Electronic Colloquium on Computational Complexity, Report No. 27 (2014)

A Tight Lower Bound on Certificate Complexity in Terms of Block Sensitivity and Sensitivity

Krišjānis Prūsis and Andris Ambainis*

Faculty of Computing, University of Latvia, Raina bulv. 19, Rga, LV-1586, Latvia

Abstract. Sensitivity, certificate complexity and block sensitivity are widely used Boolean function complexity measures. A longstanding open problem, proposed by Nisan and Szegedy [6], is whether sensitivity and block sensitivity are polynomially related. Motivated by the constructions of functions which achieve the largest known separations, we study the relation between 1-certificate complexity and 0-sensitivity and 0-block sensitivity.
Previously the best known lower bound was $C_1(f) \geq \frac{bs_0(f)}{2s_0(f)}$, achieved by Kenyon and Kutin [5]. We improve this to $C_1(f) \geq \frac{3bs_0(f)}{2s_0(f)}$. While this improvement is only by a constant factor, this is quite important, as it precludes achieving a superquadratic separation between $bs(f)$ and $s(f)$ by iterating functions which reach this bound. In addition, this bound is tight, as it matches the construction of Ambainis and Sun [3] up to an additive constant.

1 Introduction

Determining the biggest possible gap between the sensitivity $s(f)$ and block sensitivity $bs(f)$ of a Boolean function is a well-known open problem in the complexity of Boolean functions. Even though this question has been known for over 20 years, there has been quite little progress on it.

The biggest known gap is $bs(f) = \Omega(s^2(f))$. This was first discovered by Rubinstein [7], who constructed a function f with $bs(f) = \frac{s^2(f)}{2}$, and then improved by Virza [8] and Ambainis and Sun [3]. Currently, the best result is a function f with $bs(f) = \frac{2}{3}s^2(f) - \frac{1}{3}s(f)$ [3]. The best known upper bound is exponential: $bs(f) \leq s(f)2^{s(f)-1}$ [2] which improves over an earlier exponential upper bound by Kenyon and Kutin [5].

In this paper, we study a question motivated by the constructions of functions that achieve a separation between $s(f)$ and $bs(f)$. The question is as follows: Let $s_z(f)$, $bs_z(f)$ and $C_z(f)$ be the maximum sensitivity, block sensitivity and

* Supported by FP7 projects QALGO (Grant Agreement No. 600700) and RAQUEL (Grant Agreement No. 255961) and ERC Advanced Grant MQC. Part of this work was done while Andris Ambainis was visiting Institute for Advanced Study, Princeton, supported by National Science Foundation under agreement No. DMS-1128155. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

ISSN 1433-8092

certificate complexity achieved by f on inputs x: $f(x) = z$. What is the best lower bound of $C_1(f)$ in terms of $s_0(f)$ and $bs_0(f)$?

The motivation for this question is as follows. Assume that we fix $s_0(f)$ to a relatively small value m and fix $bs_0(f)$ to a substantially larger value k. We then minimize $C_1(f)$. We know that $s_1(f) \leq C_1(f)$ (because every sensitive bit has to be contained in a certificate). We have now constructed an example where both $s_0(f)$ and $s_1(f)$ are relatively small and $bs_0(f)$ large. This may already achieve a separation between $bs_0(f)$ and $s(f) = \max(s_0(f), s_1(f))$ and, if $s_1(f) > s_0(f)$, we can improve this separation by composing the function with OR (as described in [3]).

While this is just one way of achieving a gap between $s(f)$ and $bs(f)$, all the best separations between these two quantities can be cast into this framework. Therefore, we think that it is interesting to explore the limits of this approach.

The previous results are as follows:

1. Rubinstein's construction [7] can be viewed as taking a function f with $s_0(f) = 1$, $bs_0(f) = k$ and $C_1(f) = 2k$. A composition with OR yields [3] $bs(f) = \frac{1}{2}s^2(f)$;
2. Later work by Virza [8] and Ambainis and Sun [3] improve this construction by constructing f with $s_0(f) = 1$, $bs_0(f) = k$ and $C_1(f) = \left\lfloor \frac{3k}{2} \right\rfloor + 1$. A composition with OR yields $bs(f) = \frac{2}{3}s^2(f) - \frac{1}{3}s(f)$;
3. Ambainis and Sun [3] also show that, given $s_0(f) = 1$ and $bs_0(f) = k$, the certificate complexity $C_1(f) = \left\lfloor \frac{3k}{2} \right\rfloor + 1$ is the smallest that can be achieved. This means that a better bound must either start with f with $s_0(f) > 1$ or use some other approach;
4. For $s_0(f) = m$ and $bs_0(f) = k$, it is easy to modify the construction of Ambainis and Sun [3] to obtain $C_1(f) = \left\lfloor \frac{3\lceil k/m \rceil}{2} \right\rfloor + 1$ but this does not result in a better separation between $bs(f)$ and $s(f)$;
5. Kenyon and Kutin [5] have shown a lower bound of $C_1(f) \geq \frac{k}{2m}$. If this was achievable, this could result in a separation of $bs(f) = 2s^2(f)$.

The gap between the construction $C_1(f) = \frac{3k}{2m} + O(1)$ and the lower bound of $C_1(f) \geq \frac{k}{2m}$ is only a constant factor but the constant here is quite important. This gap corresponds to a difference between $bs(f) = (\frac{2}{3}+o(1))s^2(f)$ and $bs(f) = 2s^2(f)$, and, if we achieved $bs(f) > s^2(f)$, iterating the function f would yield an infinite sequence of functions with a superquadratic separation $bs(f) = s(f)^c$, where $c > 2$.

In this paper, we show that

$$C_1(f) \geq \frac{3}{2}\frac{bs_0(f)}{s_0(f)} - \frac{1}{2}$$

for any f. This matches the best construction up to an additive constant and shows that no further improvement can be achieved along the lines of [7, 8, 3]. Our bound is shown by an intricate analysis of possible certificate structures for f.

Since we now know that $bs_0(f) \leq \left(\frac{2}{3} + o(1)\right) C_1(f)s_0(f)$, it is tempting to conjecture that $bs_0(f) \leq \left(\frac{2}{3} + o(1)\right) s_1(f)s_0(f)$. If this was true, the existing separation between $bs(f)$ and $s(f)$ would be tight.

2 Preliminaries

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function on n variables. The i-th variable of input x is denoted by x_i. For an index set $S \subseteq [n]$, let x^S be the input obtained from input x by flipping every bit whose position is in S. Let a *z-input* be an input on which the function takes the value z, where $z \in \{0,1\}$.

We now briefly define the notions of sensitivity, block sensitivity and certificate complexity. For more information on them and their relations to other complexity measures (such as deterministic, probabilistic and quantum decision tree complexities), we refer the reader to the survey by Buhrman and de Wolf [4].

Definition 1. *The* sensitivity complexity $s(f,x)$ *of* f *on an input* x *is defined as* $|\{i \mid f(x) \neq f(x^{\{i\}})\}|$. *The* z-sensitivity $s_z(f)$ *of* f, *where* $z \in \{0,1\}$, *is defined as* $\max\{s(f,x) \mid x \in \{0,1\}^n, f(x) = z\}$. *The* sensitivity $s(f)$ *of* f *is defined as* $\max\{s_0(f), s_1(f)\}$.

Definition 2. *The* block sensitivity $bs(f,x)$ *of* f *on input* x *is defined as the maximum number* b *such that there are* b *pairwise disjoint subsets* $B_1, \ldots, B_b$ *of* $[n]$ *for which* $f(x) \neq f(x^{B_i})$. *We call each* B_i *a* block. *The* z-block sensitivity $bs_z(f)$ *of* f, *where* $z \in \{0,1\}$, *is defined as* $\max\{bs(f,x) \mid x \in \{0,1\}^n, f(x) = z\}$. *The* block sensitivity $bs(f)$ *of* f *is defined as* $\max\{bs_0(f), bs_1(f)\}$.

Definition 3. *A* certificate c *of* f *on input* x *is defined as a partial assignment* $c : S \to \{0,1\}, S \subseteq [n]$ *of* x *such that* f *is constant on this restriction. If* f *is always 0 on this restriction, the certificate is a* 0-certificate. *If* f *is always 1, the certificate is a* 1-certificate.

We denote specific certificates as words with $*$ in the positions that the certificate does not assign. For example, $01****$ denotes a certificate that assigns 0 to the first variable and 1 to the second variable.

We say that an input x *satisfies* a certificate c if it matches the certificate in every assigned bit.

The number of *contradictions* between an input and a certificate or between two certificates is the number of positions where one of them assigns 1 and the other assigns 0. For example, there are two contradictions between $0010**$ and $100***$ (in the 1st position and the 3rd position).

The number of *overlaps* between two certificates is the number of positions where both have assigned the same values. For example, there is one overlap between $001***$ and $*0000$ (in the second position). We say that two certificates *overlap* if there is at least one overlap between them.

We say that a certificate remains *valid* after fixing some input bits if none of the fixed bits contradicts the certificate's assignments.

Definition 4. *The* certificate complexity $C(f,x)$ *of* f *on input* x *is defined as the minimum length of a certificate that* x *satisfies. The* z-certificate complexity $C_z(f)$ *of* f, *where* $z \in \{0,1\}$, *is defined as* $\max\{C(f,x) \mid x \in \{0,1\}^n, f(x) = z\}$. *The* certificate complexity $C(f)$ *of* f *is defined as* $\max\{C_0(f), C_1(f)\}$.

3 Background

We study the following question:

Question: Assume that $s_0(g) = m$ and $bs_0(g) = k$. How small can we make $C_1(g)$?

Example 1. Ambainis and Sun [3] consider the following function construction. $g_0(x_1, \ldots, x_{2k}) = 1$ if and only if $(x_1, \ldots, x_{2k})$ satisfies one of k certificates $c_0, \ldots, c_{k-1}$ with the certificate c_i ($i \in \{0, 1, \ldots, k-1\}$) requiring that

(a) $x_{2i+1} = x_{2i+2} = 1$;
(b) $x_{2j+1} = 0$ for $j \in \{0, \ldots, k-1\}$, $j \neq i$;
(c) $x_{2j+2} = 0$ for $j \in \{i+1, \ldots, i + \lfloor k/2 \rfloor\}$ (with $i+1, \ldots, i+\lfloor k/2 \rfloor$ taken mod$k$).

Then, we have:

- $s_0(g_0) = 1$ (it can be shown that, for every 0-input of g_0, there is at most one c_i in which only one variable does not have the right value);
- $s_1(g_0) = C_1(g_0) = \lfloor 3k/2 \rfloor + 1$ (a 1-input that satisfies a certificate c_i is sensitive to changing any of the variables in c_i and c_i contains $\lfloor 3k/2 \rfloor + 1$ variables);
- $bs_0(g_0) = k$ (the 0-input $x_1 = \cdots = x_{2k} = 0$ is sensitive to changing any of the pairs (x_{2i+1}, x_{2i+2}) from $(0,0)$ to $(1,1)$).

This function can be composed with the OR-function to obtain the best known separation between $s(f)$ and $bs(f)$: $bs(f) = \frac{2}{3}s^2(f) - \frac{1}{3}s(f)$[3]. As long as $s_0(g) = 1$, the construction is essentially optimal: any g with $bs_0(g) = k$ must satisfy $C_1(g) \geq s_1(g) \geq \frac{3k}{2} - O(1)$.

In this paper, we explore the case when $s_0(g) > 1$. An easy modification of the construction from [3] gives

Theorem 1. *There exists a function g for which $s_0(g) = m$, $bs_0(g) = k$ and $C_1(f) = \left\lfloor \frac{3\lceil k/m \rceil}{2} \right\rfloor + 1$.*

Proof. To simplify the notation, we assume that k is divisible by m. Let $r = k/m$.

We consider a function $g(x_{m1}, \ldots, x_{m,2r})$ with variables $x_{i,j}$ ($i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, 2r\}$) defined by

$$g(x_{11}, \ldots, x_{m,2r}) = \vee_{i=1}^{m} g_0(x_{i,1}, \ldots, x_{i,2r}). \tag{1}$$

Equivalently, $g(x_{11}, \ldots, x_{m,2r}) = 1$ if and only if at least one of the blocks $(x_{i,1}, \ldots, x_{i,2r})$ satisfies one of the certificates $c_{i,0}, \ldots, c_{i,r-1}$ that are defined similarly to $c_0, \ldots, c_{k-1}$ in the definition of g_0.

It is easy to see [3] that the composition of a function g_0 with OR gives $s_0(g) = m\, s_0(g_0)$, $bs_0(g) = m\, bs_0(g_0)$ and $C_1(g) = C_1(g_0)$. This implies the theorem. ∎

While this function does not give a better separation between $s(f)$ and $bs(f)$, any improvement to Theorem 1 could give a better separation between $s(f)$ and $bs(f)$ by using the same composition with OR as in [3].

On the other hand, Kenyon and Kutin [5] have shown that

Theorem 2. *For any f with $s_0(g) = m$ and $bs_0(g) = k$, we have $C_1(f) \geq \frac{k}{2m}$.*

4 Separation between $C_1(f)$ and $bs_0(f)$

In this paper, we show that the example of Theorem 1 is optimal.

Theorem 3. *For any Boolean function f the following inequality holds:*

$$C_1(f) \geq \frac{3}{2}\frac{bs_0(f)}{s_0(f)} - \frac{1}{2}. \tag{2}$$

Without the loss of generality, we can assume that the maximum bs_0 is achieved on the all-0 input denoted by 0. Let $B_1, ..., B_k$ be the sensitive blocks, where $k = bs_0(f)$. Also, we can w.l.o.g. assume that these blocks are minimal and that every bit belongs to a block. (Otherwise, we can fix the remaining bits to 0. This can only decrease s_0 and C_1, strengthening the result.)

Each block B_i has a corresponding minimal 1-certificate c_i such that the word $(\{0\}^n)^{B_i}$ satisfies this certificate. Each of these certificates has a 1 in every position of the corresponding block (otherwise the block would not be minimal) and any number of 0's in other blocks.

We construct a weighted graph G whose vertices correspond to certificates $c_1, \ldots, c_k$, with edges between every two vertices. Each edge has a weight that is equal to the number of contradictions between the two certificates the edge connects. *The weight of a graph* is just the sum of the weights of its edges. We will prove

Lemma 1. *Let w be the total weight of any induced subgraph of G of size m. Then, we have*

$$w \geq \frac{3}{2}\frac{m^2}{s_0(f)} - \frac{3}{2}m. \tag{3}$$

Proof. The proof is by induction. As a basis we take induced subgraphs of size $m \leq s_0(f)$. In this case,

$$\frac{3}{2}\frac{m^2}{s_0(f)} - \frac{3}{2}m \leq 0 \tag{4}$$

and $w \geq 0$ is always true, as the number of contradictions between two certificates cannot be negative.

Let $m > s_0(f)$. We assume that the relation holds for every induced subgraph size $< m$. Let G' be an induced subgraph of size m. Let $H \subset G'$ be its induced subgraph of size $s_0(f)$ with the smallest total weight.

Lemma 2. *For any certificate in G' not belonging to this subgraph $c_i \in G' \setminus H$ the weight of the edges connecting c_i to H is ≥ 3.*

Proof. Let t be the total weight of the edges in H. Let us assume that there exists a certificate $c_j \notin H$ such that the weight of the edges connecting c_j to H is ≤ 2. Let H' be the induced subgraph $H \cup \{c_j\}$. Then the weight of H' must be $\leq t + 2$.

We define the weight of a certificate $c_i \in H'$ in H' as the sum of weights of all edges of H' that involve vertex c_i. If there exists a certificate $c_i \in H'$ such

that its weight in H' is ≥ 3, then the weight of $H \setminus \{c_i\}$ would be $< t$, which is a contradiction, as H was taken to be the induced subgraph of such size with the smallest weight. Therefore the weight of every certificate in H' is at most 2.

In the next section, we show

Lemma 3. *Let f be a Boolean function for which the following properties hold: $f(\{0\}^n) = 0$ and f has such k minimal 1-certificates that each has at most 2 contradictions with the others. Furthermore, for each input position, exactly one of these certificates assigns the value 1. Then, $s_0(f) \geq k$.*

Therefore, the 0-sensitivity of the function is the size of H' which is not possible because $|H'| = s_0(f) + 1$. ■

We now examine the graph $G' \setminus H$. It consists of $m - s_0(f)$ certificates and by the inductive assumption has a weight of at least

$$\frac{3}{2}\frac{(m - s_0(f))^2}{s_0(f)} - \frac{3}{2}(m - s_0(f)). \tag{5}$$

But there are at least $3(m - s_0(f))$ contradictions between H and $G' \setminus H$, thus the total weight of G' is at least

$$\frac{3}{2}\frac{(m - s_0(f))^2}{s_0(f)} - \frac{3}{2}(m - s_0(f)) + 3(m - s_0(f)) = \tag{6}$$

$$\frac{3}{2}\frac{(m^2 - ms_0(f) + s_0(f)^2}{s_0(f)} + \frac{3}{2}m - \frac{3}{2}s_0(f) = \tag{7}$$

$$\frac{3}{2}\frac{m^2}{s_0(f)} - \frac{3}{2}m + \frac{3}{2}s_0(f) + \frac{3}{2}m - \frac{3}{2}s_0(f) = \tag{8}$$

$$\frac{3}{2}\frac{m^2}{s_0(f)} - \frac{3}{2}m. \tag{9}$$

This completes the induction step. ■

By taking the whole of G as G', we find a lower bound on the total number of contradictions in the graph:

$$\frac{3}{2}\frac{k^2}{s_0(f)} - \frac{3}{2}k. \tag{10}$$

But each contradiction requires one 0 in one of the certificates and each 0 contributes to exactly one contradiction, therefore, by the pigeonhole principle, there exists a certificate with at least

$$\frac{3}{2}\frac{k}{s_0(f)} - \frac{3}{2} \tag{11}$$

zeroes. As each certificate contains at least one 1, we get a lower bound on the size of one of these certificates and $C_1(f)$:

$$C_1(f) \geq \frac{3}{2}\frac{bs_0(f)}{s_0(f)} - \frac{1}{2}. \tag{12}$$

■

5 Functions with $s_0(f)$ Equal to Number of 1-certificates

In this section we prove Lemma 3.

5.1 General Case: Functions with Overlaps

Let $c_1, \ldots, c_k$ be the k certificates. We start by reducing the general case of Lemma 3 to the case when there are no overlaps between any of $c_1, \ldots, c_k$.

W.l.o.g., we assume that every input bit belongs to one of minimal blocks $B_1, \ldots, B_k$ that correspond to these certificates. (Otherwise, we could fix the bits not belonging to those blocks to 0. The conditions of the lemma would still be satisfied. First, since $c_1, \ldots, c_k$ can only assign the value 1 to the positions in the corresponding blocks, they all remain valid certificates. If some of them are no longer minimal, we can minimize them by removing variables and this can only decrease the number of contradictions between them. Second, since every B_i is minimal, for every unfixed position exactly one of $c_1, \ldots, c_k$ assigns the value 1. Third, the remaining function is still 0 on the all-0 input.)

Note that certificate overlaps can only occur when two certificates assign 0 to the same position. Then a third certificate assigns 1 to that position. This produces 2 contradictions for the third certificate, therefore it has no further overlaps or contradictions. For example, here we have this situation in the 3rd position (with the first three certificates) and in the 6th position (with the last three certificates):

$$\begin{pmatrix} 1\,1\,0 * * * * * * \\ * * 1 * * * * * * \\ * * 0\,1\,1\,0 * * * * \\ * * * * * 1\,1\,1 * * \\ 0 * * * * 0 * * 1\,1 \end{pmatrix}. \tag{13}$$

Let t be the total number of such overlaps. Let D be the set of certificates assigning 1 to positions with overlaps, $|D| = t$. We fix every position where overlaps occur to 0. Since the remaining function contains the word $\{0\}^n$, it is not identically 1. Every certificate not in D is still a valid 1-certificate, as they assigned either nothing or 0 to the fixed positions. If they are no longer minimal, we can minimize them, which cannot produce any new overlaps or contradictions.

The certificates in D are, however, no longer valid. Let us examine one such certificate $c \in D$. We denote the set of positions assigned to by c by S. The certificate c assigns only the value 1 to $|S|$ positions, one of which is now fixed to 0, say i. If $|S| = 1$, then the remaining function is always sensitive to i on 0-inputs, as flipping it satisfies c.

If $|S| > 1$, we examine the $2^{|S|-1}$ subfunctions obtainable by fixing the remaining positions of S. We fix these positions to the subfunction that is not identically 1 with the highest number of bits fixed to 1. If that subfunction fixes 1 in every position, it is sensitive to i on 0-inputs, as flipping it produces a word which satisfies c. Otherwise it is sensitive on 0-inputs to every other bit fixed to 0 in S besides i, as flipping them would produce a word from a subfunction with

a higher amount of bits fixed to 1. But that subfunction is identically 1 or we would have fixed it instead.

In either case we obtain at least one sensitive bit in S on 0-inputs in the remaining function. Furthermore, every certificate not in D is still valid, if not minimal. But we can safely minimize them again.

We can repeat this procedure for every certificate in D.The resulting function is not always 1 and, on every 0-input, it has at least t sensitive bits among the bits that we fixed. Furthermore, we still have $k-t$ non-overlapping valid minimal 1-certificates with no more than 2 contradictions each. In the next section, we show that this implies that it has 0-sensitivity of at least $k - t$ (Lemma 4). Therefore, the original function has a 0-sensitivity of at least $k - t + t = k$.

5.2 Functions with No Overlaps

Lemma 4. *Let f be a Boolean function f, for which the following properties hold: f is not always 1 and f has such k non-overlapping minimal 1-certificates that each has at most 2 contradictions with the others. Then, $s_0(f) \geq k$.*

Proof. To prove this lemma, we consider the weighted graph G on these k certificates where the weight of an edge in this graph is the number of contradictions between the two certificates the edge connects.

We examine the connected components in this graph, not counting edges with weight 0. There can be only 4 kinds of components – individual certificates, two certificates with 2 contradictions between them, paths of 2 or more certificates with 1 contradiction between every two subsequent certificates in the path and cycles of 3 or more certificates with 1 contradiction between every two subsequent certificates in the cycle. As there are no overlaps between the examined certificates, each position is assigned to by certificates from at most one component.

We will now prove by induction on k that we can obtain a 0-input with as many sensitive bits in the positions of each component as there are certificates in it.

As a basis we take $k = 0$. Since f is not always 1, $s_0(f)$ is defined, but obviously $s_0(f) \geq 0$.

Then we look at each graph component type separately.

Individual Certificates. We first examine individual certificates. Let us denote the examined certificate by c and the set of positions it assigns by S. We fix all bits of S except for one according to c and we fix the remaining bit of S opposite to c. The remaining function cannot be always 1, as otherwise the last bit in S would not be necessary in c, but c is minimal. Therefore on 0-inputs the remaining function is also sensitive to this last bit, as flipping it produces a word which satisfies c.

Afterward the remaining certificates might no longer be minimal. In this case we can minimize them. This cannot produce any more contradictions and no certificate can completely disappear, as the function is not always 1. Therefore the remaining function still satisfies the conditions of this lemma and contains

$k-1$ minimal 1-certificates, with each certificate having at most 2 contradictions with the others.

Then by induction the remaining function has a 0-sensitivity of $k-1$. Together with the sensitive bit among the fixed ones, we obtain $s_0(f) \geq k$.

Certificate Paths. We can similarly reduce certificate paths. A certificate path is a structure where each certificate has 1 contradiction with the next one and there are no other contradictions. For example, here is an example of a path of length 3:

$$\begin{pmatrix} & & i & & & & \\ 1 & 1 & 0 & * & * & * & * \\ * & * & 1 & 1 & 0 & * & * \\ * & * & * & * & 1 & 1 & 1 \end{pmatrix}. \tag{14}$$

We note that every certificate in a path assigns at least 2 positions, otherwise its neighbours would not be minimal.

We then take a certificate c at the start of a path, which is next to a certificate d. Let S be the set of positions c assigns. Let i be the position where c and d contradict each other.

We then fix every bit in S but i according to c, and we fix i according to d. The remaining function cannot be always 1, as otherwise i would not be necessary in c, but c is minimal. But on 0-inputs the remaining function is also sensitive to i because flipping it produces a word which satisfies c.

We note that in the remaining function the rest of d (not all of d was fixed because d assigns at least 2 positions) is still a valid certificate, since it only assigns one of the fixed bits and it was fixed according to d. Similarly to the first case we can minimize the remaining certificates and obtain a function with $k-1$ certificates satisfying the lemma conditions.

Then by induction the remaining function has a 0-sensitivity of $k-1$. Together with the sensitive bit i, we obtain $s_0(f) \geq k$.

Two Certificates with Two Contradictions. Let us denote these 2 certificates as c and d and the two positions where they contradict as i and j. For example, we can have 2 certificates like this:

$$\begin{pmatrix} & & i & j & \\ 1 & 1 & 1 & 0 & * \\ * & * & 0 & 1 & 1 \end{pmatrix}. \tag{15}$$

Let S be the set of positions c assigns and T be the set of positions d assigns. We then fix every bit in S except j according to c but we fix j according to d. The remaining function cannot be always 1 because, otherwise, j would not be necessary in c, but c is minimal. But on 0-inputs the remaining function is also sensitive to j, as flipping it produces a word which satisfies c.

If $|T| = 2$, then on 0-inputs the remaining function is also sensitive to i because flipping the i^{th} variable produces a word which satisfies d.

If $|T| > 2$, we examine the $2^{|T|-2}$ subfunctions obtainable by fixing the remaining positions of T. We can w.l.o.g. assume that d assigns the value 0 to each of those positions. Among the subfunctions which are not identically 1 we choose one which fixes the biggest number of bits to 0. If this subfunction fixes 0 in every position, it is sensitive to i on 0-inputs, as flipping it produces a word which satisfies d. Otherwise it is sensitive on 0-inputs to every other fixed 1 in T besides i and j, as flipping them would produce a word from a subfunction with a higher amount of bits fixed to 0. But that subfunction is identically 1 or we would have chosen it instead.

Therefore we can always find at least one additional sensitive bit among T.

Similarly to the first two cases we can minimize the remaining certificates and obtain a function with $k-2$ certificates satisfying the conditions of the lemma.

Then by induction the remaining function has a 0-sensitivity of $k-2$. Together with the two additional sensitive bits found, we obtain $s_0(f) \geq k$.

Certificate Cycles. A certificate cycle is a sequence of at least 3 certificates where each certificate has 1 contradiction with the next one and the last one has 1 contradiction with the first one. For example, here is a cycle of length 5:

$$\begin{pmatrix} j_{5,1} & & j_{1,2} & j_{2,3} & & j_{3,4} & j_{4,5} & \\ 1 & 1 & 0 & * & * & * & * & * \\ * & * & 1 & 0 & * & * & * & * \\ * & * & * & 1 & 1 & 1 & * & * \\ * & * & * & * & * & 0 & 0 & * \\ 0 & * & * & * & * & * & 1 & 1 \end{pmatrix}. \tag{16}$$

Note that every certificate in a cycle assigns at least 2 positions, otherwise its neighbours in the cycle would overlap. We denote the length of the cycle by m. Let $c_1, \ldots, c_m$ be the certificates in this cycle, let $S_1, \ldots, S_m$ be the positions assigned by them, and let $j_{1,2}, \ldots, j_{m,1}$ be the positions where the certificates contradict.

We assign values to variables in $c_2, \ldots, c_m$ in the following way. We first assign values to variables in S_2 so that the variable $j_{2,3}$ contradicts c_2 and is assigned according to c_3, but all other variables are assigned according to c_2.

We have the following properties. First, the remaining function cannot be always 1, as otherwise $j_{2,3}$ would not be necessary in c_2, but c_2 is minimal. Second, any 0-input that is consistent with the assignment that we made is sensitive to $j_{2,3}$ because flipping this position produces a word which satisfies c_2. Third, in the remaining function $c_3, \ldots, c_m$ are still valid 1-certificates because we have not made any assignments that contradict them. Some of these certificates c_i may no longer be minimal. In this case, we can minimize them by removing unnecessary variables from c_i and S_i.

We then perform a similar procedure for $c_i \in \{3, \ldots, m\}$. We assume that the variables in S_2, …, S_{i-1} have been assigned values. We then assign values to variables in S_i. If c_i and c_{i+1} contradict in the variable $j_{i,i+1}$, we assign it according to c_{i+1}. (If $i = m$, we define $i + 1 = 1$.) If c_i and c_{i+1} no longer

contradict (this can happen if $j_{i,i+1}$ was removed from one of them), we choose a variable in S_i arbitrarily and assign it opposite to c_i. All other variables in S_i are assigned according to c_i.

We now have similar properties as before. The remaining function cannot be always 1 and any 0-input that is consistent with our assignment is sensitive to changing a variable in S_i. Moreover, $c_{i+1}, \ldots, c_m$ are still valid 1-certificates and, if they are not minimal, they can be made minimal by removing variables.

At the end of this process, we have obtained $m-1$ sensitive bits on 0-inputs: for each of $c_2, \ldots, c_m$, there is a bit, changing which results in an input satisfying c_i. We now argue that there should be one more sensitive bit. To find it, we consider the certificate c_1.

During the process described above, the position $j_{1,2}$ where c_1 and c_2 contradict was fixed opposite to the value assigned by c_1. The position $j_{m,1}$ where c_1 and c_m contradict is either unfixed or fixed according to c_1. All other positions of c_1 are unfixed.

If there are no unfixed positions of c_1, then changing the position $j_{1,2}$ in a 0-input (that satisfies the partial assignment that we made) leads to a 1-input that satisfies c_1. Hence, we have m sensitive bits.

Otherwise, let $T \subset S_1$ be the set of positions in c_1 that have not been assigned and let $p = |T|$. W.l.o.g, we assume that c_1 assigns the value 0 to each of those positions. We examine the 2^p subfunctions obtainable by fixing the positions of T in some way. Among the subfunctions which are not identically 1, we choose one that was obtained by fixing the biggest number of bits to 0. If this subfunction fixes 0 in every position, it is sensitive to $j_{1,2}$ on 0-inputs, as flipping it produces a word which satisfies c_1. Otherwise it is sensitive on 0-inputs to every bit in T that is fixed to 1 because flipping this bit would produce a word from a subfunction with a higher amount of bits fixed to 0 (and this subfunction must be identically 1). Hence, we have m sensitive bits in this case.

Similarly to the first three cases, we can minimize the remaining certificates and obtain a function with $k-m$ certificates satisfying the conditions of the lemma. By induction, the remaining function has a 0-sensitivity of $k-m$. Together with the m additional sensitive bits we found, we obtain $s_0(f) \geq k$. ■

6 Conclusions

In this paper, we have shown a lower bound on 1-certificate complexity in relation to the ratio of 0-block sensitivity and 0-sensitivity:

$$C_1(f) \geq \frac{3}{2}\frac{bs_0(f)}{s_0(f)} - \frac{1}{2}. \tag{17}$$

This bound is tight, as the function constructed in Theorem 1 achieves the following equality:

$$C_1(f) = \frac{3}{2}\frac{bs_0(f)}{s_0(f)} + \frac{1}{2}. \tag{18}$$

The difference of 1 appears because the proof of Theorem 3 requires only a single 1 in each certificate but the construction of Theorem 1 has two ones in each certificate.

Thus, we have completely solved the problem of finding the optimal relationship between $s_0(f)$, $bs_0(f)$ and $C_1(f)$. For functions with $s_1(f) = C_1(f)$, such as those constructed in [3, 7, 8], this means that

$$bs_0(f) \leq \left(\frac{2}{3} + o(1)\right) s_0(f)s_1(f). \tag{19}$$

That is, if we use such functions, there is no better separation between $s(f)$ and $bs(f)$ than the currently known one.

For the general case, it is important to understand how big the gap between $s_1(f)$ and $C_1(f)$ can be. Currently, we only know that

$$s_1(f) \leq C_1(f) \leq 2^{s_0(f)-1}s_1(f), \tag{20}$$

with the upper bound shown in [2]. In the general case (17) together with this bound implies only

$$bs_0(f) \leq \left(\frac{2}{3} + o(1)\right) 2^{s_0(f)-1}s_0(f)s_1(f). \tag{21}$$

However, there is no known f that comes even close to saturating the upper bound of (20) and we suspect that this bound can be significantly improved.

There are some examples of f with gaps between $C_1(f)$ and $s_1(f)$, though. For example, the 4-bit non-equality function of [1] has $s_0(NE) = s_1(NE) = 2$ and $C_1(NE) = 3$ and it is easy to use it to produce an example $s_0(NE) = 2$, $s_1(NE) = 2k$ and $C_1(NE) = 3k$. Unfortunately, we have not been able to combine this function with the function that achieves (18) to obtain a bigger gap between $bs(f)$ and $s(f)$.

Because of that, we conjecture that (19) might actually be optimal. Proving or disproving this conjecture is a very challenging problem.

References

1. A. Ambainis. Polynomial degree vs. quantum query complexity. *J. Comput. Syst. Sci.*, 72(2):220–238, 2006.
2. A. Ambainis, M. Bavarian, Y. Gao, J. Mao, X. Sun, and S. Zuo. New decision tree complexity upper bounds in terms of sensitivity. *submitted to ICALP'2014*.
3. A. Ambainis and X. Sun. New separation between $s(f)$ and $bs(f)$. *CoRR*, abs/1108.3494, 2011.
4. H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theor. Comput. Sci.*, 288(1):21–43, 2002.
5. C. Kenyon and S. Kutin. Sensitivity, block sensitivity, and l-block sensitivity of Boolean functions. *Inf. Comput.*, 189(1):43–53, 2004.
6. N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4:301–313, 1994.
7. D. Rubinstein. Sensitivity vs. block sensitivity of Boolean functions. *Combinatorica*, 15(2):297–299, 1995.
8. M. Virza. Sensitivity versus block sensitivity of Boolean functions. *Inf. Process. Lett.*, 111(9):433–435, 2011.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de