


Black-Box Hypotheses and Lower Bounds

Brynmor K. Chapman ✉ 🏠
MIT, USA

R. Ryan Williams ✉ 🏠 
MIT, USA

Abstract

What sort of code is so difficult to analyze that every potential analyst can discern essentially no information from the code, other than its input-output behavior? In their seminal work on program obfuscation, Barak, Goldreich, Impagliazzo, Rudich, Sahai, Vadhan, and Yang (CRYPTO 2001) proposed the Black-Box Hypothesis, which roughly states that every property of Boolean functions which has an efficient “analyst” and is “code independent” can also be computed by an analyst that only has black-box access to the code. In their formulation of the Black-Box Hypothesis, the “analysts” are arbitrary randomized polynomial-time algorithms, and the “codes” are general (polynomial-size) circuits. If true, the Black-Box Hypothesis would immediately imply $\text{NP} \not\subseteq \text{BPP}$.

We consider generalized forms of the Black-Box Hypothesis, where the set of “codes” \mathcal{C} and the set of “analysts” \mathcal{A} may correspond to other efficient models of computation, from more restricted models such as AC^0 to more general models such as nondeterministic circuits. We show how lower bounds of the form $\mathcal{C} \not\subseteq \mathcal{A}$ often imply a corresponding Black-Box Hypothesis for those respective codes and analysts. We investigate the possibility of “complete” problems for the Black-Box Hypothesis: problems in \mathcal{C} such that they are not in \mathcal{A} if and only if their corresponding Black-Box Hypothesis is true. Along the way, we prove an equivalence: for nondeterministic circuit classes \mathcal{C} , the “ \mathcal{C} -circuit satisfiability problem” is not in \mathcal{A} if and only if the Black-Box Hypothesis is true for analysts in \mathcal{A} .

2012 ACM Subject Classification Theory of computation → Circuit complexity

Keywords and phrases Black-Box hypothesis, circuit complexity, lower bounds

Digital Object Identifier 10.4230/LIPIcs.MFCS.2021.29

Funding Partially supported by NSF CCF-1741615, NSF CCF-1909429, and a Frank Quick Faculty Innovation Fellowship.

1 Introduction

What kind of code “behaves” like a black box to any code analyst? In particular, what programs are so difficult to analyze that every potential analyst can discern essentially no information from the code, other than its input-output behavior? Such questions are of great importance in cryptography and formal verification: what sort of code is difficult to verify without considerable resources? What kind of code can be obfuscated? What properties of functions can be automatically tested?

A priori, the answers to such questions depend on three factors:

1. The complexity of the code: what instructions are allowed in the code, the computational complexity (e.g. time/space/size/depth complexity) of the algorithm implemented by the code, and so on.
2. The complexity of the analyst: what sorts of operations the analyst can perform, and how much resources it has (time/space/size/depth) to analyze the code.
3. The actual function being computed by the code. If the function itself is trivial or extremely complicated, this could affect how “black box” it can possibly look.



© Brynmor K. Chapman and R. Ryan Williams;
licensed under Creative Commons License CC-BY 4.0

46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021).

Editors: Filippo Bonchi and Simon J. Puglisi; Article No. 29; pp. 29:1–29:22

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper, we consider these three factors carefully, and study obfuscation from a different direction compared to most existing literature on the subject. In particular, we propose generalized forms of the “Black-Box Hypothesis” considered in Barak, Goldreich, Impagliazzo, Rudich, Sahai, Vadhan, and Yang [5] and show how such questions are intimately related to lower bound questions.

A Complexity-Theoretic View

In the pioneering work of Barak et al. [5] on obfuscation, the authors also proposed a compelling conjecture about black-box obfuscation that they called a “Scaled-Down Rice’s Theorem” [5, Conjecture 5.1]; the conjecture has recently been renamed the *Black Box Hypothesis* (BBH) [20, 14]. Informally, the Black-Box Hypothesis posits that, when code is represented as a small Boolean circuit, and a code analyst is represented as an efficient (polynomial-time) randomized algorithm, the only possible analysis tasks are ones that could have been performed using only the input-output behavior of the code (and not the code itself).

While the original Black-Box Hypothesis is still a major open problem, other natural variants of the hypothesis may be tractable for us to resolve, **unconditionally**. We consider variants of the Black-Box Hypothesis in a more general complexity-theoretic setting, where the complexity of the analyst, the complexity of the code being analyzed, and the function to be obfuscated (the “box”) are carefully taken into account. For example, we consider the case where the “analyst function” is taken from a “low” complexity class \mathcal{A} (smaller than P, polynomial time), and the box is also from a “low” complexity class \mathcal{C} .

More formally, we study abstract forms of the Black-Box Hypothesis (sometimes abbreviated as BBH in the following). Let \mathcal{C} be a set of circuits and let \mathcal{A} be a complexity class that permits oracles in its definition. We say that a property $P : \mathcal{C} \rightarrow \{0, 1\}$ of \mathcal{C} is *semantic* if $P(C) = P(C')$ for all pairs of circuits C and C' in \mathcal{C} which compute the same function.

► **Hypothesis 1** (*C-Black-Box Hypothesis for \mathcal{A}*). [Informal Statement, cf. Hypothesis 14] *Let $P : \mathcal{C} \rightarrow \{0, 1\}$ be any semantic property computable by some analyst $A' \in \mathcal{A}$. Then there is a **black-box** analyst $A \in \mathcal{A}$ such that for every s and every circuit $C \in \mathcal{C}$ of size s on n inputs, $A^C(1^n 0^{s-n}) = P(C)$.*

In prior work, the class of analysts \mathcal{A} was always set to be BPP, and the class of circuits \mathcal{C} was generally set to be unrestricted circuits of fan-in two. In that full form, proving the BBH would also prove $\text{NP} \not\subseteq \text{BPP}$, so that is presently out of reach! (The BBH could also end up being false, of course.) By considering a range of natural possible choices for the weak analysts \mathcal{A} and the circuit sets \mathcal{C} , we can try to delineate precisely how weak the analysts from \mathcal{A} need to be, in order for \mathcal{C} -circuits to *provably* behave like black boxes, and to relate the corresponding Black-Box Hypotheses to other core problems within complexity.

1.1 Our Results

We demonstrate several interesting relationships between circuit lower bounds and Black-Box Hypotheses in the generalized setting. First, we prove that certain instances of the Black-Box Hypothesis are true, from known circuit lower bounds. In fact we give a generic connection from lower bounds to Black-Box Hypotheses. We also give some converse results, showing that Black-Box Hypotheses imply certain circuit lower bounds. Finally, in some settings, we can show that certain problems are “complete” for a Black-Box Hypothesis, in the sense that proving the Black-Box Hypothesis is *equivalent* to proving a lower bound against the aforementioned problem.

Black-Box Hypotheses For Restricted Analysts, From Lower Bounds

In Section 4, we explore situations in which known lower bounds imply Black-Box Hypotheses. We first consider Hypothesis 14 where the classes of analysts \mathcal{A} are restricted, and the set of potential “boxes” \mathcal{C} consists of *unrestricted circuits*. We show that one can prove a \mathcal{C} -Black-Box Hypothesis for \mathcal{A} , when the given set of boxes \mathcal{C} is sufficiently powerful and the set of analysts \mathcal{A} is limited. *We find this to be counterintuitive.* It could have been the case that, when the set of boxes \mathcal{C} is powerful, an analyst with access to the code of such a powerful box might be able to learn something interesting about it, and gain more power than if it only had black-box access. However, it turns out that when the boxes are sufficiently powerful, no analyst can learn any *semantic* property.

We find that, under very general conditions, circuit lower bounds against \mathcal{A} (as an algorithmic class) imply the Black-Box Hypothesis for \mathcal{A} (as an analyst class).

► **Theorem 2** (Informal Statement, cf. Theorem 16). *Let \mathcal{A} be a circuit class (of analysts), and let f be a Boolean function computable with (general) circuits of size at most $t(n)$. Suppose $f \notin \mathcal{A}$, and suppose \mathcal{A} is closed under projections from n variables onto $O(t(n) \log t(n))$ variables. Then the (general) Black-Box Hypothesis for \mathcal{A} is true.*

The full formal version of the theorem appears in Section 4 as Theorem 16. Intuitively, we apply a “input-switching” trick which reduces the task of computing f on an input \mathbf{y} to the task of deciding *any* non-trivial semantic property P on a circuit $D_{\mathbf{y}}$.¹ In particular, given an analyst A computing P , we show how to map every Boolean string \mathbf{y} (a potential input for f) into a circuit $D_{\mathbf{y}}$ whose input-output behavior (and in particular, whether $D_{\mathbf{y}}$ satisfies the property P) depends on the value $f(\mathbf{y})$. At a high level, $D_{\mathbf{y}}$ takes an input \mathbf{x} , evaluates $f(\mathbf{y})$, and then (depending on $f(\mathbf{y})$) evaluates and outputs either $C_1(\mathbf{x})$ or $C_2(\mathbf{x})$, where C_1 and C_2 are fixed circuits (independent of \mathbf{y}), exactly one of which satisfies the property P . In essence, we are “switching” the input \mathbf{y} with a circuit $D_{\mathbf{y}}$ which can evaluate f , and for which we can determine P . Then, we can run A on $D_{\mathbf{y}}$ *without ever evaluating f directly*, and use its answer to determine $f(\mathbf{y})$.

The conditions we impose on \mathcal{A} are quite general, so Theorem 16 has several direct corollaries. For example, recall AC^0 is the class of unbounded fan-in circuits of constant depth over AND, OR, and NOT.

► **Corollary 3.** *The BBH for (polynomial-size) AC^0 analysts is true. Moreover, the BBH for $2^{n^{o(1)}}$ -size AC^0 is true.*

In particular, Theorem 16 implies that for every subexponential-size AC^0 circuit family $\{A_n\}$ that is given the code of an arbitrary (general) circuit C as input, if $\{A_n\}$ computes a semantic property (i.e., its output depends only on the function computed by C , not the code of C) then $\{A_n\}$ must compute a trivial property (all-zeroes or all-ones). Similarly:

Let TC_2^0 be the class of unbounded fan-in circuits of depth-two over MAJORITY, AND, OR, and NOT.

► **Corollary 4.** *The BBH for (polynomial-size) TC_2^0 is true. Moreover, the BBH for $2^{n^{1-\varepsilon}}$ -size TC_2^0 analysts is true for every $\varepsilon > 0$.*

¹ At this level of generality, the idea is similar in spirit to one of the proofs of Rice’s Theorem [19] which shows that any non-trivial semantic property of Turing machines is undecidable, by way of a reduction from the Halting Problem. However, Rice’s proof techniques do not translate to finite circuits, so we prove Theorem 16 differently.

A Generalization

Next, we turn to an even more general setting of Black-Box Hypotheses, where both the class \mathcal{A} of “analysts” and the set \mathcal{C} of “boxes” can vary. Here we find that, roughly speaking, if \mathcal{A} and \mathcal{C} jointly satisfy some natural closure properties, and there are functions computable by boxes in \mathcal{C} but not by analysts in \mathcal{A} , then the \mathcal{C} -Black-Box Hypothesis for \mathcal{A} still holds.

► **Theorem 5** (Informal Statement, cf. Theorem 18 and Theorem 20). *Let \mathcal{A} be a circuit (analyst) class, let \mathcal{C} be a set of circuits, and let $f \notin \mathcal{A}$ be a Boolean function. Suppose there is an analyst in \mathcal{A} which, given input \mathbf{y} , generates a circuit $D_{\mathbf{y}} \in \mathcal{C}$ whose input-output behavior depends on the value of $f(\mathbf{y})$. Then the \mathcal{C} -Black-Box Hypothesis for \mathcal{A} is true.*

We prove two formal versions of this theorem in Section 4.1, as Theorem 18 and Theorem 20. These theorems are general enough that \mathcal{C} does not *have* to be a class of circuits *per se*: other non-uniform computational models, such as branching programs or span programs, would also work. The intuition and proof techniques are similar to those used in Theorem 16, but given the extra conditions on \mathcal{A} , we can tailor the input-switching reduction from Theorem 16 to the set \mathcal{C} in order to produce stronger results. For example:

► **Corollary 6.** *For all primes p , the $\text{AC}^0[p]$ -Black-Box Hypothesis for (poly-size) AC^0 holds.*

Theorem 20 implies that for every AC^0 circuit family $\{A_n\}$ that tries to analyze the code of a given $\text{AC}^0[p]$ circuit C , if $\{A_n\}$ computes a semantic property of C , then that property must be trivial. More generally, we can conclude the following.

► **Theorem 7.** *For all depths $d \geq 2$ and all distinct primes $p \neq q$, the $\text{AC}_d^0[p]$ -Black-Box Hypothesis for $2^{s^{o(1)}}$ -size $\text{AC}^0[q]$ analysts is true.*

That is, even if in the above, $\{A_n\}$ can have subexponential size, use MOD_q gates, and fail on input circuits C with depth greater than a fixed constant d , $\{A_n\}$ must *still* compute a trivial property. Similarly:

► **Theorem 8.** *For all depths $d \geq 2$, the AC_d^0 -Black-Box Hypothesis for $2^{s^{o(1)}}$ -size AC_{d-1}^0 analysts is true.*

Equivalences With Lower Bounds?

So far, our results show how lower bound statements of the form $\mathcal{C} \not\subseteq \mathcal{A}$ can sometimes be applied to prove the corresponding \mathcal{C} -Black-Box Hypothesis for \mathcal{A} analysts. A natural next question is, could Black-Box Hypotheses (for various pairs of boxes and analysts) be *equivalent* to proving lower bounds? As a first step, in Section 5 we prove conditional lower bounds against some analyst classes \mathcal{A} , assuming some \mathcal{C} -Black-Box Hypothesis for \mathcal{A} .

► **Theorem 9** (Informal Statement, cf. Theorem 28). *Suppose every analyst in \mathcal{A} has subexponential-size circuits, and let \mathcal{C} be a “reasonable” set of circuits (left undefined here). If the \mathcal{C} -Black-Box Hypothesis for \mathcal{A} is true, then the circuit satisfiability problem for \mathcal{C} -circuits is not in \mathcal{A} .*

Roughly speaking, we observe that if the \mathcal{C} -circuit Evaluation problem (\mathcal{C} -EVAL) is not in \mathcal{A} , then the \mathcal{C} -Black-Box Hypothesis for \mathcal{A} is **true**, and if the \mathcal{C} -circuit Satisfiability problem (\mathcal{C} -SAT) is in \mathcal{A} , then the \mathcal{C} -Black-Box Hypothesis for \mathcal{A} is **false**. However, \mathcal{C} -SAT is generally harder than \mathcal{C} -EVAL.

To better understand how lower bounds connect to Black-Box Hypotheses, we propose a notion of *BBH-completeness* for computational problems. Very roughly, we want a \mathcal{C} -BBH-complete problem Π to have the property that $\Pi \in \mathcal{C}$, and for a general analyst class \mathcal{A} , if $\Pi \notin \mathcal{A}$ then the \mathcal{C} -BBH for \mathcal{A} is true. We show that for *nondeterministic* circuit classes \mathcal{C} , both \mathcal{C} -SAT and \mathcal{C} -EVAL are \mathcal{C} -BBH-complete.

► **Theorem 10** (Informal Statement, cf. Theorem 31). *Suppose every analyst in \mathcal{A} has subexponential-size circuits, and let \mathcal{C} be a nondeterministic circuit class with “natural” closure properties. Then \mathcal{C} -EVAL and \mathcal{C} -SAT are both \mathcal{C} -BBH-complete for \mathcal{A} .*

Theorem 31 shows that lower bounds for the satisfiability problem are equivalent in some sense to proving that nondeterministic circuits behave like black boxes. Impagliazzo, Kabanets, Kolokolova, McKenzie, and Romani [14] considered the question of whether one can show the Black-Box Hypothesis is equivalent to $\text{NP} \not\subseteq \text{P/poly}$, with some partial results. A consequence of Theorem 31 is that $\text{NP} \not\subseteq \text{P/poly}$ is equivalent to the Black-Box Hypothesis when polynomial-size circuits are the analysts and *nondeterministic circuits* are the boxes. In this light, it would be very interesting if one could show the Black-Box Hypothesis is actually equivalent to $\text{NP} \not\subseteq \text{P/poly}$: it would show that two rather different-looking forms of the Black-Box Hypothesis are in fact equivalent.

Finally, we note that the aforementioned work of Impagliazzo et al. on BBH [14, 20] yields another kind of equivalence between a different variant of black-box hypothesis and a circuit lower bound.

► **Theorem 11** (Follows from [14], informal, cf. Theorem 33). *The following are equivalent:*

1. *The Circuit Satisfiability problem, CKT-SAT, is not in P/poly .*
2. *Any symmetric property P that can be decided in P/poly with white-box access to the input circuit can also be decided in P/poly with black-box access to the input circuit.*

We view this interpretation of their result as further promising evidence towards more general connections between black-box hypotheses and circuit lower bounds.

Organization

Section 2 covers significant prior work related to black-box hypotheses. Section 3 carefully discusses how to generalize the Black-Box Hypothesis for various sets of “analysts” and sets of “boxes”. Section 4 proves our main theorems, showing how circuit lower bounds imply Black-Box Hypotheses in a very generic way. Section 5 considers how we might prove equivalences between Black-Box Hypotheses and lower bounds. Section 6 concludes. The appendices include missing proofs, as well as additional related work.

2 Background

In this paper we assume basic familiarity with computational complexity, especially circuit complexity (knowledge of the first 13 chapters of Arora and Barak [4] would suffice). Throughout the paper, we will recall notation and definitions as needed. Sometimes (as is common in complexity) we will blur the distinction between the analyst class \mathcal{A} as a set of circuit *families* (computing some decision problems) and the actual decision problems computed by analysts in \mathcal{A} .

We will study generic versions of the circuit evaluation and circuit satisfiability problems. In the \mathcal{C} -EVAL problem, we are given a circuit C from a set \mathcal{C} and an input x , and wish to know if $C(x) = 1$. In the \mathcal{C} -SAT problem, we are given a circuit from a set \mathcal{C} and wish to know if there is an x such that $C(x) = 1$. The CKT-SAT problem is \mathcal{C} -SAT where \mathcal{C} is the set of arbitrary Boolean circuits (without loss of generality, each gate has fan-in two).

Historically, researchers interested in so-called “black-box hypotheses” were looking for what they called a “scaled-down” Rice’s Theorem. In the following paragraphs, we provide a brief overview of this research.

Rice’s Theorem

We briefly recall the statement and implications of Rice’s Theorem. Let \mathcal{M} be the set of Turing Machines. We say a property $P : \mathcal{M} \rightarrow \{0, 1\}$ of Turing Machines is *semantic* if $P(M)$ depends only on the (possibly partial) function computed by M . That is, for any TMs M_1 and M_2 with the same input-output behavior, $P(M_1) = P(M_2)$. A property P is *non-trivial* if there are $M_1, M_2 \in \mathcal{M}$ such that $P(M_1) \neq P(M_2)$. In his 1951 doctoral thesis, Henry Rice proved the following sweeping result:

► **Theorem 12** ([19]). *Every non-trivial semantic property of Turing Machines is undecidable.*

Rice’s powerful theorem states that any interesting property that we might want to test of a given program is undecidable, assuming the property being tested depends only on the *function computed by the program*. That is, any property that could in principle be tested using only black-box access to the program, is undecidable given a *description* of the program. Rice’s theorem generalizes (and can be proved from) the undecidability of the TM-SAT problem of determining whether a given TM accepts any string at all.

The Black Box Hypothesis

In their pioneering obfuscation work, Barak et al. [5] consider the question: *can Rice’s Theorem be scaled down in a way that would be useful to complexity theory?* Specifically, let us assume we are not interested in *all* Turing Machines, but rather in the set of efficient algorithms; for example, those represented by Boolean circuits. One can still define properties that are non-trivial and semantic when restricted to the set of Boolean circuits. In this setting, all such properties P are decidable, because the language of a circuit is simply its 2^n -bit truth table, which can be computed in finite time. However, one might want to know something about the *computational complexity* of such properties. In this setting, the circuit satisfiability problem CKT-SAT is an analogue of TM-SAT. Although CKT-SAT is decidable, it is NP-hard, so one might hope to be able to replace undecidability in Rice’s Theorem with NP-hardness.

In earlier work, Borchert and Stephan [8] note that using circuits instead of Turing Machines and NP-hardness instead of undecidability is not enough to prove an analogue of Rice’s Theorem. For every string x , the property $\{M \in \mathcal{M} : M(x) = 1\}$ is undecidable by Rice’s Theorem, but the circuit analogue is decidable in polynomial time: it is simply the circuit evaluation problem! Borchert and Stephan’s response to this issue is to look at function properties depending on more complex measures, such as the *number* of SAT assignments of a given circuit (in other words, the property is a symmetric Boolean function in the truth table of the circuit). They show that any non-trivial “counting” property of circuits is UP-hard; the UP-hardness were improved in [13].

Barak et al. [5] gave a different response to the above issue. They observe the property $\{C : C(x) = 1\}$ for circuits C is still “trivial” in some sense: it can be efficiently determined given only *black-box* oracle access to the input circuit. This observation led Barak et al. to formulate the following conjecture. For two circuits C and C' on n -bit inputs, we write $C \equiv C'$ when C and C' compute the same n -bit function.

► **Conjecture 13** (Black Box Hypothesis [5]). *Suppose $L \subseteq \{0, 1\}^*$ satisfies the property that for all C and C' such that $C \equiv C'$, we have $C \in L \iff C' \in L$. If $L \in \text{BPP}$, then there is a probabilistic polynomial time algorithm S that decides L given only oracle access to C and $0^n 1^{|C|-n}$ as input, i.e.,*

$$C \in L \implies \Pr \left[S^C \left(0^n 1^{|C|-n} \right) = 1 \right] > \frac{2}{3}$$

$$C \notin L \implies \Pr \left[S^C \left(0^n 1^{|C|-n} \right) = 1 \right] < \frac{1}{3}.$$

That is, the BBH claims that every “white box” semantic property of circuits that is decidable in randomized poly-time can also be decided in randomized poly-time with “black box” access to the circuit. If the conjecture were true, then a strong form of $\text{P} \neq \text{NP}$ would follow: $\text{P} = \text{NP}$ implies that circuit satisfiability is solvable in polynomial-time when we have “white-box” access to the input circuit, but the SAT problem requires $\Omega(2^n)$ time to solve with only black-box oracle access to the input circuit.

Impagliazzo et al. [14] proved interesting results towards understanding BBH. They show a partial converse of the observation from the previous paragraph: if the BBH is false for certain kinds of properties, then the circuit satisfiability problem has sub-exponential size circuits. Since we know that BBH implies $\text{P} \neq \text{NP}$, this suggests that it may be difficult to resolve BBH regardless of its truth or falsity. Romani’s master thesis [20] gives an excellent overview of the BBH and this work.

An additional section on “Other Related Work” appears in Appendix A.

3 Generalized Black-Box Hypotheses

We study the Black-Box Hypothesis (Conjecture 13) in a more general setting. Specifically, instead of considering $L \in \text{BPP}$ and a randomized uniform algorithm S from Conjecture 13, we study the family of hypotheses that arise when L and S come from various (possibly non-uniform) circuit classes, which may be weaker or stronger than probabilistic poly-time.

Let us set up some notation. For a circuit C , we let $\langle C \rangle$ denote the *binary description* of C . Note that if C has size s , then $\langle C \rangle$ is a binary string of length $O(s \log s)$, which we call the *description length* of C .

Let \mathcal{C} be a set of circuits. A *property* of circuits in \mathcal{C} is a function $P : \mathcal{C} \rightarrow \{0, 1\}$. A property S is *semantic* iff for any two circuits $C_1, C_2 \in \mathcal{C}$ computing the same function (that is, $\forall \mathbf{x}, C_1(\mathbf{x}) = C_2(\mathbf{x})$), $P(C_1) = P(C_2)$. Recall a circuit family is an infinite sequence of circuits, one for each possible input length; circuit families compute functions of the form $f : \{0, 1\}^* \rightarrow \{0, 1\}$ in the natural way. We say that a circuit family $\{A_s\}$ *computes* P if for every circuit $C \in \mathcal{C}$ with description length s , $A_s(\langle C \rangle) = P(C)$.

We define a **circuit class** \mathcal{A} to be a set of circuit families; our analyst classes \mathcal{A} will have this form. By convention, an *oracle circuit* C may have oracle gates of arbitrary fan-in, but we will think of C as taking an oracle O with a fixed number of inputs. If C contains oracle gates with a different number of inputs than the given oracle O , then we define such oracle gates to output the constant 0 (regardless of O).

We formulate a generalization of the Black Box Hypothesis, which we call the \mathcal{C} -Black Box Hypothesis for \mathcal{A} (\mathcal{C} -BBH for \mathcal{A}), in the following way.

► **Hypothesis 14** (Generalized Black Box Hypothesis: \mathcal{C} -BBH for \mathcal{A}). *Let P be a semantic property of circuits in \mathcal{C} . Let $\{A'_s\} \in \mathcal{A}$ be a circuit family that computes P . Then there exists a circuit family $\{A_s\} \in \mathcal{A}^{\mathcal{C}}$ such that $A_s(1^n 0^{s-n}) = 1$ iff $P(C) = 1$.*

That is, the \mathcal{C} -BBH for \mathcal{A} hypothesizes that every semantic property of \mathcal{C} -circuits that can be decided by \mathcal{A} -analysts with “white box” access to the \mathcal{C} -circuit, can also be decided by \mathcal{A} -analysts with only black-box access to the circuit. When \mathcal{C} is the set of all Boolean circuits, we refer to the \mathcal{C} -BBH for \mathcal{A} simply as the “BBH for \mathcal{A} ”. Note that if we replace \mathcal{A} in the above with BPP, we recover Conjecture 13.

3.1 Encoding Circuits

Unfortunately, if we allow the class of analysts \mathcal{A} to be an arbitrary circuit class, we can encounter some strange (and counterintuitive) consequences. For instance, suppose \mathcal{A} is AC^0 , the circuit families over AND, OR, and NOT with constant-depth, polynomial size, and unbounded fan-in. We can construct an oracle circuit family $\{A_s\}$ such that $A_s^C(1^n 0^{s-n}) = \text{PARITY}(n)$, the parity of the number of inputs of C (A_s^C ignores C , and just computes the parity of strings of the form $1^* 0^*$). Depending on how the description $\langle C \rangle$ is represented, this behavior may not be computable by *any* white-box AC^0 circuit family $\{A'_s\}$, since PARITY is not in AC^0 [1, 9]! We would like to avoid this sort of behavior, because as in Conjecture 13, the oracle circuit family A is supposed to capture some notion of *triviality*. In order for the “BBH for \mathcal{A} ” to be meaningful, it should be that the white-box circuit family A' is at least as powerful as the black-box family A . To this end, we shall require the binary descriptions of circuits to contain all the information given freely to the oracle family. Specifically, we assume that the description of a circuit C with n input wires is prefixed by $1^n 0$, and that the first n wires in $\langle C \rangle$ are the input wires.

4 Circuit Lower Bounds Imply Black-Box Hypotheses

What can we prove about the BBH for general pairs of circuit sets and analysts \mathcal{C} , \mathcal{A} ? First, we can show there are interesting pairs for which the \mathcal{C} -BBH for \mathcal{A} is true in a strong way: every semantic property is in fact trivial. The following theorem shows that, whenever lower bounds hold against a circuit class \mathcal{A} satisfying some simple conditions, the (general) BBH for \mathcal{A} is true. First, we recall a definition.

► **Definition 15.** A projection from n variables onto m variables is a function $\pi : \{0, 1\}^n \rightarrow \{0, 1\}^m$ such that for every j , there exists i such that the j^{th} coordinate of $\pi(\mathbf{x})$ depends only on the i^{th} coordinate of \mathbf{x} .

Observe that a projection is a kind of very weak reduction which can be computed not only very efficiently but also very *locally*. By requiring closure under such a weak class of reductions, we aim to keep \mathcal{A} as general as possible.

► **Theorem 16.** Let \mathcal{A} be a circuit class, $f : \{0, 1\}^* \rightarrow \{0, 1\}$ be a decision problem, and $s : \mathbb{N} \rightarrow \mathbb{N}$ be a monotone function with the properties:

1. f is computable by a size- $s(n)$ circuit family, but f is not computable by any family in \mathcal{A} .
2. Either $\{\text{OR}_n \circ \text{AND}_2\} \subseteq \mathcal{C} \in \mathcal{A}$ for some family \mathcal{C} , or $\{\text{OR}_n \circ \text{AND}_2\} \subseteq \mathcal{C} \in \mathcal{A}$ for some family \mathcal{C} . That is, either \mathcal{A} contains a family that either computes the read-once n -clause 2-DNFs on $2n$ variables, or it contains a family that computes the n -clause 2-CNFs on $2n$ variables.
3. \mathcal{A} is closed under composition with projections from n variables onto $O(s(n) \log s(n))$ variables.

Then for every property P over the set of all circuits, if P is semantic and computable in \mathcal{A} , then for all n , P restricted to circuits on n -bit inputs is also trivial. In particular, the (general) BBH for \mathcal{A} is true.

Proof. Let \mathcal{A} and f satisfy the above properties, and let $\{F_n\}$ be a size- $s(n)$ circuit family computing f . Let P be a semantic property computable in \mathcal{A} .

First, we will prove that P is trivial. The idea is that, if P is not trivial, we can use a circuit family for P to construct a circuit in \mathcal{A} for computing f , a contradiction to the assumed lower bound on f (assumption 1).

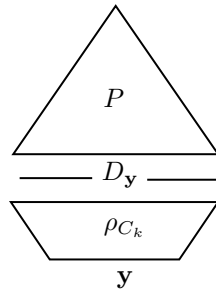
Let $k \in \mathbb{N}$. P is semantic, so assume WLOG that for every k -input circuit K_0 computing the constant 0 function, $P(K_0) = 0$. Assume for sake of contradiction that there is a k -input C_k such that $P(C_k) = 1$. Let $n \in \mathbb{N}$ be our desired input length; we want to build a circuit computing f on n -bit inputs. For an n -bit vector \mathbf{y} , define the following circuit $D_{\mathbf{y}}$ with k input wires \mathbf{x} , with \mathbf{y} hard-coded as n constant wires:

$$D_{\mathbf{y}}(\mathbf{x}) := C_k(\mathbf{x}) \wedge F_n(\mathbf{y}).$$

The circuit $D_{\mathbf{y}}$ computes some Boolean function on k input bits. For a fixed C_k , define the function ρ_{C_k} that maps the n -bit input \mathbf{y} to the description $\langle D_{\mathbf{y}} \rangle$ of $D_{\mathbf{y}}$ as defined above. Observe that for all \mathbf{x} and \mathbf{y} , $D_{\mathbf{y}}(\mathbf{x}) = C_k(\mathbf{x})$ if $f(\mathbf{y}) = 1$, and otherwise $D_{\mathbf{y}}(\mathbf{x}) = 0$. Because P is semantic, $P(D_{\mathbf{y}}) = P(C_k) = 1$ if $f(\mathbf{y}) = 1$, and $P(D_{\mathbf{y}}) = 0$ otherwise. In other words, we have $P(D_{\mathbf{y}}) = f(\mathbf{y})$ for all \mathbf{y} .

Note the size of $D_{\mathbf{y}}$ is $t(k) + \|C_k\| + 1$, where $\|C_k\|$ denotes the size of C_k (which is independent of n), so $D_{\mathbf{y}}$ has description length $O(s(n) \log s(n))$. For a fixed C_k , $\rho_{C_k}(\mathbf{y}) = \langle D_{\mathbf{y}} \rangle$ depends only the n -bit vector \mathbf{y} . In particular, within the description $\langle D_{\mathbf{y}} \rangle$, the descriptions $\langle C_k \rangle$ and $\langle F_n \rangle$ are both independent of \mathbf{y} , so the only bits in $\langle D_{\mathbf{y}} \rangle$ that vary with \mathbf{y} are those describing the hard-coded constant \mathbf{y} itself. Hence each bit in $\langle D_{\mathbf{y}} \rangle$ depends on at most one bit of \mathbf{y} . That is, ρ_{C_k} is a projection from n variables onto $O(s(n) \log s(n))$ variables.

Since \mathcal{A} is closed under such projections (assumption 3), and P is computable in \mathcal{A} by assumption, the circuit



is also computable in \mathcal{A} . However, $P(D_{\mathbf{y}}) = f(\mathbf{y})$, which is not computable in \mathcal{A} , a contradiction. It follows that for all C_k on k inputs, $P(C_k) = 0$, so P (on circuits containing k inputs) is trivial.

We now turn to proving that there exists an oracle circuit family $\{A_s\}$ in \mathcal{A} such that for any circuit C of size s on n inputs, $A_s^C(1^n 0^{s-n}) = P(C)$. In fact we prove the stronger claim that there exists a circuit family $\{A_s\}$ in \mathcal{A} (with no oracle gates) such that for any circuit C of size s on n inputs, $A_s(1^n 0^{s-n}) = P(C)$. To this end, let $X = \{n \in \mathbb{N} : \exists C \text{ on } n \text{ inputs with } P(C) = 1\}$. First, suppose that \mathcal{A} contains a family that can compute $\{\text{OR}_n \circ \text{AND}_2\}$. For $s \in \mathbb{N}$, let A_s be the circuit of the form

$$\bigvee_{i \in X \cap [s]} (x_i \wedge \neg x_{i+1}).$$

29:10 Black-Box Hypotheses and Lower Bounds

By assumptions 2 and 3 (closure under projections from n to $2n$ variables), such circuits are in \mathcal{A} . If instead \mathcal{A} contains $\{\text{AND}_n \circ \text{OR}_2\}$, we let A_s be the circuit of the form

$$\bigwedge_{i \in [s] \setminus X} (\neg x_i \vee x_{i+1}).$$

Now $A_s(1^n 0^{s-n}) = 1$ iff $n \in X$ (using no oracle gates). Since P is trivial, for all circuits C on n inputs, $A_s(1^n 0^{s-n}) = 1$ iff $P(C) = 1$, as desired. ◀

The above proof can be thought of as an “input-switching” trick. We start with the fact that P is non-trivial on some k -bit input circuits. We use the description of a k -input circuit witnessing non-triviality, along with the description of a circuit computing f on n -bit inputs, to construct the description of a larger circuit $D_{\mathbf{y}}$ with n “free variables” \mathbf{y} . By feeding n -bit \mathbf{y} into that description, and feeding that description into P , we obtain the description of an \mathcal{A} -circuit computing f .

Theorem 16 has many immediate corollaries. For example:

► **Reminder of Corollary 3.** *The BBH for (polynomial-size) AC^0 is true. Moreover, the BBH for $2^{n^{o(1)}}$ -size AC^0 is true.*

Proof. Take \mathcal{A} to be AC^0 and f to be the PARITY function in Theorem 16, using the fact that PARITY does not have subexponential-size AC^0 circuits [12]. ◀

► **Reminder of Corollary 4.** *The BBH for (polynomial-size) TC_2^0 is true. Moreover, the BBH for $2^{n^{1-\varepsilon}}$ -size TC_2^0 is true for every $\varepsilon > 0$.*

Proof. Take \mathcal{A} to be TC_2^0 and f to be the INNERPRODUCT function (mod 2) in Theorem 16, using the fact that INNERPRODUCT requires $2^{\Omega(n)}$ -size TC_2^0 circuits [3]. ◀

4.1 Generalization

The proof of Theorem 16 critically relies on the fact that the circuit $D_{\mathbf{y}}$ can be arbitrarily large and complex in comparison to its input. If we restrict \mathcal{C} to contain only “simple” circuits and allow A'_s to behave arbitrarily on circuits not in \mathcal{C} , then we would need to be more careful to ensure that $D_{\mathbf{y}}$ is still in \mathcal{C} . By extending the input-switching trick from Theorem 16, we can restrict the circuit set \mathcal{C} in some interesting ways and still prove the corresponding Black-Box Hypotheses.

► **Definition 17.** *Let \mathcal{C} be a set of circuits, and let f and g be Boolean functions. We say that a function $I : \mathcal{C} \times \{0, 1\}^* \rightarrow \{0, 1\}^*$ is an input-switching function for \mathcal{C} and f iff for some bit b , for every circuit $C \in \mathcal{C}$ and every Boolean string \mathbf{y} , $I(C, \mathbf{y})$ is the description $\langle D_{\mathbf{y}} \rangle$ of a circuit $D_{\mathbf{y}}$ with the same number of inputs as C such that $D_{\mathbf{y}}(\mathbf{x}) = C(\mathbf{x})$ when $f(\mathbf{y}) = b$ and $D_{\mathbf{y}}(\mathbf{x}) = g(\mathbf{x})$ otherwise.*

► **Theorem 18.** *Let \mathcal{A} be a circuit class, $f : \{0, 1\}^* \rightarrow \{0, 1\}$ be a decision problem, and \mathcal{C} be a set of circuits with the properties:*

1. \mathcal{A} computes neither f nor $\neg f$.
2. \mathcal{A} is closed under composition with an input-switching function I for \mathcal{C} and f , in the sense that for every function g computable by a circuit family in \mathcal{A} and for every $C \in \mathcal{C}$, the function $\mathbf{y} \mapsto g(I(C, \mathbf{y}))$ is also computable by a circuit family in \mathcal{A} .

Then for every property P over \mathcal{C} , if P is semantic and computable in \mathcal{A} , then for all input lengths n , P restricted to circuits on n -bit inputs is also trivial. Furthermore, if \mathcal{A} also contains $\{\text{OR}_n \circ \text{AND}_2\}$ (or $\{\text{AND}_n \circ \text{OR}_2\}$), then the \mathcal{C} -BBH for \mathcal{A} is true.

The proof is in Appendix B.

The preconditions for Theorem 18 are somewhat too restrictive to be applied easily in many cases, so we strengthen it further. To this end, we first define a relation \sim_n on sets of circuits.

► **Definition 19.** For sets \mathcal{C}_1 and \mathcal{C}_2 of circuits, say that $\mathcal{C}_1 \sim_n \mathcal{C}_2$ iff there exist n -input circuits $C_1 \in \mathcal{C}_1$ and $C_2 \in \mathcal{C}_2$ such that $C_1 \equiv C_2$ (that is, C_1 and C_2 compute precisely the same Boolean function).

The relation \sim_n enables us to more easily reason about semantic properties across several sets of differently structured circuits.

► **Theorem 20.** Let \mathcal{A} be a circuit class, $f : \{0, 1\}^* \rightarrow \{0, 1\}$ be a decision problem, $\mathcal{C} = \bigcup_{i \in \mathbb{N}} \mathcal{C}_i$ be a set of circuits, and $I : \mathcal{C} \times \{0, 1\}^* \rightarrow \{0, 1\}^*$ a function with the properties:

1. \mathcal{A} computes neither f nor $\neg f$.
2. \mathcal{A} is closed under composition with I .
3. For all i , the restriction of I to $\mathcal{C}_i \times \{0, 1\}^*$ is an input-switching function for \mathcal{C}_i and f .
4. For every input size $n \in \mathbb{N}$, the transitive closure of \sim_n on $\{\mathcal{C}_i\}$ is the universal relation on $\{\mathcal{C}_i\}$.

Then for every property P over \mathcal{C} , if P is semantic and computable in \mathcal{A} , then for all input lengths n , P restricted to circuits on n -bit inputs is also trivial. Furthermore, if \mathcal{A} also contains $\{OR_n \circ AND_2\}$ (or $\{AND_n \circ OR_2\}$), then the \mathcal{C} -BBH for \mathcal{A} is true.

Proof. Let P be a property over \mathcal{C} . Applying Theorem 18 to \mathcal{A} , f , and to each \mathcal{C}_i , for all n and all i , the restrictions of P to circuits in each \mathcal{C}_i with n -bit inputs is trivial. Since P is semantic, if $i \sim_n j$, then the restriction of P to circuits in $\mathcal{C}_i \cup \mathcal{C}_j$ with n -bit inputs is also trivial. Finally since the transitive closure of \sim_n is universal, by induction we have that for every n , the restriction of P to circuits in \mathcal{C} with n -bit inputs is trivial. ◀

4.2 Examples

We now define some input-switching functions. First, let f be any function computable by a circuit family $\{F_n\}$, and let $D_{\mathcal{C}, \mathbf{y}}$ be the circuit defined as follows, where \mathbf{x} are the input wires and \mathbf{y} are hard-coded as n constant wires:

$$D_{\mathcal{C}, \mathbf{y}}(\mathbf{x}) := C(\mathbf{x}) \wedge F_n(\mathbf{y}).$$

If F_n has size $s(n)$, then the map $\mathbf{y} \mapsto \langle D_{\mathcal{C}, \mathbf{y}} \rangle$ (where $\langle D_{\mathcal{C}, \mathbf{y}} \rangle$ is the description of $D_{\mathcal{C}, \mathbf{y}}$) is both an input-switching function and a projection from n variables onto the $O(s(n) \log s(n))$ variables describing $D_{\mathcal{C}, \mathbf{y}}$, so we recover Theorem 16.

Recall that $AC_d^0[p]$ denotes circuit families of depth d with unbounded fan-in AND, OR, and MOD_p gates.

► **Reminder of Corollary 6.** For all primes p , the $AC^0[p]$ -Black-Box Hypothesis for (polynomial-size) AC^0 is true. Moreover, the $AC^0[p]$ -Black-Box Hypothesis for $2^{s^{o(1)}}$ -size AC^0 is true.

Proof. Follows from Theorem 18. We make use of the fact that the MOD_p function is computable in linear size $AC^0[p]$ but requires exponential size in AC^0 , and that in AC^0 we can mask a given $AC^0[p]$ circuit with a given MOD_p function.

29:12 Black-Box Hypotheses and Lower Bounds

Let \mathcal{A} be $2^{s^{o(1)}}$ -size AC^0 , $f = \text{MOD}_p$, and $\mathcal{C} = \text{AC}^0[p]$. We now define a circuit $D_{C,\mathbf{y}}$ with the same number of inputs as C as

$$D_{C,\mathbf{y}}(\mathbf{x}) := C(\mathbf{x}) \wedge \text{MOD}_p(\mathbf{y}).$$

Then for all \mathbf{x} and \mathbf{y} , $D_{C,\mathbf{y}}(\mathbf{x}) = C(\mathbf{x})$ if $\text{MOD}_p(\mathbf{y}) = 1$, and $D_{C,\mathbf{y}}(\mathbf{x}) = 0$ otherwise. Now the map $(C, \mathbf{y}) \mapsto \langle D_{C,\mathbf{y}} \rangle$ is an input-switching function for \mathcal{C} and MOD_p . Furthermore, we can think of the map $\mathbf{y} \mapsto \langle D_{C,\mathbf{y}} \rangle$ as a projection from n variables \mathbf{y} onto $\Theta(n \log n)$ variables describing $D_{C,\mathbf{y}}$, so \mathcal{A} is closed under composition with I . Now from Theorem 18, every semantic property P over \mathcal{C} computable in \mathcal{A} is trivial, so the \mathcal{C} -BBH for \mathcal{A} is true. \blacktriangleleft

If we invoke Theorem 20 instead of Theorem 18, we can get an even stronger result.

► **Theorem 21.** *For all depths $d \geq 2$ and distinct primes $p \neq q$, the $\text{AC}_d^0[p]$ -BBH for $2^{s^{o(1)}}$ -size $\text{AC}^0[q]$ is true.*

The proof is in Appendix C. The proof of Theorem 21 relies on the fact that small $\text{AC}^0[q]$ circuits cannot evaluate some function that can be evaluated with small $\text{AC}^0[p]$ circuits (namely a single MOD_p gate). We can prove a similar result using the depth- d Sipser function, which is easy for AC^0 circuits of depth d but hard for depth $d - 1$ [22, 12].

► **Definition 22.** *The Sipser function $f^{d,n} : \{0, 1\}^{\sqrt{\frac{n}{\log n}}} \times \{0, 1\}^{n^{d-2}} \times \{0, 1\}^{\sqrt{\frac{1}{2}dn \log n}} \rightarrow \{0, 1\}$ is defined as follows:*

$$\begin{aligned} \text{If } d \text{ is odd, then } f^{d,n}(\mathbf{x}) &= \bigwedge_{i_1=1}^{\sqrt{\frac{n}{\log n}}} \bigvee_{i_2=1}^n \bigwedge_{i_3=1}^n \cdots \bigwedge_{i_d=1}^{\sqrt{\frac{1}{2}dn \log n}} x_{i_1, \dots, i_d}. \\ \text{If } d \text{ is even, then } f^{d,n}(\mathbf{x}) &= \bigwedge_{i_1=1}^{\sqrt{\frac{n}{\log n}}} \bigvee_{i_2=1}^n \bigwedge_{i_3=1}^n \cdots \bigvee_{i_d=1}^{\sqrt{\frac{1}{2}dn \log n}} x_{i_1, \dots, i_d}. \end{aligned}$$

► **Theorem 23.** *For all depths $d \geq 2$, the AC_d^0 -BBH for $2^{s^{o(1)}}$ -size AC_{d-1}^0 is true.*

The proof is in Appendix D.

5 Some Black-Box Hypotheses Imply Lower Bounds

In Section 4, we showed that many circuit lower bounds of the form $\mathcal{C}' \not\subseteq \mathcal{A}$ can be used to prove a corresponding \mathcal{C} -Black-Box Hypothesis for \mathcal{A} (for a set of boxes \mathcal{C} that suitably captures the complexity class \mathcal{C}'). Now we consider the converse question: can Black-Box Hypotheses also be used to prove circuit lower bounds? For certain sets \mathcal{C} of boxes and classes \mathcal{A} of analysts, it turns out that the \mathcal{C} -Black-Box Hypothesis for \mathcal{A} does in fact imply lower bounds against \mathcal{A} .

For a function $s : \mathbb{N} \rightarrow \mathbb{N}$, let $\text{CIRCUIT}(s(n))$ denote the set of (general) Boolean circuits on n inputs of size at most $s(n)$, for every n . (Note this is different from $\text{SIZE}(s(n))$, which is the class of *languages* computed by circuit families of size at most $s(n)$.) As a starting point, the following simple proposition was essentially noted by Barak et al. [5].

► **Proposition 24.** *If $\text{NP} \subset \text{P}/\text{poly}$, then for every polynomial p , the $\text{CIRCUIT}(p(n))$ -BBH for P/poly is false.*

Proof. Take P to be the CKT-SAT property (that is, $P(C) = 0$ iff the circuit C encodes the all-zeroes function). By assumption, $P \in \text{P/poly}$, but even with randomness, $\Omega(2^n)$ oracle queries are needed to determine whether a size- $p(n)$ circuit on n inputs is the all-zeroes function. For every polynomial q , the polynomial $q \circ p$ is $o(2^n)$, so there is no size- $q(s)$ circuit family making $\Omega(2^n)$ oracle queries on size- $p(n)$ circuits. ◀

In fact, Proposition 24 can be strengthened by replacing CKT-SAT with the property $P(C) = 1$ iff C has a satisfying assignment that sets the first k inputs to 0 (for some appropriately large k).

► **Proposition 25.** *If $\text{NP} \subset \text{SIZE}(2^{n^{o(1)}})$, then for every polynomial p , the $\text{CIRCUIT}(p(n))$ -BBH for P/poly is false.*

Propositions 24 and 25 are arguably not particularly useful, since very few researchers believe the hypotheses of these propositions. However, they still do illustrate an interesting observation, and we may be able to generalize them in a useful manner. Let \mathcal{C} -SAT be the satisfiability problem for circuits from the set \mathcal{C} . One might hope to prove the following generalization of Proposition 24, for every circuit set \mathcal{C} and every analyst class \mathcal{A} :

► **Hypothesis 26** (The Satisfiability Black-Box Hypothesis). *If $\mathcal{C}\text{-SAT} \in \mathcal{A}$, then the \mathcal{C} -BBH for \mathcal{A} is false.*

In this fully generic form, there are some simple counterexamples to Hypothesis 26. For instance, if \mathcal{A} contains *all* Boolean functions, then (for every set \mathcal{C}) $\mathcal{C}\text{-SAT} \in \mathcal{A}$. However, the \mathcal{C} -BBH for \mathcal{A} is *true*, because \mathcal{A} can decide *any* semantic property with only black-box access to the circuit being analyzed. Hence we require additional restrictions on \mathcal{C} and \mathcal{A} to make the hypothesis interesting. In particular, we would like \mathcal{A} to contain only functions of subexponential circuit complexity, and for a sufficiently simple function f , we would like \mathcal{C} circuits to be able to compute f efficiently.

Recall that a Boolean function $f : \{0,1\}^* \rightarrow \{0,1\}$ is a *point function* if there is an $\mathbf{a} \in \{0,1\}^*$ such that for all \mathbf{x} , $f(\mathbf{x}) = 1 \iff \mathbf{x} = \mathbf{a}$. The following notion of “reasonability” for circuit sets will be useful in multiple contexts.

► **Definition 27** (Reasonability). *A set \mathcal{C} of circuits is reasonable if there is a polynomial p such that for all point functions f , there is a circuit family $\{C_n\} \subset \mathcal{C}$ of size at most $p(n)$ computing f .*

We can show that if \mathcal{C} is reasonable and \mathcal{A} has subexponential-size circuits, then Hypothesis 26 is true. The following can be viewed as a kind of converse of Theorem 16.

► **Theorem 28.** *If \mathcal{C} is reasonable, $\mathcal{A} \subseteq \text{SIZE}(2^{n^{o(1)}})$, and $\mathcal{C}\text{-SAT} \in \mathcal{A}$, then the \mathcal{C} -BBH for \mathcal{A} is false.*

Proof. Assume \mathcal{C} is reasonable, \mathcal{A} has subexponential-size circuits, and $\mathcal{C}\text{-SAT} \in \mathcal{A}$. As in Proposition 24, we take P to be the satisfiability property. By assumption, $P \in \mathcal{A}$. Even with randomness, $\Omega(2^n)$ oracle queries are required to determine whether a circuit of size $p(n)$ on n inputs computes the constant 0 function. However, an \mathcal{A} circuit can make at most $2^{n^{o(1)}}$ queries to its oracle when given an input of size $p(n)$. Per the reasonableness of \mathcal{C} , there are both satisfiable and unsatisfiable \mathcal{C} -circuits of size $p(n)$, so \mathcal{A} , with only black-box access to a \mathcal{C} -circuit, cannot compute P . ◀

The preconditions for Theorem 28 are very general; most complexity classes of interest only deal with functions of subexponential complexity and can compute point functions efficiently. However, this weak condition is sufficient to remove the simple counterexamples.

5.1 A Notion of BBH-Completeness

For very general circuit sets \mathcal{C} and classes \mathcal{A} of analysts, we have shown (roughly) in Section 4 that

$$\mathcal{C} \not\subseteq \mathcal{A} \implies \mathcal{C}\text{-BBH for } \mathcal{A},$$

and in the previous paragraphs that for “reasonable” \mathcal{A} and \mathcal{C} ,

$$\mathcal{C}\text{-BBH for } \mathcal{A} \implies \mathcal{C}\text{-SAT} \not\subseteq \mathcal{A}.$$

For many pairs of classes \mathcal{C} and \mathcal{A} , we have

$$\mathcal{C}\text{-EVAL} \not\subseteq \mathcal{A} \iff \mathcal{C} \not\subseteq \mathcal{A}.$$

So the results of Section 4 imply, at least for many natural pairs \mathcal{C}, \mathcal{A} , that \mathcal{C} -EVAL lower bounds imply BBHs. However, \mathcal{C} -SAT is generally a harder problem than \mathcal{C} -EVAL, so there remains a gap between the lower bounds that provably imply a Black-Box Hypothesis, and those lower bounds provably implied by a Black-Box Hypothesis.

A natural question is then, which of these implications can be strengthened? **Is there a single problem on \mathcal{C} circuits, such that proving a lower bound for it is equivalent to proving a \mathcal{C} -Black-Box Hypothesis?** In particular, is proving either $\mathcal{C}\text{-EVAL} \not\subseteq \mathcal{A}$ or $\mathcal{C}\text{-SAT} \not\subseteq \mathcal{A}$ *equivalent* to proving the \mathcal{C} -BBH for \mathcal{A} ? Similar to other completeness notions in complexity theory, we propose a concept of BBH-completeness to study equivalences between circuit lower bounds and Black-Box Hypotheses.²

► **Definition 29** (BBH-completeness). *Let \mathcal{C} be a set of circuits and \mathcal{A} a complexity class. A Boolean function f is complete for the \mathcal{C} -BBH for \mathcal{A} (or \mathcal{C} -BBH-complete for \mathcal{A}) iff*

$$\mathcal{C}\text{-BBH for } \mathcal{A} \iff f \notin \mathcal{A}.$$

When \mathcal{A} is either implicitly understood or general, we say that f is \mathcal{C} -BBH-complete.

Are there natural pairs $(\mathcal{C}, \mathcal{A})$ for which either \mathcal{C} -EVAL or \mathcal{C} -SAT is \mathcal{C} -BBH-complete for \mathcal{A} ?

The Case of Nondeterministic Boxes. For the case of sets \mathcal{C} of nondeterministic circuits, the answer is **yes**. To state our theorem, we require one new concept. Recall that a nondeterministic circuit C has a sequence of “normal” inputs \mathbf{x} as well as a sequence of “auxiliary” nondeterministic inputs \mathbf{y} , and we say that C accepts \mathbf{x} if there is a setting of \mathbf{y} such that $C(\mathbf{x}, \mathbf{y}) = 1$.

► **Definition 30.** *For a given circuit C , a nondeterminization of C is a circuit C' in which normal inputs to C have been converted into auxiliary nondeterministic inputs. A set \mathcal{C} of circuits is closed under nondeterminization if $C \in \mathcal{C}$ implies that every nondeterminization of C is also in \mathcal{C} .*

² It must be said that both authors are not entirely comfortable with the following definition of BBH-completeness. Ideally, the following would be a consequence of f being BBH-complete, and the actual definition would involve a notion of reducibility. However, in order to give a completeness concept that fits all possible classes \mathcal{A} and \mathcal{C} at a high level of generality, it does not seem possible to use reductions: a sound reducibility notion would inevitably have to depend on \mathcal{A} (in particular, its allowed “sizes” and its closure properties) directly.

► **Theorem 31.** *Let \mathcal{C} be a reasonable set of circuits closed under nondeterminization. Assume \mathcal{A} has circuits of size $2^{n^{o(1)}}$ and that \mathcal{A} is closed under composition with an input-switching function for \mathcal{C} and \mathcal{C} -EVAL. Then \mathcal{C} -EVAL and \mathcal{C} -SAT are \mathcal{C} -BBH-complete for \mathcal{A} .*

Proof. We wish to prove that the following are equivalent:

1. \mathcal{C} -BBH for \mathcal{A}
2. \mathcal{C} -SAT $\notin \mathcal{A}$
3. \mathcal{C} -EVAL $\notin \mathcal{A}$

(1) \implies (2) Follows from Theorem 28.

(2) \implies (3) We reduce \mathcal{C} -SAT to \mathcal{C} -EVAL by observing that changing the inputs of a nondeterministic circuit into auxiliary nondeterministic inputs preserves satisfiability. Hence, given a nondeterministic circuit C , we can convert *all* of its input bits into additional nondeterministic auxiliary inputs to obtain a circuit C' , and then determine whether C' is still satisfiable. However, C' has no remaining free inputs, so determining satisfiability of C' is simply the problem of *evaluating* C' (with no inputs).

(3) \implies (1) Follows from Theorem 18. ◀

Interpreting Impagliazzo et al. as an Equivalence. Recently, Impagliazzo et al. [14] proved that if the BBH is false for certain kinds of function properties, then the circuit satisfiability problem has sub-exponential size circuits. In particular, they show that CKT-SAT has $2^{n^{o(1)}}$ -size circuits if a property P is highly sensitive on a function f that has sub-exponential size circuits.

Impagliazzo et al. indicate that in some sense CKT-SAT is BBH-complete, at least for large analyst classes \mathcal{A} . Specifically, if we consider only *symmetric* semantic properties, i.e., properties that depend only on the number of ones in the truth table of the input circuit, we can define the following conjecture:

► **Hypothesis 32 (Symmetric-BBH).** *Let P be a semantic and symmetric property of circuits. Let $\{A'_s\}$ be a polynomial size circuit family. Assume that for every circuit C of size s on n inputs, $A'_s(C) = 1$ iff $P(C) = 1$. Then there exists a polynomial size oracle circuit family $\{A_s\}$ such that $A_s^C(1^n 0^{s-n}) = 1$ iff $P(C) = 1$.*

Now [14] implies:

► **Theorem 33 (Follows from [14]).** *The following are equivalent:*

1. CKT-SAT is not in P/poly.
2. The Symmetric-BBH is true.

Proof. The forward direction is Corollary 4.3 in [14]. For the converse direction, observe that CKT-SAT is a symmetric property that requires exponentially many black-box oracle queries (and in particular, cannot be solved in P/poly with only black-box access to the input circuit). Hence if the Symmetric-BBH is true, then CKT-SAT also cannot be solved in P/poly with white-box access to the input circuit, i.e., CKT-SAT \notin P/poly. ◀

6 Conclusion

In this paper, we introduced *generalized* Black-Box Hypotheses, which parameterize both the type of “box” being analyzed, and the type of “analyst” examining such boxes. We showed that generalized Black-Box Hypotheses can follow generically from circuit lower bounds, and we showed how lower bounds for the circuit satisfiability problem are essentially equivalent to Black-Box Hypotheses where the “boxes” correspond to nondeterministic circuits. We conclude with some additional interesting directions to consider.

What Other Lower Bounds Are Implied by Black-Box Hypotheses? In Section 5 we noted a simple example of a lower bound implying a BBH: the \mathcal{C} -BBH for \mathcal{A} implies \mathcal{C} -SAT $\notin \mathcal{A}$. However, this lower bound is rather weak-looking: \mathcal{C} -SAT is NP-complete for many very simple \mathcal{C} . Are there circuit-analysis problems which are likely *not* to be NP-complete, which would still be implied by a Black-Box Hypothesis? We find this to be a very interesting question, and we currently do not have good candidates for such a problem.

Randomized Lower Bounds and Their Black-Box Hypotheses. We have shown that (deterministic) worst-case lower bounds can lead to results about analyzing circuits as boxes. What results can be derived from *average-case* or *randomized* lower bounds? We have obtained some preliminary results in this direction. For instance, if our analyst class \mathcal{A} consists of *randomized* algorithms rather than deterministic ones, we can still prove connections between lower bounds against \mathcal{A} and BBHs for \mathcal{A} , along the lines of Section 4. There are likely other connections like this to be found within the vast landscape of complexity theory.

Black-Box Hypotheses From More Lower Bounds? While we have shown that various Black-Box Hypotheses do follow from certain lower bounds in a generic way, some lower bounds don't seem to imply a Black-Box Hypothesis. For example, a circuit-size hierarchy is well-known: for nice functions s , there are functions computable with size- $s(n)$ circuits that do not have circuits of size less than $s(n) - 5n$ (cf. [15]). This suggests the possibility that, for analysts \mathcal{A} implemented by circuits of size less than $s(n) - 5n$, and boxes \mathcal{C} which are circuits of size at least $s(n)$, the \mathcal{C} -Black-Box Hypothesis for \mathcal{A} is true. However, our current methods are unable to prove such a sharp result. Are there other intermediate lower bounds (weaker than against e.g. \mathcal{C} -EVAL) that would still imply Black-Box Hypotheses?

References

- 1 Miklos Ajtai. Σ_1^1 -formulae on finite structures. *Annals of Pure and Applied Logic*, 24:1–48, 1983.
- 2 Shaul Almagor, Brynmor Chapman, Mehran Hosseini, Joël Ouaknine, and James Worrell. Effective divergence analysis for linear recurrence sequences. In *Proceedings of the 29th International Conference on Concurrency Theory (CONCUR 2018), LIPIcs 118*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
- 3 Kazuyuki Amano. On the size of depth-two threshold circuits for the inner product mod 2 function. In Alberto Leporati, Carlos Martín-Vide, Dana Shapira, and Claudio Zandron, editors, *Language and Automata Theory and Applications*, pages 235–247, Cham, 2020. Springer International Publishing.
- 4 Sanjeev Arora and Boaz Barak. *Computational complexity: A modern approach*. Cambridge University Press, Cambridge, 2009. doi:10.1017/CB09780511804090.
- 5 Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. In *Annual international cryptology conference*, pages 1–18. Springer, 2001.
- 6 Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. *Journal of the ACM (JACM)*, 65(6):39, 2018.
- 7 Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. *Algorithmica*, 79(4):1233–1285, 2017.
- 8 Bernd Borchert and Frank Stephan. Looking for an analogue of Rice's theorem in circuit complexity theory. In *Kurt Gödel Colloquium on Computational Logic and Proof Theory*, pages 114–127. Springer, 1997.

- 9 Merrick Furst, James Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984. See also FOCS’81.
- 10 Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM Journal on Computing*, 45(3):882–929, 2016.
- 11 Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Hiding secrets in software: A cryptographic approach to program obfuscation. *Communications of the ACM*, 59(5):113–120, 2016.
- 12 Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the 18th Annual ACM symposium on Theory of computing*, pages 6–20, 1986.
- 13 Lane A Hemaspaandra and Mayur Thakur. Lower bounds and the hardness of counting properties. *Theoretical computer science*, 326(1-3):1–28, 2004.
- 14 Russell Impagliazzo, Valentine Kabanets, Antonina Kolokolova, Pierre McKenzie, and Shadab Romani. Does looking inside a circuit help? In *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.
- 15 Stasys Jukna. *Boolean Function Complexity – Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-24508-4.
- 16 Joël Ouaknine and James Worrell. Decision problems for linear recurrence sequences. In *Proceedings of the 6th International Workshop on Reachability Problems (RP 2012)*, LNCS 7550. Springer, 2012.
- 17 Joël Ouaknine and James Worrell. Ultimate positivity is decidable for simple linear recurrence sequences. In *Proceedings of 41st International Colloquium on Automata, Languages, and Programming (ICALP 2014)*, LNCS 8573. Springer, 2014.
- 18 Alexander Razborov. Lower bounds on the size of bounded-depth networks over the complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.
- 19 Henry Gordon Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 74(2):358–366, 1953.
- 20 Shadab Romani. *Succinct representations of Boolean functions and the Circuit-SAT problem*. PhD thesis, Memorial University of Newfoundland, 2016.
- 21 Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 475–484. ACM, 2014.
- 22 Michael Sipser. Borel sets and circuit complexity. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 61–69, 1983. doi:10.1145/800061.808733.
- 23 Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *STOC*, pages 77–82, 1987.

A Other Related Work

Obfuscation in Cryptography

In recent years, the theory of program obfuscation has exploded into a huge subject area within cryptography, starting with an influential paper of Barak et al. [5] which crystallized several key definitions and proved key impossibility results for obfuscation. Two major concepts they proposed are *virtual black-box obfuscation* (VBB for short) and *indistinguishability obfuscation* (iO for short), which we now describe briefly.

A VBB obfuscator \mathcal{O} would take any efficient program/circuit C of size s , and output the code of an “obfuscated” $\mathcal{O}(C)$ such that, for every probabilistic polynomial time (PPT) adversary A , there is another PPT adversary A' , such that the probability A outputs 1 on

the input $\mathcal{O}(C)$ is very close to the probability that A' outputs 1 on $(1^n, 1^s)$ when *given* C as an oracle. That is, whatever computation A is doing on the code of $\mathcal{O}(C)$, A' can simulate that knowing only the size of C , its number of inputs, and with input-output access to C . Barak et al. showed that there are tasks for which VBB obfuscation is *impossible* assuming one-way functions exist. The notion of iO asks for a weaker guarantee: for all PPT A , and all pairs of size- s circuits C_1, C_2 such that $C_1 \equiv C_2$, the probability A outputs 1 on C_1 is very close to the probability A outputs 1 on C_2 . In contrast to VBB, iO is possible under plausible hardness conjectures (e.g., [11, 10, 6]), and it turns out to be very powerful, capable of implementing deniable encryption, public-key encryption from one-way functions, multiparty key exchange, and more (e.g., [21, 7]).

All of the above work on building obfuscation requires hardness assumptions that are unproved (and are typically much stronger than $P \neq NP$), and study how we might efficiently transform arbitrary code into obfuscated code, relative to some class of adversarial analysts.

We briefly note the connection between VBB and the BBH. One can think of a VBB obfuscator as an efficient mapping from general circuits to “obfuscated class of circuits”, a *restricted subclass* of circuits, such that the BBH holds when the analyzable code C must come from this restricted subclass. Namely, the VBB property says that, for any efficient analyst that takes circuits from this class as input, there is an efficient black-box analyst that can carry out essentially the same analyses. That is, when VBB is possible, there is a “promise” class of circuits (the image of the obfuscator) for which a black-box hypothesis is true. Accordingly, Barak et al. [5] showed that a “promise” version of the BBH is false, assuming one-way functions exist.

Automated Formal Verification

Additionally, settings in which the Black-Box Hypothesis is false are of great interest in automated formal verification. One central question is the following: what properties of a program’s input-output behavior can be more efficiently tested by analyzing the program’s code, than by treating it as a black box and simply running it on selected inputs? Many properties of interest depend on the program’s behavior on all possible inputs, which may be infeasible (or even impossible) to determine exhaustively. One may instead want to analyze the code of the program in order to determine whether or not it satisfies the given property. This may still be impossible, as many properties of interest are Turing-complete when considered over the space of all possible programs. However, by restricting the class of programs being tested, some such verification problems can become feasible, cf. [16, 17, 2]. In fact, in any setting where the class of programs being analyzed is restricted such that the black box hypothesis is *false*, there *must* exist properties that can be tested by analyzing the program but not by treating it as a black box.

B Proof of Theorem 18

► **Reminder of Theorem 18.** Let \mathcal{A} be a circuit class, $f : \{0, 1\}^* \rightarrow \{0, 1\}$ be a decision problem, and \mathcal{C} be a set of circuits with the properties:

1. \mathcal{A} contains neither f nor $\neg f$.
2. \mathcal{A} is closed under composition with an input-switching function I for \mathcal{C} and f , in the sense that for every function g computable by a circuit family in \mathcal{A} and for every $C \in \mathcal{C}$, the function $\mathbf{y} \mapsto g(I(C, \mathbf{y}))$ is also computable by a circuit family in \mathcal{A} .

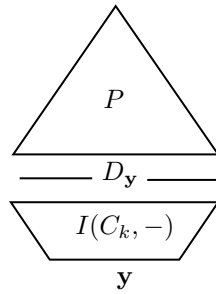
Then for every property P over \mathcal{C} , if P is semantic and computable in \mathcal{A} , then for all input lengths n , P restricted to circuits on n -bit inputs is also trivial. Furthermore, if \mathcal{A} also contains $\{\text{OR}_n \circ \text{AND}_2\}$ (or $\{\text{AND}_n \circ \text{OR}_2\}$), then the \mathcal{C} -BBH for \mathcal{A} is true.

Proof. Let \mathcal{A} and f satisfy the above properties, and let I be the input-switching function for \mathcal{C} and f . Let P be a semantic property computable in \mathcal{A} .

First, we will prove that P is trivial. The idea is that, if P is not trivial, we can use a circuit family for P to construct a circuit in \mathcal{A} for computing f , a contradiction to the assumed lower bound on f .

Let $k \in \mathbb{N}$. Assume WLOG that for every circuit G computing g on k inputs, $P(G) = 0$. Assume for sake of contradiction that there is a k -input C_k such that $P(C_k) = 1$. For an n -bit input \mathbf{y} , consider the circuit $D_{\mathbf{y}} = I(C_k, y)$. Note that $D_{\mathbf{y}}$ computes some Boolean function on k input bits. From the definition of I , $D_{\mathbf{y}}(\mathbf{x}) = C_k(\mathbf{x})$ if $f(\mathbf{y}) = b$, and otherwise $D_{\mathbf{y}}(\mathbf{x}) = G(\mathbf{x})$. Because P is semantic, $P(D_{\mathbf{y}}) = P(C_k) = 1$ if $f(\mathbf{y}) = b$, and $P(D_{\mathbf{y}}) = P(G) = 0$ otherwise. In other words, we have $P(D_{\mathbf{y}}) = b \otimes f(\mathbf{y})$ for all \mathbf{y} .

Since \mathcal{A} is closed under composition with $I(C_K, -)$, and P is computable in \mathcal{A} by assumption, the circuit



is also computable in \mathcal{A} . However, $P(D_{\mathbf{y}}) = f(\mathbf{y})$ or $\neg f(\mathbf{y})$, which are not computable in \mathcal{A} , a contradiction. It follows that for all C_k on k inputs, $P(C_k) = 0$, so P (on circuits containing k inputs) is trivial.

As in Theorem 16, there exists a circuit family $\{A_s\}$ in \mathcal{A} (with no oracle gates) such that for any circuit C of size s on n inputs, $A_s(1^n 0^{s-n}) = P(C)$. ◀

C Proof of Theorem 21

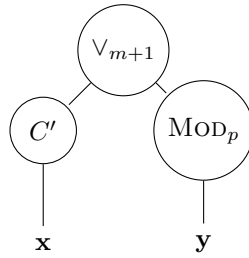
▶ **Reminder of Theorem 21.** For all depths $d \geq 2$ and distinct primes $p \neq q$, the $\text{AC}_d^0[p]$ -BBH for $2^{s^{o(1)}}$ -size $\text{AC}^0[q]$ is true.

Proof. Follows from Theorem 20. We make use of the fact that the MOD_p function is computable in linear size $\text{AC}_d^0[p]$ but requires exponential size in $\text{AC}^0[q]$ [18, 23], and that in AC^0 we can mask a given $\text{AC}_d^0[p]$ circuit with a given MOD_p function, without increasing its depth.

Let $d \geq 2$, \mathcal{A} be $2^{s^{o(1)}}$ -size $\text{AC}^0[q]$, $f = \text{MOD}_p$, $\mathcal{C} = \text{AC}_d^0[p]$, $\mathcal{C}_1 = \text{OR} \circ \text{AC}_{d-1}^0[p]$, $\mathcal{C}_2 = \text{AND} \circ \text{AC}_{d-1}^0[p]$, and $\mathcal{C}_3 = \text{MOD}_p \circ \text{AC}_{d-1}^0[p]$. (Note that $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3$.) We now define a function $I : \mathcal{C} \times \{0, 1\}^* \rightarrow \{0, 1\}^*$, so that $I(C, \mathbf{y}) = \langle D_{\mathbf{y}} \rangle$, where $D_{\mathbf{y}}$ has the same number of inputs as C . We condition on whether the input circuit C comes from \mathcal{C}_1 , \mathcal{C}_2 , or \mathcal{C}_3 .

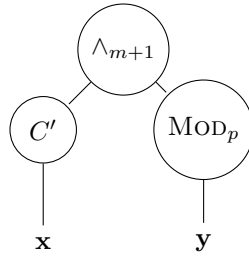
Case 1. If $C \in \mathcal{C}_1$, then it has the form $\vee_m \circ C'$, where C' is a depth- $(d-1)$ circuit with n inputs and m outputs, and \vee_m is an OR of fan-in m . For a k -bit vector \mathbf{y} , we construct $D_{\mathbf{y}}$ as follows:

29:20 Black-Box Hypotheses and Lower Bounds



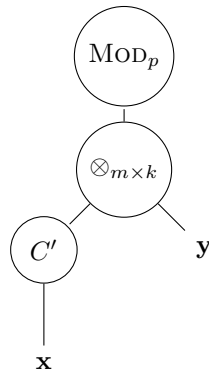
Then for all \mathbf{x} and \mathbf{y} , $D_{\mathbf{y}}(\mathbf{x}) = C(\mathbf{x})$ if $\text{MOD}_p(\mathbf{y}) = 0$, and $D_{\mathbf{y}}(\mathbf{x}) = 1$ otherwise. Hence the restriction of I to $\mathcal{C}_1 \times \{0, 1\}^*$ is an input-switching function for \mathcal{C}_1 and f .

Case 2. If $C \in \mathcal{C}_2$, we construct $D_{\mathbf{y}}$ similarly to case (1). Assuming $C = \wedge_m \circ C'$ for the same sort of C' , we can construct $D_{\mathbf{y}}$ as follows:



Then for all \mathbf{x} and \mathbf{y} , $D_{\mathbf{y}}(\mathbf{x}) = C(\mathbf{x})$ if $\text{MOD}_p(\mathbf{y}) = 1$, and $D_{\mathbf{y}}(\mathbf{x}) = 0$ otherwise. Hence the restriction of I to $\mathcal{C}_2 \times \{0, 1\}^*$ is an input-switching function for \mathcal{C}_2 and f .

Case 3. If $C \in \mathcal{C}_3$, then C is a MOD_p gate of fan-in m , composed with some depth- $(d - 1)$ circuit C' having n inputs and m outputs. We define a $\otimes_{m \times k}$ gate to take $m + k$ inputs $x_1, \dots, x_m, y_1, \dots, y_k$, and output $x_i \cdot y_j$ for all i, j , and define a circuit $D'_{\mathbf{y}}(\mathbf{x})$ as follows:



Note that for C of depth d , $D'_{\mathbf{y}}$ has depth $d + 1$. However, when treating \mathbf{y} as a constant, each $C'(\mathbf{x})_i \wedge y_j$ simplifies to a single wire (either $C'(\mathbf{x})_i$ if $y_j = 1$, or the constant 0 if $y_j = 0$). Performing these simplifications and removing the layer of AND gates, we get a circuit $D_{\mathbf{y}}$ of depth d . (Note that each bit in $\langle D_{\mathbf{y}} \rangle$ still only depends on at most one bit of \mathbf{y} .) Now for all \mathbf{x} and \mathbf{y} , $D_{\mathbf{y}}(\mathbf{x}) = \text{MOD}_p(C'(\mathbf{x}) \otimes \mathbf{y}) = \text{MOD}_p(C'(\mathbf{x})) \times \text{MOD}_p(\mathbf{y}) = C(\mathbf{x}) \times \text{MOD}_p(\mathbf{y})$. That is, $D_{\mathbf{y}}(\mathbf{x}) = C(\mathbf{x})$ if $\text{MOD}_p(\mathbf{y}) = 1$, and $D_{\mathbf{y}}(\mathbf{x}) = 0$ otherwise. Hence the restriction of I to $\mathcal{C}_3 \times \{0, 1\}^*$ is an input-switching function for \mathcal{C}_3 and f .

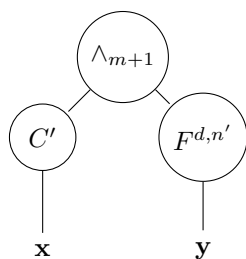
Next, we observe that in every case, each bit in $\langle D_{\mathbf{y}} \rangle$ depends on only one bit of \mathbf{y} , so \mathcal{A} is closed under composition with I (a projection). Finally, there are circuits C_1, C_2, C_3 , which have an OR, AND, and MOD_p output gate (respectively), yet $C_1 \equiv C_2 \equiv C_3$ (e.g. they can ignore their input and output the constant 0). Hence \sim_k as defined in Theorem 20 is the universal relation. Now from Theorem 20, every semantic property P over \mathcal{C} computable in \mathcal{A} is trivial, so the \mathcal{C} -BBH for \mathcal{A} is true. ◀

D Proof of Theorem 23

► **Reminder of Theorem 23.** For all depths $d \geq 2$, the AC_d^0 -BBH for $2^{s^{o(1)}}$ -size AC_{d-1}^0 is true.

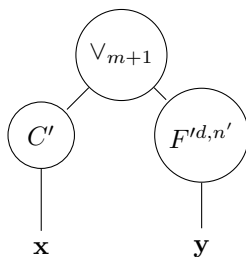
Proof. We proceed as in Theorem 21. Let $d \geq 2$, f be the depth- d Sipser function, \mathcal{A} be $2^{s^{o(1)}}$ -size AC_{d-1}^0 , $\mathcal{C} = \text{AC}_d^0$, $\mathcal{C}_1 = \text{AND} \circ \text{AC}_{d-1}^0$, and $\mathcal{C}_2 = \text{OR} \circ \text{AC}_{d-1}^0$. (Note that $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$.) We now define a function $I : \mathcal{C} \times \{0, 1\}^* \rightarrow \{0, 1\}^*$, so that $I(C, \mathbf{y}) = \langle D_{\mathbf{y}} \rangle$, where $D_{\mathbf{y}}$ has the same number of inputs as C . We condition on whether the input circuit C comes from \mathcal{C}_1 or \mathcal{C}_2 .

Case 1. If $C \in \mathcal{C}_1$, then it has the form $\wedge_m \circ C'$, where C' is a depth- $(d-1)$ circuit with n inputs and m outputs, and \wedge_m is an AND of fan-in m . Let $k \in \mathbb{N}$, and take $n' = (2k/d)^{1/(d-1)}$, so that $f^{d,n'}$ has k inputs. For a k -bit vector \mathbf{y} , we construct $D'_{\mathbf{y}}$ as follows:



Here, $F^{d,n'}$ denotes the obvious depth- d circuit computing the Sipser function $f^{d,n'}$. Now by collapsing the output AND gate of $F^{d,n'}$ into the \wedge_{m+1} , we obtain a depth- d circuit $D_{\mathbf{y}}$ on n inputs such that $D_{\mathbf{y}}(\mathbf{x}) = C(\mathbf{x})$ if $f^{d,n'}(\mathbf{y}) = 1$, and $D_{\mathbf{y}}(\mathbf{x}) = 0$ otherwise. Hence the restriction of I to $\mathcal{C}_1 \times \{0, 1\}^*$ is an input-switching function for \mathcal{C}_1 and f .

Case 2. If $C \in \mathcal{C}_2$, then it has the form $\vee_m \circ C'$. In this case, we construct the circuit $D'_{\mathbf{y}}$ as follows:



Here $F'^{d,n'}$ denotes the circuit obtained by replacing all AND gates in $F^{d,n'}$ with OR gates and vice-versa, and negating all of the input wires. By collapsing the output OR gate of $F'^{d,n'}$ into the \vee_{m+1} , we obtain a depth- d circuit $D_{\mathbf{y}}$ on n inputs such that $D_{\mathbf{y}}(\mathbf{x}) = C(\mathbf{x})$ if $f^{d,n'}(\mathbf{y}) = 1$, and $D_{\mathbf{y}}(\mathbf{x}) = 1$ otherwise. Hence the restriction of I to $\mathcal{C}_2 \times \{0, 1\}^*$ is an input-switching function for \mathcal{C}_2 and f .

29:22 Black-Box Hypotheses and Lower Bounds

As before, we observe that each bit in $\langle D_{\mathbf{y}} \rangle$ depends on at most one bit of \mathbf{y} , and that there are circuits C_1 and C_2 which have an AND and OR output gate (respectively) and compute the constant 0 function. Applying Theorem 20, every semantic property P over \mathcal{C} computable in \mathcal{A} is trivial, so the \mathcal{C} -BBH for \mathcal{A} is true. ◀