Electronic Colloquium on Computational Complexity, Report No. 132 (2016)

On the Sensitivity Conjecture for Read-k Formulas

Mitali Bafna*
Indian Institute of Technology, Madras, Chennai, India
mitali.bafna@gmail.com

Satyanarayana V. Lokam
Microsoft Research, Bangalore, India
satya@microsoft.com

Sébastien Tavenas
Microsoft Research, Bangalore, India
t-sebat@microsoft.com

Ameya Velingker*
Carnegie Mellon University, Pittsburgh, USA
avelingk@cs.cmu.edu

August 18, 2016

Abstract

Various combinatorial/algebraic parameters are used to quantify the complexity of a Boolean function. Among them, *sensitivity* is one of the simplest and *block sensitivity* is one of the most useful. Nisan (1989) and Nisan and Szegedy (1991) showed that block sensitivity and several other parameters, such as certificate complexity, decision tree depth, and degree over $\mathbb{R}$, are all polynomially related to one another. The sensitivity conjecture states that there is also a polynomial relationship between sensitivity and block sensitivity, thus supplying the "missing link".

Since its introduction in 1991, the sensitivity conjecture has remained a challenging open question in the study of Boolean functions. One natural approach is to prove it for special classes of functions. For instance, the conjecture is known to be true for monotone functions, symmetric functions, and functions describing graph properties.

In this paper, we consider the conjecture for Boolean functions computable by read-k formulas. A read-k formula is a tree in which each variable appears at most k times among the leaves and has Boolean gates at its internal nodes. We show that the sensitivity conjecture holds for read-once formulas with gates computing symmetric functions. We next consider *regular* formulas with OR and AND gates. A formula is regular if it is a leveled tree with all gates at a given level having the same fan-in

*This work was done while the author was an intern at Microsoft Research, Bangalore, India.

ISSN 1433-8092

and computing the same function. We prove the sensitivity conjecture for constant depth regular read-k formulas for constant k.

1 Introduction

Sensitivity and block sensitivity are two important complexity parameters of Boolean functions. The sensitivity conjecture states that these two parameters are polynomially related. A long-standing open question is to prove (or disprove) this conjecture. In this paper, we prove the conjecture for several subclasses of functions computable by read-k formulas.

The sensitivity $\mathrm{s}(f)$ of a Boolean function f is the maximum (over all inputs) number of coordinate dimensions along which the value of the function changes. This notion was first introduced by Cook et al. [9] to prove lower bounds on the parallel complexity (in the CREW PRAM model) of Boolean functions. Nisan [16] introduced the more general definition of *block sensitivity.* The block sensitivity $\mathrm{bs}(f)$ of a Boolean function f is the maximum (again, over all inputs) number of disjoint subsets of coordinate dimensions such that flipping all values of a given input in any of these subsets results in flipping the value of the function. Nisan proved that block sensitivity asymptotically captures the CREW PRAM complexity of all Boolean functions. Remarkably, Nisan also showed that several other complexity parameters of Boolean functions such as certificate complexity, decision tree depth, and randomized decision tree depth are polynomially related to block sensitivity. Subsequently, Nisan and Szegedy [17] showed that block sensitivity and degree of polynomials (approximately) representing a Boolean function over $\mathbb{R}$ are polynomially related.

Hence, a number of combinatorial/algebraic parameters describing complexity of Boolean functions are all polynomially related to each other, but sensitivity has so far resisted such a polynomial equivalence with any of these other parameters. In fact, Nisan and Szegedy posed this as the *sensitivity vs. block sensitivity* question and since then, this question has come to be known as the "sensitivity conjecture". More than two decades later, proving (or disproving) this conjecture still remains a foundational challenge in the study of Boolean functions. In recent times, this quest has become even more intriguing as other complexity parameters such as quantum query complexity (both exact and two-sided error versions) have been shown to be polynomially related to block sensitivity [5, 7]. At the same time, the sensitivity conjecture has been shown to be related to a number of other conjectures and open questions in Boolean function complexity, as illustrated in the survey [13].

The best known universal (applicable to all functions) upper bound on block sensitivity remains exponential in sensitivity [19] (see [14], [3], [21] for more refined upper bounds). In the other direction, Rubinstein [18] gives an example function where the gap between sensitivity and block sensitivity is quadratic (see [4] and references therein for improvements in constants). Thus the challenge is to close this gap between quadratic and exponential relations between block sensitivity and sensitivity.

Several approaches have been proposed in the literature to attack the sensitivity conjecture. Gotsman and Linial [12] showed that the degree vs. sensitivity problem is equivalent to a combinatorial problem on the maximum degree of induced subgraphs of the Boolean cube. Aaronson [1] (see also [6]) stated a problem about certain two-colorings of the integer lattice whose solution would imply the sensitivity conjecture. Recently, Gilmer et al. [10] formulated an approach to the degree vs. sensitivity problem using lower bounds on a two-party communication game. Even more recently, Gopalan et al. [11] prove an ℓ_2-approximate version of the degree vs. sensitivity conjecture (the original one needs an ℓ_∞-approximation). They also formulate the notion of tree sensitivity and a robust analog of the degree vs. sensitivity conjecture.

To make progress on our understanding of this problem, researchers also studied the conjecture on special classes of Boolean functions. It is trivial to see that the conjecture holds for monotone functions and symmetric functions. A natural question, then, is if the sensitivity conjecture holds when the function is invariant under other groups of symmetries. Turán [22] proved that for Boolean functions that describe graph properties (edges are the Boolean variables) sensitivity is $\Omega(\sqrt{n})$ and hence the conjecture holds for graph properties. Chakraborty [8] studied minterm-transitive Boolean functions and showed that for such functions sensitivity is $\Omega(n^{1/3})$, thus showing the conjecture for this class of functions. Sun [20] studied block sensitivity for Boolean functions invariant under any transitive permutation group and showed that such functions must have block sensitivity $\Omega(n^{1/3})$.

Our Results: We prove the sensitivity conjecture for another restricted class of Boolean functions, namely certain functions computed by read-k formulas. A *read-k formula* is a tree whose internal nodes are Boolean gates, e.g., AND and OR, and leaves are literals of input variables with the restriction that each variable (as negated or non-negated literal) appears at most k times among the leaves. Such a formula computes a Boolean function in a natural way from the leaves to the root. A formula is called *regular* if all gates at a given depth are the same type and have the same fan-in.

In what follows, we will mainly focus on formulas composed of OR and AND gates. In particular we show that the sensitivity conjecture is true for read-$\log n$ regular formulas whose bottom fanins are sufficiently large.

Theorem 1. Regular read-$\log n$ with large bottom fanin. *Let f be a Boolean function, dependent on n variables, computed by a regular read-$(\log n)$ formula with bottom fan-in at least $\log^2 n$. Then*

$$\mathrm{s}(f) \geq \tilde{\Omega}\left(\mathrm{bs}(f)^{1/4}\right),$$

where the $\tilde{\Omega}$ notation hides some logarithmic terms.

We would like to remove the condition on the bottom fanin. We succeed in doing so when the read and depth of the formula are constants.

Theorem 2. Regular read-constant and constant depth. *Let f be computed by a regular read-k formula of depth-d for constants k and d such that all internal gates compute non-constant AND-OR functions. Then*

$$\mathrm{s}(f) = \Omega_{k,d}(\sqrt{\mathrm{bs}(f)}),$$

where the hidden constant is a (rapidly decreasing) function of k and d.

We present our main results (Theorem 1 and 2) on regular read-k formulas with AND and OR gates in Section 4. A crucial ingredient of our proofs is an application of the *Lovász Local Lemma* (LLL) to show that some literals can be assumed to occur in their positive form in such a formula without increasing the function's sensitivity and ensuring that any satisfying assignment of such a formula must have a large Hamming weight. However, in order to apply LLL, we need the bottom fan-in of such formulas to be large enough. So, we first prove the conjecture for formulas with large bottom fan-in. We then remove the restriction on the bottom fan-in by switching AND's of OR's to OR's of AND's (or vice versa). The idea is that if the formula is sufficiently large and the depth small, there has to be a layer L with large fanin. Then, by switching, we expand the layers under L and put L close to the bottom.

When specialized to read-once formulas with symmetric gates or to read-k DNF's our lower bounds on regular read-k formulas yield better dependence on k.

Theorem 3. Read-once with symmetric gates. *Let f be a Boolean function dependent on n variables and computed by a read-once formula with symmetric gates. Then, $\mathrm{s}(f) \geq \sqrt{n}$.*

We note that Hiroki Morizumi [15] proved a similar lower bound for read-once AND-OR formulas.

Theorem 4. Read-k DNF. *Let f be a Boolean formula dependent on n variables and computed by a read-k DNF. Then*

$$\mathrm{s}(f) \geq n^{1/3}/(k+2).$$

In particular, if $k \leq n^{\frac{1}{3}-\varepsilon} - 2$, then

$$\mathrm{s}(f) \geq n^{\varepsilon} \geq \mathrm{bs}(f)^{\varepsilon}.$$

Our proof of the conjecture for read-once formulas with symmetric gates appears in Section 3. The results on DNF's appear in Section 5.

2 Notations and Preliminaries

- In this paper, log will always denote the logarithm to base two.
- We will always assume that f is a Boolean function on n variables and moreover that it depends on all its variables.

2.1 Measures on Boolean functions

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. For $x \in \{0,1\}^n$ and $S \subseteq [n]$, we denote by x^S the vector obtained by flipping all the coordinates on x in S. For $x \in \{0,1\}^n$ and $z \in \{0,1\}$, we denote by $|x|_z$ the number of coordinates of x with the value z.

Definition 5. ***Sensitivity:***

- *The* sensitivity *of f at x is defined as the number of coordinates of x, which when flipped, will flip the value of f:*

$$\mathrm{s}(f,x) := \left|\left\{i \in [n] : \; f(x) \neq f\left(x^{\{i\}}\right)\right\}\right|.$$

- *For $z \in \{0,1\}$, the z-*sensitivity *of f is defined as the maximum sensitivity of f at an input in $f^{-1}(z)$:* $\mathrm{s}_z(f) := \max\{\mathrm{s}(f,x) : \; f(x) = z\}$.

- *Finally, the* sensitivity of f *is the maximum sensitivity of f among all inputs:* $\mathrm{s}(f) := \max\{\mathrm{s}(f,x) : \; x \in \{0,1\}^n\} = \max\{\mathrm{s}_0(f), \mathrm{s}_1(f)\}$.

Definition 6. *The* block sensitivity of f at x, *denoted* $\mathrm{bs}(f,x)$ *is the maximum number of* disjoint *subsets $S_1, \ldots, S_b$ of $[n]$ such that for every i, $f(x) \neq f(x^{S_i})$. The z-block sensitivity and block sensitivity of f are defined similar to the case of sensitivity. In particular,* $\mathrm{bs}(f) := \max\{\mathrm{bs}(f,x) : \; x \in \{0,1\}^n\}$.

Definition 7. *A* certificate of f on x *is a subset $S \subseteq [n]$ such that $f(y) = f(x)$ whenever $y_i = x_i, \forall i \in S$. The size of the certificate S is $|S|$.*

The certificate complexity of f on x *denoted by* $\mathrm{C}(f,x)$ *is the size of a smallest certificate of f on x. The* certificate complexity of f *denoted by* $\mathrm{C}(f)$ *is* $\max_x \mathrm{C}(f,x)$. *For $z \in \{0,1\}$, the z-certificate complexity of f denoted by* $\mathrm{C}_z(f)$, *is* $\max_{x \in f^{-1}(z)} \mathrm{C}(f,x)$.

We will use the following known results.

Lemma 8. *For any Boolean function f and $z \in \{0,1\}$,* $\mathrm{C}_z(f) \geq \mathrm{bs}_z(f) \geq \mathrm{s}_z(f)$.

The first inequality above is from [16] and the second inequality is obvious from definitions.

Theorem 9 ([4])**.** *For any Boolean function f and $z \in \{0,1\}$,*

$$\mathrm{C}_z(f) \geq \frac{3\mathrm{bs}_{1-z}(f)}{2\mathrm{s}_{1-z}(f)} - \frac{1}{2}.$$

2.2 Formulas

Definition 10. ***Regular Read-k Formulas:***

- *A formula* $\mathtt{C}$ *is said to be $(a_1, \ldots, a_d)$-*regular *if it is a layered tree of depth d whose leaves are input variables or their negations and all internal nodes at a given layer i, $1 \leq i \leq d$, are gates of the same kind and the same fanin a_i. The layers are numbered 1 through $d+1$ from the root (output) to the leaves (inputs). We will often denote the gates at the layer d by* bottom gates. *In this paper, we only consider both formulas of alternating layers of AND and OR gates (we could start at the root with either gate and then alternate) and formulas with symmetric gates.*

- *A formula is* read-k *if each variable (either in its negated or non-negated form) appears at most k times among its leaves.*

  One can argue that by replicating the arguments, we can always assume that the formula is in regular form. However, this idea does not work here because by doing this transformation, we would increase the read-multiplicity of the formula.

2.3 Lovász local lemma

We will make use of the Lovász Local Lemmas:

Lemma 11. *[The Lovász Local Lemma: Symmetric Case] Let $A_1, \ldots, A_n$ be events in an arbitrary probability space. Suppose that each event A_i is mutually independent of a set of all the other events A_j but at most d and that $\Pr[A_i] \leq p$ for all $1 \leq i \leq n$.*

If $ep(d+1) < 1$, then $\Pr\left[\bigcap \overline{A_i}\right] > 0$.

We will also use the general version of this lemma. Both versions can be found, e.g., in [2].

Lemma 12. *[Lovász Local Lemma: General Case] Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ be events in an arbitrary probability space. For $A \in \mathcal{A}$ let $\Gamma(A)$ denote a subset of $\mathcal{A}$ such that A is independent from the collection of events $\mathcal{A} \setminus (\{A\} \cup \Gamma(A))$.*

If there exists an assignment of reals $x : \mathcal{A} \to (0,1)$ to the events such that

$$\forall A \in \mathcal{A} : \Pr(A) \leq x(A) \prod_{B \in \Gamma(A)} (1 - x(B)),$$

then $\Pr\left[\bigcap \overline{A_i}\right] > 0$.

3 Read-once formulas with symmetric gates

In this section, we prove the sensitivity conjecture for read-once formulas with *symmetric* gates. The read-multiplicity is more restrictive than the model we will consider later but the gates we allow are more powerful.

Definition 13. *Let g be a non-constant symmetric function on m inputs. We define $\tau(g)$ to be the minimal weight of an input $x \in \{0,1\}^m$ such that $g(x) \neq g(\vec{0})$:*

$$\tau(g) := \min\left\{ i \mid \ |x|_1 = i \implies g(x) \neq g(\vec{0}) \right\}.$$

Theorem 14. *Let f be a Boolean function computed by a read-once formula $\mathtt{C}$ with symmetric gates. Then, $\mathrm{s}_0(f)\mathrm{s}_1(f) \geq n$.*

Proof. We prove it by induction on the depth of the formula $\mathtt{C}$. If the depth of the formula is 1, then f is a symmetric function on n variables. Let $z = f(\vec{0})$ and $t := \tau(f)$. By Definition 13, when $|x|_1 = t - 1$, $f(x) = z$ and when $|y|_1 = t$, $f(y) = 1 - z$. It follows immediately that $s_z(f, x) \geq n - t + 1$ and $s_{1-z}(f, y) \geq t$. As $1 \leq t \leq n$, we have $\mathrm{s}_0(f)\mathrm{s}_1(f) \geq t(n - t + 1) \geq n$.

Now assume that the theorem is true for all depths $\leq d$. We prove it for depth $d + 1$.

So $f = h(g_1, \ldots, g_m)$, where h is symmetric and each g_i is computed by a read-once formula with symmetric gates, of depth at most d. Let every g_i be a function on n_i variables with $a_i = \mathrm{s}_0(g_i)$ and $b_i = \mathrm{s}_1(g_i)$. By the inductive hypothesis, we know that $a_i b_i \geq n_i$. Since $n = \sum_{i=1}^m n_i$, we have that, $\sum_{i=1}^m a_i b_i \geq n$. Without loss of generality, we may assume that $a_1 \geq a_2 \geq \ldots \geq a_m$ and $b_{\pi(1)} \geq b_{\pi(2)} \geq \ldots \geq b_{\pi(m)}$ for a suitable permutation π of $[m]$. Let $A_j := \sum_{i=1}^j a_i$ and $B_j := \sum_{i=1}^j b_{\pi(i)}$. Let $t := \tau(h)$ so $h(x) = z$ for all x with $|x|_1 = t - 1$ and $h(y) = 1 - z$ for all y with $|y|_1 = t$.

Since the formula is read-once, the g_i depend on disjoint sets of variables, and so it is easy to see that for all S with $|S| = t-1$, we can find an assignment σ to all the variables of f such that (i) $g_i(\sigma_i) = 1$ for exactly those $i \in S$ and (ii) for $i \notin S$, $g_i(\sigma_i) = 0$ and g_i has $a_i = \mathrm{s}_0(g_i, \sigma_i)$ sensitive inputs. It follows that

$$\mathrm{s}_z(f) \geq \max_{\substack{S \subseteq [n] \\ |S| = m-t+1}} \{\sum_{i \in S} a_i\} = A_{m-t+1}.$$

Similarly,

$$s_{1-z}(f) \geq \max_{\substack{S \subseteq [n] \\ |S| = t}} \{\sum_{i \in S} b_i\} = B_t.$$

So,

$$\mathrm{s}_0(f)\mathrm{s}_1(f) \geq A_{m-t+1}B_t = (a_1 + \ldots + a_{m-t+1})(b_{\pi(1)} + \ldots + b_{\pi(t)}).$$

The proof is completed by the following claim which is proved just after.

Claim 15. *For any t, $1 \leq t \leq m$, $A_{m-t+1}B_t \geq \sum_{i=1}^m a_i b_i$.*

We therefore conclude that $\mathrm{s}_0(f)\mathrm{s}_1(f) \geq A_{m-t+1}B_t \geq \sum_{i=1}^m a_i b_i \geq n$.

□

Proof of Claim 15. Let $A := A_{m-t+1} = \{a_1, \ldots, a_{m-t+1}\}$ and $B := B_t = \{b_{\pi(1)}, \ldots, b_{\pi(t)}\}$.

Let S be the set of terms in the expansion of AB, i.e. $S = \{a_j b_k \mid a_j \in A, b_k \in B\}$.

Consider the summation $\sum_{i=1}^m a_i b_i$. For every term $a_i b_i$ we will find a unique term in S which is greater equal $a_i b_i$. Let us define $A' = A \setminus \{a_{m-t+1}\}$ and $B' = B \setminus \{b_{\pi(t)}\}$.

Here are the possible cases for a_i, b_i:

- $a_i \in A', b_i \in B'$

  The term $a_i b_i$ itself.

- $a_i \in A', b_i \notin B'$

  The term $a_i b_{\pi(t)} \geq a_i b_i$ since $b_{\pi(1)} \geq \ldots \geq b_{\pi(t)} \geq \ldots b_{\pi(i)} \geq \ldots \geq b_{\pi(m)}$.

- $a_i \notin A', b_i \in B'$

  The term $a_{m-t+1} b_i \geq a_i b_i$ since $a_1 \geq \ldots \geq a_{m-t+1} \geq \ldots \geq a_m$.

- $a_i \notin A', b_i \notin B'$

  Any term in AB will be greater or equal than $a_i b_i$. Let the number of terms matched in the 3 cases above be v, so the number of terms remaining are $m - v$. As $1 \leq t \leq m$ we have that $(m-t)(t-1) = t(m-t+1) - m \geq 0$ and so $t(m-t+1) - v \geq m - v$. So we can match these remaining terms uniquely.

□

Corollary 16. *Let f be a Boolean function computed by a read once formula C with symmetric gates. Then, $s(f) \geq \sqrt{n} \geq \sqrt{bs(f)}$.*

Furthermore, this bound (i.e., the first inequality) is tight whenever n is a perfect square. To see the tightness of the bound, consider an OR of fan-in $\sqrt{n}$ over $\sqrt{n}$ disjoint AND's on $\sqrt{n}$ variables each. It is easy to see that both 0-sensitivity and 1-sensitivity of this function are exactly $\sqrt{n}$.

4 Read-k formulas

In the following, we will only consider AND-OR formulas (with positive and negative literals). In this section, we prove the sensitivity conjecture for read-k formulas with certain restrictions.

Theorem 17. *Let f be computed by a regular read-k formula of depth d with constants k and d such that any internal gate computes a non-constant function. Then, $\mathrm{s}(f) = \Omega_{k,d}(\sqrt{\mathrm{bs}(f)})$, where the hidden constant is a (rapidly decreasing) function of k and d.*

We prove this theorem in two stages:

- In Section 4.1, we first prove a lower bound for $\mathrm{s}(f)$ in terms of $\mathrm{bs}(f)$ when f is computed by a read-k regular formula with large bottom fanin.
- Then, in Section 4.2, we remove the condition on the bottom fanin by defining a normal form for formulas and then reducing a formula with small bottom fanin to one in the normal form where the previous step applies.

Notation: When $\mathtt{C}$ is an $(a_1, \ldots, a_d)$-regular formula with AND-OR gates we will use $A(\mathtt{C}, j)$ to denote the product,

$$A(\mathtt{C}, j) = \prod_{\substack{l \in [j] \\ l \text{ is a } \wedge\text{-gates level}}} a_l.$$

As most of the times, the function A will be used on the parameters $\mathtt{C}$ and $j = d - 2$, we will denote $A(\mathtt{C}, d-2)$ by A.

4.1 Large bottom fan-in

In this section, we give a lower bound for sensitivity in terms of block sensitivity for read-k regular formulas with large bottom fanin.

4.1.1 1-Sensitivity when bottom gates are AND gates

We will first prove a lower bound on the 1-sensitivity of such formulas. We will show that given a formula $\mathtt{C}$ it is possible to get an equivalent formula $\mathtt{C}'$ which has certain nice properties. Specifically, all inputs on which $\mathtt{C}'$ evaluates to 1 have large Hamming weight, which directly implies that the 1-sensitivity for this function is large.

Definition 18. *A parse tree P of a formula $\mathtt{C}$ computing f is a subcircuit which is recursively defined as follows:*

- *The output gate of $\mathtt{C}$ is in P.*

- *If an $\wedge$-gate belongs to P then all its children are also in P.*
- *If an $\vee$-gate belongs to P then exactly one of its children is in P.*

It is easy to see that f evaluates to 1 on an input x if and only if $\mathtt{C}$ contains a parse tree all of whose gates evaluate to 1. A simple induction also shows that every parse tree of a regular formula has $A(\mathtt{C}, d-1) = A$ bottom gates.

Definition 19. *The parse-read of* $\mathtt{C}$ *is the maximum number of times any variable appears in any parse tree.*

We will now consider two models. The first model is a (natural) restriction of our model of regular formulas: a variable can appear at most once under the same bottom gate. The second model is the general one without this restriction.

Lemma 20. *Let $(a_1, \ldots, a_d) \in (\mathbb{N} \setminus \{0\})^d$ with $a_d \geq 2\log 4k$. Let f be a non-constant function computed by an $(a_1, \ldots, a_d)$-regular read-k and parse-read p formula such that the bottom gates are $\wedge$-gates and such that each variable appears at most once under any bottom gate. Then*

$$\mathrm{s}_1(f) \geq \left(\frac{a_d - 2\log 4k + 1}{2p\log 4k}\right) A.$$

Proof. By regularity, any bottom gate of $\mathtt{C}$ is the parent of a_d literals. Let us group these literals into groups of size α whose value will be chosen later. The last group will be of size a_d modulo α. So we get $\lfloor a_d/\alpha \rfloor$ groups of α literals under every bottom gate. We want to modify $\mathtt{C}$ to $\mathtt{C}'$ such that each group contains at least one positive literal.

Let us randomly negate each variable. Each variable is independently chosen as positive or negative with probability $\frac{1}{2}$. Let A_i be the event that the i^{th} group has no positive literals (where the i^{th} group is taken over all groups under all bottom gates). So $\Pr[A_i] = \frac{1}{2^\alpha}$. Every event A_i is dependent on at most $k\alpha$ other A_j's. Using the symmetric version of the Lovász Local Lemma we get that, if $e(k\alpha + 1) \leq 2^\alpha$ then $\Pr\left[\bigcap \overline{A_i}\right] > 0$.

Notice that $\alpha = \lfloor 2\log(4k) \rfloor$ satisfies the previous inequality for all positive integers k. So there exists a new formula $\mathtt{C}'$ such that every group will have at least one positive literal. Let g be the function computed by $\mathtt{C}'$. Note that we now have a fixed σ such that for all x, $f(x \oplus \sigma) = g(x)$.

On any input $x \in g^{-1}(1)$ we get at least one parse tree in $\mathtt{C}'$ all of whose gates evaluate to 1. Consequently, on any input x in $g^{-1}(1)$, there are at least A bottom $\wedge$-gates of $\mathtt{C}'$ which evaluate to 1. As each variable can appear at most p times in any parse tree, we have that $\forall x \in g^{-1}(1)$,

$$|x|_1 \geq \left\lceil \frac{A}{p} \left\lfloor \frac{a_d}{\alpha} \right\rfloor \right\rceil \geq \frac{A(a_d - 2\log 4k + 1)}{2p\log 4k}.$$

Taking the input $x \in g^{-1}(1)$ with least Hamming weight we get that,

$$\mathrm{s}_1(f, x \oplus \sigma) = \mathrm{s}_1(g, x) \geq |x|_1 \geq \left(\frac{a_d - 2\log 4k + 1}{2p\log 4k}\right) A.$$

$\square$

It is interesting to notice that the proof can be turned into an algorithm for finding an input which has high sensitivity given any 1-input x. Namely, one just have to run the algorithmic version of Lovász Local Lemma to get the above bijection $\oplus\sigma$. Then find (by flipping the 1's from x) a locally minimal weight (under $\oplus\sigma$) assignment that still gives the output 1.

We will now remove the condition that every variable can occur at most once under any bottom gate. In doing so we will lose a factor of k in the lower bound while also demanding a stronger constraint on the bottom fanin.

Lemma 21. *Let $(a_1, \ldots, a_d) \in (\mathbb{N} \setminus \{0\})^d$ with $a_d \geq k \log(3k)$. Let f be a non-constant function computed by an $(a_1, \ldots, a_d)$-regular read-k and parse-read p formula such that the bottom gates are $\wedge$-gates. Then*

$$\mathrm{s}_1(f) \geq \left(\frac{a_d - k\log(3k) + 1}{kp\log(3k)}\right) A.$$

Proof. By regularity, any bottom gate of $\mathtt{C}$ is the parent of a_d literals. Let us group these literals into groups of size α (the last group will be of size a_d modulo α). So we get $\lfloor a_d/\alpha \rfloor$ groups of α literals under every bottom gate. We now get $\mathtt{C}'$ from $\mathtt{C}$ by ensuring that each group contains at least one positive literal.

Let us randomly negate each variable. Each variable is independently chosen as positive or negative with probability $\frac{1}{2}$. Let A_i be the event that the i^{th} group G_i has no positive literals and $\mathcal{A}$ be the set of these events. Let us also denote the set of variables present in G_i by V_i and its cardinality by d_i. We notice that if a variable appears in G_i positively and negatively, then $\Pr(A_i) = 0$. So, in all cases, $\Pr[A_i] \leq 1/2^{d_i}$. Let $\Gamma(A_i)$ be the set of the events in $\mathcal{A} \setminus \{A_i\}$ which are dependent of A_i. It means that for each event $A_j \in \Gamma(A_i)$ the set $V_i \cap V_j$ is not empty. As the function is read-k and as the variables in V_i already appear α times in G_i, it implies that $|\Gamma(A_i)| \leq (kd_i - \alpha)$.

Let us choose $\alpha = \lfloor k\log(3k) \rfloor \geq k\log(2k)$. We want to use Lemma 12 with the assignment of reals $x(A) = 1 - 2^{-1/k}$ for each event $A \in \mathcal{A}$. We have

$$\begin{aligned}
\frac{1}{\Pr(A_i)} x(A_i) \prod_{B \in \Gamma(A_i)} (1 - x(B)) &\geq 2^{d_i}\left(1 - 2^{-1/k}\right) 2^{-(kd_i - \alpha)/k} \\
&\geq \left(1 - 2^{-1/k}\right) 2^{\alpha/k} \\
&\geq \left(1 - \left(1 - \frac{1}{2k}\right)\right) 2^{\alpha/k} \\
&\geq \frac{2^{\alpha/k}}{2k} \\
&\geq 1.
\end{aligned}$$

Then by Lemma 12, $\Pr\left[\bigcap \overline{A_i}\right] > 0$. It means that there exists a new formula C' (where C' is derived from C by negating some variables) such that every group has at least one positive literal. Let g be the function computed by C'. As seen before, $g(x) = 1$, implies that there is at least one parse tree which has A bottom $\wedge$-gates which evaluate to 1. As each variable appears at most p times in this parse tree, we have that $\forall x \in g^{-1}(1)$,

$$|x|_1 \geq \frac{A}{p}\left\lfloor \frac{a_d}{\alpha} \right\rfloor \geq \frac{A(a_d - k\log(3k) + 1)}{kp\log(3k)}.$$

Taking the input $x \in g^{-1}(1)$ with least Hamming weight we get that,

$$\mathrm{s}_1(f,x) \geq |x|_1 \geq \frac{A(a_d - k\log(3k)+1)}{kp\log(3k)}.$$

□

4.1.2 The Sensitivity Conjecture for large bottom fan-in case

We will now combine previously known results with the statements proved in the section above to obtain some relations between sensitivity and block sensitivity.

The next lemma will help us relate the bound obtained for $\mathrm{s}_1(f)$ to $\mathrm{C}_1(f)$ of read-k regular formulas.

Lemma 22. *Let f be a Boolean function computed by an $(a_1,\ldots,a_d)$-regular formula $\mathtt{C}$. Then, $\mathrm{C}_1(f) \leq A(\mathtt{C}, d)$.*

Proof. Let us prove it by induction on d.

- If $d = 0$, then f is just a literal r associated to one variable x_i. The set $\{i\}$ is a certificate for any input of f, and so $\mathrm{C}_1(f) \leq 1$.
- Let us assume now that the lemma is true for a fixed d and prove it for formulas of depth $d+1$. So f is one of the following forms

  1. $g_1 \vee g_2 \vee \ldots \vee g_{a_1}$
  2. or $g_1 \wedge g_2 \wedge \ldots \wedge g_{a_1}$

  where $(g_1,\ldots,g_{a_1})$ are a_1 Boolean functions computed by some $(a_2,\ldots,a_{d+1})$-regular formulas.

  – In the first case, for any $x \in f^{-1}(1)$, at least one of the functions g_i is evaluated to 1. Hence $\mathrm{C}_1(f) \leq \max(\mathrm{C}_1(g_i))$ and so, by the induction hypothesis,

  $$\begin{aligned}\mathrm{C}_1(f) &\leq \prod_{\substack{l\in[2,d+1]\\ l \text{ is a } \wedge\text{-gates level}}} a_l \\ &= \prod_{\substack{l\in[1,d+1]\\ l \text{ is a } \wedge\text{-gates level}}} a_l.\end{aligned}$$

  The last equality comes from the fact that the first layer is a layer of $\vee$-gates.

  – In the second case, for any $x \in f^{-1}(1)$, all the functions g_i are evaluated to 1. Hence $\mathrm{C}_1(f) \leq \sum_{i=1}^{a_1} \mathrm{C}_1(g_i))$. Then by the induction hypothesis,

  $$\begin{aligned}\mathrm{C}_1(f) &\leq a_1 \prod_{\substack{l\in[2,d+1]\\ l \text{ is a } \wedge\text{-gates level}}} a_l \\ &= \prod_{\substack{l\in[1,d+1]\\ l \text{ is a } \wedge\text{-gates level}}} a_l.\end{aligned}$$

  The last equality comes from the fact that the first layer is a layer of $\wedge$-gates.

$\square$

Theorem 23. *Let f be a non-constant Boolean formula computed by an $(a_1, \ldots, a_d)$-regular read-k formula with parse-read p such that its bottom fanin a_d is larger or equal to $(3 \log 4k)$ and such that any variable appears at most one time under each bottom gate. Then*

$$\mathrm{s}(f) \geq \sqrt{\frac{\mathrm{bs}(f)}{10p \log 4k}}.$$

Moreover, when a variable can occur multiple times under each bottom gate and the bottom fan-in $a_d \geq 2k \log 3k$, we have

$$\mathrm{s}(f) \geq \sqrt{\frac{3\mathrm{bs}(f)}{10kp \log 3k}}.$$

Proof. Let us start by the first point of the theorem. By considering f or $\neg f$, we can assume that the bottom layer is composed of $\wedge$-gates. By Lemma 20, we have that,

$$\mathrm{s}_1(f) \geq \left(\frac{a_d - 2\log 4k + 1}{2p \log 4k} \right) A.$$

From Lemma 22 we have $C_1(f) \leq a_d A$. Since $a_d \geq 3 \log 4k$, $a_d - 2\log 4k \geq a_d/3$,

$$\mathrm{s}_1(f) \geq \frac{3A + \mathrm{C}_1(f)}{6p \log 4k} \geq \frac{\mathrm{C}_1(f) + 1/2}{6p \log 4k}.$$

Using Lemma 8 we get, $\mathrm{s}(f) \geq \mathrm{s}_1(f) \geq \frac{\mathrm{bs}_1(f)}{6p \log 4k}$. We also get by Theorem 9,

$$\mathrm{s}(f)^2 \geq \mathrm{s}_1(f) \cdot \mathrm{s}_0(f) \geq \frac{\mathrm{bs}_0(f)}{4p \log 4k}.$$

Since $\mathrm{bs}(f) = \max(\mathrm{bs}_1(f), \mathrm{bs}_0(f))$,

$$5\mathrm{s}^2 \geq 2\mathrm{s}^2 + 3\mathrm{s} \geq \frac{\mathrm{bs}_0(f)}{2p \log 4k} + \frac{\mathrm{bs}_1(f)}{2p \log 4k} \geq \frac{\mathrm{bs}(f)}{2p \log 4k}.$$

Consequently, $\mathrm{s} \geq \sqrt{\frac{\mathrm{bs}(f)}{10p \log 4k}}$, proving the first part. The second part of the theorem follows analogously using Lemma 21. $\square$

The following corollary follows from the lower bound for sensitivity proved in [19].

Corollary 24. *Let f be a non-constant Boolean formula computed by an $(a_1, \ldots, a_d)$-regular read-$(\log n)$ formula with bottom fan-in at least $\log^2 n$. Then*

$$\mathrm{s}(f) \geq \tilde{\Omega}\left(\mathrm{bs}(f)^{1/4}\right)$$

where the $\tilde{\Omega}$ notation hides some logarithmic terms.

Proof. From the hypothesis on f, $2k \log 3k \leq 2\log(n)\log(3\log n)$. Hence, for n sufficiently large (in fact as soon as $n \geq 854$), by Theorem 23,

$$\mathrm{s}(f) \geq \frac{1}{\log n}\sqrt{\frac{3\mathrm{bs}(f)}{10\log(3\log n)}}. \tag{1}$$

- Suppose $3\log n \leq \mathrm{bs}^{\frac{1}{4}}(f)$. Then, from Equation (1) we get,

$$\mathrm{s}(f) \geq \frac{3\mathrm{bs}(f)^{1/4}}{\sqrt{\log \mathrm{bs}(f)}}.$$

- Suppose, on the other hand, $3\log n > \mathrm{bs}^{\frac{1}{4}}(f)$.

  We use the following lower bound on sensitivity which was proved in [19]: for all Boolean functions f which depend on all their variables.

$$\mathrm{s}(f) \geq \frac{1}{2}\log n - \frac{1}{2}\log\log n + \frac{1}{2}.$$

  When $n \geq 2$, we have that $\log n > 2\log\log n$, so using the condition on $\mathrm{bs}(f)$ we get,

$$\mathrm{s}(f) \geq \frac{1}{4}\log n \geq \frac{1}{12}\mathrm{bs}(f)^{1/4}.$$

Combining the two cases we get when n is large enough,

$$\mathrm{s}(f) \geq \min\left(\frac{3}{\sqrt{\log \mathrm{bs}(f)}}, \frac{1}{12}\right)\mathrm{bs}(f)^{1/4}.$$

□

4.2 Removing the condition on the bottom fan-in

In this section, we complete the proof of Theorem 17. We note that when the depth is constant but the size of the formula is large enough, there has to be a level at which the fanin is sufficiently large. If one of the last two fanins is large, we can apply an argument quite similar to the one in the previous section. Otherwise, we can switch these two layers while incurring a significant blow-up (but still only as a function of depth and read-multiplicity) in certain circuit parameters, while reducing the depth of the circuit. We continue switching the last two layers until one of their fanins is sufficiently large, which is ensured because the circuit is of constant depth.

4.2.1 Normal form by switching:

For notational convenience, we number the layers of a depth-d circuit as $L_1, \ldots, L_d$ with L_1 being just the root (output) gate and L_d the bottom layer (with inputs feeding into them) of gates. Also, we define the following function over $\mathbb{N}$ for later reference:

$$H(x) := 24 \cdot (3x)^{2x} x^4 \log 3x. \tag{2}$$

As mentioned above, we will transform our formula into an equivalent formula where the fanin in the last or the last but one layer is sufficiently large. Such a representation for Boolean functions will be called a normal form:

Definition 25. *A formula is in $(k; a_1, \ldots, a_d)$-normal form if the following properties hold:*

1. *the formula is alternating and $(a_1, \ldots, a_d)$-regular, i.e., fanin of all gates in L_i is a_i,*
2. *the formula is read-k,*
3. *the bottom layer L_d is composed of $\wedge$-gates,*
4. *at least one of the two following conditions on the fanins of the two bottom layers L_{d-1} and L_d is true:*
   - $a_d \geq 2k \log 3k$,
   - *under each $\vee$-gate in L_{d-1}, i.e., one layer above the bottom layer, there are at least $H(k)$ non-constant $\wedge$-bottom gates.*

As we will switch adjacent layers of the formula, let us start by bounding the increase we get by such a procedure. Let the size and width of a DNF (respectively CNF) be the fanin of its first layer and second layer respectively.

Lemma 26. *If f is a function computed by a read-k regular DNF (respectively CNF) of size (top fanin) a and width b, then it is also computed by a read-$(kb^{(a-1)})$ CNF (respectively DNF) of size b^a and width a.*

Now we will focus on the last two layers we get after some number of switches in the formula. We will recursively define certain functions T_i below. Intuitively, T_1 is the fanin of the bottom layer without any switches and T_{i+2} is the fanin of the layer just above the bottom layer after i switching steps. Note that a depth d circuit becomes a depth $d-i$ circuit after i switches and merges of adjacent layers (after switching) of gates of the same type. Thus T_{i+2} is the fanin of layer L_{d-i-1} in the transformed circuit after i applications of switching and merging.

Formally, the family of functions $T_i : \mathbb{N}^i \to \mathbb{N}$, where i is a positive integer, is defined as

$$\begin{cases} T_0 = 1 \\ T_1(a) = a \\ T_p(a_1, \ldots, a_p) = a_1 \cdot (T_{p-2}(a_3, \ldots, a_p))^{T_{p-1}(a_2, \ldots, a_p)} & \text{if } p \geq 2. \end{cases}$$

In what follows, the function T_i will almost always be evaluated on the fanins of the last i layers of the formula. So, we will sometimes use the shorter notation $T_i(\mathbf{a})$ to designate $T_i(a_{d-i+1}, \ldots, a_d)$.

Observe that most of the non-regular formulas can be converted into a regular one by inserting gates or subtrees of gates that compute identically constant functions. Since we want to avoid this, we will define *purely regular formulas* as regular formulas in which each internal gate computes a non-constant Boolean function.

In the next claim, we compute the parameters of our new formula after several switches.

Claim 27. *Suppose f is computed by a purely $(a_1, \dots, a_d)$-regular read-k formula. Then for all integers $i \in [0, d-2]$, f is computable by an $(a_1, \dots, a_{d-i-2}, u, v)$-regular read-$\left(kuv/(\prod_{j=d-i-1}^{d} a_j)\right)$ formula where*

$$u = T_{i+2}(a_{d-i-1}, \dots, a_d) \quad \textit{and} \quad v = T_{i+1}(a_{d-i}, \dots, a_d)$$

such that under any gate in layer L_{d-i-1}, i.e., one layer above the bottom layer of gates, there are at least a_{d-i-1} non-constant bottom gates.

Proof. Let us prove the claim by induction on the number of switchings $i \in [0, d-2]$. For the sake of the proof we will modify the hypothesis by adding the condition that every gate which is not in the last layer is non-constant.

- If $i = 0$. Then,

  $$u = T_2(a_{d-1}, a_d) = a_{d-1}, \quad v = T_1(a_d) = a_d \quad \text{and} \quad k\frac{uv}{a_{d-1}a_d} = k.$$

  and indeed, f is computed by an $(a_1, \dots, a_d)$-regular read-k formula. Also, since the circuit is purely regular all the gates under the last but one bottom gate are non-constant, so that there are at least a_{d-1} non-constant bottom gates under any last but one bottom gate and every other gate is also non-constant.

- Let us assume that the claim is true for a given i in $[0, d-3]$. We will show it for $i+1$. By Induction Hypothesis, f is computed by an $(a_1, \dots, a_{d-i-2}, u, v)$-regular read-$\left(kuv/(\prod_{j=d-i-1}^{d} a_j)\right)$ formula where

  $$u = T_{i+2}(a_{d-i-1}, \dots, a_d) \quad \text{and} \quad v = T_{i+1}(a_{d-i}, \dots, a_d).$$

  Then, by switching the last two layers, Lemma 26 implies that we get a formula whose last layer is of fanin u and last but one layer of fanin v^u. Now since the last but one layer and the layer above it are composed of the same gates we can merge them to get that f is computed by an $(a_1, \dots, a_{d-i-3}, a_{d-i-2}v^u, u)$-regular read-$\left(\frac{kuv}{\prod_{j=d-i-1}^{d} a_j} v^{u-1}\right)$ formula. We have that,

  $$\begin{aligned} u' &= a_{d-i-2}v^u \\ &= a_{d-i-2} \cdot (T_{i+1}(a_{d-i}, \dots, a_d))^{T_{i+2}(a_{d-i-1}, \dots, a_d)} \\ &= T_{i+3}(a_{d-i-2}, \dots, a_d), \\ v' &= u \\ &= T_{i+2}(a_{d-i-1}, \dots, a_d), \\ \text{and } k' &= \frac{kuv}{\prod_{j=d-i-1}^{d} a_j} v^{u-1} \\ &= k\frac{u'v'}{\prod_{j=d-i-2}^{d} a_j}. \end{aligned}$$

  Since every last but one bottom gate before switching is non-constant, after switching the last two layers (but before merging), under every last but one bottom gate there is at least one non-constant bottom gate. Now when we merge the last but one layer with the layer above that we get at least a_{d-i-2} bottom gates which are non-constant under every last but one bottom gate. In this process we did not modify the function computed by any gate which is not in the last layer so all the gates which were non-constant before switching remain so even after switching. This proves the induction hypothesis and hence the claim.

$\square$

Recall the function $H(x)$ from (2). We inductively define $R_i(k)$ as

$$\begin{cases} R_0(k) = R_1(k) = k \\ R_p(k) = k \prod_{j=1}^{p-1} T_j(H(R_{j-1}(k)), \ldots, H(R_0(k)))^{T_{j+1}(H(R_j(k)),\ldots,H(R_0(k)))-1} & \text{if } p \geq 2. \end{cases}$$

Intuitively, the $R_i(k)$'s bound the read value of the formula after $i-1$ switches of the bottom layers. As the functions R_p will always be used on the parameter k (the read value of the original formula), we will usually denote $R_p(k)$ by the simpler notation R_p.

We are now ready to prove that we can transform a sufficiently large regular formula into a formula in normal form.

Lemma 28. *If f is computed by a purely $(a_1, \ldots, a_d)$-regular read-k formula with size larger than $H(R_d)$ then there exists $i \in [0, d-2]$ such that either f or $\neg f$ can be computed by a formula in $(R_{i+1}; a_1, \ldots, a_{d-i-2}, u, v)$-normal form with*

$$u = T_{i+2}(a_{d-i-1}, \ldots, a_d) \text{ and } v = T_{i+1}(a_{d-i}, \ldots, a_d).$$

Moreover,

- *the index i is such that for any $p \geq d-i$ we have $a_p \leq H(R_{d-p})$, and*
- *under each gate in one layer above the bottom one, i.e., L_{d-i-1}, there are at least a_{d-i-1} non-constant gates, where $a_{d-i-1} \geq H(R_{i+1})$.*

Proof. We need to prove here that there exists i such that after i switching steps, the four conditions of the normal form are satisfied.

Let us first notice that Claim 27 ensures that condition (1) is satisfied after any number of switching steps. Then, by considering, a posteriori, f or $\neg f$, we can always assume that the bottom layer is composed of $\wedge$-gates (which is the third condition).

Consequently, we just need to show that after some number of switches, the fourth condition is satisfied with the read value bounded by R_{i+1}.

First, if a_d is larger than $2k\log(3k)$, then the first point of the fourth condition is already satisfied and the bound $k \leq R_1$ holds. So, let us assume this is not the case. Let us consider the set

$$\mathcal{I} = \{i \in [0, d-2] \mid a_{d-i-1} \geq H(R_{i+1})\}.$$

In particular, as by hypothesis the size $a_1 \ldots a_d$ is larger than $H(R_d)$

$$a_1 \ldots a_d \geq H(R_d) \geq H(R_{d-1}) \cdot \ldots \cdot H(R_0).$$

Hence $\mathcal{I}$ is not empty (since $a_d \leq H(R_0)$) and we can define $i_0 = \min \mathcal{I}$. By minimality of i_0 we have that $a_{d-i_0-1} \geq H(R_{i_0+1})$ and for each $i \leq i_0$, $a_{d-i} < H(R_i)$.

By Claim 27, f is computed by an $(a_1, \ldots, a_{d-i_0-2}, u, v)$-regular read-$k'$ formula with

$$u = T_{i_0+2}(a_{d-i_0-1}, \ldots, a_d),\ v = T_{i_0+1}(a_{d-i_0}, \ldots, a_d),\ \text{and } k' = \frac{kuv}{\prod_{j=d-i_0-1}^{d} a_j}$$

and such that under each last but one bottom gate, there are at least a_{d-i_0-1} non-constant bottom gates. In particular,

$$\begin{aligned} k' = \frac{kuv}{\prod_{j=d-i_0-1}^{d} a_j} &= kT_{i_0}(\mathbf{a})^{T_{i_0+1}(\mathbf{a})} \frac{T_{i_0+1}(\mathbf{a})}{a_d} \prod_{j=1}^{i_0} \frac{1}{a_{d-j}} \\ &= kT_{i_0}(\mathbf{a})^{T_{i_0+1}(\mathbf{a})} \frac{T_{i_0+1}(\mathbf{a})}{T_1(\mathbf{a})} \prod_{j=1}^{i_0} \frac{T_{j-1}(\mathbf{a})^{T_j(\mathbf{a})}}{T_{j+1}(\mathbf{a})} = k \prod_{j=1}^{i_0} T_j(\mathbf{a})^{T_{j+1}(\mathbf{a})-1}. \end{aligned}$$

Since for each $i \leq i_0$, $a_{d-i} < H(R_i)$ and since the functions T_j's are non-decreasing with respect to all their variables,

$$k' \leq k \prod_{j=1}^{i_0} T_j(H(R_{j-1}), \ldots, H(R_0))^{T_{j+1}(H(R_j), \ldots, H(R_0))-1} = R_{i_0+1}.$$

Consequently, after i_0 switching steps, under each last but one bottom gate, there are at least $a_{d-i_0-1} \geq H(R_{i_0+1})$ non-constant bottom gates. This ensures the second point of the fourth condition thus concluding the proof.

$\square$

Now since our new formula's last or last but one fanin is sufficiently large, we can prove a lower bound on the sensitivity as was done in Theorem 23. The sketch of the proof is similar to the one of Theorem 23, but the fact that we now consider the last two layers (instead of the last layer only) makes details a bit more complicated.

Theorem 29. *If f is computed by a purely $(a_1, \ldots, a_d)$-regular read-k formula with size larger than $H(R_d(k))$, then*

$$\mathrm{s}(f) \geq \sqrt{\frac{3\mathrm{bs}(f)}{5R_{d-1}(k)H(R_{d-1}(k))^{(d+1)/2}}} = \Omega_{k,d}(\sqrt{\mathrm{bs}(f)}).$$

Since the function $R_{d-1}(k)$ only depends on d and k, Theorem 17 immediately follows. One can notice that the order of magnitude of the hidden constant in this theorem is approximatively the inverse of the tetration ${}^{2d-2}k = \underbrace{k^{k^{\cdot^{\cdot^{k}}}}}_{2d-2}$.

Proof of Theorem 29. By Lemma 28, we know that f (or $\neg f$) can be computed by a $(k', a_1, \ldots, a_{d'-2}, u, v)$-regular formula in normal form where $k' = R_{d-d'+1}(k)$. Here d' is the depth of the new (equivalent) formula after applying $d - d'$ switches and merges. If the bottom fanin is larger than $2k' \log(3k')$ (the first condition for the fanins in the normal form) then using Theorem 23 we get that,

$$\mathrm{s}(f) \geq \frac{1}{k'}\sqrt{\frac{3\mathrm{bs}(f)}{10\log 3k'}} \geq \sqrt{\frac{3\mathrm{bs}(f)}{5R_{d-1}(k)H(R_{d-1}(k))^{(d+1)/2}}}.$$

Otherwise we have that under each gate in $L_{d'-1}$, there are at least $a_{d'-1} \geq H(R_{d-d'+1})$ non-constant bottom gates.

In this case, we want to give a similar argument as in proof of Lemma 20 for the last but one layer instead of the last layer. Hence, we would like to have $\wedge$-gates at the last but one layer. So we will consider $(\neg f)$ if necessary. By Lemma 28, such a bottom $\wedge$-gate of $\mathtt{C}$ is the parent of at least $a_{d'-1}$ non-constant bottom $\vee$-gates. Let us group these non-constant $\vee$-gates into groups of size $\alpha = \lfloor H(k')/2 \rfloor$. We now get $\mathtt{C}'$ from $\mathtt{C}$ so that each group contains at least one $\vee$-gate which has only positive literals under it. Let g be the function computed by $\mathtt{C}'$.

Using a similar argument as in the proof of Lemma 20 we get the following claim which is proved just after,

Claim 30. *For all x in $g^{-1}(1)$*

$$|x|_1 \geq \left\lceil \frac{A'}{k'} \left\lfloor \frac{a_{d'-1}}{\alpha} \right\rfloor \right\rceil \geq \frac{A'(2a_{d'-1} - H(k') + 1)}{k'H(k')} \quad \textit{with } A' = A(\mathtt{C}, d'-2).$$

Taking the input $x \in g^{-1}(1)$ with least Hamming weight we get that,

$$\begin{aligned}
\mathrm{s}_1(g,x) &\geq |x|_1 \\
&\geq \frac{A'(2a_{d'-1} - H(k') + 1)}{k'H(k')} \\
&\geq \frac{A'a_{d'-1} + 1}{k'H(k')} \\
&\geq \frac{A(\mathtt{C}, d) + 1}{k'H(k')H(R_{d-3})^{(d-d'+1)/2}}
\end{aligned}$$

since for any $p \geq d'+1$ we have that $a_p \leq H(R_{d-p}) \leq H(R_{d-3})$ and we only need to consider alternate layers in the definitions of A and A'.

Since the circuit is in normal form we know that $k' \leq R_{d-d'+1}(k) \leq R_{d-1}(k)$. Using a proof similar to Lemma 20 we get that,

$$s(f) \geq \sqrt{\frac{3\mathrm{bs}(f)}{5R_{d-1}(k)H(R_{d-1}(k))^{(d+1)/2}}}.$$

□

Proof of Claim 30. Let us randomly negate each variable. Each variable is independently chosen as positive or negative with probability $\frac{1}{2}$. Since each variable occurs at most k' times, in every group we get a set S_i of $\frac{\alpha}{k'v}$ $\vee$-gates, no two of which share a variable. Furthermore, each $\vee$-gate in the group is not constant: a variable cannot appear in its positive and negated form under any $\vee$-gate in

S_i. Let A_i be the event that no $\vee$-gate in S_i has only positive literals. Hence since these $\vee-$gates are independent,

$$\Pr[A_i] \leq \left(1 - \frac{1}{2^v}\right)^{\frac{\alpha}{k'v}}.$$

Every event A_i is dependent on at most $k' \frac{\alpha}{k'v} v = \alpha$ other A_j's. Using the Lovász Local Lemma, if

$$e(\alpha + 1)\left(1 - \frac{1}{2^v}\right)^{\frac{\alpha}{k'v}} \leq 1 \tag{3}$$

then $\Pr\left[\bigcap \overline{A_i}\right] > 0$.

Since $v \leq 2k' \log(3k')$, $\alpha = \lfloor 12(3k')^{2k'} k'^4 \log 3k' \rfloor = \lfloor \frac{H(k')}{2} \rfloor$ satisfies Inequality (3).

So there exists a new formula $\mathtt{C}'$ (where $\mathtt{C}'$ is derived from $\mathtt{C}$ by negating some variables) such that every group will have at least one $\vee$-gate with only positive literals. Let g be the function computed by $\mathtt{C}'$.

As in Lemma 20, $g(x) = 1$ implies that the output evaluates to one and there are at least $A(\mathtt{C}', d' - 2) = A(\mathtt{C}, d' - 2) = A'$ last but one bottom $\wedge$-gates which evaluate to 1. Under each group of α $\vee$-gates there is at least one $\vee$-gate which has only positive literals. From every such $\vee$-gate we get one variable which is instantiated to 1 in x. As each variable appears at most k' times in these gates, it proves the claim. $\square$

5 Sensitivity Lower Bounds for DNFs

In this section, we get sensitivity lower bounds for functions computed by read-restricted DNFs. A DNF is said to be *minimal* if no proper sub-formula of such a DNF computes the same function.

Notation: For a DNF $\mathtt{C}$ let a_1 denote its top fanin and $a_{21}, \ldots, a_{2a_1}$ its bottom fanins, with $a_2 = a_{21} \geq a_{22} \geq \ldots \geq a_{2a_1}$.

5.1 Regular read-k DNFs of large width

We can adapt Corollary 24 in the case where the DNF in question is regular and its width is sufficiently large:

Corollary 31. *Let f be a Boolean function computed by a minimal and regular DNF of size n^c, for some $c > 0$ with width larger than or equal to $6 + 3c \log n$. Then,*

$$\mathrm{s}(f) \geq \frac{\mathrm{bs}(f)^{1/3}}{2\sqrt{5 \max(2, c)}}.$$

Proof. Since f is computed by a DNF of size n^c, the DNF is a read-n^c formula. As each variable appears only once under any bottom gate, its parse-read is 1.

By Theorem 23 we get,

$$\mathrm{s}(f) \geq \sqrt{\frac{\mathrm{bs}(f)}{10 \log 4n^c}}. \tag{4}$$

We have the following cases.

- $\log n \leq \mathrm{bs}^{\frac{1}{3}}(f)$

  From Equation (4) we get,

$$\mathrm{s}(f) \geq \sqrt{\frac{\mathrm{bs}(f)}{20 + 10c\mathrm{bs}^{1/3}}}.$$

  If $20 > 10c\mathrm{bs}^{1/3}$, then $\mathrm{s}(f) \geq \frac{\mathrm{bs}^{1/2}}{\sqrt{40}} \geq \frac{\mathrm{bs}^{1/3}}{2\sqrt{10}}$. Otherwise, $\mathrm{s}(f) \geq \frac{\mathrm{bs}^{1/3}}{2\sqrt{5c}}$.

- $\log n > \mathrm{bs}^{\frac{1}{3}}(f)$

  As in the proof of Corollary 24, we have

$$\mathrm{s}(f) \geq \frac{1}{4}\log n.$$

  So

$$\mathrm{s}(f) \geq \frac{1}{4}\mathrm{bs}(f)^{1/3} \geq \frac{1}{2\sqrt{10}}\mathrm{bs}(f)^{1/3}.$$

Combining the two cases we get,

$$\mathrm{s}(f) \geq \frac{\mathrm{bs}(f)^{1/3}}{2\sqrt{5c'}},$$

where $c' = \max(c, 2)$. □

5.2 Read-k DNFs of small size

In this section we will remove the constraints of regularity and large width for DNFs, thus proving the sensitivity conjecture for all functions computed by read-k DNFs.

The first lemma ensures a lower bound on $\mathrm{s}_0(f)$ for functions computed by read-k DNFs.

Lemma 32. *Let f be a Boolean formula computed by a minimal read-k DNF* $\mathtt{C}$. *Then*

$$\mathrm{s}_0(f) \geq \frac{a_1}{ka_2}.$$

Proof. Consider the set B of bottom $\wedge-$gates. We can get a set $I \subseteq B, |I| = \frac{a_1}{ka_2}$, of $\wedge-$gates such that no two of these gates share a common variable. This can be done using a trivial algorithm where we select a gate and remove all the other gates that share a variable with this gate, that is at most ka_2 of them, from consideration. Since no two gates in I share a variable, we negate variables to get an equivalent circuit $\mathtt{C}'$ so that all the variables under gates in I appear positively. Consequently, on any input $x \in g^{-1}(0)$, each of the gates in I gives us one variable which is instantiated to 0. We have that $\forall x \in g^{-1}(0)$, $|x|_0 \geq \frac{a_1}{ka_2}$. Taking the input $x \in g^{-1}(0)$ with largest Hamming weight, we get, $\mathrm{s}_0(f, x) \geq \frac{a_1}{ka_2}$. □

The second lemma states that the sensitivity of a read-k DNF is lower bounded by a function of its maximum bottom fanin.

Lemma 33. *Let f be a Boolean function computed by a minimal read-k DNF* $\texttt{C}$. *Then,*

$$\mathrm{s}_1(f) + (1+k)\mathrm{s}_0(f) \geq a_2.$$

Proof. Let the bottom $\wedge-$gates be $W_1, \ldots, W_{a_1}$ with fanins $a_2 = a_{21} \geq \ldots \geq a_{2a_1}$ respectively. Let the variables under W_i be $x_{i1}, \ldots, x_{ia_{2i}}$.

Let us define two sets:

- $z \in P_1$ if and only if $W_1(z) = 1$ and for all $j > 1$, $W_j(z) = 0$,
- $y \in P_2$ if and only if $W_1(y) = 1$ and y is sensitive on the variable x_{11}.

By minimality of $\texttt{C}$, we can find an input

- z_0 in P_1, otherwise removing the gate W_1 would not modify the function,
- y_0 in P_2, otherwise we can remove the leaf corresponding to x_{11} from W_1.

In fact it would be great to find an input which belongs to both P_1 and P_2, but unfortunately, it is not always possible. However, we show we can find such a pair (z, y) such that the Hamming distance between them is small.

Let us fix $z_1 = z_0$. We show that the closest input to z_1 which belongs to P_2 is close enough. We know that y_0 is in P_2, so P_2 is not empty. Let y_1 be an input in P_2 which minimizes the Hamming distance $\delta = d_H(z_1, y_1)$.

We will prove that $\delta \leq \mathrm{s}_0 - 1$. Let us consider $y' = y_1^{\{x_{11}\}}$, i.e. the input we get by flipping the variable x_{11} in y_1. We know that $f(y') = 0$ and that y' is sensitive on the variable x_{11} which appears under the gate W_1. Let v be one of the δ variables which is instantiated differently on z_1 and on y_1. As, $W_1(z_1) = W_1(y_1) = 1$, the variable v does not appear under the gate W_1. Moreover, if $f(y'^{\{v\}}) = 0$, it would imply that $y_1^{\{v\}}$ is in P_2 contradicting the minimality of δ. So y' is sensitive on the variable v. Thus, y' is sensitive on at least $\delta + 1$ variables, hence, there exists a pair of inputs $(z_1, y_1) \in P_1 \times P_2$ such that the Hamming distance between z_1 and y_1 is at most $\mathrm{s}_0(f) - 1$.

Let $J \subseteq [2, a_2]$ be the variables which appear under W_1 and which are sensitive on z_1, so $\mathrm{s}_1(f) \geq |J|$. Let $\bar{J} = [2, a_2] \setminus J$.

Let us define for all variables v in $\bar{J}$,

$$\mathcal{G}_v = \left\{ u \mid W_u \left(z_1^{\{v\}} \right) = 1 \right\}.$$

As the variables from $\bar{J}$ are not sensitive on z_1, all these sets $\mathcal{G}_v$ are non empty. Moreover for any gate W_u in $\mathcal{G}_v$, the variable v appears with a different sign in W_1 and W_u but for any other variable v' in W_1, either it appears with the same sign in W_u, or it does not appear. So we can first notice that if $v_1 \neq v_2 \in \bar{J}$, then $\mathcal{G}_{v_1} \cap \mathcal{G}_{v_2} = \emptyset$. Now we will remove from our consideration any variable v from $\bar{J}$ such that $\mathcal{G}_v$ contains a gate W_u which either depends on the variable x_{11} or depends on a variable which distinguishes z_1 from y_1. As the formula is read-k, there are at most $k\mathrm{s}_0(f) - 1$ such gates (W_1 contains x_{11} but is not such a gate), and as the $\mathcal{G}_v$'s are disjoint, they are at most $k\mathrm{s}_0(f) - 1$ such variables v. So we can extract a subset $J_0 \subseteq \bar{J}$ such that $|J_0| \geq |\bar{J}| - k\mathrm{s}_0(f) + 1$ and such that for any variable v in J_0 and every u in $\mathcal{G}_v$, the gate W_u does not depend on the variable x_{11} and on the variables which distinguish z_1 from y_1.

Let us show that all the variables from J_0 are sensitive on $y_1^{\{x_{11}\}}$. By the definition of y_1, we have $f\left(y_1^{\{x_{11}\}}\right) = 0$. For any variable v in J_0, let us choose a gate W_u corresponding to u in $\mathcal{G}_v$. As W_u does not depend on x_{11} and on the variables which distinguish z_1 from y_1, we have:

$$W_u\left(y_1^{\{x_{11},v\}}\right) = W_u\left(y_1^{\{v\}}\right) = W_u\left(z_1^{\{v\}}\right) = 1.$$

The last equality comes from the definition of $\mathcal{G}_v$. So, $f\left(y_1^{\{x_{11},v\}}\right) = 1$ as required.

Hence, $\mathrm{s}_0 \geq |\bar{J}| - k\mathrm{s}_0 + 1$, which proves the lemma.

□

Theorem 34. *Let f be a Boolean formula computed by a read-k DNF. Then*

$$(k+2)\mathrm{s}(f) \geq n^{1/3}.$$

In particular, if $k \leq n^{\frac{1}{3}-\varepsilon} - 2$, we get

$$\mathrm{s}(f) \geq n^{\varepsilon} \geq \mathrm{bs}(f)^{\varepsilon}.$$

Proof. Using Lemma 33 we get that, $(k+2)\mathrm{s}(f) \geq \mathrm{s}_1(f) + (1+k)\mathrm{s}_0(f) \geq a_2$. By Lemma 32 we know that,

$$\mathrm{s}_0(f) \geq \frac{a_1}{ka_2} \geq \frac{n}{ka_2^2}.$$

Combining these two inequalities we get,

$$\mathrm{s}^3(f) \geq \left(\frac{a_2}{k+2}\right)^2 \frac{n}{ka_2^2} \geq \frac{n}{k(k+2)^2}$$

and so $(k+2)\mathrm{s}(f) \geq n^{1/3}$. In particular, when $k+2 \leq n^{\frac{1}{3}-\varepsilon}$, we get $\mathrm{s}(f) \geq n^{\varepsilon} \geq \mathrm{bs}(f)^{\varepsilon}$.

□

References

[1] Scott Aaronson. The "sensitivity" of 2-colorings of the d-dimensional integer lattice. *blogpost*, July 2010. `http://mathoverflow.net/questions/31482/`.

[2] Noga Alon and Joel H Spencer. *The Probabilistic Method*. John Wiley and Sons, 1994.

[3] Andris Ambainis, Mohammad Bavarian, Yihan Gao, Jieming Mao, Xiaoming Sun, and Song Zuo. Tighter relations between sensitivity and other complexity measures. In *International Colloquium on Automata, Languages and Programming (ICALP) 2014, Part I*, pages 101–113, 2014.

[4] Andris Ambainis and Krisjanis Prusis. A tight lower bound on certificate complexity in terms of block sensitivity and sensitivity. In *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*, pages 33–44, 2014. URL: `http://dx.doi.org/10.1007/978-3-662-44465-8_4`, `doi:10.1007/978-3-662-44465-8_4`.

[5] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 352–361, 1998. URL: `http://dx.doi.org/10.1109/SFCS.1998.743485`, `doi:10.1109/SFCS.1998.743485`.

[6] Meena Boppana. Lattice variant of the sensitivity conjecture. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:89, 2012.

[7] Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theor. Comput. Sci.*, 288(1):21–43, 2002. URL: `http://dx.doi.org/10.1016/S0304-3975(01)00144-X`, `doi:10.1016/S0304-3975(01)00144-X`.

[8] Sourav Chakraborty. On the sensitivity of cyclically-invariant boolean functions. In *Conference on Computational Complexity (CCC)*, pages 163–167, 2005.

[9] Stephen Cook, Cynthia Dwork, and Rüdiger Reischuk. Upper and lower time bounds for parallel random access machines without simultaneous writes. *SIAM J. Comput.*, 15(1):87–97, 1986.

[10] Justin Gilmer, Michal Koucký, and Michael E. Saks. A new approach to the sensitivity conjecture. In *Innovations in Theoretical Computer Science, ITCS*, pages 247–254, 2015.

[11] Parikshit Gopalan, Rocco A. Servedio, Avishay Tal, and Avi Wigderson. Degree and sensitivity: tails of two distributions. *to appear CCC 2016*, 2016.

[12] Craig Gotsman and Nathan Linial. The equivalence of two problems on the cube. *Journal of Combinatorial Theory, Series A*, 61(1):142–146, 1992.

[13] Pooya Hatami, Raghav Kulkarni, and Denis Pankratov. Variations on the sensitivity conjecture. *Theory of Computing, Graduate Surveys*, 4:1–27, 2011.

[14] Claire Kenyon and Samuel Kutin. Sensitivity, block sensitivity, and l-block sensitivity of Boolean functions. *Information and Computation*, 189(1):43–53, 2004.

[15] Hiroki Morizumi. Sensitivity, block sensitivity, and certificate complexity of unate functions and read-once functions. *IFIP TCS*, 2014.

[16] Noam Nisan. CREW PRAMs and decision trees. In *Symposium on Theory of Computing*, pages 327–335, 1989.

[17] Noam Nisan and Mario Szegedy. On the degree of Boolean functions as real polynomial. In *Symposium on Theory of Computing*, pages 462–467, 1992.

[18] David Rubinstein. Sensitivity vs. block sensitivity of boolean functions. *Combinatorica*, 15(2):297–299, 1995.

[19] Hans-Ulrich Simon. A tight $\Omega(\log \log n)$-bound on the time for parallel RAM's to compute nondegenerated Boolean functions. *Foundations of Computation Theory, Lecture Notes in Computer Science*, 158(1):439–444, 1983.

[20] Xiaoming Sun. Block sensitivity of weakly symmetric functions. *Theor. Comput. Sci.*, 384(1):87–91, 2007.

[21] Avishay Tal. On the sensitivity conjecture. In *International Colloquium on Automata, Languages and Programming (ICALP)*, 2016.

[22] György Turán. The critical complexity of graph properties. *Inf. Process. Lett.*, 18(3):151–153, 1984.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de