)=1 \middle| X\in D_j,\ X_{i'_1}=\cdots=X_{i'_{t_j}}=1\right] \geq \frac{1}{2}, \tag{11}$$

according to (10). So, the conditions that we assumed to hold in the beginning of the j'th iteration will also hold for $j+1$, and thus our recursion can sustain itself.

Let j_0 be the index of the last protocol Π_j considered in the recursion, then it follows from the halting condition that either

$$\dim\left(C_{j_0+1}\right) < \frac{3n_{j_0+1}}{4} \tag{12}$$

or

$$\sum_{\ell=0}^{j_0} t_\ell \geq \frac{2n}{3}. \tag{13}$$

By (11) with respect to round j_0 of the recursion, $\boldsymbol{Pr}_{\mathcal{U}_{C_{j_0+1}}}[f_{j_0+1}(X)=0] \geq 1/2$. On the other hand, the affine subspace C_{j_0+1} is the outcome of a series of restrictions imposed upon $C_0 = \mathcal{GF}_2^{n_0}$, which, in particular, have constrained to "1" the values of $\sum_{\ell=0}^{j_0} t_\ell$ previously unfixed coordinates. Then (13) would imply that $f_{j_0+1} : C_{j_0+1} \to \{0,1\}$ is a restriction of f to input values of Hamming weight at least $2n/3$, and $f_{j_0+1} \equiv 1$ would follow, as f is assumed to be a $\widetilde{Maj}$-consistent function. Accordingly, (12) holds and (13) doesn't, that is, $\sum_{\ell\leq j_0} t_\ell < 2n/3$ and $\dim(C_{j_0+1}) < \frac{3n_{j_0+1}}{4}$.

By construction, $\forall\, j \leq j_0$ it holds that $n_{j+1} = n - \sum_{\ell=0}^{j} t_\ell$ and

$$\begin{aligned}\dim\left(C_{j+1}\right) &\geq \dim\left(C_j\right) - \dim\left(\overline{D_j}\right) - t_j \\ &\geq \dim\left(C_j\right) - k(n_j) - t_j \\ &\geq n - \sum_{\ell=0}^{j} k(n_\ell) - \sum_{\ell=0}^{j} t_\ell = n_{j+1} - \sum_{\ell=0}^{j} k(n_\ell),\end{aligned}$$

and so,

$$\sum_{\ell=0}^{j_0} k(n_\ell) \geq n_{j_0+1} - \dim\left(C_{j_0+1}\right) > \frac{n_{j_0+1}}{4}.$$

On the other hand,

$$n_{j_0+1} = n - \sum_{\ell=0}^{j_0} t_\ell > \frac{n}{3},$$

therefore,

$$\sum_{\ell=0}^{j_0} k(n_\ell) > \frac{n}{12} > \frac{1}{8}\cdot\sum_{\ell=0}^{j_0} t_\ell$$

and there exists $\ell_0 \leq j_0$ such that

$$k(n_{\ell_0}) > \frac{1}{8}\cdot t_{\ell_0} = \frac{1}{8}\cdot\left\lfloor\frac{n_{\ell_0}}{14\cdot k(n_{\ell_0})}\right\rfloor.$$

Therefore, $k(n_{\ell_0}) \in \Omega\left(\sqrt{n_{\ell_0}}\right) = \Omega\left(\sqrt{n}\right)$ and the result follows.

■*Theorem 1*

That is, imposing the restriction of unambiguity can turn a problem that is easy for parity queries with randomness into a hard problem.

Corollary 2.

$$R^{\oplus}(\widetilde{Maj}) \in O(1) \quad \textit{and} \quad \overline{R^{\oplus}}(\widetilde{Maj}),\ D^{\oplus}(\widetilde{Maj}) \in \Omega\left(\sqrt{n}\right).$$

Proof. An $R^{\oplus}$-protocol that queries constant number of bits at random locations can answer $\widetilde{Maj}$ with arbitrarily small constant-bounded error probability.

■*Corollary 2*

It is, in fact, not hard to see that $D^{\oplus}(\widetilde{Maj}) \in \Omega(n)$.

5 Conclusions

We believe that further investigation of the concept of unambiguity is likely to offer a peerless insight into the phenomenon of computational randomness in itself. As discussed in Section 1, restricting one's curiosity to the deterministic and the randomised regimes alone may be insufficient for an adequate understanding of computational randomness.

For the sake of speculation, next we pose two questions related to computational randomness and then use them as a basis for a hypothetical "road map" from the results of this work towards better understanding of randomness.

Efficient deterministic parity-query protocols are partitions of the input space $\mathcal{GF}_2^n$ into a (relatively) small number of same-answer *affine subspaces*; accordingly, a function with an efficient deterministic parity-query protocol must be constant on some large affine subspaces of $\mathcal{GF}_2^n$. Randomised protocols are convex combinations of deterministic ones, so there seems to be no need for a function with an efficient randomised parity-query protocol to be constant on a large affine subspace of $\mathcal{GF}_2^n$. Nevertheless, in all known cases a function over $\mathcal{GF}_2^n$ with an efficient randomised parity-query protocol *is constant over some large affine subspace*.

Question 1 *([KLMY21]). Is every $f : \mathcal{GF}_2^n \to \{0,1\}$ with an efficient randomised parity-query protocol constant on a large affine subspace of $\mathcal{GF}_2^n$?*

Similarly, while every deterministic bipartite protocol with small communication cost corresponds to a partition of the input space $\{0,1\}^n \times \{0,1\}^n$ into a small number of same-answer *combinatorial rectangles*, there seems to be no immediate reason for a function with an efficient randomised protocol to be constant on a large sub-rectangle of $\{0,1\}^n \times \{0,1\}^n$. Nevertheless, all known examples of such functions do have a large rectangle over which they are constant.

Question 2 *([CLV19]). Is every $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ with an efficient randomised communication protocol constant on a large combinatorial rectangle?*

If the answer to Question 1 were affirmative, this would imply some variation of Theorem 1, as it is easy to see that every large affine subspace of $\mathcal{GF}_2^n$ contains both elements of Hamming weight less that $n/3$ and those of Hamming weight more than $2n/3$. Accordingly, this work is a step towards answering Question 1.

Any $R^{\oplus}$-protocol for a function $f : \mathcal{GF}_2^n \to \{0,1\}$ can be emulated by a bipartite randomised communication protocol of roughly the same complexity for the function $F(x,y) \stackrel{\text{def}}{=} f(x \oplus y)$ – obviously, only requires a very special and rather restricted (but well-defined) type of randomised bipartite communication protocols. Any function $F(\cdot,\cdot)$ that can be solved efficiently by one of such restricted protocols necessarily corresponds, in the above sense, to some $f : \mathcal{GF}_2^n \to \{0,1\}$ with an efficient randomised parity-query protocol; the affirmative answer to Question 1 would imply that f is constant on a large affine subspace $C \preceq \mathcal{GF}_2^n$. A large same-answer affine subspace for f corresponds to a large same-answer combinatorial rectangle for F itself:

$$C = C' + c_0 = C' + \left(C' + c_0\right)$$

for any $c_0 \in C$, thus the rectangle $r_C \stackrel{\text{def}}{=} C' \times (C' + c_0)$ satisfies $\left\{x \oplus y \middle| (x,y) \in r_C\right\} = C$ and is monochromatic with respect to F. That is, the affirmative answer to Question 1 could be reinterpreted as affirming a special case of Question 2.

A well-known structural question is this:

Is BPP inside P^{NP} in communication complexity?

Here "*BPP*" stands for the family of all bipartite functions (with product domains) that are efficiently computable by randomised communication protocols and "P^{NP}" is the natural communication-complexity analogue of the corresponding class in computational complexity. It is known [IW10] that every function with an efficient P^{NP}-protocol is constant on a large combinatorial rectangle, so "$BPP \subseteq P^{NP}$" would imply the affirmative answer to Question 2.

And so on...

Acknowledgements

I am indebted to Shalev Ben-David, who first presented to me the thrilling idea of unambiguous complexity, and to Alex Samorodnitsky, who helped me to gain some familiarity with the fanciful realm of binary linear spaces.

References

[CLV19] A. Chattopadhyay, S. Lovett, and M. Vinyals. Equality Alone Does not Simulate Randomness. *Proceedings of the 34th IEEE Conference on Computational Complexity*, pages 1–11, 2019.

[GG11] E. Gat and S. Goldwasser. Probabilistic Search Algorithms with Unique Answers and Their Cryptographic Applications. *Electronic Colloquium on Computational Complexity 18*, 136, 2011.

[GGR13] O. Goldreich, S. Goldwasser, and D. Ron. On the Possibilities and Limitations of Pseudodeterministic Algorithms. *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, pages 127–138, 2013.

[IW10] R. Impagliazzo and R. Williams. Communication complexity with synchronized clocks. *Proceedings of the 25th IEEE Conference on Computational Complexity*, pages 259–269, 2010.

[KLMY21] A. Knop, S. Lovett, S. McGuire, and W. Yuan. Models of Computation Between Decision Trees and Communication. *SIGACT News 52(2)*, pages 46–70, 2021.

[Nis91] N. Nisan. CREW PRAMs and Decision Trees. *SIAM Journal on Computing 20(6)*, pages 999–1007, 1991.

[vN28] J. von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen 100(1)*, pages 295–320, 1928.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il