THEORY OF COMPUTING, Volume 18 (11), 2022, pp. 1–49
www.theoryofcomputing.org

Span Programs and Quantum Space Complexity

Stacey Jeffery*

Received October 7, 2020; Revised November 3, 2021; Published May 24, 2022

Abstract. While quantum computers hold the promise of significant computational speedups, the limited size of early quantum machines motivates the study of space-bounded quantum computation. We relate the quantum space complexity of computing a function f with *one-sided error* to the logarithm of its *span program size* over the reals, a classical quantity that is well-studied in attempts to prove formula size lower bounds.

In the more natural *bounded error* model, we show that the amount of space needed for a unitary quantum algorithm (i. e., an algorithm that makes no measurements until the final step) to compute f with bounded (two-sided) error is at least the logarithm of its *approximate span program size*. Approximate span programs have been introduced in the field of quantum algorithms but not studied classically. However, the approximate span program size of a function is a natural generalization of its span program size.

A conference version of this paper appeared in the Proceedings of the 11th Innovations in Theoretical Computer Science Conference, 2020 [17].

*Supported by an NWO WISE Fellowship, an NWO Veni Innovational Research Grant under project number 639.021.752, and QuantERA project QuantAlgo 680-91-03. SJ is a CIFAR Fellow in the Quantum Information Science Program.

ACM Classification: F.1.1, F.1.3

AMS Classification: 81P68

Key words and phrases: quantum computing, quantum space complexity, span programs

 DOI: 10.4086/toc.2022.v018.a011

While no non-trivial lower bound is known on the span program size (or approximate span program size) of any explicit function, a number of lower bounds are known on the *monotone span program size*. We show that the approximate monotone span program size of f is a lower bound on the space needed by quantum algorithms of a particular form, called *monotone phase estimation algorithms*, to compute f. We then give the first non-trivial lower bound on the approximate monotone span program size of an explicit function.

1 Introduction

While quantum computers hold the promise of significant speedups for a number of problems, building them is a serious technological challenge, and it is expected that early quantum computers will have quantum memories of very limited size. This motivates the theoretical question: what problems could we solve faster on a quantum computer with limited space? Or similarly, what is the minimum number of qubits needed to solve a given problem (and hopefully still get a speedup)?

We take a modest step towards answering such questions, by relating the space complexity of a function f to its *span program size* (see Definition 3.3), which is a measure that has received significant attention in theoretical computer science over the past few decades. Span programs are a model of computation introduced by Karchmer and Wigderson [20] in an entirely classical setting; they defined the span program size of a function in order to lower bound the size of *counting branching programs*. Some time later, Reichardt and Špalek [28] related span programs to quantum algorithms, and introduced the new measure of *span program complexity* (see Definition 3.4). The importance of span programs in quantum algorithms stems from the ability to compile any span program for a function f into a bounded error quantum algorithm for f [27]. In particular, there is a tight correspondence between the span program *complexity* of f, and its quantum query complexity—a rather surprising and beautiful connection for a model originally introduced outside the realm of quantum computing. In contrast, the classical notion of span program *size* had received no attention in the quantum computing literature before now.

Ref. [15] defined the notion of an approximate span program for a function f, and showed that even an approximate span program for f can be compiled into a bounded error quantum algorithm for f. In this paper, we further relax the definition of an approximate span program for f, making analysis of such algorithms significantly easier (see Definition 3.6).

Let $\mathsf{S}_U(f)$ denote the *bounded error unitary space complexity of f*, or the minimum space needed by a unitary quantum algorithm—i. e., an algorithm that makes no measurements until the final step—that computes f with bounded error (see Definition 2.3). In [10] and [12], independently, it was shown that $\mathsf{S}_U(f) = \mathsf{S}(f)$ (up to constants), where $\mathsf{S}(f)$ denotes the *bounded error space complexity of f*, without the restriction to algorithms that are unitary. Our results are proven for $\mathsf{S}_U(f)$, but the results of [10, 12] imply that they also apply to $\mathsf{S}(f)$. A similar statement is not known for the *one-sided error* unitary quantum space complexity, $\mathsf{S}_U^1(f)$, though we suspect that it also holds, and a proof of this would strengthen our results about $\mathsf{S}_U^1(f)$ to also hold for $\mathsf{S}^1(f)$.

For a function $f : \{0,1\}^n \to \{0,1\}$, we can assume that the input is accessed by queries, so that we do not need to store the full n-bit input in working memory, but we need at least $\log n$ bits of memory to store an index into the input. Thus, a lower bound of $\omega(\log n)$ on $\mathsf{S}(f)$ for some f would be considered non-trivial.

Letting $\mathsf{SP}(f)$ denote the minimum size of a span program deciding f, and $\widetilde{\mathsf{SP}}(f)$ the minimum size of a span program *approximating* f (see Definition 3.7), we have the following (see Theorem 4.1):

Theorem 1.1 (Informal)**.** *For any Boolean function f, if $\mathsf{S}(f)$ denotes its bounded error quantum space complexity, and $\widetilde{\mathsf{SP}}(f)$ its approximate span program size, then*

$$\mathsf{S}(f) \geq \log \widetilde{\mathsf{SP}}(f).$$

Similarly, if $\mathsf{S}_U^1(f)$ denotes its one-sided error unitary space complexity, and $\mathsf{SP}(f)$ its span program size, then

$$\mathsf{S}_U^1(f) \geq \log \mathsf{SP}(f).$$

In the case of bounded (two-sided) error, this lower bound is tight in the following sense (corollary of Theorem 3.1 and 3.2):

Theorem 1.2 (Informal)**.** *The class of languages decidable in bounded error by a quantum algorithm with space $O(S)$ and $2^{O(S)}$ queries*[1] *is equal to the class of languages approximated by a span program of size and complexity $2^{O(S)}$.*

The relationship between span program size and quantum space complexity is rather natural, as the span program size of f is known to lower bound the minimum size of a symmetric branching program for f, and the logarithm of the branching program size of a function f characterizes its classical deterministic space complexity.

The inequality $\mathsf{S}_U^1(f) \geq \log \mathsf{SP}(f)$ in Theorem 1.1 follows from a construction of [27] for converting a one-sided error quantum algorithm for f into a span program for f. We adapt this construction to show how to convert a bounded (two-sided) error unitary quantum algorithm for f with query complexity T and space complexity $S \geq \log T$ into an approximate span program for f with complexity $\Theta(T)$ and size $2^{\Theta(S)}$, proving $\mathsf{S}_U(f) \geq \Omega(\log \widetilde{\mathsf{SP}}(f))$, and thus $\mathsf{S}(f) \geq \Omega(\log \widetilde{\mathsf{SP}}(f))$. The connection between $\mathsf{S}(f)$ and $\log \widetilde{\mathsf{SP}}(f)$ is tight up to an additive term of the logarithm of the minimum complexity of any span program for f with optimal size, yielding Theorem 1.2. This follows from the fact that an approximate span program can be compiled into a quantum algorithm in a way that similarly preserves the correspondence between space complexity and (logarithm of) span program size, as well as the correspondence between query complexity and span program complexity (see Theorem 3.1). While the preservation of

[1]Depending on the precise model of computation, it is without loss of generality to assume that the space is at least logarithmic in the number of queries. In our model of unitary quantum algorithms (see Section 2), this is a reasonable assumption since we would need to use a counter of size at least logarithmic in the query complexity to know which unitary to apply. In the case of a quantum Turing machine that halts absolutely, if there is ever a pair of time steps $t \neq t'$ such that the state of the machine at step t and the state at step t' are non-orthogonal, then some (exponentially decreasing) branch of the computation will run forever, which is a contradiction.

the correspondence between query complexity and span program complexity (in both directions) is not necessary for our results, it may be useful in future work for studying lower bounds on time and space simultaneously.

The significance of Theorem 1.1 is that span program size has received extensive attention in theoretical computer science. Using results from [5], the connection in Theorem 1.1 immediately implies the following (Theorem 4.2):

Theorem 1.3. *For almost all Boolean functions f on n bits, $\mathsf{S}_U^1(f) = \Omega(n)$.*

If we make a uniformity assumption that the quantum space complexity of an algorithm is at least the logarithm of its time complexity, then Theorem 1.3 would follow from a lower bound of $\Omega(2^n)$ on the quantum time complexity of almost all n-bit Boolean functions. Notwithstanding, the proof via span program size is evidence of the power of the technique.

In the pursuit of lower bounds on span program size of explicit functions, several nice expressions lower bounding $\mathsf{SP}(f)$ have been derived. By adapting one such lower bound on $\mathsf{SP}(f)$ to $\widetilde{\mathsf{SP}}(f)$, we get the following (see Lemma 4.6):

Theorem 1.4 (Informal)**.** *For any Boolean function f, and* partial matrix[2] $M \in (\mathbb{R} \cup \{\star\})^{f^{-1}(0)\times f^{-1}(1)}$ *with $\|M\|_\infty \leq 1$:*

$$\mathsf{S}(f) \geq \Omega\left(\log\left(\frac{\frac{1}{2}\text{-rank}(M)}{\max_{i\in[n]}\text{rank}(M\circ\Delta_i)}\right)\right),$$

where $\circ$ denotes the entrywise product, and $\Delta_i[x,y] = 1$ if $x_i \neq y_i$ and 0 else.

Above, $\frac{1}{2}$-rank denotes the approximate rank, or the minimum rank of any matrix $\widetilde{M}$ such that $|M[x,y] - \widetilde{M}[x,y]| \leq \frac{1}{2}$ for each x, y such that $M[x,y] \neq \star$. If we replace $\frac{1}{2}$-rank(M) with rank(M), we get the logarithm of an expression called the *rank measure*, introduced by Razborov [25]. The rank measure was shown by Gàl to be a lower bound on span program size, SP [11], and thus, our results imply that the log of the rank measure is a lower bound on S_U^1. It is straightforward to extend this proof to the approximate case to get Theorem 1.4.

Theorem 1.4 seems to give some hope of proving a non-trivial—that is, $\omega(\log n)$—lower bound on the quantum space complexity of some explicit f, by exhibiting a matrix M for which the (approximate) rank measure is $2^{\omega(\log n)}$. In [25], Razborov showed that the rank measure is a lower bound on the Boolean formula size of f, motivating significant attempts to prove lower bounds on the rank measure of explicit functions. The bad news is, circuit lower bounds have been described as "Complexity theory's Waterloo" [4]. Despite significant effort, no non-trivial lower bound on span program size for any f is known.

Due to the difficulty of proving explicit lower bounds on span program size, earlier work has considered the easier problem of lower bounding *monotone* span program size, $\mathsf{mSP}(f)$. For a monotone function f, the monotone span program size of f, $\mathsf{mSP}(f)$ is the minimum size of any *monotone span program* for f (see Definition 5.1). We can similarly define the *approximate monotone span program size* of f, $\mathsf{m}\widetilde{\mathsf{SP}}(f)$ (see Definition 5.1). Although $\log \mathsf{m}\widetilde{\mathsf{SP}}(f)$ is *not* a lower bound

[2]Note that M depends on f in that it is indexed by the 0- and 1-inputs of f.

on $\mathsf{S}(f)$, even for monotone f, it is a lower bound on the space complexity of any algorithm obtained by compiling a monotone span program. We show that such algorithms are equivalent to a more natural class of algorithms called monotone phase estimation algorithms. Informally, a phase estimation algorithm is an algorithm that works by performing phase estimation of some unitary that makes one query to the input, and estimating the amplitude on a 0 in the phase register (see Definition 5.12). Any quantum algorithm can be put into this form in a way that preserves its space, query, and even time complexity. A monotone phase estimation algorithm is a phase estimation algorithm where, loosely speaking, adding 0s to the input can only make the algorithm more likely to reject (see Definition 5.13). This includes, for example, the phase estimation variant of Grover's algorithm. We can then prove the following (see Theorem 5.14):

Theorem 1.5 (Informal). *For any Boolean function f, any bounded error monotone phase estimation algorithm for f has space complexity at least $\log \widetilde{\mathsf{mSP}}(f)$, and any one-sided error monotone phase estimation algorithm for f has space complexity at least $\log \mathsf{mSP}(f)$.*

Fortunately, non-trivial lower bounds for the monotone span program complexity are known for explicit functions. In Ref. [5], Babai, Gàl and Wigderson showed a lower bound of $\mathsf{mSP}(f) \geq 2^{\Omega\left(\log^2(n)/\log\log(n)\right)}$ for some explicit function f, which was later improved to $\mathsf{mSP}(f) \geq 2^{\Omega(\log^2(n))}$ by Gàl [11]. In Ref. [29], a function f was exhibited with $\mathsf{mSP}(f) \geq 2^{n^{\epsilon}}$ for some constant $\epsilon \in (0,1)$, and in the strongest known result, Pitassi and Robere exhibited a function f with $\mathsf{mSP}(f) \geq 2^{\Omega(n)}$ [24]. Combined with our results, each of these implies a lower bound on the space complexity of one-sided error monotone phase estimation algorithms. For example, the result of [24] implies a lower bound of $\Omega(n)$ on the space complexity of one-sided error monotone phase estimation algorithms for a certain satisfiability problem f. This lower bound, and also the one in [29], are proven by choosing f based on a constraint satisfaction problem with high *refutation width*, which is a measure related to the space complexity of certain classes of SAT solvers, so it is intuitively not surprising that these problems should require a large amount of space to solve with one-sided error.

For the case of bounded error space complexity, we also prove the following (see Theorem 5.3, Corollary 5.15):

Theorem 1.6 (Informal). *There exists a function $f : \{0,1\}^n \to \{0,1\}$ such that any bounded error monotone phase estimation algorithm for f has space complexity $(\log n)^{2-o(1)}$.*

This lower bound is non-trivial, although much less so than the best known lower bound of $\Omega(n)$ for the one-sided case. The approximate monotone span program lower bound from which Theorem 1.6 follows also implies a new lower bound of $2^{(\log n)^{2-o(1)}}$ on the (non-approximate) monotone span program size of the function f in Theorem 1.6 (although, as previously mentioned, there are much better lower bounds for monotone span program size of other explicit functions).

To prove the lower bound in Theorem 1.6, we apply a new technique that leverages the best possible gap between the certificate complexity and approximate polynomial degree of a function, employing a function $g : \{0,1\}^{m^{2+o(1)}} \to \{0,1\}$ from [8],[3] whose certificate complexity is $m^{1+o(1)}$, and whose approximate degree is $m^{2-o(1)}$. Following a strategy of [29], we use this g to construct a *pattern matrix* [30] (see Definition 5.8) and use this matrix in a monotone version of Theorem 1.4 (see Theorem 5.4). The fact that certificate complexity and approximate degree of total functions are related by $\widetilde{\deg}_{1/3}(g) \leq C(g)^2$ for all g is a barrier to proving a lower bound better than $(\log n)^2$ using this technique, but we also give a generalization that has the potential to prove significantly better lower bounds (see Lemma 5.11).

Discussion and open problems The most conspicuous open problem of this work is to prove a lower bound of $\omega(\log n)$ on $\mathsf{S}(f)$ or even $\mathsf{S}_U^1(f)$ for some explicit decision function f. It is known that any space S quantum Turing machine can be simulated by a deterministic classical algorithm in space S^2 [31], so a lower bound of $\omega(\log^2 n)$ on classical space complexity would also give a non-trivial lower bound on quantum space complexity. If anything, the relationship to span program size is evidence that this task is extremely difficult.

We have shown a lower bound of $2^{(\log n)^{2-o(1)}}$ on the approximate monotone span program complexity of an explicit monotone function f, which gives a lower bound of $(\log n)^{2-o(1)}$ on the bounded error space complexity needed by a quantum algorithm of a very specific form: a monotone phase estimation algorithm. This is much worse than the best bound we can get in the one-sided case: a lower bound of $\Omega(n)$ for some explicit function. An obvious open problem is to try to get a better lower bound on the approximate monotone span program complexity of some explicit function.

Our lower bound of $(\log n)^{2-o(1)}$ only applies to the space complexity of monotone phase estimation algorithms and does not preclude the existence of a more space-efficient algorithm of a different form for f. We do know that phase estimation algorithms are fully general, in the sense that every function has a space-optimal phase estimation algorithm. Does something similar hold for monotone functions and monotone phase estimation algorithms? This would imply that $\log \mathsf{m}\widetilde{\mathsf{SP}}(f)$ is a lower bound on $\mathsf{S}(f)$ for all monotone functions f.

In this paper, we define an approximate version of the rank method, and monotone rank method, and in case of the monotone rank method, give an explicit non-trivial lower bound. The rank method is known to give lower bounds on formula size, and the monotone rank method on monotone formula size. An interesting question is whether the approximate rank method also gives lower bounds on some complexity theoretic quantity related to (classical) formulas.

Our results are a modest first step towards understanding unitary quantum space complexity, but even if we could lower bound the unitary quantum space complexity of an explicit function, there are several obstacles limiting the practical consequences of such a result. First, while an early quantum computer will have a small *quantum* memory, it is simple to augment it with a much larger classical memory. Thus, in order to achieve results with practical implications, we would need to study computational models that make a distinction between quantum and classical memories. We leave this as an important challenge for future work.

[3] An earlier version of this paper used a function described in [1] with a 7/6-separation between certificate complexity and approximate degree. We thank Robin Kothari for pointing us to the improved result of [8].

Second, we are generally only interested in running quantum algorithms when we get an advantage over classical computers in the time complexity, so results that give a lower bound on the quantum space required if we wish to keep the time complexity small, such as time-space lower bounds, are especially interesting. While we do not address time-space lower bounds in this paper, one advantage of the proposed quantum space lower bound technique, via span programs, is that span programs are also known to characterize quantum query complexity, which is a lower bound on time complexity. We leave exploration of this connection for future work.

We mention two previous characterizations of $\mathsf{S}(f)$. Ref. [19] showed that $\mathsf{S}(f)$ is equal to the logarithm of the minimum width of a *matchgate circuit* computing f, and thus our results imply that this minimum matchgate width is approximately equal to the approximate span program size of f. Separately, in Ref. [9], Fefferman and Lin showed that for every function k, inverting $2^{k(n)} \times 2^{k(n)}$ matrices is complete for the class of problems f such that $\mathsf{S}(f) \leq k(n)$. Our results imply that evaluating an approximate span program of size $2^{k(n)}$ (for some suitable definition of the problem) is similarly complete for this class. Evaluating an approximate span program boils down to deciding if $\|A(x)^+|w_0\rangle\|$, for some matrix $A(x)$ partially determined by the input x, and some initial state $|w_0\rangle$, is below a certain threshold, so these results are not unrelated.[4] We leave exploring these connections as future work.

Organization The remainder of this paper is organized as follows. In Section 2, we present the necessary notation and quantum algorithmic preliminaries, and define quantum space complexity. In Section 3, we define span programs, and describe how they correspond to quantum algorithms. In particular, we describe how a span program can be "compiled" into a quantum algorithm (Section 3.2), and how a quantum algorithm can be turned into a span program (Section 3.3), with both transformations more or less preserving the relationships between span program size and algorithmic space, and between span program complexity and query complexity. From this correspondence, we obtain, in Section 4, expressions that lower bound the quantum space complexity of a function. While we do not know how to instantiate any of these expressions to get a non-trivial lower bound for an explicit function, in Section 5, we consider to what extent monotone span program lower bounds are meaningful lower bounds on variants of quantum space complexity, and give the first non-trivial lower bound on the approximate monotone span program size of a function.

2 Preliminaries

We begin with some miscellaneous notation. For a vector $|v\rangle$, we let $\||v\rangle\|$ denote its ℓ_2-norm. In the following, let A be a matrix with i and j indexing its rows and columns. Define:

$$\|A\|_\infty = \max_{i,j} |A_{i,j}|, \quad \text{and} \quad \|A\| = \max\{\|A|v\rangle\| : \||v\rangle\| = 1\}.$$

[4]Here, $A(x) = A\Pi_{H(x)}$, where A is as in Definition 3.3, $|w_0\rangle = A^+|\tau\rangle$ for $|\tau\rangle$ as in Definition 3.3, and $H(x)$ is as in Definition 3.4. $A(x)^+$ denotes the pseudo-inverse of $A(x)$. Then one can verify that $w_+(x) = \|A(x)^+|w_0\rangle\|^2$ (see Definition 3.4).

Following [2], define the ε-rank of a matrix A as the minimum rank of any matrix B such that $\|A-B\|_\infty \leq \varepsilon$. For a matrix A with singular value decomposition $A=\sum_k \sigma_k |v_k\rangle\langle u_k|$, where we assume $\forall k$, $\sigma_k>0$, define:

$$\mathrm{col}(A)=\mathrm{span}\{|v_k\rangle\}_k, \quad \mathrm{row}(A)=\mathrm{span}\{|u_k\rangle\}_k, \quad \ker(A)=\mathrm{row}(A)^\perp, \quad A^+=\sum_k \frac{1}{\sigma_k}|u_k\rangle\langle v_k|.$$

The following lemma, from [22], is useful in the analysis of quantum algorithms.

Lemma 2.1 (Effective spectral gap lemma). *Fix orthogonal projectors Π_A and Π_B. Let $U=(2\Pi_A-I)(2\Pi_B-I)$, and let Π_Θ be the orthogonal projector onto the $\mathrm{e}^{i\theta}$-eigenspaces of U such that $|\theta|\leq\Theta$. Then if $\Pi_A|u\rangle=0$, then $\|\Pi_\Theta\Pi_B|u\rangle\|\leq\frac{\Theta}{2}\,\||u\rangle\|$.*

In general, we will let Π_V denote the orthogonal projector onto V, for a subspace V.

Unitary quantum algorithms and space complexity A *unitary quantum algorithm* $\mathcal{A}=\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ is a family (parametrized by n) of sequences of $2^{s(n)}$-dimensional unitaries $U_1^{(n)},\dots,U_{T(n)}^{(n)}$, for some $s(n)\geq\log n$ and $T(n)$. (We will generally dispense with the explicit parametrization by n). For $x\in\{0,1\}^n$, let O_x be the unitary that acts as $O_x|j\rangle=(-1)^{x_j}|j\rangle$ for $j\in[n]$, and $O_x|0\rangle=|0\rangle$. We let $\mathcal{A}(x)$ denote the random variable obtained from measuring

$$U_T O_x U_{T-1}\dots O_x U_1|0\rangle$$

with some two-outcome measurement that should be clear from context. We call $T(n)$ the *query complexity* of the algorithm, and $S(n)=s(n)+\log T(n)$ the *space complexity*. By including a $\log T(n)$ term in the space complexity, we are implicitly assuming that the algorithm must maintain a counter to know which unitary to apply next. This is a fairly mild uniformity assumption (that is, any uniformly generated algorithm uses $\Omega(\log T)$ space), and it will make the statement of our results much simpler. The requirement that $s(n)\geq\log n$ is to ensure that the algorithm has enough space to store an index $i\in[n]$ into the input.

Remark 2.2. Since T is the number of queries made by the algorithm, we may be tempted to assume that it is at most n, however, while every n-bit function can be computed in n queries, this may not be the case when space is restricted. For example, it is difficult to imagine an algorithm that uses $O(\log n)$ space and $o(n^{3/2})$ quantum queries to solve the following problem on $[q]^n\equiv\{0,1\}^{n\log q}$: Decide whether there exist distinct $i,j,k\in[n]$ such that $x_i+x_j+x_k=0$ mod q.

For a (partial) function $f:D\to\{0,1\}$ for $D\subseteq\{0,1\}^n$, we say that $\mathcal{A}$ computes f with bounded error if for all $x\in D$, $\mathcal{A}(x)=f(x)$ with probability at least $2/3$. We say that $\mathcal{A}$ computes f with one-sided error if in addition, for all x such that $f(x)=1$, $\mathcal{A}(x)=f(x)$ with probability 1.

Definition 2.3 (Unitary Quantum Space). For a family of functions $f:D\to\{0,1\}$ for $D\subseteq\{0,1\}^n$, the unitary space complexity of f, $\mathsf{S}_U(f)$, is the minimum $S(n)$ such that there is a family of unitary quantum algorithms with space complexity $S(n)$ that computes f with bounded error. Similarly, $\mathsf{S}_U^1(f)$ is the minimum $S(n)$ such that there is a family of unitary quantum algorithms with space complexity $S(n)$ that computes f with one-sided error.

In general, quantum algorithms need not be of the strict unitary form described above, as a quantum computer is not restricted to only measure at the end of the algorithm. If one only cares about time complexity, then it is without loss of generality to assume that all measurements happen in the final step of the algorithm, because one can simply set aside any register that is to be measured, to be used as a "read-only" register (that is, strictly as a control) for the remainder of the computation. However, it is not obvious that this would not increase the space complexity, since any register that should have been measured is not available for re-use. It was recently shown that, in fact, even for space complexity, there is no loss of generality in considering unitary quantum algorithms [10, 12]; if we let $\mathsf{S}(f)$ denote the minimum space complexity of *any* quantum algorithm that computes f with bounded error, $\mathsf{S}(f) = \mathsf{S}_U(f)$. Thus, we can restrict our attention to unitary quantum algorithms for the remainder of this article but all of our results in the bounded error setting also hold for non-unitary algorithms [10, 12]. At the time of writing, there is no analogous result for $\mathsf{S}_U^1(f)$, but we suspect it holds along similar lines.

Phase estimation For a unitary U acting on H and a state $|\psi\rangle \in H$, we will say we perform T *steps of phase estimation of U on $|\psi\rangle$* when we compute

$$\frac{1}{\sqrt{T}}\sum_{t=0}^{T-1}|t\rangle U^t|\psi\rangle,$$

and then perform a quantum Fourier transform over $\mathbb{Z}/T\mathbb{Z}$ on the first register, called the *phase register*. This procedure was introduced in [21]. It is easy to see that the complexity (either query or time) of phase estimation is $O(T)$ times the complexity of implementing a controlled call to U. The space complexity of phase estimation is $\log T + \log\dim(H)$.

Informally: we will use the fact that if $U|\psi\rangle = |\psi\rangle$, then performing T steps of phase estimation of U on $|\psi\rangle$ and measuring the phase register results in outcome 0 with probability 1; and if $U|\psi\rangle = e^{i\theta}|\psi\rangle$ for some $\theta \in (-\pi, \pi]$ with $|\theta| > 0$, then performing sufficiently large $T = \Omega(1/|\theta|)$ steps of phase estimation results in outcome 0 with probability bounded by a constant below 1. Formally: for the results in Section 3.2, we refer to the proof of [15, Lemma 3.2] where formal results about phase estimation are exploited; for the results in Section 5.2, we prove the specific properties of phase estimation needed for our purposes in Lemma 5.18 and 5.19.

We note that we can increase the success probability to any constant by adding some constant number k of phase registers, and doing phase estimation k times in parallel, still using a single register for U, and taking the majority. This still has space complexity $\log\dim H + O(\log T)$.

Amplitude estimation For a unitary U acting on H, a state $|\psi\rangle \in H$, and an orthogonal projector Π on H, we will say we perform M *steps of amplitude estimation of U on $|\psi\rangle$ with respect* *to* Π when we perform M steps of phase estimation of

$$U(2|\psi\rangle\langle\psi| - I)U^{\dagger}(2\Pi - I)$$

on $U|\psi\rangle$, then, if the phase register contains some $t \in \{0,\dots,M-1\}$, compute $\tilde{p} = \sin^2 \frac{\pi t}{2M}$, which is an estimate of $\|\Pi U|\psi\rangle\|^2$ in a new register. The (time or query) complexity of this is $O(M)$ times the complexity of implementing a controlled call to U, implementing a controlled call to $2\Pi - I$, and generating $|\psi\rangle$. The space complexity is $\log T + \log\dim H + O(1)$. We have the following guarantee [7]:

Lemma 2.4. *Let $p = \|\Pi U|\psi\rangle\|^2$. There exists $\Delta = \Theta(1/M)$ such that when $\tilde{p}$ is obtained as above from M steps of amplitude estimation, with probability at least $1/2$, $|\tilde{p} - p| \le \Delta$.*

We will thus also refer to M steps of amplitude estimation as *amplitude estimation to precision* $1/M$.

3 Span programs and quantum algorithms

In Section 3.1, we will define a span program, its size and complexity, and what it means for a span program to approximate a function f. In Section 3.2, we will prove the following, which implies that the first part of Theorem 1.1 is essentially tight.

Theorem 3.1. *Let $f : D \to \{0,1\}$ for $D \subseteq \{0,1\}^n$, and let P be a span program that κ-approximates f with size K and complexity C, for some constant $\kappa \in (0,1)$. Then there exists a unitary quantum algorithm $\mathcal{A}_P$ that decides f with bounded error in space $S = O(\log K + \log C)$ using $T = O(C)$ queries to x.*

Finally, in Section 3.3, we prove the following theorem, which implies Theorem 1.1:

Theorem 3.2. *Let $f : D \to \{0,1\}$ for $D \subseteq \{0,1\}^n$ and let $\mathcal{A}$ be a unitary quantum algorithm using T queries, and space S to compute f with bounded error. Then for any constant $\kappa \in (0,1)$, there is a span program $P_{\mathcal{A}}$ with size $s(P_{\mathcal{A}}) \le 2^{O(S)}$ that κ-approximates f with complexity $C_\kappa \le O(T)$. If $\mathcal{A}$ decides f with one-sided error, then $P_{\mathcal{A}}$ decides f.*

A statement similar to Theorem 3.1 for the case of exact ($\kappa = 1$) span programs[5] was proven in [27]. Later this was generalized to the case of approximate span program [15], but a slightly more constrained notion of approximation was used, which would not allow us to prove Theorem 3.2. Neither of these works explicitly mentioned space complexity, although the analysis of the space complexity follows easily.

Theorem 3.2 is proven by exhibiting a construction that maps a bounded-error quantum algorithm for f to a span program that approximates it. This is based on a similar construction in [27] that maps a *one-sided error* quantum algorithm for f to a span program that decides it exactly. Interestingly, the fact that span program complexity is a lower bound on query complexity was known even without a mapping from bounded-error quantum algorithms to

[5] See Section 3.1 for definitions of exact and approximate span programs.

span programs. This was proven by showing [27] that the semidefinite minimization problem whose solution is the minimum span program complexity of a span program that decides f is the dual of a semidefinite program that was known to be a lower bound on quantum query complexity [3, 6, 14].

3.1 Span programs

Span programs were first introduced in the context of classical complexity theory in [20], where they were used to study counting classes for nondeterministic logspace machines. While span programs can be defined with respect to any field, we will consider span programs over $\mathbb{R}$ (or equivalently, $\mathbb{C}$, when convenient, see Remark 3.10). We use the following definition, slightly modified from [20]:

Definition 3.3 (Span Program and Size)**.** A *span program on* $\{0,1\}^n$ consists of:

- Finite inner product spaces $\{H_{j,b}\}_{j\in[n],b\in\{0,1\}} \cup \{H_{\text{true}}, H_{\text{false}}\}$. We define $H = \bigoplus_{j,b} H_{j,b} \oplus H_{\text{true}} \oplus H_{\text{false}}$, and for every $x \in \{0,1\}^n$, $H(x) = H_{1,x_1} \oplus \cdots \oplus H_{n,x_n} \oplus H_{\text{true}}$.[6]
- A vector space V.
- A *target vector* $|\tau\rangle \in V$.[7]
- A linear map $A : H \to V$.

We specify this span program by $P = (H, V, |\tau\rangle, A)$, and leave the decomposition of H implicit. The *size* of the span program is $s(P) = \dim H$.

To recover the classical definition from [20], we can view A as a matrix, with each of its columns labelled by some $(j,b) \in [n] \times \{0,1\}$ (or "true" or "false").

Span programs were introduced to the study of quantum query complexity in [28]. In the context of quantum query complexity, $s(P)$ is no longer the relevant measure of the complexity of a span program. Instead, [28] introduce the following measures:

Definition 3.4 (Span Program Complexity and Witnesses)**.** For a span program $P = (H, V, |\tau\rangle, A)$ on $\{0,1\}^n$ and input $x \in \{0,1\}^n$, we say x is *accepted* by the span program if there exists $|w\rangle \in H(x)$ such that $A|w\rangle = |\tau\rangle$, and otherwise we say x is *rejected* by the span program. Let P_0 and P_1 be respectively the set of rejected and accepted inputs to P. For $x \in P_1$, define the *positive witness complexity of* x as:

$$w_+(x, P) = w_+(x) = \min\{\||w\rangle\|^2 : |w\rangle \in H(x), A|w\rangle = |\tau\rangle\}.$$

[6] We remark that while H_{true} and H_{false} may be convenient in constructing a span program, they are not necessary. We can always consider a partial function f' defined on $(n+1)$-bit strings of the form $(x,1)$ for x in the domain of f, as $f(x)$, and let $H_{n+1,1} = H_{\text{true}}$ and $H_{n+1,0} = H_{\text{false}}$.

[7] Although V has no meaningful inner product, we use Dirac notation, such as $|\tau\rangle$ and $\langle\omega|$ for the sake of our fellow quantum computing researchers.

Such a $|w\rangle$ is called a *positive witness for* x. For a domain $D \subseteq \{0,1\}^n$, we define the *positive complexity of* P (with respect to D) as:

$$W_+(P,D) = W_+ = \max_{x \in P_1 \cap D} w_+(x,P).$$

For $x \in P_0$, define the *negative witness complexity of* x as:

$$w_-(x,P) = w_-(x) = \min\{\|\langle\omega|A\|^2 : \langle\omega| \in \mathcal{L}(V,\mathbb{R}), \langle\omega|\tau\rangle = 1, \langle\omega|A\Pi_{H(x)} = 0\}.$$

Above, $\mathcal{L}(V,\mathbb{R})$ denotes the set of linear functions from V to $\mathbb{R}$. Such an $\langle\omega|$ is called a *negative witness for* x. We define the *negative complexity of* P (with respect to D) as:

$$W_-(P,D) = W_- = \max_{x \in P_0 \cap D} w_-(x,P).$$

Finally, we define the *complexity of* P (with respect to D) by $C(P,D) = \sqrt{W_+W_-}$.

For $f : D \to \{0,1\}$, we say a span program P *decides* f if $f^{-1}(0) \subseteq P_0$ and $f^{-1}(1) \subseteq P_1$.

Definition 3.5. We define the *span program size* of a function f, denoted $\mathsf{SP}(f)$, as the minimum $s(P)$ over families of span programs that decide f.

We note that originally, in [20], span program size was defined

$$s'(P) = \sum_{j,b} \dim(\mathrm{col}(A\Pi_{H_{j,b}})) = \sum_{j,b} \dim(\mathrm{row}(A\Pi_{H_{j,b}})).$$

This could differ from $s(P) = \dim(H) = \sum_{j,b} \dim(H_{j,b})$, because $\dim(H_{j,b})$ might be much larger than $\dim(\mathrm{row}(A\Pi_{H_{j,b}}))$. However, if a span program has $\dim(H_{j,b}) > \dim(\mathrm{row}(A\Pi_{H_{j,b}}))$ for some j,b, then it is a simple exercise to show that the dimension of $\dim(H_{j,b})$ can be reduced without altering the witness size of any $x \in \{0,1\}^n$, so the definition of $\mathsf{SP}(f)$ is the same as if we had used $s'(P)$ instead of $s(P)$. In any case, we will not be relying on previous results about the span program size as a black-box, and will rather prove all required statements, so this difference has no impact on our results.

While span program size has only previously been relevant outside the realm of quantum algorithms, the complexity of a span program deciding f has a fundamental correspondence with the quantum query complexity of f. Specifically, a span program P can be turned into a quantum algorithm for f with query complexity $C(P,D)$, and moreover, for every f, there exists a span program such that the algorithm constructed in this way is optimal [27]. This second direction is not constructive: there is no known method for converting a quantum algorithm with query complexity T to a span program with complexity $C(P,D) = \Theta(T)$. However, if we relax the definition of which functions are decided by a span program, then such a construction is possible, as we will show in Section 3.3. The following is a slight relaxation of [15, Definition 2.6].[8]

[8]Which was already a relaxation of the notion of a span program deciding a function.

Definition 3.6 (A Span Program that Approximately Decides a Function)**.** Let $f : D \to \{0,1\}$ for $D \subseteq \{0,1\}^n$ and $\kappa \in (0,1)$. We say that a span program P on $\{0,1\}^n$ *κ-approximates* f if $f^{-1}(0) \subseteq P_0$, and for every $x \in f^{-1}(1)$, there exists an *approximate positive witness* $|\hat{w}\rangle$ such that $A|\hat{w}\rangle = |\tau\rangle$, and $\left\|\Pi_{H(x)^\perp}|\hat{w}\rangle\right\|^2 \leq \frac{\kappa}{W_-}$. We define the approximate positive complexity as

$$\widehat{W}_+ = \widehat{W}_+^\kappa(P,D) = \max_{x\in f^{-1}(1)} \min\left\{\||\hat{w}\rangle\|^2 : A|\hat{w}\rangle = |\tau\rangle, \left\|\Pi_{H(x)^\perp}|\hat{w}\rangle\right\|^2 \leq \frac{\kappa}{W_-}\right\}.$$

If P κ-approximates f, we define the complexity of P (wrt. D and κ) as $C_\kappa(P,D) = \sqrt{\widehat{W}_+ W_-}$.

If $\kappa = 0$, the span program in Definition 3.6 *decides* f (exactly), and $\widehat{W}_+ = W_+$. By [15, Theorem 2.10], for any x,

$$\min\left\{\left\|\Pi_{H(x)^\perp}|\hat{w}\rangle\right\|^2 : A|\hat{w}\rangle = |\tau\rangle\right\} = \frac{1}{w_-(x)}.$$

Thus, since $W_- = \max_{x\in f^{-1}(0)} w_-(x)$, for every $x \in f^{-1}(0)$, there does not exist an approximate positive witness with $\left\|\Pi_{H(x)^\perp}|\hat{w}\rangle\right\|^2 < \frac{1}{W_-}$. Thus, when a span program κ-approximates f, there is a gap of size $\frac{1-\kappa}{W_-}$ between the smallest positive witness error $\left\|\Pi_{H(x)^\perp}|\hat{w}\rangle\right\|^2$ of $x \in f^{-1}(1)$, the smallest positive witness error of $x \in f^{-1}(0)$.

Definition 3.7. We define the *κ-approximate span program size* of a function f, denoted $\widetilde{\mathsf{SP}}_\kappa(f)$, as the minimum $s(P)$ over families of span programs that κ-approximate f. We let $\widetilde{\mathsf{SP}}(f) = \widetilde{\mathsf{SP}}_{1/4}(f)$.

We note that the choice of $\kappa = 1/4$ in $\widetilde{\mathsf{SP}}(f)$ is arbitrary, as it is possible to modify a span program to reduce any constant κ to any other constant without changing the complexity or the logarithm of the size asymptotically. This convenient observation is formalized in the following claim.

Claim 3.8. *Let P be a span program that κ-approximates $f : D \to \{0,1\}$ for some constant κ. For any constant $\kappa' \leq \kappa$, there exists a span program P' that κ'-approximates f with*

$$s(P') = (s(P)+2)^{2^{\frac{\log(1/\kappa')}{\log(1/\kappa)}}}, \tag{3.1}$$

and $C_{\kappa'}(P',D) \leq O(C_\kappa(P,D))$.

We prove Claim 3.8 shortly in Section 3.1.1. We have the following corollary that will be useful later, where $\mathsf{m}\widetilde{\mathsf{SP}}_\kappa$ is the *monotone approximate span program size*, defined in Definition 5.1:

Corollary 3.9. *For any $\kappa, \kappa' \in (0,1)$ with $\kappa' < \kappa$, and any Boolean function f,*

$$\widetilde{\mathsf{SP}}_\kappa(f) \geq \widetilde{\mathsf{SP}}_{\kappa'}(f)^{\frac{1}{2}\frac{\log(1/\kappa)}{\log(1/\kappa')}} - 2.$$

If f is monotone, we also have

$$\mathsf{m}\widetilde{\mathsf{SP}}_\kappa(f) \geq \mathsf{m}\widetilde{\mathsf{SP}}_{\kappa'}(f)^{\frac{1}{2}\frac{\log(1/\kappa)}{\log(1/\kappa')}} - 2.$$

Remark 3.10. It can sometimes be useful to construct a span program over $\mathbb{C}$. However, for any span program over $\mathbb{C}$, P, there is a span program over $\mathbb{R}$, P', such that for all $x \in P_0$, $w_-(x,P') \leq w_-(x,P)$, for all $x \in P_1$, $w_+(x,P') \leq w_+(x,P)$, and $s(P') \leq 2s(P)$. We define P' as follows. Without loss of generality, suppose $H_{j,b} = \mathrm{span}_{\mathbb{C}}\{|j,b,k\rangle : k \in S_{j,b}\}$. Define $H'_{j,b} = \mathrm{span}_{\mathbb{R}}\{|j,b,k,a\rangle : k \in S_{j,b}, a \in \{0,1\}\}$. Define

$$A'|j,b,k,0\rangle = \mathrm{Re}\,(A|j,b,k\rangle)\,|0\rangle + \mathrm{Im}\,(A|j,b,k\rangle)\,|1\rangle$$

$$A'|j,b,k,1\rangle = \mathrm{Re}\,(A|j,b,k\rangle)\,|1\rangle - \mathrm{Im}\,(A|j,b,k\rangle)\,|0\rangle.$$

Finally, let $|\tau'\rangle = |\tau\rangle|0\rangle$.

Suppose $|w\rangle$ is a witness in P. Then

$$\begin{aligned}|\tau\rangle &= A|w\rangle = A\mathrm{Re}(|w\rangle) + iA\mathrm{Im}(|w\rangle)\\ &= \mathrm{Re}(A\mathrm{Re}(|w\rangle)) + i\mathrm{Im}(A\mathrm{Re}(|w\rangle)) + i\mathrm{Re}(A\mathrm{Im}(|w\rangle)) - \mathrm{Im}(A\mathrm{Im}(|w\rangle)).\end{aligned}$$

Since we can assume $|\tau\rangle$ is real, we have

$$|\tau\rangle = \mathrm{Re}(A\mathrm{Re}(|w\rangle)) - \mathrm{Im}(A\mathrm{Im}(|w\rangle)) \quad \textit{and} \quad \mathrm{Im}(A\mathrm{Re}(|w\rangle)) + \mathrm{Re}(A\mathrm{Im}(|w\rangle)) = 0.$$

Define $|w'\rangle = \mathrm{Re}(|w\rangle)|0\rangle + \mathrm{Im}(|w\rangle)|1\rangle$. Then

$$A'|w'\rangle = \mathrm{Re}(A\mathrm{Re}(|w\rangle))|0\rangle + \mathrm{Im}(A\mathrm{Re}(|w\rangle))|1\rangle + \mathrm{Re}(A\mathrm{Im}(|w\rangle))|1\rangle - \mathrm{Im}(A\mathrm{Im}(|w\rangle))|0\rangle = |\tau\rangle|0\rangle = |\tau'\rangle.$$

Note that we have $\||w\rangle\| = \||w'\rangle\|$. A similar argument holds for negative witnesses.

Thus, we will restrict our attention to real span programs, but still allow constructions of span programs over $\mathbb{C}$ (in particular, in Section 3.3 and Section 5.2.1).

3.1.1 Proof of Claim 3.8

In this section, we prove Claim 3.8. The proof is somewhat technical, and may be skipped without compromising the reader's understanding of the remainder of the paper. We restate Claim 3.8 below.

Claim 3.8. Let P be a span program that κ-approximates $f : D \to \{0,1\}$ for some constant κ. For any constant $\kappa' \leq \kappa$, there exists a span program P' that κ'-approximates f with $s(P') = (s(P)+2)^{2^{\frac{\log(1/\kappa')}{\log(1/\kappa)}}}$, and $C_{\kappa'}(P',D) \leq O\left(C_\kappa(P,D)\right)$.

Let $|w_0\rangle = A^+|\tau\rangle$. We say a span program is *normalized* if $\||w_0\rangle\| = 1$. A span program can easily be normalized by scaling $|\tau\rangle$, which also scales all positive witnesses and inverse scales all negative witnesses. However, we sometimes want to normalize a span program, while also keeping all negative witness sizes bounded by a constant. We can accomplish this using the following construction, from [15].

Theorem 3.11. *Let $P = (H, V, |\tau\rangle, A)$ be a span program on $\{0,1\}^n$, and let $N = \||w_0\rangle\|^2$. For a positive real number β, define a span program $P^\beta = (H^\beta, V^\beta, |\tau^\beta\rangle, A^\beta)$ as follows, where $|\hat{0}\rangle$ and $|\hat{1}\rangle$ are not in H or V:*

$$H^\beta_{j,b} = H_{j,b}, \quad H^\beta_{\text{true}} = H_{\text{true}} \oplus \text{span}\{|\hat{1}\rangle\}, \quad H^\beta_{\text{false}} = H_{\text{false}} \oplus \text{span}\{|\hat{0}\rangle\}$$

$$V^\beta = V \oplus \text{span}\{|\hat{1}\rangle\}, \quad A^\beta = \beta A + |\tau\rangle\langle\hat{0}| + \frac{\sqrt{\beta^2+N}}{\beta}|\hat{1}\rangle\langle\hat{1}|, \quad |\tau^\beta\rangle = |\tau\rangle + |\hat{1}\rangle.$$

Then we have the following:

- $\left\|(A^\beta)^+|\tau^\beta\rangle\right\| = 1$;
- *for all $x \in P_1$, $w_+(x, P^\beta) = \frac{1}{\beta^2}w_+(x,P) + 2$;*
- *for all $x \in P_0$, $w_-(x, P^\beta) = \beta^2 w_-(x,P) + 1$.*

Corollary 3.12. *Let P be a span program on $\{0,1\}^n$, and P^β be defined as above for $\beta = \frac{1}{\sqrt{W_-(P)}}$. If P κ-approximates f, then P^β $\sqrt{\kappa}$-approximates f, with $W_-(P^\beta) \leq 2$, $\widehat{W}_+(P^\beta) \leq W_-(P)\widehat{W}_+(P) + 2$ and $s(P^\beta) \leq s(P) + 2$.*

Proof. First note that by Theorem 3.11, $W_-(P^\beta) \leq 2$. Let $|w\rangle$ be an approximate positive witness for x in P, with $\left\|\Pi_{H(x)^\perp}|w\rangle\right\|^2 \leq \frac{\kappa}{W_-(P)}$ and $\||w\rangle\|^2 \leq \widehat{W}_+(P)$. Define

$$|w'\rangle = \frac{1}{\beta(1+\kappa)}|w\rangle + \frac{\beta}{\sqrt{\beta^2+N}}|\hat{1}\rangle + \frac{\kappa}{1+\kappa}|\hat{0}\rangle.$$

One can check that $A^\beta|w'\rangle = |\tau^\beta\rangle$.

$$\begin{aligned}\left\|\Pi_{H^\beta(x)^\perp}|w'\rangle\right\|^2 &= \frac{1}{\beta^2(1+\kappa)^2}\left\|\Pi_{H(x)^\perp}|w\rangle\right\|^2 + \frac{\kappa^2}{(1+\kappa)^2} \leq \frac{1}{\beta^2(1+\kappa)^2}\frac{\kappa}{W_-(P)} + \frac{\kappa^2}{(1+\kappa)^2}\\ &= \frac{\kappa+\kappa^2}{(1+\kappa)^2} \leq \frac{2\kappa(1+\kappa)}{W_-(P^\beta)(1+\kappa)^2} = \frac{1}{W_-(P^\beta)}\frac{2\kappa}{1+\kappa} \leq \frac{\sqrt{\kappa}}{W_-(P^\beta)},\end{aligned}$$

where we have used $W_-(P^\beta) \leq 2$. We upper bound $\widehat{W}_+(P^\beta)$ by noting that:

$$\begin{aligned}\||w'\rangle\|^2 &\leq \frac{1}{\beta^2(1+\kappa)^2}\widehat{W}_+(P) + \frac{\beta^2}{\beta^2+N} + \frac{\kappa^2}{(1+\kappa)^2}\\ &\leq W_-(P)\widehat{W}_+(P) + 2.\end{aligned}$$

Finally, $s(P^\beta) = s(P) + 2$ because of the two extra degrees of freedom $|\hat{0}\rangle$ and $|\hat{1}\rangle$. □

Proof of Claim 3.8. We will first show how, given a span program P such that $\||w_0\rangle\|^2 \leq 1$, and P κ-approximates f, we can get a span program P' such that $\left\||w'_0\rangle\right\|^2 \leq 1$, $W_-(P') \leq W_-(P)^2$, P' κ^2-approximates f, $\widehat{W}_+(P') \leq 4\widehat{W}_+(P)$, and $s(P') = s(P)^2$.

Define P' as follows, where S is a *swap* operator, which acts as $S(|u\rangle|v\rangle) = |v\rangle|u\rangle$ for all $|u\rangle, |v\rangle \in H$:

$$H'_{j,b} = H_{j,b} \otimes H, \qquad A' = (A \otimes A)\left(\frac{I_{H\otimes H} + S}{2}\right), \qquad |\tau'\rangle = |\tau\rangle|\tau\rangle.$$

Observe that for any $|u\rangle, |v\rangle \in H$, we have

$$A'(|u\rangle|v\rangle - |v\rangle|u\rangle) = 0, \quad \text{and} \quad A'|u\rangle|u\rangle = A|u\rangle \otimes A|u\rangle.$$

Note that $A'(|w_0\rangle|w_0\rangle) = |\tau'\rangle$, so $\left\|A'^{+}|\tau'\rangle\right\| \leq \||w_0\rangle|w_0\rangle\| \leq 1$.

If $\langle\omega|$ is a negative witness for x in P, it is easily verified that $\langle\omega'| = \langle\omega| \otimes \langle\omega|$ is a negative witness in P', and

$$\|\langle\omega'|A'\|^2 = \left\|\frac{1}{2}(\langle\omega|A) \otimes (\langle\omega|A) + \frac{1}{2}(\langle\omega|A) \otimes (\langle\omega|A)\right\|^2 = \|\langle\omega|A\|^4,$$

so $w_-(x, P') \leq w_-(x, P)^2$, and $W_-(P') \leq W_-(P)^2$.

If $|w\rangle$ is an approximate positive witness for x in P, then define

$$|w'\rangle = |w\rangle|w\rangle - \Pi_{H(x)^\perp}|w\rangle\Pi_{H(x)}|w\rangle + \Pi_{H(x)}|w\rangle\Pi_{H(x)^\perp}|w\rangle - \Pi_{H(x)}|w\rangle\Pi_{\ker(A)}|w\rangle.$$

We have

$$A'|w'\rangle = A|w\rangle A|w\rangle - \frac{1}{2}\left(A\Pi_{H(x)}|w\rangle \otimes A\Pi_{\ker(A)}|w\rangle + A\Pi_{\ker(A)}|w\rangle \otimes A\Pi_{H(x)}|w\rangle\right) = |\tau\rangle|\tau\rangle = |\tau'\rangle.$$

We can bound the error as:

$$\begin{aligned}\left\|\Pi_{H'(x)^\perp}|w'\rangle\right\|^2 &= \left\|(\Pi_{H(x)^\perp} \otimes I)|w'\rangle\right\|^2 = \left\|\Pi_{H(x)^\perp}|w\rangle|w\rangle - \Pi_{H(x)^\perp}|w\rangle\Pi_{H(x)}|w\rangle\right\|^2 \\ &= \left\|\Pi_{H(x)^\perp}|w\rangle\Pi_{H(x)^\perp}|w\rangle\right\|^2 \leq \frac{\kappa^2}{W_-(P)^2} \leq \frac{\kappa^2}{W_-(P')}.\end{aligned}$$

Next, observe that

$$\begin{aligned}&(\Pi_{H(x)} + \Pi_{H(x)^\perp}) \otimes (\Pi_{H(x)} + \Pi_{H(x)^\perp}) - \Pi_{H(x)^\perp} \otimes \Pi_{H(x)} + \Pi_{H(x)} \otimes \Pi_{H(x)^\perp} \\ &= \Pi_{H(x)} \otimes \Pi_{H(x)} + \Pi_{H(x)} \otimes \Pi_{H(x)^\perp} + \Pi_{H(x)^\perp} \otimes \Pi_{H(x)^\perp} + \Pi_{H(x)} \otimes \Pi_{H(x)^\perp} \\ &= \Pi_{H(x)} \otimes I + I \otimes \Pi_{H(x)^\perp}\end{aligned}$$

so $|w'\rangle = \Pi_{H(x)}|w\rangle \otimes |w\rangle + |w\rangle \otimes \Pi_{H(x)^\perp}|w\rangle - \Pi_{H(x)}|w\rangle \otimes \Pi_{\ker(A)}|w\rangle.$

Thus, using the assumption $\||w_0\rangle\| \leq 1$, and the fact that $\Pi_{\mathrm{row}(A)}|w\rangle = |w_0\rangle$:

$$\begin{aligned}
\||w'\rangle\|^2 &= \left\|\Pi_{H(x)}|w\rangle|w\rangle + |w\rangle\Pi_{H(x)^\perp}|w\rangle - \Pi_{H(x)}|w\rangle\Pi_{\ker(A)}|w\rangle\right\|^2 \\
&= \left\|\Pi_{H(x)}|w\rangle\Pi_{\mathrm{row}(A)}|w\rangle + |w\rangle\Pi_{H(x)^\perp}|w\rangle\right\|^2 \\
&= \left\|\Pi_{H(x)}|w\rangle|w_0\rangle\right\|^2 + \left\||w\rangle\Pi_{H(x)^\perp}|w\rangle\right\|^2 + 2\left\|\Pi_{H(x)}|w\rangle\right\|^2 \langle w_0|\Pi_{H(x)^\perp}|w\rangle \\
&\leq \widehat{W}_+(P) + \widehat{W}_+(P)\frac{\kappa}{W_-(P)} + 2\widehat{W}_+(P)\sqrt{\frac{\kappa}{W_-(P)}} \leq (1+\kappa+2\sqrt{\kappa})\widehat{W}_+(P).
\end{aligned}$$

Note that we could assume that $\widehat{W}_-(P) \geq 1$ because $\|w_0\| \leq 1$.

We complete the proof by extending to the general case. Let P be any span program that κ-approximates f. By applying Theorem 3.11 and Corollary 3.12, we can get a span program, P_0, with $\||w_0\rangle\| = 1$, $W_-(P_0) \leq 2$, $\widehat{W}_+(P_0) \leq C(P)^2 + 2$, and $s(P_0) = s(P) + 2$, that $\sqrt{\kappa}$-approximates f. We can then apply the construction described above, iteratively, d times, to get a span program P_d that $\sqrt{\kappa}^{2^d} = \kappa^{2^{d-1}}$-approximates f, with

$$s(P_d) = s(P_0)^{2^d} = (s(P)+2)^{2^d},$$

$$W_-(P_d) \leq 2^{2^d}, \qquad \text{and} \qquad \widehat{W}_+(P_d) \leq 4^d\widehat{W}_+(P_0) \leq 4^d C(P)^2 + 2\cdot 4^d.$$

Setting $d = \log\left(\frac{\log(1/\kappa')}{\log(1/\kappa)}\right) + 1$ gives the desired κ'. □

3.2 From span programs to quantum algorithms

In this section, we will prove Theorem 3.1, which states that if a span program approximately decides a function f, then we can compile it to a quantum algorithm for f. While we hope that Theorem 3.1 will have applications in designing span program algorithms, its only relevance to the contents of this paper are its implications with respect to the tightness of the first lower bound expression in Theorem 4.1, and so this section can be safely skipped.

Theorem 3.1 is similar to [15, Lemma 3.6], the difference here is we let an approximate positive witness for x be any witness with error, $\left\|\Pi_{H(x)^\perp}|w\rangle\right\|^2$, at most κ/W_-, whereas in [15], it is required to have error as small as possible. This relaxation could potentially decrease the positive complexity $\widehat{W}_+$, since we now have more freedom in selecting positive witnesses, but more importantly, it makes it easier to analyze a span program, because we need not find the approximate positive witness with the smallest possible error. Importantly, this change in how we define a span program that approximates f does not change the most important property of such a span program: that it can be compiled into a quantum algorithm for f. To show this, we now modify the proof of [15, Lemma 3.6] to fit the new definition. We will restrict to span programs on binary strings $\{0,1\}^n$, but the proof also works for span programs on $[q]^n$ for $q > 2$.

Proof of Theorem 3.1. For a span program P on $\{0,1\}^n$ and $x \in \{0,1\}^n$, define

$$U(P,x) = (2\Pi_{\ker(A)} - I)(2\Pi_{H(x)} - I),$$

which acts on H. To prove Theorem 3.1, we will show that by performing phase estimation of $U(P,x)$ on initial state $|w_0\rangle = A^+|\tau\rangle$, and estimating the amplitude on having $|0\rangle$ in the phase register, we can distinguish 1- and 0-inputs of f with bounded error.

By Corollary 3.12 and Claim 3.8, we can assume without loss of generality that P has been scaled so that it κ-approximates f for some $\kappa < 1/4$, $|w_0\rangle = A^+|\tau\rangle$ is a unit vector, and $W_- \leq 2$. The scaled span program still has size $K^{O(1)}$ and complexity $O(C)$.

We first modify the proof of [15, Lemma 3.2] to get the following lemma:

Lemma 3.13. *Let P be a span program that κ-approximates f, with $\||w_0\rangle\|^2 = 1$. Fix any $\Theta \in (0,\pi)$, and let Π_Θ be the projector onto the $e^{i\theta}$-eigenspaces of $U(P,x)$ with $|\theta| \leq \Theta$. For any $x \in f^{-1}(1)$,*

$$\|\Pi_\Theta|w_0\rangle\|^2 \leq \Theta^2\widehat{W}_+ + \frac{4\kappa}{W_-}.$$

Proof. Suppose $x \in f^{-1}(1)$ and let $|\hat{w}_x\rangle$ be an approximate positive witness with $\left\|\Pi_{H(x)^\perp}|\hat{w}_x\rangle\right\|^2 \leq \frac{\kappa}{W_-}$ and $\||\hat{w}_x\rangle\|^2 \leq \widehat{W}_+$. Note that since $A|\hat{w}_x\rangle = |\tau\rangle$, $\Pi_{\mathrm{row}(A)}|\hat{w}_x\rangle = A^+A|\hat{w}_x\rangle = A^+|\tau\rangle = |w_0\rangle$, so

$$\Pi_{\mathrm{row}(A)}\Pi_{H(x)}|\hat{w}_x\rangle + \Pi_{\mathrm{row}(A)}\Pi_{H(x)^\perp}|\hat{w}_x\rangle = |w_0\rangle.$$

Since $\Pi_{H(x)^\perp}\Pi_{H(x)}|\hat{w}_x\rangle = 0$, we have, by the effective spectral gap lemma (Lemma 2.1):

$$\begin{aligned}
\left\|\Pi_\Theta\Pi_{\mathrm{row}(A)}\Pi_{H(x)}|\hat{w}_x\rangle\right\|^2 &\leq \frac{\Theta^2}{4}\left\|\Pi_{H(x)}|\hat{w}_x\rangle\right\|^2\\
\left\|\Pi_\Theta\left(|w_0\rangle - \Pi_{\mathrm{row}(A)}\Pi_{H(x)^\perp}|\hat{w}_x\rangle\right)\right\|^2 &\leq \frac{\Theta^2}{4}\||\hat{w}_x\rangle\|^2\\
\|\Pi_\Theta|w_0\rangle\|^2 + \left\|\Pi_\Theta\Pi_{\mathrm{row}(A)}\Pi_{H(x)^\perp}|\hat{w}_x\rangle\right\|^2 - 2\langle w_0|\Pi_\Theta\Pi_{\mathrm{row}(A)}\Pi_{H(x)^\perp}|\hat{w}_x\rangle &\leq \frac{\Theta^2}{4}\widehat{W}_+\\
\|\Pi_\Theta|w_0\rangle\|^2 - 2\,\|\Pi_\Theta|w_0\rangle\|\,\left\|\Pi_{H(x)^\perp}|\hat{w}_x\rangle\right\| &\leq \frac{\Theta^2}{4}\widehat{W}_+\\
\|\Pi_\Theta|w_0\rangle\|^2 - 2\,\|\Pi_\Theta|w_0\rangle\|\sqrt{\frac{\kappa}{W_-}} &\leq \frac{\Theta^2}{4}\widehat{W}_+.
\end{aligned}$$

This is satisfied only when

$$\begin{aligned}
\|\Pi_\Theta|w_0\rangle\| &\leq \sqrt{\frac{\kappa}{W_-}} + \sqrt{\frac{\kappa}{W_-} + \frac{\Theta^2}{4}\widehat{W}_+} \leq 2\sqrt{\frac{\Theta^2}{4}\widehat{W}_+ + \frac{\kappa}{W_-}}\\
\|\Pi_\Theta|w_0\rangle\|^2 &\leq \Theta^2\widehat{W}_+ + \frac{4\kappa}{W_-}.
\end{aligned}$$

□

We will let $\Theta^2 = \frac{1-4\kappa}{2\widehat{W}_+W_-}$. Then when $f(x) = 0$, we have

$$\|\Pi_0|w_0\rangle\|^2 = \frac{1}{w_-(x)} \geq \frac{1}{W_-} =: q_0,$$

by [15, Lemma 3.3]. On the other hand, when $f(x) = 1$, we have

$$\|\Pi_\Theta|w_0\rangle\|^2 \leq \Theta^2\widehat{W}_+ + 4\frac{\kappa}{W_-} = \frac{1-4\kappa}{2W_-} + \frac{4\kappa}{W_-} = \frac{1+4\kappa}{2W_-} =: q_1.$$

We want to distinguish these two cases using $1/\Theta$ steps of phase estimation, and then estimating the amplitude on having an estimate of 0 in the phase register to precision:

$$\Delta = \frac{q_0 - q_1}{2} = \frac{1-4\kappa}{4W_-}.$$

This will allow us to distinguish between amplitude $\geq q_0$ and amplitude $\leq q_1$. Since $\kappa < \frac{1}{4}$ is a constant, $\Delta = \Omega(1/W_-)$, and thus we use $O(1/\Delta) = O(W_-) = O(1)$ (recall that we are assuming the span program has been scaled) calls to phase estimation, each of which requires $O(1/\Theta) = O\left(\sqrt{\widehat{W}_+W_-}\right) = O(C)$ controlled calls to U (for more details, see the nearly identical proof of [15, Lemma 3.2]). Since $U(P,x)$ can be implemented in cost one query, the query complexity of this algorithm is $O(C)$.

The algorithm needs a single register of dimension $\dim H = K^{O(1)}$ to apply $U(P,x)$, $O(1)$ registers of dimension $1/\Theta$ to act as phase registers in phase estimation, and $O(1)$ registers of dimension $O(1/\Delta)$ to act as phase registers in the amplitude estimation, for a total space requirement of

$$\log\dim H + O\left(\log\frac{1}{\Delta}\right) + O\left(\log\frac{1}{\Theta}\right) = O(\log K) + O(\log C).$$

To complete the proof, we note that the algorithm is unitary, since it consists of phase estimation, composed unitarily with amplitude estimation. □

3.3 From quantum algorithms to span programs

In this section, we will show how to turn a unitary quantum algorithm into a span program, proving Theorem 3.2, which implies Theorem 1.1. The construction we use to prove Theorem 3.2 is based on a construction of Reichardt for turning any one-sided error quantum algorithm into a span program whose complexity matches the algorithm's query complexity [27, arXiv version]. We observe that a similar construction also works for two-sided error algorithms,[9] but the resulting span program only approximately decides f.

The algorithm Fix a function $f : D \to \{0,1\}$ for $D \subseteq \{0,1\}^n$, and a unitary quantum algorithm $\mathcal{A}$ such that on input $x \in f^{-1}(0)$, $\Pr[\mathcal{A}(x) = 1] \leq \frac{1}{3}$, and on input $x \in f^{-1}(1)$, $\Pr[\mathcal{A}(x) = 1] \geq 1-\varepsilon$, for $\varepsilon \in \{0, \frac{1}{3}\}$, depending on whether we want to consider a one-sided error or a bounded error algorithm. Let $p_0(x) = \Pr[\mathcal{A}(x) = 0]$, so if $f(x) = 0$, $p_0(x) \geq 2/3$, and if $f(x) = 1$, $p_0(x) \leq \varepsilon$.

[9] A preliminary version of this result appeared in [16], but there was an error in the proof, which is fixed by our new definition of approximate span programs.

We can suppose $\mathcal{A}$ acts on three registers: a query register $\mathrm{span}\{|j\rangle : j \in [n] \cup \{0\}\}$; a workspace register $\mathrm{span}\{|z\rangle : z \in \mathcal{Z}\}$ for some finite set of symbols $\mathcal{Z}$ that contains 0; and an answer register $\mathrm{span}\{|a\rangle : a \in \{0,1\}\}$. The query operator O_x acts on the query register as $O_x|j\rangle = (-1)^{x_j}|j\rangle$ if $j \geq 1$, and $O_x|0\rangle = |0\rangle$. If $\mathcal{A}$ makes T queries, the final state of $\mathcal{A}$ is:

$$|\Psi_{2T+1}(x)\rangle = U_{2T+1}O_xU_{2T-1}\dots U_3O_xU_1|0,0,0\rangle$$

for some unitaries $U_{2T+1},\dots,U_1$. The output bit of the algorithm, $\mathcal{A}(x)$, is obtained by measuring the answer register of $|\Psi_{2T+1}(x)\rangle$. We have given the input-independent unitaries odd indicies so that we may refer to the t-th query as U_{2t}.

Let $|\Psi_0(x)\rangle = |\Psi_0\rangle = |0,0,0\rangle$ denote the starting state, and for $t \in \{1,\dots,2T+1\}$, let $|\Psi_t(x)\rangle = U_t\dots U_1|\Psi_0\rangle$ denote the state after t steps.

The span program We now define a span program $P_{\mathcal{A}}$ from $\mathcal{A}$. The space H will represent all three registers of the algorithm, with an additional time counter register, and an additional register to represent a query value b.

$$H = \mathrm{span}\{|t,b,j,z,a\rangle : t \in \{0,\dots,2T+1\}, b \in \{0,1\}, j \in [n] \cup \{0\}, z \in \mathcal{Z}, a \in \{0,1\}\}.$$

We define V and A as follows, where c is some constant to be chosen later:

$$V = \mathrm{span}\{|t,j,z,a\rangle : t \in \{0,\dots,2T+1\}, j \in [n] \cup \{0\}, z \in \mathcal{Z}, a \in \{0,1\}\}$$

$$A|t,b,j,z,a\rangle = \begin{cases} |t,j,z,a\rangle - |t+1\rangle U_{t+1}|j,z,a\rangle & \text{if } t \in \{0,\dots,2T\} \text{ is even} \\ |t,j,z,a\rangle - (-1)^b|t+1,j,z,a\rangle & \text{if } t \in \{0,\dots,2T\} \text{ is odd (i.e., } U_{t+1} = O_x) \\ |t,j,z,a\rangle & \text{if } t = 2T+1,\ a = 1, \text{ and } b = 0 \\ \sqrt{cT}|t,j,z,a\rangle & \text{if } t = 2T+1,\ a = 0, \text{ and } b = 0 \\ 0 & \text{if } t = 2T+1 \text{ and } b = 1. \end{cases}$$

For $t \leq 2T$, $A|t,b,j,z,a\rangle$ should be intuitively understood as applying U_{t+1} to $|j,z,a\rangle$, and incrementing the counter register from $|t\rangle$ to $|t+1\rangle$. When t is even, this correspondence is clear (in that case, the value of b is ignored). When t is odd, so $U_{t+1} = O_x$, then as long as $b = x_j$, $(-1)^b|t+1,j,z,a\rangle = |t+1\rangle U_{t+1}|j,z,a\rangle$. We thus define

$$H_{j,b} = \mathrm{span}\{|t,b,j,z,a\rangle : t \in \{0,\dots,2T\} \text{ is odd}, z \in \mathcal{Z}, a \in \{0,1\}\}.$$

For even t, applying U_{t+1} is independent of the input, so we make the corresponding states available to every input; along with states where the query register is set to $j = 0$, meaning O_x acts input-independently; and accepting states, whose answer register is set to 1 at time $2T+1$:

$$\begin{aligned} H_{\mathrm{true}} = {} & \mathrm{span}\{|t,b,j,z,a\rangle : t \in \{0,\dots,2T\} \text{ is even}, b \in \{0,1\}, j \in [n], z \in \mathcal{Z}, a \in \{0,1\}\} \\ & \oplus \mathrm{span}\{|t,b,0,z,a\rangle : t \in \{0,\dots,2T\}, b \in \{0,1\}, z \in \mathcal{Z}, a \in \{0,1\}\} \\ & \oplus \mathrm{span}\{|2T+1,b,j,z,1\rangle : b \in \{0,1\}, j \in [n] \cup \{0\}, z \in \mathcal{Z}\}. \end{aligned}$$

The remaining part of H will be assigned to H_{false}:

$$H_{\mathrm{false}} = \mathrm{span}\{|2T+1, b, j, z, 0\rangle : b \in \{0,1\}, j \in [n] \cup \{0\}, z \in \mathcal{Z}\}.$$

Note that in defining A, we have put a large factor of $\sqrt{cT}$ in front of $A|2T+1, 0, j, z, 0\rangle$, making the vectors in H_{false} very "cheap" to use. These vectors are never in $H(x)$, but will be used as the error part of approximate positive witnesses, and the $\sqrt{cT}$ ensures they only contribute relatively small error.

Finally, we define:

$$|\tau\rangle = |0,0,0,0\rangle = |0\rangle|\Psi_0\rangle.$$

Intuitively, we can construct $|\tau\rangle$, the initial state, using a final state that has 1 in the answer register, and using the transitions $|t, j, z, a\rangle - |t+1\rangle U_{t+1}|j, z, a\rangle$ to move from the final state to the initial state. In the following analysis, we make this idea precise.

Analysis of $P_{\mathcal{A}}$ We will first show that for every x there is an approximate positive witness with error depending on its probability of being rejected by $\mathcal{A}$, $p_0(x)$.

Lemma 3.14. *For any $x \in \{0,1\}^n$, there exists an approximate positive witness $|w\rangle$ for x in $P_{\mathcal{A}}$ such that:*

$$\||w\rangle\|^2 \leq 2T+2, \text{ and } \left\|\Pi_{H(x)^\perp}|w\rangle\right\|^2 \leq \frac{p_0(x)}{cT}.$$

In particular, if $f(x) = 1$,

$$\left\|\Pi_{H(x)^\perp}|w\rangle\right\|^2 \leq \frac{\varepsilon}{cT}.$$

Proof. Let Q_x be the linear isometry that acts as

$$Q_x|j, z, a\rangle = |x_j, j, z, a\rangle \qquad \forall j \in [n] \cup \{0\}, z \in \mathcal{Z}, a \in \{0,1\},$$

where we interpret x_0 as 0. Note that for all $|j, z, a\rangle$, and $t \in \{0, \ldots, 2T\}$, we have

$$A(|t\rangle Q_x|j, z, a\rangle) = |t, j, z, a\rangle - |t+1\rangle U_{t+1}|j, z, a\rangle.$$

Let $\Pi_a = \sum_{j \in [n] \cup \{0\}, z \in \mathcal{Z}} |j, z, a\rangle\langle j, z, a|$ be the orthogonal projector onto states of the algorithm with answer register set to a. We will construct a positive witness for x from the states of the algorithm on input x, as follows:

$$|w\rangle = \sum_{t=0}^{2T} |t\rangle Q_x|\Psi_t(x)\rangle + |2T+1\rangle|0\rangle\Pi_1|\Psi_{2T+1}(x)\rangle + \frac{1}{\sqrt{cT}}|2T+1\rangle|0\rangle\Pi_0|\Psi_{2T+1}(x)\rangle.$$

To see that this is a positive witness, we compute $A|w\rangle$, using the fact that $U_{t+1}|\Psi_t(x)\rangle = |\Psi_{t+1}(x)\rangle$.

$$\begin{aligned}
A|w\rangle &= \sum_{t=0}^{2T}(|t\rangle|\Psi_t(x)\rangle - |t+1\rangle U_{t+1}|\Psi_t(x)\rangle) + |2T+1\rangle\Pi_1|\Psi_{2T+1}(x)\rangle + |2T+1\rangle\Pi_0|\Psi_{2T+1}(x)\rangle\\
&= \sum_{t=0}^{2T}|t\rangle|\Psi_t(x)\rangle - \sum_{t=0}^{2T}|t+1\rangle|\Psi_{t+1}(x)\rangle + |2T+1\rangle|\Psi_{2T+1}(x)\rangle\\
&= \sum_{t=0}^{2T+1}|t\rangle|\Psi_t(x)\rangle - \sum_{t=1}^{2T+1}|t\rangle|\Psi_t(x)\rangle = |0\rangle|\Psi_0(x)\rangle = |\tau\rangle.
\end{aligned}$$

We next consider the error of $|w\rangle$ for x, given by $\left\|\Pi_{H(x)^\perp}|w\rangle\right\|^2$. Since $Q_x|j,z,a\rangle \in H(x)$ for all j,z,a, and $|2T+1,0\rangle\Pi_1|\Psi_{2T+1}(x)\rangle \in H_{\text{true}} \subset H(x)$, $\Pi_{H(x)^\perp}|w\rangle = \frac{1}{\sqrt{cT}}|2T+1\rangle|0\rangle\Pi_0|\Psi_{2T+1}(x)\rangle$, so

$$\left\|\Pi_{H(x)^\perp}|w\rangle\right\|^2 = \frac{1}{cT}\left\|\Pi_0|\Psi_{2T+1}(x)\rangle\right\|^2 = \frac{p_0(x)}{cT}.$$

Finally, we compute an upper bound on the positive witness complexity of $|w\rangle$.

$$\begin{aligned}
\left\||w\rangle\right\|^2 &= \sum_{t=0}^{2T}\left\|Q_x|\Psi_t(x)\rangle\right\|^2 + \left\|\Pi_1|\Psi_{2T+1}(x)\rangle\right\|^2 + \frac{1}{cT}\left\|\Pi_0|\Psi_{2T+1}(x)\rangle\right\|^2\\
&\leq \sum_{t=0}^{2T}\left\||\Psi_t(x)\rangle\right\|^2 + \left\||\Psi_{2T+1}(x)\rangle\right\|^2 = 2T+2. \qquad \square
\end{aligned}$$

Next, we compute an upper bound on $w_-(x)$ whenever $f(x)=0$.

Lemma 3.15. *For any x that is rejected by $\mathcal{A}$ with probability $p_0(x) > 0$,*

$$w_-(x) \leq \frac{(c+4)T}{p_0(x)}.$$

In particular, if $f(x)=0$, $w_-(x) \leq \frac{c+4}{2/3}T$, so $W_- \leq \frac{c+4}{2/3}T$.

Proof. We will define a negative witness for x as follows. First, define

$$|\Psi_{2T+1}^0(x)\rangle = \Pi_0|\Psi_{2T+1}(x)\rangle,$$

the rejecting part of the final state. This is non-zero whenever $p_0(x) > 0$. Then for $t \in \{0,\dots,2T\}$, define

$$|\Psi_t^0(x)\rangle = U_{t+1}^\dagger \dots U_{2T+1}^\dagger|\Psi_{2T+1}^0(x)\rangle.$$

From this we can define

$$\langle\omega| = \sum_{t=0}^{2T+1}\langle t|\langle\Psi_t^0(x)|.$$

We first observe that

$$\langle\omega|\tau\rangle = \langle\Psi_0^0(x)|0,0,0\rangle = \langle\Psi_{2T+1}^0(x)|U_{2T+1}\dots U_1|0,0,0\rangle = \langle\Psi_{2T+1}^0(x)|\Psi_{2T+1}(x)\rangle = p_0(x).$$

Thus

$$\langle\bar{\omega}| = \frac{1}{p_0(x)}\langle\omega|$$

is a negative witness. Next, we show that $\langle\omega|A\Pi_{H(x)} = 0$. First, for $|t,x_j,j,z,a\rangle \in H_{j,x_j}$ (so $t < 2T$ is odd), we have

$$\begin{aligned}
\langle\omega|A|t,x_j,j,z,a\rangle &= \langle\omega|(|t,j,z,a\rangle - (-1)^{x_j}|t+1\rangle|j,z,a\rangle) \\
&= \langle\Psi_t^0(x)|j,z,a\rangle - (-1)^{x_j}\langle\Psi_{t+1}^0(x)|j,z,a\rangle \\
&= \langle\Psi_{t+1}^0(x)|U_{t+1}|j,z,a\rangle - (-1)^{x_j}\langle\Psi_{t+1}^0(x)|j,z,a\rangle \\
&= \langle\Psi_{t+1}^0(x)|O_x|j,z,a\rangle - (-1)^{x_j}\langle\Psi_{t+1}^0(x)|j,z,a\rangle = 0.
\end{aligned}$$

The same argument holds for $|t,0,0,j,z,a\rangle \in H_{\mathrm{true}}$. Similarly, for any $|t,b,j,z,a\rangle \in H_{\mathrm{true}}$ with $t \leq 2T$ even, we have

$$\begin{aligned}
\langle\omega|A|t,b,j,z,a\rangle &= \langle\omega|(|t,j,z,a\rangle - |t+1\rangle U_{t+1}|j,z,a\rangle) \\
&= \langle\Psi_t^0(x)|j,z,a\rangle - \langle\Psi_{t+1}^0(x)|U_{t+1}|j,z,a\rangle = 0.
\end{aligned}$$

Finally, for any $|2T+1,b,j,z,1\rangle \in H_{\mathrm{true}}$, we have

$$\langle\omega|A|2T+1,b,j,z,1\rangle = \langle\omega|2T+1,j,z,1\rangle = \langle\Psi_{2T+1}^0(x)|j,z,1\rangle = 0.$$

Thus $\langle\omega|A\Pi_{H(x)} = 0$ and so $\langle\bar{\omega}|A\Pi_{H(x)} = 0$, and $\langle\bar{\omega}|$ is a negative witness for x in P. To compute its witness complexity, first observe that $\langle\omega|A = \langle\omega|A\Pi_{H(x)^\perp}$, and

$$\begin{aligned}
A\Pi_{H(x)^\perp} = &\sum_{s=1}^{T}\sum_{j\in[n]\cup\{0\},z\in\mathcal{Z},a\in\{0,1\}}(|2s-1,j,z,a\rangle + (-1)^{x_j}|2s,j,z,a\rangle)\langle 2s-1,\bar{x}_j,j,z,a| \\
&+ \sum_{j\in[n]\cup\{0\},z\in\mathcal{Z}}\sqrt{cT}|2T+1,j,z,0\rangle\langle 2T+1,0,j,z,0|
\end{aligned}$$

so, using $\langle\Psi_{2s-1}^0(x)|j,z,a\rangle = \langle\Psi_{2s}^0(x)|U_{2s}|j,z,a\rangle = (-1)^{x_j}\langle\Psi_{2s}^0(x)|j,z,a\rangle$, we have:

$$\begin{aligned}
\langle\omega|A\Pi_{H(x)^\perp} = &\sum_{s=1}^{T}\sum_{j\in[n]\cup\{0\},z\in\mathcal{Z},a\in\{0,1\}}(\langle\Psi_{2s-1}^0(x)|j,z,a\rangle + (-1)^{x_j}\langle\Psi_{2s}^0(x)|j,z,a\rangle)\langle 2s-1,\bar{x}_j,j,z,a| \\
&+ \sum_{j\in[n]\cup\{0\},z\in\mathcal{Z}}\sqrt{cT}\langle\Psi_{2T+1}^0(x)|j,z,0\rangle\langle 2T+1,0,j,z,0| \\
= &\sum_{s=1}^{T}\sum_{j\in[n]\cup\{0\},z\in\mathcal{Z},a\in\{0,1\}} 2(-1)^{x_j}\langle\Psi_{2s}^0(x)|j,z,a\rangle)\langle 2s-1,\bar{x}_j,j,z,a| \\
&+ \sum_{j\in[n]\cup\{0\},z\in\mathcal{Z}}\sqrt{cT}\langle\Psi_{2T+1}^0(x)|j,z,0\rangle\langle 2T+1,0,j,z,0|.
\end{aligned}$$

Thus, the complexity of $\langle\bar{\omega}|$ is:

$$\begin{aligned}\|\langle\bar{\omega}|A\|^2 &= \frac{1}{p_0(x)^2}\left\|\langle\omega|A\Pi_{H(x)^\perp}\right\|^2 \\ &= \frac{1}{p_0(x)^2}\sum_{s=1}^{T}\sum_{\substack{j\in[n]\cup\{0\},\\ z\in\mathcal{Z},\\ a\in\{0,1\}}} 4\left|\langle\Psi_{2s}^0(x)|j,z,a\rangle\right|^2 + \frac{1}{p_0(x)^2}\sum_{\substack{j\in[n]\cup\{0\},\\ z\in\mathcal{Z}}} cT\left|\langle\Psi_{2T+1}^0(x)|j,z,0\rangle\right|^2 \\ &= \frac{4}{p_0(x)^2}\sum_{s=1}^{T}\left\||\Psi_{2s}^0(x)\rangle\right\|^2 + \frac{cT}{p_0(x)^2}\left\||\Psi_{2T+1}^0(x)\rangle\right\|^2.\end{aligned}$$

Because each U_t is unitary, we have $\left\||\Psi_{2s}^0(x)\rangle\right\|^2 = \left\||\Psi_{2T+1}^0(x)\rangle\right\|^2 = p_0(x)$, thus:

$$\|\langle\bar{\omega}|A\|^2 = \frac{4T}{p_0(x)} + \frac{cT}{p_0(x)} \leq \frac{4+c}{2/3}T \text{ when } f(x)=0. \qquad \square$$

We conclude the proof of Theorem 3.2 with the following corollary, from which Theorem 3.2 follows immediately, by appealing to Claim 3.8 with $\kappa = \frac{9}{10}$ and κ' any constant in $(0,1)$.

Corollary 3.16. *Let $c = 5$, in the definition of $P_{\mathcal{A}}$. Then:*

- $s(P_{\mathcal{A}}) = 2^{S+O(1)}$
- *If $\mathcal{A}$ decides f with one-sided error, then $P_{\mathcal{A}}$ decides f with complexity $C \leq O(T)$.*
- *If $\mathcal{A}$ decides f with bounded error, then $P_{\mathcal{A}}$ $\frac{9}{10}$-approximates f with complexity $C_\kappa \leq O(T)$.*

Proof. We first compute $s(P_{\mathcal{A}}) = \dim H$ using the fact that the algorithm uses space

$$S = \log\dim\operatorname{span}\{|j,z,a\rangle : j\in[n]\cup\{0\}, z\in\mathcal{Z}, a\in\{0,1\}\} + \log T.$$

We have:

$$\dim H = (\dim\operatorname{span}\{|t,b\rangle : t\in\{0,\dots,2T+1\}, b\in\{0,1\}\})2^{S-\log T} = 2^{S+O(1)}.$$

We prove the third statement, as the second is similar. By Lemma 3.15, using $c=5$, we have

$$W_- \leq \frac{5+4}{2/3}T = \frac{27}{2}T.$$

By Lemma 3.14, we can see that for every x such that $f(x)=1$, there is an approximate positive witness $|w\rangle$ for x with error at most

$$\frac{\varepsilon}{cT} = \frac{1/3}{5T} \leq \frac{1}{15T}\frac{\frac{27}{2}T}{W_-} = \frac{9}{10}\frac{1}{W_-}.$$

Furthermore, $\||w\rangle\|^2 \leq 2T+2$, so $\widehat{W}_+ \leq 2T+2$. Observing $C_\kappa = \sqrt{W_-\widehat{W}_+} \leq \sqrt{27T(T+1)}$ completes the proof. $\square$

4 Span programs and space complexity

Using the transformation from algorithms to span programs from Section 3.3, we immediately have the following connections between span program size and space complexity.

Theorem 4.1. *For any $f : D \to \{0,1\}$ for $D \subseteq \{0,1\}^n$, we have*

$$\mathsf{S}_U(f) \geq \Omega\left(\log \widetilde{\mathsf{SP}}(f)\right) \qquad \textit{and} \qquad \mathsf{S}_U^1(f) \geq \Omega\left(\log \mathsf{SP}(f)\right).$$

Theorem 4.1 is a corollary of Theorem 3.2. Theorem 3.1 shows that the lower bound for $\mathsf{S}_U(f)$ in Theorem 4.1 is part of a *tight* correspondence between space complexity and $\log s(P) + \log C(P)$.

Theorem 2.9 of [5] gives a lower bound of $\mathsf{SP}(f) \geq \Omega(2^{n/3}/(n \log n)^{1/3})$ for almost all n-bit Boolean functions. Combined with Theorem 4.1, we immediately have:

Theorem 4.2. *For almost all Boolean functions $f : \{0,1\}^n \to \{0,1\}$, $\mathsf{S}_U^1(f) = \Omega(n)$.*

Ideally, we would like to use the lower bound in Theorem 4.1 to prove a non-trivial lower bound for $\mathsf{S}_U(f)$ or $\mathsf{S}_U^1(f)$ for some explicit function f. Fortunately, there are somewhat nice expressions lower bounding $\mathsf{SP}(f)$ [25, 11], which we extend to lower bounds of $\widetilde{\mathsf{SP}}(f)$ in the remainder of this section. However, on the unfortunate side, there has already been significant motivation to instantiate these expressions to non-trivial lower bounds for explicit f, with no success. There has been some success in *monotone* versions of these lower bounds, which we discuss more in Section 5.

For a function $f : D \to \{0,1\}$ for $D \subseteq \{0,1\}^n$, and an index $j \in [n]$, we let $\Delta_{f,j} \in \{0,1\}^{f^{-1}(0)\times f^{-1}(1)}$ be defined by $\Delta_{f,j}[y,x] = 1$ if and only if $x_j \neq y_j$. When f is clear from context, we simply denote this by Δ_j. The following tight characterization of $\mathsf{SP}(f)$ may be found in, for example, [23].

Lemma 4.3. *For any $f : D \to \{0,1\}$ for $D \subseteq \{0,1\}^n$,*

$$\begin{aligned} \mathsf{SP}(f) = \text{minimize} \quad & \sum_{j\in[n]} \mathrm{rank}(\Lambda_j) \\ \text{subject to} \quad & \forall j \in [n], \Lambda_j \in \mathbb{R}^{f^{-1}(0)\times f^{-1}(1)} \\ & \sum_{j\in[n]} \Lambda_j \circ \Delta_j = J, \end{aligned}$$

where J is the $f^{-1}(0) \times f^{-1}(1)$ all-ones matrix.

By Theorem 4.1, the logarithm of the above is a lower bound on $\mathsf{S}_U^1(f)$. We modify Lemma 4.3 to get the following approximate version, whose logarithm lower bounds $\mathsf{S}_U(f)$ when $\kappa = \frac{1}{4}$.

Lemma 4.4. *For any $\kappa \in [0,1)$, and $f : D \to \{0,1\}$ for $D \subseteq \{0,1\}^n$,*

$$\begin{aligned} \widetilde{\mathsf{SP}}_\kappa(f) \geq \text{minimize} \quad & \sum_{j\in[n]} \text{rank}(\Lambda_j) \\ \text{subject to} \quad & \forall j \in [n], \Lambda_j \in \mathbb{R}^{f^{-1}(0)\times f^{-1}(1)} \\ & \left\| \sum_{j\in[n]} \Lambda_j \circ \Delta_j - J \right\|_\infty \leq \sqrt{\kappa}. \end{aligned} \tag{4.1}$$

Proof. Fix a span program that κ-approximates f with $s(P) = \widetilde{\mathsf{SP}}_\kappa(f)$, and let $\{\langle\omega_y| : y \in f^{-1}(0)\}$ be optimal negative witnesses, and $\{|w_x\rangle : x \in f^{-1}(1)\}$ be approximate positive witnesses with $\left\|\Pi_{H(x)}|w_x\rangle\right\|^2 \leq \frac{\kappa}{W_-}$. Letting $\Pi_{j,b}$ denote the projector onto $H_{j,b}$, define

$$\Lambda_j = \left(\sum_y |y\rangle\langle\omega_y|A\Pi_{j,\bar{y}_j}\right)\left(\sum_x \Pi_{j,x_j}|w_x\rangle\langle x|\right),$$

so Λ_j has rank at most $\dim H_j$, and so $\sum_{j\in[n]}\text{rank}(\Lambda_j) \leq s(P) = \widetilde{\mathsf{SP}}_\kappa(f)$.

We now show that $\{\Lambda_j\}_j$ is a feasible solution. Let $|\mathsf{err}(x)\rangle$ be the positive witness error of $|w_x\rangle$, $|\mathsf{err}(x)\rangle = \Pi_{H(x)^\perp}|w_x\rangle = \sum_{j=1}^n \Pi_{j,\bar{x}_j}|w_x\rangle$. Then we have:

$$\begin{aligned} \langle y|\sum_{j=1}^n \Lambda_j \circ \Delta_j|x\rangle &= \langle\omega_y|A\sum_{j:x_j\neq y_j}\Pi_{j,x_j}|w_x\rangle = \langle\omega_y|A\left(|w_x\rangle - \sum_{j:x_j=y_j}\Pi_{j,x_j}|w_x\rangle - |\mathsf{err}(x)\rangle\right) \\ &= \langle\omega_y|\tau\rangle - \langle\omega_y|A\sum_{j:x_j=y_j}\Pi_{H(y)}\Pi_{j,x_j}|w_x\rangle - \langle\omega_y|A|\mathsf{err}(x)\rangle \\ &= 1 - 0 - \langle\omega_y|A|\mathsf{err}(x)\rangle \\ \left|1 - \langle y|\sum_{j=1}^n \Lambda_j\circ\Delta_j|x\rangle\right| &\leq \left\|\langle\omega_y|A\right\| \, \left\||\mathsf{err}(x)\rangle\right\| = \sqrt{w_-(y)\frac{\kappa}{W_-}} \leq \sqrt{\kappa}. \end{aligned}$$

Above we used the fact that $\langle\omega_y|A\Pi_{H(y)} = 0$. Thus, $\{\Lambda_j\}_j$ is a feasible solution with objective value $\leq \widetilde{\mathsf{SP}}_\kappa(f)$, so the result follows. ☐

As a corollary of the above, and the connection between span program size and unitary quantum space complexity stated in Theorem 4.1, the logarithm of the expression in (4.1) with $\kappa = \frac{1}{4}$ is a lower bound on $\mathsf{S}_U(f)$, and with $\kappa = 0$, it is a lower bound on $\mathsf{S}_U^1(f)$. However, as stated, it is difficult to use this expression to prove an explicit lower bound, because it is a minimization problem. We will shortly give a lower bound in terms of a maximization problem, making it possible to obtain explicit lower bounds by exhibiting a feasible solution.

A *partial matrix* is a matrix $M \in (\mathbb{R}\cup\{\star\})^{f^{-1}(0)\times f^{-1}(1)}$. A *completion* of M is any $\overline{M} \in \mathbb{R}^{f^{-1}(0)\times f^{-1}(1)}$ such that $\overline{M}[y,x] = M[y,x]$ whenever $M[y,x] \neq \star$. For a partial matrix M, define

$\mathrm{rank}(M)$ to be the smallest rank of any completion of M, and ε-$\mathrm{rank}(M)$ to be the smallest rank of any $\tilde{M}$ such that $|M[y,x]-\tilde{M}[y,x]|\leq\varepsilon$ for all y,x such that $M[y,x]\neq\star$. Let $M\circ\Delta_i$ to be the partial matrix defined:

$$M\circ\Delta_i[y,x]=\begin{cases} M[y,x] & \text{if } \Delta_i[y,x]=1\\ 0 & \text{if } \Delta_i[y,x]=0.\end{cases}$$

Then we have the following corollary of [11, Lemma 3.2, Theorem 3.4] and Theorem 4.1:

Lemma 4.5. *For all Boolean functions $f:D\to\{0,1\}$, with $D\subseteq\{0,1\}^n$, and all partial matrices $M\in(\mathbb{R}\cup\{\star\})^{f^{-1}(0)\times f^{-1}(1)}$ such that* $\max\{|M[y,x]|:M[y,x]\neq\star\}\leq 1$*:*

$$\mathsf{S}_U^1(f)\geq\Omega\left(\log\left(\frac{\mathrm{rank}(M)}{\max_{i\in[n]}\mathrm{rank}(M\circ\Delta_i)}\right)\right).$$

In [25], Razborov showed that the expression on the right-hand side in Lemma 4.5 is a lower bound on the logarithm of the *formula size* of f (Ref. [11] related this to $\mathsf{SP}(f)$). Later, in [26], Razborov noted that when restricted to non-partial matrices, this can never give a better bound than n. Thus, to prove a non-trivial lower bound on $\mathsf{S}_U^1(f)$ using this method, one would need to use a partial matrix. We prove the following generalization to the approximate case.

Lemma 4.6. *For all Boolean functions $f:D\to\{0,1\}$, with $D\subseteq\{0,1\}^n$, and all partial matrices $M\in(\mathbb{R}\cup\{\star\})^{f^{-1}(0)\times f^{-1}(1)}$ such that* $\max\{|M[y,x]|:M[y,x]\neq\star\}\leq 1$*:*

$$\mathsf{S}_U(f)\geq\Omega\left(\log\left(\frac{\frac{1}{2}\text{-}\mathrm{rank}(M)}{\max_{i\in[n]}\mathrm{rank}(M\circ\Delta_i)}\right)\right).$$

Proof. Let $\{\Lambda_j\}_j$ be an optimal feasible solution for the expression from Lemma 4.4, so

$$\widetilde{\mathsf{SP}}_\kappa(f)\geq\sum_{j\in[n]}\mathrm{rank}(\Lambda_j),\quad\text{and}\quad\left\|\sum_{j\in[n]}\Lambda_j\circ\Delta_j-J\right\|_\infty\leq\sqrt{\kappa}.$$

Let $\overline{M}_j$ be a completion of $M\circ\Delta_j$ with $\mathrm{rank}(M\circ\Delta_j)=\mathrm{rank}(\overline{M}_j)$. Then for any x,y such that $M[y,x]\neq\star$:

$$\begin{aligned}\left|\left(\sum_{j\in[n]}\overline{M}_j\circ\Lambda_j\right)[y,x]-M[y,x]\right| &= \left|\sum_{j\in[n]}M[y,x]\Delta_j[y,x]\Lambda_j[y,x]-M[y,x]\right|\\ &\leq |M[y,x]|\left\|\sum_{j\in[n]}\Delta_j\circ\Lambda_j-J\right\|_\infty\leq\sqrt{\kappa}.\end{aligned}$$

Thus

$$\sqrt{\kappa}\text{-rank}(M) \leq \text{rank}\left(\sum_{j\in[n]} \overline{M}_j \circ \Lambda_j\right) \leq \sum_{j\in[n]} \text{rank}(\overline{M}_j \circ \Lambda_j).$$

Using the fact that for any matrices B and C, $\text{rank}(B \circ C) \leq \text{rank}(B)\text{rank}(C)$, we have

$$\sqrt{\kappa}\text{-rank}(M) \leq \sum_{j\in[n]} \text{rank}(\Lambda_j)\text{rank}(\overline{M}_j) \leq \widetilde{\mathsf{SP}}_\kappa(f) \max_{j\in[n]} \text{rank}(M \circ \Delta_j).$$

Setting $\kappa = \frac{1}{4}$, and noting that by Theorem 4.1, $\mathsf{S}_U(f) \geq \log \widetilde{\mathsf{SP}}(f) = \log \widetilde{\mathsf{SP}}_{1/4}(f)$ completes the proof. □

Unfortunately, as far as we are aware, nobody has used this lower bound to successfully prove any explicit, non-trivial formula size lower bound of $2^{\omega(\log n)}$, so it seems to be quite difficult. However, there has been some success proving lower bounds in the monotone span program case, even without resorting to partial matrices, which we discuss in the next section.

5 Monotone span programs and monotone algorithms

A monotone function is a Boolean function in which $y \leq x$ implies $f(y) \leq f(x)$, where $y \leq x$ should be interpreted bitwise. In other words, flipping 0s to 1s either keeps the function value the same, or changes it from 0 to 1. A monotone span program is a span program in which $H_{i,0} = \{0\}$ for all i, so only 1-valued queries contribute to $H(x)$, hence $H(y) \subseteq H(x)$ whenever $y \leq x$. A monotone span program can only decide or approximate a monotone function.

Definition 5.1. For a monotone function f, define the *monotone span program size*, denoted $\mathsf{mSP}(f)$, as the minimum $s(P)$ over (families of) monotone span programs P such that P decides f; and the *approximate monotone span program size*, denoted $\mathsf{m}\widetilde{\mathsf{SP}}_\kappa(f)$, as the minimum $s(P)$ over (families of) monotone span programs P such that P κ-approximates f. We let $\mathsf{m}\widetilde{\mathsf{SP}}(f) = \mathsf{m}\widetilde{\mathsf{SP}}_{1/4}(f)$.

In contrast to $\mathsf{SP}(f)$, there are non-trivial lower bounds for $\mathsf{mSP}(f)$ for explicit monotone functions f. However, this does *not* necessarily give a lower bound on $\mathsf{SP}(f)$, and in particular, may not be a lower bound on the one-sided error quantum space complexity of f. However, lower bounds on $\log \mathsf{mSP}(f)$ or $\log \mathsf{m}\widetilde{\mathsf{SP}}(f)$ do give lower bounds on the space complexity of quantum algorithms obtained from monotone span programs, and as we will soon see, $\log \mathsf{mSP}(f)$ and $\log \mathsf{m}\widetilde{\mathsf{SP}}(f)$ are lower bounds on the space complexity of *monotone phase estimation algorithms*, described in Section 5.2. The strongest known lower bound on $\mathsf{mSP}(f)$ is the following:

Theorem 5.2 ([24]). *There is an explicit Boolean function $f : D \to \{0,1\}$ for $D \subseteq \{0,1\}^n$ such that*

$$\log \mathsf{mSP}(f) \geq \Omega(n).$$

We will adapt some of the techniques used in existing lower bounds on mSP to show a lower bound on $\widetilde{\mathsf{mSP}}(f)$ for some explicit f:

Theorem 5.3. *There is an explicit Boolean function $f : D \to \{0,1\}$ for $D \subseteq \{0,1\}^n$ such that for any constant κ,*

$$\log \widetilde{\mathsf{mSP}}_\kappa(f) \geq (\log n)^{2-o(1)}.$$

In particular, this implies a lower bound of $2^{(\log n)^{2-o(1)}}$ on $\mathsf{mSP}(f)$ for the function f in Theorem 5.3. We prove Theorem 5.3 in Section 5.1. Theorem 5.3 implies that any quantum algorithm for f obtained from a monotone span program must have space complexity $(\log n)^{2-o(1)}$, which is slightly better than the trivial lower bound of $\Omega(\log n)$. In Section 5.2, we describe a more natural class of algorithms called monotone phase estimation algorithms such that $\log \widetilde{\mathsf{mSP}}(f)$ is a lower bound on the quantum space complexity of any such algorithm computing f with bounded error. Then for the specific function f from Theorem 5.3, any monotone phase estimation algorithm for f must use space $(\log n)^{2-o(1)}$.

5.1 Monotone span program lower bounds

Our main tool in proving Theorem 5.3 will be the following.

Theorem 5.4. *For any Boolean function $f : D \to \{0,1\}$, $D \subseteq \{0,1\}^n$, and any constant $\kappa \in [0,1)$:*

$$\widetilde{\mathsf{mSP}}_\kappa(f) \geq \max_{M \in \mathbb{R}^{f^{-1}(0)\times f^{-1}(1)} : \|M\|_\infty \leq 1} \frac{\sqrt{\kappa}\text{-rank}(M)}{\max_{j\in[n]} \mathrm{rank}(M \circ \Delta_{j,1})},$$

where $\Delta_{j,1}[y,x] = 1$ if $y_i = 0$ and $x_i = 1$, and 0 else.

When, $\kappa = 0$, the right-hand side of the equation in Theorem 5.4 is the (monotone) *rank measure*, defined in [25], and shown in [11] to lower bound monotone span program size. We extend the proof for the $\kappa = 0$ case to get a lower bound on approximate span program size. We could also allow for partial matrices M, as in the non-monotone case (Lemma 4.6) but unlike the non-monotone case, it is not necessary to consider partial matrices to get non-trivial lower bounds.

Proof. Fix a monotone span program that κ-approximates f with size $\widetilde{\mathsf{mSP}}_\kappa(f)$. Let $\{\langle \omega_y| : y \in f^{-1}(0)\}$ be optimal negative witnesses, and let $\{|w_x\rangle : x \in f^{-1}(1)\}$ be approximate positive witnesses with $\left\|\Pi_{H(x)^\perp}|w_x\rangle\right\|^2 \leq \frac{\kappa}{W_-}$. Letting $\Pi_{j,b}$ denote the projector onto $H_{j,b}$, define

$$\Lambda_j = \sum_{y\in f^{-1}(0)} |y\rangle\langle\omega_y|A\Pi_{j,\bar{y}_j} \sum_{x\in f^{-1}(1)} \Pi_{j,x_j}|w_x\rangle\langle x| = \sum_{\substack{y\in f^{-1}(0):\\ y_j=0}} |y\rangle\langle\omega_y|A\Pi_{j,1} \sum_{\substack{x\in f^{-1}(1):\\ x_j=1}} \Pi_{j,1}|w_x\rangle\langle x|,$$

so Λ_j has rank at most $\dim H_j$, and so $\sum_{j\in[n]} \mathrm{rank}(\Lambda_j) \leq s(P) = \widetilde{\mathsf{mSP}}_\kappa(f)$. Furthermore, Λ_j is only supported on (y,x) such that $y_j = 0$ and $x_j = 1$, so $\Lambda_j \circ \Delta_{j,1} = \Lambda_j$. Denoting the error of

$|w_x\rangle$ as $|\mathsf{err}(x)\rangle = \Pi_{H(x)^\perp}|w_x\rangle = \sum_{j:x_j=0}\Pi_{j,1}|w_x\rangle$, we have

$$\begin{aligned}\langle y|\sum_{j\in[n]}\Lambda_j|x\rangle &= \sum_{j:y_j=0,x_j=1}\langle\omega_y|A\Pi_{j,1}|w_x\rangle = \langle\omega_y|A\sum_{j:y_j=0}\Pi_{j,1}\sum_{j:x_j=1}\Pi_{j,1}|w_x\rangle\\ &= \langle\omega_y|A(|w_x\rangle - |\mathsf{err}(x)\rangle) = \langle\omega_y|A|w_x\rangle - \langle\omega_y|A|\mathsf{err}(x)\rangle\\ \left|1-\langle y|\sum_{j\in[n]}\Lambda_j|x\rangle\right| &\leq 1-1+\left\|\langle\omega_y|A\right\|\,\||\mathsf{err}(x)\rangle\| \leq \sqrt{W_-}\sqrt{\frac{\kappa}{W_-}} = \sqrt{\kappa}.\end{aligned}$$

Then for any $M\in\mathbb{R}^{f^{-1}(0)\times f^{-1}(1)}$ with $\|M\|_\infty\leq 1$, we have:

$$\left\|M - M\circ\sum_{j\in[n]}\Lambda_j\right\|_\infty \leq \|M\|_\infty\left\|J-\sum_{j\in[n]}\Lambda_j\right\|_\infty \leq \sqrt{\kappa}.$$

Thus

$$\begin{aligned}\sqrt{\kappa}\text{-rank}(M) &\leq \mathrm{rank}\left(M\circ\sum_{j\in[n]}\Lambda_j\right) \leq \sum_{j\in[n]}\mathrm{rank}(M\circ\Lambda_j)\\ &= \sum_{j\in[n]}\mathrm{rank}(M\circ\Delta_{j,1}\circ\Lambda_j) \leq \sum_{j\in[n]}\mathrm{rank}(M\circ\Delta_{j,1})\mathrm{rank}(\Lambda_j)\\ &\leq \mathsf{m}\widetilde{\mathsf{SP}}_\kappa(f)\max_{j\in[n]}\mathrm{rank}(M\circ\Delta_{j,1}).\end{aligned}$$

□

To show a lower bound on $\mathsf{m}\widetilde{\mathsf{SP}}(f)$ for *some* explicit $f:\{0,1\}^n\to\{0,1\}$, it turns out to be sufficient to find some high approximate rank matrix $M\in\mathbb{R}^{Y\times X}$ for finite sets X and Y, and a *rectangle cover* of M, $\Delta_1,\dots,\Delta_n$, where each $\Delta_i\circ M$ has low rank. Specifically, we have the following lemma, which, with rank in place of approximate rank, has been used extensively in previous monotone span program lower bounds.

Lemma 5.5. *Let $M\in\mathbb{R}^{Y\times X}$ with $\|M\|_\infty\leq 1$, for some finite sets X and Y and $X_1,\dots,X_n\subseteq X$, $Y_1,\dots,Y_n\subseteq Y$ be such that for all $(x,y)\in X\times Y$, there exists $j\in[n]$ such that $(x,y)\in X_j\times Y_j$. Define $\Delta_j\in\{0,1\}^{Y\times X}$ by $\Delta_j[y,x]=1$ if and only if $(y,x)\in Y_j\times X_j$. There exists a monotone function $f:D\to\{0,1\}$ for $D\subseteq\{0,1\}^n$ such that for any constant $\kappa\in[0,1)$:*

$$\mathsf{m}\widetilde{\mathsf{SP}}_\kappa(f)\geq\frac{\sqrt{\kappa}\text{-rank}(M)}{\max_{j\in[n]}\mathrm{rank}(M\circ\Delta_j)}.$$

Proof. For each $y\in Y$, define $t^y\in\{0,1\}^n$ by:

$$t^y_j = \begin{cases} 0 & \text{if } y\in Y_j\\ 1 & \text{else.}\end{cases}$$

Similarly, for each $x \in X$, define $s^x \in \{0,1\}^n$ by

$$s^x_j = \begin{cases} 1 & \text{if } x \in X_j \\ 0 & \text{else.} \end{cases}$$

For every $(y, x) \in Y \times X$, there is some j such that $y_j \in Y_j$ and $x_j \in X_j$, so it cannot be the case that $s^x \leq t^y$. Thus, we can define f as the unique monotone function such that $f(s) = 1$ for every $s \in \{0,1\}^n$ such that $s^x \leq s$ for some $x \in X$, and $f(t) = 0$ for all $t \in \{0,1\}^n$ such that $t \leq t^y$ for some $y \in Y$. Then we can define a matrix $M' \in \mathbb{R}^{f^{-1}(0)\times f^{-1}(1)}$ by $M'[t^y, s^x] = M[y, x]$ for all $(y, x) \in Y \times X$, and 0 elsewhere. We have $\varepsilon\text{-rank}(M') = \varepsilon\text{-rank}(M)$ for all ε, and $\text{rank}(M' \circ \Delta_{j,1}) = \text{rank}(M \circ \Delta_j)$ for all j. The result then follows from Theorem 5.4. □

We will prove Theorem 5.3 by constructing an M with high approximate rank, and a good rectangle cover. Following [29] and [24], we will make use of a technique due to Sherstov for proving communication lower bounds, called the *pattern matrix method* [30]. We begin with some definitions.

Definition 5.6 (Fourier spectrum)**.** For a real-valued function $p : \{0,1\}^m \to \mathbb{R}$, its Fourier coefficients are defined, for each $S \subseteq [m]$:

$$\hat{p}(S) = \frac{1}{2^m} \sum_{z \in \{0,1\}^m} p(z)\chi_S(z),$$

where $\chi_S(z) = (-1)^{\sum_{i\in S} z_i}$. It is easily verified that $p = \sum_{S\subseteq[m]} \hat{p}(S)\chi_S$.

Definition 5.7 (Degree and approximate degree)**.** The *degree* of a function $p : \{0,1\}^m \to \mathbb{R}$ is defined $\deg(p) = \max\{|S| : \hat{p}(S) \neq 0\}$. For any $\varepsilon \geq 0$, $\widetilde{\deg}_\varepsilon(p) = \min\{\deg(\tilde{p}) : \|p - \tilde{p}\|_\infty \leq \varepsilon\}$.

Pattern matrices, defined by Sherstov in [30], are useful for proving lower bounds in communication complexity, because their rank and approximate rank are relatively easy to lower bound. In [29], Robere, Pitassi, Rossman and Cook first used this analysis to give lower bounds on $\mathsf{mSP}(f)$ for some f. We now state the definition, using the notation from [24], which differs slightly from [30].

Definition 5.8 (Pattern matrix)**.** For a real-valued function $p : \{0,1\}^m \to \mathbb{R}$, and a positive integer λ, the (m, λ, p)-pattern matrix is defined as $F \in \mathbb{R}^{\{0,1\}^{\lambda m}\times([\lambda]^m\times\{0,1\}^m)}$ where for $y \in \{0,1\}^{\lambda m}$, $x \in [\lambda]^m$, and $w \in \{0,1\}^m$,

$$F[y, (x, w)] = p(y|_x \oplus w),$$

where by $y|_x$, we mean the m-bit string containing one bit from each λ-sized block of y as specified by the entries of x: $(y^{(1)}_{x_1}, y^{(2)}_{x_2}, \ldots, y^{(m)}_{x_m})$, where $y^{(i)} \in \{0,1\}^\lambda$ is the i-th block of y.

For comparison, what [30] calls an (n, t, p)-pattern matrix would be a $(t, n/t, p)$-pattern matrix in our notation. As previously mentioned, a pattern matrix has the nice property that its rank (or even approximate rank) can be bounded from below in terms of properties of the Fourier spectrum of p. In particular, the following is proven in [30]:

Lemma 5.9. *Let F be the (m,λ,p)-pattern matrix for $p:\{0,1\}^m\to\{-1,+1\}$. Then for any $\varepsilon\in[0,1]$ and $\delta\in[0,\varepsilon]$, we have:*

$$\mathrm{rank}(F)=\sum_{S\subseteq[m]:\hat{p}(S)\neq 0}\lambda^{|S|}\quad \textit{and}\quad \delta\text{-rank}(F)\geq\lambda^{\widetilde{\deg}_{\varepsilon}(p)}\frac{(\varepsilon-\delta)^2}{(1+\delta)^2}.$$

This shows that we can use functions p of high approximate degree to construct pattern matrices $F\in\mathbb{R}^{\{0,1\}^{\lambda m}\times([\lambda]^m\times\{0,1\}^m)}$ of high approximate rank. To apply Lemma 5.5, we also need to find a good rectangle cover of some F.

A *b-certificate* for a function p on $\{0,1\}^m$ is an assignment $\alpha:S\to\{0,1\}$ for some $S\subseteq[m]$ such that for any $x\in\{0,1\}^m$ such that $x_j=\alpha(j)$ for all $j\in S$, $f(x)=b$. The size of a certificate is $|S|$. The following shows how to use the certificates of p to construct a rectangle cover of its pattern matrix.

Lemma 5.10. *Let $p:\{0,1\}^m\to\{-1,+1\}$, and suppose there is a set of ℓ certificates for p of size at most C such that every input satisfies at least one certificate. Then for any positive integer λ, there exists a function $f:\{0,1\}^n\to\{0,1\}$ for $n=\ell(2\lambda)^C$ such that for any $\kappa\in(0,1)$ and $\varepsilon\in[\sqrt{\kappa},1]$:*

$$\mathsf{m}\widetilde{\mathsf{SP}}_{\kappa}(f)\geq\Omega\left((\varepsilon-\sqrt{\kappa})^2\lambda^{\widetilde{\deg}_{\varepsilon}(p)}\right).$$

Proof. For $i=1,\dots,\ell$, let $\alpha_i:S_i\to\{0,1\}$ for $S_i\subset[m]$ of size $|S_i|\leq C$ be one of the ℓ certificates. That is, for each i, there is some $v_i\in\{-1,+1\}$ such that for any $x\in\{0,1\}^m$, if $x_j=\alpha_i(j)$ for all $j\in S_i$, then $p(x)=v_i$ (so α_i is a v_i-certificate).

We let F be the (m,λ,p)-pattern matrix, which has $\|F\|_\infty=1$ since p has range $\{-1,+1\}$. We will define a rectangle cover as follows. For every $i\in[\ell]$, $k\in[\lambda]^{S_i}$, and $b\in\{0,1\}^{S_i}$, define:

$$\begin{aligned}X_{i,k,b}&=\{(x,w)\in[\lambda]^m\times\{0,1\}^m:\forall j\in S_i, w_j=b_j, x_j=k_j\}\\ Y_{i,k,b}&=\{y\in\{0,1\}^{\lambda m}:\forall j\in S_i, y^{(j)}_{k_j}=b_j\oplus\alpha_i(j)\}.\end{aligned}$$

We first note that this is a rectangle cover. Fix any $y\in\{0,1\}^{\lambda m}$, $x\in[\lambda]^m$ and $w\in\{0,1\}^m$. First note that for any i, if we let b be the restriction of w to S_i, and k the restriction of x to S_i, we have $(x,w)\in X_{i,k,b}$. This holds in particular for i such that α_i is a certificate for $y|_x\oplus w$, and by assumption there is at least one such i. For such an i, we have $y^{(j)}_{x_j}\oplus w_j=\alpha(j)$ for all $j\in S_i$, so $y\in Y_{i,k,b}$. Thus, we can apply Lemma 5.5.

Note that if $(x,w)\in X_{i,k,b}$, and $y\in Y_{i,k,b}$, then $(y|_x\oplus w)[j]=y^{(j)}_{x_j}\oplus w_j=\alpha_i(j)$ for all $j\in S_i$, so $p(y|_x\oplus w)=v_i$. Letting $\Delta_{i,k,b}[y,(x,w)]=1$ if $y\in Y_{i,k,b}$ and $(x,w)\in X_{i,k,b}$, and 0 else, we have that if $y\in Y_{i,k,b}$ and $(x,w)\in X_{i,k,b}$, $(F\circ\Delta_{i,k,b})[y,(x,w)]=p(y|_x\oplus w)=v_i$, and otherwise, $(F\circ\Delta_{i,k,b})[y,(x,w)]=0$. Thus $\mathrm{rank}(F\circ\Delta_{i,k,b})=\mathrm{rank}(v_i\Delta_{i,k,b})=1$. Then by Lemma 5.5, there exists $f:\{0,1\}^n\to\{0,1\}$ where $n=\sum_{i=1}^{\ell}(2\lambda)^{|S_i|}\leq\ell(2\lambda)^C$ such that:

$$\begin{aligned}\mathsf{m}\widetilde{\mathsf{SP}}_{\kappa}(f)&\geq\sqrt{\kappa}\text{-rank}(F)\\&\geq\lambda^{\widetilde{\deg}_{\varepsilon}(p)}\frac{(\varepsilon-\sqrt{\kappa})^2}{(1+\sqrt{\kappa})^2},\ \text{by Lemma 5.9.}\end{aligned}$$

□

We now prove Theorem 5.3, restated below.

Theorem 5.3. There is an explicit Boolean function $f : D \to \{0,1\}$ for $D \subseteq \{0,1\}^n$ such that for any constant κ,

$$\log \mathsf{m}\widetilde{\mathsf{SP}}_\kappa(f) \geq \Omega((\log n)^{2-o(1)}).$$

Proof. By [8, Theorem 38], there is a function p with $\widetilde{\deg}_{1/3}(p) \geq C(p)^{2-o(1)}$, which is, up to the $o(1)$ in the exponent, the best possible separation between these two quantities. In particular, this function has $\widetilde{\deg}_{1/3}(p) \geq M^{2-o(1)}$, and $C(p) \leq M^{1+o(1)}$, where $C(p)$ is the certificate complexity of p, for some parameter M (see [8] equations (64) and (65), where p is referred to as F), and p is a function on $M^{2+o(1)}$ variables (see [8], discussion above equation (64)). Thus, there are at most $\binom{M^{2+o(1)}}{M^{1+o(1)}}2^{M^{1+o(1)}}$ possible certificates of size $M^{1+o(1)}$ such that each input satisfies at least one of them.
Then by Lemma 5.10 there exists a function $f : \{0,1\}^n \to \{0,1\}$ for $n \leq \binom{M^{2+o(1)}}{M^{1+o(1)}}2^{M^{1+o(1)}}(2\lambda)^{M^{1+o(1)}}$ such that for constant $\kappa < 1/36$ and constant λ,

$$\log \mathsf{m}\widetilde{\mathsf{SP}}_\kappa(f) \geq \Omega(\widetilde{\deg}_{1/3}(p)\log\lambda) \geq M^{2-o(1)}.$$

Then we have

$$\log n \leq \log\binom{M^{2+o(1)}}{M^{1+o(1)}} + \log 2^{M^{1+o(1)}} + M^{1+o(1)}\log(2\lambda) = O(M^{1+o(1)}\log M) = M^{1+o(1)}.$$

Thus, $\log \mathsf{m}\widetilde{\mathsf{SP}}_\kappa(f) \geq (\log n)^{2-o(1)}$, and the result for *any* κ follows using Corollary 3.9. □

Since for all total functions p, $\widetilde{\deg}_{1/3}(p) \leq C(p)^2$, where $C(p)$ is the certificate complexity of p, Lemma 5.10 cannot prove a lower bound better than $\log \mathsf{m}\widetilde{\mathsf{SP}}(p) \geq (\log n)^2$ for any n-bit function. We state a more general version of Lemma 5.10 that might have the potential to prove a better bound, but we leave this for future work.

Lemma 5.11. *Fix $p : \{0,1\}^m \to \{-1,+1\}$. For $i = 1,\dots,\ell$, let $\alpha_i : S_i \to \{0,1\}$ for $S_i \subseteq [m]$ be a partial assignment such that every $z \in \{0,1\}^m$ satisfies at least one of the assignments. Let p_i denote the restriction of p to strings z satisfying the assignment α_i. Then for every positive integer λ, there exists a function $f : \{0,1\}^n \to \{0,1\}$, where $n = \sum_{i=1}^{\ell}(2\lambda)^{|S_i|}$ such that for any $\kappa \in (0,1)$ and $\varepsilon \in [\sqrt{\kappa}, 1]$:*

$$\mathsf{m}\widetilde{\mathsf{SP}}_\kappa(f) \geq \Omega\left(\frac{(\varepsilon - \sqrt{\kappa})^2\lambda^{\widetilde{\deg}_\varepsilon(p)}}{\max_{i\in[\ell]}\sum_{S\subseteq[m]\setminus S_i:\hat{p}_i(S)\neq 0}\lambda^{|S|}}\right).$$

To make use of this lemma, one needs a function p of high approximate degree, such that for every input, there is a small assignment that lowers the degree to something small. This generalizes Lemma 5.10 because a certificate is an assignment that lowers the degree of the remaining sub-function to constant. However, we note that a p with these conditions is necessary but may not be sufficient for proving a non-trivial lower bound, because while $\sum_{S:\hat{p}_i(S)\neq 0}\lambda^{|S|} \geq \lambda^{\deg(p_i)}$, it may also be much larger if p_i has a dense Fourier spectrum.

Proof. Let F be the (m,λ,p)-pattern matrix. Let $\{X_{i,k,b}\times Y_{i,k,b}\}_{i,k,b}$ be the same rectangle covered defined in the proof of Lemma 5.10, with the difference that since the α_i are no longer certificates, the resulting submatrices of F may not have constant rank.

Let $\Delta_{i,k,b}=\sum_{y\in Y_{i,k,b}}|y\rangle\sum_{(x,w)\in X_{i,k,b}}\langle x,w|$. Then

$$F\circ\Delta_{i,k,b}=\sum_{y\in Y_{i,k,b},(x,w)\in X_{i,k,b}} p(y|_x\oplus w)|y\rangle\langle x,w|.$$

Note that when $y\in Y_{i,k,b}$ and $(x,w)\in X_{i,b,k}$, $y|_x\oplus w$ satisfies α_i, so $p(y|_x\oplus w)=p_i(y'|_{x'}\oplus w')$, where y', x' and w' are restrictions of $y\in(\{0,1\}^\lambda)^m$, $x\in[\lambda]^m$ and $w\in\{0,1\}^m$ to $[m]\setminus S_i$. Thus, continuing from above, and rearranging registers, we have:

$$\begin{aligned}F\circ\Delta_{i,k,b}&=\sum_{y'\in(\{0,1\}^\lambda)^{[m]\setminus S_i}}\;\sum_{\substack{x'\in[\lambda]^{[m]\setminus S_i},\\ w'\in\{0,1\}^{[m]\setminus S_i}}} p_i(y'|_{x'}\oplus w')|y'\rangle\langle x',w'|\otimes\sum_{\substack{\bar{y}\in(\{0,1\}^\lambda)^{S_i}:\\ \bar{y}|_k=b\oplus\alpha_i}}|\bar{y}\rangle\langle k,b|\\ &=F_i\otimes J_{2^{(\lambda-1)|S_i|},1}\end{aligned}$$

where F_i is the (m,λ,p_i)-pattern matrix, and $J_{a,b}$ is the all-ones matrix of dimension a by b, which always has rank 1 for $a,b>0$. Thus

$$\mathrm{rank}(F\circ\Delta_{i,k,b})=\mathrm{rank}(F_i)\mathrm{rank}(J_{2^{(\lambda-1)|S_i|},1})=\mathrm{rank}(F_i)=\sum_{S\subseteq[m]\setminus S_i:\hat{p}_i(S)\neq 0}\lambda^{|S|},$$

by [30]. This part of the proof follows [29, Lemma IV.6].

Then by Lemma 5.5 and Lemma 5.9, we have:

$$\mathsf{m}\widetilde{\mathsf{SP}}_\kappa(f)\geq\Omega\left(\frac{\sqrt{\kappa}\text{-rank}(F)}{\max_{i,k,b}\mathrm{rank}(F\circ\Delta_{i,k,b})}\right)\geq\Omega\left(\frac{\left(\frac{\varepsilon-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^2\lambda^{\deg_\varepsilon(p)}}{\max_i\sum_{S\subseteq[m]\setminus S_i:\hat{p}_j(S)\neq 0}\lambda^{|S|}}\right).\qquad\square$$

5.2 Monotone algorithms

In Theorem 5.3, we showed a non-trivial lower bound on $\log\mathsf{m}\widetilde{\mathsf{SP}}(f)$ for some explicit monotone function f. Unlike lower bounds on $\log\widetilde{\mathsf{SP}}(f)$, this does not give us a lower bound on the quantum space complexity of f, however, at the very least it gives us a lower bound on the quantum space complexity of a certain type of quantum algorithm. Of course, this is naturally the case, since a lower bound on $\mathsf{m}\widetilde{\mathsf{SP}}(f)$ gives us a lower bound on the quantum space complexity of any algorithm for f that is obtained from a monotone span program. However, this is not the most satisfying characterization, as it is difficult to imagine what this class of algorithms looks like.

In this section, we will consider a more natural class of algorithms whose space complexity is shown to be at least $\mathsf{m}\widetilde{\mathsf{SP}}(f)$, and in some cases $\mathsf{mSP}(f)$. We will call a quantum query algorithm a *phase estimation algorithm* if it works by estimating the amplitude on $|0\rangle$ in the phase register after running phase estimation of a unitary that makes one query. We assume that the unitary for which we perform phase estimation is of the form UO_x. This is without loss of generality, because the most general form is a unitary $U_2O_xU_1$, but we have $(U_2O_xU_1)^t|\psi_0\rangle = U_1^\dagger(UO_x)^t|\psi_0'\rangle$ where $|\psi_0'\rangle = U_1|\psi_0\rangle$, and $U = U_1U_2$. The weight on a phase of $|0\rangle$ is not affected by this global (t-independent) $U_1^\dagger$. Thus, we define a phase estimation algorithm as follows:

Definition 5.12. A *phase estimation algorithm* $\mathcal{A} = (U, |\psi_0\rangle, \delta, T, M)$ for $f : D \to \{0,1\}$, $D \subseteq \{0,1\}^n$, is defined by (families of):

- a unitary U acting on $\mathcal{H} = \mathrm{span}\{|j,z\rangle : j \in [n], z \in \mathcal{Z}\}$ for some finite set $\mathcal{Z}$;
- an initial state $|\psi_0\rangle \in \mathcal{H}$;
- a bound $\delta \in [0, 1/2)$;
- positive integers T and $M \leq \frac{1}{\sqrt{\delta}}$;

such that for any $M' \geq M$ and $T' \geq T$, the following procedure computes f with bounded error:

1. Let $\Phi(x)$ be the algorithm that runs phase estimation of UO_x on $|\psi_0\rangle$ for T' steps, and then computes a bit $|b\rangle_A$ in a new register A, such that $b = 0$ if and only if the phase estimate is 0.
2. Run M' steps of amplitude estimation to estimate the amplitude on $|0\rangle_A$ after application of $\Phi(x)$. Output 0 if the amplitude is $> \delta$.

The *query complexity* of the algorithm is $O(MT)$, and, the space complexity of the algorithm is $\log \dim \mathcal{H} + \log T + \log M + 1$.

We insist that the algorithm work not only for M and T but for any larger integers as well, because we want to ensure that the algorithm is successful because M and T are large enough, and not by some quirk of the particular chosen values. When $\delta = 0$, the algorithm has one-sided error (see Lemma 5.18).

We remark on the generality of this form of algorithm. Any algorithm can be put into this form by first converting it to a span program using the construction of Section 3.3 (Theorem 3.2), and then compiling that into an algorithm using the construction of Section 3.2 (Theorem 3.1), preserving both the time and space complexity, asymptotically. However, we will consider a special case of this type of algorithm that is *not* fully general.

Definition 5.13. A *monotone* phase estimation algorithm is a phase estimation algorithm such that if $\Pi_0(x)$ denotes the orthogonal projector onto the (+1)-eigenspace of UO_x, then for any $x \in \{0,1\}^n$, $\Pi_0(x)|\psi_0\rangle$ is in the (+1)-eigenspace of O_x.

Let us consider what is "monotone" about this definition. The algorithm outputs 0 if $|\psi_0\rangle$ has high overlap with the (+1)-eigenspace of UO_x, i. e., $\Pi_0(x)|\psi_0\rangle$ is large. In a monotone phase estimation algorithm, we know that the only contribution to $\Pi_0(x)|\psi_0\rangle$ is in the (+1)-eigenspace of O_x, which is exactly the span of $|j,z\rangle$ such that $x_j = 0$. Thus, only queries that return 0 can contribute to the algorithm rejecting.

As a simple example, Grover's algorithm is a monotone phase estimation algorithm. Specifically, let $|\psi_0\rangle = \frac{1}{\sqrt{n}}\sum_{j=1}^n |j\rangle$ and $U = (2|\psi_0\rangle\langle\psi_0| - I)$. Then UO_x is the standard Grover iterate, and $|\psi_0\rangle$ is in the span of $\mathrm{e}^{i\theta}$-eigenvectors of UO_x with $\sin|\theta| = \sqrt{|x|/n}$, so phase estimation can be used to distinguish the case $|x| = 0$ from $|x| \geq 1$. So $\Pi_0(x)|\psi_0\rangle$ is either 0, when $|x| \neq 0$, or $|\psi_0\rangle$, when $|x| = 0$. In both cases, it is in the $(+1)$-eigenspace of O_x.

It is clear that a monotone phase estimation algorithm can only decide a monotone function. However, while any quantum algorithm can be converted to a phase estimation algorithm, it is not necessarily the case that any quantum algorithm for a monotone function can be turned into a monotone phase estimation algorithm (see Remark 5.17). Thus lower bounds on the quantum space complexity of any monotone phase estimation algorithm for a monotone f do not imply lower bounds on $\mathsf{S}_U(f)$. Nevertheless, if we let $\mathsf{mS}_U(f)$ represent the minimum quantum space complexity of any monotone phase estimation algorithm for f, then a lower bound on $\mathsf{mS}_U(f)$ at least tells us that if we want to compute f with space less than said bound, we must use a non-monotone phase estimation algorithm.

Similarly, we let $\mathsf{mS}_U^1(f)$ denote the minimum quantum space complexity of any monotone phase estimation algorithm with $\delta = 0$ that computes f (with one-sided error).

The main theorem of this section states that any monotone phase estimation algorithm for f with space S can be converted to a monotone span program of size $2^{\Theta(S)}$ that approximates f, so that lower bounds on $\mathsf{m}\widetilde{\mathsf{SP}}(f)$ imply lower bounds on $\mathsf{mS}_U(f)$; and that any monotone phase estimation algorithm with $\delta = 0$ and space S can be converted to a monotone span program of size $2^{\Theta(S)}$ that decides f (exactly) so that lower bounds on $\mathsf{mSP}(f)$ imply lower bounds on $\mathsf{mS}_U^1(f)$. These conversions also preserve the query complexity. We now formally state this main result.

Theorem 5.14. *Let $\mathcal{A} = (U, |\psi_0\rangle, \delta, T, M)$ be a monotone phase estimation algorithm for f with space complexity $S = \log\dim\mathcal{H} + \log T + \log M + 1$ and query complexity $O(TM)$. Then there is a monotone span program with complexity $O(TM)$ and size $2\dim\mathcal{H} \leq 2^S$ that approximates f. If $\delta = 0$, then this span program decides f (exactly). Thus*

$$\mathsf{mS}_U(f) \geq \log \mathsf{m}\widetilde{\mathsf{SP}}(f) \quad \textit{and} \quad \mathsf{mS}_U^1(f) \geq \log \mathsf{mSP}(f).$$

We prove this theorem in Section 5.2.1. As a corollary, lower bounds on $\mathsf{mSP}(f)$, such as the one from [24], imply lower bounds on $\mathsf{mS}_U^1(f)$; and lower bounds on $\mathsf{m}\widetilde{\mathsf{SP}}(f)$ such as the one in Theorem 5.3, imply lower bounds on $\mathsf{mS}_U(f)$. In particular:

Corollary 5.15. *Let $f : \{0,1\}^n \to \{0,1\}$ be the function described in Theorem 5.3. Then $\mathsf{mS}_U(f) \geq (\log n)^{2-o(1)}$. Let $g : \{0,1\}^n \to \{0,1\}$ be the function described in Theorem 5.2. Then $\mathsf{mS}_U^1(g) \geq \Omega(n)$.*

We emphasize that while this does not give a lower bound on the quantum space complexity of f, or the one-sided quantum space complexity of g, it does show that any algorithm that uses $(\log n)^c$ space to solve f with bounded error, for $c < 2$, or $o(n)$ space to solve g with one-sided error, must be of a different form than that described in Definition 5.13.

In a certain sense, monotone phase estimation algorithms completely characterize those that can be derived from monotone span programs, because the algorithm we obtain from compiling a monotone span program is a monotone phase estimation algorithm, as stated below in Lemma 5.16. However, not all monotone phase estimation algorithms can be obtained by compiling monotone span programs, and similarly, we might hope to show that an even larger class of algorithms can be converted to monotone span programs, in order to give more strength to lower bounds on $\mathsf{mS}_U(f)$.

Lemma 5.16. *Let P be an approximate monotone span program for f with size S and complexity C. Then there is a monotone phase estimation algorithm for f with query complexity $O(C)$ and space complexity $O(\log S + \log C)$.*

Proof. Fix a monotone span program, and assume it has been appropriately scaled. Without loss of generality, we can let $H_j = H_{j,1} = \mathrm{span}\{|j,z\rangle : z \in \mathcal{Z}_j\}$ for some finite set $\mathcal{Z}_j$. Then, $\mathcal{O}_x = I - 2\Pi_{H(x)}$, which is only true because the span program is monotone. Let $U = 2\Pi_{\mathrm{row}(A)} - I$. Then $U\mathcal{O}_x = (2\Pi_{\ker(A)} - I)(2\Pi_{H(x)} - I)$ is the *span program unitary*, described in Section 3.2. Then it is simple to verify that the algorithm described in [15, Lemma 3.6] (and referred to in Section 3.2) is a phase estimation algorithm for f with query complexity $O(C)$ and space complexity $O(\log S + \log C)$.

The algorithm is a monotone phase estimation algorithm because $U = 2\Pi_{\mathrm{row}(A)} - I$ is a reflection, and $|\psi_0\rangle = |w_0\rangle = A^+|\tau\rangle$ is in the (+1)-eigenspace of U, $\mathrm{row}(A)$. Since U is a reflection, the (+1)-eigenspace of $U\mathcal{O}_x$ is exactly $(\ker(A) \cap H(x)) \oplus (\mathrm{row}(A) \cap H(x)^\perp)$, and so $\Pi_0(x)|w_0\rangle \in \mathrm{row}(A) \cap H(x)^\perp \subset H(x)^\perp$. □

Remark 5.17. We mention an example of monotone functions for which the best known quantum algorithm, in terms of space complexity, is *not* a monotone phase estimation algorithm. Every function can be expressed as a Boolean formula, and every monotone function can be expressed as a monotone Boolean formula (a formula with no negation gates), but this might be much larger than the smallest (non-monotone) formula for the function. For example, the function XOR-SAT, defined in [13], can be computed by a circuit of depth $O((\log n)^2)$, which means it has a formula of size $2^{O((\log n)^2)}$, but its monotone formula complexity is $2^{\Omega(n^\varepsilon)}$ for some constant ε.[10]

For any Boolean formula of size N, there exists a quantum algorithm that can evaluate it using $O(\sqrt{N})$ queries, and $O(\log N)$ space [27, 18]. Since this algorithm is designed via span programs, it is a phase estimation algorithm, and it is monotone if and only if the formula is monotone. For a function for which there is a separation between the monotone and non-monotone formula complexities, the smallest space quantum algorithm of this type will not be monotone. For example, for XOR-SAT, we could use a quantum algorithm that evaluates a monotone formula and has space complexity n^ε. This is a monotone phase estimation algorithm, but it is not optimal. If we instead evaluate the optimal non-monotone formula, we get a quantum algorithm (that is not monotone) with space complexity $(\log n)^2$. Of course, this does not rule out that

[10]If we pad XOR-SAT with 0s so that the input length goes from n to $N = 2^{c(\log n)^2}$ for some appropriate constant c, then the formula size becomes linear in N, while the monotone formula size is still superpolynomial in N, scaling like $2^{2^{\varepsilon\sqrt{\log N}}}$. We thank Robert Robere for this observation.

there could be some other space-optimal quantum algorithm for this problem that is a monotone phase estimation algorithm.

5.2.1 Monotone algorithms to (approximate) monotone span programs

In this section, we prove Theorem 5.14. Throughout this section, we fix a phase estimation algorithm $\mathcal{A} = (U, |\psi_0\rangle, \delta, T, M)$ that computes f, with U acting on $\mathcal{H}$. For any $x \in \{0,1\}^n$ and $\Theta \in [0,\pi]$, we let $\Pi_\Theta(x)$ denote the orthogonal projector onto the span of $e^{i\theta}$-eigenvectors of UO_x for $|\theta| \leq \Theta$. We will let $\Pi_x = \sum_{j\in[n],z\in\mathcal{Z}:x_j=1} |j,z\rangle\langle j,z|$.

We begin by drawing some conclusions about the necessary relationship between the eigenspaces of UO_x and a function f whenever a monotone phase estimation computes f. The proofs are somewhat dry and are deferred to Section 5.2.2.

Lemma 5.18. *Fix a phase estimation algorithm with $\delta = 0$ that solves f with bounded error. Then if $f(x) = 0$,*

$$\|\Pi_0(x)|\psi_0\rangle\|^2 \geq \frac{1}{M^2},$$

and for any $d < \sqrt{8}/\pi$, if $f(x) = 1$, then

$$\left\|\Pi_{d\pi/T}(x)|\psi_0\rangle\right\|^2 = 0,$$

and the algorithm always outputs 1, so it has one-sided error.

Lemma 5.19. *Fix a phase estimation algorithm with $\delta \neq 0$ that solves f with bounded error. Then there is some constant $c > 0$ such that if $f(x) = 0$,*

$$\|\Pi_0(x)|\psi_0\rangle\|^2 \geq \max\{\delta(1+c), 1/M^2\}$$

and if $f(x) = 1$, for any $d < \sqrt{8}/\pi$,

$$\left\|\Pi_{d\pi/T}(x)|\psi_0\rangle\right\|^2 \leq \frac{\delta}{1 - \frac{d^2\pi^2}{8}}.$$

To prove Theorem 5.14, we will define a monotone span program $P_{\mathcal{A}}$ as follows:

$$\begin{aligned}
H_{\text{true}} &= \text{span}\{|j,z\rangle : j \in [n], z \in \mathcal{Z}\} = \mathcal{H} \\
H_{j,1} = H_j &= \text{span}\{|j,z,1\rangle : z \in \mathcal{Z}\} \\
A|j,z,1\rangle &= \frac{1}{2}(|j,z\rangle - (-1)^1|j,z\rangle) = |j,z\rangle \\
A|j,z\rangle &= (I - U^\dagger)|j,z\rangle \\
|\tau\rangle &= |\psi_0\rangle.
\end{aligned} \tag{5.1}$$

We first show that $\Pi_0(x)|\psi_0\rangle$ is (up to scaling) a negative witness for x, whenever it is nonzero:

Lemma 5.20. *For any $x \in \{0,1\}^n$, we have*

$$w_-(x) = \frac{1}{\|\Pi_0(x)|\psi_0\rangle\|^2}.$$

In particular, $\Pi_0(x)|\psi_0\rangle/\|\Pi_0(x)|\psi_0\rangle\|^2$ is an optimal negative witness for x when $\Pi_0(x)|\psi_0\rangle \neq 0$.

Proof. Suppose $\Pi_0(x)|\psi_0\rangle \neq 0$, and let $|\omega\rangle = \Pi_0(x)|\psi_0\rangle/\|\Pi_0(x)|\psi_0\rangle\|^2$. We will first show that this is a negative witness, and then show that no negative witness can have better complexity. First, we notice that

$$\langle\omega|\tau\rangle = \langle\omega|\psi_0\rangle = \frac{\langle\psi_0|\Pi_0(x)|\psi_0\rangle}{\|\Pi_0(x)|\psi_0\rangle\|^2} = 1.$$

Next, we will see that $\langle\omega|A\Pi_{H(x)} = 0$. By the monotone phase estimation property, $O_x\Pi_0(x)|\psi_0\rangle = \Pi_0(x)|\psi_0\rangle$, and so $O_x|\omega\rangle = |\omega\rangle$, and thus $\Pi_x|\omega\rangle = 0$, where Π_x is the projector onto $|j,z\rangle$ such that $x_j = 1$. Note that $H(x) = \mathrm{span}\{|j,z,1\rangle : x_j = 1, z \in \mathcal{Z}\} \oplus \mathrm{span}\{|j,z\rangle : j \in [n], z \in \mathcal{Z}\}$. Thus $\Pi_{H(x)} = \Pi_{H_{\mathrm{true}}} + \Pi_x \otimes |1\rangle\langle 1|$. We have:

$$\langle\omega|A(\Pi_x \otimes |1\rangle\langle 1|) = \langle\omega|\Pi_x = 0.$$

Since $|\omega\rangle$ is in the $(+1)$-eigenspace of UO_x, we have $UO_x|\omega\rangle = |\omega\rangle$ so since $O_x|\omega\rangle = |\omega\rangle$, $U|\omega\rangle = |\omega\rangle$. Thus

$$\langle\omega|A\Pi_{H_{\mathrm{true}}} = \langle\omega|(I - U^\dagger) \otimes \langle 1| = (\langle\omega| - \langle\omega|) \otimes \langle 1| = 0.$$

Thus $|\omega\rangle$ is a zero-error negative witness for x. Next, we argue that it is optimal.

Suppose $|\omega\rangle$ is any optimal negative witness for x, with size $w_-(x)$. Then since $\langle\omega|\Pi_x = \langle\omega|A(\Pi_x \otimes |1\rangle\langle 1|)$ must be 0, $O_x|\omega\rangle = (I - 2\Pi_x)|\omega\rangle = |\omega\rangle$, and since $\langle\omega|A\Pi_{H_{\mathrm{true}}} = \langle\omega|(I - U^\dagger)$ must be 0, $U|\omega\rangle = |\omega\rangle$. Thus $|\omega\rangle$ is a 1-eigenvector of UO_x, so

$$\|\Pi_0(x)|\psi_0\rangle\|^2 \geq \left\|\frac{|\omega\rangle\langle\omega|}{\||\omega\rangle\|^2}|\psi_0\rangle\right\|^2 = \frac{|\langle\omega|\psi_0\rangle|^2}{\||\omega\rangle\|^2} = \frac{1}{\||\omega\rangle\|^2}.$$

We complete the proof by noticing that since $\langle\omega|A\Pi_{H_{\mathrm{true}}} = 0$, we have $\langle\omega|A = \langle\omega|\langle 1|$, and $w_-(x) = \|\langle\omega|A\|^2 = \||\omega\rangle\|^2$. □

Next we find approximate positive witnesses.

Lemma 5.21. *For any $\Theta \geq 0$, the span program $P_{\mathcal{A}}$ has approximate positive witnesses for any x with error at most $\|\Pi_\Theta(x)|\psi_0\rangle\|^2$ and complexity at most $\frac{5\pi^2}{4\Theta^2}$.*

Proof. We first define a vector $|v\rangle$ by:

$$|v\rangle = (I - (UO_x)^\dagger)^+(I - \Pi_\Theta(x))|\psi_0\rangle.$$

Note that $I - (UO_x)^\dagger$ is supported everywhere except the (+1)-eigenvectors of $(UO_x)^\dagger$, which are exactly the (+1)-eigenvectors of UO_x. Thus, $(I - \Pi_\Theta(x))|\psi_0\rangle$ is contained in this support.

Next we define

$$|w\rangle = \Big(|\psi_0\rangle - (I - U^\dagger)|v\rangle\Big)\,|1\rangle + |v\rangle.$$

Then we have:

$$A|w\rangle = |\psi_0\rangle - (I - U^\dagger)|v\rangle + (I - U^\dagger)|v\rangle = |\psi_0\rangle = |\tau\rangle.$$

So $|w\rangle$ is a positive witness, and we next compute its error for x:

$$\begin{aligned}\left\|\Pi_{H(x)^\perp}|w\rangle\right\|^2 &= \left\|\Pi_{\bar{x}}\left(|\psi_0\rangle - (I - U^\dagger)|v\rangle\right)\right\|^2\\ &= \left\|\Pi_{\bar{x}}|\psi_0\rangle - \Pi_{\bar{x}}(I - U^\dagger)(I - (UO_x)^\dagger)^+(I - \Pi_\Theta(x))|\psi_0\rangle\right\|^2.\end{aligned}$$

Above, $\Pi_{\bar{x}} = I - \Pi_x$. We now observe that

$$\Pi_{\bar{x}}(I - O_x U^\dagger) = \Pi_{\bar{x}}\left(\Pi_{\bar{x}} - (\Pi_{\bar{x}} - \Pi_x)U^\dagger\right) = \Pi_{\bar{x}}(I - U^\dagger).$$

Thus, continuing from above, we have:

$$\begin{aligned}\left\|\Pi_{H(x)^\perp}|w\rangle\right\|^2 &= \left\|\Pi_{\bar{x}}|\psi_0\rangle - \Pi_{\bar{x}}(I - O_x U^\dagger)(I - O_x U^\dagger)^+(I - \Pi_\Theta(x))|\psi_0\rangle\right\|^2\\ &= \|\Pi_{\bar{x}}|\psi_0\rangle - \Pi_{\bar{x}}(I - \Pi_\Theta(x))|\psi_0\rangle\|^2 = \|\Pi_{\bar{x}}\Pi_\Theta(x)|\psi_0\rangle\|^2\\ &\leq \|\Pi_\Theta(x)|\psi_0\rangle\|^2.\end{aligned}$$

Now we compute the complexity of $|w\rangle$. First, let $UO_x = \sum_j e^{i\theta_j}|\lambda_j\rangle\langle\lambda_j|$ be the eigenvalue decomposition of UO_x. Then

$$\begin{aligned}(I - (UO_x)^\dagger)^+ &= \sum_{j:\theta_j\neq 0}\frac{1}{1 - e^{-i\theta_j}}|\lambda_j\rangle\langle\lambda_j|\\ \text{and}\quad I - \Pi_\Theta(x) &= \sum_{j:|\theta_j|>\Theta}|\lambda_j\rangle\langle\lambda_j|.\end{aligned}$$

We can thus bound $\||v\rangle\|^2$:

$$\begin{aligned}\||v\rangle\|^2 = \left\|(I - (UO_x)^\dagger)^+(I - \Pi_\Theta(x))|\psi_0\rangle\right\|^2 &= \left\|\sum_{j:|\theta_j|>\Theta}\frac{1}{1 - e^{-i\theta_j}}\langle\lambda_j|\psi_0\rangle|\lambda_j\rangle\right\|^2\\ &= \sum_{j:|\theta_j|>\Theta}\frac{1}{4\sin^2\frac{\theta_j}{2}}|\langle\lambda_j|\psi_0\rangle|^2 \leq \frac{\pi^2}{4\Theta^2}.\end{aligned}$$

Next, using $O_x + 2\Pi_x = I - 2\Pi_x + 2\Pi_x = I$, we compute

$$\begin{aligned}
\left\| |\psi_0\rangle - (I - U^\dagger)|v\rangle \right\|^2 &= \left\| |\psi_0\rangle - (I - O_x U^\dagger - 2\Pi_x U^\dagger)(I - O_x U^\dagger)^+ (I - \Pi_\Theta(x))|\psi_0\rangle \right\|^2 \\
&= \left\| |\psi_0\rangle - (I - \Pi_\Theta(x))|\psi_0\rangle + 2\Pi_x U^\dagger (I - (UO_x)^\dagger)^+ (I - \Pi_\Theta(x))|\psi_0\rangle \right\|^2 \\
&\le \left(\|\Pi_\Theta(x)|\psi_0\rangle\| + 2 \left\| \Pi_x U^\dagger \sum_{j:|\theta_j|>\Theta} \frac{1}{1 - \mathrm{e}^{-i\theta_j}} \langle \lambda_j|\psi_0\rangle |\lambda_j\rangle \right\| \right)^2 \\
&\le \left(\|\Pi_\Theta(x)|\psi_0\rangle\| + 2 \sqrt{\sum_{j:|\theta_j|>\Theta} \frac{1}{4\sin^2\frac{\theta_j}{2}} |\langle\lambda_j|\psi_0\rangle|^2} \right)^2 \\
&\le \left(\|\Pi_\Theta(x)|\psi_0\rangle\| + \frac{\pi}{\Theta} \|(I - \Pi_\Theta(x))|\psi_0\rangle\| \right)^2 \le \frac{\pi^2}{\Theta^2}.
\end{aligned}$$

Then we have the complexity of $|w\rangle$,

$$\begin{aligned}
\||w\rangle\|^2 &= \left\| |\psi_0\rangle - (I - U^\dagger)|v\rangle \right\|^2 + \||v\rangle\|^2 \\
&\le \frac{\pi^2}{\Theta^2} + \frac{\pi^2}{4\Theta^2} = \frac{5\pi^2}{4\Theta^2}.
\end{aligned}$$ □

We conclude with the following two corollaries, whose combination gives Theorem 5.14.

Corollary 5.22. *Let $\mathcal{A} = (U, |\psi_0\rangle, 0, T, M)$ be a monotone phase estimation algorithm for f with space complexity $S = \log\dim\mathcal{H} + \log T + \log M + 1$ and query complexity $O(TM)$. Then there is a monotone span program that decides f (exactly) whose size is $2\dim\mathcal{H} \le 2^S$ and whose complexity is $O(TM)$.*

Proof. If $f(x) = 0$, then by Lemma 5.18, we have $\|\Pi_0(x)|\psi_0\rangle\|^2 \ge \frac{1}{M^2}$, so by Lemma 5.20, $w_-(x) \le M^2$. Thus $W_- \le M^2$.

If $f(x) = 1$, then by Lemma 5.18, we have $\left\|\Pi_{2/T}(x)|\psi_0\rangle\right\|^2 = 0$, so by Lemma 5.21, there's an exact positive witness for x with complexity $O(T^2)$. Thus $W_+ \le O(T^2)$, and so the span program $P_\mathcal{A}$ from (5.1) has complexity $O(TM)$. The size of the span program $P_\mathcal{A}$ is $\dim H = 2\dim\mathcal{H}$. □

Corollary 5.23. *Let $\mathcal{A} = (U, |\psi_0\rangle, \delta, T, M)$ be a monotone phase estimation algorithm for f with space complexity $S = \log\dim\mathcal{H} + \log T + \log M + 1$ and query complexity $O(TM)$. Then there is a constant $\kappa \in (0,1)$ such that there exists a monotone span program that κ-approximates f whose size is $2\dim\mathcal{H} \le 2^S$ and whose complexity is $O(TM)$.*

Proof. If $f(x) = 0$, then by Lemma 5.19, we have $\|\Pi_0(x)|\psi_0\rangle\|^2 > \delta(1+c)$ for some constant $c > 0$. Thus, by Lemma 5.20, $W_- \le \frac{1}{(1+c)\delta}$.

If $f(x) = 1$, then by Lemma 5.21, setting $\Theta = d\pi/T$ for $d = \frac{2}{\pi}\sqrt{\frac{c}{1+c}}$, (where c is the constant from above), by Lemma 5.21 there is an approximate positive witness for x with error

$$e_x = \left\| \Pi_{2\sqrt{\frac{c}{1+c}}/T}(x)|\psi_0\rangle \right\|^2$$

and complexity $O(T^2)$. By Lemma 5.19, we have

$$e_x \leq \frac{\delta}{1-\frac{d^2\pi^2}{8}} = \frac{\delta}{1-\frac{c}{2(1+c)}} = \frac{\delta(1+c)}{1+c-c/2} \leq \frac{1}{1+c/2}\frac{1}{W_-}.$$

Thus, letting $\kappa = \frac{1}{1+c/2} < 1$, we have that $P_{\mathcal{A}}$ κ-approximates f. Since the positive witness complexity is $O(T^2)$, and by Lemma 5.19, we also have $W_- \leq O(M^2)$, the complexity of $P_{\mathcal{A}}$ is $O(TM)$. The size of $P_{\mathcal{A}}$ is $\dim H = 2\dim \mathcal{H}$. □

5.2.2 Proofs of Lemma 5.18 and Lemma 5.19

We will prove the lemmas as a collection of claims. Fix $T' \geq T$ and $M' \geq M$ with which to run the algorithm. Suppose $\Phi(x)$ outputs $|\psi(x)\rangle = \sqrt{p_x}|0\rangle_A|\Phi_0(x)\rangle + \sqrt{1-p_x}|1\rangle_A|\Phi_1(x)\rangle$, and let $\tilde{p}$ denote the estimate output by the algorithm. We will let $UO_x = \sum_j e^{i\sigma_j(x)}|\lambda_j^x\rangle\langle\lambda_j^x|$ be an eigenvalue decomposition.

Claim 5.24. *If $f(x) = 0$ then $\|\Pi_0(x)|\psi_0\rangle\|^2 \geq \frac{1}{M^2}$.*

Proof. Since the algorithm computes f with bounded error, the probability of accepting x is at most $1/3$, so $\tilde{p} \leq \delta$ with probability at most $1/3$.

Amplitude estimation is just phase estimation of a unitary W_Φ such that $|\psi(x)\rangle$ is in the span of $e^{\pm 2i\theta_x}$-eigenvectors of W_Φ, where $p_x = \sin^2\theta_x$, $\theta_x \in [0, \pi/2)$ [7]. One can show that the probability of outputting an estimate $\tilde{p} = 0$ is $\sin^2(M'\theta_x)/(M'^2\sin^2(\theta_x))$, so

$$\frac{1}{3} \geq \frac{\sin^2(M'\theta_x)}{M'^2\sin^2(\theta_x)}.$$

If $M'\theta_x \leq \frac{\pi}{2}$, then this would give:

$$\frac{1}{3} \geq \frac{(2M'\theta_x/\pi)^2}{M'^2\theta_x^2} = \frac{4}{\pi^2},$$

which is a contradiction. Thus, we have:

$$M'\theta_x > \frac{\pi}{2} \quad\Rightarrow\quad \frac{2\theta_x}{\pi} > \frac{1}{M'} \quad\Rightarrow\quad \sin\theta_x > \frac{1}{M'} \quad\Rightarrow\quad \sqrt{p_x} > \frac{1}{M'}.$$

Since $\Phi(x)$ is the result of running phase estimation, we have

$$p_x = \sum_j |\langle\lambda_j^x|\psi_0\rangle|^2 \frac{\sin^2(T'\sigma_j(x)/2)}{T'^2\sin^2(\sigma_j(x)/2)} \leq \|\Pi_\Theta(x)|\psi_0\rangle\|^2 + \frac{\pi^2}{T'^2\Theta^2},$$

for any Θ. In particular, if Δ is less than the spectral gap of UO_x, we have $\|\Pi_\Delta(x)|\psi_0\rangle\| = \|\Pi_0(x)|\psi_0\rangle\|$, so

$$\frac{1}{M'^2} < \|\Pi_0(x)|\psi_0\rangle\|^2 + \frac{\pi^2}{T'^2\Delta^2}.$$

This is true for any choices $T' \geq T$ and $M' \geq M$, so we must have:

$$\frac{1}{M^2} \leq \|\Pi_0(x)|\psi_0\rangle\|^2 . \qquad \square$$

Claim 5.25. *If $f(x) = 1$ and $\delta = 0$, then for any $d < \frac{\sqrt{8}}{\pi}$, $\left\|\Pi_{d\pi/T}(x)|\psi_0\rangle\right\|^2 = 0$.*

Proof. Suppose towards a contradiction that $\left\|\Pi_{d\pi/T}(x)|\psi_0\rangle\right\|^2 > 0$. Then $p_x > 0$, and some sufficiently large $M' \geq M$ would detect this and cause the algorithm to output 0, so we must actually have $\left\|\Pi_{d\pi/T}(x)|\psi_0\rangle\right\|^2 = 0$. In fact, in order to sure that no large enough value M' detects amplitude > 0 on $|0\rangle_A$, we must have $p_x = 0$ whenever $f(x) = 1$. That means that when $f(x) = 1$, the algorithm never outputs 0, so the algorithm has one-sided error. $\square$

Claim 5.26. *There is some constant c such that if $f(x) = 0$ and $\delta > 0$ then $\|\Pi_0(x)|\psi_0\rangle\|^2 > \delta(1 + c)$.*

Proof. Recall that $\tilde{p} \in \{\sin^2(\pi m/M') : m = 0, \dots, M' - 1\}$. We will restrict our attention to choices M' such that for some integer d,

$$\sin^2 \frac{d\pi}{M'} \leq \delta < \sin^2 \frac{(d + 1/3)\pi}{M'}.$$

To see that such a choice exists, let τ be such that $\delta = \sin^2 \tau$, and note that the condition holds as long as $d \leq \frac{\tau M'}{\pi} < d + 1/3$ for some d, which is equivalent to saying that $\lfloor \frac{3\tau M'}{\pi} \rfloor = 0 \mod 3$. If $K = \lfloor \frac{1}{2} \frac{\pi}{3\tau} \rfloor$, then for any $M' \geq M$, and $\ell \geq 0$, define:

$$M_\ell = M' + \ell K.$$

Then for any $\ell > 0$,

$$\frac{3\tau}{\pi} M_\ell - \frac{3\tau}{\pi} M_{\ell-1} = \frac{3\tau}{\pi} K \in \left[\frac{1}{2} - \frac{3\tau}{\pi}, \frac{1}{2}\right],$$

so there must be one $\ell \in \{0, \dots, 6\}$ such that $\lfloor \frac{3\tau}{\pi} M_\ell \rceil = 0 \mod 3$. In particular, there is some choice M_ℓ satisfying the condition such that (using some $M' \leq \frac{1}{\sqrt{\delta}}$):

$$\sqrt{\delta} M_\ell \leq \sqrt{\delta} \left(\frac{1}{\sqrt{\delta}} + 6 \frac{\pi}{6\tau} \right) = 1 + \frac{\pi \sin \tau}{\tau} \leq 1 + \pi. \tag{5.2}$$

We will use this value as our M' for the remainder of this proof.

Let $p_x = \sin^2 \theta_x$ for $\theta_x \in [0, \pi/2]$. Let z be an integer such that $\Delta = \theta_x - \pi z/M'$ has $|\Delta| \leq \frac{\pi}{2M'}$. Then the outcome $\tilde{p} = \sin^2 \frac{\pi z}{M'}$ has probability:

$$\frac{1}{M'^2} \left| \sum_{t=0}^{M'-1} \mathrm{e}^{i2t(\theta_x - \pi z/M')} \right|^2 = \frac{1}{M'^2} \left| \sum_{t=0}^{M'-1} \mathrm{e}^{i2t\Delta} \right|^2 = \frac{\sin^2(M'\Delta)}{M'^2 \sin^2 \Delta} \geq \frac{4}{\pi^2},$$

since $|M'\Delta| \leq \frac{\pi}{2}$. Thus, by correctness, we must have $\sin^2(\pi z/M') > \delta \geq \sin^2 \frac{d\pi}{M'}$. Thus $z > d$, so

$$\frac{(d+1)\pi}{M'} \leq \frac{z\pi}{M'} = \theta_x - \Delta \leq \theta_x + \frac{\pi}{2M'}.$$

Thus:

$$\begin{aligned}
\frac{(d+1/3)\pi}{M'} + \frac{2\pi}{3M'} &\leq \theta_x + \frac{\pi}{2M'} \\
\sin\left(\frac{(d+1/3)\pi}{M'} + \frac{\pi}{6M'}\right) &\leq \sin\theta_x \\
\sin\left(\frac{(d+1/3)\pi}{M'}\right)\cos\frac{\pi}{6M'} + \cos\left(\frac{(d+1/3)\pi}{M'}\right)\sin\frac{\pi}{6M'} &\leq \sqrt{p_x} \\
\sqrt{\delta}\sqrt{1-\sin^2\frac{\pi}{6M'}} + \sqrt{1-\delta}\sin\frac{\pi}{6M'} &\leq \sqrt{p_x}
\end{aligned}$$

When $\sin^2\frac{\pi}{6M'} \leq 1-\delta$, which we can assume, the above expression is minimized when $\sin^2\frac{\pi}{6M'}$ is as small as possible. We have, using $M' \leq \frac{1+\pi}{\sqrt{\delta}}$, from (5.2):

$$\sin^2\frac{\pi}{6M'} \geq \frac{4}{36M'^2} \geq \frac{\delta}{9(1+\pi)^2}.$$

Thus, continuing from above, letting $k = \frac{1}{9(1+\pi)^2}$, we have:

$$\begin{aligned}
\sqrt{\delta}\sqrt{1-k\delta} + \sqrt{1-\delta}\sqrt{k\delta} &\leq \sqrt{p_x} \\
\delta(1-k\delta) + (1-\delta)k\delta + 2\delta\sqrt{k(1-\delta)(1-k\delta)} &\leq p_x
\end{aligned}$$

Next, notice that $(1-k\delta)(1-\delta)$ is minimized when $\delta = \frac{1+k}{2k}$, but $\delta \leq \frac{1}{2} < \frac{1+k}{2k}$, so we have, using $k < 1$ and $\delta \leq 1/2$:

$$\begin{aligned}
\delta(1 + k(1-2\delta) + 2\sqrt{k}\sqrt{(1-k/2)(1-1/2)}) &\leq p_x \\
\delta(1 + 0 + \sqrt{k}) &\leq p_x.
\end{aligned}$$

Since $\Phi(x)$ is the result of running phase estimation of UO_x for $T' \geq T$ steps, we have:

$$p_x = \sum_j |\langle\lambda_j^x|\psi_0\rangle|^2 \frac{\sin^2(\frac{T'\sigma_j(x)}{2})}{(T')^2\sin^2(\frac{\sigma_j(x)}{2})},$$

so in particular, for any $\Theta \in [0, \pi)$, we have

$$p_x \leq \|\Pi_\Theta(x)|\psi_0\rangle\|^2 + \sum_{j:|\sigma_j(x)|>\Theta} |\langle\lambda_j^x|\psi_0\rangle|^2 \frac{1}{(T')^2 \sin^2(\frac{\Theta}{2})}.$$

$$\leq \|\Pi_\Theta(x)|\psi_0\rangle\|^2 + \|(I - \Pi_\Theta(x))|\psi_0\rangle\|^2 \frac{\pi^2}{(T')^2\Theta^2}.$$

In particular, for any $\Theta < \Delta$ where Δ is the spectral gap of UO_x, we have $\|\Pi_\Theta(x)|\psi_0\rangle\| = \|\Pi_0(x)|\psi_0\rangle\|$, so for any $T' \geq T$, we have

$$\|\Pi_0(x)|\psi_0\rangle\|^2 + \frac{\pi^2}{(T')^2\Delta^2} \geq p_x \geq \delta(1 + \sqrt{k}).$$

Since this holds for any $T' \geq T$, we get

$$\|\Pi_0(x)|\psi_0\rangle\|^2 \geq \delta(1 + \sqrt{k}).$$

The proof is completed by letting $c = \sqrt{k}$. □

Claim 5.27. *If $f(x) = 1$ and $\delta > 0$ then $\left\|\Pi_{d\pi/T}(x)|\psi_0\rangle\right\|^2 (1 - d^2\pi^2/8) \leq \delta$.*

Proof. If $|\lambda\rangle$ is an $\mathrm{e}^{i\theta}$-eigenvector of UO_x for some $|\theta| \leq d\pi/T < \sqrt{8}/T$, then the probability of measuring 0 in the phase register upon performing T steps of phase estimation is:

$$p_x(\theta) := \frac{1}{T^2} \left|\sum_{t=0}^{T-1} \mathrm{e}^{it\theta}\right|^2 = \frac{\sin^2 \frac{T\theta}{2}}{T^2 \sin^2 \frac{\theta}{2}}.$$

Let $\varepsilon(x) = 1 - \frac{\sin^2 x}{x^2}$ for any x. It is simple to verify that $\varepsilon(x) \leq x^2/2$ for any x, and $\varepsilon(x) \in [0, 1]$ for any x. So we have:

$$p_x(\theta) \geq \frac{(T\theta/2)^2(1 - \varepsilon(T\theta/2))}{T^2(\theta/2)^2(1 - \varepsilon(\theta/2))} \geq 1 - \varepsilon(T\theta/2) \geq 1 - \frac{T^2\theta^2}{8}.$$

Thus, we conclude that

$$p_x \geq \left\|\Pi_{d\pi/T}(x)|\psi_0\rangle\right\|^2 \left(1 - \frac{T^2}{8}\frac{d^2\pi^2}{T^2}\right) = \left\|\Pi_{d\pi/T}(x)|\psi_0\rangle\right\|^2 \left(1 - \frac{d^2\pi^2}{8}\right).$$

If this is $> \delta$, then with some sufficiently large $M' \geq M$, amplitude estimation would detect this and cause the algorithm to output 0 with high probability. □

Acknowledgements

I am grateful to Tsuyoshi Ito for discussions that led to the construction of approximate span programs from two-sided error quantum algorithms presented in Section 3.3, and to Alex B. Grilo and Mario Szegedy for insightful comments. I am grateful to Robin Kothari for pointing out the improved separation between certificate complexity and approximate degree in [8], which led to an improvement in from $(\log n)^{7/6}$ (using [1]) to $(\log n)^{2-o(1)}$ in Theorem 5.3. I thank Robert Robere for pointing me to a separation between formula size and monotone formula size for XOR-SAT. Finally, I thank the anonymous reviewers, whose feedback has improved the presentation of these results.

References

[1] SCOTT AARONSON, SHALEV BEN-DAVID, AND ROBIN KOTHARI: Separations in query complexity using cheat sheets. In *Proc. 48th STOC*, pp. 863–876. ACM Press, 2016. [doi:10.1145/2897518.2897644, arXiv:1511.01937, ECCC:TR15-175] 6, 46

[2] NOGA ALON, TROY LEE, ADI SCHRAIBMAN, AND SANTOSH VEMPALA: The approximate rank of a matrix and its algorithmic applications. In *Proc. 45th STOC*, pp. 675–684. ACM Press, 2013. [doi:10.1145/2488608.2488694, ECCC:TR12-169] 8

[3] ANDRIS AMBAINIS: Quantum lower bounds by quantum arguments. *J. Comput. System Sci.*, 64(4):750–767, 2002. Preliminary version in STOC'00. [doi:10.1006/jcss.2002.1826, arXiv:quant-ph/0002066] 11

[4] SANJEEV ARORA AND BOAZ BARAK: *Computational Complexity: A Modern Approach*. Cambridge Univ. Press, 2009. Book. 4

[5] LÁSZLÓ BABAI, ANNA GÁL, AND AVI WIGDERSON: Superpolynomial lower bounds for monotone span programs. *Combinatorica*, 19(3):301–319, 1999. [doi:10.1007/s004930050058] 4, 5, 25

[6] HOWARD BARNUM, MICHAEL E. SAKS, AND MARIO SZEGEDY: Quantum query complexity and semi-definite programming. In *Proc. 18th IEEE Conf. on Comput. Complexity (CCC'03)*, pp. 179–193. IEEE Comp. Soc., 2003. [doi:10.1109/CCC.2003.1214419] 11

[7] GILLES BRASSARD, PETER HØYER, MICHELE MOSCA, AND ALAIN TAPP: Quantum amplitude amplification and estimation. In SAMUAL J. LOMONACA AND HOWARD E. BRANDT, editors, *Quantum Computation and Quantum Information: A Millennium Volume*, pp. 53–74. Amer. Math. Soc., 2002. [doi:10.1090/conm/305, arXiv:quant-ph/0005055] 10, 42

[8] MARK BUN AND JUSTIN THALER: A nearly optimal lower bound on the approximate degree of AC^0. *SIAM J. Comput.*, 49(4):59–96, 2020. Preliminary version in FOCS'17. [doi:10.1137/17M1161737, arXiv:1703.05784, ECCC:TR17-051] 6, 33, 46

[9] BILL FEFFERMAN AND CEDRIC YEN-YU LIN: A complete characterization of unitary quantum space. In *Proc. 9th Innovations in Theoret. Comp. Sci. Conf. (ITCS'18)*, pp. 4:1–21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. [doi:10.4230/LIPIcs.ITCS.2018.4, arXiv:1604.01384] 7

[10] BILL FEFFERMAN AND ZACHARY REMSCRIM: Eliminating intermediate measurements in space-bounded quantum computation. In *Proc. 53rd STOC*, pp. 1343–1356. ACM Press, 2021. [doi:10.1145/3406325.3451051, arXiv:2006.03530, ECCC:TR20-088] 2, 9

[11] ANNA GÁL: A characterization of span program size and improved lower bounds for monotone span programs. *Comput. Complexity*, 10(4):277–296, 2001. Preliminary version in STOC'98. [doi:10.1007/s000370100001] 4, 5, 25, 27, 29

[12] UMA GIRISH, RAN RAZ, AND WEI ZHAN: Quantum logspace algorithm for powering matrices with bounded norm. In *Proc. 48th Internat. Colloq. on Automata, Languages, and Programming (ICALP'21)*, pp. 73:1–20. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2021. [doi:10.4230/LIPIcs.ICALP.2021.73, arXiv:2006.04880, ECCC:TR20-087] 2, 9

[13] MIKA GÖÖS, PRITISH KAMATH, ROBERT ROBERE, AND DMITRY SOKOLOV: Adventures in monotone complexity and TFNP. In *Proc. 10th Innovations in Theoret. Comp. Sci. Conf. (ITCS'19)*, pp. 38:1–19. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. [doi:10.4230/LIPIcs.ITCS.2019.38, ECCC:TR18-163] 37

[14] PETER HØYER, TROY LEE, AND ROBERT ŠPALEK: Negative weights make adversaries stronger. In *Proc. 39th STOC*, pp. 526–535. ACM Press, 2007. [doi:10.1145/1250790.1250867] 11

[15] TSUYOSHI ITO AND STACEY JEFFERY: Approximate span programs. *Algorithmica*, 81(6):2158–2195, 2019. Preliminary version in ICALP'16. [doi:10.1007/s00453-018-0527-1, arXiv:1507.00432] 2, 9, 10, 12, 13, 14, 17, 18, 19, 37

[16] STACEY JEFFERY: *Frameworks for Quantum Algorithms*. Ph. D. thesis, University of Waterloo, 2014. Available at `http://uwspace.uwaterloo.ca/handle/10012/8710`. 19

[17] STACEY JEFFERY: Span programs and quantum space complexity. In *Proc. 11th Innovations in Theoret. Comp. Sci. Conf. (ITCS'20)*, pp. 4:1–37. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020. [doi:10.4230/LIPIcs.ITCS.2020.4, arXiv:1908.04232] 1

[18] STACEY JEFFERY AND SHELBY KIMMEL: Quantum algorithms for graph connectivity and formula evaluation. *Quantum*, 1:26:1–40, 2017. [doi:10.22331/q-2017-08-17-26, arXiv:1704.00765] 37

[19] RICHARD JOZSA, BARBARA KRAUS, AKIMASA MIYAKE, AND JOHN WATROUS: Matchgate and space-bounded quantum computations are equivalent. *Proc. Royal Soc. A*, 466(2115):809–830, 2010. [doi:10.1098/rspa.2009.0433] 7

[20] MAURICIO KARCHMER AND AVI WIGDERSON: On span programs. In *Proc. 8th IEEE Conf. Structure in Complexity Theory (SCT'93)*, pp. 102–111. IEEE Comp. Soc., 1993. [doi:10.1109/SCT.1993.336536] 2, 11, 12

[21] ALEXEI Y. KITAEV: Quantum measurements and the Abelian stabilizer problem. *Electron. Colloq. Comput. Complexity*, TR96-003, 1996. [ECCC, arXiv:quant-ph/9511026] 9

[22] TROY LEE, RAJAT MITTAL, BEN W. REICHARDT, ROBERT ŠPALEK, AND MARIO SZEGEDY: Quantum query complexity of state conversion. In *Proc. 52nd FOCS*, pp. 344–353. IEEE Comp. Soc., 2011. [doi:10.1109/FOCS.2011.75] 8

[23] SATYANARAYANA V. LOKAM: Complexity lower bounds using linear algebra. *Found. Trends Theor. Comp. Sci.*, 4(1–2):1–155, 2009. [doi:10.1561/0400000011] 25

[24] TONIANN PITASSI AND ROBERT ROBERE: Strongly exponential lower bounds for monotone computation. In *Proc. 49th STOC*, pp. 1246–1255. ACM Press, 2017. [doi:10.1145/3055399.3055478, ECCC:TR16-188] 5, 28, 31, 36

[25] ALEXANDER A. RAZBOROV: Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica*, 10(1):81–93, 1990. [doi:10.1007/BF02122698] 4, 25, 27, 29

[26] ALEXANDER A. RAZBOROV: On submodular complexity measures. In *Proc. London Math. Soc. Symposium on Boolean Function Complexity*, pp. 76–83, 1992. Author's website. 27

[27] BEN W. REICHARDT: Span programs and quantum query complexity: The general adversary bound is nearly tight for every Boolean function. In *Proc. 50th FOCS*, pp. 544–551. IEEE Comp. Soc., 2009. [doi:10.1109/FOCS.2009.55, arXiv:0904.2759] 2, 3, 10, 11, 12, 19, 37

[28] BEN W. REICHARDT AND ROBERT ŠPALEK: Span-program-based quantum algorithm for evaluating formulas. *Theory of Computing*, 8(13):291–319, 2012. Preliminary version in STOC'08. [doi:10.4086/toc.2012.v008a013] 2, 11

[29] ROBERT ROBERE, TONIANN PITASSI, BENJAMIN ROSSMAN, AND STEPHEN A. COOK: Exponential lower bounds for monotone span programs. In *Proc. 57th FOCS*, pp. 406–415. IEEE Comp. Soc., 2016. [doi:10.1109/FOCS.2016.51, ECCC:TR16-064] 5, 6, 31, 34

[30] ALEXANDER A. SHERSTOV: The pattern matrix method. *SIAM J. Comput.*, 40(6):1969–2000, 2011. [doi:10.1137/080733644, arXiv:0906.4291] 6, 31, 34

[31] JOHN WATROUS: Space-bounded quantum complexity. *J. Comput. System Sci.*, 59(2):281–326, 1999. [doi:10.1006/jcss.1999.1655] 6

AUTHOR

Stacey Jeffery
Senior researcher
CWI & QuSoft
Amsterdam
The Netherlands
jeffery@cwi.nl
https://homepages.cwi.nl/~jeffery/

ABOUT THE AUTHOR

STACEY JEFFERY started her academic career as an undergraduate philosophy student at McMaster University, before reading the book Gödel, Escher, Bach: An Eternal Golden Braid, after which she transfered to a Computer Science program at the University of Waterloo. She got her Ph. D. in Computer Science from the University of Waterloo under the supervision of Michele Mosca in 2014, before spending two and a half years as a postdoc at Caltech at the Insitute for Quantum Information and Matter. She now lives in Amsterdam with her husband and two-year-old daughter, who enjoys bedtime stories read from Theory of Computing.