From Ignorant Decision Trees to Barriers to Infinitely-Often Circuit Lower Bounds

Hanlin Ren

September 12, 2023

1 Ignorant Decision Trees

It is known that *infinitely-often circuit upper bounds* in relativized worlds (i.e., $\mathcal{C}^{\mathcal{O}} \subseteq \mathsf{i.o.SIZE}^{\mathcal{O}}[O(n)]$) are equivalent to proving that the decision tree version of $\mathcal{C}$ cannot solve the Missing-String problem on infinitely many input lengths [VW23, Theorem 32]. Indeed, several relativization barriers to proving circuit lower bounds [BFT98, Aar06] can be interpreted as proving lower bounds for Missing-String against several kinds of decision trees ($\mathsf{MA}^{\mathsf{dt}}$, $\mathsf{PP}^{\mathsf{dt}}$). However, the circuit upper bounds in the relativized worlds constructed in [BFT98, Aar06] actually hold *almost-everywhere* ($\mathsf{MA}^{\mathcal{O}}_{\mathsf{EXP}} \subseteq \mathsf{P}^{\mathcal{O}}/_{\mathrm{poly}}$ and $\mathsf{PEXP}^{\mathcal{O}} \subseteq \mathsf{P}^{\mathcal{O}}/_{\mathrm{poly}}$). In this note, we observe that [BFT98, Aar06] actually proved something stronger than lower bounds for Missing-String, which enabled them to construct an oracle world with *almost-everywhere* circuit upper bounds.

We will call the stronger property *ignorance*. To define this property, consider the setting of the Missing-String problem. The input is a list of strings $x_1, x_2, \ldots, x_K \in \{0,1\}^n$. We have some decision tree computing a function $f : \{0,1\}^{nK} \to \{0,1\}^n$. We want to show that this function does not solve Missing-String, i.e., there exists $x_{1\sim K}$ such that $f(x_{1\sim K}) \in \{x_1, \ldots, x_K\}$. One possible approach to proving this is to show that for some inputs $x_{1\sim K}$ and index $k \in [K]$, $f(x_{1\sim K})$ is *ignorant* about some x_k, in the following sense: for every $x'_{1\sim K}$ that only differs from $x_{1\sim K}$ at x_k (i.e., $\forall k' \neq k$, $x'_{k'} = x_{k'}$), we have $f(x'_{1\sim K}) = f(x_{1\sim K})$. If $f(x_{1\sim K})$ is *ignorant* about x_k, then clearly f does not solve Missing-String: let $\hat{x}_{1\sim K} \leftarrow x_{1\sim K}$, and replace $\hat{x}_k$ with $f(x_{1\sim K})$. Since $f(x_{1\sim K})$ is ignorant about x_k, it does not notice that x_k is replaced, hence $f(\hat{x}_{1\sim K}) = f(x_{1\sim K})$. However, $f(x_{1\sim K}) = \hat{x}_k \in \{\hat{x}_1, \ldots, \hat{x}_K\}$, therefore f does not solve the Missing-String problem.

Note that the definition of ignorance still makes sense when the length of each substring x_i is different from the number of outputs. Say that f has n outputs and each x_i has length m. We can still define that $f(x_{1\sim K})$ is *ignorant* about x_k if for every $x'_{1\sim K}$ that only differs from $x_{1\sim K}$ at x_k, we have $f(x'_{1\sim K}) = f(x_{1\sim K})$.

It turns out that the relativized circuit upper bounds in [BFT98, Aar06], which are lower bounds for decision trees solving Missing-String, are actually proved by showing the ignorance of $\mathsf{MA}^{\mathsf{dt}}$ and $\mathsf{PP}^{\mathsf{dt}}$. In fact, they also show the ignorance of $\mathsf{MA}^{\mathsf{dt}}$ and $\mathsf{PP}^{\mathsf{dt}}$ in the general case when $m \neq n$. **Ignorance implies relativization barriers to proving not only almost-everywhere circuit lower bounds, but also *infinitely-often* circuit lower bounds.**

1.1 Ignorance of $\mathsf{MA}^{\mathsf{dt}}$

We define the decision tree version of MA, called $\mathsf{MA}^{\mathsf{dt}}$. An $\mathsf{MA}^{\mathsf{dt}}$ decision tree is essentially a disjunction of probabilistic decision trees; it is easy to see that for oracle $\mathcal{O}$, languages in $\mathsf{MA}^{\mathcal{O}}$ are computed by $\mathsf{MA}^{\mathsf{dt}}$ decision trees over the truth table of $\mathcal{O}$ as inputs.

Definition 1. An $\mathsf{MA}^{\mathsf{dt}}$ decision tree $M : \{0,1\}^n \to \{0,1,\perp\}$ is specified by a family of probabilistic decision trees $M_1, M_2, \ldots, M_S$, where S is the *disjunction fan-in* of M. The *decision tree depth* of M is the largest depth of any decision tree appearing in $M_1, M_2, \ldots, M_S$. On input $x \in \{0,1\}^n$:

- we say $M(x) = 1$, if there exists $s \in [S]$ such that $\Pr[M_s(x) = 1] \geq 2/3$;
- we say $M(x) = 0$, if for every $s \in [S]$, $\Pr[M_s(x) = 1] \leq 1/3$;
- otherwise, we say $M(x) = \perp$.

Theorem 2 ([BFT98])**.** *Let n, m, K, S, d be the following parameters such that $K > (1 + 4nd)n$.*

n	m	K	S	d
no. of dec. trees	*length of each string*	*no. of strings*	*disjunction fan-in*	*decision tree depth*

Let $D_1, D_2, \ldots, D_n$ be $\mathsf{MA}^{\mathsf{dt}}$ decision trees of disjunction fan-in S and decision tree depth d. Then there exist $x_1, x_2, \ldots, x_K \in \{0,1\}^m$ and an index $k \in [K]$ such that $D_i(x_{1\sim K})$ are ignorant of x_k in the following sense: There exists an answer string $ans \in \{0,1\}^n$ such that for every $\hat{x}_k \in \{0,1\}^m$, if we replace x_k with $\hat{x}_k$ and denote the obtained mK-bit input as $\hat{x}$, then $D_i(\hat{x}) \neq 1 - ans_i$ for every $i \in [n]$.

Proof. We start with $x_k = 0^m$ for all k. Then we will modify $x_{1\sim K}$ step by step until we arrive at the input $x_{1\sim K}$ that satisfies the theorem. During this process, some $k \in [K]$ will be *frozen* in the sense that the string x_k will never change again; some $i \in [n]$ will be *forced* in the sense that for every string $x'_{1\sim K}$ that is consistent with $x_{1\sim K}$ on frozen indices, it is true that $D_i(x'_{1\sim K}) \neq 0$. Initially, none of k is frozen and none of i is forced. Each step will force exactly one $i \in [n]$, so there are at most n steps. We will use $x^t = (x^t_1, \ldots, x^t_K)$ to denote the input instance at step t; thus $x^0_k = 0^n$ for every k.

Consider the t-th step ($t \geq 1$). Let k be the first unfrozen index in $[K]$ (we will show such k exists). Consider the following question: is there a string $\hat{x}_k \in \{0,1\}^m$ and an unforced index $i \in [n]$ such that, if we let $\hat{x} = x^{t-1}$ and replace x^{t-1}_k with $\hat{x}_k$, then $D_i(\hat{x}) = 1$?

If the answer is no, then we can stop here and obtain the required input $x_{1\sim K}$ and index k. In particular, let $ans \in \{0,1\}^n$ be the string such that $ans_i = 1$ if and only if i is forced. Let $\hat{x}_k \in \{0,1\}^m$ be arbitrary, $\hat{x}$ be the mK-bit input obtained by replacing x^{t-1}_k with $\hat{x}_k$. For each $i \in [n]$:

- If $D_i(\hat{x}) = 1$, then i have to be forced; otherwise the answer to the above question is yes (there exists a string $\hat{x}_k$ and an unforced index i).
- If $D_i(\hat{x}) = 0$, then i is not forced (since $\hat{x}$ and x^{t-1} are consistent on frozen indices).

On the other hand, if the answer is yes, then we freeze k and force i. We also set $x^t = \hat{x}$ as defined above, therefore $D_i(x^t) = 1$. We need to freeze some other indices as well to force i, i.e., to guarantee that $D_i(x'_{1\sim K}) \neq 0$ for every x' consistent with x^t on indices that are frozen (at the end of this step). Since $D_i(x^t) = 1$, there is a probabilistic decision tree in D_i, say D_{ij}, that has acceptance probability at least $2/3$ on input x^t. For every $k' \in [K]$ that is not frozen, if the fraction of decision trees in D_{ij} that queries $x_{k'}$ is at least $1/(4n)$, then we freeze k'. Now we can see that i is indeed forced: In the future, whenever we change some x_l, the acceptance probability of D_{ij} only gets influenced by $1/(4n)$. Since there are only n steps, the acceptance probability of D_{ij} will never drop below $2/3 - n \cdot 1/(4n) \geq 5/12 > 1/3$. It is also clear that we freeze at most $1 + 4nd$ many indices k'. Since $(1 + 4nd)n < K$, we will never run out of unfrozen indices. □

1.2 Ignorance of $\mathsf{PP}^{\mathsf{dt}}$

The decision tree version of PP, called $\mathsf{PP}^{\mathsf{dt}}$, is simply *polynomial threshold functions with bounded magnitudes*. We provide a formal definition:

Definition 3. A $\mathsf{PP}^{\mathsf{dt}}$ decision tree $M : \{0,1\}^n \to \{0,1\}$ is specified by a polynomial $P : \{0,1\}^n \to \mathbb{Z}$ such that $M(x) = 1$ if and only if $P(x) > 0$. (By replacing P with $2P - 1$, we may assume that $P(x) \neq 0$ for every $x \in \{0,1\}^n$.) The *degree* of M is the degree of P, and the *magnitude* of M is the smallest integer N such that $|P(x)| \leq N$ for every $x \in \{0,1\}^n$.

Theorem 4 ([Aar06])**.** *Let n, m, K, N, d be the following parameters such that $K \geq cn^2 d \log N$ for some large universal constant c.*

n	m	K	N	d
no. of dec. trees	*length of each string*	*no. of strings*	*magnitude*	*degree*

Let $D_1, D_2, \ldots, D_n$ be $\mathsf{PP}^{\mathsf{dt}}$ decision trees of degree d and magnitude N. Then there exist $x_1, x_2, \ldots, x_K \in \{0,1\}^m$ and an index $k \in [K]$ such that $D_i(x_{1\sim K})$ are ignorant of x_k in the following sense: For every $\hat{x}_k \in \{0,1\}^m$, if we replace x_k with $\hat{x}_k$ and denote the obtained mK-bit input as $\hat{x}$, then $D_i(\hat{x}) = D_i(x)$ for every $i \in [n]$.

Proof. Again, we start with $x_k = 0^m$ for all k. Then we proceed by steps, where in each step we make some changes to $x_{1\sim K}$. Again, let $x^t = (x_1^t, \ldots, x_K^t)$ denote the string $x_{1\sim K}$ at the t-th step.

To measure our progress, we need to define a potential function as follows. Let $P_i : \{0,1\}^{mK} \to \mathbb{Z}$ denote the polynomial underlying D_i. For each $i \in [n]$, $b \in \{\pm 1\}$ and $0 \leq l \leq \log N$, define

$$q_{i,b,l}(x) := 2^{2l-3} + (2^l + b \cdot P_i(x))^2.$$

The potential function measuring our progress would be

$$Q(x) := \prod_{i \in [n], b \in \{\pm 1\}, 0 \leq l \leq \log N} q_{i,b,l}(x).$$

In each step (say the t-th step), if there is no index k such that the input x^{t-1} and the index k satisfies the conclusion of our theorem, then we will modify x^{t-1} into x^t such that $Q(x^t) \geq 2Q(x^{t-1})$. Since for each i, b, l, we have $1/8 \leq q_{i,b,l}(x) \leq O(N^2)$, it follows that $-O(n \log N) \leq \log Q(x) \leq O(n \log^2 N)$. Hence, there will be at most $O(n \log^2 N)$ steps.

Now we specify each step. At the beginning of the t-th step, we have an input $x^{t-1} = (x_1^{t-1}, \ldots, x_K^{t-1})$. We say that $k \in [K]$ is *sensitive* to $i \in [n]$, if we can flip the sign of P_i by changing x_k^{t-1} in some way (and keep everything else unchanged). That is, there exists $\hat{x}_k \in \{0,1\}^m$ such that if we let $\hat{x} = x^{t-1}$ and replace x_k^{t-1} with $\hat{x}_k$, then $M_i(x^{t-1}) \neq M_i(\hat{x})$. By definition, if there exists a $k \in [K]$ that is not sensitive to any $i \in [n]$, then we are done.

Therefore, we consider the case that every $k \in [K]$ is sensitive to some $i \in [n]$. We start by picking a "heavy" i: there is some $i \in [n]$ such that at least K/n many indices k are sensitive to i. Let $k_1, k_2, \ldots, k_{K/n}$ be these indices. For each index k_j, there is a string $\hat{x}_{k_j}$ such that the following holds: if we let $\hat{x}^j = x^{t-1}$ and replace $(\hat{x}^j)_{k_j}$ with $\hat{x}_{k_j}$, then $M_i(x^{t-1}) \neq M_i(\hat{x}^j)$.

To monitor the potential function $Q(x)$, let us decompose $Q(x)$ as the product of two functions: $q(x) := q_{i,b,l}(x)$, and $v(x) := Q(x)/q(x)$ (everything else). Here, $b = -\mathrm{sign}(P_i(x^{t-1}))$ and l is the smallest integer such that $|P_i(x^{t-1})| \leq 2^l$. Then we have the following facts:

Fact 5. $q(x^{t-1}) \leq 3 \cdot 2^{2l-3}$.

Proof. $q(x^{t-1}) = 2^{2l-3} + (2^l + b \cdot P_i(x^{t-1}))^2 \leq 2^{2l-3} + 2^{2l-2} \leq 3 \cdot 2^{2l-3}$. ◇

Fact 6. *For every $j \in [K/n]$, $q(\hat{x}^j) \geq 9 \cdot 2^{2l-3}$.*

Proof. $q(\hat{x}^j) = 2^{2l-3} + (2^l + b \cdot P_i(\hat{x}^j))^2 \geq 2^{2l-3} + 2^{2l} \geq 9 \cdot 2^{2l-3}$. ◇

Fact 7. *For every $x \in \{0,1\}^{mK}$, $q(x) \geq 2^{2l-3}$.* ◇

Recall that our goal is to find some $x \in \{0,1\}^{mK}$ such that $Q(x) \geq 2Q(x^{t-1})$. After finding such x, we can set $x^t \leftarrow x$ and continue the next step.

Suppose that there is some $j \in [K/n]$ such that $v(\hat{x}^j) \geq (2/3) \cdot v(x^{t-1})$. Then we are done since

$$Q(\hat{x}^j) = q(\hat{x}^j) \cdot v(\hat{x}^j) \geq 3q(x^{t-1}) \cdot (2/3)v(x^{t-1}) \geq 2Q(x^{t-1}).$$

Otherwise, we use the following lemma of Nisan and Szegedy ([NS94], see also [Aar06, Lemma 1]):

Lemma 8. *Let $p : \{0,1\}^N \to \mathbb{R}$ be a real multilinear polynomial of degree at most $\sqrt{N}/7$, and suppose that $|p(x)| \leq \frac{2}{3}|p(0^N)|$ for every $x \in \{0,1\}^N$ with Hamming weight 1. Then there exists some $x \in \{0,1\}^N$ such that $|p(x)| \geq 6|p(0^N)|$.*

The idea is to change not one, but many sensitive indices k_j. Let $y \in \{0,1\}^{K/n}$ denote the set of sensitive indices we change. Define an input $\hat{x}^y$ as follows: start with $\hat{x}^y = x^{t-1}$, and for every $j \in [K/n]$ such that $y_j = 1$, replace $(\hat{x}^y)_{k_j}$ with $\hat{x}_{k_j}$. Let $w(y) = v(\hat{x}^y)$, it is easy to observe that:

Fact 9. *w is a polynomial of degree $O(nd \log N)$ over y.*

Proof. Note that each bit of $\hat{x}^y$ is a degree-1 polynomial over y, and that v is a polynomial of degree $O(nd \log N)$ over $\hat{x}^y$. ◇

Also note that for every $y \in \{0,1\}^{K/n}$ of Hamming weight 1, we have $w(y) = v(\hat{x}^j)$ for some j, and w is always positive over $\{0,1\}^{K/n}$, hence $|w(y)| \leq (2/3)|w(0^{K/n})| = (2/3)v(x^{t-1})$. Now, since $K \geq cn^2 d \log N$, we have $O(nd \log N) \leq \sqrt{K/n}/7$, and we can invoke Theorem 8 and obtain some $y \in \{0,1\}^{K/n}$ such that $w(y) \geq 6w(0^{K/n}) = 6v(x^{t-1})$. Then we are done since

$$Q(\hat{x}^y) = q(\hat{x}^y) \cdot v(\hat{x}^y) \geq (1/3)q(x^{t-1}) \cdot 6v(x^{t-1}) \geq 2Q(x^{t-1}).$$

It follows that for every x^{t-1}, either there is a k not sensitive to every i (in which we obtain what we want), or there is some x^t with $Q(x^t) \geq 2Q(x^{t-1})$. Hence the theorem is proved. □

2 Almost-Everywhere Circuit Upper Bounds in Relativized Worlds

Theorem 10 ([BFT98])**.** *There is an oracle $\mathcal{O}$ such that* pr-$\mathsf{MATIME}^{\mathcal{O}}[2^n] \subseteq \mathsf{SIZE}^{\mathcal{O}}[O(n)]$.

Proof. Let $M_1, M_2, \ldots, M_i, \ldots$ be an enumeration of $\mathsf{MATIME}[2^n]$ machines (this is a syntactical enumeration, i.e., some $M_i^{\mathcal{O}}$ might not satisfy the promise of MA computation). The interface of M_i is $M_i^{\mathcal{O}}(x, y; r)$, where $x \in \{0,1\}^n$ is the input string, $y \in \{0,1\}^{2^n}$ is Merlin's proof, and $r \in \{0,1\}^{2^n}$ is the internal randomness. We say $M_i^{\mathcal{O}}(x) = 1$ if there exists y such that $\Pr_r[M_i^{\mathcal{O}}(x, y; r) = 1] \geq 2/3$, say $M_i^{\mathcal{O}}(x) = 0$ if for every y, $\Pr_r[M_i^{\mathcal{O}}(x, y; r) = 1] \leq 1/3$, and say $M_i^{\mathcal{O}}(x) = \perp$ otherwise.

The oracle $\mathcal{O}$ receives inputs of the following form: $(a_1, a_2, \ldots, a_\ell, x, i)$. Here, for each $j \in [\ell]$, $|a_j| = 5 \cdot 2^j$; $2^{\ell-1} < |x| \leq 2^\ell$; and $i \in [|x|]$. We will also construct a sequence of strings $a_1^*, a_2^*, \ldots, a_\ell^*, \ldots$, where each a_i^* has length exactly $5 \cdot 2^i$, such that the following holds:

> For each $n \in \mathbb{N}$, $i \leq n$, and $x \in \{0,1\}^n$, let ℓ be the smallest integer such that $n \leq 2^\ell$. Then:
>
> - if $M_i^{\mathcal{O}}(x) = 1$, then $\mathcal{O}(a_1^*, a_2^*, \ldots, a_\ell^*, x, i) = 1$;
> - if $M_i^{\mathcal{O}}(x) = 0$, then $\mathcal{O}(a_1^*, a_2^*, \ldots, a_\ell^*, x, i) = 0$.

It is clear that the above implies pr-$\mathsf{MATIME}^{\mathcal{O}}[2^n] \subseteq \mathsf{SIZE}^{\mathcal{O}}[O(n)]$. Let $L \in$ pr-$\mathsf{MATIME}^{\mathcal{O}}[2^n]$ be decided by the machine M_i, then for every $n \geq i$, a circuit for L on input length n works as follows: Let ℓ be the smallest integer such that $n \leq 2^\ell$, hardcode $a_1^*, a_2^*, \ldots, a_\ell^*$, and on input $x \in \{0,1\}^n$, return $\mathcal{O}(a_1^*, \ldots, a_\ell^*, x, i)$.

Now we construct the strings $\{a_i^*\}$. Our construction works in stages. For each $\ell \geq 1$, the ℓ-th stage will construct a_ℓ^* given $a_1^*, \ldots, a_{\ell-1}^*$. We will guarantee that at the end of the ℓ-th stage, every entry of $\mathcal{O}$ of the form

$$\mathcal{O}(a_1^*, a_2^*, \ldots, a_{\ell-1}^*, a_\ell^*, a'_{\ell+1}, \ldots, a'_{\ell'}, x', i') \quad \text{where } \ell' > \ell \tag{1}$$

is *not* fixed, and every entry *not* of this form is fixed. We will also guarantee that at the end of the ℓ-th stage, every $M_i^{\mathcal{O}}(x)$ where $2^{\ell-1} < |x| \leq 2^\ell$ and $i \leq |x|$ will be fixed in the following sense: there is a bit $ans(i, x)$ such that no matter how we modify the entries in Equation 1 in the future, $M_i^{\mathcal{O}}(x) \neq 1 - ans(i, x)$ always holds.

Suppose that the strings $a_1^*, \ldots, a_{\ell-1}^*$ are already constructed, we want to construct a_ℓ^*. We enumerate every $2^{\ell-1} < n \leq 2^\ell$, $i \in [n]$, and $x \in \{0,1\}^n$. Each (i, x) induces an $\mathsf{MA}^{\mathsf{dt}}$ decision tree $M_i^{\mathcal{O}}(x)$. Values n, i, x are hardcoded in the decision tree, and the input of this decision tree consists of the portions of $\mathcal{O}$ on inputs of the following form:

$$\mathcal{O}(\underbrace{a_1^*, a_2^*, \ldots, a_{\ell-1}^*}_{\text{fixed}}, \underbrace{a'_\ell, a'_{\ell+1}, \ldots, a'_{\ell'}, x', i'}_{\text{arbitrary}}) \quad \text{where } \ell' > \ell - 1.$$

We partition these inputs into $K := 2^{5 \cdot 2^\ell}$ blocks according to a'_ℓ. For each $a'_\ell \in \{0,1\}^{5 \cdot 2^\ell}$, the a'_ℓ-th block consists of the portions of $\mathcal{O}$ on inputs of the following form:

$$\mathcal{O}(\underbrace{a_1^*, a_2^*, \ldots, a_{\ell-1}^*, a'_\ell}_{\text{fixed}}, \underbrace{a'_{\ell+1}, \ldots, a'_{\ell'}, x', i'}_{\text{arbitrary}}) \quad \text{where } \ell' > \ell - 1.$$

Note that each block has length $m \leq 2^{2^n}$. Also note that each $M_i^{\mathcal{O}}(x)$ has disjunction fan-in $S \leq 2^{2^n}$ and decision tree depth $d \leq 2^n$. There are at most $n' := \sum_{n=2^{\ell-1}+1}^{2^\ell} n 2^n \leq 2^{2^\ell + \ell + 1}$ $\mathsf{MA}^{\mathsf{dt}}$ decision trees we need to consider.

Now, since $K > (1 + 4n'd)n'$, we invoke Theorem 2 and obtain an oracle $\mathcal{O}$ and an index a_ℓ^* such that the following holds. There exists a function ans that maps each (i, x) into a bit, such that for every oracle $\mathcal{O}'$ that is the same as $\mathcal{O}$ except on inputs of the following form:

$$\mathcal{O}(\underbrace{a_1^*, a_2^*, \ldots, a_{\ell-1}^*, a_\ell^*}_{\text{fixed}}, \underbrace{a'_{\ell+1}, \ldots, a'_{\ell'}, x', i'}_{\text{arbitrary}}) \quad \text{where } \ell' > \ell - 1, \tag{2}$$

and for every (i, x), we have $M_i^{\mathcal{O}'}(x) \neq 1 - ans(i, x)$.

Hence, we fix every input of $\mathcal{O}$ not of the form Equation 2. We are free to modify inputs of the form Equation 2 in the future and it will never happen that $M_i^{\mathcal{O}}(x) = 1 - ans(n, i, x)$. We also fix

$$\mathcal{O}(a_1^*, a_2^*, \ldots, a_\ell^*, x, i) = ans(i, x)$$

for every (i, x). Now, the unfixed portion of $\mathcal{O}$ is exactly Equation 1 and this concludes the ℓ-th step. □

I believe that using the same argument, we can show a relativized world where $\mathsf{PTIME}^{\mathcal{O}}[2^n] \subseteq \mathsf{SIZE}^{\mathcal{O}}[O(n)]$.

Remark 11 (Bounded Relativization)**.** In Theorem 2, we can find $x_1, x_2, \ldots, x_k$ in $\mathrm{poly}(mnKSd)$ time with a PH oracle. However, in the oracle construction in Theorem 10, we need to feed Theorem 2 with instances of length $m \leq 2^{2^n}$.[1] Therefore, the trivial upper bound on the complexity of $\mathcal{O}$ is only *doubly exponential time* with PH oracles.

[1]The instance length is doubly exponential because we want *exponential-time* MA classes to have small circuits. If we want *super-polynomial time* MA classes to have small circuits, we still need instances of length $2^{n^{\omega(1)}}$.

References

[Aar06] Scott Aaronson. Oracles are subtle but not malicious. In *Conference on Computational Complexity* (CCC), pages 340–354. IEEE Computer Society, 2006. doi:10.1109/CCC.2006.32. 1, 3, 4

[BFT98] Harry Buhrman, Lance Fortnow, and Thomas Thierauf. Nonrelativizing separations. In *CCC*, pages 8–12, 1998. doi:10.1109/CCC.1998.694585. 1, 2, 4

[NS94] Noam Nisan and Mario Szegedy. On the degree of Boolean functions as real polynomials. *Comput. Complex.*, 4:301–313, 1994. doi:10.1007/BF01263419. 4

[VW23] Nikhil Vyas and R. Ryan Williams. On oracles and algorithmic methods for proving lower bounds. In *Innovations in Theoretical Computer Science Conference* (ITCS), volume 251 of *LIPIcs*, pages 99:1–99:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.99. 1