Electronic Colloquium on Computational Complexity, Report No. 75 (2010)

Least span program witness size equals the general adversary lower bound on quantum query complexity

Ben W. Reichardt
Institute for Quantum Computing
University of Waterloo

Abstract

Span programs form a linear-algebraic model of computation, with span program "size" used in proving classical lower bounds. Quantum query complexity is a coherent generalization, for quantum algorithms, of classical decision-tree complexity. It is bounded below by a semi-definite program (SDP) known as the general adversary bound. We connect these classical and quantum models by proving that for any boolean function, the optimal "witness size" of a span program for that function coincides exactly with the general adversary bound.

A consequence is an optimal quantum algorithm for evaluating "balanced," read-once formulas over any finite boolean gate set. For example, the gate set may be taken to be all functions $\{0,1\}^k \to \{0,1\}$ with $k \leq 1000$. A formula is a tree whose nodes are associated to functions from the gate set. The notion of balance is technical, but it includes layered formulas. A previous quantum algorithm optimally evaluates formulas for which an optimal span program is given for each constant-size gate. However, span programs have been found only by hand. The SDP automates this procedure, and its value surprisingly always matches the lower bound. Other implications of the SDP include an exact composition rule for the general adversary bound, and that the general adversary bound upper-bounds the sign degree.

The connection can also be seen as half of a universality result for span programs. For any boolean function, there exists a span program with witness size at most the function's quantum query complexity. Conversely, solutions to the SDP give span programs, and therefore also new quantum algorithms—beyond evaluating formulas. Subsequent work has bounded the query complexity by the witness size, thus implying that the general adversary bound is tight.

1 Introduction

Quantum algorithms for evaluating formulas have developed rapidly since Farhi, Goldstone and Gutmann's breakthrough AND-OR formula-evaluation algorithm [FGG08]. The set of allowed gates in the formula has increased from just AND and OR gates to include all boolean functions on up to three bits, and many four-bit functions, with certain technical balance conditions [RŠ08]. The new algorithms can be interpreted as evaluating span programs, a certain linear-algebraic computational model [KW93]. A function can be added to the gate set if a span program is found whose complexity matches an adversary lower bound.

This paper is motivated by three main puzzles:

1. Can the gate set allowed in the formula-evaluation algorithm be extended further? Given that the search for optimal span programs has been entirely ad hoc, yet still quite successful, it seems that the answer must be yes. How far can it be extended, though?

ISSN 1433-8092

2. The adversary bounds are lower bounds on the number of queries to the input that a quantum algorithm needs to evaluate a function [Amb02, ŠS06, HLŠ07]. There are two different adversary bounds, $\mathrm{Adv} \leq \mathrm{Adv}^{\pm}$ (Definition 4.1), but the power of the latter bound is not fully understood. What is the relationship between span program complexity, or "witness size" (Definition 3.3), and the adversary lower bounds on quantum query complexity? There appears to be a close connection. For example, so far all known optimal span programs are for functions f with $\mathrm{Adv}(f) = \mathrm{Adv}^{\pm}(f)$.

3. Aside from their applications to formula evaluation, can span programs be used to derive other quantum algorithms?

We answer the first two questions, and give a partial answer to the third. Unexpectedly, we find that for any boolean function f, the optimal span program has witness size equal to the general adversary bound $\mathrm{Adv}^{\pm}(f)$. This result allows us to optimally evaluate balanced, read-once formulas over any finite gate set, quantumly. Classically, optimal formula-evaluation algorithms are known only for a limited class of formulas using AND and OR gates, and a few other special cases.

The optimization result also suggests that there is a fundamental connection between span programs and quantum algorithms. A good quantum query algorithm implies a low adversary bound and hence the existence of a good span program. Conversely, based on the adversary lower bound, one can construct a span program. The span program can be turned into a quantum algorithm using ideas from [RŠ08]. This provides a new quantum algorithm design technique for problems beyond formula evaluation. Unfortunately, it has not been known how to analyze the algorithm's query complexity, or more precisely, how to set the query complexity in order to ensure correctness. Subsequent work [Rei09c, Rei10] has resolved this question, showing that the query complexity can be bounded by the witness size. In combination, the two results imply that the general adversary bound, $\mathrm{Adv}^{\pm}$, is tight for every boolean function. The results also imply that quantum computers, measured by query complexity, and span programs, measured by witness size, are equivalent computational models.

Some further background material is needed to introduce and place in context the results. Farhi, Goldstone and Gutmann in 2007 gave a nearly optimal quantum algorithm for evaluating balanced binary AND-OR formulas [FGG08, CCJY07]. This was extended by Ambainis et al. to a nearly optimal quantum algorithm for evaluating all AND-OR formulas, and an optimal quantum algorithm for evaluating "approximately balanced" AND-OR formulas [ACR$^+$07, Rei09a].

Reichardt and Špalek gave an optimal quantum algorithm for evaluating "adversary-balanced" formulas over a considerably extended gate set [RŠ08], including in particular:

- All functions $\{0,1\}^n \rightarrow \{0,1\}$ for $n \leq 3$, such as AND, OR, PARITY and MAJ_3.
- 69 of the 92 inequivalent functions $f : \{0,1\}^4 \rightarrow \{0,1\}$ with $\mathrm{Adv}(f) = \mathrm{Adv}^{\pm}(f)$.

They derived this result by generalizing the previous approaches to consider span programs, a computational model introduced by Karchmer and Wigderson [KW93]. They then derived a quantum algorithm for evaluating certain concatenated span programs, with a query complexity upper-bounded by the span program witness size. Thus in fact the allowed gate set includes all functions $f : \{0,1\}^n \rightarrow \{0,1\}$, with $n = O(1)$, for which we have a span program P computing f and with witness size $\mathrm{wsize}(P) = \mathrm{Adv}^{\pm}(f)$. A special case of [RŠ08, Theorem 4.7] is:

Theorem 1.1 ([RŠ08])**.** *Fix a function $f : \{0,1\}^n \to \{0,1\}$. For $k \in \mathbf{N}$, define $f^k : \{0,1\}^{n^k} \to \{0,1\}$ as follows: $f^1 = f$ and $f^k(x) = f^{k-1}\big(f(x_1, \ldots, x_n), \ldots, f(x_{n^k-n+1}, \ldots, x_{n^k})\big)$ for $k > 1$. If span program P computes f, then the bounded-error quantum query complexity of f^k, $Q(f^k)$, satisfies*

$$Q(f^k) = O(\mathrm{wsize}(P)^k) \ . \tag{1.1}$$

[RŠ08] followed an ad hoc approach to finding optimal span programs for various functions. Although successful so far, continuing this method seems daunting for a few reasons:

- For most functions f, probably $\mathrm{Adv}^\pm(f) > \mathrm{Adv}(f)$. Indeed, there are 222 four-bit boolean functions, up to the natural equivalences, and for only 92 of them does $\mathrm{Adv}^\pm = \mathrm{Adv}$ hold. For no function with a gap has a span program matching $\mathrm{Adv}^\pm(f)$ been found. This suggests that perhaps span programs can only work well for the special cases when $\mathrm{Adv}^\pm = \mathrm{Adv}$.

- Moreover, for all the functions for which we know an optimal span program, it turns out that an optimal span program can be built just by using AND and OR gates with optimized weights. (This fact has not been appreciated; see [Rei09c, App. A].) On the other hand, there is no reason to think that optimal span programs will in general have such a limited form.

- Finally, it can be difficult to prove a span program's optimality. For several functions, we have found span programs whose witness sizes match Adv numerically, but we lack a proof.

In any case, the natural next step is to try to automate the search for good span programs. A main difficulty is that there is considerable freedom in the span program definition, e.g., span programs are naturally continuous, not discrete objects. The search space needs to be narrowed.

We show that it suffices to consider span programs written in so-called "canonical" form. This form was introduced by [KW93], but its significance for developing quantum algorithms was not at first appreciated. We then find a semi-definite program (SDP) for varying over span programs written in canonical form, optimizing the witness size. This automates the search for span programs.

Remarkably, the SDP has a value that corresponds exactly to the general adversary bound $\mathrm{Adv}^\pm$, in a new formulation. Thus we characterize optimal span program witness size:

Theorem 1.2. *For any function $f : \{0,1\}^n \to \{0,1\}$,*

$$\inf_P \mathrm{wsize}(P) = \mathrm{Adv}^\pm(f) \ , \tag{1.2}$$

where the infimum is over span programs P computing f. Moreover, this infimum is achieved.

This result greatly extends the gate set over which the formula-evaluation algorithm of [RŠ08] works optimally. In fact, it allows the algorithm to run on formulas with any finite gate set (Theorem 6.5). A factor is lost that depends on the gates, but for a finite gate set, this will be a constant. As another corollary, Theorem 1.2 also settles the question of how the general adversary bound behaves under function composition (Theorem 4.8), and it implies that the sign degree of a boolean function is upper-bounded by the general adversary bound (Corollary 6.4).

By Theorems 1.1 and 1.2 together, we obtain that for any function $f : \{0,1\}^n \to \{0,1\}$,

$$\lim_{k\to\infty} Q(f^k)^{1/k} = \mathrm{Adv}^\pm(f) \tag{1.3}$$

(Theorem 6.2). It seems natural to ask whether the limit is necessary. Is in fact $Q(f) = \Theta(\mathrm{Adv}^{\pm}(f))$, with the optimal algorithm perhaps based on the optimal span program? Can span programs be used to develop quantum algorithms for problems beyond formula evaluation? Indeed, the quantum walk technique of [RŠ08] can be applied to evaluate arbitrary span programs. However, the analysis in [RŠ08] only holds for the repeated concatenation of constant-size programs, i.e., for evaluating formulas. Recent work [Rei09c, Rei10] has extended the analysis to arbitrary span programs, giving

$$Q(f) = O(\mathrm{wsize}(P)) \ . \tag{1.4}$$

for any span program P computing f. Combined with Theorem 1.2, this implies that the general adversary bound is tight, and that span programs are equivalent to quantum query algorithms for evaluating boolean functions.

This article is based on a portion of the arXiv preprint [Rei09c], an abbreviated version of which has appeared in [Rei09d].

2 Notation

For a natural number $n \in \mathbf{N}$, let $[n] = \{1, 2, \ldots, n\}$. Let $B = \{0, 1\}$. For a bit $b \in B$, let $\bar{b} = 1 - b$ denote its complement. A function f with codomain B is a (total) boolean function if its domain is B^n for some $n \in \mathbf{N}$; f is a partial boolean function if its domain is a subset $\mathcal{D} \subseteq B^n$.

The complex numbers are denoted by $\mathbf{C}$. For a finite set X, let $\mathbf{C}^X$ be the inner product space $\mathbf{C}^{|X|}$ with orthonormal basis $\{|x\rangle : x \in X\}$. We assume familiarity with ket notation, e.g., $\sum_{x \in X} |x\rangle\langle x| = \mathbf{1}$ the identity on $\mathbf{C}^X$. The ℓ_2 vector and operator norms are denoted by $\| \cdot \|$. For vector spaces V and W over $\mathbf{C}$, let $\mathcal{L}(V, W)$ denote the set of all linear transformations from V into W, and let $\mathcal{L}(V) = \mathcal{L}(V, V)$.

3 Span programs and canonical span programs

A span program P is a certain linear-algebraic way of specifying a boolean function f_P [KW93, GP03]. Roughly, a span program consists of a target $|t\rangle$ in a vector space V, and a collection of subspaces $V_{j,b} \subseteq V$, for $j \in [n]$, $b \in B$. For an input $x \in B^n$, $f_P(x) = 1$ when the target can be reached using a linear combination of vectors in $\cup_{j \in [n]} V_{j,x_j}$. For our complexity measure on span programs, however, it will be necessary to fix a set of "input vectors" that span each subspace $V_{j,b}$. We desire to span the target using a linear combination of these vectors with small coefficients.

Formally we therefore define a span program as follows:

Definition 3.1 (Span program [KW93])**.** *Let $n \in \mathbf{N}$. A span program P consists of a "target" vector $|t\rangle$ in a finite-dimensional inner-product space V over $\mathbf{C}$, together with "input" vectors $|v_i\rangle \in V$ for $i \in I$. Here the index set I is a disjoint union $I = I_{\mathrm{free}} \sqcup \bigsqcup_{j \in [n], b \in B} I_{j,b}$.*

To P corresponds a function $f_P : B^n \to B$, defined by

$$f_P(x) = \begin{cases} 1 & \text{if } |t\rangle \in \mathrm{Span}(\{|v_i\rangle : i \in I_{\mathrm{free}} \cup \bigcup_{j \in [n]} I_{j,x_j}\}) \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Additionally, let $A \in \mathcal{L}(\mathbf{C}^I, V)$ be the linear operator whose columns are the input vectors. For an input x, let $\Pi(x) \in \mathcal{L}(\mathbf{C}^I)$ be the projection onto the indices of the input vectors available for x,

$$A = \sum_{i \in I} |v_i\rangle\langle i| \tag{3.2}$$

$$\Pi(x) = \sum_{i \in I_{\text{free}} \cup \bigcup_{j \in [n]} I_{j,x_j}} |i\rangle\langle i| \ . \tag{3.3}$$

Lemma 3.2. *For a span program P, $f_P(x) = 1$ if and only if $|t\rangle \in \operatorname{Range}(A\Pi(x))$. Equivalently, $f_P(x) = 0$ if and only if $\Pi(x)A^\dagger|t\rangle \in \operatorname{Range}\Big[\Pi(x)A^\dagger\Big(\mathbf{1} - \frac{|t\rangle\langle t|}{\|t\|^2}\Big)\Big]$.*

Lemma 3.2 follows from Eq. (3.1). Therefore exactly when $f_P(x) = 1$ is there a "witness" $|w\rangle \in \mathbf{C}^I$ satisfying $A\Pi(x)|w\rangle = |t\rangle$. Exactly when $f_P(x) = 0$, there is a witness $|w'\rangle \in V$ satisfying $\langle t|w'\rangle \neq 0$ and $\Pi(x)A^\dagger|w'\rangle = 0$, i.e., $|w'\rangle$ has nonzero inner product with the target vector and is orthogonal to the available input vectors.

The complexity measure we use to characterize span programs is the witness size [RŠ08]:

Definition 3.3 (Witness size with costs [RŠ08])**.** *Consider a span program P, and a vector $s \in [0,\infty)^n$ of nonnegative "costs." Let $S = \sum_{j \in [n], b \in B, i \in I_{j,b}} \sqrt{s_j}|i\rangle\langle i|$. For each input $x \in B^n$, define the witness size of P on x with costs s, $\operatorname{wsize}_s(P,x)$, as follows:*

- *If $f_P(x) = 1$, then $|t\rangle \in \operatorname{Range}(A\Pi(x))$, so there is a witness $|w\rangle \in \mathbf{C}^I$ satisfying $A\Pi(x)|w\rangle = |t\rangle$. Then $\operatorname{wsize}_s(P,x)$ is the minimum squared length of any such witness, weighted by the costs s:*

$$\operatorname{wsize}_s(P,x) = \min_{|w\rangle :\, A\Pi(x)|w\rangle = |t\rangle} \|S|w\rangle\|^2 \ . \tag{3.4}$$

- *If $f_P(x) = 0$, then $|t\rangle \notin \operatorname{Range}(A\Pi(x))$. Therefore there is a witness $|w'\rangle \in V$ satisfying $\langle t|w'\rangle = 1$ and $\Pi(x)A^\dagger|w'\rangle = 0$. Then*

$$\operatorname{wsize}_s(P,x) = \min_{\substack{|w'\rangle :\, \langle t|w'\rangle = 1 \\ \Pi(x)A^\dagger|w'\rangle = 0}} \|SA^\dagger|w'\rangle\|^2 \ . \tag{3.5}$$

The witness size of P with costs s, restricted to domain $\mathcal{D} \subseteq B^n$, is

$$\operatorname{wsize}_s(P,\mathcal{D}) = \max_{x \in \mathcal{D}} \operatorname{wsize}_s(P,x) \ . \tag{3.6}$$

Numerous examples are given in [RŠ08, Rei09c]. For the common case that $\mathcal{D} = B^n$, let $\operatorname{wsize}_s(P) = \operatorname{wsize}_s(P, B^n)$. For $j \in [n]$, s_j can intuitively be thought of as a charge for evaluating the jth input bit. When the subscript s is omitted, the costs are taken to be uniform, $s = \vec{1} = (1,1,\ldots,1)$, e.g., $\operatorname{wsize}(P) = \operatorname{wsize}_{\vec{1}}(P)$. The extra generality of allowing nonuniform costs is necessary for considering unbalanced formulas [Rei09b, Rei09a].

Note that replacing the target vector $|t\rangle$ by $c|t\rangle$, for $c \neq 0$, changes the witness sizes by a factor of $|c|^2$ or $1/|c|^2$, depending on whether $f_P(x) = 1$ or 0. Thus we might just as well have defined the witness size as the geometric mean of $\max_{x: f_P(x)=0} \operatorname{wsize}_s(P,x)$ and $\max_{x: f_P(x)=1} \operatorname{wsize}_s(P,x)$. Explicit formulas for $\operatorname{wsize}_s(P,x)$ can be written in terms of Moore-Penrose pseudoinverses of certain matrices, and are given in [RŠ08, Lemma A.3].

Classical applications of span programs have used a different complexity measure, the "size" of P being the number of input vectors, $|I|$. This measure has been characterized in [Gál01].

Call a span program *strict* if $I_{\mathrm{free}} = \emptyset$. The original definition of span programs [KW93] considered only strict span programs. If our aim is to minimize the witness size, then we may restrict to considering strict span programs:

Proposition 3.4. *For any span program P, there exists a strict span program P' with $f_{P'} = f_P$ and $\mathrm{wsize}_s(P', x) = \mathrm{wsize}_s(P, x)$ for all $s \in [0, \infty)^n$ and $x \in B^n$.*

Proof. Construct P' by projecting P's target vector $|t\rangle$ and input vectors $\{|v_i\rangle : i \in I \smallsetminus I_{\mathrm{free}}\}$ to the space orthogonal to the span of the free input vectors. That is, let $\overline{\Delta}_{\mathrm{free}}$ be the projection onto the space orthogonal to $\mathrm{Span}(\{|v_i\rangle : i \in I_{\mathrm{free}}\})$. Then the target vector of P' is $\overline{\Delta}_{\mathrm{free}}|t\rangle$ and the input vectors are $\{\overline{\Delta}_{\mathrm{free}}|v_i\rangle : i \in I \smallsetminus I_{\mathrm{free}}\}$.

Then $f_{P'} = f_P$. Indeed, if $f_P(x) = 1$, i.e., $|t\rangle = A\Pi(x)|w\rangle$ for some witness $|w\rangle$, then $|w\rangle$ is also a witness for $f_{P'}(x) = 1$. Conversely, if $f_{P'}(x) = 1$, i.e., for some $|w\rangle$, $\overline{\Delta}_{\mathrm{free}}|t\rangle = \overline{\Delta}_{\mathrm{free}} A\Pi(x)|w\rangle$, then $|t\rangle - A\Pi(x)|w\rangle \in \mathrm{Range}(\{|v_i\rangle : i \in I_{\mathrm{free}}\})$, so $f_P(x) = 1$. Fix $s \in [0, \infty)^n$.

If $f_P(x) = 1$, then let $\Pi_{\mathrm{free}} = \sum_{i \in I_{\mathrm{free}}} |i\rangle\langle i|$ and $\Pi'(x) = \Pi(x) - \Pi_{\mathrm{free}}$. We have

$$\begin{aligned}
\mathrm{wsize}_s(P, x) &= \min_{|w\rangle : A\Pi(x)|w\rangle = |t\rangle} \left\| S\Pi'(x)|w\rangle \right\|^2 \\
&= \min_{\substack{|w\rangle : \Pi'(x)|w\rangle = |w\rangle \\ A|w\rangle - |t\rangle \in \mathrm{Range}(A\Pi_{\mathrm{free}})}} \left\| S|w\rangle \right\|^2 \\
&= \min_{|w\rangle : \overline{\Delta}_{\mathrm{free}} A\Pi'(x)|w\rangle = \overline{\Delta}_{\mathrm{free}}|t\rangle} \left\| S|w\rangle \right\|^2 = \mathrm{wsize}_s(P', x) \enspace .
\end{aligned} \tag{3.7}$$

If $f_P(x) = 0$, then

$$\begin{aligned}
\mathrm{wsize}_s(P, x) &= \min_{\substack{|w'\rangle : \langle t|w'\rangle = 1 \\ \Pi(x)A^\dagger|w'\rangle = 0}} \left\| SA^\dagger|w'\rangle \right\|^2 \\
&= \min_{\substack{|w'\rangle : \langle t|w'\rangle = 1 \\ \Pi(x)A^\dagger|w'\rangle = 0 \\ \overline{\Delta}_{\mathrm{free}}|w'\rangle = |w'\rangle}} \left\| SA^\dagger|w'\rangle \right\|^2 \\
&= \min_{\substack{|w'\rangle : \langle t|\overline{\Delta}_{\mathrm{free}}|w'\rangle = 1 \\ \Pi(x)A^\dagger\overline{\Delta}_{\mathrm{free}}|w'\rangle = 0}} \left\| SA^\dagger\overline{\Delta}_{\mathrm{free}}|w'\rangle \right\|^2 = \mathrm{wsize}_s(P', x) \enspace ,
\end{aligned} \tag{3.8}$$

where the second equality is because $\Pi(x)A^\dagger|w'\rangle = 0$ implies that $\langle v_i|w'\rangle = 0$ for all $i \in I_{\mathrm{free}}$. □

Allowing free input vectors can be convenient for defining and composing span programs, and may be necessary for developing span-program-based quantum algorithms that are *time* efficient as well as query efficient [Rei09b, Rei09a, Rei09c]. The problem for time complexity is that applying the projection $\overline{\Delta}_{\mathrm{free}}$ affects the sparsity of $|t\rangle$ and A, as well as the norm of the entry-wise absolute-value of A, and this complicates the implementation of a certain quantum walk. In this article, though, we henceforth consider only strict span programs.

In fact, we may restrict even further, and consider only "canonical" span programs [KW93]:

Definition 3.5 (Canonical span program [KW93]). *Let P be a strict span program computing $f_P : B^n \to B$, with inner product space V, target vector $|t\rangle$ and input vectors $|v_i\rangle$ for $i \in \bigsqcup_{j\in[n],b\in B} I_{j,b}$. P is canonical if:*

- *$V = \mathbf{C}^{F_0}$ where $F_0 = \{x \in B^n : f_P(x) = 0\}$. V has orthonormal basis $\{|x\rangle : x \in F_0\}$.*
- *The target is given by $|t\rangle = \sum_{x\in F_0} |x\rangle$.*
- *For all $x \in F_0$, $j \in [n]$ and $i \in I_{j,x_j}$, $\langle x|v_i\rangle = 0$; thus $\langle x|A\Pi(x) = 0$.*

Theorem 3.6. *For any cost vector $s \in [0,\infty)^n$, a span program P can be converted to a canonical span program $\hat{P}$ that computes the same function $f_{\hat{P}} = f_P$, with $\mathrm{wsize}_s(\hat{P},x) \le \mathrm{wsize}_s(P,x)$ for all $x \in B^n$. In fact, for all $x \in B^n$ with $f_P(x) = 0$, $\mathrm{wsize}_s(\hat{P},x) = \mathrm{wsize}_s(P,x)$, with $|x\rangle$ itself an optimal witness for $f_{\hat{P}}(x) = 0$.*

Proof. This theorem is analogous to [KW93, Theorem 6], and we use the same conversion procedure, except additionally analyzing the witness size.

Let P be a strict span program with target vector $|t\rangle \in V$ and input vectors $|v_i\rangle$ for $i \in I = \bigcup_{j\in[n],b\in B} I_{j,b}$. Fix $s \in [0,\infty)^n$ and let $S = \sum_{j\in[n],b\in B,i\in I_{j,b}} \sqrt{s_j}|i\rangle\langle i|$.

For $x \in B^n$, let $|w(x)\rangle$ or $|w'(x)\rangle$ be optimal witnesses for $f_P(x)$ being 1 or 0, respectively, with costs s. That is, let

$$\begin{aligned} |w(x)\rangle &= \arg\min_{|w\rangle : A\Pi(x)|w\rangle = |t\rangle} \| S|w\rangle \|^2 & \text{if } f_P(x) = 1 \\ |w'(x)\rangle &= \arg\min_{\substack{|w'\rangle:\langle t|w'\rangle = 1 \\ \Pi(x)A^\dagger|w'\rangle = 0}} \| SA^\dagger|w'\rangle \|^2 & \text{if } f_P(x) = 0 \end{aligned} \tag{3.9}$$

(See [RŠ08, Lemma A.3] for explicit formulas for $|w(x)\rangle$ and $|w'(x)\rangle$.)

Let $F_0 = \{x \in B^n : f_P(x) = 0\}$. To construct $\hat{P}$ from P, simply apply to P's target and input vectors the map $\sum_{x\in F_0} |x\rangle\langle w'(x)| \in \mathcal{L}(V, \mathbf{C}^{F_0})$. Then

- The target vector becomes $|\hat{t}\rangle = \sum_{x\in F_0} |x\rangle\langle w'(x)|t\rangle = \sum_{x\in F_0} |x\rangle \in \mathbf{C}^{F_0}$, as required for a canonical span program. The input vectors become, for $i \in I$, $|\hat{v}_i\rangle = \sum_{x\in F_0} |x\rangle\langle w'(x)|v_i\rangle$.
- For any $x \in F_0$ and $i \in \cup_{j\in[n]} I_{j,x_j}$, since $\langle w'(x)|v_i\rangle = 0$, $\langle x|\hat{v}_i\rangle = 0$.

Therefore $\hat{P}$ is a canonical span program. Let $\hat{A} = \sum_{i\in I} |\hat{v}_i\rangle\langle i| = \sum_{x\in F_0} |x\rangle\langle w'(x)|A$.

For $x \in F_0$, note that $\langle \hat{t}|x\rangle = 1$ and

$$\hat{A}^\dagger|x\rangle = A^\dagger|w'(x)\rangle \ . \tag{3.10}$$

In particular, $\Pi(x)\hat{A}^\dagger|x\rangle = 0$, so $|x\rangle$ is a witness for $f_{\hat{P}}(x) = 0$. Also, $\mathrm{wsize}_s(\hat{P},x) \le \|S\hat{A}^\dagger|x\rangle\|^2 = \|SA^\dagger|w'(x)\rangle\|^2 = \mathrm{wsize}_s(P,x)$. In fact, $|x\rangle$ is an optimal witness for $f_{\hat{P}}(x) = 0$. Indeed, assume otherwise, and let $|\hat{u}\rangle = \sum_{y\in F_0} \hat{u}_y|y\rangle$ satisfy $\langle \hat{t}|\hat{u}\rangle = \sum_{y\in F_0} \hat{u}_y = 1$, $\Pi(x)\hat{A}^\dagger|\hat{u}\rangle = 0$ and $\|S\hat{A}^\dagger|\hat{u}\rangle\|^2 < \|S\hat{A}^\dagger|x\rangle\|^2$. Let $|u\rangle = \sum_{y\in F_0} \hat{u}_y|w'(y)\rangle$, so $A^\dagger|u\rangle = \hat{A}^\dagger|\hat{u}\rangle$. Then $\langle t|u\rangle = 1$, $\Pi(x)A^\dagger|u\rangle = 0$, and $\|SA^\dagger|u\rangle\|^2 = \|S\hat{A}^\dagger|\hat{u}\rangle\|^2 < \mathrm{wsize}_s(P,x)$, a contradiction.

Next consider an $x \in B^n$ such that $f_P(x) = 1$. Then

$$\begin{aligned} \hat{A}\Pi(x)|w(x)\rangle &= \sum_{y \in F_0} |y\rangle\langle w'(y)|A\Pi(x)|w(x)\rangle \\ &= \sum_{y \in F_0} |y\rangle\langle w'(y)|t\rangle \\ &= |\hat{t}\rangle \ . \end{aligned} \tag{3.11}$$

Thus $|w(x)\rangle$ is a witness for $f_{\hat{P}}(x) = 1$, and $\mathrm{wsize}_s(\hat{P}, x) \le \| S|w(x)\rangle \|^2 = \mathrm{wsize}_s(P, x)$. □

4 Adversary lower bounds

The adversary method lower bounds the quantum query complexity of a function by a hybrid argument that considers the system's entanglement when run on a superposition of inputs [HŠ05]. The technique was introduced by Ambainis [Amb02]. A number of variants of Ambainis's bound were soon discovered, including weighted versions [HNS02, BS04, Amb06, Zha05], a spectral version [BSS03], and a version based on Kolmogorov complexity [LM04]. These variants can be asymptotically stronger than the original unweighted bound, but are equivalent to each other [ŠS06], and we denote them all by Adv.

The lower bound Adv is known to be loose for some functions, as it runs up against certificate complexity and property testing barriers. Høyer, Lee and Špalek discovered a strict generalization $\mathrm{Adv}^\pm$ of Adv [HLŠ07]. For example, for a certain four-bit function f studied in [Amb06], $\mathrm{Adv}(f^k) = 2.5^k$, whereas $\mathrm{Adv}^\pm(f^k) \ge 2.51^k$. $\mathrm{Adv}^\pm$ breaks the certificate complexity and property testing barriers, and no similar limits on its power have been found. In particular, for no function f is it known that the quantum query complexity of f is $\omega(\mathrm{Adv}^\pm(f))$.

In this section, we define the two adversary bounds. On account of how their definitions differ, we call Adv the "nonnegative-weight" adversary bound, and $\mathrm{Adv}^\pm$ the "general" adversary bound. We then derive a new dual formulation of the semi-definite program for the general adversary bound for boolean functions, and apply it to prove a composition theorem.

Definition 4.1 (Adversary bounds with costs [HLŠ05, HLŠ07])**.** *For finite sets C and E, and $\mathcal{D} \subseteq C^n$, let $f : \mathcal{D} \to E$ and let $s \in [0, \infty)^n$ be a vector of nonnegative costs. An adversary matrix for f is a nonzero, $|\mathcal{D}| \times |\mathcal{D}|$ real, symmetric matrix Γ that satisfies $\langle x|\Gamma|y\rangle = 0$ for all $x, y \in \mathcal{D}$ with $f(x) = f(y)$.*

Define the nonnegative-weight adversary bound for f, with costs s, as

$$\mathrm{Adv}_s(f) = \max_{\substack{\text{adversary matrices } \Gamma: \\ \forall x,y \in \mathcal{D},\, \langle x|\Gamma|y\rangle \ge 0 \\ \forall j \in [n],\, \|\Gamma \circ \Delta_j\| \le s_j}} \|\Gamma\| \ , \tag{4.1}$$

where $\Gamma \circ \Delta_j$ denotes the entry-wise matrix product between Γ and $\Delta_j = \sum_{x,y \in \mathcal{D}: x_j \ne y_j} |x\rangle\langle y|$, and the norm is the operator norm.

The general adversary bound for f, with costs s, is

$$\mathrm{Adv}_s^\pm(f) = \max_{\substack{\text{adversary matrices } \Gamma: \\ \forall j \in [n],\, \|\Gamma \circ \Delta_j\| \le s_j}} \|\Gamma\| \ . \tag{4.2}$$

In this maximization, the entries of Γ *need not be nonnegative. In particular,* $\mathrm{Adv}^{\pm}_s(f) \geq \mathrm{Adv}_s(f)$.

Letting $\vec{1} = (1, 1, \ldots, 1)$*, the nonnegative-weight adversary bound for* f *is* $\mathrm{Adv}(f) = \mathrm{Adv}_{\vec{1}}(f)$ *and the general adversary bound for* f *is* $\mathrm{Adv}^{\pm}(f) = \mathrm{Adv}^{\pm}_{\vec{1}}(f)$.

The adversary bounds are primarily of interest because, with uniform costs $s = \vec{1}$, they give lower bounds on quantum query complexity [BSS03, HLŠ07].

Definition 4.2. *For* $f : \mathcal{D} \to E$*, with* $\mathcal{D} \subseteq C^n$*, let* $Q_\epsilon(f)$ *be the* ϵ*-bounded-error quantum query complexity of* f*, and let* $Q(f) = Q_{1/10}(f)$.

Theorem 4.3 ([BSS03, HLŠ07])**.** *For any function* $f : \mathcal{D} \to E$*, with* $\mathcal{D} \subseteq C^n$*, the* ϵ*-bounded-error quantum query complexity of* f *is lower-bounded as*

$$\begin{aligned} Q_\epsilon(f) &\geq \frac{1 - 2\sqrt{\epsilon(1-\epsilon)}}{2} \mathrm{Adv}(f) \\ Q_\epsilon(f) &\geq \frac{1 - 2\sqrt{\epsilon(1-\epsilon)} - 2\epsilon}{2} \mathrm{Adv}^{\pm}(f) \enspace . \end{aligned} \tag{4.3}$$

In particular, $Q(f) = \Omega(\mathrm{Adv}^{\pm}(f))$*. Moreover, if* $D = \{0, 1\}$*, then*

$$Q_\epsilon(f) \geq \frac{1 - 2\sqrt{\epsilon(1-\epsilon)}}{2} \mathrm{Adv}^{\pm}(f) \enspace . \tag{4.4}$$

Let us now prove a dual formulation of Eq. (4.2) for $\mathrm{Adv}^{\pm}_s(f)$, that holds when the input alphabet is binary or the codomain is boolean.

Theorem 4.4. *For finite sets* $\mathcal{D} \subseteq C^n$*, and* E*, let* $f : \mathcal{D} \to E$*, and let* $s \in [0, \infty)^n$ *be a vector of nonnegative costs. If either* $C = \{0, 1\}$ *or* $E = \{0, 1\}$*, then the general adversary bound for* f*, with costs* s*, equals*

$$\mathrm{Adv}^{\pm}_s(f) = \min_{\substack{X \succeq 0: \\ \forall (x,y) \in F, \sum_{j \in [n] : x_j \neq y_j} \langle x, j | X | y, j \rangle = 1}} \max_{x \in \mathcal{D}} \sum_{j \in [n]} s_j \langle x, j | X | x, j \rangle \enspace . \tag{4.5}$$

Here X *is required to be a positive semi-definite,* $(n|\mathcal{D}|) \times (n|\mathcal{D}|)$ *matrix, with coordinates labeled by* $\mathcal{D} \times [n]$*, and* $F = \{(x, y) \in \mathcal{D} \times \mathcal{D} : f(x) \neq f(y)\}$*. The optimum is achieved.*

Proof. The proof is by a standard application of duality theory to the semi-definite program (SDP) given in Definition 4.1. Nonetheless, this expression for $\mathrm{Adv}^{\pm}_s(f)$ is new, and is somewhat simpler than the expression that was known before, Eq. (4.9) below. Therefore we include a proof, based on the following immediate observation:

Claim 4.5. *Let* $M = \sum_{j,k \in [m]} M_{jk} |j\rangle\langle k| \in \mathcal{L}(\mathbf{C}^{[m]})$ *be an* $m \times m$ *Hermitian matrix. Assume that either* M *is entry-wise nonnegative, i.e.,* $M_{jk} \geq 0$ *for all* $j, k \in [m]$*, or that* M *is bipartite, i.e., for some* $l \in [m-1]$*,* $M = \sum_{j \leq l, k > l} (M_{jk} |j\rangle\langle k| + M_{kj} |k\rangle\langle j|)$*. Then* $M \preceq \mathbf{1}$ *if and only if* $\|M\| \leq 1$.

Taking the dual of the SDP on the right-hand side of Eq. (4.5), we obtain

$$\max_{\substack{\tilde{\Gamma} = \sum_F \alpha_{xy} |x\rangle\langle y| \\ \{\beta_x \geq 0\}}} \sum_F \alpha_{xy} \quad \text{such that} \quad \sum_x \beta_x \leq 1, \ \ \forall j, \ \tilde{\Gamma}_j \preceq s_j \sum_x \beta_x |x\rangle\langle x| \enspace . \tag{4.6}$$

Here $\tilde{\Gamma}_j = \tilde{\Gamma} \circ \Delta_j = \sum_{x,y \in \mathcal{D}: x_j \neq y_j} |x\rangle\langle x|\tilde{\Gamma}|y\rangle\langle y|$ as in Definition 4.1. Also, $\tilde{\Gamma}_j \preceq s_j \sum_x \beta_x |x\rangle\langle x|$ means that the difference $(s_j \sum_{x \in \mathcal{D}} \beta_x |x\rangle\langle x|) - \tilde{\Gamma}_j$ is a positive semi-definite matrix. In particular, this constraint implies that if $s_j = 0$ then $\alpha_{xy} = 0$ for all x, y with $x_j \neq y_j$; and that if $\alpha_{xy} \neq 0$, then $\beta_x > 0$ and $\beta_y > 0$.

Thus we can change variables, letting $\Gamma = \sum_{(x,y) \in \Delta: \alpha_{xy} \neq 0} \frac{\alpha_{xy}}{\sqrt{\beta_x \beta_y}} |x\rangle\langle y|$. Like $\tilde{\Gamma}$, Γ can vary over the set of adversary matrices, i.e., symmetric matrices supported only on those $|x\rangle\langle y|$ with $f(x) \neq f(y)$. The objective function becomes $\sum_F \langle x|\Gamma|y\rangle \sqrt{\beta_x \beta_y}$, and, for $j \in [n]$, the constraint on $\tilde{\Gamma}_j$ becomes $\Gamma_j \preceq s_j \mathbf{1}$, where $\Gamma_j = \Gamma \circ \Delta_j$.

Now if $C = \{0,1\}$, then the matrices Δ_j are bipartite—perhaps in a permuted basis—so each Γ_j is also bipartite. If $E = \{0,1\}$, then Γ is bipartite since it is supported only on F. In either case, by Claim 4.5 the condition $\Gamma_j \preceq s_j \mathbf{1}$ is equivalent to $\|\Gamma_j\| \leq s_j$. Therefore, after changing variables, the SDP becomes

$$\max_{\substack{\text{adversary matrices } \Gamma \\ \{\beta_x \geq 0\}}} \sum_F \langle x|\Gamma|y\rangle \sqrt{\beta_x \beta_y} \quad \text{such that} \quad \sum_x \beta_x \leq 1, \;\; \forall j, \; \|\Gamma_j\| \leq s_j \;\; . \tag{4.7}$$

Since any negative signs on the coordinates of the principal eigenvector of Γ can be absorbed into the matrix, without affecting the norms of the Γ_j, the objective function in Eq. (4.7) simplifies to $\|\Gamma\|$, so we obtain $\mathrm{Adv}^{\pm}(f)$. Since the dual SDP in Eq. (4.6) is clearly strictly feasible, by the duality principle [Lov03, Theorem 3.4] the primal optimum equals the dual optimum and the primal optimum is achieved. Eq. (4.5) follows. $\square$

For completeness, we state without proof the dual forms of the adversary bounds for the case of functions without a binary input alphabet or boolean codomain:

Theorem 4.6. *For finite sets $\mathcal{D} \subseteq C^n$, and E, let $f : \mathcal{D} \to E$, and let $s \in [0, \infty)^n$. Let $F = \sum_{x,y \in \mathcal{D}: f(x) \neq f(y)} |x\rangle\langle y|$. As in Definition 4.1, let $\Delta_j = \sum_{x,y \in \mathcal{D}: x_j \neq y_j} |x\rangle\langle y|$ for $j \in [n]$, and let $\circ$ denote entry-wise matrix multiplication.*

Then the nonnegative-weight adversary bound for f, with costs s, equals

$$\mathrm{Adv}_s(f) = \min_{\substack{X_j \succeq 0: \\ \sum_j X_j \circ \Delta_j \circ F \geq F}} \max_{x \in \mathcal{D}} \sum_{j \in [n]} s_j \langle x|X_j|x\rangle \;\; . \tag{4.8}$$

The minimization is over $|\mathcal{D}| \times |\mathcal{D}|$ positive semi-definite matrices X_j, $j \in [n]$, that satisfy the entry-wise inequality $\sum_j X_j \circ \Delta_j \circ F \geq F$. (Note that Eq. (4.5) has the same form, except with the requirement that $\sum_j X_j \circ \Delta_j \circ F = F$.)

The general adversary bound for f, with costs s, equals

$$\mathrm{Adv}_s^{\pm}(f) = \min_{\substack{X_j, Y_j \succeq 0: \\ \sum_j (X_j - Y_j) \circ \Delta_j \circ F = F}} \max_{x \in \mathcal{D}} \sum_{j \in [n]} s_j \langle x|(X_j + Y_j)|x\rangle \;\; . \tag{4.9}$$

For boolean functions, the nonnegative-weight adversary bound composes multiplicatively, but this was not known to hold for the general adversary bound [HLŠ07]. Theorem 4.4 allows us to prove that the general adversary bound composes in the same way.

Theorem 4.7 (Adversary bound composition [HLŠ07, Amb06, LLS06, HLŠ05])**.** *Let $f : \{0,1\}^n \to \{0,1\}$ and, for $j \in [n]$, let $f_j : \{0,1\}^{m_j} \to \{0,1\}$. Define $g : \{0,1\}^{m_1} \times \cdots \times \{0,1\}^{m_n} \to \{0,1\}$ by*

$$g(x) = f\big(f_1(x_1), \ldots, f_n(x_n)\big) \ . \tag{4.10}$$

Let $s \in [0,\infty)^{m_1} \times \cdots \times [0,\infty)^{m_n}$, and let $\alpha_j = \mathrm{Adv}_{s_j}(f_j)$ and $\beta_j = \mathrm{Adv}^{\pm}_{s_j}(f_j)$ for $j \in [n]$. Then

$$\mathrm{Adv}_s(g) = \mathrm{Adv}_\alpha(f) \tag{4.11}$$
$$\mathrm{Adv}^{\pm}_s(g) \geq \mathrm{Adv}^{\pm}_\beta(f) \ . \tag{4.12}$$

In particular, if $\mathrm{Adv}_{s_1}(f_1) = \cdots = \mathrm{Adv}_{s_n}(f_n) = \alpha$, then $\mathrm{Adv}_s(g) = \alpha \,\mathrm{Adv}(f)$.

Theorem 4.8 (General adversary bound composition)**.** *Under the conditions of Theorem 4.7,*

$$\mathrm{Adv}^{\pm}_s(g) = \mathrm{Adv}^{\pm}_\beta(f) \ . \tag{4.13}$$

In particular, if $\mathrm{Adv}^{\pm}_{s_1}(f_1) = \cdots = \mathrm{Adv}^{\pm}_{s_n}(f_n) = \beta$, then $\mathrm{Adv}^{\pm}_s(g) = \beta \,\mathrm{Adv}^{\pm}(f)$.

Proof. We aim to show that Eq. (4.12) is tight. Recall that $B = \{0,1\}$. For $x \in B^{m_1} \times \cdots \times B^{m_n}$, let $y(x) = (f_1(x), \ldots, f_n(x))$, so $g(x) = f(y(x))$.

For $y \in B^n$ and $j \in [n]$, fix vectors $|v_{yj}\rangle \in V$ that achieve $\mathrm{Adv}^{\pm}_\beta(f)$, i.e., $\sum_{j : y_j \neq y_{j'}} \langle v_{yj} | v_{y'j} \rangle = 1$ for all $y, y' \in B^n$ with $f(y) \neq f(y')$, and $\mathrm{Adv}^{\pm}_\beta(f) = \max_{y \in B^n} \sum_{j \in [n]} \beta_j \| |v_{yj}\rangle \|^2$. For $j \in [n]$, fix vectors $|v^j_{zk}\rangle \in V^j$ for $z \in B^{m_j}$, $k \in [m_j]$, that achieve $\mathrm{Adv}^{\pm}_{s_j}(f_j)$, i.e., $\sum_{k : z_k \neq z'_k} \langle v^j_{zk} | v^j_{z'k} \rangle = 1$ for all $z, z' \in B^{m_j}$ with $f_j(z) \neq f_j(z')$.

Based on these solutions, we construct a feasible solution for the dual formulation of $\mathrm{Adv}^{\pm}_s(g)$. For $x \in B^{m_1} \times \cdots \times B^{m_n}$, $j \in [n]$ and $k \in [m_j]$, let

$$|w_{xjk}\rangle = |v_{y(x)j}\rangle \otimes |v^j_{x_j k}\rangle \otimes |\delta_{g(x), f_j(x_j)}\rangle \in V \otimes (\oplus_{j \in n} V^j) \otimes \mathbf{C}^2 \ . \tag{4.14}$$

Here, the third register is spanned by the orthonormal basis $\{|0\rangle, |1\rangle\}$, and $\delta_{a,b}$ is 1 if $a = b$ and 0 otherwise.

Consider $x, x' \in B^{m_1} \times \cdots \times B^{m_n}$ such that $g(x) \neq g(x')$. In particular, $y(x) \neq y(x')$. Then

$$\begin{aligned}
\sum_{\substack{j \in [n], k \in [m_j]: \\ x_{jk} \neq x'_{jk}}} \langle w_{xjk} | w_{x'jk} \rangle &= \sum_{j \in [n]} \langle v_{y(x)j} | v_{y(x')j} \rangle \sum_{k \in [m_j] : x_{jk} \neq x'_{jk}} \langle v^j_{x_j k} | v^j_{x'_j k} \rangle (1 - \delta_{f_j(x_j), f_j(x'_j)}) \\
&= \sum_{j \in [n] : y(x)_j \neq y(x')_j} \langle v_{y(x)j} | v_{y(x')j} \rangle \sum_{k \in [m_j] : x_{jk} \neq x'_{jk}} \langle v^j_{x_j k} | v^j_{x'_j k} \rangle \\
&= \sum_{j \in [n] : y(x)_j \neq y(x')_j} \langle v_{y(x)j} | v_{y(x')j} \rangle \\
&= 1 \ .
\end{aligned} \tag{4.15}$$

Hence indeed the vectors $|w_{xjk}\rangle$ give a feasible solution. We conclude that

$$\begin{aligned}
\mathrm{Adv}_s^{\pm}(g) &\leq \max_{x \in B^{m_1} \times \cdots \times B^{m_n}} \sum_{j \in [n], k \in [m_j]} s_{jk} \| |w_{xjk}\rangle \|^2 \\
&= \max_x \sum_{j \in [n]} \| |v_{y(x)j}\rangle \|^2 \sum_{k \in [m_j]} s_{jk} \| |v^j_{x_j k}\rangle \|^2 \\
&\leq \max_x \sum_{j \in [n]} \beta_j \| |v_{y(x)j}\rangle \|^2 \\
&= \mathrm{Adv}_\beta^{\pm}(f) \ .
\end{aligned} \tag{4.16}$$

The last step is clearly an inequality, which is all that is needed to finish the proof. It is in fact an equality, though, because $y(x)$ varies over all strings in B^n as x varies over $B^{m_1} \times \cdots \times B^{m_n}$. $\square$

Finally, Theorem 4.7 allows connecting span program witness size to the adversary lower bounds:

Theorem 4.9 ([RŠ08])**.** *For any span program P computing $f_P : \{0,1\}^n \to \{0,1\}$,*

$$\mathrm{wsize}(P) \geq \mathrm{Adv}^{\pm}(f_P) \geq \mathrm{Adv}(f_P) \ . \tag{4.17}$$

There is a direct proof that $\mathrm{wsize}(P) \geq \mathrm{Adv}(f_P)$ in [RŠ08, Sec. 5.3], but the inequality $\mathrm{wsize}(P) \geq \mathrm{Adv}^{\pm}(f_P)$ is only implicit in [RŠ08]. The argument is as follows. Letting $f^k : \{0,1\}^{n^k} \to \{0,1\}$ be the k-times-iterated composition of f on itself, $Q(f_P^k) = O_k(\mathrm{wsize}(P)^k)$ by Theorem 1.1. Now by Theorem 4.7, $\mathrm{Adv}^{\pm}(f)^k \leq \mathrm{Adv}^{\pm}(f^k) = O(Q(f_P^k))$. Putting these results together and letting $k \to \infty$ gives $\mathrm{Adv}^{\pm}(f) \leq \mathrm{wsize}(P)$. A full and direct proof will be given below in Theorem 5.1.

5 Span program witness size and the general adversary bound

In this section, we will use Theorem 3.6 to formulate a semi-definite program for the optimal span program computing a boolean function f. Remarkably, the optimal span program witness size is exactly equal to the general adversary bound. This result has several corollaries, in quantum algorithms and in complexity theory, that we give in Section 6.

Theorem 5.1. *For any function $f : \mathcal{D} \to B$, with $\mathcal{D} \subseteq B^n$, and any cost vector $s \in [0,\infty)^n$,*

$$\inf_{P :\, f_P|_{\mathcal{D}} = f} \mathrm{wsize}_s(P, \mathcal{D}) = \mathrm{Adv}_s^{\pm}(f) \ , \tag{5.1}$$

where the infimum is over span programs P that compute a function agreeing with f on $\mathcal{D}$. Moreover, this infimum is achieved.

Proof. Lemma 5.2 constructs an SDP whose solution is the optimal witness size of a span program computing f.

Lemma 5.2. *Let $f : \mathcal{D} \to B$, with $\mathcal{D} \subseteq B^n$, be a partial boolean function. For $b \in B$, let $F_b = \{x \in \mathcal{D} : f(x) = b\}$. Then for any cost vector $s \in [0,\infty)^n$,*

$$\inf_{P :\, f_P|_{\mathcal{D}} = f} \mathrm{wsize}_s(P, \mathcal{D}) = \inf_{\substack{m \in \mathbf{N}, \\ \{|v_{xj}\rangle \in \mathbf{C}^m :\, x \in \mathcal{D}, j \in [n]\} : \\ \forall (x,y) \in F_0 \times F_1,\, \sum_{j \in [n] : x_j \neq y_j} \langle v_{xj} | v_{yj} \rangle = 1}} \max_{x \in \mathcal{D}} \sum_{j \in [n]} s_j \| |v_{xj}\rangle \|^2 \ . \tag{5.2}$$

Proof. The proof is by establishing a correspondence between solutions to the constraints on the right-hand side of Eq. (5.2) and canonical span programs computing $f_P|_{\mathcal{D}} = f$ with $\max_{j\in[n],b\in B} |I_{j,b}| \leq m$.

First let us prove the $\leq$ direction. Given a solution $\{|v_{xj}\rangle\}$, let P be a span program with target $|t\rangle = \sum_{x\in F_0} |x\rangle \in \mathbf{R}^{F_0}$ and $I_{j,b} = [m]$ for all $j \in [n]$, $b \in B$. These sets are not disjoint, so for $k \in I_{j,b}$, use $|v_{jbk}\rangle$ to denote the corresponding input vector, defined by $|v_{jbk}\rangle = \sum_{x\in F_0 : x_j \neq b} \langle v_{xj}|k\rangle |x\rangle$. Thus

$$\begin{aligned} A :=& \sum_{j\in[n],b\in B,k\in[m]} |v_{jbk}\rangle\langle j,b,k| \\ =& \sum_{x\in F_0, j\in[n]} |x\rangle\langle j,\bar{x}_j| \otimes \langle v_{xj}| \enspace . \end{aligned} \tag{5.3}$$

For $x \in F_0$, $|w'\rangle = |x\rangle$ is a witness for $f_P(x) = 0$; $\langle x|t\rangle = 1$ but $\langle x|v_{jx_jk}\rangle = 0$ for all j, k. The witness size is $\| A^\dagger |x\rangle \|^2 = \sum_j s_j \| |v_{xj}\rangle \|^2$.

For $x \in F_1$, let $|w\rangle = \sum_j |j, x_j\rangle \otimes |v_{xj}\rangle$. The condition that $\sum_{j:x_j\neq y_j} \langle v_{yj}|v_{xj}\rangle = 1$ implies that $|w\rangle$ is a witness, $A\Pi(x)|w\rangle = A|w\rangle = |t\rangle$, so $f_P(x) = 1$. The witness size is $\| |w\rangle \|^2 = \sum_j s_j \| |v_{xj}\rangle \|^2$.

Thus $f_P|_{\mathcal{D}} = f$ and $\mathrm{wsize}_s(P, \mathcal{D}) \leq \max_x \sum_j s_j \| |v_{xj}\rangle \|^2$.

Now let us prove the $\geq$ direction. Let P be a span program computing f_P, with $f_P|_{\mathcal{D}} = f$. By Theorem 3.6, we may assume that P is in canonical form, and that for each $x \in F_0$, $|x\rangle$ is an optimal witness for $f_P(x) = 0$: $\mathrm{wsize}_s(P, x) = \| S A^\dagger |x\rangle \|^2$.

Thus the target vector $|t\rangle = \sum_{x\in F_0} |x\rangle$ and the input vectors lie in the inner product space $\mathbf{C}^{F_0}$. Let $m = \max_{j\in[n],b\in B} |I_{j,b}|$. Without loss of generality, we may assume that $|I_{j,b}| = m$ for all $j \in [n]$ and $b \in B$; if some index set $I_{j,b}$ is smaller, then we can pad it with zero vectors without affecting the witness size. Therefore, let $I_{j,b} = [m]$ for all $j \in [n]$ and $b \in B$. These sets are not disjoint, so for $k \in I_{j,b}$, use $|v_{jbk}\rangle$ to denote the corresponding input vector.

For $x \in F_0$, note that since the span program is canonical, $\langle x|v_{jx_jk}\rangle = 0$ for all $j \in [n]$ and $k \in [m]$. For $j \in [n]$, let $|v_{xj}\rangle = \sum_{k\in[m]} \langle v_{j\bar{x}_jk}|x\rangle |k\rangle$. Then Eq. (5.3) again holds. Moreover, $\mathrm{wsize}_s(P, x) = \| S A^\dagger |x\rangle \|^2 = \sum_{j\in[n]} s_j \| |v_{xj}\rangle \|^2$. Thus $\max_{x\in F_0} \sum_j s_j \| |v_{xj}\rangle \|^2 \leq \mathrm{wsize}_s(P, \mathcal{D})$.

For $x \in F_1$, on the other hand, let $|w_x\rangle$ be an optimal witness vector, i.e., satisfying $|w_x\rangle = \Pi(x)|w_x\rangle = \sum_{j\in[n],k\in[m]} |j, x_j, k\rangle\langle j, x_j, k|w_x\rangle$, $A|w_x\rangle = |t\rangle$ and $\mathrm{wsize}_s(P, x) = \| S|w_x\rangle \|^2$. For $j \in [n]$, let $|v_{xj}\rangle = \sum_{k\in[m]} |k\rangle\langle j, x_j, k|w_x\rangle$. Then

$$A|w_x\rangle = |t\rangle \qquad \Longrightarrow \qquad \forall\, y \in F_0, \sum_{j:x_j\neq y_j} \langle v_{yj}|v_{xj}\rangle = 1 \enspace . \tag{5.4}$$

Finally, $\mathrm{wsize}_s(P, x) = \sum_j s_j \| |v_{xj}\rangle \|^2$, so $\max_{x\in F_1} \sum_j s_j \| |v_{xj}\rangle \|^2 \leq \mathrm{wsize}_s(P, \mathcal{D})$. $\square$

Now the expression on the right-hand side of Eq. (5.1) is just the Cholesky decomposition of the solution to the SDP in Eq. (4.5). We conclude that $\inf_{P: f_P|_{\mathcal{D}}=f} \mathrm{wsize}_s(P) = \mathrm{Adv}^\pm_s(f)$, as claimed. $\square$

This result may be somewhat surprising, because the optimal span programs known previously were all for functions f with $\mathrm{Adv}(f) = \mathrm{Adv}^\pm(f)$ [RŠ08]. It is not clear why earlier attempts to find optimal span programs did not succeed for any function f with $\mathrm{Adv}(f) < \mathrm{Adv}^\pm(f)$.

Before stating some corollaries of Theorem 5.1, let us make a remark on the proof:

Lemma 5.3. *For a function $f : \mathcal{D} \to B$, with $\mathcal{D} \subseteq B^n$, assume that there is a rank-k optimal solution X to Eq. (4.5) for $\mathrm{Adv}^{\pm}(f)$. Note that $k \leq n|\mathcal{D}| \leq n2^n$. Then by the proof of Lemma 5.2 there is an optimal span program computing f with $|I_{j,b}| = k$ for all $j \in [n]$ and $b \in B$.*

[HLŠ07, Theorem 18] states in particular that Eq. (4.8) always has a rank-one optimal solution. The proof takes the Cholesky decomposition of a solution $X = \sum_{x,y,j,j'} |x,j\rangle\langle v_{xj}|v_{yj'}\rangle\langle y,j'|$, and replaces each vector $|v_{xj}\rangle$ by the scalar $\||v_{xj}\rangle\|$. That is, let $X' = \sum_{x,y,j,j'} \||v_{xj}\rangle\|\||v_{yj'}\rangle\||x,j\rangle\langle y,j'|$, a rank-one matrix. Then by the Cauchy-Schwarz inequality, $\langle x,j|X'|y,j\rangle \geq \langle x,j|X|y,j\rangle$, with equality when $y = x$, so X' is as good a solution to Eq. (4.8) as X is. However, note that even when $\mathrm{Adv}_s(f) = \mathrm{Adv}_s^{\pm}(f)$, this argument does not imply that Eq. (4.5) has a rank-one optimal solution [Špa09].

6 Consequences of the SDP for optimal witness size

This section will state several corollaries of Theorem 5.1.

Theorem 6.1. *For any function $f : \mathcal{D} \to \{0,1\}$, with $\mathcal{D} \subseteq \{0,1\}^n$, there exists a span program P computing $f_P|_{\mathcal{D}} = f$ with witness size upper-bounded by the bounded-error quantum query complexity of f,*

$$\mathrm{wsize}(P, \mathcal{D}) = O(Q(f)) \ . \tag{6.1}$$

Proof. By Theorem 4.3, the quantum query complexity of f is lower-bounded by the general adversary bound for f, which by Theorem 5.1 equals the best span program witness size:

$$\begin{aligned} Q(f) &= \Omega(\mathrm{Adv}^{\pm}(f)) \qquad (6.2)\\ &= \Omega\big(\inf_{P : f_P|_{\mathcal{D}} = f} \mathrm{wsize}(P, \mathcal{D})\big) \ . \end{aligned}$$

□

It is an interesting problem to prove Theorem 6.1 based directly on a quantum query algorithm that evaluates f, as in the proof of [Rei09c, Theorem 3.1] for the one-sided error case.

By substituting Theorem 5.1 into Theorem 1.1, we obtain an exact asymptotic expression for the quantum query complexity of a boolean function f composed on itself.

Theorem 6.2. *For any function $f : \{0,1\}^n \to \{0,1\}$, define $f^k : \{0,1\}^{n^k} \to \{0,1\}$ as the function f composed on itself repeatedly to a depth of k, as in Theorem 1.1. Then*

$$\lim_{k\to\infty} Q(f^k)^{1/k} = \mathrm{Adv}^{\pm}(f) \ . \tag{6.3}$$

Proof. By Theorems 4.3 and 4.7, $Q(f^k) = \Omega(\mathrm{Adv}^{\pm}(f^k)) = \Omega(\mathrm{Adv}^{\pm}(f)^k)$. Hence $\liminf_{k\to\infty} Q(f^k)^{1/k} \geq \mathrm{Adv}^{\pm}(f)$. Theorem 5.1 together with the formula-evaluation algorithm Theorem 1.1 implies $Q(f^k) = O_k(\mathrm{Adv}^{\pm}(f)^k)$. Hence $\limsup_{k\to\infty} Q(f^k)^{1/k} \leq \mathrm{Adv}^{\pm}(f)$. □

Theorem 6.2 implies a new asymptotic upper bound on the sign degree of a boolean function f.

Definition 6.3. *The* sign degree *of a function $f : \{0,1\}^n \to \{0,1\}$ is the least degree of a real multivariate polynomial $p(x_1, \ldots, x_n)$ such that for all $x \in \{0,1\}^n$, $p(x) \geq 0$ if and only if $f(x) = 1$.*

Corollary 6.4 ([Lee09])**.** *For any function $f : \{0,1\}^n \to \{0,1\}$,*

$$\text{sign-degree}(f) \leq \mathrm{Adv}^{\pm}(f) \ . \tag{6.4}$$

Proof. By the polynomial method [BBC$^+$01, NC00], sign-degree$(f) \leq 2Q(f)$. Thus

$$\limsup_{k\to\infty} \text{sign-degree}(f^k)^{1/k} \leq \lim_{k\to\infty} Q(f^k)^{1/k} = \mathrm{Adv}^\pm(f) \ . \tag{6.5}$$

Now use sign-degree$(f)^k \leq$ sign-degree(f^k) [Lee09]. □

Corollary 6.4 is a classical statement, with positive consequences in classical learning theory; an n-bit function with sign-degree d can be learned, in several models, in time $2^{\tilde{O}(d)}$ [KS01, KOS04]. However, its proof uses quantum algorithms [DW09]. A more direct proof is not known.

For example, if f is a read-once AND-OR formula on n variables, $\mathrm{Adv}(f) = \mathrm{Adv}^\pm(f) = \sqrt{n}$ [BS04]. Indeed, these bounds can be computed by showing $\mathrm{Adv}_s(\mathrm{AND}_m) = \mathrm{Adv}_s^\pm(\mathrm{AND}_m) = \sqrt{\sum_{j\in[m]} s_j^2}$, where AND_m denotes the AND gate on m variables, and then using Theorems 4.7 and 4.8 to compose the adversary bounds. Therefore sign-degree$(f) \leq \sqrt{n}$. This inequality is tight for some n, and resolves an open problem posed in [OS03]. The best previous upper bound was sign-degree$(f) = n^{1/2+o(1)}$ based on a quantum algorithm [ACR$^+$07]. Of course, Eq. (6.4) can be loose, for example for disjunctive normal form (DNF) formulas [KS01]. Given Eq. (1.4), this is not surprising, as $2Q(f)$ in fact upper-bounds the approximate polynomial degree, which lies above the sign degree.

Theorem 5.1 is very useful for developing quantum algorithms for evaluating formulas. Theorem 1.1 is only a special case of the formula-evaluation result from [RŠ08]. That article's main result can also be extended. Ref. [RŠ08] used the nonnegative-weight adversary bound Adv instead of the general adversary bound $\mathrm{Adv}^\pm$ throughout, because only for functions f with $\mathrm{Adv}(f) = \mathrm{Adv}^\pm(f)$ had matching span programs been found. Theorem 5.1, however, gives optimal span programs for every boolean function f. Thus we can simply modify [RŠ08, Def. 4.5], defining adversary-balanced formulas, to refer to $\mathrm{Adv}^\pm$ instead of Adv. That is, a formula is adversary-balanced if at every gate the input subformulas have equal general adversary bounds. Letting $\mathcal{S}$ be any finite gate set of boolean functions, [RŠ08, Theorem 4.7] becomes:

Theorem 6.5. *There exists a quantum algorithm that evaluates an adversary-balanced formula $\varphi(x)$ over $\mathcal{S}$ using $O(\mathrm{Adv}^\pm(\varphi))$ input queries. After efficient classical preprocessing independent of the input x, and assuming $O(1)$-time coherent access to the preprocessed classical string, the running time of the algorithm is $\mathrm{Adv}^\pm(\varphi)(\log \mathrm{Adv}^\pm(\varphi))^{O(1)}$.*

Aside from changing Adv to $\mathrm{Adv}^\pm$, the proof from [RŠ08] is unchanged. Note that layered formulas, in which gates at the same depth are the same, are a special case of adversary-balanced formulas.

Notice that Theorem 6.5 is superior to the algorithm behind Eq. (1.4), from [Rei10], because Theorem 6.5 bounds *time* complexity as well as query complexity. In fact, though, the more general analysis of quantum algorithms for evaluating span programs in [Rei09c] is superior to the analysis in [RŠ08] even for the special case of evaluating formulas. Using this improved analysis, Theorem 6.5 has been generalized to hold even for "almost-balanced" formulas, for which the input subformulas of a gate are allowed to have general adversary bounds that differ by a constant factor [Rei09b].

Finally, combined with Theorem 4.8 and simple monotonicity arguments for the witness size [RŠ08, Remark A.5], Theorem 5.1 gives a composition result for span programs:

Corollary 6.6 (Span program composition)**.** *Under the conditions of Theorem 4.7, let P be a span program computing $f_P = f$ and, for $j \in [n]$, let P_j be a span program computing $f_{P_j} = f_j$. Then* *there exists a span program Q computing the composed function $f_Q = g$, and such that, for any $s \in [0,\infty)^{m_1} \times \cdots \times [0,\infty)^{m_n}$, with $\beta_j = \mathrm{wsize}_{s_j}(P_j)$,*

$$\mathrm{wsize}_s(Q) \leq \mathrm{wsize}_\beta(P) \ . \tag{6.6}$$

In particular, $\mathrm{wsize}_s(Q) \leq \mathrm{wsize}(P) \max_{j\in[n]} \mathrm{wsize}_{s_j}(P_j)$*.*

This corollary should not be emphasized, though, as there is a direct and explicit composition theorem for span programs [Rei09c, Theorem 4.3]. In fact, Theorem 4.8 was first proved indirectly, by combining the span program composition theorem with Theorem 5.1 (see [Rei09c, Theorem 7.2]).

7 Conclusion

We have shown that for any boolean function f, the general adversary bound $\mathrm{Adv}^\pm(f)$ exactly characterizes the optimal span program witness size. As discussed below Theorem 6.5, this and subsequent work [Rei09b] has largely resolved the problem of evaluating formulas quantumly, with optimal query complexity and near-optimal time complexity, except for formulas including gates of unbounded size or very unbalanced formulas. The formula-evaluation algorithms exploit the ease of composing span programs. Span programs, and the semi-definite program for finding optimal span programs, may also be useful for developing other quantum algorithms.

Another open question is to determine the relationship, if any, between span programs and quantum query algorithms for non-boolean functions, and especially for functions with a non-binary input alphabet. Of course an input in $[k]^n$ can always be encoded into binary, $\{0,1\}^{n\lceil \log_2 k\rceil}$, and then a span program built from the general adversary bound SDP. However, this encoding might increase $\mathrm{Adv}^\pm$ significantly, possibly by more than the expected $O(\log k)$ factor. Neither adversary bound is known to be stable under different encodings of the input. A natural and more direct approach is to define generalized canonical span programs and extend Lemma 5.2 to characterize an optimal generalized witness size. Although this may lead to new quantum query algorithms, it will likely be insufficient for obtaining provably optimal or near-optimal algorithms, since the SDP on the right-hand side of Eq. (4.5) is greater than $\mathrm{Adv}^\pm$ in general; see Eq. (4.9). Moreover, there are surjective functions $[3]^2 \to [3]$ for which both $\mathrm{Adv}^\pm$ and the SDP in Eq. (4.5) compose strictly sub-multiplicatively. This indicates, not surprisingly, that the formula-evaluation problem with non-boolean gate sets may be much complicated.

Acknowledgements

I thank Troy Lee and Robert Špalek for many helpful discussions and for feedback on early versions of this work. Research supported in part by NSERC and ARO.

References

[ACR+07] Andris Ambainis, Andrew M. Childs, Ben W. Reichardt, Robert Špalek, and Shengyu Zhang. Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. In *Proc. 48th IEEE FOCS*, pages 363–372, 2007.

[Amb02] Andris Ambainis. Quantum lower bounds by quantum arguments. *J. Comput. Syst. Sci.*, 64:750–767, 2002, arXiv:quant-ph/0002066. Earlier version in STOC'00.

[Amb06] Andris Ambainis. Polynomial degree vs. quantum query complexity. *J. Comput. Syst. Sci.*, 72(2):220–238, 2006, arXiv:quant-ph/0305028. Earlier version in FOCS'03.

[BBC$^+$01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001, arXiv:quant-ph/9802049.

[BS04] Howard Barnum and Michael Saks. A lower bound on the quantum query complexity of read-once functions. *J. Comput. Syst. Sci.*, 69(2):244–258, 2004, arXiv:quant-ph/0201007.

[BSS03] Howard Barnum, Michael Saks, and Mario Szegedy. Quantum decision trees and semidefinite programming. In *Proc. 18th IEEE Complexity*, pages 179–193, 2003.

[CCJY07] Andrew M. Childs, Richard Cleve, Stephen P. Jordan, and David Yeung. Discrete-query quantum algorithm for NAND trees. 2007, arXiv:quant-ph/0702160.

[DW09] Andrew Drucker and Ronald de Wolf. Quantum proofs for classical theorems. 2009, arXiv:0910.3376 [quant-ph].

[FGG08] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum algorithm for the Hamiltonian NAND tree. *Theory of Computing*, 4:169–190, 2008, arXiv:quant-ph/0702144.

[Gál01] Anna Gál. A characterization of span program size and improved lower bounds for monotone span programs. *Computational Complexity*, 10:277–296, 2001.

[GP03] Anna Gál and Pavel Pudlák. A note on monotone complexity and the rank of matrices. *Information Processing Letters*, 87(6):321–326, 2003.

[HLŠ05] Peter Høyer, Troy Lee, and Robert Špalek. Tight adversary bounds for composite functions. 2005, arXiv:quant-ph/0509067.

[HLŠ07] Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proc. 39th ACM STOC*, pages 526–535, 2007, arXiv:quant-ph/0611054.

[HNS02] Peter Høyer, Jan Neerbek, and Yaoyun Shi. Quantum complexities of ordered searching, sorting, and element distinctness. *Algorithmica*, 34(4):429–448, 2002, arXiv:quant-ph/0102078. Special issue on Quantum Computation and Cryptography.

[HŠ05] Peter Høyer and Robert Špalek. Lower bounds on quantum query complexity. *EATCS Bulletin*, 87:78–103, October 2005, arXiv:quant-ph/0509153.

[KOS04] Adam R. Klivans, Ryan O'Donnell, and Rocco A. Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer and System Sciences*, 68(4):808–840, 2004.

[KS01] Adam R. Klivans and Rocco A. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. In *Proc. 33rd ACM STOC*, pages 258–265, 2001.

[KW93] Mauricio Karchmer and Avi Wigderson. On span programs. In *Proc. 8th IEEE Symp. Structure in Complexity Theory*, pages 102–111, 1993.

[Lee09] Troy Lee. A note on the sign degree of formulas. 2009, arXiv:0909.4607 [cs.CC].

[LLS06] Sophie Laplante, Troy Lee, and Mario Szegedy. The quantum adversary method and classical formula size lower bounds. *Computational Complexity*, 15:163–196, 2006, arXiv:quant-ph/0501057. Earlier version in Complexity'05.

[LM04] Sophie Laplante and Frédéric Magniez. Lower bounds for randomized and quantum query complexity using Kolmogorov arguments. In *Proc. 19th IEEE Complexity*, pages 294–304, 2004, arXiv:quant-ph/0311189.

[Lov03] László Lovász. Semidefinite programs and combinatorial optimization. In B. A. Reed and C. Linhares Sales, editors, *Recent Advances in Algorithms and Combinatorics*, volume 11 of *CMS Books Math.*, pages 137–194. Springer, 2003.

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information.* Cambridge University Press, Cambridge, 2000.

[OS03] Ryan O'Donnell and Rocco A. Servedio. New degree bounds for polynomial threshold functions. In *Proc. 35th ACM STOC*, pages 325–334, 2003.

[Rei09a] Ben W. Reichardt. Faster quantum algorithm for evaluating game trees. 2009, arXiv:0907.1623 [quant-ph].

[Rei09b] Ben W. Reichardt. Span-program-based quantum algorithm for evaluating unbalanced formulas. 2009, arXiv:0907.1622 [quant-ph].

[Rei09c] Ben W. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function. 2009, arXiv:0904.2759 [quant-ph].

[Rei09d] Ben W. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function. In *Proc. 50th IEEE FOCS*, pages 544–551, 2009.

[Rei10] Ben W. Reichardt. Reflections for quantum query complexity: The general adversary bound is tight for every boolean function. unpublished, 2010.

[RŠ08] Ben W. Reichardt and Robert Špalek. Span-program-based quantum algorithm for evaluating formulas. In *Proc. 40th ACM STOC*, pages 103–112, 2008, arXiv:0710.2630 [quant-ph].

[Špa09] Robert Špalek. private communication, 2009.

[ŠS06] Robert Špalek and Mario Szegedy. All quantum adversary methods are equivalent. *Theory of Computing*, 2(1):1–18, 2006, arXiv:quant-ph/0409116. Earlier version in ICALP'05.

[Zha05] Shengyu Zhang. On the power of Ambainis's lower bounds. *Theoretical Computer Science*, 339(2-3):241–256, 2005, arXiv:quant-ph/0311060. Earlier version in ICALP'04.

ECCC ISSN 1433-8092
http://eccc.hpi-web.de