Approximate Degree Composition for Recursive Functions

Sourav Chakraborty
Indian Statistical Institute, Kolkata, India

Chandrima Kayal
Indian Statistical Institute, Kolkata, India

Rajat Mittal
Indian Institute of Technology Kanpur, India

Manaswi Paraashar
University of Copenhagen, Denmark

Nitin Saurabh
Indian Institute of Technology Hyderabad, India

Abstract

Determining the approximate degree composition for Boolean functions remains a significant unsolved problem in Boolean function complexity. In recent decades, researchers have concentrated on proving that approximate degree composes for special types of inner and outer functions. An important and extensively studied class of functions are the recursive functions, i.e. functions obtained by composing a base function with itself a number of times. Let h^d denote the standard d-fold composition of the base function h. The main result of this work is to show that the approximate degree composes if either of the following conditions holds:

- The outer function $f : \{0,1\}^n \to \{0,1\}$ is a recursive function of the form h^d, with h being any base function and $d = \Omega(\log\log n)$.
- The inner function is a recursive function of the form h^d, with h being any constant arity base function (other than AND and OR) and $d = \Omega(\log\log n)$, where n is the arity of the outer function.

In terms of proof techniques, we first observe that the lower bound for composition can be obtained by introducing majority in between the inner and the outer functions. We then show that majority can be *efficiently eliminated* if the inner or outer function is a recursive function.

2012 ACM Subject Classification Theory of computation

Keywords and phrases Approximate degree, Boolean function, Composition theorem

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2024.71

Category RANDOM

Related Version *Full Version*: `https://arxiv.org/abs/2407.08385` [16]

Funding *Sourav Chakraborty*: supported by the Science & Engineering Research Board of the DST, India, through the MATRICS grant MTR/2021/000318.
Manaswi Paraashar: supported by ERC grant (QInteract, Grant Agreement No 101078107).
Nitin Saurabh: supported by the seed grant (SG/IITH/F285/2022-23/SG-123) from IIT Hyderabad.

1 Introduction

Representations of Boolean functions $f\colon \{0,1\}^n \to \{0,1\}$ in terms of multivariate polynomials $p(x)$ play a pivotal role in theoretical computer science. There are different notions of representations;

- *exact* representation: $f(x) = p(x)$ for all $x \in \{0,1\}^n$,
- *approximate* representation: $|f(x) - p(x)| \leq 1/3$ for all $x \in \{0,1\}^n$, and
- *sign* representation: $(1 - 2f(x))p(x) > 0$ for all $x \in \{0,1\}^n$.

Arguably the most important measure associated with a polynomial is its (total) *degree*. Let $\deg(f)$, $\widetilde{\deg}(f)$, and $\deg_{\pm}(f)$ denote the minimal possible degree of a real polynomial *exactly*, *approximately*, and *sign* representing f, respectively. These different notions of degrees capture notions of efficiency in many different models of computation (e.g., decision trees, quantum query, perceptrons), and are thus well-studied in literature (see, e.g., [6, 7, 14] and the references therein).

For instance, $\deg_{\pm}(f)$ (called sign degree) has strong connections to – separations among complexity classes [7], designing efficient learning algorithm [28, 27], and lower bounds against circuits, formulas, communication complexity, etc. [12, 18]. Similarly, upper bounds on $\widetilde{\deg}(f)$ (called approximate degree), has strong connections to learning theory [25, 29, 37], approximate inclusion-exclusion [24, 43], differentially private data release [47, 17], etc. While the lower bounds on approximate degree lead to lower bounds in quantum query complexity [5, 2, 1], communication complexity [43, 38], circuit complexity [3], etc.

Despite decades of work in this area, there are many important problems that are yet to be resolved completely. One such problem pertains to the composition of approximate degrees. For any two Boolean functions $f\colon \{0,1\}^n \to \{0,1\}$ and $g\colon \{0,1\}^m \to \{0,1\}$, define the *composed* function $f \circ g\colon \{0,1\}^{nm} \to \{0,1\}$ as follows

$$f \circ g(x_{11}, \ldots, x_{1m}, \ldots\ldots, x_{n1}, \ldots, x_{nm}) = f(g(x_1), \ldots, g(x_n)),$$

where $x_i = (x_{i1}, \ldots, x_{im}) \in \{0,1\}^m$ for $i \in [n]$. The function f is called the outer function and g the inner function.

Investigating the behaviour of complexity measures under composition has been a quintessential tool in our quest to gain insights into relationships among different measures. In particular, composition has been used successfully on numerous occasions to show separations between various complexity measures associated with Boolean functions, see, e.g., [36, 33, 23, 4, 46, 19]. A big open problem in this context is to understand how approximate degree behaves under composition. More formally, it asks whether for all Boolean functions $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$,

$$\widetilde{\deg}(f \circ g) = \widetilde{\Theta}(\widetilde{\deg}(f) \cdot \widetilde{\deg}(g))?$$

The tilde in the $\widetilde{\Theta}$ notation hides a factor polynomial in $\log(n+m)$. This problem is often referred to as the "approximate degree composition" problem.

The upper bound, $\widetilde{\deg}(f \circ g) = O(\widetilde{\deg}(f) \cdot \widetilde{\deg}(g))$, was established in a seminal work [42] of Sherstov. Thus to completely resolve the problem it remains to prove a matching lower bound on the approximate degree of a composed function in terms of the approximate degree of the individual functions. In other words, does the following hold for all Boolean functions f and g,

$$\widetilde{\deg}(f \circ g) = \widetilde{\Omega}\left(\widetilde{\deg}(f) \cdot \widetilde{\deg}(g)\right)?$$

In this article we will refer to the aforementioned (lower bound) question by the phrase "approximate degree composition" problem.

Numerous works, including those by [33, 4, 39, 41, 40, 13, 8, 15], actively pursued these lower bounds, leading to newer connections with several important problems in the field. However, establishing the lower bound $\widetilde{\deg}(f \circ g) = \widetilde{\Omega}\left(\widetilde{\deg}(f)\, \widetilde{\deg}(g)\right)$ even for specific functions or restricted classes of functions is often very challenging. For example, consider the composed function $\mathsf{OR} \circ \mathsf{AND}$, it took a long series of work [33, 44, 4, 41, 40, 13] over nearly two decades to prove that $\widetilde{\deg}(\mathsf{OR} \circ \mathsf{AND}) = \Omega\left(\widetilde{\deg}(\mathsf{OR})\,\widetilde{\deg}(\mathsf{AND})\right)$. Till date we know that the approximate degree composes in the following cases:

- when the outer function f has full approximate degree, i.e., $\Theta(n)$ [39],
- when the outer function f is a symmetric function [8],
- when the outer function f has minimal approximate degree with respect to its block sensitivity, i.e., $\widetilde{\deg}(f) = O(\sqrt{\mathrm{bs}(f)})$ [15], and
- when the sign degree of the inner function is same as its approximate degree [39, 30].

This work focuses on the behavior of approximate degree when *recursive functions* are composed with other general functions (as outer or inner function). Here, by recursive functions, we mean the functions of the kind h^d (h composed with itself d times) where the arity of h is small. The function h is often called the base function and the function f is called the recursive-h function.

Recursive functions are an important class of Boolean functions that are studied in various different contexts in the analysis of Boolean functions, mainly in proving various lower bounds [4, 45, 36, 33, 34, 9]. For example, the Kushilevitz's function [34] which is the only known non-trivial example of functions with low degree and high sensitivity is a recursive function of a carefully chosen base function. Recursive majority, MAJ_3^d, is another recursive function that has been studied extensively in the literature for its different properties [36, 22, 31, 32]. Boppana (see, e.g., [36]) used it to provide the first evidence that the randomized query is more powerful than deterministic query [36]. In the same article, they show a similar separation using recursive $\mathsf{AND}_2 \circ \mathsf{OR}_2$ function too. In a different application of recursive $\mathsf{AND}_2 \circ \mathsf{OR}_2$, [23] show separation between deterministic tree-size complexity and number of monomials in the minimal DNF or CNF.

The approximate degree composition was not known when the outer or inner function is a recursive function, in general. For some special recursive functions, however, it was known that the approximate degree composes. For example, the OR function on $n = 3^d$ bits is same as OR_3^d. After a series of works ([33, 4, 40, 13, 41]), it was proven that the approximate degree composition holds when the outer function is OR, and in general symmetric [8]. Similarly, from the result of [39, 30] it can be observed that the lower bound holds when either the inner or outer function is recursive PARITY. Unfortunately, these results can't be applied in general even when the base function is symmetric or it has full approximate degree.

This scenario leads to the natural question:

Can we prove that $\widetilde{\deg}(f \circ g) = \Omega(\widetilde{\deg}(f) \cdot \widetilde{\deg}(g))$ when the outer function f or the inner function g is recursive?

1.1 Our Results

Let $h : \{0,1\}^k \to \{0,1\}$ be a function on k-bits. Let h^d denote the Boolean function represented by the complete k-ary tree of depth d such that each internal node of the tree is labelled by h and the leaves of tree are labelled by distinct variables. Our main result shows that the composition theorem holds for any h^d (except a few specific h's), either as the outer function with any inner function or as the inner function with any outer function.

▶ **Theorem 1.** *Let* $f\colon \{0,1\}^n \to \{0,1\}$ *and* $g\colon \{0,1\}^m \to \{0,1\}$ *be two Boolean functions and* $d \geq C \log\log n$ *for a large enough constant* C*. Then,*

$$\widetilde{\deg}(f \circ g) = \Omega\left(\frac{\widetilde{\deg}(f)\widetilde{\deg}(g)}{\mathrm{polylog}(n)}\right),$$

if either of the following conditions hold:
1. $f = h^d$*, for any Boolean function* h*.*
2. $g = h^d$*, for any Boolean function* h *with constant arity and not equal to* AND *or* OR*.*

In light of the above theorem, understanding the composition of approximate degree when inner function is OR is the central case for making progress towards the general composition question.

We would like to emphasize that there are not many results which prove composition theorem for a general class of inner functions. Theorem 1 shows that the composition property holds if the inner function is recursive irrespective of the outer function.

We further note that Theorem 1 doesn't follow from the known results even when the composition theorem is known to hold for the base function. Firstly, it is known that the composition lower bound holds when the outer function is symmetric [8]; though, a repeated composition of a symmetric function will incur the factor of $(\log n)^d$ (because of the $\log n$ factor hiding in the $\widetilde{\Omega}$ notation). Secondly, while the majority function, MAJ_n, has full approximate degree ($\Theta(n)$), MAJ_3^d doesn't have full approximate degree. Thus, Sherstov's result [39] that proves composition theorem holds for functions with full approximate degree cannot be applied in the case of recursive majority. The situation is similar for the inner function as well.

Moving ahead, the proof of Theorem 1 uses two ideas.
- We first prove that a similar theorem works for the specific case of $h = \mathsf{MAJ}_3$ and $h = \mathsf{AND}_2 \circ \mathsf{OR}_2$ functions.
- Then, we use a general h to *simulate* $\mathsf{AND}_2 \circ \mathsf{OR}_2$; hence, proving composition for the general case.

The case of recursive $h = \mathsf{MAJ}_3$ and $h = \mathsf{AND}_2 \circ \mathsf{OR}_2$ functions is in itself very interesting. There have been several works towards exploring the approximate degree and other properties of these two functions [21, 26, 36, 23]. Given their importance, and the fact that it is a central step in our main result (Theorem 1), we state the composition theorem for these two functions separately.

▶ **Theorem 2.** *Let* f *and* g *be two Boolean functions. Then,*

$$\widetilde{\deg}(f \circ h^d) = \widetilde{\Omega}(\widetilde{\deg}(f)\ \widetilde{\deg}(h^d)) \text{ and } \widetilde{\deg}(h^d \circ g) = \widetilde{\Omega}(\widetilde{\deg}(h^d)\ \widetilde{\deg}(g)),$$

where h *is either* $\mathsf{MAJ}_3 : \{0,1\}^3 \to \{0,1\}$ *or* $\mathsf{AND}_2 \circ \mathsf{OR}_2 : \{0,1\}^4 \to \{0,1\}$*,* n *is the arity of the outer function,* $d \geq C \log\log n$ *for a large enough constant* C*, and* $\widetilde{\Omega}(\cdot)$ *hides* $\mathrm{polylog}(n)$ *factors.*

To prove Theorem 2 we will need the following lemma. Even though the lemma can be obtained from a combination of known results (e.g., [39] and [10]) with appropriate parameters, we give a self-contained simpler proof of the lemma, inspired by the primal-dual perspective of [40].

▶ **Lemma 3.** *For any Boolean functions* $f : \{0,1\}^n \to \{0,1\}$ *and* $g : \{0,1\}^m \to \{0,1\}$*,*

$$\widetilde{\deg}(f \circ \mathsf{MAJ}_t \circ g) = \Omega(\widetilde{\deg}(f)\widetilde{\deg}(g)) \tag{1.1}$$

for $t \geq C \log n$ *for a large enough constant* C*.*

Note that, Lemma 3 gives a way to settle the composition question affirmatively. In particular, if $\widetilde{\deg}(f \circ \mathsf{MAJ}_t \circ g) = \widetilde{O}(\widetilde{\deg}(f \circ g))$, where t is $\Theta(\log n)$ and n is the arity of f, then it follows that the composition holds for f and g.

We also highlight that a tighter lower bound can be obtained when the middle function MAJ is replaced by an "amplifier function" in Lemma 3. Define H to be a *strong hardness amplifier function* for g if

$$\widetilde{\deg}_{\frac{1-2^{-\Omega(t)}}{2}}(H \circ g) = \Omega(\widetilde{\deg}(H) \circ \widetilde{\deg}(g)).$$

We also observe that,

$$\widetilde{\deg}(f \circ H \circ g) = \Omega(\widetilde{\deg}(f)\widetilde{\deg}(H)\widetilde{\deg}(g)), \tag{1.2}$$

when H is a strong hardness amplifier function for g. We discuss this improvement in the full version of the paper [16].

1.2 Proof Ideas

To address the lower bound for the composition of two Boolean functions f and g, $f \circ g$, we will call f to be the "outer function" and g to be the "inner function". In the case of three layered composed functions $(f \circ H \circ g)$, we will call H to be the "hardness amplifier" and f and g to be the outer and inner functions respectively.

Primal dual approach to composition. Our proof technique is based on the primal-dual view used by [40] for proving the composition of $\mathsf{AND}_n \circ \mathsf{OR}_n$. Here, instead of using "dual-composition method" (see [13, 14]) we will be using only the dual witness of the inner function. The primal-dual approach is to construct an approximating polynomial for f with smaller degree than $\widetilde{\deg}(f)$ by applying a linear operator L on the assumed approximating polynomial for $f \circ g$ (say p, with smaller degree than claimed), leading to a contradiction. The linear operator L is defined by taking the input to f, extending it to a probability distribution (which depends upon the dual of g) over the inputs of $f \circ g$ and outputting the expectation.

Let ψ be the dual witness of g, we get μ_0 and μ_1 by restricting ψ on support which takes positive and negative values respectively; by the properties of dual witness, μ_1 (and μ_0) will mostly be supported on inputs x such that $g(x) = 1$ (and $g(x) = 0$ respectively). The input to f is expanded bit by bit using μ_0 and μ_1, creating a distribution on inputs of $f \circ g$.

Formally, L takes a general function $h : \{0,1\}^{mn} \to \{0,1\}$ and gives $Lh : \{0,1\}^n \to \mathbb{R}$.

$$Lh(z_1, \dots, z_n) = \mathop{\mathbb{E}}_{x_1 \sim \mu_{z_1}} \mathop{\mathbb{E}}_{x_2 \sim \mu_{z_2}} \cdots \mathop{\mathbb{E}}_{x_n \sim \mu_{z_n}} [h(x_1, x_2, \dots, x_n)], \tag{1.3}$$

where $x_i \in \{0,1\}^m$ for all $i \in \{1, 2, \dots, n\}$.

To complete the proof, the following two properties of L are required:

1. Showing that the polynomial Lp indeed approximates f in l_∞ norm. Intuitively this happens because the restricted distributions (μ_0 and μ_1) are a pretty good indicator of the value of g.
2. The degree of Lp is small, intuitively because L reduces the degree of every monomial by a factor of $\widetilde{\deg}(g)$.

Problem with the primal dual approach. Unfortunately, the recipe described above doesn't work well in general due to the error introduced by the expectation over μ_0 and μ_1 in the string $(z_1, \ldots, z_n)$. To handle a noisy string in place of a Boolean string, the approximating polynomial p needs to be robust. A polynomial is robust to noise $\frac{1}{3}$, if for all inputs x and for all $\Delta \in \left[-\frac{1}{3}, \frac{1}{3}\right]^m$, $|p(x) - p(x+\Delta)| < \varepsilon$.

While any polynomial p can be made robust up to error ε with degree at most $\deg(p) + \log(\frac{1}{\varepsilon})$ (see Theorem 11 by [42]), such polynomials are not known to be multilinear, making the analysis of expectation difficult. [11] gives a robust multilinear polynomial for any Boolean function $f : \{0,1\}^n \to \{0,1\}$; though, the polynomial is defined on a perturbation matrix of input x instead of x itself. We now discuss how to overcome this problem.

We give the proof ideas of Theorem 1, Theorem 2 and Lemma 3 in the reverse order, the way they are obtained from each other.

Proof idea of Lemma 3. We will use MAJ_t to get past this difficulty; it helps to reduce the noise in the input of f to error $\frac{1}{n}$. Using the fact that any multilinear polynomial on n variables is robust up to error $\frac{1}{n}$, we have our lower bound for the function $\widetilde{\deg}(f \circ \mathsf{MAJ}_t \circ g)$ where $t = \Omega(\log n)$.

Proof idea of Theorem 2. Using previously known constructions ([48, 20]), $\mathsf{MAJ}_{\log n}$ can be projected to MAJ_3^d and $(\mathsf{AND}_2 \circ \mathsf{OR}_2)^d$, where $d \geq C \log\log n$. We now replace $\mathsf{MAJ}_{\log n}$ in Lemma 3 with these recursive functions; by using the associativity of the composition of functions and the approximate degree upper bound [42], we finish the proof of the theorem. Note that we only lose a factor of polylog(n) in the lower bound since we only need to simulate $\mathsf{MAJ}_{\log n}$.

Now we give the idea about how to replace $\mathsf{AND}_2 \circ \mathsf{OR}_2$ with almost any recursive function to get our main result.

Proof idea of Theorem 1. Given Theorem 2, it is natural to ask, what other recursive functions satisfy the composition property. We show that almost any h can be used to replace the $\mathsf{AND}_2 \circ \mathsf{OR}_2$ function. This is done by simulating AND_2 and OR_2 using restrictions of h and its powers. The proof of this simulation is divided into two cases: monotone h and non-monotone h.

For the monotone case (except when h is AND or OR): We show that both AND_2 and OR_2 will be present as sub-cubes of the original Boolean hypercube of h.

For the non-monotone case (except when h is PARITY or $\neg\mathsf{PARITY}$): The proof requires more work here because of these two issues. First, there need not be both functions AND_2 and OR_2 as sub-cubes (though, we show that at least one will be present). Second, the sub-cube could be rotated. The resolution to both these issues is same. We use the non-monotonicity to construct the negation function. This allows us to rotate the sub-cube as well as construct $\mathsf{AND}_2/\mathsf{OR}_2$ from the other one.

A slight technical point to note is that when h is a non-constant arity function and h^d is the inner function, then the loss in the lower bound will be larger than polylog(n). However, even for the case when the base function h has arity that is a "slowly" growing function of n we still obtain a non-trivial lower bound composition result.

The remaining cases of Theorem 1, that is,
(i) when f or g equals h^d for $h \in \{\mathsf{PARITY}, \neg\mathsf{PARITY}\}$ follows from [39], and (ii) when $f = h^d$ and $h \in \{\mathsf{AND}, \mathsf{OR}\}$ follows from [8].

2 Notations and Preliminaries

In this paper, we will assume a Boolean function has domain $\{0,1\}^n$ and range $\{0,1\}$. We start with some of the important definitions.

▶ **Definition 4** (Generalized Composition of functions). *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$ and n Boolean functions $g_1, g_2, \ldots, g_n$, define the composed function*

$$f \circ (g_1, g_2, \ldots, g_n)(x_1, x_2, \ldots, x_n) = f(g_1(x_1), g_2(x_2), \ldots, g_n(x_n)),$$

where g_i's can have different arities and $x_i \in \mathrm{Dom}(g_i)$ for all $i \in [n]$.

When all the copies of g_i are the same function g then the composed function is denoted by $f \circ g$.

▶ **Definition 5** (Recursive functions). *For any Boolean function $f : \{0,1\}^t \to \{0,1\}$ we define recursive function $f^d : \{0,1\}^{t^d} \to \{0,1\}$ by $f^d = \underbrace{f \circ f \circ \ldots \circ f}_{d \text{ times}}$.*

▶ **Definition 6** (Approximate degree ($\widetilde{\deg}$)). *For some constant $0 < \varepsilon < 1/2$, a polynomial $p : \mathbb{R}^n \to \mathbb{R}$ is said to ε-approximate a Boolean function $f : \{0,1\}^n \to \{0,1\}$ if for all $x \in \{0,1\}^n$, $|p(x) - f(x)| \le \varepsilon$. The ε-approximate degree of f, $\widetilde{\deg}_\varepsilon(f)$, is the minimum possible degree of a polynomial that ε-approximates f. Conventionally we use $\widetilde{\deg}(\cdot)$ as the shorthand for $\widetilde{\deg}_{1/3}(\cdot)$.*

Note that the constant ε in the above definition can be replaced by any constant strictly smaller than $1/2$ which changes $\widetilde{\deg}_\varepsilon(f)$ by only a constant factor. We note this well-known fact about error reduction.

▶ **Lemma 7** (Error reduction). *For any $\varepsilon > 0$, $\widetilde{\deg}_\varepsilon(f) = \Theta_\varepsilon(\widetilde{\deg}(f))$, where $\Theta_\varepsilon(\cdot)$ denotes that the constant in $\Theta(\cdot)$ depends on ε.*

▶ **Lemma 8** ([38, 39]). *Let $f : \{0,1\}^n \to \mathbb{R}$ be a function and $\varepsilon > 0$. Then, $\widetilde{\deg}_\varepsilon(f) \ge d$ iff there exists a function $\psi : \{0,1\}^n \to \mathbb{R}$ such that*

$$\sum_{x \in \{0,1\}^n} |\psi(x)| = 1, \tag{2.1}$$

$$\sum_{x \in \{0,1\}^n} \psi(x) \cdot f(x) > \varepsilon, \text{ and} \tag{2.2}$$

$$\sum_{x \in \{0,1\}^n} \psi(x) \cdot p(x) = 0 \quad \text{for every polynomial } p \text{ of degree} < d. \tag{2.3}$$

In a seminal work, Sherstov [42] showed that the approximate degree can increase at most multiplicatively under composition.

▶ **Theorem 9** ([42]). *For all Boolean function $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$, $\widetilde{\deg}(f \circ g) = O(\widetilde{\deg}(f) \cdot \widetilde{\deg}(g))$.*

At times we will be working with inputs that are not Boolean *but* are close to Boolean. So we would also need the following notion of *robust* approximating polynomials.

▶ **Definition 10** ((δ, ε)-robust approximating polynomial). *Let $p : \{0,1\}^m \to [0,1]$ be a polynomial. Then, for δ, $\varepsilon > 0$, a (δ, ε)-robust approximating polynomial for p is a polynomial $p_{robust} : \mathbb{R}^m \to \mathbb{R}$ such that for all $x \in \{0,1\}^m$ and for all $\Delta \in [-\delta, \delta]^m$,*

$$|p(x) - p_{robust}(x + \Delta)| < \varepsilon.$$

Sherstov [42] proved that for any Boolean function $f : \{0,1\}^n \to \{0,1\}$ there exists a robust approximating polynomial with degree at most $O(\widetilde{\deg}(f) + \log(1/\varepsilon))$.

▶ **Theorem 11** ([42]). *A (δ, ε)-robust approximating polynomial for $p : \{0,1\}^n \to [0,1]$ of degree $O_\delta(\deg(p) + \log(1/\varepsilon))$ exists. Here $O_\delta(\cdot)$ denotes that the constant in $O(\cdot)$ depends on δ.*

Note that a robust approximating polynomial need not to be multilinear. For our purposes, we need a multilinear robust approximating polynomial.

▶ **Theorem 12** (Folklore). *Any multilinear polynomial $p : \{0,1\}^n \to \{0,1\}$ is $\left(\frac{\delta}{n}, \delta\right)$-robust.*

A proof of the theorem above can be found at [11, Lemma 3].

▶ **Theorem 13** ([8]). *For any symmetric Boolean function $f : \{0,1\}^n \to \{0,1\}$ and any Boolean function $g : \{0,1\}^m \to \{0,1\}$,*

$$\widetilde{\deg}(f \circ g) = \Omega\left(\frac{\widetilde{\deg}(f)\widetilde{\deg}(g)}{\log n}\right).$$

Finally, we define projection of functions.

▶ **Definition 14** (Projection of functions). *Let $f : \{0,1\}^n \to \mathbb{R}$ and $g\colon \{0,1\}^m \to \mathbb{R}$ be two functions. We say that f is a* projection *of g, denoted $f \leq_{proj} g$, iff*

$$f(x_1, \ldots, x_n) = g(a_1, \ldots, a_m)$$

for some $a_i \in \{0,1\} \cup \{x_1, x_2, \ldots, x_n\}$. That is, f is obtained from g by substitutions of variables of g by variables of f or constants in $\{0,1\}$.

We need the following theorems about computing MAJ_n using a projection of recursive functions.

▶ **Theorem 15** ([20]). *There exists a constant $C > 0$, such that $\mathsf{MAJ}_n\colon \{0,1\}^n \to \{0,1\}$ is a projection of MAJ_3^d where $d = C \log n$.*

▶ **Theorem 16** ([48]). *There exists a constant $C > 0$, such that $\mathsf{MAJ}_n\colon \{0,1\}^n \to \{0,1\}$ is a projection of $(\mathsf{AND}_2 \circ \mathsf{OR}_2)^d$ where $d = C \log n$.*

3 Composition theorem for recursive Majority and alternating AND-OR trees

In this section we give a proof of Theorem 2. We begin with a proof highlight of Lemma 3. The missing proofs are in the full version of the paper [16].

3.1 Proof of Lemma 3

▶ **Lemma 3.** *For any Boolean functions $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$,*

$$\widetilde{\deg}(f \circ \mathsf{MAJ}_t \circ g) = \Omega(\widetilde{\deg}(f)\widetilde{\deg}(g)) \qquad (1.1)$$

for $t \geq C \log n$ for a large enough constant C.

Proof. We will present a proof inspired by the primal-dual view of [40]. Fix any constant $0 < \varepsilon < 1/2$. Let $h := f \circ \mathsf{MAJ}_t \circ g$ be the composed function, and $p_h : \{0,1\}^{ntm} \to \mathbb{R}$ be an ε-approximating polynomial for h.

Further, define $d := \widetilde{\deg}_{\frac{1-\varepsilon}{2}}(g)$. Then, by Lemma 8, there exists a function $\psi : \{0,1\}^m \to \mathbb{R}$ such that

$$\sum_{x \in \{0,1\}^m} |\psi(x)| = 1, \tag{3.1}$$

$$\sum_{x \in \{0,1\}^m} \psi(x) \cdot g(x) > \frac{1-\varepsilon}{2}, \text{ and} \tag{3.2}$$

$$\sum_{x \in \{0,1\}^m} \psi(x) \cdot p(x) = 0 \text{ for every polynomial } p \text{ of degree} < d. \tag{3.3}$$

Let μ be the probability distribution on $\{0,1\}^m$ given by $\mu(x) = |\psi(x)|$ for $x \in \{0,1\}^m$. From (3.3), we have $\sum_{x \in \{0,1\}^m} \psi(x) = 0$. Therefore, the sets $\{x \mid \psi(x) < 0\}$ and $\{x \mid \psi(x) > 0\}$ are weighted equally by μ. Let μ_0 and μ_1 be the probability distributions obtained by conditioning μ on the sets $\{x \mid \psi(x) < 0\}$ and $\{x \mid \psi(x) > 0\}$ respectively. Hence,

$$\mu = \frac{1}{2}\mu_0 + \frac{1}{2}\mu_1, \quad \text{and} \quad \psi = \frac{1}{2}\mu_1 - \frac{1}{2}\mu_0.$$

We note an important property of the distributions μ_0 and μ_1 which shows that the error between $\text{sign}(\psi(x))$ and $g(x)$ is low.

▶ **Lemma 17.** $\mathbb{E}_{x \sim \mu_1}[g(x)] > 1 - \varepsilon$.

▶ **Lemma 18.** $\mathbb{E}_{x \sim \mu_0}[g(x)] < \varepsilon$.

Consider the following linear operator L that maps functions $h : \{0,1\}^{ntm} \to \mathbb{R}$ to functions $Lh : \{0,1\}^n \to \mathbb{R}$,

$$Lh(z) = \mathop{\mathbb{E}}_{\substack{x_{11} \sim \mu_{z_1} \\ x_{12} \sim \mu_{z_1} \\ \vdots \\ x_{1t} \sim \mu_{z_1}}} \mathop{\mathbb{E}}_{\substack{x_{21} \sim \mu_{z_2} \\ x_{22} \sim \mu_{z_2} \\ \vdots \\ x_{2t} \sim \mu_{z_2}}} \cdots \mathop{\mathbb{E}}_{\substack{x_{n1} \sim \mu_{z_n} \\ x_{n2} \sim \mu_{z_n} \\ \vdots \\ x_{nt} \sim \mu_{z_n}}} [h(x_{11}, \ldots, x_{1t}, x_{21}, \ldots, x_{2t}, \ldots, x_{n1}, \ldots, x_{nt})]. \tag{3.4}$$

Recall $h = f \circ \mathsf{MAJ}_t \circ g$ and p_h be ε-approximating polynomial for h. Thus by convexity of L we have $\|L(h - p_h)\|_\infty \leq \varepsilon$. We will now observe some useful properties of the linear operator L.

▶ **Lemma 19.** $\deg(Lp_h) \leq \deg(p_h)/d$, *where* $d = \widetilde{\deg}_{\frac{1-\varepsilon}{2}}(g)$.

We now show that Lp_h is in fact an approximating polynomial for f.

▶ **Lemma 20.** *Fix* $0 < \delta < 1/2$. *Recall* p_h *is an* ε*-approximating polynomial for* $h = f \circ \mathsf{MAJ}_t \circ g$. *Let* $t = \Theta(\log n + \log(1/\delta))$ *where the constant in* $\Theta(\cdot)$ *depends on* ε. *Then,* Lp_h *is a* $(\delta + \varepsilon)$*-approximating polynomial for* f. *That is,*

$$\|f - Lp_h\|_\infty \leq \|f - Lh\|_\infty + \|Lh - Lp_h\|_\infty \leq \delta + \varepsilon.$$

Proof. It suffices to show $\|f - Lh\|_\infty \le \delta$. To this end, consider $Lh(z)$.

$$\begin{aligned} Lh(z) &= \mathop{\mathbb{E}}_{\substack{x_{11}\sim\mu_{z_1}\\ x_{12}\sim\mu_{z_1}\\ \vdots \\ x_{1t}\sim\mu_{z_1}}} \mathop{\mathbb{E}}_{\substack{x_{21}\sim\mu_{z_2}\\ x_{22}\sim\mu_{z_2}\\ \vdots \\ x_{2t}\sim\mu_{z_2}}} \cdots \mathop{\mathbb{E}}_{\substack{x_{n1}\sim\mu_{z_n}\\ x_{n2}\sim\mu_{z_n}\\ \vdots \\ x_{nt}\sim\mu_{z_n}}} [f \circ \mathsf{MAJ}_t \circ g(x_{11}, \ldots, x_{1t}, \ldots, x_{n1}, \ldots, x_{nt})] \\ &= f\left(\mathsf{MAJ}_t\left(\mathop{\mathbb{E}}_{\mu_{z_1}}[g], \ldots, \mathop{\mathbb{E}}_{\mu_{z_1}}[g]\right), \ldots, \mathsf{MAJ}_t\left(\mathop{\mathbb{E}}_{\mu_{z_n}}[g], \ldots, \mathop{\mathbb{E}}_{\mu_{z_n}}[g]\right)\right) \\ &= f(z'_1, z'_2, \ldots, z'_n), \end{aligned}$$

where $\|z - z'\|_\infty \le \delta/n$ because $t = \Theta_\varepsilon(\log n + \log(1/\delta))$ and Lemmas 18 and 17.

Therefore, for any $z \in \{0,1\}^n$, $|f(z) - Lh(z)| = |f(z) - f(z')| \le \delta$, since $\|z - z'\|_\infty \le \delta/n$ and Lemma 12. ◀

Since Lp_h is a $(\delta+\varepsilon)$-approximating polynomial for f, we also have $\deg(Lp_h) \ge \widetilde{\deg}_{\delta+\varepsilon}(f)$. We therefore have the following inequalities

$$\widetilde{\deg}_{\delta+\varepsilon}(f) \le \deg(Lp_h) \le \frac{\deg(p_h)}{\widetilde{\deg}_{\frac{1-\varepsilon}{2}}(g)}.$$

Rewriting we have

$$\widetilde{\deg}_\varepsilon(f \circ \mathsf{MAJ}_t \circ g) = \deg(p_h) \ge \widetilde{\deg}_{\delta+\varepsilon}(f) \cdot \widetilde{\deg}_{\frac{1-\varepsilon}{2}}(g). \tag{3.5}$$

This completes the proof of Lemma 3. ◀

3.2 Proof of Theorem 2

We note an easy to observe fact about approximate degree of projections of functions.

► **Fact 3.6.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ and $g\colon \{0,1\}^m \to \{0,1\}$ be such that $f \le_{proj} g$, i.e., f is a projection of g. Then, for any $\varepsilon \in (0,1/2)$, $\widetilde{\deg}_\varepsilon(f) \le \widetilde{\deg}_\varepsilon(g)$.*

Consider the recursive-majority function MAJ_3^d given by the complete 3-ary tree of height d with internal nodes labeled by MAJ_3 and the leaves are labeled by distinct variables. Fix $d \ge C \log\log n$ for a large enough constant C.

First, observe that MAJ_3^d is *not* a symmetric function. Secondly, it doesn't have *full* approximate degree ([35]). And finally, its approximate degree is *not* equal to $\Theta\left(\sqrt{\mathrm{bs}(\mathsf{MAJ}_3^d)}\right)$ (it follows from the fact that $\mathrm{bs}(\mathsf{MAJ}_3^d)$ is linear with $\widetilde{\deg}(\mathsf{MAJ}_3^d)$. See the full version [16] for a proof of $\widetilde{\deg}(\mathsf{MAJ}_3^d) = 2^d$). Thus, none of the previous works [39, 8, 15] imply that approximate degree composes when one of the (inner or outer) functions is recursive-majority MAJ_3^d.

Proof of Theorem 2. Let MAJ_3^d be the recursive-majority function obtained by the complete 3-ary tree of height d with internal nodes labeled by MAJ_3 and the leaves are labeled by distinct variables. Let $f\colon \{0,1\}^n \to \{0,1\}$ be an arbitrary function and consider the approximate degree of the composed function $f \circ \mathsf{MAJ}_t \circ \mathsf{MAJ}_3^d$ where $t = \Theta(\log n)$.

$$\widetilde{\deg}(f \circ \mathsf{MAJ}_t \circ \mathsf{MAJ}_3^d) \le \widetilde{\deg}(f \circ \mathsf{MAJ}_3^{C\log t} \circ \mathsf{MAJ}_3^d) = \widetilde{\deg}(f \circ \mathsf{MAJ}_3^d \circ \mathsf{MAJ}_3^{C\log t}) \tag{3.7}$$

$$= O(\widetilde{\deg}(f \circ \mathsf{MAJ}_3^d) \cdot \widetilde{\deg}(\mathsf{MAJ}_3^{C\log t})) \tag{3.8}$$

$$= O(\widetilde{\deg}(f \circ \mathsf{MAJ}_3^d) \cdot \mathrm{poly}(t)). \tag{3.9}$$

The first inequality in (3.7) follows from the fact that MAJ_t is a projection of $\mathsf{MAJ}_3^{C\log t}$ (Theorem 15) and Fact 3.6. Then (3.8) follows from Theorem 9.

On the other hand, from Lemma 3, for $t = \Omega(\log n)$ we have

$$\widetilde{\deg}(f \circ \mathsf{MAJ}_t \circ \mathsf{MAJ}_3^d) = \Omega(\widetilde{\deg}(f) \cdot \widetilde{\deg}(\mathsf{MAJ}_3^d)).$$

Combining with (3.9), we obtain the lower bound

$$\widetilde{\deg}(f \circ \mathsf{MAJ}_3^d) = \Omega\left(\frac{\widetilde{\deg}(f) \cdot \widetilde{\deg}(\mathsf{MAJ}_3^d)}{\mathrm{polylog}(n)}\right).$$

A similar argument shows the following inequalities, where in the last two inequalities we use Theorem 16 instead of Theorem 15, for $d = \Omega(\log n)$,

- $\widetilde{\deg}(\mathsf{MAJ}_3^d \circ f) = \widetilde{\Omega}(\widetilde{\deg}(f) \cdot \widetilde{\deg}(\mathsf{MAJ}_3^d))$,
- $\widetilde{\deg}(f \circ (\mathsf{AND}_2 \circ \mathsf{OR}_2)^d) = \widetilde{\Omega}(\widetilde{\deg}(f) \cdot \widetilde{\deg}((\mathsf{AND}_2 \circ \mathsf{OR}_2)^d))$, and
- $\widetilde{\deg}((\mathsf{AND}_2 \circ \mathsf{OR}_2)^d \circ f) = \widetilde{\Omega}(\widetilde{\deg}(f) \cdot \widetilde{\deg}((\mathsf{AND}_2 \circ \mathsf{OR}_2)^d))$. ◀

4 Composition theorem for recursive functions

In this section we prove our main theorem (Theorem 1). It shows that the approximate degree composes when either the inner function or the outer function is a recursive function. More formally,

▶ **Theorem 1.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ and $g\colon \{0,1\}^m \to \{0,1\}$ be two Boolean functions and $d \geq C \log\log n$ for a large enough constant C. Then,*

$$\widetilde{\deg}(f \circ g) = \Omega\left(\frac{\widetilde{\deg}(f)\widetilde{\deg}(g)}{\mathrm{polylog}(n)}\right),$$

if either of the following conditions hold:

1. *$f = h^d$, for any Boolean function h.*
2. *$g = h^d$, for any Boolean function h with constant arity and not equal to* AND *or* OR.

The following cases of Theorem 1 follows from prior works:

1. f or g equals h^d for $h \in \{\mathsf{PARITY}, \neg\mathsf{PARITY}\}$ [39].
2. $f = h^d$ and $h \in \{\mathsf{AND}, \mathsf{OR}\}$ [8].

Therefore, it remains to prove Theorem 1 when $h \notin \{\mathsf{PARITY}, \neg\mathsf{PARITY}, \mathsf{AND}, \mathsf{OR}\}$. A crucial technical insight that makes the proof work is that when $h \notin \{\mathsf{PARITY}, \neg\mathsf{PARITY}, \mathsf{AND}, \mathsf{OR}\}$ then AND_2 and OR_2 are projections of h^3. We can thus simulate MAJ using a small power of h. Thereafter, Lemma 3 is used to conclude Theorem 1. We now work out the details. We first state the main technical lemma we need for Theorem 1 and then complete the proof of the theorem. Finally, we prove the technical lemma in Section 4.1.

▶ **Lemma 21.** *Let $h : \{0,1\}^t \to \{0,1\}$ (where $t \geq 2$) be a Boolean function which depends on all t variables and is not equal to* $\mathsf{PARITY}/\neg\mathsf{PARITY}/\mathsf{OR}/\mathsf{AND}$. *The function* AND_2 *(and similarly* OR_2*) can be obtained by setting all but two variables to constants in h^k for $k \leq 3$.*

We now present the proof of Theorem 1 using Lemma 21.

Proof of Theorem 1. Let $h\colon \{0,1\}^t \to \{0,1\}$ be any Boolean function such that $h \notin \{\mathsf{PARITY}, \neg\mathsf{PARITY}, \mathsf{AND}, \mathsf{OR}\}$. We know from Lemma 3 that $\widetilde{\deg}(f \circ \mathsf{MAJ}_k \circ h^d) = \Omega(\widetilde{\deg}(f)\widetilde{\deg}(h^d))$ where $k = \Theta(\log n)$. Like in the proof of Theorem 2, we will simulate MAJ_k using h^ℓ for sufficiently large ℓ. From Lemma 21, it follows that $(\mathsf{AND}_2 \circ \mathsf{OR}_2)^\ell$ is a projection of $h^{6\ell}$. Therefore, we obtain from Theorem 16 that MAJ_k is a projection of $h^{C\log k}$ for some constant $C > 0$. We thus have the following sequence of inequalities,

$$\begin{aligned}\widetilde{\deg}(f \circ h^d) \geq \widetilde{\deg}(f \circ \mathsf{MAJ}_k \circ h^{(d-C\log k)})\\ &= \Omega(\widetilde{\deg}(f)\widetilde{\deg}(h^{(d-C\log k)}))\\ &= \Omega\left(\frac{\widetilde{\deg}(f)\widetilde{\deg}(h^d)}{t^{C\log k}}\right)\\ &= \Omega\left(\frac{\widetilde{\deg}(f)\widetilde{\deg}(h^d)}{\mathrm{polylog}(n)}\right).\end{aligned}$$

Note that the last equality above uses the fact that t is a constant. When h^d is the outer function then we don't need t to be a constant, while the rest of the argument remains the same to give

$$\widetilde{\deg}(h^d \circ g) = \Omega\left(\frac{\widetilde{\deg}(h^d)\widetilde{\deg}(g)}{\mathrm{polylog}(n)}\right). \blacktriangleleft$$

This completes the proof of the main theorem. We now present a proof of Lemma 21.

4.1 Proof of the main technical lemma (Lemma 21)

We proceed by proving an intermediate result (Lemma 22) before going to the proof of Lemma 21.

Suppose we are allowed to *modify* a Boolean function by two operations: negating some of its variables, and restricting some of the variables to constant values. Lemma 22 proves that almost every Boolean function can be modified to either an AND_2 or an OR_2 function. A restriction of the variables amounts to looking at a smaller hypercube translated to a new point, and negating a variable amounts to rotating the smaller hypercube. In other words, we want to show that there is a *shifted* AND_2 or OR_2 in the Boolean hypercube of h (see Figure 1 for an example).

This shifted $\mathsf{AND}_2/\mathsf{OR}_2$ in the Boolean hypercube of a Boolean function can be concretely defined by the concept of a sensitive block. For a block of variables $S \subseteq [n]$ and an input $x \in \{0,1\}^n$, define $x^{\oplus S} \in \{0,1\}^n$ to be the input which flips exactly the variables in S at the input x. Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, a block S is called sensitive on x iff $f(x) \neq f(x^{\oplus S})$. A block S is called *minimal sensitive* for x at f, if no subset of S is sensitive for x at f.

Notice that a shifted $\mathsf{AND}_2/\mathsf{OR}_2$ is a square with three vertices labelled 0 and one vertex labelled 1 or vice versa. This gives us a minimal sensitive block on the vertex opposite to the unique value. It can be easily verified that the converse is also true. So, we define a function to have a shifted $\mathsf{AND}_2/\mathsf{OR}_2$ iff it has a minimal sensitive block of size 2.

We show below that almost all functions have a minimal sensitive block of size 2.

▶ **Lemma 22.** *Let $h : \{0,1\}^t \to \{0,1\}$ (where $t \geq 2$) be a Boolean function which depends on all t variables and is not equal to* $\mathsf{PARITY}/\neg\mathsf{PARITY}$. *Then, there exists an $x \in \{0,1\}^t$ such that h has a minimal sensitive block of size 2 on x.*

Proof. We will prove the result using induction on the variables. The statement can be easily verified for $t = 2$.

Define g_0 (and g_1) to be the restrictions of h by setting $x_t = 0$ (and $x_t = 1$) respectively. Let e_y be the edge $((y,0),(y,1))$ in the Boolean hypercube, and $S_t := \{e_y : y \in \{0,1\}^{t-1}\}$. Color an edge e_y red if $g_0(y) = g_1(y)$, and blue otherwise.

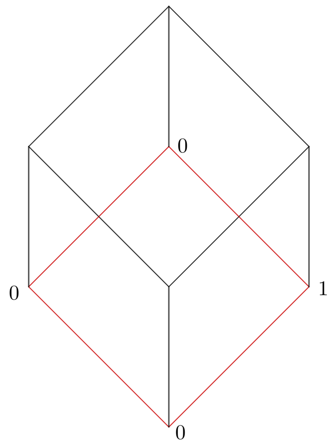

Figure 1 A function on 3 bits with a shifted OR marked with red edges.

Notice that not all the edges in S_t can be red, otherwise h does not depend on x_t. Suppose all the edges in S_t are blue, i.e, $g_1 = \neg g_0$ (in other words, $h = g_0 \oplus x_t$). Since h depends on all variables, then g_0 depends on all variables $x_1, x_2, \cdots, x_{t-1}$. If g_0 is PARITY/¬PARITY, then h is also PARITY/¬PARITY. Implying that g_0 is dependent on all its variables and is not PARITY/¬PARITY. By induction, there exists a minimal sensitive block of size 2 for g_0 (and hence h).

For the rest of the proof, we can assume that there exists both a red and a blue edge in S_t.

Let e_x be red and e_y be blue, this means that $g_0(x) = g_1(x)$ but $g_0(y) \neq g_1(y)$. If x and y were at Hamming distance 1, then vertices $(x, 0), (x, 1), (y, 0)$ and $(y, 1)$ will give us the required minimal sensitive block of size 2.

If x, y are not at Hamming distance 1, look at any path from x to y in the $t-1$ dimensional hypercube, say $z_0 = x, z_1, z_2, \cdots, z_l = y$. The edge e_{z_0} is red and e_{z_l} is blue. Since the color needs to switch at some point, there exist z_i, z_{i+1} at Hamming distance 1 such that e_{z_i} is red and $e_{z_{i+1}}$ is blue. Again, the vertices $(z_i, 0), (z_i, 1), (z_{1+1}, 0)$ and $(z_{i+1}, 1)$ will give us the required minimal sensitive block of size 2. ◀

We are prepared to prove Lemma 21 which shows: given a Boolean function h, AND_2 (and OR_2) can be obtained by restricting some of the variables to constants in a very small power of h. Compared to Lemma 22, we need to remove negation and simulate both AND_2 and OR_2 and not just one of them.

We just show how to obtain AND_2, the case for OR_2 is similar. We handle the case of h being monotone and non-monotone separately.

Monotone h

This case is simpler, and AND_2 can be obtained as a restriction of h itself. Let a minimal 1-input be a $x \in \{0, 1\}^t$ such that setting any 1 bit of x to 0 changes the value of h. If there is a minimal 1-input x of Hamming weight more than 2, we get a AND_2 by choosing any two indices which are 1 in x. The following claim finishes the proof for monotone functions.

▷ Claim 23. Let $h : \{0, 1\}^t \to \{0, 1\}$ be a monotone Boolean function which depends on all variables. If there is no minimal 1-input with Hamming weight more than 2, then h is the OR function.

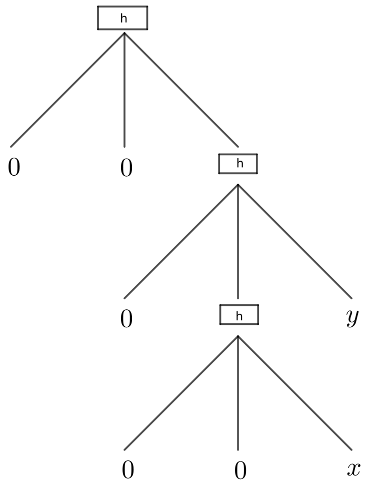

Figure 2 An example for constructing AND_2 using a non-monotone function. Let $\mathsf{h} : \{0,1\}^3 \to \{0,1\}$ be 0 at $x = 001$ and 1 otherwise. Use the shifted OR_2/minimal sensitive block at 001 with indices $\{2,3\}$.

Proof. By abusing the notation, let 0 denote the all 0 input. Since the function is monotone but not constant, we know that $h(0) = 0$. Let $S \subseteq [t]$ capture the indices such that the corresponding Hamming weight 1-input has function value 0,

$$S = \{i : h(0^{\oplus i}) = 0\}.$$

For a $y \in \{0,1\}^t$, if the set of 1-indices are not a subset of S, then $h(y) = 1$ by monotonicity. If the set of 1-indices are a subset of S, then $h(y) = 0$ because there is no minimal 1-input with Hamming weight more than 2.

In other words, h is the OR function on the remaining $[t] \setminus S$ variables. Since h depends on all the t variables, h is the OR function. ◁

Non-monotone h

Since h is a non-monotone function, there exists an input $a \in \{0,1\}^t$ and an index $i \in [t]$ such that $h(a) = 1$, $a_i = 0$ and $h(a^{\oplus i}) = 0$. Restricting the variables according to a (except the i-th bit) gives $h_1(x_i) = \neg x_i$.

From Lemma 22, there exists a $b \in \{0,1\}^t$ such that h has a minimal sensitive block of size 2 on b (shifted $\mathsf{AND}_2/\mathsf{OR}_2$). The main idea of this proof is to use negation and this shifted $\mathsf{AND}_2/\mathsf{OR}_2$ (Figure 2 gives an example).

For the formal proof, without loss of generality assume that the block have indices $1, 2$ (that means $h(b) = h(b^{\oplus\{1\}}) = h(b^{\oplus\{2\}}) \neq h(b^{\oplus\{1,2\}})$). We will finish the proof by considering the two cases $h(b) = 0$ and $h(b) = 1$.

- $h(b) = 0$ (shifted AND_2): Suppose $b_1 = 0$ and $b_2 = 1$ (other cases can be handled similarly). Notice that $\mathsf{AND}_2(x, y) = h(x, \neg y, b_3, \cdots, b_t)$, giving us $\mathsf{AND}_2(x, y) = h(x, h_1(y), b_3, \cdots, b_t)$.
- $h(b) = 1$ (shifted OR_2): Suppose $b_1 = 1$ and $b_2 = 0$ (other cases can be handled similarly). Notice that $\mathsf{OR}_2(x, y) = h(x, \neg y, b_3, \cdots, b_t)$; using De Morgan's law,

$$\mathsf{AND}_2(x, y) = \neg\mathsf{OR}_2(\neg x, \neg y) = \neg h(\neg x, y, b_3, \cdots, b_t) = h_1(h(h_1(x), y, b_3, \cdots, b_t))$$

Since h_1 is also a restriction of h, the proof is complete.

5 Conclusion

Towards the main open problem of approximate degree composition, we have the following immediate question in light of Lemma 3. Can we upper bound $\widetilde{\deg}(f \circ \mathsf{MAJ}_t \circ g)$ in terms of $\widetilde{\deg}(f \circ g)$? Precisely,

▶ **Open question 24.** *Is $\widetilde{\deg}(f \circ \mathsf{MAJ}_t \circ g) = \widetilde{O}(\widetilde{\deg}(f \circ g))$, where $t = \Theta(\log n)$ and n is the arity of the outer function f?*

Observe that an affirmative solution to the above question solves the composition question for approximate degree in positive. Another interesting question is to find other classes of functions for which the analogue of Equation 1.2 holds.

▶ **Open question 25.** *Find non-trivial classes of functions H such that $\widetilde{\deg}(f \circ h \circ g) = \widetilde{\Omega}(\widetilde{\deg}(f) \cdot \widetilde{\deg}(h) \cdot \widetilde{\deg}(g))$ for all $h \in H$?*

It has the following two useful implications. First, this gives composition for functions $h \in H$. In particular, when one of the functions h (inner or outer) belongs to the class H then $\widetilde{\deg}(f \circ h \circ g) = \widetilde{\Omega}(\widetilde{\deg}(f) \cdot \widetilde{\deg}(h) \cdot \widetilde{\deg}(g))$ along with Theorem 9 implies

$$\widetilde{\deg}(h \circ g) = \widetilde{\Omega}(\widetilde{\deg}(h) \cdot \widetilde{\deg}(g)) \quad \text{and} \quad \widetilde{\deg}(f \circ h) = \widetilde{\Omega}(\widetilde{\deg}(f) \cdot \widetilde{\deg}(h)).$$

Second, a function $h \in H$ can be used as "hardness amplifier" functions.

Another very interesting question that may provide us insights to make progress towards the main question of approximate degree composition is to prove that approximate degree composes when the inner function is OR.

▶ **Open question 26.** *Show that $\widetilde{\deg}(f \circ \mathsf{OR}) = \widetilde{\Omega}(\widetilde{\deg}(f).\widetilde{\deg}(\mathsf{OR}))$.*

References

1. S. Aaronson. Impossibility of Succinct Quantum Proofs for Collision-Freeness. *Quantum Inf. Comput.*, 12(1-2):21–28, 2012. doi:10.26421/QIC12.1-2-3.
2. S. Aaronson and Y. Shi. Quantum Lower Bounds for the Collision and the Element Distinctness Problems. *J. ACM*, 51(4):595–605, 2004. doi:10.1145/1008731.1008735.
3. E. Allender. A Note on the Power of Threshold Circuits. In *FOCS*, pages 580–584, 1989. doi:10.1109/SFCS.1989.63538.
4. A. Ambainis. Polynomial Degree and Lower Bounds in Quantum Complexity: Collision and Element Distinctness with Small Range. *Theory Comput.*, 1(1):37–46, 2005. doi:10.4086/toc.2005.v001a003.
5. R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum Lower Bounds by Polynomials. *J. ACM*, 48(4):778–797, 2001. doi:10.1145/502090.502097.

6 R. Beigel. The Polynomial Method in Circuit Complexity. In *Proceedings of the Eigth Annual Structure in Complexity Theory Conference*, pages 82–95, 1993.

7 R. Beigel. Perceptrons, PP, and the Polynomial Hierarchy. *Comput. Complex.*, 4:339–349, 1994. doi:10.1007/BF01263422.

8 S. Ben-David, A. Bouland, A. Garg, and R. Kothari. Classical Lower Bounds from Quantum Upper Bounds. In *FOCS*, pages 339–349, 2018. doi:10.1109/FOCS.2018.00040.

9 S. Ben-David, P. Hatami, and A. Tal. Low-Sensitivity Functions from Unambiguous Certificates. In *ITCS*, volume 67, pages 28:1–28:23, 2017. doi:10.4230/LIPIcs.ITCS.2017.28.

10 A. Bouland, L. Chen, D. Holden, J. Thaler, and P. N. Vasudevan. On the Power of Statistical Zero Knowledge. In *FOCS*, pages 708–719, 2017. doi:10.1109/FOCS.2017.71.

11 H. Buhrman, I. Newman, H. Röhrig, and R. de Wolf. Robust Polynomials and Quantum Algorithms. *Theory Comput. Syst.*, 40(4):379–395, 2007. doi:10.1007/S00224-006-1313-Z.

12 H. Buhrman, N. K. Vereshchagin, and R. de Wolf. On Computation and Communication with Small Bias. In *CCC*, pages 24–32, 2007. doi:10.1109/CCC.2007.18.

13 M. Bun and J. Thaler. Dual Lower Bounds for Approximate Degree and Markov-Bernstein Inequalities. In *ICALP*, volume 7965, pages 303–314, 2013. doi:10.1007/978-3-642-39206-1_26.

14 M. Bun and J. Thaler. Approximate Degree in Classical and Quantum Computing. *Found. Trends Theor. Comput. Sci.*, 15(3-4):229–423, 2022. doi:10.1561/0400000107.

15 S. Chakraborty, C. Kayal, R. Mittal, M. Paraashar, S. Sanyal, and N. Saurabh. On the Composition of Randomized Query Complexity and Approximate Degree. In *APPROX/RANDOM*, volume 275, pages 63:1–63:23, 2023. doi:10.4230/LIPICS.APPROX/RANDOM.2023.63.

16 Sourav Chakraborty, Chandrima Kayal, Rajat Mittal, Manaswi Paraashar, and Nitin Saurabh. Approximate Degree Composition for Recursive Functions, 2024. arXiv:2407.08385.

17 K. Chandrasekaran, J. Thaler, J. R. Ullman, and A. Wan. Faster Private Release of Marginals on Small Databases. In *ITCS*, pages 387–402, 2014. doi:10.1145/2554797.2554833.

18 L. Chen. Adaptivity vs. Postselection, and Hardness Amplification for Polynomial Approximation. In *ISAAC*, volume 64 of *LIPIcs*, pages 26:1–26:12, 2016. doi:10.4230/LIPICS.ISAAC.2016.26.

19 J. Gilmer, M. Saks, and S. Srinivasan. Composition Limits and Separating Examples for Some Boolean Function Complexity Measures. *Combinatorica*, 36(3):265–311, 2016. doi:10.1007/s00493-014-3189-x.

20 O. Goldreich. On (Valiant's) Polynomial-Size Monotone Formula for Majority. In *Computational Complexity and Property Testing - On the Interplay Between Randomness and Computation*, volume 12050 of *Lecture Notes in Computer Science*, pages 17–23. Springer, 2020. doi:10.1007/978-3-030-43662-9_3.

21 M. Göös and T. S. Jayram. A Composition Theorem for Conical Juntas. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 5:1–5:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.CCC.2016.5.

22 T. S. Jayram, R. Kumar, and D. Sivakumar. Two Applications of Information Complexity. In *STOC*, pages 673–682, 2003. doi:10.1145/780542.780640.

23 S. Jukna, A. Razborov, P. Savický, and I. Wegener. On P versus NP ∩ co-NP for Decision trees and Read-Once Branching Programs. *Comput. Complex.*, 8(4):357–370, 1999.

24 J. Kahn, N. Linial, and A. Samorodnitsky. Inclusion-Exclusion: Exact and Approximate. *Comb.*, 16(4):465–477, 1996. doi:10.1007/BF01271266.

25 A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically Learning Halfspaces. *SIAM J. Comput.*, 37(6):1777–1805, 2008. doi:10.1137/060649057.

26 P. Kamath and P. Vasudevan. Approximate Degree of AND-OR trees, 2014. URL: https://www.scottaaronson.com/showcase3/kamath-pritish-vasudevan-prashant.pdf.

27 A. Klivans, R. O'Donnell, and R. Servedio. Learning Intersections and Thresholds of Halfspaces. *J. Comput. Syst. Sci.*, 68(4):808–840, 2004. doi:10.1016/J.JCSS.2003.11.002.

28 A. Klivans and R. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. *J. Comput. Syst. Sci.*, 68(2):303–318, 2004. doi:10.1016/J.JCSS.2003.07.007.

29 A. Klivans and R. Servedio. Toward Attribute Efficient Learning of Decision Lists and Parities. *J. Mach. Learn. Res.*, 7:587–602, 2006. URL: http://jmlr.org/papers/v7/klivans06a.html.

30 T. Lee. A Note on the Sign Degree of Formulas. *CoRR*, abs/0909.4607, 2009. arXiv:0909.4607.

31 N. Leonardos. An Improved Lower Bound for the Randomized Decision Tree Complexity of Recursive Majority. In *ICALP*, volume 7965, pages 696–708, 2013. doi:10.1007/978-3-642-39206-1_59.

32 F. Magniez, A. Nayak, M. Santha, J. Sherman, G. Tardos, and D. Xiao. Improved Bounds for the Randomized Decision tree Complexity of Recursive Majority. *Random Struct. Algorithms*, 48(3):612–638, 2016. doi:10.1002/rsa.20598.

33 N. Nisan and M. Szegedy. On the Degree of Boolean Functions as Real Polynomials. *Comput. Complex.*, 4:301–313, 1994. doi:10.1007/BF01263419.

34 N. Nisan and A. Wigderson. On Rank vs. Communication Complexity. *Combinatorica*, 15(4):557–565, 1995. doi:10.1007/BF01192527.

35 B. Reichardt and R. Špalek. Span-Program-Based Quantum Algorithm for Evaluating Formulas. *Theory Comput.*, 8(1):291–319, 2012. doi:10.4086/TOC.2012.V008A013.

36 M. Saks and A. Wigderson. Probabilistic Boolean Decision Trees and the Complexity of Evaluating Game Trees. In *FOCS*, pages 29–38, 1986. doi:10.1109/SFCS.1986.44.

37 R. Servedio, L.-Y. Tan, and J. Thaler. Attribute-Efficient Learning and Weight-Degree Tradeoffs for Polynomial Threshold Functions. In *COLT*, volume 23, pages 14.1–14.19. JMLR.org, 2012. URL: http://proceedings.mlr.press/v23/servedio12/servedio12.pdf.

38 A. Sherstov. The Pattern Matrix Method. *SIAM J. Comput.*, 40(6):1969–2000, 2011. doi:10.1137/080733644.

39 A. Sherstov. Strong Direct Product Theorems for Quantum Communication and Query Complexity. *SIAM J. Comput.*, 41(5):1122–1165, 2012. doi:10.1137/110842661.

40 A. Sherstov. Approximating the AND-OR Tree. *Theory Comput.*, 9:653–663, 2013. doi:10.4086/toc.2013.v009a020.

41 A. Sherstov. The Intersection of Two Halfspaces has High Threshold Degree. *SIAM Journal on Computing*, 42(6):2329–2374, 2013. doi:10.1137/100785260.

42 A. Sherstov. Making Polynomials Robust to Noise. *Theory Comput.*, 9:593–615, 2013. doi:10.4086/TOC.2013.V009A018.

43 A. A. Sherstov. Approximate Inclusion-Exclusion for Arbitrary Symmetric Functions. *Comput. Complex.*, 18(2):219–247, 2009. doi:10.1007/S00037-009-0274-4.

44 Y. Shi. Approximating Linear Restrictions of Boolean functions, 2002.

45 M. Snir. Lower Bounds on Probabilistic Linear Decision Trees. *Theoretical Computer Science*, 38:69–82, 1985. doi:10.1016/0304-3975(85)90210-5.

46 A. Tal. Properties and Applications of Boolean Function Composition. In *ITCS*, pages 441–454, 2013. doi:10.1145/2422436.2422485.

47 J. Thaler, J. R. Ullman, and S. P. Vadhan. Faster Algorithms for Privately Releasing Marginals. In *ICALP*, volume 7391 of *Lecture Notes in Computer Science*, pages 810–821. Springer, 2012. doi:10.1007/978-3-642-31594-7_68.

48 L. Valiant. Short Monotone Formulae for the Majority Function. *Journal of Algorithms*, 5(3):363–366, 1984. doi:10.1016/0196-6774(84)90016-6.