DECISION-TREE COMPLEXITY

SHALEV BEN-DAVID

ABSTRACT. Two separate results related to decision-tree complexity are presented. The first uses a topological approach to generalize some theorems about the evasiveness of monotone boolean functions to other classes of functions. The second bounds the gap between the deterministic decision-tree complexity of functions on the permutation group S_n and their zero-error randomized decision-tree complexity.

1. Introduction

The problem of decision-tree complexity is the following. Given oracle access to input string, how many of its bits must be queried to determine the value of some function of the string? For example, the function may be OR, which takes in a boolean string and outputs 0 if the string is all zeros and 1 otherwise. In that case, it is clear that any deterministic algorithm must, in the worst case, query all bits of the input.

The worst-case query complexity of the best possible deterministic algorithm for computing a function f is called the decision-tree complexity of f (the name comes from the fact that such an algorithm can be represented as a rooted tree whose internal nodes indicate bits to query and whose edges are labeled by 0 or 1). One can define the randomized and quantum decision-tree complexities of f as the minimum worst-case expected number of queries of randomized or quantum algorithms computing f (for quantum algorithms, we allow queries in superposition).

A function f is called evasive if its (deterministic) decision-tree complexity is the number of input bits – that is, a function is evasive if any deterministic algorithm must, in the worst case, query all bits of the input. Early study of decision-tree complexity has focused on showing that certain classes of boolean functions – notably monotone functions with some symmetry – are evasive. In the first part of this project, we prove these results and generalize them to non-boolean functions.

Date: December 21, 2012.

We also present a model in which queries to the input string may be rejected, and show that our results transfer to this model.

The gaps between deterministic, randomized, and quantum decision-tree complexities for different classes of functions have also been studied, often with the goal of understanding when randomized and quantum algorithms provide a substantial increase in computational power.

In the second part of this project, we investigate these gaps for functions defined on permutations. We show that the deterministic and zero-error-randomized complexities are polynomially related for such functions. We also show that the certificate complexity is polynomially related to the quantum decision-tree complexity for these functions.

2. Basic Definitions

In this section, we provide the basic definitions that will be required for the subsequent sections. Generally speaking, we will let M be a fixed natural number which is at least 2, and denote by $[M]$ the set $\{0, 1, \ldots, M-1\}$. We will also let n be a natural number, and S a subset of $[M]^n$.

Definition 2.1. *For a fixed set $S \subseteq [M]^n$, the* deterministic decision-tree complexity *of a function $f : S \to \{0,1\}$, denoted $D(f)$, is the minimum number of adaptive queries to the characters of x that are required to determine the value of $f(x)$, in the worst case over possible values of $x \in S$. If $D(f) = n$, we say that f is* evasive*.*

One can also provide similar definitions for the decision-tree complexity in a randomized or quantum context, as follows.

Definition 2.2. *For a fixed set $S \subseteq [M]^n$, the* randomized decision-tree complexity *of a function $f : S \to \{0,1\}$, denoted $R(f)$, is the minimum number of adaptive random queries to the characters of x that are required to determine the value of $f(x)$ with probably of error at most $\frac{1}{3}$, in the worst case over possible values of $x \in \{0,1\}^n$. Similarly, $Q(f)$ is the minimum number of quantum queries needed (where we allow the characters of x to be queried in superposition), and $R_0(f)$ is the randomized complexity when the probability of error must be 0.*

For our analysis, we will be interested in special classes of functions, including monotone functions and functions closed under a group action.

Definition 2.3. *For $x, y \in [M]^n$ and $k \in [M]$ we write $x \leq_k y$ if y can be obtained from x by replacing some of the characters of x with the character k. A function $f : S \to \{0,1\}$ with $S \subseteq [M]^n$ is called* monotone in k *if whenever $x \leq_k y$ we have $f(x) \leq f(y)$.*

Definition 2.4. *Let G be a group action on $\{1, 2, \ldots, n\}$ and let $S \subseteq [M]^n$. For $x \in [M]^n$, we denote the characters of x by $x_1, x_2, \ldots, x_n$. We say that S is* closed *under G if for all $x \in S$ and $\sigma \in G$ we have $x_{\sigma(1)}x_{\sigma(2)} \ldots x_{\sigma(n)} \in S$. We say that a function $f : S \to \{0, 1\}$ is* invariant *under G if S is closed under G and for all $x \in S$, $\sigma \in G$ we have $f(x) = f(x_{\sigma(1)}x_{\sigma(2)} \ldots x_{\sigma(n)})$.*

Finally, the notions of a partial assignment and a certificate will play a major role in both parts of the project.

Definition 2.5. *Let $S \subseteq [M]^n$. For $x \in S$, a* partial assignment *of x is a set of pairs $D = \{(i_1, x_{i_1}), (i_2, x_{i_2}), \ldots\}$. The* inputs *of D are $i_1, i_2, \ldots$, and the* outputs *of D are $x_{i_1}, x_{i_2}, \ldots$ An* extension *of D is any $y \in S$ that agrees with x on the inputs of D. D is called a* certificate *for a function $f : S \to \{0, 1\}$ if $f(y)$ is fixed for all extensions y of D. D is called a 1-certificate of the fixed value is 1, and a 0-certificate if it is 0. The* size *of D refers to the number of pairs in D.*

Definition 2.6. *Let $S \subseteq [M]^n$ and $x \in S$. A* subcertificate *of x with respect to $f : S \to \{0, 1\}$ is a partial assignment of x which is a certificate. A subcertificate of x is minimum if it has the smallest size out of all subcertificates of x. The size of the minimum subcertificate of x is denoted by $C_x(f)$. We define the certificate complexity of f, denoted by $C(f)$, to be be the maximum of $C_x(f)$ over all $x \in S$.*

Part 1. Evasiveness Using Topology

3. Topological Background

In this section we introduce the basic topological concepts that will be needed in our analysis. The basic building block for the topological tools we will apply is the simplicial complex.

Definition 3.1. *A* simplicial complex *is a set system which is closed under subsets. A simplicial complex will have an associated* ground set*, which contains the union of the sets in the system, but may be larger.*

Given a simplicial complex Γ with ground set of size n, we identify the ground set with n affinely independent points in $\mathbb{R}^m$ (with $m \geq n-1$). Furthermore, for each set $A \in \Gamma$, we identify A with the simplex given by the convex hall of the elements of A. We identify the simplicial complex Γ with the union of the simplicies A for all $A \in \Gamma$.

Note that the topology of the above representation of a simplicial complex is independent of the choices of m and of the points identified with the ground set. This topology is the only property of the representation that we will use.

Next, we introduce some notation which will allow us to discuss collapsible simplicial complexes.

Definition 3.2. *For a given simplicial complex Γ, the sets in Γ are called* faces*; a face is called* maximal *if it is not contained in any other face; a face is* free *if it is not maximal and is contained in only one maximal face.*

Definition 3.3. *An* elementary collapse *of a simplicial complex Γ is a new simplicial complex given by deleting a free face and all faces that contain it. A simplicial complex is called* collapsible *if, through a sequence of elementary collapses, it is possible to reduce the simplicial complex to the empty simplicial complex (in other words, if it is possible to reduce the set system to one with no sets).*

Collapsibility of a simplicial complex is still a fairly combinatorial property, but it is intimately related to the topological notion of contractibility.

Definition 3.4. *A topological space T is* contractible *if there is a continuous map $\Phi : T \times [0,1] \to T$ such that some fixed $p \in T$ and for all $x \in T$, we have $\Phi(x,0) = x$ and $\Phi(x,1) = p$.*

Theorem 3.5. *If a simplicial complex is collapsible then it is contractible as a topological space.*

The previous theorem is standard in topology, and we omit its proof.

Finally, we define the operations of link and delete in order to develop tools for demonstrating that a simplicial complex is collapsible.

Definition 3.6. *For a simplicial complex Γ and an element u of its ground set, the* link *of u in Γ is the simplicial complex*

$$\Gamma/u := \{A\backslash\{x\} : x \in A \in \Gamma\}$$

and Γ delete u is

$$\Gamma\backslash u := \{A : x \notin A \in \Gamma\}.$$

Lemma 3.7. *For a simplicial complex Γ, if Γ/u and $\Gamma\backslash u$ are collapsible for all u, then Γ is collapsible.*

The previous theorem is not difficult, and has an entirely combinatorial proof. We omit the proof because we prove a generalization of it in the next section.

4. Topology and Decision Trees

The connection between decision trees and topology was first demonstrated in [1], in which the following simplicial complex was defined.

Definition 4.1. *The simplicial complex Γ_f of a monotone boolean function $f : \{0,1\}^n \to \{0,1\}$ is the collection of all sets of coordinates such that if $x \in \{0,1\}^n$ is zero exactly on these coordinates, then $f(x) = 1$.*

The following theorem was proved in that paper:

Theorem 4.2. *If a monotone boolean function f is not evasive, then Γ_f is collapsible.*

We generalize the above definition both to functions that are not monotone and to functions on a larger character set. We then prove a generalized version of the previous theorem.

Definition 4.3. *Given a function $f : [M]^n \to \{0,1\}$, let*

$$f^r : [M] \cup \{*\} \to \{0,1\}$$

be given by $f^r(x) = 0$ if and only if x is a certificate for f when treated as a partial assignment for $[M]$ (with $*$ representing an unassigned coordinate).

Note that f^r has a character set that is one larger than that of f. Also note that f is monotone in $*$.

Definition 4.4. *Let $f : [M+1]^n \to \{0,1\}$ be monotone in M. We define a simplicial complex Γ_f as follows. The ground set will be all the pairs (i,k) for $i = 1, 2, \ldots, n$ and $k \in [M]$. Call a subset A of the ground set* valid *if each $i = 1, 2, \ldots, n$ occurs as the first coordinate of at most one pair in A. Note that each valid subset of the ground set corresponds to a partial assignment for $[M+1]$. Let x_A be the extension of that partial assignment given by placing M in each unassigned coordinate. We define the faces of the simplicial complex Γ_f to be the sets A such that A is valid and $f(x_A) = 1$.*

Note that since f is monotone in M, Γ_f is closed under subsets, so it is a simplicial complex. Now, for any (not necessarily monotone) function $f : [M]^n \to \{0,1\}$, we will consider the simplicial complex Γ_{f^r}. Note that the faces of Γ_{f^r} correspond exactly to the partial assignments of $[M]$ that are not certificates for f.

We now obtain the following generalization of 4.2.

Theorem 4.5. *Let $f : [M+1]^n \to \{0,1\}$ be monotone in M. If f is not evasive then Γ_f is collapsible.*

To prove this, we first prove a generalization of 3.7.

Lemma 4.6. *Let Γ be a simplicial complex with ground set C. Suppose that C can be written as the disjoint union of sets $A_1, A_2, \dots, A_n$ such that two elements from the same set A_i never occur together in any face of Γ. Suppose further that for some set A_k, it is the case that $\Gamma \backslash A_k$ is collapsible and that Γ/x is collapsible for all $x \in A_k$. Then Γ is collapsible.*

proof (of lemma 4.6). Note that if x and y are both in A_k, then Γ/x and Γ/y share no faces. Note further that B is a free face of Γ/x if and only if $B \cup \{x\}$ is a free face of Γ. It follows that we may apply the sequences of elementary collapses that collapse Γ/x to Γ/a_i for each $a_i \in A_k$. Because these simplicial complexes don't share faces, there will be no conflicts when collapsing all of these faces. The result of this will be the simplicial complex $\Gamma \backslash A_k$. Since $\Gamma \backslash A_k$ is collapsible, so is Γ. $\square$

Note that the ground set of Γ_f splits into disjoint sets $A_i := \{(i,k) : k \in [M]\}$ such that no face of Γ_f contains two members of A_i. Moreover, note that the simplicial complex of the function $f|_{x_i=k}$ is $\Gamma/(i,k)$ if $k \neq M$, and $\Gamma \backslash A_i$ if $k = M$.

proof (of theorem 4.5). By way of contradiction, consider the smallest n for which the theorem does not hold, and let f be not evasive with Γ_f not collapsible. It is easy to verify that $n > 1$. Consider an algorithm that finds $f(x)$ by querying less than n elements, for all x. Let i be the index of the first character that this algorithm examines. Then after examining this character, the function f becomes $f|_{x_i=k}$, so $f|_{x_i=k}$ is not evasive for all $k \in [M]$. Now, since $f|_{x_i=k}$ is a function on $n-1$ characters, it follows from the previous theorem that its simplicial complex is collapsible. But this implies that $\Gamma \backslash A_k$ is collapsible and that Γ/x is collapsible for all $x \in A_k$, so by the previous theorem, Γ_f is collapsible, giving the desired contradiction. $\square$

Next, we need the following definitions.

Definition 4.7. *An* automorphism *of a topological space T is a map $\phi : T \to T$ which is continuous, invertible, and has continuous inverse.*

Definition 4.8. *Let Γ be a simplicial complex with ground set C. Let σ be a permutation of C. We define an automorphism $\sigma_\Gamma : \Gamma \to \Gamma$ of Γ as follows. The representation of c in $\mathbb{R}^m$ gets mapped to the representation of $\sigma(c)$ for all $c \in C$. Then, since every other point in the topological space Γ can be written as a convex combination of points in C, we linearly extend the definition of σ_Γ to the rest of Γ.*

It is easy to see that σ_Γ is indeed an automorphism.

Definition 4.9. *Let $f : [M+1]^n \to \{0,1\}$ be monotone in M. We say f is* proper *if $f(M^n) = 1$ and $f(x) = 0$ whenever x contains no instances of the character M.*

Finally, we make the following observation.

Lemma 4.10. *Let $f : [M+1]^n \to \{0,1\}$ be monotone in M and proper. Let G be a transitive group action on $\{1,2,\dots,n\}$ and let f be invariant under G. We let G act on the ground set of Γ_f by letting it act on the first coordinate of pairs (i,k). Then there is no point z in the topological space Γ_f with the property that $\sigma_{\Gamma_f}(z) = z$ for all $\sigma \in G$.*

Proof. Let the points in the topological space Γ_f that represent the ground set be denoted by $c_(i,k)$ for all pairs (i,k). If a fixed point z with the desired properties existed, then it would lie on some face of Γ_f. The verticies of that face would be some set of points $A = \{c_(i_1,k_1), c_(i_2,k_2), \dots\}$. We would then have $i_a \neq i_b$ whenever $a \neq b$, since no two pairs with the same first coordinate lie in the same face. It follows that each σ_{Γ_f} maps points in A to points in A (since otherwise, it could not fix z). Now, since G is transitive, we conclude that A contains a pair $(i,*)$ for all values of i. This means the partial assignment x_A given by the face A is actually a full assignment. Now, by the definition of Γ_f, x_A contains no instances of the character M, and $f(x_A) = 1$. This contradicts the assumption that f is proper. □

We can now use fixed-point theorems from topology to show evasiveness. One basic fixed-point theorem is the following.

Theorem 4.11. (Lefshet's Fixed point theorem) *Any automorphism of a contractible space to itself has a fixed point.*

Using this we get the following result.

Corollary 4.12. *Let $f : [M+1]^n \to \{0,1\}$ be monotone in M and proper. If f is invariant under a cyclic group action then f is evasive. In particular, if $f : [M]^n \to \{0,1\}$ is non-trivial and invariant under a cyclic group action, then f^r is evasive.*

Another useful fixed-point theorem is the following.

Theorem 4.13. *Let Ψ be a group of automorphisms on (the representation of) a contractible simplicial complex Γ. Suppose there is a normal subgroup Ψ_1 of Ψ of prime power order. Suppose further that Ψ/Ψ_1 is cyclic. Then there is a point z on (the representation of) Γ such that for all $\psi \in \Psi$, $\psi(z) = z$.*

From this we get the following corollary.

Corollary 4.14. *Let $f : [M+1]^n \to \{0,1\}$ be monotone in M and proper. Let G be a transitive group action with a normal subgroup of prime power order such that the quotient group is cyclic. If f is invariant under the G then f is evasive. In particular, if $f : [M]^n \to \{0,1\}$ is non-trivial and invariant under such a group action, then f^r is evasive.*

In [3] it is shown that such a group action occurs in the graph setting, as a subgroup of the group action that relabels vertices of a graph, as long as the number of vertices of the graph is a prime power. From this construction, we conclude the following.

Corollary 4.15. *Let $f : [M+1]^{\frac{n(n-1)}{2}} \to \{0,1\}$ be monotone in M and proper, and let n be a prime power. Consider $[M+1]$ as labelling the edges of a graph, and let f be invariant under relabelling of the vertices of the graph. Then f is evasive.*

Part 2. Decision-Tree Complexity on Permutations

In this part of the project we analyze the decision-tree complexity of functions whose inputs are permutations. We prove that $R_0(f)$ and $D(f)$ are polynomially (indeed, cubically) related for such functions. We also define sensitivity and block sensitivity for these functions, and prove a quadratic relationship between the block sensitivity $bs(f)$ and the certificate complexity $C(f)$. It follows from this that there is a polynomial relationship between $C(f)$ and $Q(f)$ for functions on permutations.

$S_n \subseteq [M]^n$ will denote the symmetric group on n elements. We analyze functions $f : S_n \to \{0,1\}$. For permutations $\sigma, \tau \in S_n$, we will write $\sigma(i)$ instead of σ_i to denote the permutation applied to $i \in \{1,2,\ldots,n\}$, and we will write $\sigma\tau$ to denote the product in S_n or function composition (and not concatenation).

5. $D(f)$ and $R_0(f)$ on Permutations

In this section we prove that the gap between $R_0(f)$ and $D(f)$ is at most cubic. The main work involved is in proving the following lemma. The lemma says that if we fix some small number of values of a permutation fixed, we can make it so that all small certificates have an output belonging to some fixed small set.

Lemma 5.1. *If $k \leq \frac{1}{2}\sqrt{D(f)}$ then there are two sets $A, B \subseteq \{1,2,\ldots,n\}$, each of size at most $4k^2$, and a permutation $\sigma \in S_n$ with $\sigma(A) \cap B = \emptyset$ such that for any $\tau \in S_n$ that agrees with σ on A, all subcertificates of τ of size at most k have at least one output in B.*

Proof. Consider any 0-certificate C_0 and any 1-certificate C_1. These certificates cannot be subcertificates of the same permutation; we say that they *conflict*. In particular, either there is some input for which C_0 claims a different output than C_1 or else there is some output for which C_0 gives a different input than C_1. In the former case, we say that these certificates *disagree on an input*; in the latter case we say they *disagree on an output*.

We now present an algorithm that, when run on input permutation σ, will either output $f(\sigma)$ or else will find A and B satisfying the desired conditions. The algorithm will query σ on the inputs of different certificates and keep track of D, the partial assignment given by these queries. In addition, the algorithm will keep track of the set B of the predicted outputs of the certificates. In the beginning, $B = D = \emptyset$. The algorithm is given by the following loop.

```
loop
    (1) Choose a certificate $C$ of size at most $k$ which is consistent with
        the partial assignment $D$ and which has no outputs in $B$. If no
        such certificate exists, halt.
    (2) Add the outputs of $C$ to $B$.
    (3) Query all the inputs of $C$, and add these queries to $D$.
    (4) If any outputs of $D$ are in $B$, remove them from $B$.
    (5) If $D$ is a certificate, output the corresponding value of $f(\tau)$ and
        halt.
end loop
```

We first show that this loop repeats at most $4k$ times. To do this, we define the *hidden information* of a certificate C with respect to B and D, denoted $h_{B,D}(C)$. If C conflicts with D, we define $h_{B,D}(C) = 0$. Otherwise, we define $h_{B,D}(C)$ to be 2 times (the number of entries of C that are not entries of D) minus (the number of outputs of C that are in B). Note that when the algorithm starts, $h_{\emptyset,\emptyset}(C)$ is twice the size of C.

We show that in each iteration of the loop, if the certificate chosen in that iteration is a 0-certificate, then the hidden information of all 1-certificates decreases. Similarly, if the chosen certificate is a 1-certificate, then the hidden information of all 0-certificates decreases.

Indeed, consider any iteration of the loop, and suppose without loss if generality that the chosen certificate in that iteration is a 0-certificate, say C_0. Then C_0 agrees with D and has no outputs in B. Let C_1 be any 1-certificate. Then C_1 must conflict with C_0. If they disagree on an input, then this input gets queried, which decreases the hidden information of C_1 by at least 1. If they disagree on an output, then this output gets added to B, which decreases the hidden information of C_1 by 1 (note that this output may be removed from B in the same iteration, but this would mean that either a conflict with C_1 was found or else an entry of C_1 was revealed - in either case, this decreases the hidden information of C_1 further).

It follows that each iteration decreases by 1 the hidden information either of all 0-certificates or of all 1-certificates. It is easy to see that the loop will stop if the hidden information of all 0-certificates of size at most k is 0, and similarly for 1-certificates. Since the hidden information of certificates of size at most k starts at $2k$, we conclude that the number of iterations before the loop halts is less than $4k$.

Next, we observe that since the main loop breaks in $4k$ iterations and queries at most k inputs of σ in each iteration, it uses less than $4k^2$ queries on each input permutation σ. Now, since $k \leq \frac{1}{2}\sqrt{D(f)}$, the algorithm uses less than $D(f)$ queries, so there is some σ for which it does not output $f(\sigma)$. Hence, when run on σ, the algorithm outputs B and D such that every certificate of size at most k either has an output in B or else conflicts with D.

Let A be the set of all inputs of D; in other words, A is the set of all inputs that were queried. Then $|A| < 4k^2$ and $|B| < 4k^2$. Moreover, if τ agrees with σ on A, then it extends D, and thus all its subcertificates of size at most k have an output in B.

□

We are now ready to prove the main result.

Theorem 5.2. $R_0(f) = \Omega(\min(\sqrt{D(f)}, n^{\frac{1}{3}}))$.

Proof. Apply lemma 5.1 with $k = \min(\frac{1}{2}\sqrt{D(f)}, n^{\frac{1}{3}})$ to find σ, A, and B. Consider a uniform distribution over the permutations that agree with σ on A. This distributes the elements of B uniformly over all inputs that are not in A. Now, finding a certificate of size less than k requires finding an element of B. There are at most $4k^2$ such elements, and they are placed uniformly in at least $n-4k^2$ spots, so it follows that expected time $\Omega(\frac{n}{k^2})$ is needed to find one. Thus finding a certificate of size less than k takes $\Omega(\frac{n}{k^2})$ queries. Finding a certificate of size at least k takes at least k queries. It follows that $R_0(f) = \Omega(\min(\sqrt{D(f)}, n^{\frac{1}{3}}))$, as desired.

□

6. Block Sensitivity

In this section, we define analogues of sensitivity and block sensitivity for functions of permutations. We prove a quadratic relationship between block sensitivity and certificate complexity for these functions, and conclude that the quantum query complexity of such functions is polynomially related to the certificate complexity.

Definition 6.1. *We say that two permutations $\sigma, \tau \in S_n$ are* disjoint *if for all $i = 1, 2, \ldots, n$ either $\sigma(i) = i$ or $\tau(i) = i$.*

Definition 6.2. *A* sensitive block *of an input $\sigma \in S_n$ with respect to $f : S_n \to \{0, 1\}$ is a permutation $\tau \in S_n$ such that $f(\sigma\tau) \neq f(\sigma)$. The* block sensitivity *of σ is the maximum size of a set of disjoint sensitive blocks of σ. The block sensitivity of σ is denoted by $bs_\sigma(f)$. The block sensitivity of the function f is the maximum value of $bs_\sigma(f)$ over all $\sigma \in S_n$.*

Definition 6.3. *The* length *of a permutation $\tau \in S_n$, denoted by $l(\tau)$, is the number of inputs which it does not fix. The* sensitivity *of $\sigma \in S_n$ with respect to $f : S_n \to \{0, 1\}$, denoted by $s_\sigma(f)$, is the maximum number of disjoint sensitive blocks of σ, all of which are of length 2. The sensitivity of a function f is the maximum of $s_\sigma(f)$ over all values of $\sigma \in S_n$.*

Definition 6.4. *A sensitive block τ for σ is* minimal *if no sensitive block of σ fixes a strict subset of the inputs that τ fixes.*

Lemma 6.5. *Let $\tau \in S_n$ be a minimal sensitive block for $\sigma \in S_n$ with respect to $f : S_n \to \{0, 1\}$. Then $s_{\sigma\tau}(f) \geq \frac{1}{3}l(\tau)$, and in particular, $bs(f) \geq s(f) \geq \frac{1}{3}l(\tau)$.*

Proof. Let the cycle decomposition of τ be

$$\tau = (a_{11}a_{12}\ldots a_{1k_1})(a_{21}a_{22}\ldots a_{2k_2})\ldots(a_{m1}a_{m2}\ldots a_{mk_m}).$$

For each consecutive pair $(a_{ij}, a_{i(j+1)})$ in one of the cycles, consider the permutation ϕ_{ij} that flips it. Then $\tau\phi_{ij}(a_{i(j+1)}) = \tau(a_{ij}) = a_{i(j+1)}$, so the fixed points of $\tau\phi_{ij}$ are a superset of the fixed point of τ. By the minimality of the sensitive block τ, we must have $f(\sigma\tau\phi_{ij}) = f(\sigma)$, whence $f(\sigma\tau\phi_{ij}) \neq f(\sigma\tau)$. It follows that each ϕ_{ij} is sensitive for f on $\sigma\tau$. We now select a disjoint set of such flips. It's possible to select $\lfloor\frac{k_1}{2}\rfloor + \lfloor\frac{k_2}{2}\rfloor + \cdots + \lfloor\frac{k_m}{2}\rfloor \geq \frac{k_1}{3} + \frac{k_2}{3} + \cdots + \frac{k_m}{3} = \frac{1}{3}l(\tau)$ such flips, so it follows that $s_{\sigma\tau} \geq \frac{1}{3}l(\tau)$. □

Lemma 6.6. *For all $f : S_n \to \{0, 1\}$, $C(f) \leq 3bs(f)^2$.*

Proof. Let $\sigma \in S_n$ have minimum subcertificate of size $C(f)$. Let $B \subseteq S_n$ be a set of sensitive blocks of size $bs_\sigma(f)$ whose sensitive blocks are minimal. Let $G \subseteq \{1, 2, \ldots, n\}$ be the set of inputs that are not fixed by B.

Claim: the set $\{(g, \sigma(g)) : g \in G\}$ forms a subcertificate of σ. Indeed, if not, then there would be some $\tau \in S_n$ that agrees with σ on G for which $f(\tau) \neq f(\sigma)$. Let $\phi = \sigma^{-1}\tau$. Then for each $g \in G$, we have $\phi(g) = \sigma^{-1}(\tau(g)) = \sigma^{-1}(\sigma(g)) = g$, so ϕ is disjoint from all members of B. In addition, $f(\sigma\phi) = f(\sigma\sigma^{-1}\tau) = f(\tau) \neq f(\sigma)$. Thus $B \cup \{\phi\}$ is a larger sensitive block, contradicting the assumption that $|B| = bs_\sigma(f)$.

Hence, we conclude that $|G| \geq C_\sigma(f) = C(f)$. Each input in G is not fixed by at least one member of B. By the pigeonhole principle, there is some τ in B of length at least $\frac{|G|}{bs_\sigma(f)}$. Then by lemma 3.5, $bs(f) \geq \frac{|G|}{3bs_\sigma(f)}$, so $C(f) \leq |G| \leq 3bs_\sigma(f)bs(f) \leq 3bs(f)^2$. □

We can relate these results to the quantum query complexity by using the following lemma.

Lemma 6.7. *For $f : S_n \to \{0,1\}$, $Q(f) = \Omega(\sqrt{bs(f)})$.*

Proof. Consider a permutation σ with $bs(f)$ sensitive blocks $\tau_1, \tau_2, \ldots, \tau_{bs(f)}$. The value of f on each of the inputs $\sigma\tau_1, \sigma\tau_2, \ldots, \sigma\tau_{bs(f)}$ is the same, and is different from the value of f on σ. Moreover, all of these inputs differ from σ in mutually exclusive sets of coordinates. It then follows from the lower bound on the Grover search problem that $Q(f) = \Omega(\sqrt{bs(f)})$. □

Finally, we get the following corollary.

Corollary 6.8. *For $f : S_n \to \{0,1\}$, we have $Q(f) = \Omega(C(f)^{\frac{1}{4}})$.*

We note, however, that for some functions, such as f defined by $f(\sigma) = 1 \iff \sigma^{-1}(1) < \frac{n}{2}$, we have $C(f) = 1$ while $Q(f)$ grows polynomially with n.

References

[1] J. Kahn, M. Saks, D. Sturtevant, A Topological Approach to Evasiveness, *Combinatorica* **4**(4) (1984), 297-306.

[2] L. Lovász, N. Young, Lecture Notes on Evasiveness of Graphs, arXiv:cs/0205031

[3] D. Du, K. Ko *Theory of Computational Complexity*, Wiley-Interscience, 2000.