Electronic Colloquium on Computational Complexity, Report No. 110 (2010)

Span programs and quantum query algorithms

Ben W. Reichardt

Abstract

Quantum query complexity measures the number of input bits that must be read by a quantum algorithm in order to evaluate a function. Høyer et al. (2007) have generalized the adversary semi-definite program that lower-bounds quantum query complexity. By giving a matching quantum algorithm, we show that the general adversary lower bound is tight for every boolean function.

The proof is based on span programs, a linear-algebraic computational model without inherent dynamics. Span programs correspond to solutions to the dual semi-definite program, and to bipartite graphs. The analysis shows that properties of eigenvalue-zero eigenvectors of the graphs in fact imply an "effective" spectral gap around zero We thus develop a quantum algorithm for evaluating span programs. It follows that span programs, measured by witness size, and quantum algorithms, measured by query complexity, are equivalent computational models, up to a constant factor.

The result efficiently characterizes the quantum query complexity of a read-once formula over any finite gate set. It also implies that the quantum query complexity of the composition $f \circ (g, \ldots, g)$ of two boolean functions matches the product of the query complexities of f and g, without a logarithmic factor for error reduction. The algorithm alternates a fixed reflection with input queries. Originally introduced for solving the unstructured search problem, this structure is therefore a universal feature of quantum query algorithms.

We give a second algorithm for evaluating span programs that has the further potential to be time-efficient. Subsequent applications have derived nearly time-optimal quantum algorithms for evaluating many read-once formulas. Span programs may have promise for developing more quantum algorithms.

1 Introduction

The query complexity, or decision-tree complexity, of a function measures the number of input bits that must be read in order to evaluate the function. Computation between queries is not counted. Quantum algorithms can run in superposition, and the quantum query complexity therefore allows coherent access to the input string (Figure 1). Quantum query complexity with bounded error can be far smaller than classical randomized query complexity [BV97, Sim97, Sho97, Aar10], but for total functions [BBC+01] or functions satisfying certain symmetries [AA09] the two measures are polynomially related; see the survey [BW02].

Although the query complexity of a function can fall well below its time complexity, studying query complexity has historically given insight into the power of quantum computers. For example, the quantum part of Shor's algorithms for integer factorization and discrete logarithm is a quantum query algorithm for period finding [Sho97]. Grover's unstructured database search algorithm is a quantum query algorithm for evaluating the n-bit OR function, with $\Theta(\sqrt{n})$ queries [Gro96, BBHT98].

Unlike for time complexity, there are also strong information-theoretic techniques for placing lower bounds on quantum query complexity. These lower-bound techniques can be broadly classified

ISSN 1433-8092

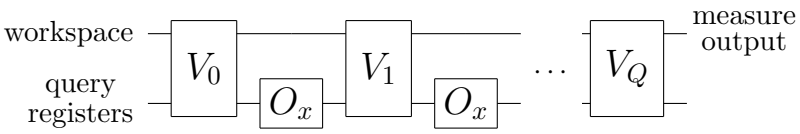

Figure 1: Beginning in a fixed initial state, a quantum query algorithm alternates arbitrary unitaries V_q that do not depend on the input with coherent input oracle queries O_x. For a boolean function, O_x is the unitary that maps $|j,b\rangle$ to $|j,x_j \oplus b\rangle$, for $j = 1,\ldots,n$ and $b \in \{0,1\}$. Finally, the output is measured. The algorithm's query complexity is the number of calls made to O_x, i.e., Q in the illustrated circuit diagram.

as using either the polynomial method [BBC+01] or the adversary method [Amb02, ŠS06]. Høyer and Špalek [HŠ05] have surveyed the development of these two methods and their multitude of applications. In particular, Ambainis's original lower bound [Amb02] has several stronger variants [HNS02, BS04, Amb06, Zha05, BSS03, LM04] that are equivalent to each other [ŠS06]. The two techniques are incomparable. For instance, for the n-input collision problem, the best adversary lower bound is of $O(1)$, whereas the correct complexity, determined by the polynomial method [AS04, Kut05, Amb05] and a matching algorithm [BHT98] is $\Theta(n^{1/3})$. Ambainis defined functions f^k that can be represented exactly by a polynomial of degree 2^k, which limits the polynomial method, but for which the adversary lower bound is 2.5^k [Amb06].

However, Høyer, Lee and Špalek [HLŠ07] have discovered a strict generalization of the adversary bound that remains a lower bound on quantum query complexity:

Definition 1.1 (Adversary bounds)**.** *For finite sets C and E, and $\mathcal{D} \subseteq C^n$, let $f : \mathcal{D} \to E$. An adversary matrix for f is a $|\mathcal{D}| \times |\mathcal{D}|$ real, symmetric matrix Γ that satisfies $\Gamma_{xy} = 0$ for all $x, y \in \mathcal{D}$ with $f(x) = f(y)$. Define the adversary and general adversary bounds for f by*

$$\mathrm{Adv}(f) = \max_{\Gamma \geq 0} \frac{\|\Gamma\|}{\max_{j \in [n]} \|\Gamma^{(j)}\|} \tag{1.1}$$

$$\mathrm{Adv}^{\pm}(f) = \max_{\Gamma} \frac{\|\Gamma\|}{\max_{j \in [n]} \|\Gamma^{(j)}\|} \ . \tag{1.2}$$

Both maximizations are over adversary matrices Γ, required to be entry-wise nonnegative in $\mathrm{Adv}(f)$. $\Gamma^{(j)}$ denotes the restriction of Γ to those entries (x,y) with $x_j \neq y_j$.

Observe that $\mathrm{Adv}^{\pm}(f) \geq \mathrm{Adv}(f)$ always. Although the two bounds are sometimes equal, it appears that they usually differ [HLŠ06]. For Ambainis's functions f^k, $\mathrm{Adv}^{\pm}(f^k) > 2.51^k$.

Theorem 1.2 ([BSS03, HLŠ07])**.** *For any function f, let $Q(f)$ be the quantum query complexity for evaluating f with error probability at most $1/10$. Then $Q(f) = \Omega\big(\mathrm{Adv}^{\pm}(f)\big)$ and in particular,*

$$Q(f) \geq \frac{1}{5}\mathrm{Adv}(f) \qquad \text{and} \qquad Q(f) \geq \frac{1}{10}\mathrm{Adv}^{\pm}(f) \ . \tag{1.3}$$

The proof of Theorem 1.2 is similar to a classical hybrid argument. It works by considering a superposition of inputs, determined by a principal eigenvector of the adversary matrix Γ. Roughly, a successful algorithm must learn information nearly $\|\Gamma\|$, with at most $\max_{j \in [n]} \|\Gamma^{(j)}\|$ learned from any one query. The ratio of these quantities thus lower-bounds $Q(f)$.

Although the two bounds have very similar definitions, the general adversary bound $\mathrm{Adv}^{\pm}$ is in fact much more powerful than the adversary bound Adv. Our main result is that the general adversary lower bound is tight for every boolean function:

Theorem 1.3. *For any function $f : \mathcal{D} \to \{0,1\}$, with $\mathcal{D} \subseteq \{0,1\}^n$, the general adversary bound characterizes quantum query complexity:*

$$Q(f) = \Theta\big(\mathrm{Adv}^{\pm}(f)\big) \ . \tag{1.4}$$

Based on binary input encodings, Lee et al. have generalized Theorem 1.3 to show that the general adversary bound is tight up to logarithmic factors for every function f:

Theorem 1.4 ([LMRŠ10])**.** *For finite sets C and E, and $\mathcal{D} \subseteq C^n$, let $f : \mathcal{D} \to E$. Then*

$$Q(f) = \Omega\big(\mathrm{Adv}^{\pm}(f)\big) \qquad \text{and} \qquad Q(f) = O\big(\mathrm{Adv}^{\pm}(f) \log |C|\big) \ . \tag{1.5}$$

Theorems 1.3 and 1.4 suggest that one way of developing new quantum query algorithms may be to solve for the general adversary bound. More precisely, Eq. (1.2) is a semi-definite program (SDP), and for algorithms we need solutions to the dual SDP (see Lemma 3.1). The SDP is typically exponentially large, depending on $|\mathcal{D}|$. Even so, solving the equally large adversary bound SDP is often straightforward. For instance, there is a solution that implies $\mathrm{Adv}(f) \leq \sqrt{n \min_b C_b(f)}$ for a partial boolean function, or $\mathrm{Adv}(f) \leq \sqrt{C_0(f) C_1(f)}$ for a total boolean function, where $C_b(f)$ is the certificate complexity on inputs x with $f(x) = b$ [Sze03, LM04, Zha05, ŠS06]. The general adversary lower bound is fortunately not limited by this certificate complexity barrier [HLŠ07], but no equally simple dual SDP solution is known. Removing from the primal SDP the constraints $\Gamma_{xy} \geq 0$ changes dual SDP inequalities to stronger, equality constraints. Another useful property of the adversary bound is that the dual SDP always has a rank-one solution [HLŠ07, Theorem 18], which does not appear to hold for $\mathrm{Adv}^{\pm}$ [Rei10a].

With the above qualifications, studying the general adversary bound is a promising approach to developing new quantum algorithms and to furthering our understanding of quantum query complexity. In particular, Theorem 1.3 resolves the query complexity for composed functions and read-once formulas, because the general adversary bound composes easily (see Section 1.1 below). The algorithm used to prove Theorem 1.3 has an especially simple form, based on a single fixed reflection, that may be useful for other applications (see Section 1.2). Finally, the dual SDP for $\mathrm{Adv}^{\pm}$ turns out to be closely related to the span program computational model, which has been well-studied in classical complexity theory. One application of Theorem 1.3 is to show that span programs, measured by the witness size complexity measure, and quantum algorithms, measured by query complexity, are equivalent computational models, up to a constant factor. Thus efficient quantum query algorithms can be derived by finding new span programs, which are essentially linear-algebraic objects or can also be seen as weighted bipartite graphs. This algorithmic approach has consequences for time complexity as well as query complexity (see Section 1.3).

Barnum, Saks and Szegedy [BSS03] have given a family of SDPs that characterize quantum query complexity according to their feasibility or infeasibility, instead of according to the optimum value of a single SDP. The BSS SDPs work for any specified error rate, including zero. The general adversary bound is a polynomially smaller SDP, since it does not need separate terms for every query, but of course the truth table of a function is typically exponentially long anyway. Whereas our algorithm uses a workspace of $n + O(\log n)$ qubits to evaluate an n-bit boolean function (by [Rei10a, Lemma 5.3]), $n + 1$ qubits suffice by [BSS03].

1.1 Composition of functions, algorithms and lower bounds

A fundamental problem in the theory of computation is how the query complexity transforms under function composition. In the simplest case, for $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$, let $f \bullet g$ be the function $\{0,1\}^{nm} \to \{0,1\}$ defined by

$$(f \bullet g)(x) = f\big(g(x_1, \ldots, x_m), \ldots, g(x_{(n-1)m+1}, \ldots, x_{nm})\big) \ . \tag{1.6}$$

How does the query complexity of $f \bullet g$ relate to the query complexities of f and g? For deterministic classical query complexity D, $D(f \bullet g) = D(f)D(g)$, and the optimal algorithm for evaluating $f \bullet g$ is the composition of the optimal algorithms for evaluating f and for evaluating g. For bounded-error randomized and quantum query complexities, R and Q, respectively, $R(f \bullet g) = O\big(R(f)R(g)\log R(f)\big)$ and $Q(f \bullet g) = O\big(Q(f)Q(g)\log Q(f)\big)$. Indeed, in these cases, a bounded-error algorithm for evaluating $f \bullet g$ can be built directly from bounded-error algorithms for f and for g, if we use repetition to reduce the error rate of the inner algorithm to below roughly $1/R(f)$ or $1/Q(f)$. However, the extra logarithmic factors are not always necessary, for example when $f = g$ is an OR or PARITY function.

A beautiful property of the adversary bounds is that they transform multiplicatively under function composition. That is, $\mathrm{Adv}(f \bullet g) = \mathrm{Adv}(f)\mathrm{Adv}(g)$ [Amb06, LLS06] and

$$\mathrm{Adv}^{\pm}(f \bullet g) = \mathrm{Adv}^{\pm}(f)\mathrm{Adv}^{\pm}(g) \ . \tag{1.7}$$

Here, the $\geq$ direction is from [HLŠ07] and the $\leq$ direction is from [Rei10a], inspired by span program composition rules. Using Eq. (1.7), Theorem 1.3 implies:

Theorem 1.5. *Let $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$. Then*

$$Q(f \bullet g) = \Theta\big(Q(f)Q(g)\big) \ . \tag{1.8}$$

The novelty of Theorem 1.5 is that the $\log Q(f)$ factor can always be removed in the quantum case, and that there is a matching lower bound. Moreover, by applying Eq. (1.7) recursively, we obtain that for any boolean functions $f_1, f_2, \ldots, f_k$, $Q(f_1 \bullet \cdots \bullet f_k) = \Theta\big(\mathrm{Adv}^{\pm}(f_1) \cdots \mathrm{Adv}^{\pm}(f_k)\big)$, where the hidden constants depend neither on k nor the f_j. Naïve composition of the algorithms for $f_1, \ldots, f_k$ works poorly in this case, since the product of the error-reduction factors grows exponentially with k.

More generally, a read-once formula φ over $\mathcal{S}$, a set of boolean functions, is a rooted tree in which each node with m children is associated to a m-bit function from $\mathcal{S}$, for $m \in \mathbf{N}$. Any such tree with n leaves naturally defines a function $\varphi : \{0,1\}^n \to \{0,1\}$, by evaluating the functions recursively toward the root. Note that the different input subproblems to any given node of φ are independent of each other, which is important for lower bounds.

Definition 1.1 can be extended to allow weights $s \in (0,\infty)^n$, by letting

$$\mathrm{Adv}^{\pm}_s(f) = \max_{\Gamma} \frac{\|\Gamma\|}{\max_{j \in [n]} \frac{1}{s_j}\|\Gamma^{(j)}\|} \ . \tag{1.9}$$

Then for $f : \{0,1\}^k \to \{0,1\}$ and $g_j : \{0,1\}^{m_j} \to \{0,1\}$, defining $h : \{0,1\}^{m_1} \times \cdots \times \{0,1\}^{m_k} \to \{0,1\}$ by $h(x^1, \ldots, x^k) = f\big(g_1(x^1), \ldots, g_k(x^k)\big)$, Eq. (1.7) generalizes to give

$$\mathrm{Adv}^{\pm}(h) = \mathrm{Adv}^{\pm}_s(f) \ , \tag{1.10}$$

where $s_j = \mathrm{Adv}^{\pm}(g_j)$ [HLŠ07, Rei10a]. This equation implies that we can calculate the general adversary bound, and hence the quantum query complexity, of the entire formula φ merely by computing weighted general adversary bounds for individual functions in $\mathcal{S}$. For example, one can compute the weighted general adversary bounds for the n-bit AND and OR functions as $\mathrm{Adv}^{\pm}_{s}(\mathrm{OR}_n) = \mathrm{Adv}^{\pm}_{s}(\mathrm{AND}_n) = \sqrt{\sum_j s_j^2}$ [BS04]. It thus follows:

Corollary 1.6. *The general adversary bound and quantum query complexity of a read-once AND-OR formula with n inputs are $\sqrt{n}$ and $\Theta(\sqrt{n})$, respectively.*

Previous work has shown that the quantum query complexity of evaluating a read-once AND-OR formula on n inputs is $\sqrt{n}\,2^{O(\sqrt{\log n})}$ [ACR+10, FGG08], and has characterized the query complexity of "adversary-balanced" formulas when the set $\mathcal{S}$ has constant size [RŠ08]. This and other previous and subsequent work on formula-evaluation will be discussed in Section 1.3 below.

Classically, the randomized query complexity even of AND-OR formulas is unknown, except in the "well-balanced" case [Sni85, SW86, San95], and the best general lower bound is $\Omega(n^{0.51})$ [HW91], conjectured to be quite loose. This emphasizes the power of Eq. (1.10) for quantum query complexity.

Furthermore, the quantum query algorithm for evaluating $f \bullet g$ in Theorem 1.5 is not the composition of an algorithm for evaluating f with an algorithm for evaluating g. Instead, it corresponds to a solution to the general adversary bound dual SDP for $f \bullet g$, i.e., a span program, that is a certain composition of dual SDP solutions—span programs—for f and for g. This notion of composition is natural, but also very different from the usual way of composing classical or quantum algorithms. See [RŠ08, Rei09c, Rei09b, Rei09a] for more details on span program composition.

1.2 Reflection structure of the algorithm

The algorithm behind Theorem 1.3 has a simple structure. On input x, the algorithm applies in alternation the input oracle O_x and a certain fixed reflection, which is specified precisely in Section 3. In fact, the input oracle is itself a reflection, $O_x^2 = \mathbf{1}$. It follows that every boolean function can be evaluated optimally, with bounded error, by alternating two fixed reflections. This is similar to Grover's search algorithm, in which the state vector rotates through a certain two-dimensional space. During our algorithm, the state vector rotates steadily through each of *many* orthogonal, reflection-invariant, two-dimensional spaces.

Two curious properties of this algorithm are that the operations between oracle queries are *reflections*, and that they are all the *same*. Neither property is surprising on its own. Indeed, considering an arbitrary quantum query algorithm, shown in Figure 1, we may add a clock register and define $V = \sum_{q=0}^{Q} |q+1\rangle\langle q| \otimes V_q$; when the clock reads q, V applies V_q and increments the clock (mod $Q+1$). Then the algorithm that alternates V and O_x has the same effect as the original algorithm. Similarly, if some V_q is not a reflection, we may add a one-qubit register for that q, and then use the reflection $\tilde{V}_q = |1\rangle\langle 0| \otimes V_q + |0\rangle\langle 1| \otimes V_q^{\dagger}$ instead of V_q. However, if we combine these two algorithmic transformations, in either order, then the desired property from the first transformation is lost. A black-box conversion of an algorithm into the two-fixed-reflections form, that preserves the completeness and soundness parameters, does not appear to be possible.

The second reflection is about the eigenvalue-zero subspace of the adjacency matrix A_G for a certain graph G derived from a dual SDP for $\mathrm{Adv}^{\pm}$. A previous algorithm, in [Rei09a], roughly simulates A_G as a Hamiltonian in continuous time. The problem it faces is that the norm of A_G can be super-constant, which slows the simulation. Fortunately, a relationship between the discrete-

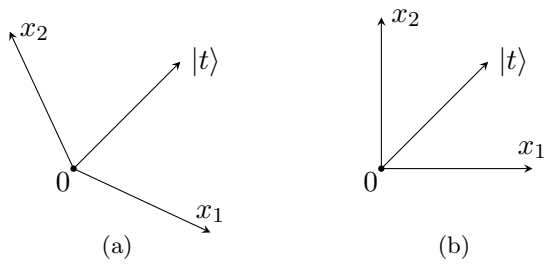

Figure 2: The span programs in (a) and (b) each consist of a target vector $|t\rangle$ and two input vectors, labeled by the literals x_1 and x_2. Each span program computes the function $x_1 \wedge x_2$, since a linear combination of *both* input vectors is needed to span the target. The span program in (b) has a smaller witness size, though, because a shorter combination of the input vectors spans the target.

and continuous-time quantum query complexity models, from [CGM+09], allows for removing the norm dependency, at a sub-logarithmic cost to the query complexity. By instead reflecting about the eigenvalue-zero subspace of A_G, we efficiently remove the dependence on higher-energy portions of A_G. This new approach is inspired by Ambainis's AND-OR formula-evaluation algorithm [Amb07].

Intuitively, the underlying reason why the two-reflections structure is possible seems to come from the simple form of the general adversary bound SDP, Eq. (1.2). Since neither this SDP nor its dual has different terms for different query times, an algorithm based on the dual SDP is naturally symmetrical, without requiring the above clock-register trick. Note that in order to match the general adversary lower bound, most queries made by the algorithm must be roughly equally effective, in order to learn greedily the maximum possible amount of information, $\max_{j\in[n]} \|\Gamma^{(j)}\|$.

Although aesthetically appealing, the two-reflections property has no immediate practical implications. In Section 9 we speculate on its use for fault-tolerant function evaluation. Essentially, the algorithm's structured form may allow for other black-box algorithm transformations. While known algorithms can in principle be converted into this two-reflections form [Rei09a, Theorems 3.1, 5.2], we do not know an explicit closed form for the second reflection, e.g., for the collision problem.

1.3 Span programs

As explained in Section 1.1, a main application of Theorem 1.3 is to evaluating formulas. In fact, the formula-evaluation problem has fostered the development of Theorem 1.3, beginning with Farhi, Goldstone and Gutmann's nearly optimal quantum algorithm for evaluating balanced binary AND-OR formulas [FGG08, ACR+10]. Reichardt and Špalek [RŠ08] extend the allowed gate set considerably by drawing a connection to span programs, a computational model introduced by Karchmer and Wigderson [KW93]. A span program defines a function according to whether subsets of "input vectors" span a certain "target vector," and the witness size complexity measure, from [RŠ08], roughly measures how long a combination is needed to reach the target. Figure 2 shows two examples, and the formal definitions are in Section 6. Ref. [RŠ08] gives a quantum algorithm for evaluating certain *composed* span programs, with a query complexity upper-bounded by the witness size. Thus in fact the allowed formula 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 $\text{wsize}(P) = \text{Adv}^{\pm}(f)$.

Subsequent work [Rei10a] has provided a systematic method for finding optimal span programs. Ref. [Rei10a] shows that it suffices to consider so-called "canonical" span programs, a specialization

introduced by [KW93], and then derives an SDP for optimizing the witness size of canonical span programs. Remarkably, the SDP value corresponds exactly to the general adversary bound:

Theorem 1.7 ([Rei10a]). *For any function $f : \mathcal{D} \to \{0,1\}$, with $\mathcal{D} \subseteq \{0,1\}^n$,*

$$\inf_P \text{wsize}(P, \mathcal{D}) = \text{Adv}^{\pm}(f) \ , \tag{1.11}$$

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 constant-size gate set.

Unfortunately, for reasons explained in Section 1.4 below, the algorithm from [RŠ08] can only evaluate composed constant-size span programs, not general span programs. The new result behind Theorem 1.3 is a quantum algorithm for evaluating arbitrary span programs, with only a constant query overhead on the witness size. For simplicity, we will present the algorithm and prove its correctness, in Sections 3 through 5, based directly on a solution to the dual $\text{Adv}^{\pm}$ SDP, without going through the span program formalism. From Theorems 1.3 and 1.7 it then follows:

Corollary 1.8. *For any function $f : \mathcal{D} \to \{0,1\}$, with $\mathcal{D} \subseteq \{0,1\}^n$,*

$$\inf_P \text{wsize}(P, \mathcal{D}) = \Theta\big(Q(f)\big) \ , \tag{1.12}$$

where the infimum is over span programs P computing f. Therefore, up to a constant factor, span programs are equivalent to quantum query algorithms.

Although we prove Theorem 1.3 directly, in order to understand the algorithm's origins it is still useful to study span programs. Therefore, in Sections 6 and 7, we define span programs and their correspondence with bipartite graphs. We also define two complexity measures, the witness size and the full witness size. For an optimal canonical span program, these measures differ by at most one.

In Section 8, we will present another quantum algorithm for evaluating a span program P, that uses $O\big(\text{fwsize}(P)\|\text{abs}(A_{G_P})\|\big)$ queries to evaluate f, where $\text{fwsize}(P)$ is the full witness size and $\text{abs}(A_{G_P})$ is the entry-wise absolute value of the weighted adjacency matrix for the corresponding bipartite graph (Theorem 8.1). The algorithm is based on Szegedy's quantum walk model [Sze04], which also uses two reflections. This alternative algorithm is useful since it often has a *time*-efficient implementation, when $\|\text{abs}(A_{G_P})\|$ and the maximum degree of G_P can be bounded. There is considerable freedom in designing a span program for a function, and these quantities should be minded. Subsequent work has applied Theorem 8.1 to derive a query-optimal and nearly time-optimal quantum algorithm for evaluating a large class of read-once formulas over any finite gate set [Rei09b], and to derive a nearly query- and time-optimal quantum algorithm for evaluating arbitrary read-once AND-OR formulas [Rei09c].

1.4 Analysis of the algorithm

The main technical difficulty in the proof of Theorem 1.3 is analyzing the spectrum of a certain bipartite graph. In Section 4, we show that properties of eigenvalue-zero eigenvectors of bipartite graphs in fact imply an "effective" spectral gap around zero for perturbed graphs. This small-eigenvalue analysis is the key step that allows us to evaluate span programs on a quantum computer.

The [RŠ08] formula-evaluation algorithm works by plugging together optimal span programs for the individual gates in a formula 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 $1/\text{wsize}(P)$ steps in order to compute $\varphi(x)$. 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 the algorithm can detect a stationary component in the walk. Second, for soundness, it is shown that if $\varphi(x) = 0$, then $A_{G_P(x)}$ has a spectral gap of $\Omega(1/\text{wsize}(P))$ for eigenvectors supported on the output vertex. The inverse spectral gap, as a kind of mixing time, 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 [RŠ08] 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, the approach fails in general. Span programs can be constructed for which the associated graphs simply do not have an $\Omega(1/\text{wsize}(P))$ spectral gap in the 0 case. (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 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 the formula-evaluation results in [FGG08, ACR+10, RŠ08].

This article is based on two arXiv preprints: a portion of [Rei09a], and [Rei10b]. The technical report [Rei10a] contains another portion of the former preprint's results, namely the connection between the general adversary bound and optimal span program witness size (Theorem 1.7).

2 Definitions

For a natural number $n \in \mathbf{N}$, let $[n] = \{1, 2, \ldots, n\}$. For a bit $b \in \{0, 1\}$, let $\bar{b} = 1 - b$. For a finite set X, let $\mathbf{C}^X$ be the Hilbert 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 $\mathbf{C}^X$. 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 $\mathcal{L}(V) = \mathcal{L}(V, V)$. $\|A\|$ is the spectral norm of an operator A.

A weighted bipartite graph G can be specified by its weighted biadjacency matrix B_G. G has a

vertex for every row and for every column of B_G, and edges between the row and column vertices have weights specified by the matrix entries. The weighted adjacency matrix of G is

$$A_G = \begin{pmatrix} 0 & B_G \\ B_G^\dagger & 0 \end{pmatrix} \ . \tag{2.1}$$

3 The algorithms

For a boolean function, taking the dual of the general adversary bound SDP in Definition 1.1 gives:

Lemma 3.1 ([Rei10a, Theorem 4.4])**.** *Let $f : \mathcal{D} \to \{0,1\}$, with $\mathcal{D} \subseteq \{0,1\}^n$. For $b \in \{0,1\}$, let $F_b = \{x \in \mathcal{D} : f(x) = b\}$. Then*

$$\mathrm{Adv}^{\pm}(f) = \min_{\substack{m\in\mathbf{N},\{|v_{xj}\rangle\in\mathbf{C}^m : x\in\mathcal{D}, j\in[n]\}: \\ \forall(x,y)\in F_0\times F_1, \sum_{j\in[n]:x_j\neq y_j}\langle v_{xj}|v_{yj}\rangle=1}} \max_{x\in\mathcal{D}} \sum_{j\in[n]} \||v_{xj}\rangle\|^2 \ . \tag{3.1}$$

Based on a feasible solution to this SDP with objective value $W(\geq 1)$, we will give three algorithms for evaluating f, each with query complexity $O(W)$. (A feasible solution corresponds to a span program in canonical form, and its value equals the span program witness size [Rei10a].)

Let $I = [n] \times \{0,1\} \times [m]$. Let $|t\rangle \in \mathbf{C}^{F_0}$ and $A \in \mathcal{L}(\mathbf{C}^I, \mathbf{C}^{F_0})$ be given by

$$\begin{aligned} |t\rangle &= \frac{1}{3\sqrt{W}} \sum_{x\in F_0} |x\rangle \\ A &= \sum_{x\in F_0, j\in[n]} |x\rangle\langle j, \bar{x}_j| \otimes \langle v_{xj}| \ . \end{aligned} \tag{3.2}$$

Let G be the weighted bipartite graph with biadjacency matrix $B_G \in \mathcal{L}(\mathbf{C}^{\{\emptyset\}} \oplus \mathbf{C}^I, \mathbf{C}^{F_0})$:

$$B_G = \begin{pmatrix} |t\rangle & A \end{pmatrix} \ . \tag{3.3}$$

The vertex set of G is the disjoint union $F_0 \cup \{\emptyset\} \cup I$.

Let $\Delta \in \mathcal{L}(\mathbf{C}^{F_0\cup\{\emptyset\}\cup I})$ be the orthogonal projection onto the span of all eigenvalue-zero eigenvectors of the weighted adjacency matrix A_G. For an input $x \in \mathcal{D}$, let $\Pi_x \in \mathcal{L}(\mathbf{C}^{F_0\cup\{\emptyset\}\cup I})$ be the projection

$$\Pi_x = \mathbf{1} - \sum_{j\in[n],k\in[m]} |j, \bar{x}_j, k\rangle\langle j, \bar{x}_j, k| \ . \tag{3.4}$$

That is, Π_x is a diagonal matrix that projects onto all vertices except those associated to the input bit complements $\bar{x}_j$. Finally, let

$$U_x = (2\Pi_x - \mathbf{1})(2\Delta - \mathbf{1}) \ . \tag{3.5}$$

U_x consists of the two reflections $2\Delta - \mathbf{1}$ and $2\Pi_x - \mathbf{1}$. The first reflection does not depend on the input x. The second reflection can be implemented using a single call to the input oracle O_x.

We present three related algorithms, each slightly simpler than the one before:

Algorithm 1:

1. Prepare the initial state $|\emptyset\rangle \in \mathbf{C}^{F_0 \cup \{\emptyset\} \cup I}$.
2. Run phase estimation on U_x, with precision $\delta_p = \frac{1}{100W}$ and error rate $\delta_e = \frac{1}{10}$.
3. Output 1 if the measured phase is zero. Otherwise output 0.

Algorithm 2:

1. Prepare the initial state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |\emptyset\rangle \in \mathbf{C}^2 \otimes \mathbf{C}^{F_0 \cup \{\emptyset\} \cup I}$.
2. Pick $T \in [\lceil 100W \rceil]$ uniformly at random. Apply the controlled unitary $|0\rangle\langle 0| \otimes \mathbf{1} + |1\rangle\langle 1| \otimes U_x^T$.
3. Measure the first register in the basis $\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. Output 1 if the measurement result is $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and output 0 otherwise.

Algorithm 3:

1. Prepare the initial state $|\emptyset\rangle \in \mathbf{C}^{F_0 \cup \{\emptyset\} \cup I}$.
2. Pick $T \in [\lceil 10^5 W \rceil]$ uniformly at random. Apply U_x^T.
3. Measure the vertex. Output 1 if the measurement result is $|\emptyset\rangle$, and output 0 otherwise.

Phase estimation on a unitary V with precision δ_p and error rate δ_e can be implemented using $O\left(\frac{1}{\delta_p} \log \frac{1}{\delta_e}\right)$ controlled applications of V [KOS07, NWZ09], so the first algorithm has $O(W)$ query complexity. The second algorithm essentially applies a simplified version of phase estimation. Intuitively, it works because it suffices to distinguish zero from nonzero phase. The third algorithm does away with any phase estimation. Intuitively, this is possible because U_x is the product of two reflections, so its spectrum is symmetrical. The second and third algorithms clearly have $O(W)$ query complexity. The factor of 10^5 in the third algorithm's query complexity can be reduced by three orders of magnitude by adjusting downward the scaling factor for $|t\rangle$ in Eq. (3.2).

The time, or number of elementary operations, required to implement the reflection $2\Delta - \mathbf{1}$ is unclear. In practice it may be preferable to use the potentially less query-efficient quantum walk algorithm from Theorem 8.1 below.

In the following two sections, we will show that all three algorithms correctly evaluate $f(x)$, with constant gaps between the soundness error and completeness parameters.

4 Effective spectral gaps for perturbed bipartite graphs

In this section, we give a general argument that relates properties of eigenvalue-zero eigenvectors of weighted bipartite graphs to what are in a certain sense "effective" spectral gaps. As explained in Section 1.4, this small-eigenvalue analysis substantially extends the analysis in [RŠ08].

The main result of this section is:

Theorem 4.1. *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 weighted adjacency matrix A_G has an eigenvalue-zero eigenvector $|\psi\rangle$ with*

$$|\langle t|\psi_T\rangle|^2 \geq \delta \||\psi\rangle\|^2 \ . \tag{4.1}$$

Let G' be the same as G except with a new vertex, $\emptyset$, added to the U side, with outgoing edges weighted by the entries of $|t\rangle$. That is, the biadjacency matrix of G' is

$$B_{G'} = \begin{matrix} \emptyset & U \end{matrix} \\ \begin{pmatrix} |t\rangle & B_G \end{pmatrix} T \tag{4.2}$$

Let $\{|\alpha\rangle\}$ be a complete set of orthonormal eigenvectors of the weighted adjacency matrix $A_{G'}$, with corresponding eigenvalues $\rho(\alpha)$. Then for all $\Upsilon \geq 0$, the squared length of the projection of $|\emptyset\rangle$ onto the span of the eigenvectors α with $|\rho(\alpha)| \leq \Upsilon$ satisfies

$$\sum_{\alpha:\, |\rho(\alpha)| \leq \Upsilon} |\langle \alpha|\emptyset\rangle|^2 \leq 8\Upsilon^2/\delta \ . \tag{4.3}$$

To motivate our approach to proving Theorem 4.1, let us recall some basic properties about the eigenvalues and eigenvectors of bipartite graphs.

Proposition 4.2. *Let G be a weighted bipartite graph with biadjacency matrix B_G and adjacency matrix $A_G = \begin{pmatrix} 0 & B_G \\ B_G^\dagger & 0 \end{pmatrix}$.*

Assume that $|\psi\rangle = (|\psi_T\rangle, |\psi_U\rangle) \in \mathbf{C}^T \oplus \mathbf{C}^U$ is an eigenvalue-ρ eigenvector of A_G, for some $\rho \neq 0$. Then $(|\psi_T\rangle, -|\psi_U\rangle)$ is an eigenvector of A_G with eigenvalue $-\rho$. Moreover, $|\psi_T\rangle = \frac{1}{\rho} B_G |\psi_U\rangle$ is an eigenvector of $B_G B_G^\dagger$ and $|\psi_U\rangle = \frac{1}{\rho} B_G^\dagger |\psi_T\rangle$ is an eigenvector of $B_G^\dagger B_G$, both with corresponding eigenvalues ρ^2.

Conversely, if $|\varphi\rangle \in \mathbf{C}^T$ is an eigenvalue-λ eigenvector of $B_G B_G^\dagger$ for $\lambda > 0$, then $B_G^\dagger |\varphi\rangle \in \mathbf{C}^U$ is an eigenvalue-λ eigenvector of $B_G^\dagger B_G$ and $|\psi_\pm\rangle = (|\varphi\rangle, \pm\frac{1}{\sqrt{\lambda}} B_G^\dagger |\varphi\rangle) \in \mathbf{C}^T \oplus \mathbf{C}^U$ are eigenvectors of A_G with corresponding eigenvalues $\pm\sqrt{\lambda}$.

The proof is immediate.

Thus the spectrum of A_G is symmetrical around zero, and nonzero-eigenvalue eigenvectors of the positive semi-definite matrix $B_G B_G^\dagger$ are in exact correspondence to symmetrical pairs of nonzero-eigenvalue eigenvectors of A_G.

Proposition 4.2 allows us to translate the claims of Theorem 4.1 into claims on spectral properties of positive semi-definite matrices. We will start, though, by proving the necessary result for positive semi-definite matrices, Theorem 4.3 below. After proving Theorem 4.3, we will give the translation to prove Theorem 4.1.

Theorem 4.3. *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 corresponding eigenvalues $\lambda(\beta) \geq 0$. Assume that there exists a $|\varphi\rangle \in \mathrm{Ker}(X)$ with $|\langle t|\varphi\rangle|^2 \geq \delta \||\varphi\rangle\|^2$. Then for any $\Lambda \geq 0$,*

$$\delta \sum_{\substack{\beta:\, \lambda(\beta) \leq \Lambda \\ \langle t|\beta\rangle \neq 0}} \frac{1}{\lambda(\beta)} |\langle t|\beta\rangle|^2 \leq 4\Lambda \ . \tag{4.4}$$

Proof. The sum is well-defined, with no division by zero, because any $|\beta\rangle$ with $\langle t|\beta\rangle \neq 0$ must have $\lambda(\beta) = \langle\beta|X'|\beta\rangle = \langle\beta|X|\beta\rangle + |\langle t|\beta\rangle|^2 > 0$.

The key lemma for proving Theorem 4.3 is:

Lemma 4.4. *Under the conditions of Theorem 4.3, for any $|\xi\rangle \in V$,*

$$\delta|\langle t|\xi\rangle|^2 \leq \|X'|\xi\rangle\|^2 \ . \tag{4.5}$$

Moreover, if $|\xi\rangle$ is a linear combination of eigenvectors with corresponding eigenvalues at most κ, i.e., $|\xi\rangle = \sum_{\beta:\lambda(\beta)\leq\kappa} \langle\beta|\xi\rangle|\beta\rangle$, then

$$\delta|\langle t|\xi\rangle|^2 \leq \kappa^2\||\xi\rangle\|^2 \ . \tag{4.6}$$

Proof. We will write the matrices X and X' out in coordinates. Fixing $\langle t|\xi\rangle$, we will use straightforward calculus to minimize $\|X'|\xi\rangle\|^2$.

Let $|1\rangle, \ldots, |m\rangle$ be a complete, orthonormal set of eigenvectors for $\big(\mathbf{1} - \frac{|t\rangle\langle t|}{\||t\rangle\|^2}\big)X\big(\mathbf{1} - \frac{|t\rangle\langle t|}{\||t\rangle\|^2}\big)$, with corresponding eigenvalues $a_1, \ldots, a_m$. In the coordinates $\big(\frac{|t\rangle}{\||t\rangle\|}, |1\rangle, \ldots, |m\rangle\big)$, X and X' are given by

$$X = \begin{pmatrix} a & \bar{b}_1 & \ldots & \bar{b}_m \\ b_1 & a_1 & & 0 \\ \vdots & & \ddots & \\ b_m & 0 & & a_m \end{pmatrix} \qquad X' = \begin{pmatrix} a + \||t\rangle\|^2 & \bar{b}_1 & \ldots & \bar{b}_m \\ b_1 & a_1 & & 0 \\ \vdots & & \ddots & \\ b_m & 0 & & a_m \end{pmatrix} \tag{4.7}$$

where $a = \langle t|X|t\rangle/\||t\rangle\|^2$ and $b_j = \langle a_j|X\frac{|t\rangle}{\||t\rangle\|}$, for $j \in [m]$.

By incorporating any phases into the basis vectors $|j\rangle$, we may assume that all $b_j \geq 0$. Furthermore, we may assume without loss of generality that all $b_j > 0$. Indeed, if some $b_j = 0$, then the $|j\rangle$ coordinate lies in a different block of X' from $|t\rangle$, so removing this coordinate will not affect $\min_{|\psi\rangle} \|X'|\psi\rangle\|/|\langle t|\psi\rangle|$. Since $X \succeq 0$, all $a_j \geq 0$. Moreover, if some $a_j = 0$, then since $\left(\begin{smallmatrix} a & b_j \\ b_j & 0 \end{smallmatrix}\right)$ is a (positive semi-definite) submatrix of X, it must be that $b_j = 0$. Hence we may assume that $a_j > 0$ for all $j \in [m]$.

We are given the existence of a $|\varphi\rangle \in \mathrm{Ker}(X)$ with $|\langle t|\varphi\rangle|^2 \geq \delta\||\varphi\rangle\|^2$. Let us write out this condition in coordinates. By scaling $|\varphi\rangle$, we may assume that $\langle t|\varphi\rangle = \||t\rangle\|$. Thus, written in coordinates, $|\varphi\rangle = (1, -\frac{b_1}{a_1}, \ldots, -\frac{b_m}{a_m})$ and $\langle t|X|\varphi\rangle = 0$ implies that

$$a = \sum_{j=1}^{m} b_j^2/a_j \ . \tag{4.8}$$

The condition $|\langle t|\varphi\rangle|^2 \geq \delta\||\varphi\rangle\|^2$, in coordinates, is

$$\||t\rangle\|^2 \geq \delta\Big(1 + \sum_{j=1}^{m}\Big(\frac{b_j}{a_j}\Big)^2\Big) \ . \tag{4.9}$$

We can now solve the minimization problem:

Claim 4.5.

$$\min_{|\xi\rangle:\,\langle t|\xi\rangle = \||t\rangle\|} \|X'|\xi\rangle\|^2 = \frac{\||t\rangle\|^4}{1 + \sum_j \big(\frac{b_j}{a_j}\big)^2} \geq \delta\||t\rangle\|^2 \ . \tag{4.10}$$

Proof. Since X' is a symmetric matrix, we may assume that $|\xi\rangle$ has real coordinates. Introduce variables $c_1, \ldots, c_m$ and let $|\xi\rangle = (1, c_1, \ldots, c_m)$. For $j \in [m]$, let $\gamma_j = a_j\big(\frac{a_j}{b_j}c_j + 1\big)$. Then

$$\begin{aligned}
\|X'|\xi\rangle\|^2 &= \big(a + \||t\rangle\|^2 + \sum_j b_j c_j\big)^2 + \sum_j (b_j + a_j c_j)^2 \\
&= \left(a + \||t\rangle\|^2 + \sum_j \frac{b_j^2}{a_j}\Big(\frac{\gamma_j}{a_j} - 1\Big)\right)^2 + \sum_j \Big(\frac{b_j}{a_j}\gamma_j\Big)^2 \\
&= \left(\||t\rangle\|^2 + \sum_j \Big(\frac{b_j}{a_j}\Big)^2 \gamma_j\right)^2 + \sum_j \Big(\frac{b_j}{a_j}\Big)^2 \gamma_j^2 \ , \qquad (4.11)
\end{aligned}$$

where we have substituted $c_j = \frac{b_j}{a_j}\big(\frac{\gamma_j}{a_j} - 1\big)$ and then used Eq. (4.8) to cancel a from the first term.

A global minimum exists and will satisfy, for all $j \in [m]$,

$$\begin{aligned}
0 &= \frac{\partial}{\partial \gamma_j}\|X'|\xi\rangle\|^2 \\
&= 2\Big(\frac{b_j}{a_j}\Big)^2\left(\gamma_j + \||t\rangle\|^2 + \sum_k \Big(\frac{b_k}{a_k}\Big)^2 \gamma_k\right) \ . \qquad (4.12)
\end{aligned}$$

Thus we should set all γ_j equal, $\gamma_j = \gamma$ for $j \in [m]$, where $\gamma = -\||t\rangle\|^2/(1+S)$ and $S = \sum_j \big(\frac{b_j}{a_j}\big)^2$. Substituting back into Eq. (4.11), $\|X'|\xi\rangle\|^2$ at the minimum is

$$\begin{aligned}
\|X'|\xi\rangle\|^2 &= (\||t\rangle\|^2 + S\gamma)^2 + S\gamma^2 \\
&= \||t\rangle\|^4/(1+S) \ , \qquad (4.13)
\end{aligned}$$

as claimed. □

Eq. (4.5) follows. Eq. (4.6) is an immediate consequence, since $|\xi\rangle = \sum_{\beta:\lambda(\beta)\le\kappa} \langle\beta|\xi\rangle|\beta\rangle$ implies $\|X'|\xi\rangle\| \le \kappa\||\xi\rangle\|$. This completes the proof of Lemma 4.4. □

Now let us derive Eq. (4.4) by bootstrapping Lemma 4.4. We aim to bound

$$\begin{aligned}
\delta \sum_{\substack{\beta:\,\lambda(\beta)\le\Lambda \\ \langle t|\beta\rangle\neq 0}} \frac{1}{\lambda(\beta)}|\langle t|\beta\rangle|^2 &= \delta \sum_{k=0}^{\infty} \sum_{\frac{\Lambda}{2^{k+1}} < \lambda(\beta) \le \frac{\Lambda}{2^k}} \frac{1}{\lambda(\beta)}|\langle t|\beta\rangle|^2 \\
&\le \frac{\delta}{\Lambda}\sum_{k=0}^{\infty} 2^{k+1} \sum_{\frac{\Lambda}{2^{k+1}} < \lambda(\beta) \le \frac{\Lambda}{2^k}} |\langle t|\beta\rangle|^2 \\
&= \frac{\delta}{\Lambda}\sum_{k=0}^{\infty} 2^{k+1}\langle t|t_k\rangle \ , \qquad (4.14)
\end{aligned}$$

where $|t_k\rangle = \sum_{\beta:\frac{\Lambda}{2^{k+1}}<\lambda(\beta)\le\frac{\Lambda}{2^k}} \langle\beta|t\rangle|\beta\rangle$, the projection of $|t\rangle$ onto the span of the eigenvectors with eigenvalues in $\left(\frac{\Lambda}{2^{k+1}}, \frac{\Lambda}{2^k}\right]$. Therefore $\langle t|t_k\rangle = \langle t_k|t_k\rangle = |\langle t|t_k\rangle|^2/\||t_k\rangle\|^2$ when $|t_k\rangle \neq 0$, so Eq. (4.6) can be applied with $|\xi\rangle = |t_k\rangle$ and $\kappa = \Lambda/2^k$ to continue:

$$\begin{aligned}\delta \sum_{\substack{\beta:\,\lambda(\beta)\le\Lambda \\ \langle t|\beta\rangle\neq 0}} \frac{1}{\lambda(\beta)}|\langle t|\beta\rangle|^2 &\le \frac{1}{\Lambda}\sum_{k=0}^{\infty} 2^{k+1}\left(\frac{\Lambda}{2^k}\right)^2 \\ &= 2\Lambda\sum_{k=0}^{\infty}\frac{1}{2^k} \\ &= 4\Lambda \ , \end{aligned} \tag{4.15}$$

as claimed. □

With Theorem 4.3 in hand, we can now apply Proposition 4.2 to prove Theorem 4.1.

Proof of Theorem 4.1. We are given an eigenvalue-zero eigenvector of A_G, $(|\psi_T\rangle, 0) \in \mathbf{C}^T \oplus \mathbf{C}^U$ with $|\langle t|\psi_T\rangle|^2 \ge \delta\||\psi_T\rangle\|^2$. In particular, $B_G^\dagger|\psi_T\rangle = 0$.

An eigenvalue-zero eigenvector $|\zeta\rangle = (|\zeta_T\rangle, \zeta_\emptyset, |\zeta_U\rangle) \in \mathbf{C}^T \oplus \mathbf{C}^{\{\emptyset\}} \oplus \mathbf{C}^U$ has to satisfy

$$\begin{aligned} 0 &= B_{G'}(\zeta_\emptyset, |\zeta_U\rangle) \\ &= \zeta_\emptyset|t\rangle + B_G|\zeta_U\rangle \ . \end{aligned} \tag{4.16}$$

Since $|\langle t|\psi_T\rangle|^2 > 0$ and $B_G^\dagger|\psi_T\rangle = 0$, $|t\rangle$ cannot lie in the range of B_G, so $\zeta_\emptyset$ must be zero. Thus follows the claim for $\Upsilon = 0$, that $A_{G'}$ has no eigenvalue-zero eigenvectors supported on $|\emptyset\rangle$.

Now to show Eq. (4.3) for $\Upsilon > 0$, note that for each eigenvector $|\alpha\rangle$ of $A_{G'}$, $\rho(\alpha)\langle\emptyset|\alpha\rangle = \langle\emptyset|A_{G'}|\alpha\rangle = \langle t|\alpha_T\rangle$. Therefore

$$\sum_{\alpha:\,|\rho(\alpha)|\le\Upsilon} |\langle\alpha|\emptyset\rangle|^2 = \sum_{\alpha:\,0<|\rho(\alpha)|\le\Upsilon} \frac{1}{\rho(\alpha)^2}|\langle t|\alpha_T\rangle|^2 \ . \tag{4.17}$$

Let $X' = B_{G'}B_{G'}^\dagger$. Let $\{|\beta\rangle\}$ be a complete set of orthonormal eigenvectors of X', with corresponding eigenvalues $\lambda(\beta)$. By Proposition 4.2, each eigenvector $|\beta\rangle$ with $\lambda(\beta) \neq 0$ corresponds to a pair of eigenvectors of $A_{G'}$ with eigenvalues $\pm\sqrt{\lambda(\beta)}$. The above sum therefore equals

$$2\sum_{\beta:\,0<\lambda(\beta)\le\Upsilon^2} \frac{1}{\lambda(\beta)}|\langle t|\beta\rangle|^2 \ . \tag{4.18}$$

Now apply Theorem 4.3 with $X = X' - |t\rangle\langle t| = B_G B_G^\dagger \succeq 0$, $|\varphi\rangle = |\psi_T\rangle$ and $\Lambda = \Upsilon^2$, to obtain the claimed upper bound of $8\Upsilon^2/\delta$. □

5 Analysis of the algorithms

To analyze the algorithms from Section 3, we shall study the spectrum of the unitary $U_x = (2\Pi_x - \mathbf{1})(2\Delta - \mathbf{1})$.

For this purpose, it will be useful to introduce two new graphs, following [RŠ08, Rei09a]. Let $\overline{\Pi}(x) = \sum_{j\in[n]} |j\rangle\langle j| \otimes |\bar{x}_j\rangle\langle\bar{x}_j| \otimes \mathbf{1}_{\mathbf{C}^{[m]}} \in \mathcal{L}(\mathbf{C}^I)$, and let $G(x)$ and $G'(x)$ be the weighted bipartite graphs with biadjacency matrices

$$B_{G(x)} = \begin{pmatrix} |t\rangle & A \\ 0 & \overline{\Pi}(x) \end{pmatrix} \qquad \text{and} \qquad B_{G'(x)} = \begin{pmatrix} A \\ \overline{\Pi}(x) \end{pmatrix} \ . \tag{5.1}$$

Based on the constraints of the SDP in Lemma 3.1, we can immediately construct eigenvalue-zero eigenvectors for $G(x)$ or $G'(x)$, depending on whether $f(x) = 1$ or $f(x) = 0$:

Lemma 5.1. *If $f(x) = 1$, then the vector*

$$|\psi\rangle = -3\sqrt{W}|\emptyset\rangle + \sum_{j\in[n]} |j, x_j\rangle \otimes |v_{xj}\rangle \in \mathbf{C}^{\{\emptyset\}\cup I} \tag{5.2}$$

satisfies $B_{G(x)}|\psi\rangle = 0$ and $|\langle\emptyset|\psi\rangle|^2/\||\psi\rangle\|^2 \geq 9/10$.

If $f(x) = 0$, then

$$|\psi\rangle = -|x\rangle + \sum_{j\in[n]} |j, \bar{x}_j\rangle \otimes |v_{xj}\rangle \in \mathbf{C}^{F_0\cup I} \tag{5.3}$$

satisfies $B^\dagger_{G'(x)}|\psi\rangle = 0$ and $|\langle t|\psi\rangle|^2/\||\psi\rangle\|^2 \geq 1/(9W(W+1))$.

Substituting Lemma 5.1 into Theorem 4.1, we thus obtain the key statement:

Lemma 5.2. *If $f(x) = 1$, then $A_{G(x)}$ has an eigenvalue-zero eigenvector $|\psi\rangle$, supported on the column vertices, with*

$$\frac{|\langle\emptyset|\psi\rangle|^2}{\||\psi\rangle\|^2} \geq \frac{9}{10} \ . \tag{5.4}$$

If $f(x) = 0$, let $\{|\alpha\rangle\}$ be a complete set of orthonormal eigenvectors of $A_{G(x)}$ with corresponding eigenvalues $\rho(\alpha)$. Then for any $c \geq 0$,

$$\sum_{\alpha:|\rho(\alpha)|\leq c/W} |\langle\alpha|\emptyset\rangle|^2 \leq 72\Big(1 + \frac{1}{W}\Big)c^2 \ . \tag{5.5}$$

By choosing c a small positive constant, Eq. (5.5) gives an $O(1/W)$ "effective spectral gap" for eigenvectors of $A_{G(x)}$ supported on $|\emptyset\rangle$; it says that $|\emptyset\rangle$ has small squared overlap on the subspace of $O(1/W)$-eigenvalue eigenvectors.

It remains to translate Lemma 5.2 into analogous statements for U_x:

Lemma 5.3. *If $f(x) = 1$, then U_x has an eigenvalue-one eigenvector $|\varphi\rangle$ with*

$$\frac{|\langle\emptyset|\varphi\rangle|^2}{\||\varphi\rangle\|^2} \geq \frac{9}{10} \ . \tag{5.6}$$

If $f(x) = 0$, let $\{|\beta\rangle\}$ be a complete set of orthonormal eigenvectors of U_x with corresponding eigenvalues $e^{i\theta(\beta)}$, $\theta(\beta) \in (-\pi, \pi]$. Then for any $\Theta \geq 0$,

$$\sum_{\beta:|\theta(\beta)|\leq\Theta} |\langle\beta|\emptyset\rangle|^2 \leq \Big(2\sqrt{6\Theta W} + \frac{\Theta}{2}\Big)^2 \ . \tag{5.7}$$

Assuming Lemma 5.3, the algorithms from Section 3 are both complete and sound. If $f(x) = 1$, then the first, phase-estimation-based algorithm outputs 1 with probability at least $9/10 - \delta_e = 4/5$. If $f(x) = 0$, then setting $\Theta = \delta_p = \frac{1}{100W}$, the algorithm outputs 1 with probability at most $\delta_e + (2\sqrt{6\Theta W} + \frac{\Theta}{2})^2 < 2/5$. The probability the second algorithm outputs 1 is the expectation versus T of $\frac{1}{4}\|(\mathbf{1} + U_x^T)|\phi\rangle\|^2$. If $f(x) = 1$, this is at least 9/10 for all T. If $f(x) = 0$, let $\tau = \lceil 100W \rceil$ and simplify

$$\begin{aligned} \mathrm{E}_{T\in_R[\tau]}\Big[\frac{1}{4}\|(\mathbf{1} + U_x^T)|\phi\rangle\|^2\Big] &= \mathrm{E}_{T\in_R[\tau]}\Big[\frac{1}{4}\sum_\beta |1 + e^{i\theta(\beta)T}|^2 |\langle\phi|\beta\rangle|^2\Big] \\ &= \frac{1}{4}\sum_\beta |\langle\phi|\beta\rangle|^2\Big[2 + \frac{1}{\tau}\Big(\frac{e^{i\theta(\beta)(\tau+1)} - e^{-i\theta(\beta)\tau}}{e^{i\theta(\beta)} - 1} - 1\Big)\Big] \ . \end{aligned} \tag{5.8}$$

Setting $\Theta = \frac{1}{50W}$ and $\xi = (2\sqrt{6\Theta W} + \frac{\Theta}{2})^2$, we see that the algorithm outputs 1 with probability at most $\xi + (1-\xi)\big(\frac{1}{2} + 1/(4\tau \sin\frac{\Theta}{2})\big) < 88\%$ for $W \geq 1$. As its analysis requires more care, we defer consideration of the third algorithm to the end of this section.

For the proof of Lemma 5.3 we will use the following characterization of the eigen-decomposition of the product of reflections, from [NWZ09] using observations by Jordan [Jor75]. Its use is common in quantum computation, e.g., [Sze04, MW05].

Lemma 5.4. *Given two projections Π and Δ, the Hilbert space can be decomposed into orthogonal one- and two-dimensional subspaces invariant under Π and Δ. On the one-dimensional invariant subspaces, $(2\Pi - \mathbf{1})(2\Delta - \mathbf{1})$ acts as either $+1$ or -1. Each two-dimensional subspace is spanned by an eigenvalue-λ eigenvector $|v\rangle$ of $\Delta\Pi\Delta$, with $\lambda \in (0,1)$, and $|v^\perp\rangle = (\mathbf{1} - \Delta)\Pi|v\rangle / \|(\mathbf{1} - \Delta)\Pi|v\rangle\|$. Letting $\theta = 2\arccos\sqrt{\lambda} \in (0,\pi)$, so $\Pi|v\rangle/\|\Pi|v\rangle\| = \cos\frac{\theta}{2}|v\rangle + \sin\frac{\theta}{2}|v^\perp\rangle$, the eigenvectors and corresponding eigenvalues of $(2\Pi - \mathbf{1})(2\Delta - \mathbf{1})$ on this subspace are, respectively,*

$$\frac{|v\rangle \mp i|v^\perp\rangle}{\sqrt{2}} \qquad \text{and} \qquad e^{\pm i\theta} \ . \tag{5.9}$$

Proof of Lemma 5.3. Notice from Eqs. (3.3) and (5.1) that G is naturally a subgraph of $G(x)$. Since $A_G\Delta = 0$ by definition of Δ, $A_{G(x)}\Delta = S(\mathbf{1} - \Pi_x)$, where S is a permutation matrix.

First consider the case $f(x) = 1$. Let $|\varphi\rangle$ be the restriction of $|\psi\rangle$ from Eq. (5.4) to the vertices of G. Since $|\psi\rangle$ has no support on the extra vertices of $G(x)$, $\||\varphi\rangle\| = \||\psi\rangle\|$ and $|\varphi\rangle$ is an eigenvalue-zero eigenvector of A_G; $\Delta|\varphi\rangle = |\varphi\rangle$. Also $\Pi_x|\varphi\rangle = |\varphi\rangle$, so indeed $U_x|\varphi\rangle = |\varphi\rangle$.

Next consider the case $f(x) = 0$. Let

$$|\zeta\rangle = \sum_{\beta:|\theta(\beta)|\leq\Theta} |\beta\rangle\langle\beta|\phi\rangle \tag{5.10}$$

be the projection of $|\phi\rangle$ onto the space of eigenvectors with phase at most Θ in magnitude. Our aim is to upper bound $\||\zeta\rangle\|^2 = \langle\phi|\zeta\rangle = |\langle\phi|\hat{\zeta}\rangle|^2$, where $|\hat{\zeta}\rangle = |\zeta\rangle/\||\zeta\rangle\|$. Notice that $|\hat{\zeta}\rangle$ is supported only on eigenvectors $|\beta\rangle$ with $\theta(\beta) \neq 0$, i.e., on the two-dimensional invariant subspaces of Π_x and Δ. Indeed, if $U_x|\beta\rangle = |\beta\rangle$, then either $|\beta\rangle = \Pi_x|\beta\rangle = \Delta|\beta\rangle$ or $|\beta\rangle = (\mathbf{1} - \Pi_x)|\beta\rangle = (\mathbf{1} - \Delta)|\beta\rangle$. The first possibility implies $A_{G(x)}|\beta\rangle = 0$, so by Eq. (5.5) with $c = 0$, $\langle\phi|\beta\rangle = 0$. In the second possibility, also $\langle\phi|\beta\rangle = \langle\phi|\Pi_x|\beta\rangle = 0$ since $|\phi\rangle = \Pi_x|\phi\rangle$.

We can split $\langle \phi|\hat{\zeta}\rangle$ as

$$\begin{aligned}\langle \phi|\hat{\zeta}\rangle &= \langle \phi|\Delta|\hat{\zeta}\rangle + \langle \phi|\Pi_x(\mathbf{1}-\Delta)|\hat{\zeta}\rangle \\ &\le |\langle \phi|\Delta|\hat{\zeta}\rangle| + |\langle \phi|\Pi_x(\mathbf{1}-\Delta)|\hat{\zeta}\rangle| \\ &\le |\langle \phi|\Delta|\hat{\zeta}\rangle| + \|\Pi_x(\mathbf{1}-\Delta)|\hat{\zeta}\rangle\| \ .\end{aligned} \tag{5.11}$$

Start by bounding the second term, $\|\Pi_x(\mathbf{1}-\Delta)|\hat{\zeta}\rangle\|$. Intuitively, this term is small because $|\hat{\zeta}\rangle$ is supported only on two-dimensional invariant subspaces where Δ and Π_x are close. Indeed, let $|-\beta\rangle = (2\Delta - \mathbf{1})|\beta\rangle$, an eigenvector of A_G with phase $\theta(-\beta) = -\theta(\beta)$. Expanding $|\hat{\zeta}\rangle = \sum_\beta c_\beta|\beta\rangle$,

$$\begin{aligned}\|\Pi_x(\mathbf{1}-\Delta)|\hat{\zeta}\rangle\|^2 &= \|\sum_\beta \Pi_x(\mathbf{1}-\Delta)c_\beta|\beta\rangle\|^2 \\ &= \sum_{\beta:\theta(\beta)>0} \|\Pi_x(\mathbf{1}-\Delta)(c_\beta|\beta\rangle + c_{-\beta}|-\beta\rangle)\|^2 \\ &= \sum_{\beta:\theta(\beta)>0} \sin^2\frac{\theta(\beta)}{2}\,\|(\mathbf{1}-\Delta)(c_\beta|\beta\rangle + c_{-\beta}|-\beta\rangle)\|^2 \\ &\le \left(\frac{\Theta}{2}\right)^2 \|(\mathbf{1}-\Delta)|\hat{\zeta}\rangle\|^2 \ .\end{aligned} \tag{5.12}$$

It remains to bound $|\langle \phi|\Delta|\hat{\zeta}\rangle| = |\langle \phi|w\rangle|\|\Delta|\hat{\zeta}\rangle\|$, where $|w\rangle = \Delta|\hat{\zeta}\rangle/\|\Delta|\hat{\zeta}\rangle\|$ is an eigenvalue-zero eigenvector of A_G. Intuitively, if $|\langle \phi|w\rangle| = |\langle \phi|\Pi_x|w\rangle|$ is large, then since A_G and $A_{G(x)}$ are the same on Π_x, $\|A_{G(x)}|w\rangle\| = \|S(\mathbf{1}-\Pi_x)|w\rangle\|$ will be small. This in turn will imply that $|w\rangle$ has large support on the small-eigenvalue subspace of $A_{G(x)}$, contradicting Eq. (5.5).

Beginning the formal argument, we have

$$\begin{aligned}\|A_{G(x)}\Delta|\hat{\zeta}\rangle\|^2 &= \|(\mathbf{1}-\Pi_x)\Delta|\hat{\zeta}\rangle\|^2 \\ &= \sum_{\beta:\theta(\beta)>0} \|(\mathbf{1}-\Pi_x)\Delta(c_\beta|\beta\rangle + c_{-\beta}|-\beta\rangle)\|^2 \\ &= \sum_{\beta:\theta(\beta)>0} \sin^2\frac{\theta(\beta)}{2}\,\|\Delta(c_\beta|\beta\rangle + c_{-\beta}|-\beta\rangle)\|^2 \\ &\le \left(\frac{\Theta}{2}\right)^2 \|\Delta|\hat{\zeta}\rangle\|^2 \ .\end{aligned} \tag{5.13}$$

Hence $\|A_{G(x)}|w\rangle\| \le \Theta/2$.

Now split $|w\rangle = |w_{\text{small}}\rangle + |w_{\text{big}}\rangle$, where for a certain $d > 0$ to be determined,

$$\begin{aligned}|w_{\text{small}}\rangle &= \sum_{\alpha:|\rho(\alpha)|\le d\Theta/2} |\alpha\rangle\langle\alpha|w\rangle \\ |w_{\text{big}}\rangle &= \sum_{\alpha:|\rho(\alpha)|> d\Theta/2} |\alpha\rangle\langle\alpha|w\rangle \ .\end{aligned} \tag{5.14}$$

Then

$$|\langle \phi|\Delta|\hat{\zeta}\rangle| \le |\langle \phi|w\rangle| \le |\langle \phi|w_{\text{small}}\rangle| + |\langle \phi|w_{\text{big}}\rangle| \ . \tag{5.15}$$

From Eq. (5.5) with $c = d\Theta W/2$, $|\langle \phi|w_{\text{small}}\rangle| \le \sqrt{72(1+1/W)}c\||w_{\text{small}}\rangle\| \le 6d\Theta W$.

Since $A_{G(x)}|w\rangle = \sum_\alpha \rho(\alpha)|\alpha\rangle\langle\alpha|w\rangle$, we have

$$\begin{aligned}\left(\frac{\Theta}{2}\right)^2 &\geq \|A_{G(x)}|w\rangle\|^2 \\ &= \|A_{G(x)}|w_{\text{small}}\rangle\|^2 + \|A_{G(x)}|w_{\text{big}}\rangle\|^2 \\ &\geq d^2\left(\frac{\Theta}{2}\right)^2 \||w_{\text{big}}\rangle\|^2 \ .\end{aligned} \tag{5.16}$$

Hence $\||w_{\text{big}}\rangle\| \leq 1/d$.

Combining our calculations gives

$$\sqrt{\sum_{\beta:|\theta(\beta)|\leq\Theta} |\langle\beta|\phi\rangle|^2} = \langle\phi|\hat{\zeta}\rangle \leq |\langle\phi|\Delta|\hat{\zeta}\rangle| + \|\Pi_x(\mathbf{1}-\Delta)|\hat{\zeta}\rangle\| \leq 6d\Theta W + \frac{1}{d} + \frac{\Theta}{2} \ . \tag{5.17}$$

The right-hand side is $2\sqrt{6\Theta W} + \Theta/2$, as claimed, for $d = 1/\sqrt{6\Theta W}$. □

Having proved Lemma 5.3, we return to the correctness proof for the third algorithm.

Proposition 5.5. *If $f(x) = 1$, then the third algorithm outputs 1 with probability at least 64%. If $f(x) = 0$, then the third algorithm outputs 1 with probability at most 61%.*

Proof. Letting $\tau = \lceil 10^5 W \rceil$, the third algorithm outputs 1 with probability

$$p_1 := \mathrm{E}_{T\in_R[\tau]}\left[|\langle\phi|U_x^T|\phi\rangle|^2\right] = \mathrm{E}_{T\in_R[\tau]}\left[\left|\sum_\beta e^{i\theta(\beta)T}|\langle\beta|\phi\rangle|^2\right|^2\right] \ . \tag{5.18}$$

If $f(x) = 1$, then a crude bound puts p_1 at least $(9/10 - 1/10)^2 = 64\%$.

Assume $f(x) = 0$. Recall the notation that for an eigenvector $|\beta\rangle$ with $|\theta(\beta)| \in (0,\pi)$, $|-\beta\rangle = (2\Delta - \mathbf{1})|\beta\rangle$ denotes the corresponding eigenvector with eigenvalue phase $\theta(-\beta) = -\theta(\beta)$. The key observation for this proof is that

$$\langle\phi|\beta\rangle = e^{-i\theta(\beta)}\langle\phi|-\beta\rangle \ . \tag{5.19}$$

This equal splitting of $|\langle\phi|\beta\rangle|$ and $|\langle\phi|-\beta\rangle|$ will allow us to bound p_1 close to $1/2$ instead of the trivial bound $p_1 \leq 1$. The intuition is that after applying U_x a suitable number of times, eigenvectors $|\beta\rangle$ and $|-\beta\rangle$ will accumulate roughly opposite phases, so their overlaps with $|\phi\rangle$ will roughly cancel out. For this argument to work, though, the eigenvalue phase $\theta(\beta)$ should be bounded away from zero and from $\pm\pi$. Therefore define the projections

$$\begin{aligned}\Delta_\Theta &= \sum_{\beta:|\theta(\beta)|\leq\Theta} |\beta\rangle\langle\beta| \\ \overline{\Delta}_\Lambda &= \sum_{\beta:|\theta(\beta)|>\Lambda} |\beta\rangle\langle\beta| \\ \Sigma &= \mathbf{1} - \Delta_\Theta - \overline{\Delta}_\Lambda \ ,\end{aligned} \tag{5.20}$$

where Θ and Λ, $0 < \Theta < \Lambda < \pi$, will be determined below. Lemma 5.3 immediately gives the bound $\|\Delta_\Theta|\phi\rangle\| \leq 2\sqrt{6\Theta W} + \frac{\Theta}{2}$. We can also place a bound on $\|\overline{\Delta}_\Lambda|\phi\rangle\|$, using

$$2(\Delta - \mathbf{1})|\phi\rangle = (U_x^\dagger - \mathbf{1})|\phi\rangle = \sum_\beta (e^{-i\theta(\beta)} - 1)|\beta\rangle\langle\beta|\phi\rangle \ . \tag{5.21}$$

Expanding the squared norm of both sides gives

$$\|(U_x^\dagger - \mathbf{1})|\phi\rangle\|^2 = 4\sum_\beta \sin^2\frac{\theta(\beta)}{2}|\langle\beta|\phi\rangle|^2 \geq \|\overline{\Delta}_\Lambda|\phi\rangle\|^2 \cdot 4\sin^2\frac{\Lambda}{2} \tag{5.22}$$

and

$$\|(U_x^\dagger - \mathbf{1})|\phi\rangle\|^2 = 4(1 - \|\Delta|\phi\rangle\|^2) \leq 2/5 \ . \tag{5.23}$$

In the second step we have used that $\|\Delta|\phi\rangle\|^2 \geq 9/10$; provided that f is not the constant zero function, A_G must have an eigenvalue-zero eigenvector with large overlap on $|\phi\rangle$. Combining Eqs. (5.22) and (5.23) gives

$$\|\overline{\Delta}_\Lambda|\phi\rangle\|^2 \leq \frac{1}{10\sin^2\frac{\Lambda}{2}} \ . \tag{5.24}$$

Returning to Eq. (5.18), we have

$$\begin{aligned} p_1 &\leq \mathrm{E}_{T\in_R[\tau]}\Big[\Big(\|\Delta_\Theta|\phi\rangle\|^2 + \|\overline{\Delta}_\Lambda|\phi\rangle\|^2 + \Big|\sum_{\beta:\theta(\beta)\in(\Theta,\Lambda]}|\langle\beta|\phi\rangle|^2\big(e^{i\theta(\beta)T} + e^{-i\theta(\beta)T}\big)\Big|\Big)^2\Big] \\ &\leq (\|\Delta_\Theta|\phi\rangle\|^2 + \|\overline{\Delta}_\Lambda|\phi\rangle\|^2)(2 - \|\Delta_\Theta|\phi\rangle\|^2 - \|\overline{\Delta}_\Lambda|\phi\rangle\|^2) \\ &\quad + \mathrm{E}_{T\in_R[\tau]}\Big[\Big(\sum_{\beta:\theta(\beta)\in(\Theta,\Lambda]}|\langle\beta|\phi\rangle|^2\big(e^{i\theta(\beta)T} + e^{-i\theta(\beta)T}\big)\Big)^2\Big] \ . \end{aligned} \tag{5.25}$$

The algorithm chooses T at random to allow bounding the last term. Expanding this term gives

$$\begin{aligned} &\mathrm{E}_{T\in_R[\tau]}\Big[\sum_{\beta,\beta':\theta(\beta),\theta(\beta')\in(\Theta,\Lambda]}|\langle\beta|\phi\rangle|^2|\langle\beta'|\phi\rangle|^2\big(e^{i\theta(\beta)T} + e^{-i\theta(\beta)T}\big)\big(e^{i\theta(\beta')T} + e^{-i\theta(\beta')T}\big)\Big] \\ &= \mathrm{E}\Big[\sum_{\theta,\theta'\in(\Theta,\Lambda]}|\langle\beta|\phi\rangle|^2|\langle\beta'|\phi\rangle|^2\big((e^{i(\theta+\theta')T} + e^{-i(\theta+\theta')T}) + (e^{i(\theta-\theta')T} + e^{-i(\theta-\theta')T})\big)\Big] \\ &\leq \frac{1}{2}\|\Sigma|\phi\rangle\|^4 + \mathrm{E}\Big[\sum_{\theta,\theta'\in(\Theta,\Lambda]}|\langle\beta|\phi\rangle|^2|\langle\beta'|\phi\rangle|^2\big(e^{i(\theta+\theta')T} + e^{-i(\theta+\theta')T}\big)\Big] \end{aligned} \tag{5.26}$$

Here for brevity we have written θ and θ' for $\theta(\beta)$ and $\theta(\beta')$, respectively. In the second step, we have used $\sum_{\theta\in(\Theta,\Lambda]}|\langle\beta|\phi\rangle|^2 = \frac{1}{2}\|\Sigma|\phi\rangle\|^2$. We continue by simplifying the expectation,

$$\begin{aligned} &= \frac{1}{2}\|\Sigma|\phi\rangle\|^4 + \frac{1}{\tau}\sum_{\theta,\theta'\in(\Theta,\Lambda]}|\langle\beta|\phi\rangle|^2|\langle\beta'|\phi\rangle|^2\Big(\frac{e^{i(\theta+\theta')(\tau+1)} - e^{-i(\theta+\theta')\tau}}{e^{i(\theta+\theta')} - 1} - 1\Big) \\ &\leq \frac{1}{2}\Big(1 + \frac{1/\tau}{\min_{\theta,\theta'\in(\Theta,\Lambda]}|e^{i(\theta+\theta')} - 1|}\Big)\|\Sigma|\phi\rangle\|^4 \\ &\leq \frac{1}{2}\Big(1 + \frac{1}{2\tau\min\{\sin\Theta, \sin\Lambda\}}\Big)\|\Sigma|\phi\rangle\|^4 \ . \end{aligned} \tag{5.27}$$

In the last step, we have used $|e^{i(\theta+\theta')} - 1| = 2\sin\frac{\theta+\theta'}{2} \geq 2\min\{\sin\Theta, \sin\Lambda\}$. Substituting the result back into Eq. (5.25) gives

$$\begin{aligned} p_1 &\leq 1 - \frac{1}{2}\Big(1 - \frac{1}{2\tau\min\{\sin\Theta, \sin\Lambda\}}\Big)\|\Sigma|\phi\rangle\|^4 \\ &\leq 1 - \frac{1}{2}\Big(1 - \frac{1}{2\tau\min\{\sin\Theta, \sin\Lambda\}}\Big)\max\Big[1 - \frac{1}{10\sin^2\frac{\Lambda}{2}} - \Big(2\sqrt{6\Theta W} + \frac{\Theta}{2}\Big)^2, 0\Big]^2 \ . \end{aligned} \tag{5.28}$$

Setting $\Lambda = \pi - \Theta$ and $\Theta = 1/(2000W)$, for $W \geq 1$ a calculation yields $p_1 \leq 61\%$. □

6 Span programs

Having proved Theorem 1.3, we now turn attention to span programs. In this section, we will define span programs, from [KW93], and the witness size span program complexity measure from [RŠ08]. Additionally, we define a new complexity measure, the "full witness size," that charges even for the "free" inputs. Section 7 develops a correspondence between span programs and bipartite graphs that, together with Definition 6.4 below, underlies the construction of the graph G in Eqs. (3.2) and (3.3). Theorems 7.2 and 7.3 then generalize Lemmas 5.1 and 5.2, respectively. Section 8 will apply this framework to develop a fourth quantum algorithm for evaluating span programs. Compared to the algorithms in Section 3, it has greater potential for having a time-efficient implementation.

A span program P is a certain linear-algebraic way of specifying a boolean function f_P. Roughly, P consists of a target vector $|t\rangle$ in a vector space V, and a collection of subspaces $V_{j,b} \subseteq V$, for $j \in [n]$ and $b \in \{0,1\}$. For an input $x \in \{0,1\}^n$, $f_P(x) = 1$ when the target can be reached using a linear combination of vectors in $\cup_{j\in[n]} V_{j,x_j}$. For our complexity measures 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 (see Figure 2).

Formally we therefore define a span program as follows:

Definition 6.1 (Span program [KW93])**.** *A span program P consists of a natural number n, a finite-dimensional inner product space V over $\mathbf{C}$, a "target" vector $|t\rangle \in V$, disjoint sets I_{free} and $I_{j,b}$ for $j \in [n]$, $b \in \{0,1\}$, and "input vectors" $|v_i\rangle \in V$ for $i \in I_{\text{free}} \cup \bigcup_{j\in[n],b\in\{0,1\}} I_{j,b}$.*

To P corresponds a function $f_P : \{0,1\}^n \to \{0,1\}$, defined on $x \in \{0,1\}^n$ by

$$f_P(x) = \begin{cases} 1 & \text{if } |t\rangle \in \text{Span}(\{|v_i\rangle : i \in I_{\text{free}} \cup \bigcup_{j\in[n]} I_{j,x_j}\}) \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

We say that I_{free} indexes the set of "free" input vectors, while $I_{j,b}$ indexes input vectors "labeled by" (j,b). We say that P "computes" the function f_P. For $x \in \{0,1\}^n$, $f_P(x)$ evaluates to 1, or true, when the target can be reached using a linear combination of the "available" input vectors, i.e., input vectors that are either free or labeled by (j, x_j) for $j \in [n]$.

Some additional notation is convenient. Fix a span program P. Let $I = I_{\text{free}} \cup \bigcup_{j\in[n],b\in\{0,1\}} I_{j,b}$. Let $A \in \mathcal{L}(\mathbf{C}^I, V)$ be the linear operator

$$A = \sum_{i\in I} |v_i\rangle\langle i| \ . \tag{6.2}$$

Written as a matrix, the columns of A are the input vectors of P.

For an input $x \in \{0,1\}^n$, let $I(x)$ be the set of available input vector indices and $\Pi(x) \in \mathcal{L}(\mathbf{C}^I)$ the projection thereon,

$$I(x) = I_{\text{free}} \cup \bigcup_{j \in [n]} I_{j,x_j} \tag{6.3}$$

$$\Pi(x) = \sum_{i \in I(x)} |i\rangle\langle i| \ . \tag{6.4}$$

Then $f_P(x) = 1$ if $|t\rangle \in \text{Range}(A\Pi(x))$. A vector $|w\rangle \in \mathbf{C}^I$ is said to be a witness for $f_P(x) = 1$ if $\Pi(x)|w\rangle = |w\rangle$ and $A|w\rangle = |t\rangle$. A vector $|w'\rangle \in V$ is said to be a witness for $f_P(x) = 0$ if $\langle t|w'\rangle = 1$ and $\Pi(x)A^\dagger|w'\rangle = 0$.

Lemma 6.2. *For a span program P, $f_P(x) = 1$ if and only if $|t\rangle \in \text{Range}(A\Pi(x))$. Equivalently, $f_P(x) = 0$ if and only if $\Pi(x)A^\dagger|t\rangle \in \text{Range}\Big[\Pi(x)A^\dagger\Big(\mathbf{1} - \frac{|t\rangle\langle t|}{\||t\rangle\|^2}\Big)\Big]$.*

Lemma 6.2 follows from Eq. (6.1). 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 main complexity measure we use to characterize span programs is the witness size [RŠ08]:

Definition 6.3 (Witness size)**.** *Consider a span program P, and a vector $s \in [0,\infty)^n$ of nonnegative "costs." Let $S = \sum_{j \in [n], b \in \{0,1\}, i \in I_{j,b}} \sqrt{s_j}|i\rangle\langle i| \in \mathcal{L}(\mathbf{C}^I)$. For each input $x \in \{0,1\}^n$, define the witness size of P on x with costs s, $\text{wsize}_s(P,x)$, as follows:*

$$\text{wsize}_s(P,x) = \begin{cases} \min_{|w\rangle:\, A\Pi(x)|w\rangle = |t\rangle} \|S|w\rangle\|^2 & \text{if } f_P(x) = 1 \\ \min\limits_{\substack{|w'\rangle:\, \langle t|w'\rangle = 1 \\ \Pi(x)A^\dagger|w'\rangle = 0}} \|SA^\dagger|w'\rangle\|^2 & \text{if } f_P(x) = 0 \end{cases} \tag{6.5}$$

The witness size of P with costs s, restricted to domain $\mathcal{D} \subseteq \{0,1\}^n$, is

$$\text{wsize}_s(P,\mathcal{D}) = \max_{x \in \mathcal{D}} \text{wsize}_s(P,x) \ . \tag{6.6}$$

Define the full witness size $\text{fwsize}_s(P,\mathcal{D})$ by letting $S^f = S + \sum_{i \in I_{\text{free}}} |i\rangle\langle i|$ and

$$\text{fwsize}_s(P,x) = \begin{cases} \min_{|w\rangle:\, A\Pi(x)|w\rangle = |t\rangle} (1 + \|S^f|w\rangle\|^2) & \text{if } f_P(x) = 1 \\ \min\limits_{\substack{|w'\rangle:\, \langle t|w'\rangle = 1 \\ \Pi(x)A^\dagger|w'\rangle = 0}} (\||w'\rangle\|^2 + \|SA^\dagger|w'\rangle\|^2) & \text{if } f_P(x) = 0 \end{cases} \tag{6.7}$$

$$\text{fwsize}_s(P,\mathcal{D}) = \max_{x \in \mathcal{D}} \text{fwsize}_s(P,x) \ . \tag{6.8}$$

When not specified, $\mathcal{D}$ is assumed to be $\{0,1\}^n$ and the costs s are taken to be uniform $s = \vec{1} = (1,1,\ldots,1)$, e.g., $\text{wsize}(P) = \text{wsize}_{\vec{1}}(P,\{0,1\}^n)$. In the latter case, note that $S^f = \mathbf{1}$. The extra generality of allowing nonuniform costs is necessary for considering unbalanced formulas. For $j \in [n]$, s_j can intuitively be thought of as the charge for evaluating the jth input bit.

The above definition of span programs differs slightly from the original definition in [KW93]. Call a span program *strict* if $I_{\text{free}} = \emptyset$. Ref. [KW93] considers only strict span programs. For the

witness size complexity measure, span programs and strict span programs are equivalent, since free input vectors can be projected away without changing the witness size [Rei10a, Prop. 3.4]. Allowing free input vectors is often convenient for defining and composing span programs, though, and may be necessary for developing time-efficient quantum algorithms based on span programs. For succinctly presenting span programs, Ref. [RŠ08] uses an even more relaxed span program definition, letting each input vector be labeled by a subset of $[n] \times \{0,1\}$. This definition is also equivalent to ours. 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].

To help explain the derivation of the algorithms in Section 3, define canonical span programs:

Definition 6.4 ([KW93])**.** *A strict span program P on n bits is* canonical *if $V = \mathbf{C}^{F_0}$ where $F_0 = \{x \in \{0,1\}^n : f_P(x) = 0\}$, the target is $|t\rangle = \sum_{x \in F_0} |x\rangle$, and for all $x \in F_0$ and $i \in I(x)$, $\langle x|v_i\rangle = 0$.*

The full witness size is not the same as the witness size. When $f_P(x) = 1$, the full witness size counts the portion of $|w\rangle$ supported on I_{free}, whereas the witness size does not. When $f_P(x) = 0$, the term $\||w'\rangle\|^2$ in Eq. (6.7) is not necessarily bounded by the witness size. However, any span program can be converted into canonical form, so that the two measures essentially agree:

Theorem 6.5 ([Rei10a, Theorem 3.6])**.** *For any span program P and cost vector s, there exists a canonical span program $\hat{P}$ with $f_{\hat{P}} = f_P$ and such that for all inputs x, $\text{wsize}_s(\hat{P}, x) \leq \text{wsize}_s(P, x)$ and $\text{fwsize}_s(\hat{P}, x) \leq \text{wsize}_s(P, x) + 1$.*

Several further span program transformations, such as converting a span program computing f into one computing $\neg f$, are given in [Rei09a]. Three methods of composing span programs, direct sum, tensor product and reduced tensor product composition, are studied in [Rei09a, Rei09b, Rei09c]. Many explicit examples are given in [RŠ08, Rei09a].

7 Correspondence between span programs and bipartite graphs

In this section, we define a correspondence between span programs and weighted bipartite graphs. We relate span program witness size to properties of eigenvalue-zero eigenvectors of these graphs, following [RŠ08]. We then apply Theorem 4.1 to derive effective spectral gaps.

Definition 7.1 (Graphs $G_P(x)$ [RŠ08])**.** *Let P be a span program with target vector $|t\rangle$ and input vectors $|v_i\rangle$ for $i \in I$, in inner product space V. Fix an arbitrary orthonormal basis $\{|k\rangle : k \in [\dim(V)]\}$ for V. For $x \in \{0,1\}^n$, let $G_P(x)$ be the weighted bipartite graph with biadjacency matrix*

$$B_{G_P(x)} = \begin{pmatrix} \emptyset & I \\ |t\rangle & A \\ 0 & \overline{\Pi}(x) \end{pmatrix} \begin{matrix} V \\ I \end{matrix} \tag{7.1}$$

where

$$\overline{\Pi}(x) = \mathbf{1} - \Pi(x) = \sum_{i \in I \setminus I(x)} |i\rangle\langle i| \ . \tag{7.2}$$

The row vertices of $G_P(x)$ are $T = [\dim(V)] \cup I$, and the column vertices are $U = \{\emptyset\} \cup I$, both disjoint unions. The vertex $\emptyset$ is called the "output vertex."

Let G_P be the bipartite graph with biadjacency matrix given by

$$B_{G_P} = \begin{pmatrix} \emptyset & I \\ |t\rangle & A \\ 0 & \overline{\Pi}_{\text{free}} \end{pmatrix} \begin{matrix} V \\ I \end{matrix}$$
$$\overline{\Pi}_{\text{free}} = \sum_{i \in I \setminus I_{\text{free}}} |i\rangle\langle i| \ . \tag{7.3}$$

Note that for each $i \in I$, $G_P(x)$ has two vertices, connected by a weight-one edge if $i \notin I(x)$.

7.1 Eigenvalue-zero spectral analysis of the graphs $G_P(x)$

We now connect eigenvalue-zero eigenvectors of $A_{G_P(x)}$ to span program witnesses, in a straightforward extension of [RŠ08, Theorems 2.5 and A.7].

Some more notation will be useful. Any vector $|\psi\rangle \in \mathbf{C}^T \oplus \mathbf{C}^U$ can be uniquely expanded as $|\psi\rangle = (|\psi_T\rangle, |\psi_U\rangle)$, with $|\psi_T\rangle \in \mathbf{C}^T$ and $|\psi_U\rangle \in \mathbf{C}^U$. For the graphs $G_P(x)$, $\mathbf{C}^T = V \oplus \mathbf{C}^I$ and $\mathbf{C}^U = \mathbf{C}^{\{\emptyset\}} \oplus \mathbf{C}^I$, so $|\psi\rangle$ can be further expanded as $\big((|\psi_{T,V}\rangle, |\psi_{T,I}\rangle), (\psi_{U,\emptyset}, |\psi_{U,I}\rangle)\big)$.

With this notation, the eigenvalue-ρ eigenvector equation of $A_{G_P(x)}$,

$$\rho|\psi\rangle = A_{G_P(x)}|\psi\rangle \ , \tag{7.4}$$

is equivalent to the four equations

$$\rho|\psi_{T,V}\rangle = \psi_{U,\emptyset}|t\rangle + A|\psi_{U,I}\rangle \tag{7.5a}$$
$$\rho|\psi_{T,I}\rangle = \overline{\Pi}(x)|\psi_{U,I}\rangle \tag{7.5b}$$
$$\rho\,\psi_{U,\emptyset} = \langle t|\psi_{T,V}\rangle \tag{7.5c}$$
$$\rho|\psi_{U,I}\rangle = A^\dagger|\psi_{T,V}\rangle + \overline{\Pi}(x)|\psi_{T,I}\rangle \ . \tag{7.5d}$$

Theorem 7.2 ([RŠ08])**.** *For a span program P and input $x \in \{0,1\}^n$, consider all the eigenvalue-zero eigenvector equations of the weighted adjacency matrix $A_{G_P(x)}$, except for the constraint at the output vertex $\emptyset$, i.e., Eqs. (7.5) except (7.5c) at $\rho = 0$.*

These equations have a solution $|\psi\rangle$ with $\psi_{U,\emptyset} \neq 0$ if and only if $f_P(x) = 1$, and have a solution $|\psi\rangle$ with $\langle t|\psi_{T,V}\rangle \neq 0$ if and only if $f_P(x) = 0$. More quantitatively, let $s \in [0,\infty)^n$ be a vector of nonnegative costs, and recall from Definition 6.3 the cost matrix S^f. Then

- *If $f_P(x) = 1$, $A_{G_P(x)}$ has an eigenvalue-zero eigenvector $|\psi\rangle = (0, \psi_{U,\emptyset}, |\psi_{U,I}\rangle) \in \mathbf{C}^T \oplus \mathbf{C}^{\{\emptyset\}} \oplus \mathbf{C}^I$ with*
$$\frac{|\psi_{U,\emptyset}|^2}{|\psi_{U,\emptyset}|^2 + \||S^f|\psi_{U,I}\rangle\|^2} = \frac{1}{\text{fwsize}_s(P,x)} \ . \tag{7.6}$$
- *If $f_P(x) = 0$, then there is a solution $|\psi\rangle = (|\psi_{T,V}\rangle, |\psi_{T,I}\rangle, 0) \in V \oplus \mathbf{C}^I \oplus \mathbf{C}^U$ to Eqs. (7.5a,b,d) at $\rho = 0$, with*
$$\frac{|\langle t|\psi_{T,V}\rangle|^2}{\||\psi_{T,V}\rangle\|^2 + \||S^f|\psi_{T,I}\rangle\|^2} = \frac{1}{\text{fwsize}_s(P,x)} \ . \tag{7.7}$$

Proof. Let $\rho = 0$. Since $G_P(x)$ is bipartite, the ψ_T terms do not interact with the ψ_U terms. Thus Eqs. (7.5c,d) (resp. 7.5a,b) can always be satisfied by setting the ψ_T (resp. ψ_U) terms to zero.

Now Eqs. (7.5a,b) are equivalent to $-\psi_{U,\emptyset}|t\rangle = A|\psi_{U,I}\rangle$ and $|\psi_{U,I}\rangle = \Pi(x)|\psi_{U,I}\rangle$. If these equations have a solution with $\psi_{U,\emptyset} \neq 0$, then $-|\psi_{U,I}\rangle/\psi_{U,\emptyset}$ is a witness for $f_P(x) = 1$. Conversely, if $f_P(x) = 1$, then let $|w\rangle \in \mathbf{C}^I$ be an optimal witness, satisfying $A\Pi(x)|w\rangle = |t\rangle$ and $\text{fwsize}_s(P, x) = \||S^f|w\rangle\|^2$. Let $\psi_{U,\emptyset} = -1$ and $|\psi_{U,I}\rangle = \Pi(x)|w\rangle$. Then $|\psi\rangle = (0, \psi_{U,\emptyset}, |\psi_{U,I}\rangle)$ satisfies Eqs. (7.5), and Eq. (7.6) with equality.

Next, assume $|\psi\rangle$ solves Eq. (7.5d) with $\langle t|\psi_{T,V}\rangle \neq 0$. Then $\Pi(x)A^\dagger|\psi_{T,V}\rangle = -\Pi(x)\overline{\Pi}(x)|\psi_{T,I}\rangle = 0$, so $|\psi_{T,V}\rangle/\langle t|\psi_{T,V}\rangle$ is a witness for $f_P(x) = 0$. Conversely, assume that $f_P(x) = 0$ and let $|w'\rangle$ be an optimal witness, achieving the full witness size Eq. (6.7). Let $|\psi_{T,V}\rangle = |w'\rangle$ and $|\psi_{T,I}\rangle = -A^\dagger|w'\rangle$. Then $|\psi\rangle = (|\psi_{T,V}\rangle, |\psi_{T,I}\rangle, 0)$ satisfies Eqs. (7.5a,b,d), and Eq. (7.7) with equality. □

Note that with costs $s = \vec{1}$, $S^f = \mathbf{1}$, so $\||S^f|\psi_{U,I}\rangle\|^2 = \||\psi_{U,I}\rangle\|^2$ and $\||S^f|\psi_{T,I}\rangle\|^2 = \||\psi_{T,I}\rangle\|^2$. Then the denominators on the left-hand sides of Eqs. (7.6) and (7.7) are both $\||\psi\rangle\|^2$.

7.2 Amplification and effective spectral gaps for the graphs

Theorem 7.2 implies in particular that when $f_P(x) = 0$, $A_{G_P(x)}$ does not have any eigenvalue-zero eigenvectors supported on the output vertex. Therefore $A_{G_P(x)}$ has some spectral gap around zero for eigenvectors overlapping $|\emptyset\rangle$. This spectral gap can be arbitrarily small, though. In fact, though, the lower bound Eq. (7.7) can be translated into a good lower bound on an "effective" spectral gap. That is, we can upper-bound the total squared support of $|\emptyset\rangle$ on small-magnitude eigenvalues of $A_{G_P(x)}$. This is strong enough for applying quantum phase estimation, and is the key result that allows span programs to be evaluated on a quantum computer.

Theorem 7.3. *Let P be a span program and $\mathcal{D} \subseteq \{0,1\}^n$. Let P' be the same as P except with the target vector scaled by a factor of $1/\sqrt{\text{fwsize}(P, \mathcal{D})}$. Then $f_{P'} = f_P$ and, for all $x \in \mathcal{D}$,*

- *If $f_P(x) = 1$, then there is an eigenvalue-zero eigenvector $|\psi\rangle$ of $A_{G_{P'}(x)}$ with*

$$\frac{|\langle\emptyset|\psi\rangle|^2}{\||\psi\rangle\|^2} \geq \frac{1}{2} \ . \tag{7.8}$$

- *If $f_P(x) = 0$, let $\{|\alpha\rangle\}$ be a complete set of orthonormal eigenvectors of $A_{G_{P'}(x)}$, with corresponding eigenvalues $\rho(\alpha)$. Then for any $\Upsilon \geq 0$, the squared length of the projection of $|\emptyset\rangle$ onto the span of the eigenvectors α with $|\rho(\alpha)| \leq \Upsilon$ satisfies*

$$\sum_{\alpha:\,|\rho(\alpha)|\leq\Upsilon} |\langle\alpha|\emptyset\rangle|^2 \leq 8\Upsilon^2\,\text{fwsize}(P, \mathcal{D})^2 \ . \tag{7.9}$$

Proof. Scaling the target vector does not change the function: $f_{P'} = f_P$. Moreover, for all $x \in \mathcal{D}$,

$$\text{fwsize}(P', x) \leq \begin{cases} 2 & \text{if } f_P(x) = 1 \\ \text{fwsize}(P, \mathcal{D})^2 & \text{if } f_P(x) = 0 \end{cases} \tag{7.10}$$

This scaling, or "amplification," step is essentially the same as a technique used in [ACR+10, RŠ08].

For the case $f_P(x) = 1$, Eq. (7.8) now follows from Eq. (7.6) in Theorem 7.2.

For the case $f_P(x) = 0$, we will combine Theorem 7.2 and Theorem 4.1. Let G be the graph $G_{P'}(x)$ with the output vertex and all incident edges deleted. Thus G's biadjacency matrix is the same as $B_{G_{P'}(x)}$ from Eq. (7.1), except with the first column deleted. Theorem 7.2 implies that A_G has an eigenvalue-zero eigenvector $|\psi\rangle = (|\psi_{T,V}\rangle, |\psi_{T,I}\rangle, 0) \in V \oplus \mathbf{C}^I \oplus \mathbf{C}^I$ satisfying

$$\frac{|\langle t'|\psi_{T,V}\rangle|^2}{\||\psi\rangle\|^2} \geq \frac{1}{\text{fwsize}(P,\mathcal{D})^2} \ . \tag{7.11}$$

Eq. (7.9) now follows by Eq. (4.3) in Theorem 4.1 with $G' = G_{P'}(x)$ and $\delta = 1/\text{fwsize}(P,\mathcal{D})^2$. □

Note that the graph G defined by Eq. (3.3), on which the algorithms in Section 3 are based, corresponds under Definition 7.1 to a span program. Undoing the amplification step by multiplying the target vector by $3\sqrt{W}$—see Eq. (3.2)—yields a canonical span program P that satisfies $f_P|_{\mathcal{D}} = f$, $\text{wsize}(P,\mathcal{D}) = W = \text{Adv}^{\pm}(f)$ and $\text{fwsize}(P,\mathcal{D}) \leq W + 1$. The eigenvectors in Lemma 5.1 are constructed from witnesses as in Theorem 7.2. (For $x \in F_0$, a witness to $f_P(x) = 0$ is $|w'\rangle = |x\rangle$.) This explains the origin of G. For further details, see the proof of Theorem 1.7 in [Rei10a].

8 Quantum algorithm to evaluate a span program based on its full witness size

Based on the graph spectral properties from Theorem 7.3, a quantum query algorithm for evaluating span programs based on the full witness size can be constructed. The algorithms in Section 3 do this implicitly for an optimal span program. In this section, we will give a different quantum algorithm, based on a certain quantum walk. Its query complexity loses a factor of $\|\text{abs}(A_{G_P})\|$, where $\text{abs}(\cdot)$ denotes the entry-wise absolute value. The advantage of this algorithm, though, is that there is often a clear time-efficient implementation of the basic operations, provided the graph G_P has polynomially bounded maximum degree. In particular, the time-efficient formula-evaluation applications [Rei09b, Rei09c] use this algorithm; by working with span programs that are not in canonical form, they trade off $\text{fwsize}(P)$ versus $\|\text{abs}(A_{G_P})\|$ and the maximum degree.

Theorem 8.1. *Let P be a span program and $\mathcal{D} \subseteq \{0,1\}^n$. Then $f_P|_{\mathcal{D}}$ can be evaluated using*

$$Q = O\big(\text{fwsize}(P,\mathcal{D})\,\|\text{abs}(A_{G_P})\|\big) \tag{8.1}$$

quantum queries, with error probability at most $1/3$. Moreover, if the maximum degree of a vertex in G_P is d, then the algorithm's time complexity is at most a factor of $(\log d)\big(\log(Q\log d)\big)^{O(1)}$ worse, after classical preprocessing and assuming constant-time coherent access to the preprocessed string.

The intuition behind this theorem is that f_P can be evaluated by starting at the output vertex and "measuring" $A_{G_{P'}(x)}$ to an appropriate precision Υ. (More precisely, this is implemented by applying phase estimation to a certain unitary operator.) Output 1 if and only if the measurement returns 0. Eq. (7.8) implies completeness when $f(x) = 1$, because the initial state has large overlap with an eigenvalue-zero eigenstate. Eq. (7.9) implies soundness when $f(x) = 0$.

The proof of Theorem 8.1 is based on Szegedy's correspondence between continuous- and discrete-time quantum walks [Sze04]. The proof is nearly the same as in [RŠ08, Appendix B.2], which in turn is based on the algorithms in [CRŠZ07, ACR+10]. The difference is that we are only

assuming an effective spectral gap in the case $f(x) = 0$, instead of a spectral gap. This weaker assumption means that establishing the algorithm's soundness requires somewhat more care.

We use a formulation of Szegedy's correspondence theorem from [ACR+10]:

Theorem 8.2 ([Sze04])**.** *Let V be a finite set. For each $v \in V$, let $|\varphi_v\rangle \in \mathbf{C}^V$ be a length-one vector. Define $T \in \mathcal{L}(\mathbf{C}^V, \mathbf{C}^V \otimes \mathbf{C}^V)$, $S, U \in \mathcal{L}(\mathbf{C}^V \otimes \mathbf{C}^V)$ and $M \in \mathcal{L}(\mathbf{C}^V)$ by*

$$T = \sum_{v \in V} (|v\rangle \otimes |\varphi_v\rangle)\langle v| \qquad S = \sum_{v,w \in V} |v,w\rangle\langle w,v| \tag{8.2}$$

$$U = (2TT^\dagger - \mathbf{1})S \qquad M = T^\dagger S T = \sum_{v,w \in V} \langle\varphi_v|w\rangle\langle v|\varphi_w\rangle |v\rangle\langle w| \tag{8.3}$$

Since $T^\dagger T = \mathbf{1}$, U is a unitary. (U is a swap followed by a reflection about the span of the vectors $\{|v\rangle \otimes |\varphi_v\rangle : v \in V\}$.) M is a Hermitian matrix with $\|M\| \leq 1$. Let $\{|\alpha\rangle\}$ be a complete set of orthonormal eigenvectors of M with respective eigenvalues $\rho(\alpha)$.

Then the spectral decomposition of U corresponds to that of M as follows: Let $R_\alpha = \mathrm{Span}\{T|\alpha\rangle, ST|\alpha\rangle\}$. Then $R_\alpha \perp R_{\alpha'}$ for $\alpha \neq \alpha'$; let $R = \oplus_\alpha R_\alpha$. U is $-S$ on $R^\perp$, and U preserves each subspace R_α.

If $|\rho(\alpha)| < 1$, then R_α is two-dimensional, and within it the eigenvectors and corresponding eigenvalues of U are given by

$$\begin{aligned} |\alpha, \pm\rangle &= \Big(\mathbf{1} - \big(\rho(\alpha) \mp i\sqrt{1 - \rho(\alpha)^2}\big)S\Big)T|\alpha\rangle \\ \lambda(\alpha, \pm) &= \rho(\alpha) \pm i\sqrt{1 - \rho(\alpha)^2} \ . \end{aligned} \tag{8.4}$$

If $\rho(\alpha) \in \{1, -1\}$, then $ST|\alpha\rangle = \rho(\alpha)T|\alpha\rangle$, so R_α is one-dimensional; let $|\alpha, +\rangle = T|\alpha\rangle$ and $\lambda(\alpha, +) = \rho(\alpha)$ be the corresponding eigenvalue of U.

We will need slightly more control over the eigenvectors $|\alpha, \pm\rangle$:

Lemma 8.3. *With the setup of Theorem 8.2, for any $|\psi\rangle \in \mathbf{C}^V$, the eigenvectors $|\alpha, \pm\rangle$ with $|\rho(\alpha)| < 1$ satisfy $\||\alpha, \pm\rangle\| = \sqrt{2(1 - \rho(\alpha)^2)}$ and*

$$\frac{|\langle\psi|T^\dagger|\alpha, \pm\rangle|^2}{\||\alpha, \pm\rangle\|^2} = \frac{1}{2}|\langle\psi|\alpha\rangle|^2 \ . \tag{8.5}$$

When $|\rho(\alpha)| = 1$, $\||\alpha, +\rangle\| = 1$ and $\langle\psi|T^\dagger|\alpha, +\rangle = \langle\psi|\alpha\rangle$.

Proof. Fix an eigenvector $|\alpha\rangle$ of $A_{G(x)}$ and let $\rho = \rho(\alpha)$. Assume that $|\rho| < 1$. We have

$$\begin{aligned} \||\alpha, \pm\rangle\|^2 &= \langle\alpha|T^\dagger(\mathbf{1} - e^{\pm i \arccos\rho} S)(\mathbf{1} - e^{\mp i \arccos\rho} S)T|\alpha\rangle \\ &= \langle\alpha|T^\dagger 2(\mathbf{1} - \rho S)T|\alpha\rangle \\ &= 2(1 - \rho\langle\alpha|T^\dagger S T|\alpha\rangle) \\ &= 2(1 - \rho^2) \ , \end{aligned} \tag{8.6}$$

where we have used $S^2 = T^\dagger T = \mathbf{1}$, $\||\alpha\rangle\| = 1$, and $T^\dagger S T = M$. Also, then, we compute

$$\begin{aligned} \langle\psi|T^\dagger|\alpha, \pm\rangle &= \langle\psi|T^\dagger(\mathbf{1} - e^{\mp i \arccos\rho} S)T|\alpha\rangle \\ &= \langle\psi|T^\dagger T|\alpha\rangle - e^{\mp i \arccos\rho}\langle\psi|T^\dagger S T|\alpha\rangle \\ &= \langle\psi|\alpha\rangle(1 - \rho\, e^{\mp i \arccos\rho}) \\ &= \langle\psi|\alpha\rangle(1 - \rho^2 \pm i\rho\sqrt{1 - \rho^2}) \ , \end{aligned} \tag{8.7}$$

so $|\langle\psi|T^\dagger|\alpha,\pm\rangle|^2 = |\langle\psi|\alpha\rangle|^2(1-\rho^2)$. Eq. (8.5) follows.

When $|\rho(\alpha)| = 1$, the claims are immediate from $|\alpha,+\rangle = T|\alpha\rangle$ and $T^\dagger T = \mathbf{1}$. □

We can now prove Theorem 8.1.

Proof of Theorem 8.1. Let P' be the span program with amplified target vector defined in Theorem 7.3. Let G be the same as the graph $G_{P'}$, defined in Eq. (7.3), except scaled down by a factor of $\|\mathrm{abs}(A_{G_{P'}})\| \le \|\mathrm{abs}(A_{G_P})\|$. That is, $A_G = A_{G_{P'}}/\|\mathrm{abs}(A_{G_{P'}})\|$, so $\|\mathrm{abs}(A_G)\| = 1$. Let V be the vertex set of G, E the edge set, and for $j \in [n]$ and $b \in \{0,1\}$, let $V_{j,b}$ be the set of row vertices corresponding to $I_{j,b}$. For an input x, let $G(x)$ be $G_{P'}(x)$ scaled by $1/\|\mathrm{abs}(A_{G_{P'}})\|$. From Eqs. (7.8) and (7.9), we obtain that if $f_P(x) = 1$, then there is an eigenvalue-zero eigenvector $|\psi\rangle$ of $A_{G(x)}$ with

$$\frac{|\langle\emptyset|\psi\rangle|^2}{\||\psi\rangle\|^2} \ge \frac{1}{2} \tag{8.8}$$

and if $f_P(x) = 0$, then for all $\Upsilon \ge 0$,

$$\sum_{\alpha:\,|\rho(\alpha)|\le\Upsilon} |\langle\alpha|\emptyset\rangle|^2 \le 8\Upsilon^2\,\mathrm{fwsize}(P,\mathcal{D})^2\,\|\mathrm{abs}(A_{G_P})\|^2 \tag{8.9}$$

where $|\alpha\rangle$ and $\rho(\alpha)$ denote the eigenvectors and respective eigenvalues of $A_{G(x)}$.

Assume that G is a connected graph; otherwise, discard all components other than the one containing the output vertex $\emptyset$. Therefore $\mathrm{abs}(A_G)$ has a single principal eigenvector $|\delta\rangle$, $\mathrm{abs}(A_G)|\delta\rangle = |\delta\rangle$, with $\langle v|\delta\rangle > 0$ for all $v \in V$.

Put an arbitrary total order "$<$" on the vertices in V. For each $v \in V$, let

$$|\varphi_v\rangle = \frac{1}{\sqrt{\langle v|\delta\rangle}}\left(\sqrt{\langle v|A_G|v\rangle\langle v|\delta\rangle}|v\rangle + \sum_{w\in V:\,w<v}\sqrt{|\langle v|A_G|w\rangle|\,\langle w|\delta\rangle}|w\rangle + \sum_{\substack{w\in V:\,v<w\\ \langle v|A_G|w\rangle\neq 0}}\frac{\langle w|A_G|v\rangle}{\sqrt{|\langle v|A_G|w\rangle|}}\sqrt{\langle w|\delta\rangle}|w\rangle\right) \tag{8.10}$$

Then

$$\||\varphi_v\rangle\|^2 = \frac{1}{\langle v|\delta\rangle}\sum_{w\in V}\langle v|\,\mathrm{abs}(A_G)|w\rangle\langle w|\delta\rangle = 1\ . \tag{8.11}$$

Therefore Theorem 8.2 will apply; define T, S, U and M from Eqs. (8.2) and (8.3). Also let $\tilde{O}_x$ be the unitary

$$\tilde{O}_x|v,w\rangle = \begin{cases} -|v,w\rangle & \text{if } v \in V_{j,x_j} \text{ for some } j\in[n] \\ |v,w\rangle & \text{otherwise}\end{cases} \tag{8.12}$$

One controlled call to $\tilde{O}_x$ can be implemented using one call to the standard input oracle O_x of Figure 1.

The algorithm has three steps:

Algorithm 4:

1. Prepare the initial state $T|\emptyset\rangle$.
2. Let $\Upsilon = 1/\big(8\,\mathrm{fwsize}(P,\mathcal{D})\|\mathrm{abs}(A_{G_P})\|\big)$. Run phase estimation on $W_x = i\,\tilde{O}_x U$, with precision $\delta_p = \frac{2}{\pi}\Upsilon$ and error rate $\delta_e = 1/8$.
3. Output 1 if the measured phase is 0 or π. Otherwise output 0.

The query complexity of this algorithm is $O\big(\frac{1}{\delta_p}\log\frac{1}{\delta_e}\big) = O(1/\Upsilon) = O\big(\mathrm{fwsize}(P,\mathcal{D})\|\mathrm{abs}(A_{G_P})\|\big)$, as claimed. We will not go into the details, but as in Section 3 the phase-estimation wrapper can be removed by using a smaller amplification factor. It remains to prove completeness and soundness.

Fix an input $x \in \{0,1\}^n$. For $v \in V$, let

$$|\varphi_v^x\rangle = \begin{cases} |v\rangle & \text{if } v \in V_{j,x_j} \text{ for some } j \in [n] \\ |\varphi_v\rangle & \text{otherwise} \end{cases} \tag{8.13}$$

Apply Theorem 8.2 using the vectors $|\varphi_v^x\rangle$ to define T_x, U_x and M_x.

Lemma 8.4. *$M = A_G$ and $M_x = A_{G(x)}$. Moreover, letting $\mathbf{C}^E = \mathrm{Span}(\{|v,w\rangle : (v,w) \in E\}) \subseteq \mathbf{C}^V \otimes \mathbf{C}^V$ be the span of the edges of G, $U_x|_{\mathbf{C}^E} = \tilde{O}_x U|_{\mathbf{C}^E}$ and $T_x|\emptyset\rangle = T|\emptyset\rangle \in \mathbf{C}^E$.*

Proof. First, note that for any vertices $v, w \in V$, from Eq. (8.3) and Eq. (8.10),

$$\begin{aligned} \langle v|M|w\rangle &= \langle\varphi_v|w\rangle\langle v|\varphi_w\rangle \\ &= \langle v|A_G|w\rangle\sqrt{\frac{\langle v|\delta\rangle}{\langle w|\delta\rangle}}\sqrt{\frac{\langle w|\delta\rangle}{\langle v|\delta\rangle}} \\ &= \langle v|A_G|w\rangle \ . \end{aligned} \tag{8.14}$$

Therefore $M = A_G$.

Recall that $G(x)$ is the same as G except with the edges to vertices in $\cup_{j\in[n]}V_{j,x_j}$ removed. Consider a $v \in V_{j,x_j}$ and let $w \neq v$ be its single neighbor, so $|\varphi_v\rangle = |w\rangle$. Since $|\varphi_v^x\rangle = |v\rangle$, $\langle v|M_x|w\rangle = \langle\varphi_v^x|w\rangle\langle v|\varphi_w^x\rangle = 0$. However, for all pairs (v,w) that do not make an edge leaving some V_{j,x_j}, $\langle v|M_x|w\rangle = \langle v|M|w\rangle$. Therefore $M_x = A_{G(x)}$.

Next, we aim to show that U_xS and $\tilde{O}_xUS$ are the same when restricted to $\mathbf{C}^E$. Note that

$$\begin{aligned} US &= 2TT^\dagger - \mathbf{1}_{\mathbf{C}^V\otimes\mathbf{C}^V} \\ &= 2\sum_{v\in V}|v\rangle\langle v| \otimes |\varphi_v\rangle\langle\varphi_v| - \mathbf{1}_{\mathbf{C}^V\otimes\mathbf{C}^V} \\ &= \sum_{v\in V}|v\rangle\langle v| \otimes (2|\varphi_v\rangle\langle\varphi_v| - \mathbf{1}_{\mathbf{C}^V}) \ . \end{aligned} \tag{8.15}$$

Similarly $U_xS = \sum_v |v\rangle\langle v| \otimes (2|\varphi_v^x\rangle\langle\varphi_v^x| - \mathbf{1}_{\mathbf{C}^V})$. Therefore,

$$\begin{aligned} (US)^\dagger U_xS &= \sum_v |v\rangle\langle v| \otimes \big[(2|\varphi_v\rangle\langle\varphi_v| - \mathbf{1})(2|\varphi_v^x\rangle\langle\varphi_v^x| - \mathbf{1})\big] \\ &= \sum_{v\notin\cup_j V_{j,x_j}} |v\rangle\langle v| \otimes \mathbf{1} + \sum_{\substack{j\in[n], v\in V_{j,x_j} \\ w\sim v}} |v\rangle\langle v| \otimes (\mathbf{1} - 2|v\rangle\langle v| - 2|w\rangle\langle w|) \ , \end{aligned} \tag{8.16}$$

where in the second term w is v's single neighbor in G. On the other hand, from its definition in Eq. (8.12),

$$\tilde{O}_x = \mathbf{1}_{\mathbf{C}^V\otimes\mathbf{C}^V} - 2\sum_{j\in[n], v\in V_{j,x_j}} |v\rangle\langle v| \otimes \mathbf{1}_{\mathbf{C}^V} \ . \tag{8.17}$$

By inspection, this is the same as Eq. (8.16) on $\mathbf{C}^E$.

Finally, since by assumption $\emptyset \notin V_{\text{input}}$, $T_x|\emptyset\rangle = |\emptyset\rangle \otimes |\varphi_\emptyset^x\rangle = |\emptyset\rangle \otimes |\varphi_\emptyset\rangle = T|\emptyset\rangle$. $T|\emptyset\rangle \in \mathbf{C}^E$ by Eq. (8.10). □

The initial state $T|\emptyset\rangle = T_x|\emptyset\rangle$ lies in $\mathrm{Range}(T_x) \subseteq \mathbf{C}^E$. Also, U_x fixes $\mathbf{C}^E$; in fact, it even fixes the join of the ranges of T_x and ST_x, which could be smaller than $\mathbf{C}^E$. By Lemma 8.4, $\tilde{O}_x U$ and U_x are the same restricted to $\mathbf{C}^E$. Therefore, the algorithm behaves the same as if it were running phase estimation on iU_x instead of $W_x = i\tilde{O}_x U$.

Based on Eq. (8.8), the algorithm is complete:

Lemma 8.5. *If $x \in \mathcal{D}$ and $f(x) = 1$, then the algorithm outputs 1 with probability at least $1/2 - \delta_e$.*

Proof. Assume that $f(x) = 1$. From Eq. (8.8), $A_{G(x)}$ has an eigenvalue-zero eigenvector $|\alpha\rangle \in \mathbf{C}^V$ with $\||\alpha\rangle\| = 1$ and $|\langle\emptyset|\alpha\rangle|^2 \geq 1/2$. By Theorem 8.2 with $\rho(\alpha) = 0$, U_x has eigenvectors $|\alpha, \pm\rangle = (1 \pm iS)T_x|\alpha\rangle$ with respective eigenvalues $\pm i$. By Lemma 8.3, these satisfy

$$\frac{|\langle\emptyset|T_x^\dagger|\alpha,+\rangle|^2}{\||\alpha,+\rangle\|} + \frac{|\langle\emptyset|T_x^\dagger|\alpha,-\rangle|^2}{\||\alpha,-\rangle\|} = |\langle\emptyset|\alpha\rangle|^2 \geq \frac{1}{2} \ . \tag{8.18}$$

Thus the algorithm measures a phase of 0 or π, and outputs 1, with probability at least $1/2 - \delta_e$. □

Based on Eq. (8.9), since the phase estimation precision is $\delta_p = \frac{2}{\pi}\Upsilon$, the algorithm is sound:

Lemma 8.6. *If $x \in \mathcal{D}$ and $f(x) = 0$, then the algorithm outputs 1 with probability at most $1/8 + \delta_e$.*

Proof. Let $\{|\alpha\rangle\}$ be a complete set of orthonormal eigenvectors of $A_{G(x)}$, with corresponding eigenvalues $\rho(\alpha)$. The initial state $T|\emptyset\rangle = T_x|\emptyset\rangle$ lies in the range of T_x, and therefore is in the span of the eigenvectors $\{|\alpha, \pm\rangle\}$, i.e., the space $R = \oplus_\alpha R_\alpha$ from Theorem 8.2. The probability that the algorithm outputs 1 is therefore at most δ_e plus

$$\sum_{\substack{|\alpha,b\rangle: \\ \arg(\lambda(\alpha,b)) \in [\frac{\pi}{2}-\delta_p, \frac{\pi}{2}+\delta_p] \cup [-\frac{\pi}{2}-\delta_p, -\frac{\pi}{2}+\delta_p]}} \frac{|\langle\alpha,b|\emptyset\rangle|^2}{\||\alpha,b\rangle\|^2} = \sum_{\alpha:|\arcsin\rho(\alpha)| \leq \delta_p} \left(\frac{|\langle\alpha,+|\emptyset\rangle|^2}{\||\alpha,+\rangle\|^2} + \frac{|\langle\alpha,-|\emptyset\rangle|^2}{\||\alpha,-\rangle\|^2} \right) \tag{8.19}$$

where in the first sum b can be either $+$ or $-$, and we have used $\lambda(\alpha, \pm) = e^{\pm i \arccos\rho(\alpha)}$, so $\arg(\lambda(\alpha, \pm)) = \pm(\frac{\pi}{2} - \arcsin\rho(\alpha))$.

Since $|\arcsin\rho(\alpha)| \leq \frac{\pi}{2}|\rho(\alpha)|$, and by Lemma 8.3, the above sum is at most

$$\sum_{\alpha:|\rho(\alpha)| \leq \Upsilon} |\langle\alpha|\emptyset\rangle|^2 \ , \tag{8.20}$$

which is at most $1/8$ by Eq. (8.9). □

Therefore, the algorithm is correct. The constant gap $3/8 - 2\delta_e = 1/8$ between its completeness and soundness parameters can be amplified as usual.

Finally, for the time-complexity claim, the algorithm uses a discrete-time quantum walk on a scaled version of the graph $G_{P'}(x)$. If the maximum degree of a vertex in G_P is d, then each coin reflection can be implemented using $O(\log d)$ single-qubit unitaries and queries to the preprocessed string [GR02, CNW10]. The additional $\big(\log(Q \log d)\big)^{O(1)}$ factor comes from applying the Solovay-Kitaev Theorem [KSV02] to compile the single-qubit unitaries into products of elementary gates, to precision $1/O(Q \log d)$. □

9 Open problems

We have shown that for any boolean function f, the general adversary lower bound $\mathrm{Adv}^{\pm}(f)$ actually characterizes bounded-error quantum query complexity $Q(f)$. In proving this statement, we have also shown that quantum algorithms, judged by query complexity, and span programs, judged by witness size, are equivalent computational models for evaluating boolean functions. One consequence is an efficiently computable characterization of the quantum query complexity for a read-once formula over any constant-size gate set.

Span programs may also be useful for developing other quantum algorithms. They have a rich mathematical structure, and their potential has not been fully explored. Span programs can be constructed directly, but another approach is to solve for the general adversary bound. The $\mathrm{Adv}^{\pm}$ SDP is simpler than the previous known SDPs from [BSS03]. The simplifications may ease the inference of structure from numerical investigations. For the ordered search problem in particular, Childs and Lee have closely characterized $\mathrm{Adv}^{\pm}$ [CL08]. To the author's knowledge, neither the [BSS03] SDPs nor $\mathrm{Adv}^{\pm}$ have ever been solved *directly* for a substantially nontrivial, asymptotically large family of functions, with better than a constant-factor improvement over the adversary bound. It is the easy composition rule that allows for computing $\mathrm{Adv}^{\pm}$ for a read-once formula, by multiplying the bounds computed for constant-size gates.

It may be that the simple form of our algorithm will allow for further progress in the understanding of quantum query algorithms. For example, this form and Theorem 1.5 suggest that it may be possible to adapt the algorithm to evaluate any boolean function f with a *bounded-error* input oracle with the same asymptotic number $\Theta(\mathrm{Adv}^{\pm}(f))$ of quantum queries, following [HMW03] for the OR function. Classically, in the noisy decision-tree model, an extra logarithmic factor for error reduction is sometimes required [FRPU94], but this factor is not known to be needed for any quantum query algorithm [BNRW05].

Regarding time complexity, new techniques are needed for developing span programs P such that G_P is sparse and $\|\mathrm{abs}(A_{G_P})\| = O(1)$, so that the algorithm in Theorem 8.1 can be implemented efficiently. Some partial progress is made in [Rei09b, Rei09c]. Investigating the tradeoffs involved in designing span programs for query-optimal and nearly time-optimal quantum algorithms is an important area for further research.

By a construction in [Rei09a, Theorem 3.1], a one-sided-error quantum algorithm can be converted into a span program, and by Theorem 8.1, the span program can be converted back into a quantum algorithm. Overall, this transformation can be seen to preserve time complexity, after preprocessing. To some degree, this complements the equivalence results for best span program witness size and bounded-error quantum *query* complexity, Corollary 1.8. However, the algorithm from Theorem 8.1 has two-sided error. Perhaps a definition of "approximate" span programs would allow a closer equivalence between algorithm time complexity and span program full witness size.

Acknowledgements

I would like to thank Troy Lee and Robert Špalek for helpful discussions and feedback on early drafts. I also thank Sergio Boixo, Chen-Fu Chiang, Stephen Jordan, Valentine Kabanets, Julia Kempe, Rajat Mittal and Rolando Somma for useful comments. Research supported by NSERC, ARO/DTO and MITACS.

References

[AA09] Scott Aaronson and Andris Ambainis. The need for structure in quantum speedups. 2009, arXiv:0911.0996 [quant-ph].

[Aar10] Scott Aaronson. BQP and the polynomial hierarchy. In *Proc. 42nd ACM STOC*, pages 141–150, 2010, arXiv:0910.4698 [quant-ph].

[ACR+10] 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. *SIAM J. Comput.*, 39(6):2513–2530, 2010. Earlier version in FOCS'07.

[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.

[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, arXiv:quant-ph/0305179.

[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.

[Amb07] Andris Ambainis. A nearly optimal discrete query quantum algorithm for evaluating NAND formulas. 2007, arXiv:0704.3628 [quant-ph].

[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. Earlier version in FOCS'98.

[BBHT98] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5):493–505, 1998, arXiv:quant-ph/9605034. Earlier version in *Proc. 4th Workshop on Physics and Computation*, pp. 36-43, 1996.

[BHT98] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum algorithm for the collision problem. In *Proc. 3rd LATIN*, LNCS vol. 1380, pages 163–169, 1998, arXiv:quant-ph/9705002.

[BNRW05] Harry Buhrman, Ilan Newman, Hein Röhrig, and Ronald de Wolf. Robust polynomials and quantum algorithms. In *Proc. 22nd STACS*, LNCS vol. 3404, pages 593–604, 2005, arXiv:quant-ph/0309220.

[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 query complexity and semidefinite programming. In *Proc. 18th IEEE Complexity*, pages 179–193, 2003.

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997. Earlier version in STOC'93.

[BW02] Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002.

[CGM+09] Richard Cleve, Daniel Gottesman, Michele Mosca, Rolando D. 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].

[CNW10] Chen-Fu Chiang, Daniel Nagaj, and Pawel Wocjan. Efficient circuits for quantum walks. *Quantum Inf. Comput.*, 10(5&6):420–434, 2010, arXiv:0903.3465 [quant-ph].

[CRŠZ07] Andrew M. Childs, Ben W. Reichardt, Robert Špalek, and Shengyu Zhang. Every NAND formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. 2007, arXiv:quant-ph/0703015.

[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.

[FRPU94] Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM J. Comput.*, 23(5):1001–1018, 1994. Earlier version in STOC'90.

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

[GR02] Lov K. Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. 2002, arXiv:quant-ph/0208112.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. 28th ACM STOC*, pages 212–219, 1996, arXiv:quant-ph/9605043.

[HLŠ06] Peter Høyer, Troy Lee, and Robert Špalek. Source codes of semidefinite programs for ADV$^{\pm}$. http://www.ucw.cz/~robert/papers/adv/, 2006.

[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.

[HMW03] Peter Høyer, Michele Mosca, and Ronald de Wolf. Quantum search on bounded-error inputs. In *Proc. 30th ICALP*, pages 291–299, 2003, LNCS 2719. arXiv:quant-ph/0304052.

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

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

[HW91] Rafi Heiman and Avi Wigderson. Randomized vs. deterministic decision tree complexity for read-once boolean functions. *Computational Complexity*, 1(4):311–329, 1991. Earlier version in Structure in Complexity Theory '91.

[Jor75] Camille Jordan. Essai sur la géométrie à n dimensions. *Bulletin de la S. M. F.*, 3:103–174, 1875.

[KOS07] Emanuel Knill, Gerardo Ortiz, and Rolando D. Somma. Optimal quantum measurements of expectation values of observables. *Phys. Rev. A*, 75:012328, 2007, arXiv:quant-ph/0607019.

[KSV02] Alexei Yu. Kitaev, Alexander H. Shen, and Mikhail N. Vyalyi. *Classical and Quantum Computation*, volume 47 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, Rhode Island, 2002.

[Kut05] Samuel A. Kutin. Quantum lower bound for the collision problem with small range. *Theory of Computing*, 1:29–36, 2005, arXiv:quant-ph/0304162.

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

[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.

[LMRŠ10] Troy Lee, Rajat Mittal, Ben W. Reichardt, and Robert Špalek. In preparation, 2010.

[MW05] Chris Marriott and John Watrous. Quantum Arthur-Merlin games. *Computational Complexity*, 14(2):122152, 2005, arXiv:cs/0506068 [cs.CC].

[NWZ09] Daniel Nagaj, Pawel Wocjan, and Yong Zhang. Fast amplification of QMA. *Quantum Inf. Comput.*, 9:1053–1068, 2009, arXiv:0904.1549 [quant-ph].

[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]. Extended abstract in *Proc. 50th IEEE FOCS*, pages 544–551, 2009.

[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].

[Rei10a] Ben W. Reichardt. Least span program witness size equals the general adversary lower bound on quantum query complexity. Technical Report TR10-075, Electronic Colloquium on Computational Complexity, http://eccc.hpi-web.de, 2010.

[Rei10b] Ben W. Reichardt. Reflections for quantum query algorithms. 2010, arXiv:1005.1601 [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].

[San95] Miklos Santha. On the Monte Carlo decision tree complexity of read-once formulae. *Random Structures and Algorithms*, 6(1):75–87, 1995. Earlier version in *Proc. 6th IEEE Structure in Complexity Theory*, 1991.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997, arXiv:quant-ph/0508027. Earlier version in FOCS'94.

[Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM J. Computing*, 26(5):1474–1483, 1997. Earlier version in FOCS'94.

[Sni85] Marc Snir. Lower bounds on probabilistic linear decision trees. *Theoretical Computer Science*, 38:69–82, 1985.

[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.

[SW86] Michael Saks and Avi Wigderson. Probabilistic Boolean decision trees and the complexity of evaluating game trees. In *Proc. 27th IEEE FOCS*, pages 29–38, 1986.

[Sze03] Mario Szegedy. On the quantum query complexity of detecting triangles in graphs. 2003, arXiv:quant-ph/0310107.

[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.

Institute for Quantum Computing, University of Waterloo
E-mail address: breic@iqc.ca

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