A Tight Composition Theorem for the Randomized Query Complexity of Partial Functions
(Extended Abstract†)

Shalev Ben-David
Eric Blais
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
(shalev.b | eric.blais)@uwaterloo.ca

***Abstract*—We prove two new results about the randomized query complexity of composed functions. First, we show that the randomized composition conjecture is false: there are families of partial Boolean functions f and g such that $\mathrm{R}(f \circ g) \ll \mathrm{R}(f)\,\mathrm{R}(g)$. In fact, we show that the left hand side can be polynomially smaller than the right hand side (though in our construction, both sides are polylogarithmic in the input size of f).**

Second, we show that for all f and g, $\mathrm{R}(f \circ g) = \Omega(\mathrm{noisyR}(f)\,\mathrm{R}(g))$, where $\mathrm{noisyR}(f)$ is a measure describing the cost of computing f on noisy oracle inputs. We show that this composition theorem is the strongest possible of its type: for any measure $M(\cdot)$ satisfying $\mathrm{R}(f \circ g) = \Omega(M(f)\,\mathrm{R}(g))$ for all f and g, it must hold that $\mathrm{noisyR}(f) = \Omega(M(f))$ for all f. We also give a clean characterization of the measure $\mathrm{noisyR}(f)$: it satisfies $\mathrm{noisyR}(f) = \Theta(\mathrm{R}(f \circ \textsc{GapMaj}_n)/\,\mathrm{R}(\textsc{GapMaj}_n))$, where n is the input size of f and $\textsc{GapMaj}_n$ is the $\sqrt{n}$-gap majority function on n bits.

***Keywords*-Query complexity; Randomized computation; Function composition; Noisy oracle; Gap Majority function**

I. Introduction

In any computational model, one may ask the following basic question: is computing a function g on n independent inputs roughly n times as hard as computing g on a single input? If so, a natural followup question arises: how hard is computing some function $f\colon \{0,1\}^n \to \{0,1\}$ of the value of g on n inputs? Can this be characterized in terms of the complexity of the function f?

Query complexity is one of the simplest settings in which one can study these joint computation questions. In query complexity, a natural conjecture is that for any such functions f and g, the cost of computing f on the value of g on n inputs is roughly the cost of computing f times the cost of computing g. Indeed, using $f \circ g$ to denote the composition of f with n copies of g, it is known that the deterministic query complexity (also known as the decision tree complexity) of composed functions satisfies $\mathrm{D}(f \circ g) = \mathrm{D}(f)\,\mathrm{D}(g)$ [Tal13; Mon14]. It is also known that the quantum query complexity (in the bounded-error setting) of composed functions satisfies $\mathrm{Q}(f \circ g) = \Theta(\mathrm{Q}(f)\,\mathrm{Q}(g))$ [Rei11; LMR+11; Kim13].

However, despite significant interest, the situation for randomized query complexity is not well understood, and it is currently unknown whether $\mathrm{R}(f \circ g) = \widetilde{\Theta}(\mathrm{R}(f)\,\mathrm{R}(g))$ holds for all Boolean functions f and g. It is known that the upper bound of $\mathrm{R}(f \circ g) = O(\mathrm{R}(f)\,\mathrm{R}(g)\log \mathrm{R}(f))$ holds. This follows from running an algorithm for f on the outside, and then using an algorithm for g to answer each query made by the algorithm for f. (The log factor in the bound is due to the need to amplify the success probability of the algorithm for g so that it has small error.) The randomized composition conjecture in query complexity posits that there is a lower bound that matches this upper bound up to logarithmic factors; this conjecture is the focus of the current work.

Main Question. *Do all Boolean functions f and g satisfy $\mathrm{R}(f \circ g) = \Omega\big(\mathrm{R}(f)\,\mathrm{R}(g)\big)$?*

Note that there are actually two different versions of this question, depending on whether f and g are allowed to be partial functions. A *partial function* is a function $f\colon S \to \{0,1\}$ where S is a subset of $\{0,1\}^n$, and a randomized algorithm computing it is only required to be correct on the domain of f. (Effectively, the input string is promised to be inside this domain.) When composing partial functions f and g, we get a new partial function $f \circ g$, whose domain is the set of strings for which the computation of f and of each copy of g are all well-defined. Since partial functions are a generalization of total Boolean functions, it is possible that the composition conjecture holds for total functions but not for partial functions. In this work, we will mainly focus on the more general partial function setting; when we do not mention anything about f or g, they should be assumed to be partial Boolean functions.

† The full version of the article is available on arXiv as report number 2002.10809.

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00031

A. Previous work

Direct sum and product theorems: Direct sum theorems and direct product theorems study the complexity of $\mathrm{ID} \circ g$, where g is an arbitrary Boolean function but $\mathrm{ID}\colon \{0,1\}^n \to \{0,1\}^n$ is the identity function. These are not directly comparable to composition theorems, but they are of a similar flavor.

Jain, Klauck, and Santha [JKS10] showed that randomized query complexity satisfies a direct sum theorem. Drucker [Dru12] showed that randomized query complexity also satisfies a direct product theorem, which means that $\mathrm{ID} \circ g$ cannot be solved too quickly even with small success probability. More recently, Blais and Brody [BB19] proved a strong direct sum theorem, showing that computing n copies of g can be even harder for randomized query complexity than n times the cost of computing g (due to the need for amplification).

Composition theorems for other complexity measures: Several composition theorems are known for measures that lower bound $\mathrm{R}(f)$; as such, these theorems can be used to lower bound $\mathrm{R}(f \circ g)$ in terms of some smaller measure of f and g.

First, though it is not normally phrased this way, the composition theorem for quantum query complexity [Rei11; LMR+11] can be viewed as a composition theorem for a measure which lower bounds $\mathrm{R}(f)$, since $\mathrm{Q}(f) \le \mathrm{R}(f)$ for all f. Interestingly, as a lower bound technique for $\mathrm{R}(f)$, $\mathrm{Q}(f)$ turns out to be incomparable to the other lower bounds on randomized query complexity for which composition is known, meaning that this composition theorem can sometimes be stronger than everything we know how to do using classical techniques.

Tal [Tal13] and independently Gilmer, Saks, and Srinivasan [GSS16] studied the composition behavior of simple measures like sensitivity, block sensitivity, and fractional block sensitivity. The behavior turns out to be somewhat complicated, but is reasonably well characterized in these works.

Göös and Jayram [GJ16] studied the composition behavior of conical junta degree, also known as approximate nonnegative degree. This measure is a powerful lower bound technique for randomized algorithms and seems to be equal to $\mathrm{R}(f)$ for all but the most artificial functions; however, Göös and Jayram were only able to prove a composition theorem for a variant of conical junta degree, and the variant appears to be weaker in some cases (or at least harder to use).

Ben-David and Kothari [BK18] showed a composition theorem for a measure they defined called randomized sabotage complexity, denoted $\mathrm{RS}(f)$. They showed that this measure is larger than fractional block sensitivity, and incomparable to quantum query complexity and conical junta degree. It is also nearly quadratically related to $\mathrm{R}(f)$ for total functions.

Composition theorems with a loss in g: There are also composition theorems are known that lower bound $\mathrm{R}(f \circ g)$ in terms of $\mathrm{R}(f)$ and some smaller measure of g.

Ben-David and Kothari [BK18] also showed that $\mathrm{R}(f \circ g) = \Omega(\mathrm{R}(f)\,\mathrm{RS}(g))$, for the randomized sabotage complexity measure $\mathrm{RS}(g)$ mentioned above. Anshu et al. [AGJ+17] showed that $\mathrm{R}(f \circ g) = \Omega(\mathrm{R}(f)\,\mathrm{R}_{1/2-n^{-4}}(g))$, where $\mathrm{R}_{1/2-n^{-4}}(g)$ is the randomized query complexity of g to bias n^{-4}. These two results can also be used to give composition theorems of the form $\mathrm{R}(f \circ h \circ g) = \Omega(\mathrm{R}(f)\,\mathrm{R}(h)\,\mathrm{R}(g))$, where f and g are arbitrary Boolean functions but h is a fixed small gadget designed to break up any "collusion" between f and g. [BK18] proved such a theorem when h is the index function, while [AGJ+17] proved it when h is the parity function of size $O(\log n)$.

Finally, Gavinsky, Lee, Santha, and Sanyal [GLSS19] showed that $\mathrm{R}(f \circ g) = \Omega(\mathrm{R}(f)\overline{\chi}(g))$, where $\overline{\chi}(g)$ is a measure they define. They showed that $\overline{\chi}(g) = \Omega(\mathrm{RS}(g))$ and that $\overline{\chi}(g) = \Omega(\sqrt{\mathrm{R}(g)})$ (even for partial functions g), which means their theorem also shows $\mathrm{R}(f \circ g) = \Omega(\mathrm{R}(f)\sqrt{\mathrm{R}(g)})$.

Composition theorems with a loss in f: There have been very few composition theorems of the form $\mathrm{R}(f \circ g) = \Omega(M(f)\,\mathrm{R}(g))$ for some measure $M(f)$. Göös, Jayram, Pitassi, and Watson [GJPW18] showed that $\mathrm{R}(\mathrm{AND}_n \circ g) = \Omega(n\,\mathrm{R}(g))$, which can be generalized to $\mathrm{R}(f \circ g) = \Omega(\mathrm{s}(f)\,\mathrm{R}(g))$, where $\mathrm{s}(f)$ denotes the sensitivity of f.

Extremely recently, in work concurrent with this one, Bassilakis, Drucker, Göös, Hu, Ma, and Tan [BDG+20] showed that $\mathrm{R}(f \circ g) = \Omega(\mathrm{fbs}(f)\,\mathrm{R}(g))$, where $\mathrm{fbs}(f)$ is the fractional block sensitivity of f. (This result also follows from our independent work in this paper.)

A relational counterexample to composition: Gavinsky, Lee, Santha, and Sanyal [GLSS19] showed that the randomized composition conjecture is false when f is allowed to be a *relation*. Relations are generalizations of partial functions, in which f has non-Boolean output alphabet and there can be multiple allowed outputs for each input string. The authors exhibited a family of relations f_n and a family of partial functions g_n such that $\mathrm{R}(f_n) = \Theta(\sqrt{n})$, $\mathrm{R}(g_n) = \Theta(n)$, but $\mathrm{R}(f_n \circ g_n) = \Theta(n) \ll n^{3/2}$.

This counterexample of Gavinsky, Lee, Santha, and Sanyal does not directly answer the randomized composition conjecture (which usually refers to Boolean functions only), but it does place restrictions on the types of tools which might prove it true, since it appears that most or all of the composition theorems mentioned above do not use the fact that f has Boolean outputs and apply equally well when f is a relation—meaning those techniques cannot be used to prove the composition conjecture true without major new ideas.

B. Our results

Our first result shows that the randomized composition conjecture is false for partial functions.

Theorem 1. *There is a family of partial Boolean functions f_n and a family of partial Boolean functions g_n such that $\mathrm{R}(f_n) \to \infty$ and $\mathrm{R}(g_n) \to \infty$ as $n \to \infty$, but*

$$\mathrm{R}(f_n \circ g_n) = O\left(\mathrm{R}(f_n)^{2/3}\,\mathrm{R}(g_n)^{2/3}\log^{2/3}\mathrm{R}(f_n)\right).$$

In this counterexample, $\mathrm{R}(f \circ g)$ is polynomially smaller than what it was conjectured to be in the randomized composition conjecture. However, this counterexample actually uses functions f and g for which $\mathrm{R}(f)$ and $\mathrm{R}(g)$ are logarithmic in the input size of f. Therefore, the following slight weakening of the original randomized composition conjecture is still viable.

Conjecture 2. *For all partial Boolean functions f and g,*

$$\mathrm{R}(f \circ g) = \Omega\left(\frac{\mathrm{R}(f)\,\mathrm{R}(g)}{\log n}\right),$$

where n is the input size of f.

Hence, even for partial functions, the composition story is far from complete. This is in contrast to the setting in which f is a relation, where in the counterexample of [GLSS19], the query complexity $\mathrm{R}(f \circ g)$ is smaller than $\mathrm{R}(f)\,\mathrm{R}(g)$ by a polynomial factor even relative to the input size.

Our second contribution is a new composition theorem for randomized algorithms with a loss only in terms of f.

Theorem 3. *For all partial functions f and g,*

$$\mathrm{R}(f \circ g) = \Omega(\mathrm{noisyR}(f)\,\mathrm{R}(g)).$$

Here $\mathrm{noisyR}(f)$ is a measure we introduce, which is defined as the cost of computing f when given noisy oracle access to the input bits; for a full definition, see the full version of the paper. As it turns out, $\mathrm{noisyR}(f)$ has a very natural interpretation, as the following theorem shows.

Theorem 4. *For all partial functions f, we have*

$$\mathrm{noisyR}(f) = \Theta\left(\frac{\mathrm{R}(f \circ \textsc{GapMaj}_n)}{n}\right),$$

where n is the input size of f and $\textsc{GapMaj}_n$ is the majority function on n bits with the promise that the Hamming weight of the input is either $\lceil \frac{n}{2} + \sqrt{n} \rceil$ or $\lfloor \frac{n}{2} - \sqrt{n} \rfloor$. Note that $\mathrm{R}(\textsc{GapMaj}_n) = \Theta(n)$.

In other words, $\mathrm{noisyR}(f)$ characterizes the cost of computing f when the inputs to f are given as $\sqrt{n}$-gap majority instances (divided by n, so that $\mathrm{noisyR}(f) \le \mathrm{R}(f)$). This means that our composition theorem reduces the randomized composition problem on arbitrary f and g to the randomized composition problem of f with $\textsc{GapMaj}_n$.

Corollary 5. *For all partial functions f and g, we have*

$$\mathrm{R}(f \circ g) = \Omega\left(\frac{\mathrm{R}(f \circ \textsc{GapMaj}_n)}{\mathrm{R}(\textsc{GapMaj}_n)} \cdot \mathrm{R}(g)\right),$$

where n is the input size of f.

These results hold even when f is a relation. We also note that the counterexamples to composition theorems—the one for partial functions in Theorem 1 and the relational one in [GLSS19]—use the same function $\textsc{GapMaj}$ as the inner function g (or close variants of it). Therefore, there is a strong sense in which $g = \textsc{GapMaj}$ function is the only interesting case for studying the randomized composition behavior of $\mathrm{R}(f \circ g)$.

Next, we observe that our composition theorem is the strongest possible theorem of the form $\mathrm{R}(f \circ g) = \Omega(M(f)\,\mathrm{R}(g))$ for any complexity measure M of f. Formally, we have the following.

Lemma 6. *Let $M(\cdot)$ be any positive-real-valued measure of Boolean functions. Suppose that for all (possibly partial) Boolean functions f and g, we have $\mathrm{R}(f \circ g) = \Omega(M(f)\,\mathrm{R}(g))$. Then for all f, we have $\mathrm{noisyR}(f) = \Omega(M(f))$.*

Proof: By Theorem 4, we have

$$n \cdot \mathrm{noisyR}(f) = \Omega\big(\mathrm{R}(f \circ \textsc{GapMaj}_n)\big),$$

where n in the input size of f. Now, by our assumption on $M(\cdot)$, taking $g = \textsc{GapMaj}_n$ we obtain

$$\mathrm{R}(f \circ \textsc{GapMaj}_n) = \Omega\big(M(f)\,\mathrm{R}(\textsc{GapMaj}_n)\big) = \Omega\big(M(f) \cdot n\big).$$

Hence $\mathrm{noisyR}(f) = \Omega(M(f))$, as desired. ∎

The natural next step is to study the measure $\mathrm{noisyR}(f) = \mathrm{R}(f \circ \textsc{GapMaj}_n)/n$. We observe in the full version of the paper that $\mathrm{noisyR}(f) = \Omega(\mathrm{fbs}(f))$. However, we believe that a much stronger lower bound should be possible. The following conjecture is equivalent to Conjecture 2.

Conjecture 7 (Equivalent to Conjecture 2)**.** *For all (possibly partial) Boolean functions f,*

$$\mathrm{noisyR}(f) = \Omega\left(\frac{\mathrm{R}(f)}{\log n}\right).$$

The equivalence of the two conjectures follows from Theorem 3 in one direction, and from Lemma 6 in the other direction (taking $M(f) = \mathrm{R}(f)/\log n$).

One major barrier for proving Conjecture 7 is that it is false for relations. Indeed, the family of relations f from [GLSS19] has $\mathrm{noisyR}(f) = O(1)$ and $\mathrm{R}(f) = \Omega(\sqrt{n})$. Any lower bound $M(\cdot)$ for $\mathrm{noisyR}(\cdot)$ must therefore either be specific to functions (and not work for relations), or else must satisfy $M(f) = O(1)$ for that family of relations, even though $\mathrm{R}(f) = \Omega(\sqrt{n})$ (which means $M(f)$ is a poor lower bound on $\mathrm{R}(f)$, at least for some relations).

We are able to overcome this "relational barrier" for proving $\mathrm{noisyR}(f)$ lower bounds in the setting of non-adaptive algorithms. Let $\mathrm{R}^{\mathrm{NA}}(f)$ denote the non-adaptive randomized query complexity of f and let $\mathrm{noisyR}^{\mathrm{NA}}(f)$ denote the non-adaptive version of $\mathrm{noisyR}(f)$. Then for

the family of relations f from [GLSS19], it is still the case that $\mathrm{noisyR}^{\mathrm{NA}}(f) = O(1)$ and $\mathrm{R}^{\mathrm{NA}}(f) = \Omega(\sqrt{n})$. Despite this relational barrier, we have the following theorem for the non-adaptive setting.

Theorem 8. *For all (possibly partial) Boolean functions f, we have* $\mathrm{noisyR}^{\mathrm{NA}}(f) = \Theta(\mathrm{R}^{\mathrm{NA}}(f))$.

Since Theorem 8 is false for relations, its proof necessarily "notices" whether f is a relation or a partial function. Such proofs are unusual in query complexity. We hope that the techniques we used in the proof of Theorem 8 will assist future work in settling the relationship between $\mathrm{noisyR}(f)$ and $\mathrm{R}(f)$ (perhaps resolving Conjecture 7).

C. Our techniques

1) Main idea for the counterexample: The main idea for the counterexample to composition is to take $g = \mathrm{GAPMAJ}_m$ and to construct a function f that only requires some of its bits to be computed to bias $1/\sqrt{m}$ instead of exactly. Achieving bias $1/\sqrt{m}$ will be disproportionately cheap for an input to $f \circ g$ compared to an input to f.

This is the same principle used for the relational counterexample of [GLSS19]. There, the authors took f to be the relational problem of taking an input $x \in \{0,1\}^n$ and returning an output $y \in \{0,1\}^n$ with the property that $|x - y| \leq n/2 - \sqrt{n}$. This can be done using either $\sqrt{n}$ exact queries to x, or using n queries to x with bias $1/\sqrt{n}$ each. When f is composed with g and $n = m$, it's not hard to verify that $\mathrm{R}(f \circ g) = O(n)$, even though $\mathrm{R}(f) = \Omega(\sqrt{n})$ and $\mathrm{R}(g) = \Omega(m) = \Omega(n)$.

To convert f into a partial Boolean function, we use the indexing trick. We let the first m bits of f represent a string x, and we want to force an algorithm to find a string y that's within Hamming weight $m/2 - \sqrt{m}$ of x. To do so, we can try adding an array of length 2^m to the input of f, with entries indexed by y. We'll fill the array with $*$ on positions indexed by strings y that are far from x. On positions corresponding to strings y within $m/2 - \sqrt{m}$ of x, we'll put either all 0s or all 1s, and we'll require the algorithm to output 0 in the former case and 1 in the latter case (promised one of the two cases hold).

The above construction doesn't quite work, because a randomized algorithm can cheat: instead of finding a string y close to x, it can simply search the array for a non-$*$ bit and output that bit. Since a constant fraction of the Boolean hypercube is within $m/2 - \sqrt{m}$ of x, this strategy will succeed after a constant number of queries. To fix this, all we need to do is increase the gap from $\sqrt{m}$ to $10\sqrt{m \log m}$, so that y is required to be within $m/2 - 10\sqrt{m \log m}$ of x. Now the non-$*$ positions in the array will fill only a $1/m^{\Omega(1)}$ fraction of the array, and a randomized algorithm has no hope of finding one of those positions with a small number of random guesses. The input size of f will be $n = m + 2^m$. Then we have $\mathrm{R}(f) = \Theta(\sqrt{m \log m}))$, $\mathrm{R}(g) = \Theta(m)$, but $\mathrm{R}(f \circ g) = \Theta(m \log m)$ as we can solve $f \circ g$ by querying each of the first m copies of g $O(\log m)$ times each, getting bias $\Omega(\sqrt{(\log m)/m})$ for each of the m bits of x, which provides a good string y with high probability.

2) Main idea for the composition theorem: The main idea for proving the composition theorem $\mathrm{R}(f \circ g) = \Omega(\mathrm{noisyR}(f)\,\mathrm{R}(g))$ is to try to turn an algorithm for $f \circ g$ into an algorithm for f. This is the standard approach for most composition theorems, and the main question becomes how to solve f when we only have an algorithm A which makes queries to an nm-length input for $f \circ g$. When the algorithm queries bit j inside copy i of g, and we only have an n-bit input x to f, what do we query?

One solution would be to fix hard distributions μ_0 and μ_1 for g, and then, when A makes a query to bit j inside copy i of g, we can query x_i, sample an m-bit string from μ_{x_i}, and then return the j-th bit of that string. However, this uses a lot of queries: in the worst case, one query to x would be needed for each query A makes, giving only the upper bound $\mathrm{R}(f) \leq \mathrm{R}(f \circ g)$ instead of something closer to $\mathrm{R}(f) \leq \mathrm{R}(f \circ g)/\mathrm{R}(g)$. The goal is to simulate the behavior of A while avoiding making queries to x as much as possible.

One insight (also used in previous work) is that if bit j is queried inside copy i of g, we only need to query x_i from the real input x if μ_0 and μ_1 disagree on the j-th bit with substantial probability. In [GLSS19], the approach was to first try to generate the answer j from μ_0 and μ_1, and see if they happen to agree; this way, querying the real input x_i is only needed in case they disagree.

We do something slightly different: we assume we have access to a (very) noisy oracle for x_i, and use calls to the oracle to generate bit j from μ_{x_i} without actually finding out x_i. In effect, this lets us use the squared-Hellinger distance between the marginal distributions $\mu_0|_j$ and $\mu_1|_j$ as the cost of generating the sample, instead of using the total variation distance between $\mu_0|_j$ and $\mu_1|_j$. That is, we charge a cost for the noisy oracle calls in a special way, which ensures that the total cost of the noisy oracle calls will be proportional to the squared-Hellinger distance between the transcript of A when run on μ_0 and when run on μ_1. In other words, the cost our $\mathrm{R}(f)$ algorithm pays for simulating A will be proportional to how much A solved the copies of g, as tracked by the Hellinger distance of the transcript of A (i.e. its set of queries and query answers) on μ_0 vs. μ_1. It turns out this way of tracking the progress of A in solving g is tight, at least for the appropriate choice of hard distributions μ_0 and μ_1 for g. Therefore, this will give us an algorithm for f that has only $\mathrm{R}(f \circ g)/\mathrm{R}(g)$ cost, though this algorithm for f will require noisy oracles for the bits of the input—that is to say, it will be a $\mathrm{noisyR}(f)$ algorithm instead of an $\mathrm{R}(f)$ algorithm.

One wrinkle is that the hard distribution produced by Yao's minimax theorem is not sufficient to give the hardness guarantee we will need from μ_0 and μ_1. Roughly speaking,

we will need μ_0 and μ_1 to be such that distinguishing them with squared-Hellinger distance ϵ requires at least $\Omega(\epsilon \mathrm{R}(g))$ queries, uniformly across all choices of ϵ. To get such a hard distribution, we use our companion paper [BB20]. The concurrent work of [BDG+20] also gives a sufficiently strong hard distribution for g (though it is phrased somewhat differently).

3) Noisy oracle model: The noisy oracle model we will use is the following. There is a hidden bit $b \in \{0,1\}$ known to the oracle. The oracle will accept queries with any parameter $\gamma \in [0,1]$, and will return a bit $\tilde{b}$ that has bias γ towards b—that is, a bit from $\mathrm{Bernoulli}\left(\frac{1-(-1)^b\gamma}{2}\right)$ (independently sampled for each query call). This oracle can be called any number of times with possibly different parameters, but each call with parameter γ costs γ^2. (The cost γ^2 is a natural choice, as it would take $O(1/\gamma^2)$ bits of bias γ to determine the bit with constant error.)

The measure $\mathrm{noisyR}(f)$ is defined as the cost of computing f (to worst-case bounded error) using noisy oracle access to each bit in the input of x. That is, instead of receiving query access to the n-bit string x, we now have access to n noisy oracles, one for each bit x_i of x. We can call each oracle with any parameter γ of our choice, at the cost of γ^2 per such call. The goal is to compute f to bounded error using minimum expected cost (measured in the worst case over inputs x). We note that by using $\gamma = 1$ each time, this reverts to the usual query complexity of f, meaning that $\mathrm{noisyR}(f) \leq \mathrm{R}(f)$.

The key to our composition theorem lies in using such a noisy oracle for a bit x_i to generate a sample from a distribution $\mu_{x_i}|_j$ (distribution μ_{x_i} marginalized to bit j) without learning x_i. More generally, suppose we have two distributions, p_0 and p_1, and we wish to sample from one of them, but we don't know which one. The choice of which distribution to sample from depends on a hidden bit b, and we have noisy oracle access to b. Suppose we know that p_0 and p_1 are close, say $\mathrm{h}^2(p_0,p_1) = \epsilon$. How many queries to this noisy oracle do we need to make in order to generate this sample?

We show that using such noisy oracle calls, we can return a sample from p_b with an expected cost of $O(\mathrm{h}^2(p_0,p_1))$. When p_0 and p_1 are close, this is a much lower cost than the $\Omega(1)$ cost of extracting b. In other words, when the distributions are close, we can return a sample from p_b (without any error) without learning the value of the bit b! This is the key insight that allows our composition result to work.

4) Main idea for characterizing $\mathrm{noisyR}(f)$*:* In order to show that $\mathrm{noisyR}(f) = \Theta(\mathrm{R}(f \circ \textsc{GapMaj}_n)/n)$, we first note that the upper bound follows from our composition theorem: that is, $\mathrm{R}(f \circ \textsc{GapMaj}_n) = \Omega(\mathrm{noisyR}(f)\,\mathrm{R}(\textsc{GapMaj}_n))$, and $\mathrm{R}(\textsc{GapMaj}_n) = \Theta(n)$. For the lower bound direction, we need to convert a $\mathrm{noisyR}(f)$ algorithm (which makes noisy oracle calls to the input bits, with cost γ^2 for a noisy oracle call with parameter γ) into an algorithm for $\mathrm{noisyR}(f \circ \textsc{GapMaj}_n)$ where each query costs $1/n$. Recalling that $\textsc{GapMaj}_n$ is the majority function with the promise that the Hamming weight of the input is $n/2 \pm \lfloor\sqrt{n}\rfloor$, it's not hard to see that a single random query to a $\textsc{GapMaj}_n$ gadget (with cost $1/n$ each) is the same thing as a noisy oracle query with $\gamma \approx 1/\sqrt{n}$. Also, querying all n bits in a $\textsc{GapMaj}_n$ (with cost 1 in total) is the same thing as a noisy oracle query with $\gamma = 1$.

To finish the argument, all we have to show is that a $\mathrm{noisyR}(f)$ algorithm can always be assumed to make only queries with $\gamma = 1/\sqrt{n}$ or $\gamma = 1$. Now, it is well-known that an oracle with bias γ can be amplified to an oracle with bias $\gamma' > \gamma$ by calling it $O(\gamma'^2/\gamma^2)$ times and taking the majority of the answers. Since oracle calls with parameter γ cost us γ^2, this fact ensures that we only need to make noisy oracle calls with parameter either $\gamma = \hat{\gamma}$ or $\gamma = 1$, where $\hat{\gamma}$ is extremely small – smaller than anything used by an optimal (or at least near-optiomal) $\mathrm{noisyR}(f)$ algorithm. This is because for any desired bias level larger than $\hat{\gamma}$, we could simply amplify the $\hat{\gamma}$ calls.

Hence it only remains to show how to simulate noisy oracle queries with an arbitrarily small parameter $\hat{\gamma}$ using noisy oracle queries with parameter $1/\sqrt{n}$. For this, we consider a random walk on a line that starts at 0 and flips a $\mathrm{Bernoulli}\left(\frac{1-(-1)^b\hat{\gamma}}{2}\right)$ coin when deciding whether step forwards or backwards. Consider making this walk starting at 0, walking until either k or $-k$ is reached, and then stopping (where k is some fixed integer). Note that the probability that neither k or $-k$ is ever reached after infinitely many steps is 0. We then make the following key observation: the probability distribution over the sequence steps of this walk, *conditioned* on reaching k before $-k$, is the same whether $b = 0$ or $b = 1$. Therefore, it is possible to generate the full walk by generating the sequence of multiples of k the walk will reach (in a way that depends on b), and then completely separately – and independently of b – generating the sequence of steps between one multiple and the next, up to negation.

To simulate a bias $\hat{\gamma}$ oracle with a bias $1/\sqrt{n}$ oracle, we can use latter to generate the sequence of multiples of k described above, with $k = O(1/(\sqrt{n}\hat{\gamma}))$. We generate this sequence one at a time. For each one, we can then generate ℓ calls to the bias $\hat{\gamma}$ oracle, where ℓ is the (random) number of steps the random walk takes to go from one multiple of k to the next. This simulation is perfect: is produces the distribution of any number of calls to the $\hat{\gamma}$-bias oracle. It also turns out to use the right number of noisy oracle queries in the long run. The only catch is that if the algorithm makes only one noisy oracle call with bias $\hat{\gamma}$, this still requires one call to the oracle of bias $1/\sqrt{n}$, at a cost of $1/n$ instead of $1/\hat{\gamma}^2$. Since there are n total bits, this means the simulation can suffer an additive cost of 1. To complete the argument,

we then show that $\mathrm{noisyR}(f) = \Omega(1)$ for every non-constant Boolean function f.

5) Main idea for bypassing the relational barrier in the non-adaptive setting: The trick for showing $\mathrm{noisyR}^{\mathrm{NA}}(f) = \Theta(\mathrm{R}^{\mathrm{NA}}(f))$ for partial functions is to use an information-theoretic characterization of this statement. First, using a Yao-style minimax theorem, we can assume we are working against a hard distribution μ for $\mathrm{R}^{\mathrm{NA}}(f)$. Then we consider a non-adaptive randomized algorithm that uses noisy oracle queries (that is, a $\mathrm{noisyR}^{\mathrm{NA}}(f)$ algorithm) that solves f against μ. By some simple modifications and reductions, we can assume that this algorithm simply makes one noisy query to each bit of the input, with bias parameter $1/\sqrt{n}$. In other words, if X is the random variable for a string sampled from μ, and if Y is the random variable we get by flipping each bit of X independently with probability $(1-1/\sqrt{n})/2$, then we can assume a $\mathrm{noisyR}^{\mathrm{NA}}(f)$ algorithm just has access to the string Y and tries to compute $f(X)$ using Y. Our reductions change the length of the string (by duplicating bits of the input), and the cost of this noisy randomized algorithm will roughly be $|X|/n$, where $|X|$ is the length of the string X and n is the length of the original string.

What we wish to show is that such a noisy non-adaptive randomized algorithm (which computes $f(X)$ using Y) can be converted into a regular non-adaptive randomized algorithm which computes $f(X)$ by querying only around $|X|/n$ bits of X. To do so, we use a theorem of Samorodnitsky [Sam16; PW17], which states that the erasure channel with parameter ρ^2 – which deletes each bit of X with probability $1-\rho^2$ – preserves more information about any function $f(X)$ than the noisy channel with parameter ρ (which flips each bit of X with probability $(1-\rho)/2$). Hence, if $f(X)$ can be computed from Y, it can also be computed from the string Z which is formed by deleting each bit of X with probability $1-1/n$. Since Z reveals only $|X|/n$ bits on expectation, this can be used to define a non-adaptive randomized algorithm whose cost is at most $\mathrm{noisyR}^{\mathrm{NA}}(f)$, and which still succeeds in computing f against μ to bounded error. This shows $\mathrm{R}^{\mathrm{NA}}(f) = O(\mathrm{noisyR}^{\mathrm{NA}}(f))$.

We note that the step where we used the fact that f is a partial function is the step where we said that if Z gives *information* about $f(X)$, seeing Z can be used to *compute* $f(X)$ to bounded error. This statement holds when $f(X)$ is a Boolean-valued random variable, but it has no good analogue in the relational setting (and indeed, we know that $\mathrm{noisyR}^{\mathrm{NA}}(f)$ does not equal $\mathrm{R}^{\mathrm{NA}}(f)$ for relations).

References

[Aar08] Scott Aaronson. "Quantum certificate complexity". In: *Journal of Computer and System Sciences* (2008). Previous version in CCC 2003. DOI: 10.1016/j.jcss.2007.06.020. arXiv: 1506.04719.

[AGJ+17] Anurag Anshu, Naresh B. Goud, Rahul Jain, Srijita Kundu, and Priyanka Mukhopadhyay. "Lifting randomized query complexity to randomized communication complexity". Preprint, retracted, 2017. arXiv: 1703.07521.

[BB19] Eric Blais and Joshua Brody. "Optimal Separation and Strong Direct Sum for Randomized Query Complexity". In: *Proceedings of the 34th Conference on Computational Complexity (CCC)*. 2019. DOI: 10.4230/LIPICS.CCC.2019.29. arXiv: 1908.01020.

[BB20] Shalev Ben-David and Eric Blais. "A New Minimax Theorem for Randomized Algorithms". Preprint, 2020. arXiv: 2002.10809.

[BDG+20] Andrew Bassilakis, Andrew Drucker, Mika Göös, Lunjia Hu, Weiyun Ma, and Li-Yang Tan. "The Power of Many Samples in Query Complexity". In: *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*. 2020. DOI: 10.4230/LIPIcs.ICALP.2020.9. arXiv: 2002.10654.

[BK18] Shalev Ben-David and Robin Kothari. "Randomized Query Complexity of Sabotaged and Composed Functions". In: *Theory of Computing* (2018). Previous version in ICALP 2016. DOI: 10.4086/toc.2018.v014a005. arXiv: 1605.09071.

[BW02] Harry Buhrman and Ronald de Wolf. "Complexity measures and decision tree complexity: a survey". In: *Theoretical Computer Science* (2002). DOI: 10.1016/S0304-3975(01)00144-X.

[Dru12] Andrew Drucker. "Improved direct product theorems for randomized query complexity". In: *Computational Complexity* (2012). Previous version in CCC 2011. DOI: 10.1007/s00037-012-0043-7. arXiv: 1005.0644.

[DZ91] Persi Diaconis and Sandy Zabell. "Closed form summation for classical distributions: variations on a theme of de Moivre". In: *Statistical Science* (1991). DOI: 10.1214/ss/1177011699.

[Fel57] William Feller. *An Introduction to Probability Theory and Its Applications*. Vol. 1. John Wiley & Sons, Inc., 1957. ISBN: 978-0-471-25708-0. URL: https://archive.org/details/AnIntroductionToProbabilityTheoryAndItsApplicationsVolume1.

[GJ16] Mika Göös and T. S. Jayram. "A composition theorem for conical juntas". In: *Proceedings of the 31st Conference on Computational Complexity (CCC)*. 2016. DOI: 10.4230/LIPIcs.CCC.2016.5. ECCC: 2015/167.

[GJPW18] Mika Göös, T. S. Jayram, Toniann Pitassi, and Thomas Watson. "Randomized Communication versus Partition Number". In: *ACM Transactions on Computation Theory* (2018). Previous version in ICALP 2017. DOI: 10.1145/3170711. ECCC: 2015/169.

[GLSS19] Dmitry Gavinsky, Troy Lee, Miklos Santha, and Swagato Sanyal. "A Composition Theorem for Randomized Query Complexity via Max-Conflict Complexity". In: *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*. 2019. DOI: 10.4230/LIPICS.ICALP.2019.64. arXiv: 1811.10752.

[GSS16] Justin Gilmer, Michael Saks, and Sudarshan Srinivasan. "Composition limits and separating examples for some Boolean function complexity measures". In: *Combinatorica* (2016). Previous version in CCC 2013. DOI: 10.1007/s00493-014-3189-x. arXiv: 1306.0630.

[HR70] M. Hellman and J. Raviv. "Probability of error, equivocation, and the Chernoff bound". In: *IEEE Transactions on Information Theory* (1970). DOI: 10.1109/tit.1970.1054466.

[JKS10] Rahul Jain, Hartmut Klauck, and Miklos Santha. "Optimal direct sum results for deterministic and randomized decision tree complexity". In: *Information Processing Letters* (2010). DOI: 10.1016/j.ipl.2010.07.020. arXiv: 1004.0105.

[Kim13] Shelby Kimmel. "Quantum Adversary (Upper) Bound". In: *Chicago Journal of Theoretical Computer Science* (2013). Previous version in ICALP 2012. DOI: 10.4086/cjtcs.2013.004. arXiv: 1101.0797.

[KT16] Raghav Kulkarni and Avishay Tal. "On Fractional Block Sensitivity". In: *Chicago Journal of Theoretical Computer Science* (2016). DOI: 10.4086/cjtcs.2016.008. ECCC: 2013/168.

[LMR+11] Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. "Quantum query complexity of state conversion". In: *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 2011. DOI: 10.1109/FOCS.2011.75. arXiv: 1011.3020.

[MCAL17] Marianthi Markatou, Yang Chen, Georgios Afendras, and Bruce G. Lindsay. "Statistical Distances and Their Role in Robustness". In: *New Advances in Statistics and Data Science* (2017). DOI: 10.1007/978-3-319-69416-0_1. arXiv: 1612.07408.

[Mon14] Ashley Montanaro. "A composition theorem for decision tree complexity". In: *Chicago Journal of Theoretical Computer Science* (2014). DOI: 10.4086/cjtcs.2014.006. arXiv: 1302.4207.

[PW17] Yury Polyanskiy and Yihong Wu. "Strong Data-Processing Inequalities for Channels and Bayesian Networks". In: *Convexity and Concentration*. 2017. DOI: 10.1007/978-1-4939-7005-6_7. arXiv: 1508.06025.

[Rei11] Ben W. Reichardt. "Reflections for quantum query algorithms". In: *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*. 2011. DOI: 10.1137/1.9781611973082.44. arXiv: 1005.1601.

[Sam16] Alex Samorodnitsky. "On the Entropy of a Noisy Function". In: *IEEE Transactions on Information Theory* (2016). DOI: 10.1109/tit.2016.2584625. arXiv: 1508.01464.

[Sha03] Ronen Shaltiel. "Towards proving strong direct product theorems". In: *Computational Complexity* (2003). Previous version in CCC 2001. DOI: 10.1007/s00037-003-0175-x. ECCC: 2001/009.

[Sta01] Pantelimon Stanica. "Good lower and upper bounds on binomial coefficients". In: *Journal of Inequalities in Pure and Applied Mathematics* (2001). URL: https://eudml.org/doc/121842.

[Tal13] Avishay Tal. "Properties and Applications of Boolean Function Composition". In: *Proceedings of the 4th Innovations in Theoretical Computer Science Conference (ITCS)*. 2013. DOI: 10.1145/2422436.2422485. ECCC: 2012/163.

[Tøp00] Flemming Tøpsoe. "Some inequalities for information divergence and related measures of discrimination". In: *IEEE Transactions on Information Theory* (2000). DOI: 10.1109/18.850703.