<table>
<thead>
<tr>
<th>Title</th>
<th>A composition theorem for randomized query complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Anshu, Anurag; Gavinsky, Dmitry; Jain, Rahul; Kundu, Srijita; Lee, Troy; Mukhopadhyay, Priyanka; Santha, Miklos; Sanyal, Swagato</td>
</tr>
<tr>
<td>Date</td>
<td>2018</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10220/46216">http://hdl.handle.net/10220/46216</a></td>
</tr>
<tr>
<td>Rights</td>
<td>© 2018 Anurag Anshu, Dmitry Gavinsky, Rahul Jain, Srijita Kundu, Troy Lee, Priyanka Mukhopadhyay, Miklos Santha, and Swagato Sanyal; licensed under Creative Commons License CC-BY.</td>
</tr>
</tbody>
</table>
A Composition Theorem for Randomized Query Complexity

Anurag Anshu¹, Dmitry Gavinsky¹², Rahul Jain³, Srijita Kundu⁴, Troy Lee⁵, Priyanka Mukhopadhyay⁶, Miklos Santha⁷, and Swagato Sanyal⁸

1 Centre for Quantum Technologies, National University of Singapore, Block S15, 3 Science Drive 2, Singapore 117543
a0109169@u.nus.edu
2 Institute of Mathematics, Czech Academy of Sciences, 115 67 Žitna 25, Praha 1, Czech Republic
3 Centre for Quantum Technologies, National University of Singapore, Block S15, 3 Science Drive 2, Singapore 117543 and MajuLab, UMI 3654, Singapore rahul@comp.nus.edu.sg
4 Centre for Quantum Technologies, National University of Singapore, Block S15, 3 Science Drive 2, Singapore 117543 and MajuLab, UMI 3654, Singapore srijita.kundu@u.nus.edu
5 SPMS, Nanyang Technological University, 21 Nanyang Link, Singapore 637371 and Centre for Quantum Technologies, National University of Singapore, Block S15, 3 Science Drive 2, Singapore 117543 troyjlee@gmail.com
6 Centre for Quantum Technologies, National University of Singapore, Block S15, 3 Science Drive 2, Singapore 117543 and MajuLab, UMI 3654, Singapore a0109168@u.nus.edu
7 IRIF, Université Paris Diderot, CNRS, 75205 Paris, France and Centre for Quantum Technologies, National University of Singapore, Block S15, 3 Science Drive 2, Singapore 117543 santha@irif.fr
8 SPMS, Nanyang Technological University, 21 Nanyang Link, Singapore 637371 and Centre for Quantum Technologies, National University of Singapore, Block S15, 3 Science Drive 2, Singapore 117543 ssanyal@ntu.edu.sg

Abstract

Let the randomized query complexity of a relation for error probability \( \epsilon \) be denoted by \( R(\cdot) \). We prove that for any relation \( f \subseteq \{0,1\}^n \times R \) and Boolean function \( g : \{0,1\}^m \rightarrow \{0,1\} \),
\[
R_{1/3}(f \circ g^n) = \Omega(R_{4/9}(f) \cdot R_{1/2-1/n}(g)),
\]
where \( f \circ g^n \) is the relation obtained by composing \( f \) and \( g \). We also show using an XOR lemma that
\[
R_{1/3}(f \circ (g_{O(\log n)}^{\oplus})) = \Omega(\log n \cdot R_{4/9}(f) \cdot R_{1/3}(g)),
\]
where \( g_{O(\log n)}^{\oplus} \) is the function obtained by composing the XOR function on \( O(\log n) \) bits and \( g \).

1998 ACM Subject Classification F.1.1 Models of Computation, F.1.3 Complexity Measures and Classes

* This work was partially supported by the National Research Foundation, including under NRF RF Award No. NRF-NRF2013-13, the Prime Minister’s Office, Singapore and the Ministry of Education, Singapore under the Research Centres of Excellence programme and by Grant No. MOE2012-T3-1-009.
† D.G. is partially funded by the grant P202/12/G061 of GA ČR and by RVO: 67985840.
‡ M. S. is partially funded by the ANR Blanc program under contract ANR-12-BS02-005 (RDAM project).

© Anurag Anshu, Dmitry Gavinsky, Rahul Jain, Srijita Kundu, Troy Lee, Priyanka Mukhopadhyay, Miklos Santha, and Swagato Sanyal; licensed under Creative Commons License CC-BY

Editors: Satya Lokam and R. Ramanujam; Article No. 10; pp. 10:1–10:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
Keywords and phrases  Query algorithms and complexity, Decision trees, Composition theorem, XOR lemma, Hardness amplification

Digital Object Identifier  10.4230/LIPIcs.FSTTCS.2017.10

1 Introduction

Given two Boolean functions $f : \{0, 1\}^n \to \{0, 1\}$ and $g : \{0, 1\}^m \to \{0, 1\}$, the composed function $f \circ g^n : \{(0, 1)^m\}^n \to \{0, 1\}$ is defined as follows: For $x = (x^{(1)}, \ldots, x^{(n)}) \in \{(0, 1)^m\}^n$, $f \circ g^n(x) = f(g(x^{(1)}), \ldots, g(x^{(n)}))$. Composition of Boolean functions has long been a topic of active research in complexity theory. In many works, composition of Boolean function is studied in the context of a certain complexity measure. The objective is to understand the relation between the complexity of the composed function in terms of the complexities of the individual functions. Let $D(\cdot)$ denote the deterministic query complexity. It is easy to see that $D(f \circ g^n) \leq D(f) \cdot D(g)$ since $f \circ g$ can be computed by simulating an optimal query algorithm of $f$; whenever the algorithm makes a query, we simulate an optimal query algorithm of $g$ and serve the query. It can be shown by an adversary argument that this is an optimal query algorithm and $D(f \circ g^n) = D(f) \cdot D(g)$ [11, 15].

However, such a characterization is not so obvious for randomized query complexity. Although a similar upper bound still holds true (possibly accommodating a logarithmic overhead), it is no more as clear that it also asymptotically bounds the randomized query complexity of $f \circ g^n$ from below. Let $R_{\epsilon}(\cdot)$ denote the $\epsilon$-error randomized query complexity. Our main theorem in this work is the following.

**Theorem 1 (Main Theorem).** For any relation $f \subseteq \{0, 1\}^n \times \mathcal{R}$ and Boolean function $g : \{0, 1\}^m \to \{0, 1\}$,

$$R_{1/3}(f \circ g^n) = \Omega(R_{4/9}(f) \cdot R_{1/2-1/n^4}(g)).$$

See Section 2 for definitions of composition and various complexity measures of relations. Theorem 1 implies that if $g$ is a function that is hard to compute with error $1/2 - 1/n^4$, $f \circ g^n$ is hard to compute with error $1/3$.

In the special case where $f$ is a function, Theorem 1 implies that $R_{1/3}(f \circ g^n) = \Omega(R_{1/3}(f) \cdot R_{1/2-1/n^4}(g))$, since the success probability of query algorithms for functions can be boosted from $5/9$ to $2/3$ by constantly many independent repetitions followed by taking a majority of the different outputs.

Theorem 1 is useful only when the function $g$ is hard against randomized query algorithms even for error $1/2 - 1/n^4$. In Section 3.1 we prove the following consequence of Theorem 1.

**Theorem 2.** Let $f \subseteq \{0, 1\}^n \times \mathcal{R}$ be any relation. Let $g : \{0, 1\}^m \to \{0, 1\}$ be a function. Let $g^{\oplus}_i : \{(0, 1)^m\}^i \to \{0, 1\}$ be defined as follows: for $x = (x^{(1)}, \ldots, x^{(i)}) \in \{(0, 1)^m\}^i$, $g^{\oplus}_i(x) = \oplus_{i=1}^i g(x^{(i)})$. Then,

$$R_{1/3} \left( f \circ \left( g^{\oplus}_{O(\log n)} \right)^n \right) = \Omega(\log n \cdot R_{4/9}(f) \cdot R_{1/3}(g)).$$

Theorem 2 is proved by establishing, via an XOR lemma by Andrew Drucker [5], that if $g$ is hard for error $1/3$ then $g^{\oplus}_{O(\log n)}$ is hard for error $1/2 - 1/n^4$. 
Composition theorems for randomized query complexity have been extensively studied in the past. Göös and Jayram [6] showed a composition theorem for a constrained version of conical junta degree, which is a lower bound on randomized query complexity. Composition theorem for approximate degree (which also bounds randomized query complexity from below) for the special case of TRIBES function has seen a long line of research culminating in independent works of Sherstov [14] and Bun and Thaler [3] who settle the question by proving optimal bounds.

Composition theorem has been studied and shown in the context of communication and query complexities by the works of Göös, Pitassi and Watson [7, 8], Chattopadhyay et al. [4] when the function \(g\) is the indexing function or the inner product function with large enough arity. The work of Hatami, Hosseini and Lovett [9] proves a composition theorem in the context of communication and parity query complexities when the function \(g\) is the two-bit XOR function. Ben-David and Kothari [1] proved a composition theorem for the sabotage complexity of Boolean functions, a novel complexity measure defined in the same work that the authors prove to give quadratically tight bound on the randomized query complexity.

Composition theorems have also been successfully used in the past in constructing separating examples for various complexity measures, and bounding one complexity measure in terms of another. Kulkarni and Tal [10] proved an upper bound on fractional block sensitivity in terms of degree by analyzing the behavior of fractional block sensitivity under function composition. Separation between block sensitivity and degree was obtained by composing Kushilevitz’s hemi-icosahedron function repeatedly with itself [13, 12, 2]. Separation between parity decision tree complexity and Fourier sparsity has been obtained by O’Donnell et al. by studying the behavior of parity kill number under function composition [13].

### 1.1 Our techniques

In this section, we give a high level overview of our proof of Theorem 1. We refer the reader to Section 2 for formal definitions of composition and various complexity measures of relations.

Let \(\epsilon = 1/2 - 1/n^4\). Let \(\mu\) be the distribution over the domain \(\{0,1\}^m\) of \(g\) for which \(R_e(g)\) is achieved, i.e., \(R_e(g) = D^\mu(g)\) (see Fact 3). For \(b \in \{0,1\}\), let \(\mu_b\) denote the distribution obtained by conditioning \(\mu\) to the event that \(g(x) = b\) (see Section 2 for a formal definition).

We show that for every probability distribution \(\lambda\) over the domain \(\{0,1\}^n\) of \(f\), there exists a deterministic query algorithm \(A\) with worst case query complexity at most \(R_{1/\lambda}(f \circ g^n)/R_e(g)\), such that \(\Pr_{z \sim \lambda}[(z, A(z)) \in f] \geq 5/9\). By the minimax principle (Fact 3) this proves Theorem 1.

Now using the distribution \(\lambda\) over \(\{0,1\}^n\) we define a probability distribution \(\gamma\) over \((\{0,1\}^m)^n\). To define \(\gamma\), we begin by defining a family of distributions \(\{\gamma^z : z \in \{0,1\}^n\}\) over \((\{0,1\}^m)^n\). For a fixed \(z = (z_1, \ldots, z_n) \in \{0,1\}^n\), we define \(\gamma^z\) by giving a sampling procedure:

1. For each \(i = 1, \ldots, n\), sample \(x^{(i)} = (x_1^{(i)}, \ldots, x_m^{(i)})\) from \(\{0,1\}^m\) independently according to \(\mu_{z_i}\).
2. Return \(x = (x^{(1)}, \ldots, x^{(n)})\).

Thus for \(z = (z_1, \ldots, z_n) \in \{0,1\}^n\) and \(x = (x^{(1)}, \ldots, x^{(n)}) \in (\{0,1\}^m)^n\), \(\gamma^z(x) = \Pi_{i=1}^n \mu_{z_i}(x^{(i)})\). Note that \(\gamma^z\) is supported only on strings \(x\) for which the following is true: for each \(r \in R\), \((x, r) \in f \circ g^n\) if and only if \((z, r) \in f\).
Having defined the distributions $\gamma^z$, we define the distribution $\gamma$ by giving a sampling procedure:

1. Sample a $z = (z_1, \ldots, z_n)$ from $\{0,1\}^n$ according to $\lambda$.
2. Sample an $x = (x^{(1)}, \ldots, x^{(n)})$ from $\{(0,1)^m\}^n$ according to $\gamma^z$. Return $x$.

By minimax principle (Fact 3), there is a deterministic query algorithm $B$ of worst case complexity at most $R_{1/3}(f \circ g^n)$ such that $Pr_{x \sim \gamma}(f(x),B(x) \in f \circ g^n) \geq 2/3$. We will use $B$ to construct a randomized query algorithm $A'$ for $f$ with the desired properties. A deterministic query algorithm $A$ for $f$ with required performance guarantees can then be obtained by appropriately fixing the randomness of $A'$.

See Algorithm 1 for a formal description of $A'$. Given an input $z = (z_1, \ldots, z_n), A'$ simulates $B$. Recall that an input to $B$ is an $nm$ bit long string $(x_j^{(i)})_{j=1,\ldots,m}$. Whenever $B$ asks for (queries) an input bit $x_j^{(i)}$, a response bit is appropriately generated and passed to $B$. To generate a response to a query by $B$, a bit in $z$ may be queried; those queries will contribute to the query complexity of $A'$. The queries are addressed as follows. Let the simulation of $B$ request bit $x_j^{(i)}$.

- If less than $D^\mu_i(g)$ queries have been made into $x^{(i)}$ (including the current query) then a bit $b$ is sampled from the marginal distribution of $x_j^{(i)}$ according to $\mu$, conditioned on the responses to the past queries. $b$ is passed to the simulation of $B$.
- If $D^\mu_i(g)$ queries have been made into $x^{(i)}$ (including the current query) then first the input bit $z_i$ is queried; then a bit $b$ is sampled from the marginal distribution of $x_j^{(i)}$ according to $\mu_{z_i}$, conditioned on the responses to the past queries. $b$ is passed to the simulation of $B$.

The simulation of $B$ continues until $B$ terminates in a leaf. Then $A'$ also terminates and outputs the label of the leaf.

We use Claims 8 and 9 to prove that for a fixed $z \in \{0,1\}^n$, the probability distribution induced by $A'$ on the leaves of $B$ is statistically close to the probability distribution induced by $B$ on its leaves for a random input from $\gamma^z$. Averaging over different $z$'s, the correctness of $A'$ follows from the correctness of $B$. The reader is referred to Section 3 for the details.

## 2 Preliminaries

In this section, we define some basic concepts, and set up our notations. We begin with defining the 2-sided error randomized and distributional query complexity measures of relations. The relations considered in this work will all be between the Boolean hypercube $\{0,1\}^k$ of some dimension $k$, and an arbitrary set $S$. The strings $x \in \{0,1\}^n$ will be called as inputs to the relation, and $\{0,1\}^n$ will be referred to as the input space and the domain of relations.

**Definition 1** (2-sided Error Randomized Query Complexity). Let $S$ be any set. Let $h \subseteq \{0,1\}^k \times S$ be any relation and $\epsilon \in [0,1/2)$. The 2-sided error randomized query complexity $R_\epsilon(h)$ is the minimum number of queries made in the worst case by a randomized query algorithm $A$ (the worst case is over inputs and the internal randomness of $A$) that on each input $x \in \{0,1\}^k$ satisfies $Pr[(x,A(x)) \in h] \geq 1 - \epsilon$ (where the probability is over the internal randomness of $A$).
Definition 2 (Distributional Query Complexity). Let \( h \subseteq \{0,1\}^k \times \mathcal{S} \) be any relation, \( \mu \) a distribution on the input space \( \{0,1\}^k \) of \( h \), and \( \epsilon \in [0,1/2) \). The distributional query complexity \( D^\epsilon(h) \) is the minimum number of queries made in the worst case (over inputs) by a deterministic query algorithm \( A \) for which \( \Pr_{x \sim \mu}(x, A(x)) \subseteq h \geq 1 - \epsilon \).

In particular, if \( h \) is a function and \( A \) is a randomized or distributional query algorithm computing \( h \) with error \( \epsilon \), then \( \Pr[h(x) = A(x)] \geq 1 - \epsilon \), where the probability is over the respective sources of randomness.

The following theorem is von Neumann’s minimax principle stated for decision trees.

Fact 3 (Minimax principle). For any integer \( k \), set \( \mathcal{S} \), and relation \( h \subseteq \{0,1\}^k \times \mathcal{S} \),
\[
R_k(h) = \max_{\mu} \mathbb{D}^\epsilon(h).
\]

Let \( g : \{0,1\}^m \rightarrow \{0,1\} \) be a Boolean function. Let \( \mu \) be a probability distribution on \( \{0,1\}^m \) which intersects non-trivially both with \( g^{-1}(0) \) and with \( g^{-1}(1) \). For each \( z \in \{0,1\} \), let \( \mu_z \) be the distribution obtained by restricting \( \mu \) to \( g^{-1}(z) \). Formally,
\[
\mu_z(x) = \begin{cases} 
0 & \text{if } g(x) \neq z \\
\frac{\mu(x)}{\mu \left( g^{-1}(z) \right)} & \text{if } g(x) = z
\end{cases}
\]

Notice that \( \mu_0 \) and \( \mu_1 \) are defined with respect to some Boolean function \( g \), which will always be clear from the context.

Definition 4 (Subcube, Co-dimension). A subset \( C \) of \( \{0,1\}^m \) is called a subcube if there exists a set \( S \subseteq \{1,\ldots,m\} \) of indices and an assignment function \( A : S \rightarrow \{0,1\} \) such that \( C = \{ x \in \{0,1\}^m : \forall i \in S, x_i = A(i) \} \). The co-dimension \( \text{codim}(C) \) of \( C \) is defined to be \( |S| \).

Let \( C \subseteq \{0,1\}^m \) be a subcube and \( \mu \) be a probability distribution on \( \{0,1\}^m \). We will often abuse notation and use \( C \) to denote the event that a random string \( x \) belongs to the subcube \( C \). The probability \( \Pr_{x \sim \mu}[x \in C] \) will be denoted by \( \Pr_{\mu}[C] \). For subcubes \( C_1 \) and \( C_2 \), the conditional probability \( \Pr_{x \sim \mu}[x \in C_2 | x \in C_1] \) will be denoted by \( \Pr_{\mu}[C_2 | C_1] \).

Definition 5 (Bias of a subcube). Let \( g : \{0,1\}^m \rightarrow \{0,1\} \) be a Boolean function. Let \( \mu \) be a probability distribution over \( \{0,1\}^m \). Let \( C \subseteq \{0,1\}^m \) be a subcube such that \( \Pr_{\mu}[C] > 0 \). The bias of \( C \) with respect to \( \mu \), \( \text{bias}^\mu(C) \), is defined to be:
\[
\text{bias}^\mu(C) = \left| \Pr_{x \sim \mu}[g(x) = 0 | x \in C] - \Pr_{x \sim \mu}[g(x) = 1 | x \in C] \right|
\]

A Boolean function \( g \) is implicit in the definition of bias, which will always be clear from the context.

Proposition 6. Let \( g : \{0,1\}^m \rightarrow \{0,1\} \) be a Boolean function, and \( \mathbb{D}^\epsilon(g) > 0 \). Then,
\[
\min_{b \in \{0,1\}} \{ \Pr_{x \sim \mu}[g(x) = b] \} > \epsilon.
\]

In particular, \( \text{bias}^\mu(\{0,1\}^m) < 1 - 2\epsilon \).

Proof. Towards a contradiction, assume that \( \min_{b \in \{0,1\}} \{ \Pr_{x \sim \mu}[g(x) = b] \} \leq \epsilon \). Then, the algorithm that outputs \( \arg \max_{b \in \{0,1\}} \{ \Pr_{x \sim \mu}[g(x) = b] \} \) makes 0 query and is correct with probability at least \( 1 - \epsilon \). This contradicts the hypothesis that \( \mathbb{D}^\epsilon(g) > 0 \).
Now we define composition of two relations.

**Definition 7** (Composition of relations). Let \( f \subseteq \{0,1\}^n \times \mathcal{R} \) and \( g \subseteq \{0,1\}^m \times \{0,1\} \) be two relations. The composed relation \( f \circ g^\mu \subseteq \{(0,1)^m\}^n \times \mathcal{R} \) is defined as follows: For \( x = (x^{(1)}, \ldots, x^{(n)}) \in \{(0,1)^m\}^n \) and \( r \in \mathcal{R} \), \((x, r) \in f \circ g^\mu \) if and only if there exists \( b = (b^{(1)}, \ldots, b^{(m)}) \in \{0,1\}^m \) such that for each \( i = 1, \ldots, n \), \((x^{(i)}, b^{(i)}) \in g \) and \((b, r) \in f \).

We will often view a deterministic query algorithm as a binary decision tree. In each vertex \( v \) of the tree, an input variable is queried. Depending on the outcome of the query, the computation goes to a child of \( v \). The child of \( v \) corresponding to outcome \( b \) to the query made is denoted by \( v_b \). It is well known that the set of inputs that lead the computation of a decision tree to a certain vertex forms a subcube. We will denote the subcube corresponding to a vertex \( v \) by \( C_v \).

We next prove two claims about bias, probability and co-dimension of subcubes that will be useful. Claim 8 states that for a function with large distributional query complexity, the bias of most shallow leaves of any deterministic query procedure is small.

**Claim 8.** Let \( g : \{0,1\}^m \rightarrow \{0,1\} \) be a Boolean function. Let \( \epsilon \in [1/4, 1/2) \) and let \( \delta = 1/2 - \epsilon \). Let \( \mu \) be a probability distribution on \( \{0,1\}^m \), and \( D_y^\mu(g) = c > 0 \). Let \( \mathcal{B} \) be any deterministic query algorithm for strings in \( \{0,1\}^m \). For each \( y \in \{0,1\}^m \), let \( \ell_y \) be the unique leaf of \( \mathcal{B} \) that contains \( y \). Then,

\[
\text{(a) } \Pr_{y \sim \mu}[\text{codim}(\ell_y) < c \text{ and } \text{bias}^\mu(\ell_y) \geq 2\delta^{1/2}] < \delta^{1/2}.
\]

\[
\text{(b) For each } b \in \{0,1\}, \Pr_{y \sim \mu_a}[\text{codim}(\ell_y) < c \text{ and } \text{bias}^\mu(\ell_y) \geq 2\delta^{1/2}] < 4\delta^{1/2}.
\]

In the above claim \( \mathcal{B} \) could just be a deterministic procedure that makes queries and eventually terminates; whether or not it makes any output upon termination is not of any consequence here.

**Proof.** We first show that part (a) implies part (b). To this end, assume part (a) and fix a \( b \in \{0,1\} \). Let \( a(y) \) be the indicator variable for the event \( \text{codim}(\ell_y) < c \) and \( \text{bias}^\mu(\ell_y) \geq 2\delta^{1/2} \). Thus, part (a) states that \( \Pr_{y \sim \mu}[a(y) = 1] < \delta^{1/2} \). Now,

\[
\Pr_{y \sim \mu}[\text{codim}(\ell_y) < c \text{ and } \text{bias}^\mu(\ell_y) \geq 2\delta^{1/2}]
= \sum_{y : a(y) = 1} \mu_b(y)
= \sum_{y : a(y) = 1} \frac{1}{\Pr_{y \sim \mu}[a(y) = 1]} \mu(y) \quad \text{(From the definition of } \mu_b)\n< \frac{1}{\epsilon} \Pr_{y \sim \mu}[a(y) = 1] \quad \text{(From Proposition 6)}\n< 4\delta^{1/2}. \quad \text{(By the hypothesis } \epsilon \geq 1/4 \text{ and part (a)})
\]

We now prove part (a). Towards a contradiction assume that

\[
\Pr_{y \sim \mu}[\text{codim}(\ell_y) < c \text{ and } \text{bias}^\mu(\ell_y) \geq 2\delta^{1/2}] \geq \delta^{1/2}.
\]

Now consider the following query algorithm \( \mathcal{A} \) on \( m \) bit strings:

Begin simulating \( \mathcal{B} \). Let \( C \) be the subcube associated with the current node of \( \mathcal{B} \) in the simulation. Simulate \( \mathcal{B} \) unless one of the following happens.
The next claim states that if a subcube has low bias with respect to a distribution 

\[ \text{bias}^\mu(C) \geq 2\delta^{1/2}. \]

Upon termination, if \( \text{bias}^\mu(C) \geq 2\delta^{1/2} \), output \( \max_{b \in \{0,1\}} \Pr_{y \sim \mu}[g(y) = b \mid y \in C] \). Else output a uniformly random bit.

It immediately follows that the worst case query complexity of \( A \) is at most \( c - 1 \). Now, we will prove that \( \Pr_{y \sim \mu}[A(y) = g(y)] \geq 1 - \epsilon \). This will contradict the hypothesis that \( D^\mu_p(g) = c \).

Let \( L \) be the node of \( B \) at which the computation of \( A \) ends. Let \( \Pr_{y \sim \mu}[\text{bias}^\mu(L) \geq 2\delta^{1/2}] = p \).

By our assumption, the probability (over \( \mu \)) that \( L \) is a leaf and \( \text{bias}^\mu(L) \geq 2\delta^{1/2} \) is at least \( \delta^{1/2} \); in particular \( p \geq \delta^{1/2} \). Now,

\[
\Pr_{y \sim \mu}[A(y) = g(y)] = \Pr_{y \sim \mu}[\text{bias}^\mu(L) \geq 2\delta^{1/2}] \cdot \Pr_{y \sim \mu}[A(y) = g(y) \mid \text{bias}^\mu(L) \geq 2\delta^{1/2}] + \Pr_{y \sim \mu}[\text{bias}^\mu(L) < 2\delta^{1/2}] \cdot \Pr_{y \sim \mu}[A(y) = g(y) \mid \text{bias}^\mu(L) < 2\delta^{1/2}]
\]

\[
\geq p \cdot (1/2 + \delta^{1/2}) + (1 - p) \cdot \frac{1}{2} \quad \text{(from our assumption)}
\]

\[
= 1/2 + p \cdot \delta^{1/2}
\]

\[
\geq 1/2 + \delta \quad \text{(since } p \geq \delta^{1/2} \text{)}
\]

\[
= 1 - \epsilon.
\]

This completes the proof. ▶

The next claim states that if a subcube has low bias with respect to a distribution \( \mu \), then the distributions \( \mu_0 \) and \( \mu_1 \) ascribe almost the same probability to it.

**Claim 9.** Let \( g : \{0,1\}^m \rightarrow \{0,1\} \) be a Boolean function and \( \delta \in (0, \frac{1}{2}) \). Let \( \mu \) be a distribution on \( \{0,1\}^m \). Let \( C \) be a subcube such that \( \Pr_{\mu}[C] > 0 \) and \( \text{bias}^\mu(C) \leq \delta \). Also assume that \( \text{bias}^\mu(\{0,1\}^m) \leq \delta \). Then for any \( b \in \{0,1\} \) we have,

(a) \( \Pr_{\mu}[C] \leq (1 + 4\delta) \cdot \Pr_{\mu_b}[C] \),

(b) \( \Pr_{\mu}[C] \geq (1 - 4\delta) \cdot \Pr_{\mu_b}[C] \).

**Proof.** We prove part (a) of the claim. The proof of part (b) is similar.

By the definition of \( \text{bias} \) and the hypothesis, for each \( b \in \{0,1\} \),

\[
\sum_{y \in \mathcal{H}_m, g(y) = b} \mu(y) \leq \left( \frac{1}{2} + \frac{\delta}{2} \right) \cdot \sum_{y \in \mathcal{H}_m} \mu(y) = \frac{1}{2} + \frac{\delta}{2},
\]

\[
\sum_{y \in C, g(y) = b} \mu(y) \geq \left( \frac{1}{2} - \frac{\delta}{2} \right) \cdot \sum_{y \in C} \mu(y) > 0.
\]

Now,

\[
\Pr_{\mu_b}[C] = \sum_{y \in C} \mu_b(y)
\]

\[
= \frac{\sum_{y \in \mathcal{H}_m, g(y) = b} \mu(y)}{\sum_{y \in \mathcal{H}_m} \mu(y)}
\]

\[
= \frac{\sum_{y \in \mathcal{H}_m, g(y) = b} \mu(y)}{\sum_{y \in \mathcal{H}_m} \mu(y)}
\]

\[
\geq (1 - 4\delta) \cdot \Pr_{\mu_b}[C].
\]
\[ \geq \frac{(1/2 - \delta/2) \cdot \sum_{y \in C} \mu(y)}{1/2 + \delta/2} \]  
(From Equations (1) and (2))

\[ = \frac{1/2 - \delta/2}{1/2 + \delta/2} \cdot \Pr_{\mu}[C] \]

Thus,

\[ \Pr_{\mu}[C] \leq \frac{1/2 + \delta/2}{1/2 - \delta/2} \cdot \Pr_{\mu}[C] \leq (1 + 4\delta) \cdot \Pr_{\mu}[C]. \]

(see Equations (1) and (2)

### 3 Composition Theorem

In this section we prove our main theorem. We restate it below.

**Theorem 1** (Main Theorem). For any relation \( f \subseteq \{0,1\}^n \times \mathcal{R} \) and Boolean function \( g: \{0,1\}^m \rightarrow \{0,1\} \),

\[ R_{1/3}(f \circ g^n) = \Omega(R_{4/9}(f) \cdot R_{1/2-1/n^4}(g)). \]

**Proof.** We begin by recalling the notations defined in Section 1.1 that we will use in this proof.

Let \( \epsilon = 1/2 - 1/n^4 \). Let \( \mu \) be the distribution over the domain \( \{0,1\}^m \) of \( g \) for which \( R_{\epsilon}(g) \) is achieved, i.e., \( R_{\epsilon}(g) = D_\mu^n(g) \). (see Fact 3)

We show that for every probability distribution \( \lambda \) over the input space \( \{0,1\}^n \) of \( f \), there exists a deterministic query algorithm \( A \) with worst case query complexity at most \( R_{1/3}(f \circ g^n)/R_{\epsilon}(g) \), such that \( \Pr_{z \sim \lambda}[\langle z, A(z) \rangle \in f] \geq 5/9 \). By the minimax principle (Fact 3) this will prove Theorem 1.

Using \( \lambda \), we define a probability distribution \( \gamma \) over \( \{0,1\}^n \). We first define a family of distributions \( \{\gamma_z^i : z \in \{0,1\}^n\} \) over \( \{0,1\}^n \). For a fixed \( z \in \{0,1\}^n \), we define \( \gamma_z^i \) by giving a sampling procedure:

1. For each \( i = 1, \ldots, n \), sample \( x^{(i)} = (x_1^{(i)}, \ldots, x_m^{(i)}) \) from \( \{0,1\}^m \) independently according to \( \mu_{x_i} \).
2. Return \( x = (x^{(1)}, \ldots, x^{(n)}) \).

Thus for \( z = (z_1, \ldots, z_n) \in \{0,1\}^n \) and \( x = (x^{(1)}, \ldots, x^{(n)}) \in \{0,1\}^n \), \( \gamma_z(x) = \Pi_{i=1}^n \mu_{z_i}(x^{(i)}) \). Note that \( \gamma_z \) is supported only on strings \( x \) for which the following is true: for each \( r \in \mathcal{R} \), \( \langle x, r \rangle \in f \circ g^n \) if and only if \( \langle z, r \rangle \in f \).

Now, we define the distribution \( \gamma \) by giving a sampling procedure:

1. Sample a \( z = (z_1, \ldots, z_n) \) from \( \{0,1\}^n \) according to \( \lambda \).
2. Sample an \( x = (x^{(1)}, \ldots, x^{(n)}) \) from \( \{0,1\}^n \) according to \( \gamma_z \). Return \( x \).

By the minimax principle (Fact 3), there is a deterministic query algorithm \( B \) of worst case complexity at most \( R_{1/3}(f \circ g^n) \) such that \( \Pr_{x \sim \gamma}[\langle x, B(x) \rangle \in f \circ g^n] \geq 2/3 \). We will use \( B \) to construct a randomized query algorithm \( A' \) for \( f \) with the desired properties. A deterministic query algorithm \( A \) for \( f \) with required performance guarantees can then be obtained by appropriately fixing the randomness of \( A' \). Algorithm 1 formally defines the algorithm \( A' \) that we construct.
We are left with the task of bounding the error of internal randomness of the computation of \( \mathcal{A}' \). Now, consider a run of \( \mathcal{A}' \). For each \( \ell \in \mathcal{L} \) let the corresponding subcube be \( C_{v_\ell} \), where \( C_{v_\ell} \) is a subcube of the domain of the \( i \)-th copy of \( g \) (corresponding to the input \( x^{(i)} \)). Recall from Section 2 that for \( b \in \{0,1\} \), \( v_b \) denotes the \( b \)-th child of \( v \).

For each leaf \( \ell \in \mathcal{L} \) and \( i = 1, \ldots, n \), define \( \text{snip}^{(i)}(\ell) \) to be 1 if there is a node \( t \) in the unique path from the root of \( \mathcal{B} \) to \( \ell \) such that \( \text{codim}(C_t^{(i)}) < D_{\ell}^{(i)}(g) \) and \( \text{bias}^{(i)}(C_t^{(i)}) \geq \frac{2^n}{\epsilon} \), and 0 otherwise. Define \( \text{snip}(\ell) := \bigvee_{i=1}^n \text{snip}^{(i)}(\ell) \).

For each \( \ell \in \mathcal{L} \), define \( p_\ell^z \) to be the probability that for an input drawn from \( \gamma^z \), the computation of \( \mathcal{B} \) terminates at leaf \( \ell \). We have,

\[
P_{z \sim \gamma^z} [(x, \mathcal{B}(x)) \in f \circ g^n] = \sum_{\ell \in \mathcal{L}(z, b_\ell) \in f} p_\ell^z. \tag{3}
\]

From our assumption about \( \mathcal{B} \) we also have that,

\[
P_{z \sim \gamma^z} [(x, \mathcal{B}(x)) \in f \circ g^n] = \mathbb{E}_{z \sim \lambda} \mathbb{E}_{z \sim \gamma^z} [P_{z \sim \gamma^z} [(x, \mathcal{B}(x)) \in f \circ g^n] \geq \frac{2}{3}. \tag{4}
\]

Now, consider a run of \( \mathcal{A}' \) on \( z \). For each \( \ell \in \mathcal{L} \) of \( \mathcal{B} \), define \( q_\ell^z \) to be the probability that the computation of \( \mathcal{A}' \) on \( z \) terminates at leaf \( \ell \) of \( \mathcal{B} \). Note that the probability is over the internal randomness of \( \mathcal{A}' \).

\[\]
To finish the proof, we need the following two claims. The first one states that the leaves $\ell \in \mathcal{L}$ are sampled with similar probabilities by $\mathcal{B}$ and $\mathcal{A'}$.

**Claim 10.** For each $\ell \in \mathcal{L}$ such that $\text{snip}(\ell) = 0$, and for each $z \in \{0, 1\}^n$, $\frac{8}{9} p^\ell_z \leq q^\ell_z \leq \frac{10}{9} p^\ell_z$.

The next Claim states that for each $z$, the probability according to $\gamma^z$ of the leaves $\ell$ for which $\text{snip}(\ell) = 1$ is small.

**Claim 11.** \(\forall z \in \{0, 1\}^n, \sum_{\ell \in \mathcal{L}, \text{snip}(\ell) = 1} p^\ell_z \leq \frac{4}{n}\).

We first finish the proof of Theorem 1 assuming Claims 10 and 11, and then prove the claims.

For a fixed input $z \in \{0, 1\}^n$, the probability that $\mathcal{A'}$, when run on $z$, outputs an $r$ such that $(z, r) \in f$, is at least

\[
\sum_{\ell \in \mathcal{L}, (z, b^\ell_z) \in f, \text{snip}(\ell) = 0} q^\ell_z \geq \sum_{\ell \in \mathcal{L}, (z, b^\ell_z) \in f, \text{snip}(\ell) = 0} \frac{8}{9} p^\ell_z \quad (\text{By Claim 10})
\]

\[
= \frac{8}{9} \left( \sum_{\ell \in \mathcal{L}, (z, b^\ell_z) \in f} p^\ell_z - \sum_{\ell \in \mathcal{L}, (z, b^\ell_z) \in f, \text{snip}(\ell) = 1} p^\ell_z \right) \geq \frac{8}{9} \left( \sum_{\ell \in \mathcal{L}, (z, b^\ell_z) \in f} p^\ell_z - \frac{4}{n} \right) \quad (\text{By Claim 11})
\]

Thus, the success probability of $\mathcal{A'}$ is at least

\[
\mathbb{E}_{z \sim \lambda} \sum_{\ell \in \mathcal{L}, (z, b^\ell_z) \in f, \text{snip}(\ell) = 0} q^\ell_z \geq \frac{8}{9} \left( \mathbb{E}_{z \sim \lambda} \sum_{\ell \in \mathcal{L}, (z, b^\ell_z) \in f} p^\ell_z - \frac{4}{n} \right) \quad (\text{By Equation (5)})
\]

\[
\geq \frac{8}{9} \left( \frac{2}{3} - \frac{4}{n} \right) \quad (\text{By Equations (3) and (4)})
\]

\[
\geq \frac{5}{9} \quad (\text{For large enough } n)
\]

We now give the proofs of Claims 10 and 11.

**Proof of Claim 10.** We will prove the first inequality. The proof of the second inequality is similar.¹

Fix a $z \in \{0, 1\}^n$ and a leaf $\ell \in \mathcal{L}$ such that $\text{snip}(\ell) = 0$. For each $i = 1, \ldots, n$, assume that $\text{codim}(c^{(i)}_\ell) = d^{(i)}$, and in the path from the root of $\mathcal{B}$ to $\ell$ the variables $x_1^{(i)}, \ldots, x_{d^{(i)}}^{(i)}$ are set to bits $b_1^{(i)}, \ldots, b_{d^{(i)}}^{(i)}$ in this order.

We first observe that if $v$ is a vertex of $\mathcal{B}$ for which the condition in step 10 of $\mathcal{A'}$ is satisfied, then co-dimension of $c^{(i)}_v$ is at most $D^v(g) - 1$, and $\text{bias}(c^{(i)}_v) = 1$; hence each leaf $\ell'$ of $\mathcal{B}$ reachable from $v$ has $\text{snip}(\ell') = 1$.

¹ Note that only the first inequality is used in the proof of Theorem 1.
We invoke Claim 9(b) with setting for each that in strictly less than

This completes the proof of Theorem 1.

3.1 Hardness Amplification Using XOR Lemma

In this section we prove Theorem 2.
A Composition Theorem for Randomized Query Complexity

Theorem 1 is useful only when the function $g$ is hard against randomized query algorithms even for error $1/2 - 1/n^4$. In this section we use an XOR lemma to show a procedure that, given any $g$ that is hard against randomized query algorithms with error $1/3$, obtains another function on a slightly larger domain that is hard against randomized query algorithms with error $1/2 - 1/n^4$. This yields the proof of Theorem 2.

Let $g : \{0, 1\}^m \rightarrow \{0, 1\}$ be a function. Let $g^\oplus_t : (\{0, 1\}^m)^t \rightarrow \{0, 1\}$ be defined as follows. For $x = (x^{(1)}, \ldots, x^{(t)}) \in (\{0, 1\}^m)^t$,

$$g^\oplus_t(x) = \oplus_{i=1}^t g(x^{(i)}).$$

The following theorem is obtained by specializing Theorem 3 of Andrew Drucker’s paper [5] to this setting.


$$R_{1/2-2^{-O(t)}}(g^\oplus_t) = \Omega(t \cdot R_{1/3}(g)).$$

Theorem 2 (restated below) follows by setting $t = \Theta(\log n)$ and combining Theorem 12 with Theorem 1.

▶ **Theorem 2.** Let $f \subseteq \{0, 1\}^n \times R$ be any relation. Let $g : \{0, 1\}^m \rightarrow \{0, 1\}$ be a function. Let $g^\oplus_t : (\{0, 1\}^m)^t \rightarrow \{0, 1\}$ be defined as follows: for $x = (x^{(1)}, \ldots, x^{(t)}) \in (\{0, 1\}^m)^t$, $g^\oplus_t(x) = \oplus_{i=1}^t g(x^{(i)}).$ Then,

$$R_{1/3}(f \circ (g^\oplus_{O(\log n)})^n) = \Omega(\log n \cdot R_{4/9}(f) \cdot R_{1/3}(g)).$$

References


