A Composition Theorem via Conflict Complexity

Swagato Sanyal *

January 11, 2018

Abstract

Let $R(\cdot)$ stand for the bounded-error randomized query complexity. We show that for any relation $f \subseteq \{0,1\}^n \times S$ and partial Boolean function $g \subseteq \{0,1\}^n \times \{0,1\}$, $R_{1/3}(f \circ g^n) = \Omega(R_{4/9}(f) \cdot \sqrt{R_{1/3}(g)})$. Independently of us, Gavinsky, Lee and Santha [3] proved this result. By an example demonstrated in their work, this bound is optimal. We prove our result by introducing a novel complexity measure called the conflict complexity of a partial Boolean function $g$, denoted by $\chi(g)$, which may be of independent interest. We show that $\chi(g) = \Omega(\sqrt{R(g)})$ and $R(f \circ g^n) = \Omega(R(f) \cdot \chi(g))$.

1 Introduction

Let $f \subseteq \{0,1\}^n \times S$ be a relation and $g \subseteq \{0,1\}^n \times \{0,1\}$ be a partial Boolean function. In this work, we bound the bounded-error randomized query complexity of the composed relation $f \circ g^n$ from below in terms of the bounded-error query complexities of $f$ and $g$. Our main theorem is as follows.

Theorem 1 (Main Theorem). For any relation $f \subseteq \{0,1\}^n \times S$ and partial Boolean function $g \subseteq \{0,1\}^n \times \{0,1\}$,

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

Prior to this work, Anshu et. al. [1] proved that $R_{1/3}(f \circ g^n) = \Omega(R_{4/9}(f) \cdot R_{1/2-1/\log R}(g))$. Although in the statement of their result $g$ is stated to be a Boolean function, their result holds even when $g$ is a partial Boolean function.

In the special case of $g$ being a total Boolean function, Ben-David and Kothari [2] showed that $R(f \circ g^n) = \Omega(R(f) \cdot \sqrt{R(g) \log R(g)})$.

Gavinsky, Lee and Santha [3] independently proved Theorem 1 (possibly with different values for the error parameters). They also prove this bound to be tight by exhibiting an example that matches this bound. We believe that our proof is sufficiently different and significantly shorter and simpler than theirs. We draw on and refine the ideas developed in the works of Anshu et. al. and Ben-David and Kothari to prove our result.

We define a novel measure of complexity of a partial Boolean function $g$ that we refer to as the conflict complexity of $g$, denoted by $\chi(g)$ (see Section 3 for a definition). This quantity is inspired by the Sabotage complexity introduced by ben-David and Kothari. However, the two measures also have important differences. For example, we could show that for any partial function $g$, $\chi(g)$ and $R(g)$ are related as follows.

*Division of Mathematical Sciences, Nanyang Technological University, Singapore and Centre for Quantum Technologies, National University of Singapore, Singapore. ssanyal@ntu.edu.sg
Theorem 2. For any partial Boolean function \( g \subseteq \{0,1\}^m \times \{0,1\} \),

\[
\chi(g) = \Omega \left( \sqrt{R_{1/3}(g)} \right).
\]

See Section 3 for a proof of Throrem 2. Sabotage complexity is known to be similarly related to the bounded-error randomized query complexity (up to a logarithmic factor) when \( g \) is a total Boolean function. For partial Boolean functions, unbounded separation is possible between sabotage complexity and \( R(\cdot) \).

We next prove the following composition theorem.

Theorem 3. Let \( S \) be an arbitrary set, \( f \subseteq \{0,1\}^n \times S \) be a relation and \( g \subseteq \{0,1\}^m \times \{0,1\} \) be a partial Boolean function. Then,

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

To prove Theorem 3 we draw on the techniques developed by Anshu et. al. and ben-David and Kothari. See Section 5 for a proof of Theorem 3. Theorem 1 follows from Theorems 2 and 3.

2 Preliminaries

A partial Boolean function \( g \) is a relation in \( \{0,1\}^m \times \{0,1\} \). For \( b \in \{0,1\} \), \( g^{-1}(b) \) is defined to be the set of strings \( x \) in \( \{0,1\}^n \) for which \((x,b) \in g \) and \((x,b) \notin g \). \( g^{-1}(0) \cup g^{-1}(0) \) is referred to as the set of valid inputs to \( g \). We assume that for all strings \( y \notin g^{-1}(0) \cup g^{-1}(1) \), both \((y,0)\) and \((y,1)\) are in \( g \). For a string \( x \in g^{-1}(0) \cup g^{-1}(1) \), \( g(x) \) refers to the unique bit \( b \) such that \((x,b) \in g \). All the probability distributions \( \mu \) over the domain of a partial Boolean function \( g \) in this paper are assumed to be supported entirely on \( g^{-1}(0) \cup g^{-1}(1) \). Thus \( g(x) \) is well-defined for any \( x \) in the support of \( \mu \).

Definition 1 (Bounded-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 \( h \subseteq \{0,1\}^k \times S \) 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 \( D_\epsilon^\mu(h) \) is the minimum number of queries made in the worst case (over inputs) by a deterministic query algorithm \( A \) for which \( \Pr_{x \sim \mu}[(x,A(x)) \in h] \geq 1 - \epsilon \).

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 1 (minimax principle). For any integer \( k \), set \( S \), and relation \( h \subseteq \{0,1\}^k \times S \),

\[
R_\epsilon(h) = \max_\mu D_\epsilon^\mu(h).
\]

Let \( \mu \) be a probability distribution over \( \{0,1\}^k \). \( x \sim \mu \) implies that \( x \) is a random string drawn from \( \mu \). Let \( C \subseteq \{0,1\}^k \) be arbitrary. Then \( \mu \mid C \) is defined to be the probability distribution obtained by conditioning \( \mu \) on the event that the sampled string belongs to \( C \), i.e.,

\[
\mu \mid C(x) = \begin{cases} 
0 & \text{if } x \notin C \\
\frac{\mu(x)}{\sum_{y \in C} \mu(y)} & \text{if } x \in C
\end{cases}
\]
For a partial Boolean function \( g : \{0,1\}^m \rightarrow \{0,1\} \), probability distribution \( \mu \) and bit \( b \),

\[ \mu_b := \mu \mid g^{-1}(b). \]

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

Now we define composition of two relations.

**Definition 4 (Composition of relations).** We now reproduce from the Section 1 the definition of composed relations. Let \( f \subseteq \{0,1\}^n \times \mathcal{S} \) and \( g \subseteq \{0,1\}^m \times \{0,1\} \) be two relations. The composed relation \( f \circ g \subseteq \{(0,1)^m\}^n \times \mathcal{S} \) is defined as follows: For \( x = (x^{(1)}, \ldots, x^{(n)}) \in ((0,1)^m)^n \) and \( s \in \mathcal{S} \), \( (x,s) \in f \circ g \) if and only if there exists \( b = (b^{(1)}, \ldots, b^{(n)}) \in \{0,1\}^n \) such that for each \( i = 1, \ldots, n \), \( (x^{(i)}, b^{(i)}) \in g \) and \((b,s) \in f \).

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 use the same symbol (e.g. \( v \)) to refer to a vertex as well as the subcube associated with it.

The depth of a vertex \( v \) in a tree is the number of vertices on the unique path from the root of the tree to \( v \) in the tree. Thus, the depth of the root is 1.

**Definition 5.** Let \( \mathcal{A} \) be a decision tree on \( m \) bits. Let \( \eta_0 \) and \( \eta_1 \) be two probability distributions with disjoint supports. Let \( v \) be a vertex in \( \mathcal{A} \). Let variable \( x_i \) be queried at \( v \). Then,

\[ \Delta^{(v)} := \left\{ \begin{array}{ll} \left| \Pr_{x \sim \eta_0}|x_i = 0| - \Pr_{x \sim \eta_1}|x_i = 0| \right| & \text{if } v \neq \bot, \\ 1 & \text{if } v = \bot. \end{array} \right. \]

Note that \( \Delta^{(v)} \) is defined with respect to distributions \( \eta_0 \) and \( \eta_1 \). In our application, we will often consider a decision tree \( \mathcal{A} \), a partial Boolean function \( g \) and a probability distributions \( \mu \) over the inputs. \( \Delta^{(v)} \), for a vertex \( v \) of \( \mathcal{A} \), will then be assumed to be with respect to the distributions \( (\mu_b \mid v)_{b \in \{0,1\}} \).

**Claim 2.** Let \( \mathcal{A} \) be a decision tree on \( m \) bits. Let \( g \) be a partial Boolean function. Let \( x \sim \{0,1\}^n \) be sampled from a distribution \( \mu \). Let \( v \) be a vertex in \( \mathcal{A} \). Let variable \( x_i \) be queried at \( v \). Then,

\[ l_\mu(g(x) : x_i \mid x \in v) = l_\mu_\mu(g(x) : x_i) \geq 32 \left( \Pr_{x \sim \mu}|g(x) = 0| \cdot \Pr_{x \sim \mu}|g(x) = 1| \cdot \Delta^{(v)} \right)^2, \]

where \( \Delta^{(v)} \) is with respect to the distributions \( (\mu_b \mid v)_{b \in \{0,1\}} \).

**Proof of Claim 2.** Define \( b := g(x) \). Condition on the event \( x \in v \). Let \( (b \otimes x_i) \) be the distribution over pairs of bits, where the bits are distributed independently according to the distributions of \( b \) and \( x_i \) respectively. We use the equivalence: \( I(b : x_i) = D((b, x_i)||b \otimes x_i)) \). Now, an application of Pinsker’s inequality implies that

\[ D((b, x_i)||b \otimes x_i) \geq 2||\langle b, x_i \rangle - (b \otimes x_i)\rangle^2. \]
Next, we bound $|(b, x_i) - (b \otimes x_i)||_1$. To this end, we fix bits $z_1, z_2 \in \{0, 1\}$, and bound $|\Pr[(b, x_i) = (z_1, z_2)] - \Pr[(b \otimes x_i) = (z_1, z_2)]|$. We have that,

$$\Pr[(b, x_i) = (z_1, z_2)] = \Pr[b = z_1] \Pr[x_i = z_2 \mid b = z_1].$$

(2)

Now,

$$\Pr[(b \otimes x_i) = (z_1, z_2)] = \Pr[b = z_1] \Pr[x_i = z_2] = \Pr[b = z_1] (\Pr[b = z_1] \Pr[x_i = z_2 \mid b = z_1] + \Pr[b = \overline{z_1}] \Pr[x_i = z_2 \mid b = \overline{z_1}]).$$

(3)

Taking the absolute difference of (3) and (2) we have that,

$$|\Pr[(b, x_i) = (z_1, z_2)] - \Pr[(b \otimes x_i) = (z_1, z_2)]| = \Pr[b = z_1] \cdot \Pr[b = \overline{z_1}] \cdot \Delta(v) = \Pr[b = 0] \cdot \Pr[b = 1] \cdot \Delta(v)$$

(4)

The Claim follows by adding (4) over $z_1, z_2$ and using (1).

\[\square\]

3 Conflict Complexity

In this section, we introduce a randomized process $P$ (formally given in Algorithm 1). This process is going to play a central role in the proof of our composition theorem (Theorem 3). Later in the section, we use $P$ to define the conflict complexity of a partial boolean function $g$.

Let $n > 0$ be any integer and $B$ be any deterministic query algorithm that runs on inputs in $(\{0, 1\}^m)^n$. $B$ can be though of as just a query procedure that queries various input variables, and then terminates without producing any output. Let $x = (x^{(j)}_{i})_{j=1,...,n}$ be a generic input to $B$, and $x_{i}$ stand for $(x_{i}^{(j)})_{j=1,...,m}$. For a vertex $v$ of $B$, $v^{(i)}$ denotes the subcube in $v$ corresponding to $x_i$, i.e., $v = x_{i=1}^{n} v^{(i)}$. Recall from Section 2 that for $b \in \{0, 1\}$, $v_b$ stands for the child of $v$ corresponding to the query outcome being $b$. Let $\mu_0$ and $\mu_1$ be any two probability distributions supported on $g^{-1}(0)$ and $g^{-1}(1)$ respectively. Let $z = (z_1, \ldots, z_n) \in \{0, 1\}^n$ be arbitrary. Now consider the probabilistic process $P$ given by Algorithm 1. Note that $P$ can be thought of as a randomized query algorithm on input $z \in \{0, 1\}^n$, where a query to $z_i$ corresponds to an assignment of $0$ to NOQUERY, in line 14. This view of $P$ will be adopted in Section 5.

We now prove an important structural result about $P$ which will be used many times in our proofs. Consider the following distribution $\gamma_z$ on $(\{0, 1\}^m)^n$: For each $i$, sample $x_i$ independently from $\mu_{z_i}$.

Let $v$ be a vertex of $B$. Let $A_B(v)$ be the event that process $P$ reaches node $v$, and $B_B(v)$ be the event that for a random input $x$ sampled from $\gamma_z$, the computation of $B$ reaches node $v$.

Claim 3. For each vertex $v$ of $B$,

$$\Pr[A_B(v)] = \Pr[B_B(v)].$$

Proof. We will prove by induction on the depth $t$ of $v$, i.e., the number of vertices on the unique path from the root to $v$ in $B$.

Base case: $t = 1$. $v$ is the root of $B$. Thus $\Pr[A_B(v)] = \Pr[B_B(v)] = 1$.

Inductive step: Assume that $t \geq 2$, and that the statement is true for all vertices at depth at most $t - 1$. Since $t \geq 2$, $v$ is not the root of $B$. Let $u$ be the ancestor of $v$, and variable $x_{i}^{(j)}$ be queried at $u$. Without loss of generality assume that $v$ is the child of $u$ corresponding to $x_{i}^{(j)} = 0$. We split the proof into the following two cases.
\textbf{Algorithm 1:} \(P\) on \(B, \mu_0, \mu_1, z\)
\begin{algorithmic}[1]
\For{\(1 \leq k \leq n\)}
\State \textbf{NOQUERY}\(_k \leftarrow 1\).
\State \textbf{N}_k \leftarrow 0.
\EndFor
\State \(v \leftarrow \text{Root of } B\) \hfill // Corresponds to \(\{0, 1\}^m\)
\While{\(v\) is not a leaf of \(B\)}
\State Let \(B\) query \(x^{(j)}_i\) at \(v\).
\If{\textbf{NOQUERY}\(_i = 1\)}
\State Sample a fresh real number \(r \sim [0, 1]\) uniformly at random.
\If{\(r \leq \min_{B} \text{Pr}_{x_i \sim \mu_k}[x^{(j)}_i = 0 \mid x_i \in v_{i}]\)}
\State \(v \leftarrow v_0\).
\Else\(\quad\) if \(r \geq \max_{B} \text{Pr}_{x_i \sim \mu_k}[x^{(j)}_i = 0 \mid x_i \in v^{(i)}]\) then
\State \(v \leftarrow v_1\).
\Else
\State \textbf{NOQUERY}\(_i \leftarrow 0\).
\If{\(r \leq \text{Pr}_{x_i \sim \mu_k}[x^{(j)}_i = 0 \mid x_i \in v^{(i)}]\)}
\State \(v \leftarrow v_0\).
\Else
\State \(v \leftarrow v_1\).
\EndIf
\EndIf
\State \textbf{N}_i \leftarrow \textbf{N}_i + 1.
\EndIf
\State Sample \(b\) from the distribution \(\mu_{z_i}\) conditioned on the event \(x_i \in v^{(i)}\).
\State \(v \leftarrow v_b\).
\EndWhile
\EndFor
\end{algorithmic}

- **Case 1:** \(\text{Pr}_{x_i \sim \mu_k}[x^{(j)}_i = 0 \mid x_i \in u_{i}] \leq \text{Pr}_{x_i \sim \mu_k}[x^{(j)}_i = 0 \mid x_i \in u_{i}]\).
  Condition on \(A_{B}(u)\) and \textbf{NOQUERY}\(_i = 0\). The probability that \(P\) reaches \(v\) is \(\text{Pr}_{x_i \sim \mu_k}[x^{(j)}_i = 0 \mid x_i \in u_{i}]\). Now, condition on \(A_{B}(u)\) and \textbf{NOQUERY}\(_i = 1\). The probability that \(P\) reaches \(v\) is exactly equal to the probability that the real number \(r\) sampled at \(v\) lies in \([0, \text{Pr}_{x_i \sim \mu_k}[x^{(j)}_i = 0 \mid x_i \in u_{i}]\]), which is equal to \(\text{Pr}_{x_i \sim \mu_k}[x^{(j)}_i = 0 \mid x_i \in u_{i}]\).

\begin{equation}
\text{Pr}[A_{B}(v) = \text{Pr}[A_{B}(u)], \text{Pr}[A_{B}(v) \mid A_{B}(u)] = \text{Pr}[A_{B}(u)] \cdot \text{Pr}_{x_i \sim \mu_k}[x^{(j)}_i = 0 \mid x_i \in u_{i}].}
\end{equation}

Now condition on \(B_{G}(u)\). The probability that \(B\) reaches \(v\) is exactly equal to the probability that \(x^{(j)}_i = 0\) when \(x\) is sampled according to the distribution \(\gamma_{z}\) conditioned on the event that \(x \in u\). Note that in the distribution \(\gamma_{z}\), the \(x_k\)'s are independently distributed. Thus,

\begin{equation}
\text{Pr}[B_{G}(v) = \text{Pr}[B_{G}(u)], \text{Pr}[B_{G}(v) \mid B_{G}(u)] = \text{Pr}[B_{G}(u)] \cdot \text{Pr}_{x_i \sim \mu_k}[x^{(j)}_i = 0 \mid x_i \in u_{i}].}
\end{equation}

By the inductive hypothesis, \(\text{Pr}[A_{B}(u)] = \text{Pr}[B_{G}(u)]\). The claim follows from (5) and (6).

- **Case 2:** \(\text{Pr}_{x_i \sim \mu_k}[x^{(j)}_i = 0 \mid x_i \in u_{i}] > \text{Pr}_{x_i \sim \mu_k}[x^{(j)}_i = 0 \mid x_i \in u_{i}]\). Let \(v'\) be the child of \(u\) corresponding to \(x^{(j)}_i = 1\). By an argument similar to Case 1, we have that

\begin{equation}
\text{Pr}[A_{B}(v')] = \text{Pr}[B_{G}(v')].
\end{equation}
Now,
\[
\Pr[A_B(v)] = \Pr[A_B(u)] - \Pr[A_B(v')] = \Pr[B_B(u)] - \Pr[A_B(v')] = \Pr[B_B(u)] - \Pr[B_B(v')] = \Pr[B_B(v)].
\]

Let \( n = 1, z \in \{0, 1\} \), and \( B \) be a decision tree that computes \( g \). Consider process \( P \) on \( B, \mu_0, \mu_1, z \). Note that \( \text{NOQUERY}_1 \) is set to 0 with probability 1. To see this observe that as long as \( \text{NOQUERY}_1 = 1 \), the current subcube \( v \) contains strings from the supports of both \( \mu_0 \) and \( \mu_1 \), and hence from both \( g^{-1}(0) \) and \( g^{-1}(1) \). If \( \text{NOQUERY}_1 \) is not set to 0 for the entire run of \( P \), then there exist inputs \( x \in g^{-1}(0), x' \in g^{-1}(1) \) which belong to the same leaf of \( B \), contradicting the hypothesis that \( B \) computes \( g \). Let the random variable \( N \) stand for the value of the variable \( N_1 \) after the termination of \( P \). Note that \( N \) is equal to the the index of the iteration of the while loop in which \( \text{NOQUERY}_1 \) is set to 0. The distribution of \( N \) depends on \( \mu_0, \mu_1 \) and \( B \), which in our applications will either be clear from the context, or clearly specified. Note that the distribution of \( N \) is independent of the value of \( z \).

**Definition 6.** The conflict complexity of a partial Boolean function \( g \) with respect to distributions \( \mu_0 \) and \( \mu_1 \) supported on \( g^{-1}(0) \) and \( g^{-1}(1) \) respectively, and decision tree \( B \) computing \( g \), is defined as:

\[
\chi(\mu_0, \mu_1, B) = E[N].
\]

The conflict complexity of \( g \) is defined as:

\[
\chi(g) = \max_{\mu_0, \mu_1} \min_B \chi(\mu_0, \mu_1, B).
\]

Where the maximum is over distributions \( \mu_0 \) and \( \mu_1 \) supported on \( g^{-1}(0) \) and \( g^{-1}(1) \) respectively, and the minimum is over decision trees \( B \) computing \( g \).

For a pair \((\mu_0, \mu_1)\) of distributions, let \( B \) be the decision tree computing \( g \) such that \( E[N] \) is minimized. We call such a decision tree an optimal decision tree for \( \mu_0, \mu_1 \). We conclude this section by making an important observation about the structure of optimal decision trees. Let \( v \) be any node of \( B \). Let \( \mu'_0 := \mu_0 \mid v \) and \( \mu'_1 := \mu_1 \mid v \). Let \( B_v \) denote the subtree of \( B \) rooted at \( v \). We observe that \( B_v \) is an optimal tree for \( \mu'_0 \) and \( \mu'_1 \); if it is not then we could replace it by an optimal tree for \( \mu'_0 \) and \( \mu'_1 \), and for the resultant tree, the expected value of \( N \) with respect to \( \mu_0 \) and \( \mu_1 \) will be smaller than that in \( B \). This will contradict the optimality of \( B \). This recursive sub-structure property of optimal trees will be helpful to us.

## 4 Conflict Complexity and Randomized Query Complexity

In this section, we will prove Theorem 2 (restated below).

**Theorem 2.** For any partial Boolean function \( g \subseteq \{0, 1\}^n \times \{0, 1\} \),

\[
\chi(g) = \Omega \left( \sqrt{R_{1/3}(g)} \right).
\]

---

\(^1\text{As observed before, the choices of } \mu_0, \mu_1 \text{ and } B \text{ are built into the definition of } N.\)
Proof. We will bound the distributional query complexity of \( g \) for each input distribution \( \mu \) with respect to error \( 47/95 < 1/2 \), \( D_{47/95}^\mu(g) \), from above by \( O(\chi(g)^2) \). Theorem 2 will follow from the minimax principle (Fact 1), and the observation that the error can be brought down to \( 1/3 \) by constantly many independent repetitions followed by a selection of the majority of the answers. It is enough to consider distributions \( \mu \) supported on valid inputs of \( g \). To this end, fix a distribution \( \mu \) supported only on \( g^{-1}(0) \cup g^{-1}(1) \).

Let \( \chi(g) = d \). Let \( \mu_b \) be the distribution obtained by conditioning \( \mu \) on the event \( g(x) = b \). Let \( \mathcal{B} \) be an optimal decision tree for distributions \( \mu_b \) and \( \mu_1 \). Clearly \( E[\mathcal{N}] \leq \chi(g) = d \).

We first prove some structural results about \( \mathcal{B} \). Let \( \mathcal{B} \) be run on a random input \( x \) sampled according to \( \mu \). Let \( v_t \) be the random vertex at which the \( t \)-th query is made; if \( \mathcal{B} \) terminates before making \( t \) queries, define \( v_t := \bot \). Let \( \mathcal{E} \) be any event which is a collection of possible transcripts of \( \mathcal{B} \), such that \( \Pr[\mathcal{E}] \geq \frac{3}{4} \). Recall from Section 2 that for any vertex \( v \) of \( \mathcal{B} \), \( \Delta(v) \) is assumed to be with respect to the probability distribution \( \mu \mid v \).

Claim 4.

\[ \sum_{t=1}^{10d} E[\Delta(u_t) \mid \mathcal{E}] \geq \frac{13}{20}. \]

Proof. Let us sample vertices \( u_t \) of \( \mathcal{B} \) as follows:

1. Set \( z = \begin{cases} 1 & \text{with probability } \Pr_{x \sim \mu}[g(x) = 1], \\ 0 & \text{with probability } \Pr_{x \sim \mu}[g(x) = 0] \end{cases} \)

2. Run process \( \mathcal{P} \) for \( \mathcal{B}, \mu_0, \mu_1, z \).

3. Let \( u_t \) be the vertex \( v \) in the beginning of the \( t \)-th iteration of the \textit{while} loop of Algorithm 1.

Return \((u_t)_{t=1}^{10d}\). If the simulation stops after \( i \) iterations, set \( u_t := \bot \) for all \( t > i \).

By Claim 3, and since \( z \) has the same distribution as that of \( g(x) \) where \( x \) is sampled from \( \mu \), the vertices \( u_t \) and \( v_t \) have the same distribution. In the above sampling process for each \( t = 1, \ldots, 10d \), let \( E_t \) be the event that \( \text{NOQUERY}_1 = 1 \) in the beginning of the \( t \)-th iteration of the \textit{while} loop of Algorithm 1. Conditioned on \( \mathcal{E} \), the probability that \( \text{NOQUERY}_1 \) is set to 0 in the \( t \)-th iteration is \( \Pr[E_t \mid \mathcal{E}] \cdot E[\Delta(u_t) \mid E_t, \mathcal{E}]^2 \). By union bound we have that,

\[
\sum_{t=1}^{10d} E[\Delta(u_t) \mid \mathcal{E}] = \sum_{t=1}^{10d} E[\Delta(u_t) \mid \mathcal{E}]
\geq \sum_{t=1}^{10d} \Pr[E_t \mid \mathcal{E}] \cdot E[\Delta(u_t) \mid E_t, \mathcal{E}]
\geq \Pr \left[ \bigcap_{t=1}^{10d} E_t \mid \mathcal{E} \right] - \Pr[\mathcal{E}] .
\]

Now, since \( E[\mathcal{N}] \leq \chi(g) = d \), we have by Markov’s inequality that the probability that the process \( \mathcal{P} \), when run for \( \mathcal{B}, \mu_0, \mu_1 \) and the random bit \( z \) generated as above\(^1\), sets \( \text{NOQUERY}_1 \) to 0 within first

\(^1\)Note that conditioned on \( E_t \), \( u_t \neq \bot \).

\(^2\)Recall that the distribution of \( \mathcal{N} \) is independent of \( z \).
10d iterations of the while loop, is at least 9/10. Thus we have that,

$$\Pr\left[\bigcap_{t=1}^{10d} E_t\right]^c \geq \frac{9}{10}. \quad (9)$$

The claim follows from (8), (9) and the hypothesis $\Pr[\mathcal{E}] \geq \frac{1}{4}$.

The next Lemma follows from Claim 4 and the recursive sub-structure property of optimal trees discussed in the last paragraph of Section 3.

**Lemma 5.** Let $i$ be any positive integer. Then,

$$\sum_{t=1}^{10di} \mathbb{E}[\Delta(v_t) | \mathcal{E}] \geq \frac{13i}{20}.$$  \(10\)

Notice that if $B$ terminates before making $t$ queries, $v_t = \bot$ and $\Delta(v_t) = 1$.

**Proof of Lemma 5.** For $j = 1, \ldots, i$, let $w$ be any vertex at depth $10jd + 1$. Consider the subtree $T$ of $B$ rooted at $w$. By the recursive sub-structure property of $B$, $T$ is an optimal tree for distributions $\mu'_0 := \mu_0 | w, \mu'_1 := \mu_1 | w$. Let $w_t$ be the random vertex at depth $t$ of $T$, when $T$ is run on a random input from $\mu | w$. By Claim 4, we have that,

$$\sum_{t=1}^{10d} \mathbb{E}[\Delta(v_t) | \mathcal{E}] \geq \frac{13}{20}. \quad (10)$$

In (10), $\Delta(v_t)$ is with respect to distributions $\mu'_0 | w_t = \mu_0 | w_t, \mu'_1 | w_t = \mu_1 | w_t$. Now, when $w$ is the random vertex $v_{10jd+1}$, $w_t$ is the random vertex $v_{10jd+t}$. Thus from (10) we have that,

$$\sum_{t=10jd+1}^{10(j+1)d} \mathbb{E}[\Delta(v_t) | \mathcal{E}] \geq \frac{13}{20}. \quad (11)$$

The claim follows by adding (11) over $j = 0, \ldots, i - 1$. \qed

We now finish the proof of Theorem 2 by showing that $D^\mu(g) = O(d^2)$. Let $x$ be distributed according to $\mu$, and $B$ be run on $x$. Let BIASED denote the event that in at most $10d^2$ queries, the computation of $B$ reaches a vertex $v$ for which $\Pr_{x \sim \mu}[g(x) = 0 \mid x \in v] \cdot \Pr_{x \sim \mu}[g(x) = 1 \mid x \in v] \leq \frac{1}{3}$. Let STOP denote the event that $B$ terminates after making at most $10d^2$ queries. Let $\mathcal{E} := \text{BIASED} \lor \text{STOP}$. Consider the following decision tree $B'$: Start simulating $B$. Terminate the simulation if one of the following events occurs. The outputs in each case is specified below.

1. (Event STOP) If $B$ terminates, terminate and output what $B$ outputs.

2. If $10d^2$ queries have been made and the computation is at a vertex $v$, terminate and output $\arg\max_b \Pr[g(x) = b \mid x \in v]$.

By construction, $B'$ makes at most $10d^2$ queries in the worst case. We shall show that $\Pr_{x \sim \mu}[B'(x) \neq g(x)] \leq \frac{47}{95} < \frac{1}{2}$. This will prove Theorem 2.

We split the proof into the following two cases.

8
Case 1: \( \Pr[\mathcal{E}] \geq \frac{1}{4} \).

First, condition on the event that the computation reaches a vertex \( v \) for which \( \Pr_{x \sim \mu}[g(x) = 0 \mid x \in v] \cdot \Pr_{x \sim \mu}[g(x) = 1 \mid x \in v] \leq \frac{1}{4} \) holds. Thus one of \( \Pr_{x \sim \mu}[g(x) = 0 \mid x \in v] \) and \( \Pr_{x \sim \mu}[g(x) = 1 \mid x \in v] \) is at most \( 1/3 \). Hence, \( \Pr_{x \sim \mu}[g(x) = 0 \mid x \in v] - \Pr_{x \sim \mu}[g(x) = 1 \mid x \in v] \geq 2/3 \). Let \( m \) be the random leaf of the subtree of \( B' \) rooted at \( v \) at which the computation ends. The probability that \( B' \) errs is at most

\[
\mathbb{E}_{x \sim \mu \mid v} \left[ \frac{1}{2} - \frac{1}{2} \left| \Pr_{x \sim \mu}[g(x) = 0 \mid x \in m] - \Pr_{x \sim \mu}[g(x) = 1 \mid x \in m] \right| \right] \\
\leq \frac{1}{2} - \frac{1}{2} \mathbb{E}_{x \sim \mu \mid v} \left[ \Pr_{x \sim \mu}[g(x) = 0 \mid x \in m] - \mathbb{E}_{x \sim \mu \mid v} \Pr_{x \sim \mu}[g(x) = 1 \mid x \in m] \right] \\
= \frac{1}{2} - \frac{1}{2} \left| \Pr_{x \sim \mu}[g(x) = 0 \mid x \in v] - \Pr_{x \sim \mu}[g(x) = 1 \mid x \in v] \right| \leq \frac{1}{3}.
\]

(By Jensen’s inequality)

Then, condition on the event \( \text{STOP} \). The probability that \( B' \) errs is 0 \( \leq \frac{1}{3} \).

Thus we have shown that conditioned on \( \mathcal{E} \) the probability that \( B' \) errs is at most \( \frac{1}{3} \). Thus the probability that \( B' \) errs is at most \( \frac{1}{4} \cdot \frac{1}{3} + \frac{3}{4} \cdot \frac{1}{2} = \frac{11}{24} < \frac{47}{95} \).

Case 2: \( \Pr[\mathcal{E}] < \frac{1}{4} \).

By Claim 5 we have that

\[
\sum_{i=1}^{10d^2} \mathbb{E} \left[ \Delta^{(i)} \mid \mathcal{E} \right] \geq \frac{13d}{20}.
\] (12)

Let \( a_i := (x_i, b_i) \) be the tuple formed by the random input variable \( x_i \) queried at the \( i \)-th step by \( B' \), and the outcome \( b_i \) of the query; if \( B' \) terminates before \( i \)-th step, \( a_i := \bot \). Notice that the vertex \( v_i \) at which the \( i \)-th query is made is determined by \((a_1, \ldots, a_{i-1})\) and vice versa. We have,

\[
\begin{align*}
\text{l}(a_1, \ldots, a_{10d^2} : g(x)) & = \sum_{i=1}^{10d^2} \text{l}(a_i : g(x) \mid a_1, \ldots, a_{i-1}) \quad \text{(Chain rule of mutual information)} \\
& = \sum_{i=1}^{10d^2} \text{l}(b_i : g(x) \mid v_i) \\
& \geq 32 \sum_{i=1}^{10d^2} \mathbb{E} \left[ 1_{v_i \neq \bot} \cdot \left[ \Pr[g(x) = 0 \mid x \in v_i] \cdot \Pr[g(x) = 1 \mid x \in v_i] \cdot \Delta^{(v_i)} \right]^2 \right] \\
& \geq 32 \sum_{i=1}^{10d^2} \Pr[\mathcal{E}] \cdot \mathbb{E} \left[ \left[ \Pr[g(x) = 0 \mid x \in v_{i-1}] \cdot \Pr[g(x) = 1 \mid x \in v_{i-1}] \cdot \Delta^{(v_i)} \right]^2 \mid \mathcal{E} \right] \\
& \quad \text{(Conditioned on } \mathcal{E}, v_i \neq \bot) \\
& \geq 32 \sum_{i=1}^{10d^2} 3 \cdot \frac{1}{9} \cdot \mathbb{E}[\Delta^{(v_i)}^2 \mid \mathcal{E}] \\
& \geq \frac{3}{4} \cdot \frac{1}{9} \cdot \sum_{i=1}^{10d^2} \mathbb{E}[\Delta^{(v_i)}^2 \mid \mathcal{E}] \\
& = \frac{8}{3} \sum_{i=1}^{10d^2} \mathbb{E}[\Delta^{(v_i)}^2 \mid \mathcal{E}] \quad \text{(By the assumption } \Pr[\mathcal{E}] \leq \frac{1}{4})
\end{align*}
\]
Observe that for each $z, x, A$ algorithm $R$, we shall prove that for each distribution $\rho$ and $\eta, \mu$ respectively, such that for every decision tree $B$, we have

$$\Pr_{z \sim \rho}[g(x) = b | x \in \ell] \leq \frac{2}{3}.$$ Conditioned on $(a_1, \ldots, a_{10d^2}) \in \mathcal{L}$, the probability that $B'$ err is at most $\frac{2}{5}$. By Markov's inequality and (14), it follows that $\Pr[(a_1, \ldots, a_{10d^2}) \in \mathcal{L}] \geq \frac{1}{19}$. Thus $B'$ err with probability at most $\frac{1}{19} \cdot \frac{2}{5} + \frac{18}{19} \cdot \frac{1}{2} = \frac{47}{95}$.

$$\square$$

## 5 The Composition Theorem

In this section we prove Theorem 3 (restated below).

**Theorem 3.** Let $\mathcal{S}$ be an arbitrary set, $f \subseteq \{0,1\}^n \times \mathcal{S}$ be a relation and $g \subseteq \{0,1\}^m \times \{0,1\}$ be a partial Boolean function. Then,

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

**Proof.** We shall prove that for each distribution $\eta$ on the inputs to $f$, there is a query algorithm $A$ making $O(R(f \circ g^n) / \chi(g))$ queries in the worst case, for which $\Pr_{x \in \mu}[g(x) = b | x \in \ell] \leq \frac{2}{3}$. Conditioned on $(a_1, \ldots, a_{10d^2}) \in \mathcal{L}$, the probability that $B'$ err is at most $\frac{2}{5}$. By Markov's inequality and (14), it follows that $\Pr[(a_1, \ldots, a_{10d^2}) \in \mathcal{L}] \geq \frac{1}{19}$. Thus $B'$ err with probability at most $\frac{1}{19} \cdot \frac{2}{5} + \frac{18}{19} \cdot \frac{1}{2} = \frac{47}{95}$.

Now, let $\gamma_{\eta}$ be the distribution over $(\{0,1\}^m)^n$ that is given by the following sampling procedure:

1. For $i = 1, \ldots, n$, sample $x_i = (x_i^{(j)})_{j=1,\ldots,m}$ from $\mu_{z_i}$, independently for each $i$.
2. Return $x = (x_i)_{i=1,\ldots,n}$.

Now, let $\gamma_0$ be the distribution over $(\{0,1\}^m)^n$ that is given by the following sampling procedure:

1. Sample $z = (z_1, \ldots, z_n)$ from $\eta$.
2. Sample $x = (x_i)_{i=1,\ldots,n}$ from $\gamma_z$. Return $x$.

Observe that for each $z, x$ sampled as above, for each $s \in \mathcal{S}$, $(z, s) \in f$ if and only if $(x, s) \in f \circ g^n$.

Assume that $R_{1/3}(f \circ g^n) = t$. Yao’s minimax principle implies that there is a deterministic query algorithm $A'$ for inputs from $(\{0,1\}^m)^n$, that makes at most $t$ queries in the worst case, such that $\Pr_{x \in \gamma}[g(x, A'(x)) \in f \circ g^n] \geq \frac{2}{3}$. We will first use $A'$ to construct a randomized algorithm $T$ for $f$, whose accuracy is as desired, and for which the expected number of queries made is small.

$$\text{(13)}$$

Hence, from (13) we have

$$H(g(x) | a_1, \ldots, a_{10d^2}) \leq 1 - \frac{1}{10} = \frac{9}{10}.$$ (14)
Algorithm 2: $T$ on $z$

1. for $1 \leq k \leq n$ do
2.   \begin{align*}
   & \text{NOQUERY}_k \leftarrow 1. \\
   & N_k \leftarrow 0.
\end{align*}
3. $v \leftarrow \text{Root of $\mathcal{A}'$}$ \quad // Corresponds to $\{0,1\}^m$
4. while $v$ is not a leaf of $\mathcal{A}'$ do
5.   \begin{align*}
   & \text{Let $\mathcal{A}'$ query $x^{(j)}_i$ at $v$.} \\
   & \text{if $\text{NOQUERY}_i = 1$ then} \\
   & \quad \begin{align*}
   & \text{Sample a fresh real number } r \sim [0,1] \text{ uniformly at random.} \\
   & \text{if } r \leq \min_b \Pr_{x_i \sim \mu_b} [x^{(j)}_i = 0 \mid x_i \in v_i] \text{ then} \\
   & \quad v \leftarrow v_0. \\
   & \text{else if } r \geq \max_b \Pr_{x_i \sim \mu_b} [x^{(j)}_i = 0 \mid x_i \in v^{(i)}] \text{ then} \\
   & \quad v \leftarrow v_1. \\
   & \text{else} \\
   & \quad \text{NOQUERY}_i \leftarrow 0. \\
   & \quad \text{Query $z_i$.} \\
   & \text{if } r \leq \Pr_{x_i \sim \mu_{z_i}} [x^{(j)}_i = 0 \mid x_i \in v^{(i)}] \text{ then} \\
   & \quad v \leftarrow v_0. \\
   & \text{else} \\
   & \quad v \leftarrow v_1. \\
   & N_i \leftarrow N_i + 1.
   \end{align*}
   \text{else}
   \begin{align*}
   & \text{Sample $b$ from the distribution $\mu_{z_i}$ conditioned on the event $x_i \in v^{(i)}$.} \\
   & v \leftarrow v_b.
\end{align*}
\end{align*}

$T$, described formally in Algorithm 2, is essentially viewing the process $P$ for $z, \mu_0, \mu_1, \mathcal{A}'$ as a query algorithm running on input $z$; an assignment of 0 to $\text{NOQUERY}_i$ corresponds to a query to $z_i$. By Claim 3, we have that for each $z \in \{0,1\}^n$, $\Pr[(z, T(z)) \in f] = \Pr_{x_i \sim \gamma_i} [(x, \mathcal{A}'(x)) \in f \circ \gamma^*]$. Thus, $\Pr_{x_i \sim \eta} [(z, T(z)) \in f] = \Pr_{x_i \sim \gamma_i} [(x, \mathcal{A}'(x)) \in f \circ \gamma^*] \geq \frac{2}{3}$.

We now bound the expected number of queries made by $T$ on each $z$. For doing that we consider the following randomized process $Q$ that acts on $z$. Let $\mathcal{B}$ be an optimal tree for distributions $\mu_0, \mu_1$. $Q$ is described formally in Algorithm 3. Since $\mathcal{B}$ computes $g$, process $Q$ is guaranteed to set $\text{NOQUERY}_i$

Algorithm 3: $Q$ on $z$

1. Run $T$ on $z$.
2. for $1 \leq i \leq n$ do
3.   \begin{align*}
   & \text{if $\text{NOQUERY}_i = 1$ then} \\
   & \quad \text{Run process $P$ on } \mathcal{B}, \mu_0, \mu_1, x_i \text{ until } \text{NOQUERY}_i \text{ is set to 0.}
\end{align*}

to 0 for each $i$. In steps 1 and 4, the process $P$ is run with trees $\mathcal{A}'$ and $\mathcal{B}$, and the trees make queries inside the for loop of $P$. These queries can be thought of as being made to an $mn$ bit string $(x^{(j)}_i)_{i=1}^n$. Let the random variable $X_i$ stand for the total number of queries made by these trees in $x_i$. $X = \sum_{i=1}^n X_i$ is the total number of queries in $Q$, i.e., the total number of iterations of the for loop of $P$ in all the runs of $P$ in $Q$. The next claim bounds $E_X$ from below.
Claim 6. \( \mathbb{E}X \geq nd. \)

Proof. Towards a contradiction assume that \( \mathbb{E}X < nd. \) Thus there exists an \( i \) such that \( \mathbb{E}X_i < d. \) Notice that this expectation is over the random real numbers sampled in the for loop of \( \mathcal{P}. \) Thus, there exists a fixing of those real numbers \( r \) that are sampled in those iterations of the for loop of \( \mathcal{P} \) that correspond to queries into \( x_j \) for \( j \neq i, \) such that conditioned on that fixing, \( \mathbb{E}X_i < d. \) However, under that fixing, process \( \mathcal{Q} \) is equivalent to process \( \mathcal{P} \) for some deterministic decision tree \( T' \) that computes \( g(z_i) \) (since \( \text{NOQUERY}_i \) is set to 0 with probability 1), \( \mu_0, \mu_1 \) and \( z_i. \) Thus \( \mathbb{E}X_i < d \) conditioned on the above-mentioned fixing of randomness contradicts the assumption that \( \min_{\beta} \chi(\beta, \mu_0, \mu_1) = \chi(g) = d, \) where the minimum is taken over all deterministic decision tree \( \beta \) that computes \( g. \)

Now, let \( Y \) denote the size of the random set \( \{i \mid \text{NOQUERY}_i \text{ is set to 0 in step 1 in } \mathcal{Q}\}. \) Now, conditioned on the event \( Y = b, \) the expected number of queries made in step 4 of \( \mathcal{Q} \) is \( (n-b)d = nd - bd. \) So under this conditioning the total number of queries \( X \) made by \( \mathcal{Q} \) is at most \( t + nd - bd. \) Taking expectation over \( b, \) and using Claim 6 we have that

\[
t + nd - d \cdot \mathbb{E}Y \geq nd \implies \mathbb{E}Y \leq \frac{t}{d}.
\]

Observing that for each \( z, \) \( Y \) has the same distribution as the number of queries made by \( T \) when run on \( z, \) we conclude that for each \( z, \) \( T \) makes at most \( t/d \) queries on expectation. By Markov’s inequality, the probability that \( T \) makes more than \( 9t/d \) queries is at most \( 1/9. \) Thus the probabilistic algorithm \( \mathcal{A}' \) obtained by terminating \( T \) after \( 10t/d \) queries computes \( f \) with probability at least \( 2/3 - 1/9 = 5/9 > 1/2 \) on a random input from \( \eta. \) By fixing the randomness of \( \mathcal{A}' \) appropriately we get a deterministic algorithm \( \mathcal{A} \) of complexity \( O(t/d) = O(\mathbb{R}(f \circ g)/\chi(g)) \) such that \( \Pr_{z \sim \eta}[\{z, \mathcal{A}(z)\} \in f] \geq \frac{5}{9}. \)

Acknowledgements. I thank Rahul Jain for helpful discussions.

This material is based on research supported by the Singapore National Research Foundation under NRF RF Award No. NRF-NRFF2013-13.

References

[1] Anurag Anshu, Dmitry Gavinsky, Rahul Jain, Srijita Kundu, Troy Lee, Priyanka Mukhopadhyay, Miklos Santha, and Swagato Sanyal. A composition theorem for randomized query complexity. In FSTTCS, 2017.

[2] Shalev Ben-David and Robin Kothari. Randomized query complexity of sabotaged and composed functions. In 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy, pages 60:1–60:14, 2016.

[3] Dmitry Gavinsky, Troy Lee, and Miklos Santha. On the randomised query complexity of composition. CoRR, abs/1801.02226, 2018.