Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function

Ben W. Reichardt[*]

Abstract

The general adversary bound is a semi-definite program (SDP) that lower-bounds the quantum query complexity of a function. We turn this lower bound into an upper bound, by giving a quantum walk algorithm based on the dual SDP that has query complexity at most the general adversary bound, up to a logarithmic factor.

In more detail, the proof has two steps, each based on "span programs," a certain linear-algebraic model of computation. First, we give an SDP that outputs for any boolean function a span program computing it that has optimal "witness size." The optimal witness size is shown to coincide with the general adversary lower bound. Second, we give a quantum algorithm for evaluating span programs with only a logarithmic query overhead on the witness size.

The first result is motivated by a quantum algorithm for evaluating composed span programs. The algorithm is known to be optimal for evaluating a large class of formulas. The allowed gates include all constant-size functions for which there is an optimal span program. So far, good span programs have been found in an ad hoc manner, and the SDP automates this procedure. Surprisingly, the SDP's value equals the general adversary bound. A corollary is an optimal quantum algorithm for evaluating "balanced" formulas over any finite boolean gate set.

The second result broadens span programs' applicability beyond the formula-evaluation problem. We extend the analysis of the quantum algorithm for evaluating span programs. The previous analysis shows that a corresponding bipartite graph has a large spectral gap, but only works when applied to the composition of constant-size span programs. We show generally that properties of eigenvalue-zero eigenvectors in fact imply an "effective" spectral gap around zero.

A strong universality result for span programs follows. A good quantum query algorithm for a problem implies a good span program, and vice versa. Although nearly tight, this equivalence is nontrivial. Span programs are a promising model for developing more quantum algorithms.

[*] School of Computer Science and Institute for Quantum Computing, University of Waterloo.

1 Introduction

Quantum algorithms for evaluating formulas have developed rapidly since the breakthrough AND-OR formula-evaluation algorithm [FGG07]. 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. Operationally, the new algorithms can be interpreted as evaluating "span programs," a certain linear-algebraic computational model (Def. 2.1). Discovering an optimal span program for a function immediately allows it to be added to the gate set [RŠ08].

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?

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, $\mathrm{Adv} \leq \mathrm{Adv}^{\pm}$ (Def. 2.4), but the power of the latter bound is not fully understood. What is the relationship between span program complexity, or "witness size" (Def. 2.2), 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?

Our first result answers the first two questions. 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 is surprising because of its broad scope. It allows us to optimally evaluate 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.

This result suggests a new technique for developing quantum algorithms for other problems. Based on the adversary lower bound, one can construct a span program, and hopefully turn this into an algorithm, i.e., an upper bound. Unfortunately, it has not been known how to evaluate general span programs. The second result of this paper is a quantum algorithm for evaluating span programs, with only a logarithmic query overhead on the witness size. The main technical difficulty is showing that a corresponding bipartite graph has a large spectral gap. We show that properties of eigenvalue-zero eigenvectors in fact imply an "effective" spectral gap around zero.

In combination, the two results imply that the general adversary bound, $\mathrm{Adv}^{\pm}$, is tight up to a logarithmic factor for every boolean function. This is surprising because $\mathrm{Adv}^{\pm}$ is closely connected to the nonnegative-weight adversary bound Adv, which has strong known limitations [Zha05, ŠS06, HLŠ07]. It implies a significantly simpler semi-definite program for quantum query complexity than has been known [BSS03]. The results also imply that quantum computers, measured by query complexity, and span programs, measured by witness size, are equivalent computational models, up to a logarithmic factor.

Some further background material is needed to place the results in context.

Quantum algorithms for evaluating formulas

Farhi, Goldstone and Gutmann in 2007 gave a nearly optimal quantum algorithm for evaluating balanced binary AND-OR formulas [FGG07, 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].

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 $\{0,1\}^n \to \{0,1\}$ for $n \leq 3$, such as AND, OR, PARITY and MAJ_3.

- 69 of the 92 inequivalent functions $f : \{0,1\}^4 \to \{0,1\}$ 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 gave 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 $f : \{0,1\}^n \to \{0,1\}$, 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, Thm. 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 span programs with optimized weights. (This fact has not been appreciated.) 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. The search space needs to be narrowed down.

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.

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 $f : \{0,1\}^n \to \{0,1\}$,*

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

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. A factor is lost that depends on the gates, but for a finite gate set, this will be a constant. As another corollary, Thm. 1.2 also settles the question of how the general adversary bound behaves under function composition, and it implies a new upper bound on the sign-degree of boolean functions.

Quantum algorithm for evaluating span programs

Now that we know there are span programs with witness size matching the general adversary bound, it is of more interest to extend the formula-evaluation algorithm to evaluate arbitrary span programs. Unfortunately, though, a key theorem from [RŠ08] does not hold general span programs.

The [RŠ08] algorithm to evaluate a formula φ works by plugging together optimal span programs for the individual gates in φ to construct a composed span program P that computes φ. Then a family of related graphs $G_P(x)$, one for each input x, is constructed. For an input x, the algorithm starts at a particular "output vertex" of the graph, and runs a quantum walk for about $1/\operatorname{wsize}(P)$ steps. The algorithm's analysis has two parts. First, for completeness, it is shown that when $\varphi(x) = 1$, there exists a normalized, eigenvalue-zero eigenvector of the weighted adjacency matrix $A_{G_P(x)}$ with large overlap on the output vertex. Thus there is a large stationary component to the walk, which is the algorithm detects. Second, for soundness, it is shown that if $\varphi(x) = 0$, then $A_{G_P(x)}$ has an $\Omega(1/\operatorname{wsize}(P))$ spectral gap around zero for eigenvectors supported on the output vertex. This spectral gap determines the algorithm's query complexity.

The completeness step of the proof comes from relating the definition of $G_P(x)$ to the witness size definition. Eigenvalue-zero eigenvectors correspond exactly to span program "witnesses," with the squared support on the output vertex corresponding to the witness size. This argument straightforwardly extends to arbitrary span programs.

For soundness, the proof essentially inverts the matrix $A_{G_P(x)} - \rho\mathbf{1}$ gate by gate, span program by span program, starting at the inputs and working recursively toward the output vertex. In this way, it roughly computes the Taylor series about $\rho = 0$ of the eigenvalue-ρ eigenvectors in order eventually to find a contradiction for $|\rho|$ small. One would not expect this method to extend to arbitrary span programs, because it loses a constant factor that depends badly on the individual span programs used for each gate. Indeed, it fails in general. Span programs can be constructed for which the associated graphs simply do not have an $\Omega(1/\operatorname{wsize}(P))$ spectral gap. (For example, take a large span program and add an AND gate to the top whose other input is 0. The composed span program computes the constant 0 function and has constant witness size, but the spectral gaps of the associated large graphs need not be $\Omega(1)$.)

It has not been fully understood why the [RŠ08] analysis works so well when applied to balanced compositions of constant-size optimal span programs. In particular, the correspondence between graphs and span programs by definition relates the witness size to properties of eigenvalue-zero eigenvectors. Why does the same witness size quantity also appear in the spectral gap?

We show that this is not a coincidence, that in general an eigenvalue-zero eigenvector of a bipartite graph implies an "effective" spectral gap for a perturbed graph. Somewhat more precisely, the inference is that the total squared overlap on the output vertex of small-eigenvalue eigenvectors is small. This argument leads to a substantially more general small-eigenvalue spectral analysis. It also implies simpler proofs of Thm. 1.1 as well as of the AND-OR formula-evaluation result in [ACR+07].

This small-eigenvalue analysis is the key step that allows us to evaluate span programs on a quantum computer. Besides showing an effective spectral gap, though, we would also need to bound $\|A_{G_P}\|$ in order to generalize [RŠ08]. However, recent work by Cleve et al. shows that this norm does not matter if we are willing to concede a logarithmic factor in the query complexity [CGM+09]. We thus obtain:

Theorem 1.3. *Let P be a span program computing $f : \{0,1\}^n \rightarrow \{0,1\}$. Then*

$$Q(f) = O\left(\mathrm{wsize}(P) \frac{\log \mathrm{wsize}(P)}{\log\log \mathrm{wsize}(P)}\right) . \tag{1.3}$$

We can now prove the main result of this paper, that for any boolean function f the general adversary bound on the quantum query complexity is tight up to a logarithmic factor:

Theorem 1.4. *For any function $f : \{0,1\}^n \rightarrow \{0,1\}$, the quantum query complexity of f satisfies*

$$Q(f) = \Omega(\mathrm{Adv}^{\pm}(f)) \quad \textit{and} \quad Q(f) = O\left(\mathrm{Adv}^{\pm}(f) \frac{\log \mathrm{Adv}^{\pm}(f)}{\log\log \mathrm{Adv}^{\pm}(f)}\right) . \tag{1.4}$$

Proof. The lower bound is due to [HLŠ07] (see Thm. 2.5). For the upper bound, use the SDP from Thm. 1.2, to construct a span program P computing f, with $\mathrm{wsize}(P) = \mathrm{Adv}^{\pm}(f)$. Then apply Thm. 1.3 to obtain a bounded-error quantum query algorithm that evaluates f. □

Thus the $\mathrm{Adv}^{\pm}$ semi-definite program is in fact an SDP for quantum query complexity, up to a logarithmic factor. Previously, Barnum et al. have already given an SDP for quantum query complexity [BSS03], and have shown that the nonnegative-weight adversary bound Adv can be derived by strengthening it, but their SDP is quite different. In particular, the $\mathrm{Adv}^{\pm}$ SDP is “greedy,” in the sense that it considers only how much information can be learned using a single query; see Def. 2.4 below. The [BSS03] SDP, on the other hand, has separate terms for every query. It is surprising that a small modification to Adv can not only break the certicate complexity and property testing barriers [HLŠ07], but in fact be nearly optimal always. For example, for the Element Distinctness problem with the input in $[n]^n$ specified in binary, $\mathrm{Adv}(f) = O(\sqrt{n} \log n)$ [ŠS06] but $Q(f) = \Omega(n^{2/3})$ by the polynomial method [AS04, Amb05]. Thm. 1.4 implies that $\mathrm{Adv}^{\pm}(f) = \Omega(n^{2/3}/\log n)$.

2 Definitions

Let $B = \{0,1\}$. For a natural number n, let $[n] = \{1, 2, \ldots, n\}$. For a finite set X, let $\mathbf{C}^X$ be the inner product space $\mathbf{C}^{|X|}$ with orthonormal basis $\{|x\rangle : x \in X\}$.

2.1 Span programs

A span program P is a certain linear-algebraic way of specifying a boolean function f_P [KW93]. The complexity measure we use to characterize span programs is the witness size [RŠ08].

Definition 2.1 ([KW93])**.** *A span program P on $n \in \mathbf{N}$ bits 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 $I = \bigsqcup_{j \in [n], b \in B} I_{j,b}$.*

P "computes" a function $f_P : B^n \to B$, defined by

$$f_P(x) = \begin{cases} 1 & \text{if } |t\rangle \in \operatorname{Span}(\{|v_i\rangle : i \in \bigcup_{j \in [n]} I_{j,x_j}\}) \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

Definition 2.2 ([RŠ08])**.** *Consider a span program P. Let $A = \sum_{i \in I} |v_i\rangle\langle i| \in \mathcal{L}(\mathbf{C}^I, V)$. For each input $x \in B^n$, let $I(x) = \bigcup_{j \in [n]} I_{j,x_j}$ and $\Pi(x) = \sum_{i \in I(x)} |i\rangle\langle i| \in \mathcal{L}(\mathbf{C}^I)$, and*

- *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 $A\Pi(x)|w\rangle = |t\rangle$. Let $\operatorname{wsize}(P, x)$ be the minimum squared length of any such witness:*

$$\operatorname{wsize}(P, x) = \min_{|w\rangle :\, A\Pi(x)|w\rangle = |t\rangle} \| |w\rangle \|^2 \ . \tag{2.2}$$

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

$$\operatorname{wsize}(P, x) = \min_{\substack{|w'\rangle :\, \langle t|w'\rangle = 1 \\ \Pi(x)A^\dagger|w'\rangle = 0}} \|A^\dagger|w'\rangle\|^2 \ . \tag{2.3}$$

Let the witness size of P be $\operatorname{wsize}(P) = \max_{x \in B^n} \operatorname{wsize}(P, x)$.

The ease with which span programs compose is one of their nicest features:

Theorem 2.3. *Consider boolean functions f and, for $j \in [n]$, f_j. Let $g(x) = f\big(f_1(x), f_2(x), \ldots, f_n(x)\big)$. Let span programs P_j compute $f_{P_j} = f_j$, and P compute $f_P = f$. Then there exists a span program Q computing $f_Q = g$, with $\operatorname{wsize}(Q) \leq \operatorname{wsize}(P) \max_{j \in [n]} \operatorname{wsize}(P_j)$.*

2.2 Adversary lower bounds

There are essentially two techniques for lower-bounding quantum query complexity, the polynomial method, due to Beals et al. [BBC$^+$01], and the adversary bounds, first introduced by Ambainis [Amb02]. The two methods are incomparable. The adversary bound Adv, from [ŠS06, HNS02, BS04, Amb06, Zha05, BSS03, LM04], is subject to a certificate complexity barrier: for f a total boolean function, $\operatorname{Adv}(f) \leq \sqrt{C_0(f)C_1(f)}$, where $C_b(f)$ is the certificate complexity of the inputs x with $f(x) = b$ [Zha05, ŠS06]. The polynomial method can surpass this barrier, for example for the Element Distinctness problem mentioned earlier. Adv also suffers a "property testing barrier" on partial functions. On the other hand, the polynomial method can also be loose. Ambainis gave a total boolean function f^k on $n = 4^k$ bits for which the polynomial method cannot give a bound larger than 2^k, but for which $\operatorname{Adv}(f^k) = 2.5^k$ [Amb06].

In 2007, though, Høyer et al. discovered a strict generalization $\operatorname{Adv}^\pm$ of Adv [HLŠ07]. For example, for Ambainis's function, $\operatorname{Adv}^\pm(f^k) \geq 2.51^k$. $\operatorname{Adv}^\pm$ also breaks the certificate complexity and property testing barriers. No similar limits on its power have been found.

Let us now define the two adversary bounds. On account of how their definitions differ, we call Adv the "nonnegative-weight" adversary bound, and $\operatorname{Adv}^\pm$ the "general" adversary bound.

Definition 2.4. *Let $f : B^n \to B$. An adversary matrix for f is a $2^n \times 2^n$ real, symmetric matrix Γ that, for $x, y \in B^n$ with $f(x) = f(y)$, has a zero (x, y) entry, $\langle x|\Gamma|y\rangle = 0$. The general adversary bound for f is*

$$\mathrm{Adv}^{\pm}(f) = \max_{\substack{\text{adversary matrices } \Gamma: \\ \forall j \in [n], \|\Gamma \circ \Delta_j\| \leq 1}} \|\Gamma\| \tag{2.4}$$

where $\Gamma \circ \Delta_j$ denotes the entry-wise matrix product between Γ and $\Delta_j = \sum_{x,y \in B^n : x_j \neq y_j} |x\rangle\langle y|$, and the norm is the operator norm. The nonnegative-weight adversary bound, $\mathrm{Adv}(f)$*, is the same, except with the entries of Γ required to be nonnegative; thus* $\mathrm{Adv}(f) \leq \mathrm{Adv}^{\pm}(f)$.

Theorem 2.5 ([HLŠ07])**.** *For any function $f : B^n \to B$, the bounded-error quantum query complexity of f satisfies $Q(f) = \Omega(\mathrm{Adv}^{\pm}(f))$.*

Theorem 2.6 ([RŠ08])**.** *For any span program P,* $\mathrm{wsize}(P) \geq \mathrm{Adv}^{\pm}(f_P)$.

3 Canonical span programs

In looking for an optimal span program, we prove that it suffices to search over span programs with a very restricted form, so-called canonical span programs. This reduction will be essential not only for the SDP for span programs, but also for the quantum algorithm for evaluating span programs.

Definition 3.1 ([KW93])**.** *A span program P on n bits is* canonical *if $V = \mathbf{C}^{F_0}$ where $F_0 = \{x \in B^n : f_P(x) = 0\}$, the target vector is $|t\rangle = \sum_{x \in F_0} |x\rangle$, and for all $x \in F_0$ and $i \in I(x)$, $\langle x|v_i\rangle = 0$.*

Theorem 3.2. *A span program P can be converted to a canonical span program $\hat{P}$ that computes $f_{\hat{P}} = f_P$, with* $\mathrm{wsize}(\hat{P}) \leq \mathrm{wsize}(P)$*. For all x with $f_{\hat{P}}(x) = 0$, $|x\rangle$ itself is an optimal witness.*

Proof. We use the conversion procedure from [KW93, Thm. 6], and analyze the witness size. □

4 Span program witness size and the general adversary bound

We now formulate a semi-definite program for the optimal span program computing a function f. Remarkably, this SDP turns out to be the dual of the SDP in Eq. (2.4) for $\mathrm{Adv}^{\pm}(f)$.

Proof of Thm. 1.2. Lemma 4.1 constructs an SDP whose value is the optimal witness size of a span program computing f. Thm. 4.2 takes the dual of this SDP to show that it evaluates to $\mathrm{Adv}^{\pm}(f)$.

Lemma 4.1. *For a function $f : B^n \to B$, let $F_b = \{x : f(x) = b\}$ for $b \in B$. Then,*

$$\inf_{P : f_P = f} \mathrm{wsize}(P) = \inf_{\substack{m \in \mathbf{N}, \\ \{|v_{xj}\rangle \in \mathbf{R}^m : x \in B^n, j \in [n]\} : \\ \forall (x,y) \in F_0 \times F_1, \sum_{j : x_j \neq y_j} \langle v_{xj}|v_{yj}\rangle = 1}} \max_{x \in B^n} \sum_{j \in [n]} \||v_{xj}\rangle\|^2 \ . \tag{4.1}$$

Proof. We prove the $\leq$ direction here. Given a solution $\{|v_{xj}\rangle\}$, let P be a span program with target $|t\rangle = \sum_{x \in F_0} |x\rangle$ 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

$$A := \sum_{j \in [n], b \in B, k \in [m]} |v_{jbk}\rangle\langle j, b, k| = \sum_{x \in F_0, j \in [n]} |x\rangle\langle j, \bar{x}_j| \otimes \langle v_{xj}| \ . \tag{4.2}$$

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 $\| A^\dagger |x\rangle \|^2 = \sum_j \| |v_{xj}\rangle \|^2$.

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

Theorem 4.2. *Let $f : B^n \to B$. Let $F = \{(x, y) \in B^n \times B^n : f(x) \neq f(y)\}$. Then*

$$\mathrm{Adv}(f) = \min_{\substack{\{X_j \succeq 0\}: \\ \forall (x,y) \in F, \sum_{j : x_j \neq y_j} \langle x|X_j|y\rangle \geq 1}} \max_{x \in B^n} \sum_{j \in [n]} \langle x|X_j|x\rangle \tag{4.3}$$

$$\mathrm{Adv}^\pm(f) = \min_{\substack{\{X_j \succeq 0\}: \\ \forall (x,y) \in F, \sum_{j : x_j \neq y_j} \langle x|X_j|y\rangle = 1}} \max_{x \in B^n} \sum_{j \in [n]} \langle x|X_j|x\rangle \ . \tag{4.4}$$

Proof. The proof is by standard SDP duality theory. Eq. (4.3) is from [HLŠ07]. The expression (4.4) for $\mathrm{Adv}^\pm(f)$ is new, though, and is somewhat simpler than the dual SDP known before. □

Now the right-hand side of Eq. (4.1) is the Cholesky decomposition of Eq. (4.4)'s SDP solution. □

Let us state three of Thm. 1.2's corollaries. First, using also Thm. 1.1, is an exact asymptotic expression for the quantum query complexity of a boolean function f composed on itself, and therefore a new upper bound on the sign-degree.

Theorem 4.3. *For any function $f : B^n \to B$, define $f^k : B^{n^k} \to B$ as the function f composed on itself repeatedly to a depth of k, as in Thm. 1.1. Then*

$$\limsup_{k \to \infty} \text{sign-degree}(f^k)^{1/k} \leq \lim_{k \to \infty} Q(f^k)^{1/k} = \mathrm{Adv}^\pm(f) \ . \tag{4.5}$$

Lee and Servedio have recently shown that $\text{sign-degree}(f)^k \leq \text{sign-degree}(f^k)$ [Lee09], based on which the above gives an upper bound of the sign-degree of f itself.

The general adversary bound composes multiplicatively for boolean functions:

Theorem 4.4. *Let f and, for $j \in [n]$, f_j be boolean functions. Let $g(x) = f\big(f_1(x_1), \ldots, f_n(x_n)\big)$. If $\mathrm{Adv}^\pm(f_j) = \beta$ for all j, then $\mathrm{Adv}^\pm(g) = \beta \, \mathrm{Adv}^\pm(f)$.*

Proof. The $\geq$ direction is from [HLŠ07]. For $\leq$, apply Thm. 2.3 and Thm. 1.2. □

Thm. 1.1 was a special case of the main [RŠ08] formula-evaluation result, which can also be extended. Modifying [RŠ08, Def. 4.5], defining adversary-balanced formulas, to refer to $\mathrm{Adv}^\pm$ instead of Adv, and letting $\mathcal{S}$ be any finite set of boolean functions, [RŠ08, Thm. 4.7] becomes:

Theorem 4.5. *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 $\mathrm{Adv}^\pm(\varphi)(\log \mathrm{Adv}^\pm(\varphi))^{O(1)}$.*

The proof from [RŠ08] goes through entirely. Note that layered formulas, in which gates at the same depth are the same, are a special case of adversary-balanced formulas.

5 Correspondence between span programs and bipartite graphs

We now define a correspondence between span programs and weighted bipartite graphs, slightly generalizing the construction in [RŠ08]. We analyze the spectra of these graphs, focusing on eigenvalues near zero and eigenvectors supported on one particular "output vertex."

Definition 5.1 (Graphs $G_P(x)$)**.** *Let P be a span program with target vector $|t\rangle$ and input vectors $|v_i\rangle$ for $i \in I = \bigcup_{j\in[n],b\in B} I_{j,b}$, in inner product space V. Let G_P be the weighted bipartite graph with $T = [\dim(V)] \sqcup I$, $U = \{\mu\} \sqcup I$ and the biadjacency matrix $B_{G_P} \in \mathcal{L}(\mathbf{C}^U, \mathbf{C}^T)$,*

$$B_{G_P} = \begin{array}{c} \begin{array}{cc} \mu & I \end{array} \\ \begin{pmatrix} |t\rangle & A \\ 0 & \mathbf{1} \end{pmatrix} \end{array} \begin{array}{l} \\ V \\ I \end{array} \tag{5.1}$$

The vertex μ is called the "output vertex."

Note that G_P has two vertices for each $i \in I$, with a weight-one edge between them. For $x \in B^n$, let $G_P(x)$ be the same as G_P except with these weight-one edges deleted for all $i \in I(x)$.

Our main result relates spectral quantities of interest to the span program witness size. This is the key theorem that allows span programs to be evaluated on a quantum computer.

Theorem 5.2. *Let P be a span program. Then a span program P' can be constructed such that $f_{P'} = f_P$ and, for all $x \in B^n$, letting $\{|\alpha\rangle\}$ be a complete set of orthonormal eigenvectors of the weighted adjacency matrix $A_{G_{P'}(x)}$, with corresponding eigenvalues $\rho(\alpha)$, for all $c \geq 0$,*

$$f_P(x) = 1 \;\Rightarrow\; \sum_{\alpha\colon \rho(\alpha)=0} |\langle\alpha|\mu\rangle|^2 \geq 1/2 \tag{5.2}$$

$$f_P(x) = 0 \;\Rightarrow\; \sum_{\alpha\colon |\rho(\alpha)| \leq c/\mathrm{wsize}(P)} |\langle\alpha|\mu\rangle|^2 \leq 16c^2 \tag{5.3}$$

The two main ingredients required for proving Thm. 5.2 are an eigenvalue-zero analysis of $A_{G_P}(x)$ and an analysis relating eigenvalue-zero eigenvectors to an effective spectral gap. The eigenvalue-zero analysis is a straightforward extension of [RŠ08, Thms. 2.5 and A.7]:

Theorem 5.3 ([RŠ08])**.** *For a span program P and input $x \in B^n$,*

- *If $f_P(x) = 1$, $A_{G_P(x)}$ has an eigenvalue-zero eigenvector $|\psi\rangle = (|\psi_T\rangle, |\psi_U\rangle) \in \mathbf{C}^T \oplus \mathbf{C}^U$ with*

$$\frac{|\langle\mu|\psi_U\rangle|^2}{\||\psi\rangle\|^2} \geq \frac{1}{1 + \mathrm{wsize}(P, x)} \quad . \tag{5.4}$$

- *If $f_P(x) = 0$, let $|w'\rangle \in V$ be an optimal witness (see Def. 2.2). Then there is a solution $|\psi\rangle$ to all the eigenvalue-zero equations of $A_{G_P(x)}$, except for the constraint at μ, with*

$$\frac{|\langle t|\psi_T\rangle|^2}{\||\psi\rangle\|^2} \geq \frac{1}{\||w'\rangle\|^2 + \mathrm{wsize}(P, x)} \quad . \tag{5.5}$$

The difficult step in proving Thm. 5.2 is applying Eq. (5.5) to show Eq. (5.3), which is in a certain sense an "effective" spectral gap around zero. We give a general argument that relates properties of eigenvalue-zero eigenvectors of weighted bipartite graphs to effective spectral gaps.

Theorem 5.4. *Let G be a weighted bipartite graph with biadjacency matrix $B_G \in \mathcal{L}(\mathbf{C}^U, \mathbf{C}^T)$. Assume that for some $\delta > 0$ and $|t\rangle \in \mathbf{C}^T$, the adjacency matrix A_G has an eigenvalue-zero eigenvector $|\psi\rangle$ with*

$$\left|\langle t|\psi_T\rangle\right|^2 \geq \delta \||\psi\rangle\|^2 \ . \tag{5.6}$$

Let G' be the same as G except with a new vertex, μ, added to the U side, and for $i \in T$ the new edge (i, μ) weighted by $\langle i|t\rangle$. That is, the biadjacency matrix of G' is

$$B_{G'} = \begin{array}{ccc} \mu & U & \\ \big(|t\rangle & B_G\big) & T \end{array} \tag{5.7}$$

Take $\{|\alpha\rangle\}$ a complete set of orthonormal eigenvectors of $A_{G'}$, with eigenvalues $\rho(\alpha)$. Then

$$\forall\, \Upsilon \geq 0, \qquad \sum_{\alpha:\, |\rho(\alpha)| \leq \Upsilon} \left|\langle \alpha|\mu\rangle\right|^2 \leq 8\Upsilon^2/\delta \ . \tag{5.8}$$

Proving Thm. 5.4 requires some involved linear algebra in order to show the key lemma:

Lemma 5.5. *Let $X \in \mathcal{L}(V)$ be a positive semi-definite matrix, $|t\rangle \in V$ a vector, and let $X' = X + |t\rangle\langle t|$. Let $\{|\beta\rangle\}$ be a complete set of orthonormal eigenvectors of X', with eigenvalues $\lambda(\beta) \geq 0$. Assume that there exists a $|\varphi\rangle \in \mathrm{Ker}(X)$ with $\left|\langle t|\varphi\rangle\right|^2 \geq \delta \||\varphi\rangle\|^2$. Then for any $\Lambda \geq 0$,*

$$\delta \sum_{\substack{\beta:\, \lambda(\beta) \leq \Lambda \\ \langle t|\beta\rangle \neq 0}} \frac{1}{\lambda(\beta)} \left|\langle t|\beta\rangle\right|^2 \leq 4\Lambda \ . \tag{5.9}$$

Proof of Thm. 5.2. Let $\hat{P}$ be the canonical span program from Thm. 3.2. In particular, when $f_P(x) = 0$, $|x\rangle$ itself is an optimal witness. Let P' be the same as $\hat{P}$ except with the target vector scaled by a factor of $1/\sqrt{\mathrm{wsize}(P)}$. Now combine Thm. 5.3 and Thm. 5.4. □

6 Quantum algorithm for evaluating span programs

We will next connect quantum algorithms to the graph spectral properties that follow from Thm. 5.2.

Theorem 6.1. *Let $G = (V, E)$ be a complex-weighted graph with $\langle v|A_G|v\rangle \geq 0$ for all $v \in V$, and let V_{input} be a subset of degree-one vertices of G whose incident edges have weight one. Let V_{input} be partitioned as $V_{input} = \bigsqcup_{j \in [n], b \in B} V_{j,b}$. For $x \in B^n$, define $G(x)$ from G by deleting all edges to vertices in $\bigcup_{j \in [n]} V_{j,x_j}$.*

Let $f : B^n \to B$, $\mu \in V \smallsetminus V_{input}$, $\epsilon = \Omega(1)$ and $\Lambda > 0$. For $x \in B^n$ let $\{|\alpha\rangle\}$ be a complete set of orthonormal eigenvectors of $A_{G(x)}$, with corresponding eigenvalues $\rho(\alpha)$, and assume that

$$f(x) = 1 \ \Rightarrow \sum_{\alpha:\, \rho(\alpha) = 0} \left|\langle \alpha|\mu\rangle\right|^2 \geq \epsilon \tag{6.1}$$

$$f(x) = 0 \ \Rightarrow \sum_{\alpha:\, |\rho(\alpha)| \leq \Lambda} \left|\langle \alpha|\mu\rangle\right|^2 \leq \epsilon/2 \tag{6.2}$$

Let $\mathrm{abs}(A_G)$ be the entry-wise absolute value of A_G. Then

$$Q(f) = O\left(\min\left\{\frac{\|\mathrm{abs}(A_G)\|}{\Lambda},\ \frac{1}{\Lambda}\frac{\log\frac{1}{\Lambda}}{\log\log\frac{1}{\Lambda}}\right\}\right) \ . \tag{6.3}$$

The intuition is that f can be evaluated by starting at $|\mu\rangle \in \mathbf{C}^V$ and "measuring" $A_{G(x)}$ to precision Λ. Output 1 iff the measurement result is zero. Eq. (6.1) implies completeness, because the initial state has large overlap with an eigenvalue-zero eigenstate. Eq. (6.2) implies soundness.

In fact, the proof requires two quantum algorithms, one for each bound in Eq. (6.3).

1. The $O(\| \operatorname{abs}(A_G) \| / \Lambda)$ bound is based on Szegedy's correspondence between continuous- and discrete-time quantum walks [Sze04], and is similar to [RŠ08, App. B.2]. This is the algorithm behind the formula-evaluation applications, Thm. 1.1, Thm. 4.3 and Thm. 4.5.

2. The second bound, $Q(f) = \tilde{O}(1/\Lambda)$, is applicable in the more typical case when we do not know an upper bound on $\| \operatorname{abs}(A_G) \|$. The idea is to apply phase estimation to $e^{iA_{G(x)}}$. Since A_G is independent of the input x, recent work by Cleve et al. shows that its norm does not matter if we can concede a logarithmic factor [CGM+09]. For applying phase estimation, there is still the problem that eigenvalues can wrap around the circle, e.g., $e^{2\pi i} = e^{0i}$, leading to false positives. To avoid these, we scale $A_{G(x)}$ by a uniformly random $R \in (0, 144/\epsilon^2)$.

Although Thm. 6.1 refers only to query complexity, the first algorithm's time complexity can also often be bounded under reasonable assumptions on G [RŠ08, CNW09].

For a span program P, the graphs G_P and $G_P(x)$ from Def. 5.1 are of the form required by Thm. 6.1, and the assumptions Eqs. (6.1) and (6.2) correspond to the conclusion of Thm. 5.2. Therefore, as a corollary of Thm. 6.1 we obtain a quantum algorithm for evaluating span programs:

Corollary 6.2. *Let P be a span program. Then the quantum query complexity of f_P satisfies*

$$Q(f_P) = O\left(\operatorname{wsize}(P) \frac{\log \operatorname{wsize}(P)}{\log \log \operatorname{wsize}(P)} \right) . \tag{6.4}$$

Proof. Set $c = 1/8$ in Thm. 5.2 and apply Thm. 6.1 with $\epsilon = 1/2$ and $\Lambda = c/\operatorname{wsize}(P)$. □

7 Open problems

Span programs appear to be a useful tool for developing quantum algorithms, especially for evaluating formulas, but their potential has not been fully explored. There remain a number of unresolved problems even in formula evaluation, especially how best to evaluate unbalanced formulas [Rei09b, Rei09c].

Although this extended abstract has focused on query complexity, Thm. 6.1 is more than an information-theoretic statement; it gives explicit algorithms whose time complexity can be analyzed. The full version [Rei09a] has further discussion of this issue, including some pertinent new theorems.

It is an interesting problem to consider functions with non-binary input alphabet and non-boolean codomain. Then the natural generalization of the SDP in Eq. (4.4) is not dual to $\mathrm{Adv}^{\pm}$. Moreover, there are functions $[3]^2 \to [3]$ for which both $\mathrm{Adv}^{\pm}$ and the generalization of Eq. (4.4) compose strictly sub-multiplicatively [Špa09].

Although it can be difficult to compute $\mathrm{Adv}^{\pm}$, it is often easy to guess Adv [CL08]. If the two bounds are close, then perhaps the Adv bound can also be turned into a quantum walk algorithm.

Finally, we ask whether the logarithmic overhead can be removed from Thm. 1.4.

Acknowledgements

I would like to thank Robert Špalek and Troy Lee for many helpful discussions.

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. Earlier version in STOC'00.

[Amb05] Andris Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing*, 1:37–46, 2005.

[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 *Proc. 44th IEEE FOCS*, 2003.

[AS04] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problem. *J. ACM*, 51(4):595–605, 2004.

[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.

[CGM$^+$09] Richard Cleve, Daniel Gottesman, Michele Mosca, Rolando Somma, and David L. Yonge-Mallo. Efficient discrete-time simulations of continuous-time quantum query algorithms. In *Proc. 41st ACM STOC*, pages 409–416, 2009, arXiv:0811.4428 [quant-ph].

[CL08] Andrew M. Childs and Troy Lee. Optimal quantum adversary lower bounds for ordered search. In *Proc. 35th ICALP*, LNCS vol. 5125, pages 869–880, 2008, arXiv:0708.3396 [quant-ph].

[CNW09] Chen-Fu Chiang, Daniel Nagaj, and Pawl Wocjan. An efficient circuit for the quantum walk update rule. 2009, arXiv:0903.3465 [quant-ph].

[FGG07] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum algorithm for the Hamiltonian NAND tree. 2007, arXiv:quant-ph/0702144.

[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.

[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. private communication, 2009.

[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.

[Rei09a] 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].

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

[Rei09c] Ben W. Reichardt. Faster quantum algorithm for evaluating game trees. 2009, arXiv:0907.1623 [quant-ph].

[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.

[Sze04] Mario Szegedy. Quantum speed-up of Markov chain based algorithms. In *Proc. 45th IEEE FOCS*, pages 32–41, 2004.

[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.