

Span Programs and Quantum Space Complexity

Stacey Jeffery

CWI and QuSoft, the Netherlands

<https://homepages.cwi.nl/~jeffery/>

jeffery@cwi.nl

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*, 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 to compute f with bounded (two-sided) error is lower bounded by the logarithm of its *approximate span program size*. Approximate span programs were 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.

While no non-trivial lower bound is known on the span program size (or approximate span program size) of any concrete 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 span program size of an explicit function.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases Quantum space complexity, span programs

Digital Object Identifier [10.4230/LIPIcs.ITCS.2020.4](https://doi.org/10.4230/LIPIcs.ITCS.2020.4)

Related Version A full version of the paper is available at <https://arxiv.org/abs/1908.04232>

Funding 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.

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*, 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 [10] in an entirely classical setting. They defined a span program for a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ as a matrix A with each of its columns labelled by an index $i \in [n]$ and a bit $b \in \{0, 1\}$, and some fixed target vector in the columnspace of A . The span program *decides* f if for all x such that $f(x) = 1$, the target vector is in the span of the vectors labelled by (i, x_i) for $i \in [n]$. The *size* of the span program is the sum over i of the dimension of the span of the columns labelled



© Stacey Jeffery;

licensed under Creative Commons License CC-BY

11th Innovations in Theoretical Computer Science Conference (ITCS 2020).

Editor: Thomas Vidick; Article No. 4; pp. 4:1–4:36



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

46 by $(i, 0)$ or $(i, 1)$ (see also [Definition 12](#)). The span program size of f is then the minimum
 47 size of any span program deciding f , and was originally defined to lower bound the size of
 48 *counting branching programs*.

49 Several decades after the introduction of span programs, Reichardt and Špalek [18] related
 50 them to quantum algorithms, and introduced the new measure of *span program complexity*
 51 (see [Definition 13](#)). The importance of span programs in quantum algorithms stems from the
 52 ability to compile any span program for a function f into a bounded error quantum algorithm
 53 for f [17]. In particular, there is a tight correspondence between the span program *complexity*
 54 of f , and its quantum query complexity – a rather surprising and beautiful connection for
 55 a model originally introduced outside the realm of quantum computing. In contrast, the
 56 classical notion of span program *size* had received no attention in the quantum computing
 57 literature before now.

58 Ref. [8] defined the notion of an approximate span program for a function f . Loosely
 59 speaking, a span program *approximates* f if for every x such that $f(x) = 1$, the target is
 60 *close to* the span of the columns labelled by $\{(i, x_i)\}_{i \in [n]}$, and otherwise, the target is far
 61 from this span. They showed that even an approximate span program for f can be compiled
 62 into a bounded error quantum algorithm for f . In this work, we further relax the definition
 63 of an approximate span program for f , making analysis of such algorithms significantly easier
 64 (see [Definition 15](#)).

65 Let $S_U(f)$ denote the *bounded error unitary space complexity* of f , or the minimum
 66 space needed by a unitary quantum algorithm¹ that computes f with bounded error (see
 67 [Definition 7](#)). For a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, we can assume that the input is accessed
 68 by queries, so that we do not need to store the full n -bit input in working memory, but we
 69 need at least $\log n$ bits of memory to store an index into the input. Thus, a lower bound of
 70 $\omega(\log n)$ on $S_U(f)$ for some f would be non-trivial.

71 Letting $\text{SP}(f)$ denote the minimum size of a span program deciding f , and $\widetilde{\text{SP}}(f)$ the
 72 minimum size of a span program *approximating* f (see [Definition 16](#)), we have the following
 73 (see [Theorem 24](#)):

► **Theorem 1 (Informal).** *For any Boolean function f , if $S_U(f)$ denotes its bounded error unitary space complexity, and $\widetilde{\text{SP}}(f)$ its approximate span program size, then*

$$S_U(f) \geq \log \widetilde{\text{SP}}(f).$$

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

$$S_U^1(f) \geq \log \text{SP}(f).$$

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

78 The inequality $S_U^1(f) \geq \log \text{SP}(f)$, although not observed previously, follows straight-
 79 forwardly from a construction of [17] for converting a one-sided error quantum algorithm
 80 for f into a span program for f – one need only observe that the size of the resulting

¹ A unitary quantum algorithm is a quantum algorithm in which all measurements are delayed until the end. In contrast to time complexity, the space complexity of an algorithm may be significantly smaller if we allow intermediate measurements. See [6] for a discussion of the distinction between unitary and non-unitary quantum space.

span program is closely related to the space complexity of the algorithm. We adapt this construction to show how to convert a bounded (two-sided) error 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 $S_U(f) \geq \Omega(\log \widetilde{\text{SP}}(f))$. The connection between $S_U(f)$ and $\log \widetilde{\text{SP}}(f)$ is tight up to an additive term of the logarithm of the minimum complexity of any span program for f with optimal size. 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 17](#)). While the preservation of 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 – somewhat analogous to branching programs, which capture both the time and space of classical algorithms.

The significance of [Theorem 1](#) is that span program size has received extensive attention in theoretical computer science. Using results from [\[3\]](#), the connection in [Theorem 1](#) immediately implies the following ([Theorem 25](#)):

► **Theorem 2.** *For almost all Boolean functions f on n bits, $S_U^1(f) = \Omega(n)$.*

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 2](#) 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 concrete functions, several nice expressions lower bounding $\text{SP}(f)$ have been derived. By adapting one such lower bound on $\text{SP}(f)$ to $\widetilde{\text{SP}}(f)$, we get the following (see [Lemma 29](#)):

► **Theorem 3 (Informal).** *For any Boolean function f , and partial matrix $M \in (\mathbb{R} \cup \{\star\})^{f^{-1}(0) \times f^{-1}(1)}$ with $\|M\|_\infty \leq 1$:*

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

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 $\text{rank}(M)$, we get the logarithm of an expression called the *rank measure*, introduced by Razborov [\[15\]](#). The rank measure was shown by Gál to be a lower bound on span program size, SP [\[7\]](#), 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 3](#).

[Theorem 3](#) seems to give some hope of proving a non-trivial – that is, $\omega(\log n)$ – lower bound on the unitary space complexity of some explicit f , by exhibiting a matrix M for which the (approximate) rank measure is $2^{\omega(\log n)}$. In [\[15\]](#), 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” [\[2\]](#). 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, $\text{mSP}(f)$.

4:4 Span Programs and Quantum Space Complexity

122 A monotone span program is a span program where the columns of A are labelled by $(i, 1)$
 123 for $i \in [n]$ (i.e. there are no columns associated with $(i, 0)$). For a monotone function f ,
 124 the monotone span program size of f , $\text{mSP}(f)$ is the minimum size of any *monotone span*
 125 *program* for f . We can similarly define the *approximate monotone span program size* of f ,
 126 $\widetilde{\text{mSP}}(f)$. Although $\log \widetilde{\text{mSP}}(f)$ is *not* a lower bound on $\text{S}_U(f)$, even for monotone f , it is a
 127 lower bound on the space complexity of any algorithm obtained by compiling a monotone
 128 span program. We show that such algorithms are equivalent to a more natural class of
 129 algorithms called monotone phase estimation algorithms. Informally, a phase estimation
 130 algorithm is an algorithm that works by performing phase estimation of some unitary that
 131 makes a single query to the input, and estimating the amplitude on a 0 in the phase register
 132 (see [Definition 41](#)). Phase estimation algorithms are completely general, in the sense that
 133 any unitary quantum algorithm can be transformed into a phase estimation in a way that
 134 asymptotically preserves its space and query complexity. A monotone phase estimation
 135 algorithm is a phase estimation algorithm where, loosely speaking, adding 0s to the input
 136 can only make the algorithm more likely to reject (see [Definition 42](#)). We can then prove the
 137 following (see [Theorem 43](#)):

138 ► **Theorem 4 (Informal).** *For any Boolean function f , any bounded error monotone phase*
 139 *estimation algorithm for f has space complexity at least $\log \widetilde{\text{mSP}}(f)$, and any one-sided error*
 140 *monotone phase estimation algorithm for f has space complexity at least $\log \text{mSP}(f)$.*

141 Fortunately, non-trivial lower bounds for the monotone span program complexity are
 142 known for explicit functions. In Ref. [3], Babai, Gál and Wigderson showed a lower bound
 143 of $\text{mSP}(f) \geq 2^{\Omega(\log^2(n)/\log \log(n))}$ for some explicit function f , which was later improved to
 144 $\text{mSP}(f) \geq 2^{\Omega(\log^2(n))}$ by Gál [7]. In Ref. [19], a function f was exhibited with $\text{mSP}(f) \geq 2^{n^\epsilon}$
 145 for some constant $\epsilon \in (0, 1)$, and in the strongest known result, Pitassi and Robere exhibited a
 146 function f with $\text{mSP}(f) \geq 2^{\Omega(n)}$ [14]. Combined with our results, each of these implies a lower
 147 bound on the space complexity of one-sided error monotone phase estimation algorithms.
 148 For example, the result of [14] implies a lower bound of $\Omega(n)$ on the space complexity of
 149 one-sided error monotone phase estimation algorithms for a certain satisfiability problem f .
 150 This lower bound, and also the one in [19], are proven by choosing f based on a constraint
 151 satisfaction problem with high *refutation width*, which is a measure related to the space
 152 complexity of certain classes of SAT solvers, so it is intuitively not surprising that these
 153 problems should require a large amount of space to solve with one-sided error.

154 For the case of bounded error space complexity, we also prove the following (see [The-](#)
 155 [orem 32, Corollary 44](#)):

156 ► **Theorem 5 (Informal).** *There exists a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ such that any bounded*
 157 *error monotone phase estimation algorithm for f has space complexity $(\log n)^{2-o(1)}$.*

158 This lower bound is non-trivial, although much less so than the best known lower bound of
 159 $\Omega(n)$ for the one-sided case. Our result also implies a new lower bound of $2^{(\log n)^{2-o(1)}}$ on the
 160 monotone span program complexity of the function f in [Theorem 5](#).

161 To prove the lower bound in [Theorem 5](#), we apply a new technique that leverages the
 162 best possible gap between the certificate complexity and approximate polynomial degree
 163 of a function, employing a function $g : \{0, 1\}^{m^{2+o(1)}} \rightarrow \{0, 1\}$ from [5]², whose certificate

² An earlier version of this work 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 [5].

164 complexity is $m^{1+o(1)}$, and whose approximate degree is $m^{2-o(1)}$. Following a strategy of
 165 [19], we use this g to construct a *pattern matrix* [20] (see Definition 37) and use this matrix
 166 in a monotone version of Theorem 3 (see Theorem 33). The fact that certificate complexity
 167 and approximate degree of total functions are related by $\widetilde{\deg}_{1/3}(g) \leq C(g)^2$ for all g is a
 168 barrier to proving a lower bound better than $(\log n)^2$ using this technique, but we also
 169 give a generalization that has the potential to prove significantly better lower bounds (see
 170 Lemma 40).

171 1.0.0.1 Discussion and open problems

172 The most conspicuous open problem of this work is to prove a lower bound of $\omega(\log n)$ on
 173 $S_U(f)$ or even $S_U^1(f)$ for some explicit decision function f . It is known that any space S
 174 quantum Turing machine can be simulated by a deterministic classical algorithm in space S^2
 175 [21] so a lower bound of $\omega(\log^2 n)$ on classical space complexity would also give a non-trivial
 176 lower bound on quantum space complexity. If anything, the relationship to span program
 177 size is evidence that this task is extremely difficult.

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

185 Our lower bound of $(\log n)^{2-o(1)}$ only applies to the space complexity of monotone phase
 186 estimation algorithms and does not preclude the existence of a more space-efficient algorithm
 187 for f of a different form. We do know that phase estimation algorithms are fully general,
 188 in the sense that every problem has a space-optimal phase estimation algorithm. Does
 189 something similar hold for monotone phase estimation algorithms? This would imply that
 190 $\log \widetilde{\text{mSP}}(f)$ is a lower bound on $S_U(f)$ for all monotone functions f .

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

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

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

211 connection for future work.

212 We mention two previous characterizations of $S_U(f)$. Ref. [9] showed that $S_U(f)$ is equal
 213 to the logarithm of the minimum width of a *matchgate circuit* computing f , and thus our
 214 results imply that this minimum matchgate width is approximately equal to the approximate
 215 span program size of f . Separately, in Ref. [6], Fefferman and Lin showed that for every
 216 function k , inverting $2^{k(n)} \times 2^{k(n)}$ matrices is complete for the class of problems f such
 217 that $S_U(f) \leq k(n)$. Our results imply that evaluating an approximate span program of
 218 size $2^{k(n)}$ (for some suitable definition of the problem) is similarly complete for this class.
 219 Evaluating an approximate span program boils down to deciding if $\|A(x)^+|w_0\rangle\|$ is below a
 220 certain threshold, where $A(x)$ is the span program matrix A restricted to the rows labeled
 221 by $\{(i, x_i)\}_{i \in [n]}$, and $|w_0\rangle$ is some input-independent initial state; so these results are not
 222 unrelated³. We leave exploring these connections as future work.

223 1.0.0.2 Organization

224 The remainder of this paper is organized as follows. In Section 2, we present the necessary
 225 notation and quantum algorithmic preliminaries, and define quantum space complexity. In
 226 Section 3, we define span programs, and describe how they correspond to quantum algorithms.
 227 In particular, we describe how a span program can be “compiled” into a quantum algorithm,
 228 and in Section 3.2, show how a quantum algorithm can be turned into a span program, with
 229 both transformations moreorless preserving the relationships between span program size and
 230 algorithmic space, and between span program complexity and query complexity. From this
 231 correspondence, we obtain, in Section 4, expressions that lower bound the quantum space
 232 complexity of a function. While we do not know how to instantiate any of these expressions
 233 to get a non-trivial lower bound for a concrete function, in Section 5, we consider to what
 234 extent monotone span program lower bounds are meaningful lower bounds on quantum space
 235 complexity, and give the first non-trivial lower bound on the approximate monotone span
 236 program size of a function.

237 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:

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

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 $A = \sum_k \sigma_k |v_k\rangle\langle u_k|$, define:

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

238 The following lemma, from [12], is useful in the analysis of quantum algorithms.

239 ► **Lemma 6** (Effective spectral gap lemma). *Fix orthogonal projectors Π_A and Π_B . Let*
 240 *$U = (2\Pi_A - I)(2\Pi_B - I)$, and let Π_Θ be the orthogonal projector onto the $e^{i\theta}$ -eigenspaces of*
 241 *U such that $|\theta| \leq \Theta$. Then if $\Pi_A|u\rangle = 0$, $\|\Pi_\Theta\Pi_B|u\rangle\| \leq \frac{\Theta}{2} \| |u\rangle \|$.*

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

³ In the notation of Definition 12, $A(x) = A\Pi_{H(x)}$, and $|w_0\rangle = A^+|\tau\rangle$ for $|\tau\rangle$ the target. Then one can
 verify that the *positive witness size* of x is $w_+(x) = \|A(x)^+|w_0\rangle\|^2$ (see Definition 13).

2.0.0.1 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 \mathcal{O}_x be the unitary that acts as $\mathcal{O}_x|j\rangle = (-1)^{x_j}|j\rangle$ for $j \in [n]$, and $\mathcal{O}_x|0\rangle = |0\rangle$. We let $\mathcal{A}(x)$ denote the random variable obtained from measuring

$$U_T \mathcal{O}_x U_{T-1} \dots \mathcal{O}_x U_1 |0\rangle$$

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.

For a (partial) function $f : D \rightarrow \{0, 1\}$ 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 7** (Unitary Quantum Space). For a family of functions $f : D \rightarrow \{0, 1\}$ for $D \subseteq \{0, 1\}^n$, the *unitary space complexity* of f , $S_U(f)$, 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, $S_U^1(f)$ 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.

► **Remark 8.** 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 $x_i + x_j + x_k = 0 \pmod q$.

2.0.0.2 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 [11]. 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 $\log T + \log \dim(H)$. We will use the following properties:

► **Lemma 9** (Phase Estimation). If $U|\psi\rangle = e^{i\theta}|\psi\rangle$, then performing T steps of phase estimation of U on $|\psi\rangle$ and measuring the phase register results in outcome 0 with probability 1. If $U|\psi\rangle = e^{i\theta}|\psi\rangle$ for $|\theta| \in (\pi/T, \pi]$, then performing T steps of phase estimation of U on $|\psi\rangle$ results in outcome 0 with probability at most $\frac{\pi}{T\theta}$.

4:8 Span Programs and Quantum Space Complexity

276 We note that we can increase the success probability to any constant by adding some
 277 constant number k of phase registers, and doing phase estimation k times in parallel, still
 278 using a single register for U , and taking the majority. This still has space complexity
 279 $\log \dim H + O(\log T)$.

2.0.0.3 Amplitude estimation

For a unitary U acting on H , a state $|\psi_0\rangle \in H$, 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

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

281 on $U|\psi\rangle$, then, if the phase register contains some $t \in \{0, \dots, M-1\}$, compute $\tilde{p} = \sin^2 \frac{\pi t}{2M}$,
 282 which is an estimate of $\|\Pi U|\psi\rangle\|^2$ in a new register. The (time or query) complexity of this is
 283 $O(M)$ times the complexity of implementing a controlled call to U , implementing a controlled
 284 call to $2\Pi - I$, and generating $|\psi\rangle$. The space complexity is $\log T + \log \dim H + O(1)$. We
 285 have the following guarantee [4]:

286 ► **Lemma 10.** *Let $p = \|\Pi U|\psi\rangle\|^2$. There exists $\Delta = \Theta(1/M)$ such that when \tilde{p} is obtained
 287 as above from M steps of amplitude estimation, with probability at least $1/2$, $|\tilde{p} - p| \leq \Delta$.*

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

3 Span Programs and Quantum Algorithms

291 In Section 3.1, we will define a span program, its size and complexity, and what it means for
 292 a span program to approximate a function f . In Section 3.2, we prove the following theorem,
 293 which implies Theorem 1:

294 ► **Theorem 11.** *Let $f : D \rightarrow \{0, 1\}$ for $D \subseteq \{0, 1\}^n$ and let \mathcal{A} be a unitary quantum
 295 algorithm using T queries, and space S to compute f with bounded error. Then for any
 296 constant $\kappa \in (0, 1)$, there is a span program $P_{\mathcal{A}}$ with size $s(P_{\mathcal{A}}) \leq 2^{O(S)}$ that κ -approximates
 297 f with complexity $C_\kappa \leq O(T)$. If \mathcal{A} decides f with one-sided error, then $P_{\mathcal{A}}$ decides f .*

3.1 Span Programs

299 Span programs were first introduced in the context of classical complexity theory in [10],
 300 where they were used to study counting classes for nondeterministic logspace machines.
 301 While span programs can be defined with respect to any field, we will consider span programs
 302 over \mathbb{R} (or equivalently, \mathbb{C} , when convenient, see Remark 20). We use the following definition,
 303 slightly modified from [10]:

304 ► **Definition 12 (Span Program and Size).** *A span program on $\{0, 1\}^n$ consists of:*
 305 \blacksquare *Finite inner product spaces $\{H_{j,b}\}_{j \in [n], b \in \{0,1\}} \cup \{H_{\text{true}}, H_{\text{false}}\}$. We then define $H =$
 306 $\bigoplus_{j,b} H_{j,b} \oplus H_{\text{true}} \oplus H_{\text{false}}$, and for every $x \in \{0, 1\}^n$, $H(x) = H_{1,x_1} \oplus \dots \oplus H_{n,x_n} \oplus H_{\text{true}}$.⁴*

⁴ 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 $H_{n+1,0} = H_{\text{false}}$.

307 ■ A vector space V .

308 ■ A target vector $|\tau\rangle \in V$.

309 ■ A linear map $A : H \rightarrow V$.

310 We specify this span program by $P = (H, V, |\tau\rangle, A)$, and leave the decomposition of H implicit.

311 The size of the span program is $s(P) = \dim H$.

312 To recover the classical definition from [10], we can view $A = \sum_{j,b} A\Pi_{H_{j,b}}$ as a matrix,
313 with each of the columns of $A\Pi_{H_{j,b}}$ labeled by (j, b) .

314 Span programs were introduced to the study of quantum query complexity in [18]. In
315 the context of quantum query complexity, $s(P)$ is no longer the relevant measure of the
316 complexity of a span program. Instead, [18] introduce the following measures:

► **Definition 13** (Span Program Complexity and Witnesses). For $P = (H, V, |\tau\rangle, A)$ a span program 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:

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

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:

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

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

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

319 ► **Definition 14.** We define the span program size of a function f , denoted $\text{SP}(f)$, as the
320 minimum $s(P)$ over families of span programs that decide f .

We note that originally, in [10], span program size was defined

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

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

4:10 Span Programs and Quantum Space Complexity

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 [17]. 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 this situation can be improved. The following is a slight relaxation of [8, Definition 2.6]⁵.

► **Definition 15** (A Span Program that Approximately Decides a Function). *Let $f : D \rightarrow \{0, 1\}$ 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 $\|\Pi_{H(x)^\perp}|\hat{w}\rangle\|^2 \leq \frac{\kappa}{W_-}$. We define the approximate positive complexity as*

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

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 15 decides f (exactly), and $\widehat{W}_+ = W_+$. By [8], for any x ,

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

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 $\|\Pi_{H(x)^\perp}|\hat{w}\rangle\|^2 < \frac{1}{W_-}$. Thus, when a span program κ -approximates f , there is a gap of size $\frac{1-\kappa}{W_-}$ between the smallest positive witness error $\|\Pi_{H(x)^\perp}|\hat{w}\rangle\|^2$ of $x \in f^{-1}(1)$, the smallest positive witness error of $x \in f^{-1}(0)$.

► **Definition 16.** *We define the κ -approximate span program size of a function f , denoted $\widetilde{\text{SP}}_\kappa(f)$, as the minimum $s(P)$ over families of span programs that κ -approximate f . We let $\widetilde{\text{SP}}(f) = \widetilde{\text{SP}}_{1/4}(f)$.*

Then we have the following theorem, whose proof is nearly identical to that of [8, Lemma 3.6]. The only difference between [8, Lemma 3.6] and Theorem 17 below is that here we let an approximate positive witness for x be any witness with error $\|\Pi_{H(x)^\perp}|w\rangle\|^2$ at most κ/W_- , whereas in [8], an approximate positive witness must have error as small as possible. This relaxation has negligible effect on the proof.

► **Theorem 17.** *Let $f : D \rightarrow \{0, 1\}$ 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 .*

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

⁵ Which was already a relaxation of the notion of a span program deciding a function.

358 \triangleright **Claim 18.** Let P be a span program that κ -approximates $f : D \rightarrow \{0, 1\}$ for some constant
 359 κ . For any constant $\kappa' \leq \kappa$, there exists a span program P' that κ' -approximates f with
 360 $s(P') = (s(P) + 2)^{2 \frac{\log \frac{1}{\kappa'}}{\log \frac{1}{\kappa}}}$, and $C_{\kappa'}(P', D) \leq O(C_{\kappa}(P, D))$.

361 We prove [Claim 18](#) in [Appendix A](#). We have the following corollary that will be useful
 362 later, where $\widetilde{\text{mSP}}_{\kappa}$ is the *monotone approximate span program size*, defined in [Definition 30](#):

► **Corollary 19.** For any $\kappa, \kappa' \in (0, 1)$ with $\kappa' < \kappa$, and any Boolean function f ,

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

If f is monotone, we also have

$$\widetilde{\text{mSP}}_{\kappa}(f) \geq \widetilde{\text{mSP}}_{\kappa'}(f)^{\frac{1}{2} \frac{\log \frac{1}{\kappa}}{\log \frac{1}{\kappa'}}} - 2.$$

Proof. Let P κ -approximate f with optimal size, so $s(P) = \widetilde{\text{SP}}_{\kappa}(f)$. Then by [Claim 18](#), there is a span program P' that κ' -approximates f with size

$$\widetilde{\text{SP}}_{\kappa'}(f) \leq s(P') = \left(\widetilde{\text{SP}}_{\kappa}(f) + 2 \right)^{2 \frac{\log \frac{1}{\kappa'}}{\log \frac{1}{\kappa}}}.$$

363 The first result follows. The second is similar, but also includes the observation that if P is
 364 monotone, so is P' . ◀

365 ► **Remark 20.** It can sometimes be useful to construct a span program over \mathbb{C} . However, for
 366 any span program over \mathbb{C} , P , there is a span program over \mathbb{R} , P' , such that for all $x \in P_0$,
 367 $w_-(x, P') \leq w_-(x, P)$, for all $x \in P_1$, $w_+(x, P') \leq w_+(x, P)$, and $s(P') \leq 2s(P)$. Thus,
 368 we will restrict our attention to real span programs, but still allow constructions of span
 369 programs over \mathbb{C} (in particular, in [Section 3.2](#) and [Section 5.2.1](#)).

370 3.2 From Quantum Algorithms to Span Programs

371 In this section, we will show how to turn a unitary quantum algorithm into a span program,
 372 proving [Theorem 11](#), which implies [Theorem 1](#). The construction we use to prove [Theorem 11](#)
 373 is based on a construction of Reichardt for turning any one-sided error quantum algorithm
 374 into a span program whose complexity matches the algorithm's query complexity [[17](#), arXiv
 375 version]. We observe that the logarithm of the span program's size is closely related to the
 376 algorithm's space complexity. We also show that a similar construction works for two-sided
 377 error algorithms, but the resulting span program only approximately decides f .

378 3.2.0.1 The algorithm

379 Fix a function $f : D \rightarrow \{0, 1\}$ for $D \subseteq \{0, 1\}^n$, and a unitary quantum algorithm \mathcal{A} such that
 380 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
 381 $\varepsilon \in \{0, \frac{1}{3}\}$, depending on whether we want to consider a one-sided error or a bounded error
 382 algorithm. Let $p_0(x) = \Pr[\mathcal{A}(x) = 0]$, so if $f(x) = 0$, $p_0(x) \geq 2/3$, and if $f(x) = 1$, $p_0(x) \leq \varepsilon$.

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

$$|\Psi_{2T+1}(x)\rangle = U_{2T+1}\mathcal{O}_xU_{2T-1}\dots U_3\mathcal{O}_xU_1|0, 0, 0\rangle$$

4:12 Span Programs and Quantum Space Complexity

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

386 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
 387 $|\Psi_t(x)\rangle = U_t \dots U_1 |\Psi_0\rangle$ denote the state after t steps.

3.2.0.2 The span program

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

$$392 \quad H = \text{span}\{|t, b, j, z, a\rangle : t \in \{0, \dots, 2T+1\}, b \in \{0, 1\}, j \in [n] \cup \{0\}, z \in \mathcal{Z}, a \in \{0, 1\}\}.$$

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

$$395 \quad V = \text{span}\{|t, j, z, a\rangle : t \in \{0, \dots, 2T+1\}, j \in [n] \cup \{0\}, z \in \mathcal{Z}, a \in \{0, 1\}\}$$

$$396 \quad A|t, b, j, z, a\rangle = \begin{cases} |t, j, z, a\rangle - |t+1\rangle U_{t+1}|j, z, a\rangle & \text{if } t \in \{0, \dots, 2T\} \text{ is even} \\ |t, j, z, a\rangle - (-1)^b |t+1, j, z, a\rangle & \text{if } t \in \{0, \dots, 2T\} \text{ is odd} \\ |t, j, z, a\rangle & \text{if } t = 2T+1, a = 1, \text{ and } b = 0 \\ \sqrt{cT} |t, j, z, a\rangle & \text{if } t = 2T+1, a = 0, \text{ and } b = 0 \\ 0 & \text{if } t = 2T+1 \text{ and } b = 1. \end{cases}$$

398 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
 399 incrementing the counter register from $|t\rangle$ to $|t+1\rangle$. When t is even, this correspondence is
 400 clear (in that case, the value of b is ignored). When t is odd, so $U_{t+1} = \mathcal{O}_x$, then as long as
 401 $b = x_j$, $(-1)^b |t+1, j, z, a\rangle = |t+1\rangle U_{t+1}|j, z, a\rangle$. We thus define

$$402 \quad H_{j,b} = \text{span}\{|t, b, j, z, a\rangle : t \in \{0, \dots, 2T\} \text{ is odd}, z \in \mathcal{Z}, a \in \{0, 1\}\}.$$

404 For even t , applying U_{t+1} is independent of the input, so we make the corresponding states
 405 available to every input; along with states where the query register is set to $j = 0$, meaning
 406 \mathcal{O}_x acts input-independently; and accepting states, whose answer register is set to 1 at time
 407 $2T+1$:

$$408 \quad H_{\text{true}} = \text{span}\{|t, b, j, z, a\rangle : t \in \{0, \dots, 2T\} \text{ is even}, b \in \{0, 1\}, j \in [n], z \in \mathcal{Z}, a \in \{0, 1\}\}$$

$$409 \quad \oplus \text{span}\{|t, b, 0, z, a\rangle : t \in \{0, \dots, 2T\}, b \in \{0, 1\}, z \in \mathcal{Z}, a \in \{0, 1\}\}$$

$$410 \quad \oplus \text{span}\{|2T+1, b, j, z, 1\rangle : b \in \{0, 1\}, j \in [n] \cup \{0\}, z \in \mathcal{Z}\}.$$

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

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

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

416 Finally, we define:

$$417 \quad |\tau\rangle = |0, 0, 0, 0\rangle = |0\rangle |\Psi_0\rangle.$$

419 Intuitively, we can construct $|\tau\rangle$, the initial state, using a final state that has 1 in the answer
 420 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
 421 to the initial state. In the following analysis, we make this idea precise.

422 **3.2.0.3 Analysis of $P_{\mathcal{A}}$**

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

► **Lemma 21.** *For any $x \in \{0, 1\}^n$, there exists an approximate positive witness $|w\rangle$ for x in $P_{\mathcal{A}}$ such that:*

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

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

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

Proof. Let Q_x be the linear isometry that acts as

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

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

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

425 To see that this is a positive witness, we compute $A|w\rangle$, using the fact that $U_{t+1} |\Psi_t(x)\rangle =$
426 $|\Psi_{t+1}(x)\rangle$:

$$\begin{aligned} 427 \quad A|w\rangle &= \sum_{t=0}^{2T} (|t\rangle |\Psi_t(x)\rangle - |t+1\rangle U_{t+1} |\Psi_t(x)\rangle) \\ 428 \quad &\quad + |2T+1\rangle \Pi_1 |\Psi_{2T+1}(x)\rangle + |2T+1\rangle \Pi_0 |\Psi_{2T+1}(x)\rangle \\ 429 \quad &= \sum_{t=0}^{2T} |t\rangle |\Psi_t(x)\rangle - \sum_{t=0}^{2T} |t+1\rangle |\Psi_{t+1}(x)\rangle + |2T+1\rangle |\Psi_{2T+1}(x)\rangle \\ 430 \quad &= \sum_{t=0}^{2T+1} |t\rangle |\Psi_t(x)\rangle - \sum_{t=1}^{2T+1} |t\rangle |\Psi_t(x)\rangle = |0\rangle |\Psi_0(x)\rangle = |\tau\rangle. \\ 431 \end{aligned}$$

432 We next consider the error of $|w\rangle$ for x , given by $\|\Pi_{H(x)^\perp} |w\rangle\|^2$. Since $Q_x |j, z, a\rangle \in$
433 $H(x)$ for all j, z, a , and $|2T+1, 0\rangle \Pi_1 |\Psi_{2T+1}(x)\rangle \in H_{\text{true}} \subset H(x)$, $\Pi_{H(x)^\perp} |w\rangle = \frac{1}{\sqrt{cT}} |2T+$
434 $1\rangle |0\rangle \Pi_0 |\Psi_{2T+1}(x)\rangle$, so

$$435 \quad \|\Pi_{H(x)^\perp} |w\rangle\|^2 = \frac{1}{cT} \|\Pi_0 |\Psi_{2T+1}(x)\rangle\|^2 = \frac{p_0(x)}{cT}.$$

437 Finally, we compute the positive witness complexity of $|w\rangle$:

$$\begin{aligned} 438 \quad \| |w\rangle \|^2 &= \sum_{t=0}^{2T} \|Q_x |\Psi_t(x)\rangle\|^2 + \|\Pi_1 |\Psi_{2T+1}(x)\rangle\|^2 + \frac{1}{cT} \|\Pi_0 |\Psi_{2T+1}(x)\rangle\|^2 \\ 439 \quad &\leq \sum_{t=0}^{2T} \| |\Psi_t(x)\rangle \|^2 + \| |\Psi_{2T+1}(x)\rangle \|^2 = 2T + 2. \quad \blacktriangleleft \\ 440 \end{aligned}$$

4:14 Span Programs and Quantum Space Complexity

441 Next, we upper bound $w_-(x)$ whenever $f(x) = 0$:

► **Lemma 22.** *For any x that is rejected by \mathcal{A} with probability $p_0(x) > 0$,*

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

442 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

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

443 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
444 $t < 2T$ is odd), we have

$$\begin{aligned} 445 \langle \omega | A | t, x_j, j, z, a \rangle &= \langle \omega | (|t, j, z, a\rangle - (-1)^{x_j} |t+1\rangle |j, z, a\rangle) \\ 446 &= \langle \Psi_t^0(x) | j, z, a \rangle - (-1)^{x_j} \langle \Psi_{t+1}^0(x) | j, z, a \rangle \\ 447 &= \langle \Psi_{t+1}^0(x) | U_{t+1} | j, z, a \rangle - (-1)^{x_j} \langle \Psi_{t+1}^0(x) | j, z, a \rangle \\ 448 &= \langle \Psi_{t+1}^0(x) | \mathcal{O}_x | j, z, a \rangle - (-1)^{x_j} \langle \Psi_{t+1}^0(x) | j, z, a \rangle = 0. \end{aligned}$$

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

$$\begin{aligned} 452 \langle \omega | A | t, b, j, z, a \rangle &= \langle \omega | (|t, j, z, a\rangle - |t+1\rangle |j, z, a\rangle) \\ 453 &= \langle \Psi_t^0(x) | j, z, a \rangle - \langle \Psi_{t+1}^0(x) | U_{t+1} | j, z, a \rangle = 0. \end{aligned}$$

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

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

455 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
456 compute its witness complexity, first observe that $\langle \omega | A = \langle \omega | A \Pi_{H(x)^\perp}$, and

$$\begin{aligned} 457 A \Pi_{H(x)^\perp} &= \sum_{s=1}^T \sum_{\substack{j \in [n] \cup \{0\}, \\ z \in \mathcal{Z}, a \in \{0,1\}}} (|2s-1, j, z, a\rangle + (-1)^{x_j} |2s, j, z, a\rangle) \langle 2s-1, \bar{x}_j, j, z, a | \\ 458 &+ \sum_{j \in [n] \cup \{0\}, z \in \mathcal{Z}} \sqrt{cT} |2T+1, j, z, 0\rangle \langle 2T+1, 0, j, z, 0 | \\ 459 \end{aligned}$$

460 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:

$$\begin{aligned}
461 & \langle \omega | A \Pi_{H(x)^\perp} \\
462 &= \sum_{s=1}^T \sum_{\substack{j \in [n] \cup \{0\}, \\ z \in \mathcal{Z}, a \in \{0,1\}}} (\langle \Psi_{2s-1}^0(x)|j, z, a \rangle + (-1)^{x_j} \langle \Psi_{2s}^0(x)|j, z, a \rangle) \langle 2s-1, \bar{x}_j, j, z, a | \\
463 &+ \sum_{j \in [n] \cup \{0\}, z \in \mathcal{Z}} \sqrt{cT} \langle \Psi_{2T+1}^0(x)|j, z, 0 \rangle \langle 2T+1, 0, j, z, 0 | \\
464 &= \sum_{s=1}^T \sum_{j \in [n] \cup \{0\}, z \in \mathcal{Z}, a \in \{0,1\}} 2(-1)^{x_j} \langle \Psi_{2s}^0(x)|j, z, a \rangle \langle 2s-1, \bar{x}_j, j, z, a | \\
465 &+ \sum_{j \in [n] \cup \{0\}, z \in \mathcal{Z}} \sqrt{cT} \langle \Psi_{2T+1}^0(x)|j, z, 0 \rangle \langle 2T+1, 0, j, z, 0 |. \\
466 &
\end{aligned}$$

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

$$\begin{aligned}
468 & \|\langle \bar{\omega} | A\|^2 = \frac{1}{p_0(x)^2} \|\langle \omega | A \Pi_{H(x)^\perp}\|^2 \\
469 &= \frac{1}{p_0(x)^2} \sum_{s=1}^T \sum_{\substack{j \in [n] \cup \{0\}, \\ z \in \mathcal{Z}, \\ a \in \{0,1\}}} 4 |\langle \Psi_{2s}^0(x)|j, z, a \rangle|^2 + \frac{1}{p_0(x)^2} \sum_{\substack{j \in [n] \cup \{0\}, \\ z \in \mathcal{Z}}} cT |\langle \Psi_{2T+1}^0(x)|j, z, 0 \rangle|^2 \\
470 &= \frac{4}{p_0(x)^2} \sum_{s=1}^T \|\Psi_{2s}^0(x)\|^2 + \frac{cT}{p_0(x)^2} \|\Psi_{2T+1}^0(x)\|^2. \\
471 &
\end{aligned}$$

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

$$473 \|\langle \bar{\omega} | A\|^2 = \frac{4T}{p_0(x)} + \frac{cT}{p_0(x)} \leq \frac{4+c}{2/3} T \text{ when } f(x) = 0. \quad \blacktriangleleft$$

475 We conclude the proof of [Theorem 11](#) with the following corollary, from which [Theorem 11](#)
476 follows immediately, by appealing to [Claim 18](#) with $\kappa = \frac{9}{10}$ and κ' any constant in $(0, 1)$.

477 **► Corollary 23.** *Let $c = 5$, in the definition of P_A . Then:*

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

Proof. We first compute $s(P_A) = \dim H$ using the fact that the algorithm uses space

$$481 S = \log \dim \text{span}\{|j, z, a\rangle : j \in [n] \cup \{0\}, z \in \mathcal{Z}, a \in \{0, 1\}\} + \log T :$$

$$482 \dim H = (\dim \text{span}\{|t, b\rangle : t \in \{0, \dots, 2T+1\}, b \in \{0, 1\}\}) 2^{S-\log T} = 2^{S+O(1)}.$$

We prove the third statement, as the second is similar. By [Lemma 22](#), using $c = 5$, we have

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

By [Lemma 21](#), 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:

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

4:16 Span Programs and Quantum Space Complexity

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

4 Span Programs and Space Complexity

487 Using the transformation from algorithms to span programs from [Section 3.2](#), we immediately
 488 have the following connections between span program size and space complexity.

489 **► Theorem 24.** *For any $f : D \rightarrow \{0, 1\}$ for $D \subseteq \{0, 1\}^n$, we have*

$$490 \quad S_U(f) \geq \Omega(\log \widetilde{\text{SP}}(f)) \quad \text{and} \quad S_U^1(f) \geq \Omega(\log \text{SP}(f)).$$

491 [Theorem 24](#) is a corollary of [Theorem 11](#). [Theorem 17](#) shows that the lower bound for
 492 $S_U(f)$ in [Theorem 24](#) is part of a *tight* correspondence between space complexity and
 493 $\log s(P) + \log C(P)$.

494 [Theorem 2.9](#) of [\[3\]](#) gives a lower bound of $\text{SP}(f) \geq \Omega(2^{n/3}/(n \log n)^{1/3})$ for almost all
 495 n -bit Boolean functions. Combined with [Theorem 24](#), we immediately have:

496 **► Theorem 25.** *For almost all Boolean functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$, $S_U^1(f) = \Omega(n)$.*

497 Ideally, we would like to use the lower bound in [Theorem 24](#) to prove a non-trivial
 498 lower bound for $S_U(f)$ or $S_U^1(f)$ for some concrete f . Fortunately, there are somewhat nice
 499 expressions lower bounding $\text{SP}(f)$ [\[15, 7\]](#), which we extend to lower bounds of $\widetilde{\text{SP}}(f)$ in
 500 the remainder of this section. However, on the unfortunate side, there has already been
 501 significant motivation to instantiate these expressions to non-trivial lower bounds for concrete
 502 f , with no success. There has been some success in *monotone* versions of these lower bounds,
 503 which we discuss more in [Section 5](#).

504 For a function $f : D \rightarrow \{0, 1\}$ for $D \subseteq \{0, 1\}^n$, and an index $j \in [n]$, we let $\Delta_{f,j} \in$
 505 $\{0, 1\}^{f^{-1}(0) \times f^{-1}(1)}$ be defined by $\Delta_{f,j}[y, x] = 1$ if and only if $x_j \neq y_j$. When f is clear from
 506 context, we simply denote this by Δ_j . The following tight characterization of $\text{SP}(f)$ may be
 507 found in, for example, [\[13\]](#).

508 **► Lemma 26.** *For any $f : D \rightarrow \{0, 1\}$ for $D \subseteq \{0, 1\}^n$,*

$$509 \quad \text{SP}(f) = \text{minimize} \quad \sum_{j \in [n]} \text{rank}(\Lambda_j)$$

$$510 \quad \text{subject to} \quad \forall j \in [n], \Lambda_j \in \mathbb{R}^{f^{-1}(0) \times f^{-1}(1)}$$

$$511 \quad \sum_{j \in [n]} \Lambda_j \circ \Delta_j = J,$$

$$512 \quad \sum_{j \in [n]} \Lambda_j \circ \Delta_j = J,$$

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

514 By [Theorem 24](#), the logarithm of the above is a lower bound on $S_U^1(f)$. We modify [Lemma 26](#)
 515 to get the following approximate version, whose logarithm lower bounds $S_U(f)$ when $\kappa = \frac{1}{4}$.

516 **► Lemma 27.** *For any $\kappa \in [0, 1)$, and $f : D \rightarrow \{0, 1\}$ for $D \subseteq \{0, 1\}^n$,*

$$517 \quad \widetilde{\text{SP}}_\kappa(f) \geq \text{minimize} \quad \sum_{j \in [n]} \text{rank}(\Lambda_j) \tag{1}$$

$$518 \quad \text{subject to} \quad \forall j \in [n], \Lambda_j \in \mathbb{R}^{f^{-1}(0) \times f^{-1}(1)}$$

$$519 \quad \left\| \sum_{j \in [n]} \Lambda_j \circ \Delta_j - J \right\|_\infty \leq \sqrt{\kappa}.$$

$$520$$

Proof. Fix a span program that κ -approximates f with $s(P) = \widetilde{\text{SP}}_\kappa(f)$, 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 $\|\Pi_{H(x)}|w_x\rangle\|^2 \leq \frac{\kappa}{W_-}$. Letting $\Pi_{j,b}$ denote the projector onto $H_{j,b}$, define

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

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

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

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

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 $\leq \widetilde{\text{SP}}_\kappa(f)$, so the result follows. \blacktriangleleft

As a corollary of the above, and the connection between span program size and unitary quantum space complexity stated in [Theorem 24](#), the logarithm of the expression in (1) with $\kappa = \frac{1}{4}$ is a lower bound on $S_U(f)$, and with $\kappa = 0$, it is a lower bound on $S_U^1(f)$. 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 $\bar{M} \in \mathbb{R}^{f^{-1}(0) \times f^{-1}(1)}$ such that $\bar{M}[y, x] = M[y, x]$ whenever $M[y, x] \neq \star$. For a partial matrix M , define $\text{rank}(M)$ to be the smallest rank of any completion of M , and ε - $\text{rank}(M)$ 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:

► **Lemma 28.** *For all Boolean functions $f : D \rightarrow \{0, 1\}$, 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 $\max\{|M[y, x]| : M[y, x] \neq \star\} \leq 1$:*

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

In [15], Razborov showed that the expression on the right-hand side in [Lemma 28](#) is a lower bound on the logarithm of the *formula size* of f (Ref. [7] related this to $\text{SP}(f)$). Later, in [16],

4:18 Span Programs and Quantum Space Complexity

545 Razborov noted that when restricted to non-partial matrices, this can never give a better
 546 bound than n . Thus, to prove a non-trivial lower bound on $S_U^1(f)$ using this method, one
 547 would need to use a partial matrix. We prove the following generalization to the approximate
 548 case.

549 ► **Lemma 29.** *For all Boolean functions $f : D \rightarrow \{0, 1\}$, with $D \subseteq \{0, 1\}^n$, and all partial*
 550 *matrices $M \in (\mathbb{R} \cup \{\star\})^{f^{-1}(0) \times f^{-1}(1)}$ such that $\max\{|M[y, x]| : M[y, x] \neq \star\} \leq 1$:*

$$551 \quad S_U(f) \geq \Omega \left(\log \left(\frac{\frac{1}{2}\text{-rank}(M)}{\max_{i \in [n]} \text{rank}(M \circ \Delta_i)} \right) \right).$$

Proof. Let $\{\Lambda_j\}_j$ be an optimal feasible solution for the expression from Lemma 27, so

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

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

$$555 \quad \left| \left(\sum_{j \in [n]} \overline{M}_j \circ \Lambda_j \right) [y, x] - M[y, x] \right| = \left| \sum_{j \in [n]} M[y, x] \Delta_j [y, x] \Lambda_j [y, x] - M[y, x] \right|$$

$$556 \quad \leq |M[y, x]| \left\| \sum_{j \in [n]} \Delta_j \circ \Lambda_j - J \right\|_\infty \leq \sqrt{\kappa}.$$

558 Thus

$$559 \quad \sqrt{\kappa} \cdot \text{rank}(M) \leq \text{rank} \left(\sum_{j \in [n]} \overline{M}_j \circ \Lambda_j \right) \leq \sum_{j \in [n]} \text{rank}(\overline{M}_j \circ \Lambda_j).$$

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

$$562 \quad \sqrt{\kappa} \cdot \text{rank}(M) \leq \sum_{j \in [n]} \text{rank}(\Lambda_j) \text{rank}(\overline{M}_j) \leq \widetilde{\text{SP}}_\kappa(f) \max_{j \in [n]} \text{rank}(M \circ \Delta_j).$$

564 Setting $\kappa = \frac{1}{4}$, and noting that by Theorem 24, $S_U(f) \geq \log \widetilde{\text{SP}}(f) = \log \widetilde{\text{SP}}_{1/4}(f)$ completes
 565 the proof. ◀

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

570 **5 Monotone Span Programs and Monotone Algorithms**

571 A monotone function is a Boolean function in which $y \leq x$ implies $f(y) \leq f(x)$, where $y \leq x$
 572 should be interpreted bitwise. In other words, flipping 0s to 1s in the input either keeps the
 573 function value the same, or changes it from 0 to 1. A monotone span program is a span
 574 program in which $H_{i,0} = \{0\}$ for all i , so only 1-valued queries contribute to $H(x)$, and
 575 $H(y) \subseteq H(x)$ whenever $y \leq x$. A monotone span program can only decide or approximate a
 576 monotone function.

577 **► Definition 30.** For a monotone function f , define the monotone span program size,
 578 denoted $\text{mSP}(f)$, as the minimum $s(P)$ over (families of) monotone span programs P such
 579 that P decides f ; and the approximate monotone span program size, denoted $\widetilde{\text{mSP}}_\kappa(f)$, as
 580 the minimum $s(P)$ over (families of) monotone span programs P such that P κ -approximates
 581 f . We let $\widetilde{\text{mSP}}(f) = \widetilde{\text{mSP}}_{1/4}(f)$.

582 In contrast to $\text{SP}(f)$, there are non-trivial lower bounds for $\text{mSP}(f)$ for explicit monotone
 583 functions f . However, this does *not* necessarily give a lower bound on $\text{SP}(f)$, and in particular,
 584 may not be a lower bound on the one-sided error quantum space complexity of f . However,
 585 lower bounds on $\log \text{mSP}(f)$ or $\log \widetilde{\text{mSP}}(f)$ do give lower bounds on the space complexity
 586 of quantum algorithms obtained from monotone span programs, and as we will soon see,
 587 $\log \text{mSP}(f)$ and $\log \widetilde{\text{mSP}}(f)$ are lower bounds on the space complexity of *monotone phase*
 588 *estimation algorithms*, described in [Section 5.2](#). The strongest known lower bound on $\text{mSP}(f)$
 589 is the following:

► Theorem 31 ([14]). *There is an explicit Boolean function $f : D \rightarrow \{0, 1\}$ for $D \subseteq \{0, 1\}^n$ such that*

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

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

► Theorem 32. *There is an explicit Boolean function $f : D \rightarrow \{0, 1\}$ for $D \subseteq \{0, 1\}^n$ such that for any constant κ ,*

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

592 In particular, this implies a lower bound of $2^{(\log n)^{2-o(1)}}$ on $\text{mSP}(f)$ for the function f in
 593 [Theorem 32](#). We prove [Theorem 32](#) in [Section 5.1](#). [Theorem 32](#) implies that any quantum
 594 algorithm for f obtained from a monotone span program must have space complexity
 595 $(\log n)^{2-o(1)}$, which is slightly better than the trivial lower bound of $\Omega(\log n)$. In [Section 5.2](#),
 596 we describe a more natural class of algorithms called monotone phase estimation algorithms
 597 such that $\log \widetilde{\text{mSP}}(f)$ is a lower bound on the quantum space complexity of any such algorithm
 598 computing f with bounded error. Then for the specific function f from [Theorem 32](#), any
 599 monotone phase estimation algorithm for f must use space $(\log n)^{2-o(1)}$.

600 5.1 Monotone Span Program Lower Bounds

601 Our main tool in proving [Theorem 32](#) will be the following.

► Theorem 33. *For any Boolean function $f : D \rightarrow \{0, 1\}$, $D \subseteq \{0, 1\}^n$, and any constant $\kappa \in [0, 1)$:*

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

602 where $\Delta_{j,1}[y, x] = 1$ if $y_j = 0$ and $x_j = 1$, and 0 else.

603 When, $\kappa = 0$, the right-hand side of the equation in [Theorem 33](#) is the (monotone) *rank*
 604 *measure*, defined in [15], and shown in [7] to lower bound monotone span program size. We
 605 extend the proof for the $\kappa = 0$ case to get a lower bound on approximate span program
 606 size. We could also allow for partial matrices M , as in the non-monotone case ([Lemma 29](#))
 607 but unlike the non-monotone case, it is not necessary to consider partial matrices to get
 608 non-trivial lower bounds.

4:20 Span Programs and Quantum Space Complexity

609 **Proof.** Fix a monotone span program that κ -approximates f with size $\text{mSP}_\kappa(f)$. Let
 610 $\{|\omega_y\rangle : y \in f^{-1}(0)\}$ be optimal negative witnesses, and let $\{|w_x\rangle : x \in f^{-1}(1)\}$ be approximate
 611 positive witnesses with $\|\Pi_{H(x)^\perp}|w_x\rangle\|^2 \leq \frac{\kappa}{W_-}$. Letting $\Pi_{j,b}$ denote the projector onto $H_{j,b}$,
 612 define

$$\begin{aligned} 613 \quad \Lambda_j &= \sum_{y \in f^{-1}(0)} |y\rangle \langle \omega_y| A \Pi_{j, \bar{y}_j} \sum_{x \in f^{-1}(1)} \Pi_{j, x_j} |w_x\rangle \langle x| \\ 614 \quad &= \sum_{\substack{y \in f^{-1}(0): \\ y_j = 0}} |y\rangle \langle \omega_y| A \Pi_{j, 1} \sum_{\substack{x \in f^{-1}(1): \\ x_j = 1}} \Pi_{j, 1} |w_x\rangle \langle x|, \\ 615 \end{aligned}$$

616 so Λ_j has rank at most $\dim H_j$, and so $\sum_{j \in [n]} \text{rank}(\Lambda_j) \leq s(P) = \text{mSP}_\kappa(f)$. Furthermore,
 617 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
 618 error of $|w_x\rangle$ as $|\text{err}(x)\rangle = \Pi_{H(x)^\perp}|w_x\rangle = \sum_{j: x_j=0} \Pi_{j,1}|w_x\rangle$, we have

$$\begin{aligned} 619 \quad \langle y | \sum_{j \in [n]} \Lambda_j |x\rangle &= \sum_{j: y_j=0, x_j=1} \langle \omega_y | A \Pi_{j,1} |w_x\rangle = \langle \omega_y | A \sum_{j: y_j=0} \Pi_{j,1} \sum_{j: x_j=1} \Pi_{j,1} |w_x\rangle \\ 620 \quad &= \langle \omega_y | A (|w_x\rangle - |\text{err}(x)\rangle) = \langle \omega_y | A |w_x\rangle - \langle \omega_y | A |\text{err}(x)\rangle \\ 621 \quad \left| 1 - \langle y | \sum_{j \in [n]} \Lambda_j |x\rangle \right| &\leq 1 - 1 + \|\langle \omega_y | A \|\|\text{err}(x)\rangle\| \leq \sqrt{W_-} \sqrt{\frac{\kappa}{W_-}} = \sqrt{\kappa}. \\ 622 \end{aligned}$$

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

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

623 Thus

$$\begin{aligned} 624 \quad \sqrt{\kappa}\text{-rank}(M) &\leq \text{rank} \left(M \circ \sum_{j \in [n]} \Lambda_j \right) \leq \sum_{j \in [n]} \text{rank}(M \circ \Lambda_j) = \sum_{j \in [n]} \text{rank}(M \circ \Delta_{j,1} \circ \Lambda_j) \\ 625 \quad &\leq \sum_{j \in [n]} \text{rank}(M \circ \Delta_{j,1}) \text{rank}(\Lambda_j) \leq \text{mSP}_\kappa(f) \max_{j \in [n]} \text{rank}(M \circ \Delta_{j,1}). \quad \blacktriangleleft \\ 626 \end{aligned}$$

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

632 **► Lemma 34.** *Let $M \in \mathbb{R}^{Y \times X}$ with $\|M\|_\infty \leq 1$, for some finite sets X and Y and*
 633 *$X_1, \dots, X_n \subseteq X$, $Y_1, \dots, Y_n \subseteq Y$ be such that for all $(x, y) \in X \times Y$, there exists $j \in [n]$ such*
 634 *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 $(y, x) \in Y_j \times X_j$.*
 635 *There exists a monotone function $f : D \rightarrow \{0, 1\}$ for $D \subseteq \{0, 1\}^n$ such that for any constant*
 636 *$\kappa \in [0, 1)$:*

$$637 \quad \text{mSP}_\kappa(f) \geq \frac{\sqrt{\kappa}\text{-rank}(M)}{\max_{j \in [n]} \text{rank}(M \circ \Delta_j)}.$$

638

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

$$640 \quad t_j^y = \begin{cases} 0 & \text{if } y \in Y_j \\ 1 & \text{else.} \end{cases}$$

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

$$643 \quad s_j^x = \begin{cases} 1 & \text{if } x \in X_j \\ 0 & \text{else.} \end{cases}$$

645 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 can't be the
646 case that $s^x \leq t^y$. Thus, we can define f as the unique monotone function such that $f(s) = 1$
647 for every $s \in \{0, 1\}^n$ such that $s^x \leq s$ for some $x \in X$, and $f(t) = 0$ for all $t \in \{0, 1\}^n$
648 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
649 $M'[t^y, s^x] = M[y, x]$ for all $(y, x) \in Y \times X$, and 0 elsewhere. We have ε -rank(M') = ε -rank(M)
650 for all ε , and rank($M' \circ \Delta_{j,1}$) = rank($M \circ \Delta_j$) for all j . The result then follows from
651 [Theorem 33](#). ◀

652 We will prove [Theorem 32](#) by constructing an M with high approximate rank, and a
653 good rectangle cover $\{X_j \times Y_j\}_j$. Following [\[19\]](#) and [\[14\]](#), we will make use of a technique
654 due to Sherstov for proving communication lower bounds, called the *pattern matrix method*
655 [\[20\]](#). We begin with some definitions.

▶ **Definition 35** (Fourier spectrum). For a real-valued function $p : \{0, 1\}^m \rightarrow \mathbb{R}$, its Fourier coefficients are defined, for each $S \subseteq [m]$:

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

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

657 ▶ **Definition 36** (Degree and approximate degree). The degree of a function $p : \{0, 1\}^m \rightarrow \mathbb{R}$
658 is defined $\deg(p) = \max\{|S| : \hat{p}(S) \neq 0\}$. For any $\varepsilon \geq 0$, $\deg_\varepsilon(p) = \min\{\deg(\tilde{p}) : \|p - \tilde{p}\|_\infty \leq$
659 $\varepsilon\}$.

660 Pattern matrices, defined by Sherstov in [\[20\]](#), are useful for proving lower bounds in
661 communication complexity, because their rank and approximate rank are relatively easy to
662 lower bound. In [\[19\]](#), Robere, Pitassi, Rossman and Cook first used this analysis to give
663 lower bounds on $\text{mSP}(f)$ for some f . We now state the definition, using the notation from
664 [\[14\]](#), which differs slightly from [\[20\]](#).

▶ **Definition 37** (Pattern matrix). For a real-valued function $p : \{0, 1\}^m \rightarrow \mathbb{R}$, 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 $w \in \{0, 1\}^m$,

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

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

667 For comparison, what [\[20\]](#) calls an (n, t, p) -pattern matrix would be a $(t, n/t, p)$ -pattern
668 matrix in our notation. As previously mentioned, a pattern matrix has the nice property
669 that its rank (or even approximate rank) can be lower bounded in terms of properties of the
670 Fourier spectrum of p . In particular, the following is proven in [\[20\]](#):

4:22 Span Programs and Quantum Space Complexity

671 ► **Lemma 38.** *Let F be the (m, λ, p) -pattern matrix for $p : \{0, 1\}^m \rightarrow \{-1, +1\}$. Then for
672 any $\varepsilon \in [0, 1]$ and $\delta \in [0, \varepsilon]$, we have:*

$$673 \quad \text{rank}(F) = \sum_{S \subseteq [m]: \hat{p}(S) \neq 0} \lambda^{|S|} \quad \text{and} \quad \delta\text{-rank}(F) \geq \lambda^{\widetilde{\deg}_\varepsilon(p)} \frac{(\varepsilon - \delta)^2}{(1 + \delta)^2}.$$

674

675 This shows that we can use functions p of high approximate degree to construct pattern
676 matrices $F \in \mathbb{R}^{\{0,1\}^{\lambda^m} \times ([\lambda]^m \times \{0,1\}^m)}$ of high approximate rank. To apply [Lemma 34](#), we also
677 need to find a good rectangle cover of some F .

678 A *b-certificate* for a function p on $\{0, 1\}^m$ is an assignment $\alpha : S \rightarrow \{0, 1\}$ for some
679 $S \subseteq [m]$ 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
680 of a certificate is $|S|$. The following shows how to use the certificates of p to construct a
681 rectangle cover of its pattern matrix.

► **Lemma 39.** *Let $p : \{0, 1\}^m \rightarrow \{-1, +1\}$, 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 $f : \{0, 1\}^n \rightarrow \{0, 1\}$ for $n = \ell(2\lambda)^C$ such that for
any $\kappa \in (0, 1)$ and $\varepsilon \in [\sqrt{\kappa}, 1]$:*

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

682 **Proof.** For $i = 1, \dots, \ell$, let $\alpha_i : S_i \rightarrow \{0, 1\}$ for $S_i \subseteq [m]$ of size $|S_i| \leq C$ be one of the ℓ
683 certificates. That is, for each i , there is some $v_i \in \{-1, +1\}$ such that for any $x \in \{0, 1\}^m$, if
684 $x_j = \alpha_i(j)$ for all $j \in S_i$, then $p(x) = v_i$ (so α_i is a v_i -certificate).

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

$$688 \quad X_{i,k,b} = \{(x, w) \in [\lambda]^m \times \{0, 1\}^m : \forall j \in S_i, w_j = b_j, x_j = k_j\}$$

$$689 \quad Y_{i,k,b} = \{y \in \{0, 1\}^{\lambda^m} : \forall j \in S_i, y_{k_j}^{(j)} = b_j \oplus \alpha_i(j)\}.$$

690

691 We first note that this is a rectangle cover. Fix any $y \in \{0, 1\}^{\lambda^m}$, $x \in [\lambda]^m$ and $w \in \{0, 1\}^m$.
692 First note that for any i , if we let b be the restriction of w to S_i , and k the restriction of x
693 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
694 $y|_x \oplus w$, and by assumption there is at least one such i . For such an i , we have $y_{x_j}^{(j)} \oplus w_j = \alpha_i(j)$
695 for all $j \in S_i$, so $y \in Y_{i,k,b}$. Thus, we can apply [Lemma 34](#).

696 Note that if $(x, w) \in X_{i,k,b}$, and $y \in Y_{i,k,b}$, then $(y|_x \oplus w)[j] = y_{x_j}^{(j)} \oplus w_j = \alpha_i(j)$ for all
697 $j \in S_i$, 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
698 0 else, we have that if $y \in Y_{i,k,b}$ and $(x, w) \in X_{i,k,b}$, $(F \circ \Delta_{i,k,b})[y, (x, w)] = p(y|_x \oplus w) = v_i$,
699 and otherwise, $(F \circ \Delta_{i,k,b})[y, (x, w)] = 0$. Thus $\text{rank}(F \circ \Delta_{i,k,b}) = \text{rank}(v_i \Delta_{i,k,b}) = 1$. Then
700 by [Lemma 34](#), there exists $f : \{0, 1\}^n \rightarrow \{0, 1\}$ where $n = \sum_{i=1}^{\ell} (2\lambda)^{|S_i|} \leq \ell(2\lambda)^C$ such that:

$$701 \quad \text{m}\widetilde{\text{SP}}_\kappa(f) \geq \sqrt{\kappa}\text{-rank}(F) \geq \lambda^{\widetilde{\deg}_\varepsilon(p)} \frac{(\varepsilon - \sqrt{\kappa})^2}{(1 + \sqrt{\kappa})^2}, \text{ by } \text{Lemma 38.} \quad \blacktriangleleft$$

702

703 We now prove [Theorem 32](#), restated below:

▷ **Theorem 32.** There is an explicit Boolean function $f : D \rightarrow \{0, 1\}$ for $D \subseteq \{0, 1\}^n$ such
that for any constant κ ,

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

704 **Proof.** By [5, Theorem 38], there is a function p with $\widetilde{\deg}_{1/3}(p) \geq C(p)^{2-o(1)}$, which is, up
 705 to the $o(1)$ in the exponent, the best possible separation between these two quantities. In
 706 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
 707 certificate complexity of p , for some parameter M (see [5] equations (64) and (65), where
 708 p is referred to as F), and p is a function on $M^{2+o(1)}$ variables (see [5], discussion above
 709 equation (64)). Thus, there are at most $\binom{M^{2+o(1)}}{M^{1+o(1)}}$ possible certificates of size $M^{1+o(1)}$ such
 710 that each input satisfies at least one of them.

711 Then by Lemma 39 there exists a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ for some n such that
 712 $n \leq \binom{M^{2+o(1)}}{M^{1+o(1)}} (2\lambda)^{M^{1+o(1)}}$ such that for constant $\kappa < 1/36$ and constant λ :

$$713 \log \widetilde{\text{mSP}}_{\kappa}(f) \geq \Omega(\widetilde{\deg}_{1/3}(p) \log \lambda) \geq M^{2-o(1)}.$$

715 Then we have:

$$716 \log n \leq \log \binom{M^{2+o(1)}}{M^{1+o(1)}} + M^{1+o(1)} \log(2\lambda) = O(M^{1+o(1)} \log M) = M^{1+o(1)}.$$

718 Thus, $\log \widetilde{\text{mSP}}_{\kappa}(f) \geq (\log n)^{2-o(1)}$, and the result for any κ follows using Corollary 19. ◀

719 Since for all total functions p , $\widetilde{\deg}_{1/3}(p) \leq C(p)^2$, where $C(p)$ is the certificate complexity
 720 of p , Lemma 39 can't prove a lower bound better than $\log \widetilde{\text{mSP}}(p) \geq (\log n)^2$ for any n -bit
 721 function. We state a more general version of Lemma 39 that might have the potential to
 722 prove a better bound, but we leave this as future work.

723 ► **Lemma 40.** Fix $p : \{0, 1\}^m \rightarrow \{-1, +1\}$. For $i = 1, \dots, \ell$, let $\alpha_i : S_i \rightarrow \{0, 1\}$ for $S_i \subseteq [m]$
 724 be a partial assignment such that every $z \in \{0, 1\}^m$ satisfies at least one of the assignments.
 725 Let p_i denote the restriction of p to strings z satisfying the assignment α_i . Then for every
 726 positive integer λ , there exists a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, where $n = \sum_{i=1}^{\ell} (2\lambda)^{|S_i|}$ such
 727 that for any $\kappa \in (0, 1)$ and $\varepsilon \in [\sqrt{\kappa}, 1]$:

$$728 \widetilde{\text{mSP}}_{\kappa}(f) \geq \Omega \left(\frac{(\varepsilon - \sqrt{\kappa})^2 \lambda^{\widetilde{\deg}_{\varepsilon}(p)}}{\max_{i \in [\ell]} \sum_{S \subseteq [m] \setminus S_i : \hat{p}_i(S) \neq 0} \lambda^{|S|}} \right).$$

730 To make use of this lemma, one needs a function p of high approximate degree, such that for
 731 every input, there is a small assignment that lowers the degree to something small. This
 732 generalizes Lemma 39 because a certificate is an assignment that lowers the degree of the
 733 remaining sub-function to constant. However, we note that a p with these conditions is
 734 necessary but may not be sufficient for proving a non-trivial lower bound, because while
 735 $\sum_{S: \hat{p}_i(S) \neq 0} \lambda^{|S|} \geq \lambda^{\deg(p_i)}$, it may also be much larger if p_i has a dense Fourier spectrum.

736 **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
 737 covered defined in the proof of Lemma 39, with the difference that since the α_i are no longer
 738 certificates, the resulting submatrices of F may not have constant rank.

739 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

$$740 F \circ \Delta_{i,k,b} = \sum_{y \in Y_{i,k,b}, (x,w) \in X_{i,k,b}} p(y|_x \oplus w) |y\rangle \langle x, w|.$$

742 Note that when $y \in Y_{i,k,b}$ and $(x, w) \in X_{i,k,b}$, $y|_x \oplus w$ satisfies α_i , so $p(y|_x \oplus w) = p_i(y'|_{x' \oplus w'})$,
 743 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 $[m] \setminus S_i$.

744 Thus, continuing from above, and rearranging registers, we have:

$$\begin{aligned}
 745 \quad F \circ \Delta_{i,k,b} &= \sum_{y' \in (\{0,1\}^\lambda)^{[m] \setminus S_i}} \sum_{\substack{x' \in [\lambda]^{[m] \setminus S_i}, \\ w' \in \{0,1\}^{[m] \setminus S_i}}} p_i(y'|x' \oplus w')|y'\rangle\langle x', w'| \otimes \sum_{\substack{\bar{y} \in (\{0,1\}^\lambda)^{S_i}: \\ \bar{y}|_k = b \oplus \alpha_i}} |\bar{y}\rangle\langle k, b| \\
 746 \quad &= F_i \otimes J_{2^{(\lambda-1)|S_i|}, 1} \\
 747
 \end{aligned}$$

748 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 ,
 749 which always has rank 1 for $a, b > 0$. Thus

$$750 \quad \text{rank}(F \circ \Delta_{i,k,b}) = \text{rank}(F_i) \text{rank}(J_{2^{(\lambda-1)|S_i|}, 1}) = \text{rank}(F_i) = \sum_{S \subseteq [m] \setminus S_i: \hat{p}_i(S) \neq 0} \lambda^{|S|},$$

751

752 by [20]. This part of the proof follows [19, Lemma IV.6].

753 Then by Lemma 34 and Lemma 38, we have:

$$754 \quad \text{m}\widetilde{\text{SP}}_\kappa(f) \geq \Omega\left(\frac{\sqrt{\kappa} \cdot \text{rank}(F)}{\max_{i,k,b} \text{rank}(F \circ \Delta_{i,k,b})}\right) \geq \Omega\left(\frac{\left(\frac{\varepsilon - \sqrt{\kappa}}{1 + \sqrt{\kappa}}\right)^2 \lambda^{\deg_\varepsilon(p)}}{\max_i \sum_{S \subseteq [m] \setminus S_i: \hat{p}_j(S) \neq 0} \lambda^{|S|}}\right). \quad \blacktriangleleft$$

755

756 5.2 Monotone Algorithms

757 In Theorem 32, we showed a non-trivial lower bound on $\log \text{m}\widetilde{\text{SP}}(f)$ for some explicit monotone
 758 function f . Unlike lower bounds on $\log \widetilde{\text{SP}}(f)$, this does not give us a lower bound on the
 759 quantum space complexity of f , however, at the very least it gives us a lower bound on
 760 the quantum space complexity of a certain type of quantum algorithm. Of course, this is
 761 naturally the case, since a lower bound on $\text{m}\widetilde{\text{SP}}(f)$ gives us a lower bound on the quantum
 762 space complexity of any algorithm for f that is obtained from a monotone span program.
 763 However, this is not the most satisfying characterization, as it is difficult to imagine what
 764 this class of algorithms looks like.

765 In this section, we will consider a more natural class of algorithms whose space complexity
 766 is lower bounded by $\text{m}\widetilde{\text{SP}}(f)$, and in some cases $\text{m}\text{SP}(f)$. We will call a quantum query
 767 algorithm a *phase estimation algorithm* if it works by estimating the amplitude on $|0\rangle$ in
 768 the phase register after running phase estimation of a unitary that makes one query. We
 769 assume that the unitary for which we perform phase estimation is of the form $U\mathcal{O}_x$. This
 770 is without loss of generality, because the most general form is a unitary $U_2\mathcal{O}_xU_1$, but we
 771 have $(U_2\mathcal{O}_xU_1)^t|\psi_0\rangle = U_1^\dagger(U\mathcal{O}_x)^t|\psi'_0\rangle$ where $|\psi'_0\rangle = U_1|\psi_0\rangle$, and $U = U_1U_2$. The weight on
 772 a phase of $|0\rangle$ is not affected by this global (t -independent) U_1^\dagger . Thus, we define a phase
 773 estimation algorithm as follows:

774 ► **Definition 41.** A phase estimation algorithm $\mathcal{A} = (U, |\psi_0\rangle, \delta, T, M)$ for $f : D \rightarrow \{0, 1\}$,
 775 $D \subseteq \{0, 1\}^n$, is defined by (families of):

776 ■ a unitary U acting on $\mathcal{H} = \text{span}\{|j, z\rangle : j \in [n], z \in \mathcal{Z}\}$ for some finite set \mathcal{Z} ;

777 ■ an initial state $|\psi_0\rangle \in \mathcal{H}$;

778 ■ a bound $\delta \in [0, 1/2)$;

779 ■ positive integers T and $M \leq \frac{1}{\sqrt{\delta}}$;

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

782 1. Let $\Phi(x)$ be the algorithm that runs phase estimation of $U\mathcal{O}_x$ on $|\psi_0\rangle$ for T' steps, and
 783 then computes a bit $|b\rangle_A$ in a new register A , such that $b = 0$ if and only if the phase
 784 estimate is 0.

785 2. Run M' steps of amplitude estimation to estimate the amplitude on $|0\rangle_A$ after application
786 of $\Phi(x)$. Output 0 if the amplitude is $> \delta$.

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

789 We insist that the algorithm work not only for M and T but for any larger integers as
790 well, because we want to ensure that the algorithm is successful because M and T are large
791 enough, and not by some quirk of the particular chosen values. When $\delta = 0$, the algorithm
792 has one-sided error (see Lemma 46).

793 We remark on the generality of this form of algorithm. Any algorithm can be put into
794 this form by first converting it to a span program, and then compiling that into an algorithm,
795 preserving both the time and space complexity, asymptotically. However, we will consider a
796 special case of this type of algorithm that is *not* fully general.

797 ► **Definition 42.** A monotone phase estimation algorithm is a phase estimation algorithm
798 such that if $\Pi_0(x)$ denotes the orthogonal projector onto the $(+1)$ -eigenspace of $U\mathcal{O}_x$, then
799 for any $x \in \{0, 1\}^n$, $\Pi_0(x)|\psi_0\rangle$ is in the $(+1)$ -eigenspace of \mathcal{O}_x .

800 Let us consider what is “monotone” about this definition. The algorithm rejects if $|\psi_0\rangle$
801 has high overlap with the $(+1)$ -eigenspace of $U\mathcal{O}_x$, i.e., $\Pi_0(x)|\psi_0\rangle$ is large. In a monotone
802 phase estimation algorithm, we know that the only contribution to $\Pi_0(x)|\psi_0\rangle$ is in the
803 $(+1)$ -eigenspace of \mathcal{O}_x , which is exactly the span of $|j, z\rangle$ such that $x_j = 0$. Thus, only
804 0-queries can contribute to the algorithm rejecting.

805 As a simple example, Grover’s algorithm is a monotone phase estimation algorithm.
806 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 $U\mathcal{O}_x$ is the standard
807 Grover iterate, and $|\psi_0\rangle$ is in the span of $e^{i\theta}$ -eigenvectors of $U\mathcal{O}_x$ with $\sin|\theta| = \sqrt{|x|/n}$, so
808 phase estimation can be used to distinguish the case $|x| = 0$ from $|x| \geq 1$. So $\Pi_0(x)|\psi_0\rangle$ is
809 either 0, when $|x| \neq 0$, or $|\psi_0\rangle$, when $|x| = 0$. In both cases, it is in the $(+1)$ -eigenspace of
810 \mathcal{O}_x .

811 It is clear that a monotone phase estimation algorithm can only decide a monotone
812 function. However, while any quantum algorithm can be converted to a phase estimation
813 algorithm, it is not necessarily the case that any quantum algorithm for a monotone function
814 can be turned into a monotone phase estimation algorithm. Thus lower bounds on the
815 quantum space complexity of any monotone phase estimation algorithm for f do not imply
816 lower bounds on $S_U(f)$. Nevertheless, if we let $\mathbf{mS}_U(f)$ represent the minimum quantum
817 space complexity of any monotone phase estimation algorithm for f , then a lower bound on
818 $\mathbf{mS}_U(f)$ at least tells us that if we want to compute f with space less than said bound, we
819 must use a non-monotone phase estimation algorithm.

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

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

► **Theorem 43.** 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

$$\text{mS}_U(f) \geq \log \widetilde{\text{mSP}}(f) \quad \text{and} \quad \text{mS}_U^1(f) \geq \log \text{mSP}(f).$$

829 We prove this theorem in [Section 5.2.1](#). As a corollary, lower bounds on $\text{mSP}(f)$, such as the
830 one from [\[14\]](#), imply lower bounds on $\text{mS}_U^1(f)$; and lower bounds on $\widetilde{\text{mSP}}(f)$ such as the one
831 in [Theorem 32](#), imply lower bounds on $\text{mS}_U(f)$. In particular:

832 ► **Corollary 44.** *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be the function described in [Theorem 32](#). Then*
833 *$\text{mS}_U(f) \geq (\log n)^{2-o(1)}$. Let $g : \{0, 1\}^n \rightarrow \{0, 1\}$ be the function described in [Theorem 31](#).*
834 *Then $\text{mS}_U^1(g) \geq \Omega(n)$.*

835 We emphasize that while this does not give a lower bound on the quantum space complexity
836 of f , or the one-sided quantum space complexity of g , it does show that any algorithm
837 that uses $(\log n)^c$ space to solve f with bounded error, for $c < 2$, or $o(n)$ space to solve g
838 with one-sided error, must be of a different form than that described in [Definition 41](#) and
839 [Definition 42](#).

840 In a certain sense, monotone phase estimation algorithms completely characterize those
841 that can be derived from monotone span programs, because the algorithm we obtain from
842 compiling a monotone span program is a monotone phase estimation algorithm, as stated
843 below in [Lemma 45](#). However, not all monotone phase estimation algorithms can be obtained
844 by compiling monotone span programs, and similarly, we might hope to show that an even
845 larger class of algorithms can be converted to monotone span programs, in order to give
846 more strength to lower bounds on $\text{mS}_U(f)$.

847 ► **Lemma 45.** *Let P be an approximate monotone span program for f with size S and*
848 *complexity C . Then there is a monotone algorithm for f with query complexity $O(C)$ and*
849 *space complexity $O(\log S + \log C)$.*

850 **Proof.** Fix a monotone span program, and assume it has been appropriately scaled. Without
851 loss of generality, we can let $H_j = H_{j,1} = \text{span}\{|j, z\rangle : z \in \mathcal{Z}_j\}$ for some finite set \mathcal{Z}_j .
852 Then, $\mathcal{O}_x = I - 2\Pi_{H(x)}$, which is only true because the span program is monotone. Let
853 $U = 2\Pi_{\text{row}(A)} - I$. Then $U\mathcal{O}_x = (2\Pi_{\ker(A)} - I)(2\Pi_{H(x)} - I)$ is the *span program unitary*,
854 described in [\[8\]](#). The algorithm obtained from compiling a span program works by performing
855 $O(C)$ steps of phase estimation of this unitary, applied to $|w_0\rangle = A^+|\tau\rangle$, and estimating
856 the amplitude on 0 in the phase register to constant precision (see [\[8, Lemma 3.6\]](#)). This is
857 clearly a phase estimation algorithm for f with query complexity $O(C)$ and space complexity
858 $O(\log S + \log C)$.

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

863 5.2.1 Monotone Algorithms to (Approximate) Monotone Span 864 Programs

865 In this section, we prove [Theorem 43](#). Throughout this section, we fix a phase estimation
866 algorithm $\mathcal{A} = (U, |\psi_0\rangle, \delta, T, M)$ that computes f , with U acting on \mathcal{H} . For any $x \in \{0, 1\}^n$
867 and $\Theta \in [0, \pi]$, we let $\Pi_\Theta(x)$ denote the orthogonal projector onto the span of $e^{i\theta}$ -eigenvectors
868 of $U\mathcal{O}_x$ for $|\theta| \leq \Theta$. We will let $\Pi_x = \sum_{j \in [n], z \in \mathcal{Z}: x_j=1} |j, z\rangle\langle j, z|$.

869 We begin by drawing some conclusions about the necessary relationship between the
 870 eigenspaces of $U\mathcal{O}_x$ and a function f whenever a monotone phase estimation computes f .
 871 The proofs are somewhat dry and are relegated to [Appendix B](#).

► **Lemma 46.** *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

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

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

► **Lemma 47.** *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 $d < \sqrt{8}/\pi$,

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

873 To prove [Theorem 43](#), we will define a monotone span program $P_{\mathcal{A}}$ as follows:

$$\begin{aligned} 874 \quad H_{\text{true}} &= \text{span}\{|j, z\rangle : j \in [n], z \in \mathcal{Z}\} = \mathcal{H} \\ 875 \quad H_{j,1} &= H_j = \text{span}\{|j, z, 1\rangle : z \in \mathcal{Z}\} \\ 876 \quad A|j, z, 1\rangle &= \frac{1}{2}(|j, z\rangle - (-1)^1|j, z\rangle) = |j, z\rangle \\ 877 \quad A|j, z\rangle &= (I - U^\dagger)|j, z\rangle \\ 878 \quad |\tau\rangle &= |\psi_0\rangle. \end{aligned} \tag{2}$$

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

► **Lemma 48.** *For any $x \in \{0, 1\}^n$, we have*

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

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

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, $\mathcal{O}_x \Pi_0(x) | \psi_0 \rangle = \Pi_0(x) | \psi_0 \rangle$, and so $\mathcal{O}_x | \omega \rangle = | \omega \rangle$, and thus $\Pi_x | \omega \rangle = 0$, where Π_x is the

projector onto $|j, z\rangle$ such that $x_j = 1$. Note that $H(x) = \text{span}\{|j, z, 1\rangle : x_j = 1, z \in \mathcal{Z}\} \oplus \text{span}\{|j, z\rangle : j \in [n], z \in \mathcal{Z}\}$. Thus $\Pi_{H(x)} = \Pi_{H_{\text{true}}} + \Pi_x \otimes |1\rangle\langle 1|$. 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 $U\mathcal{O}_x$, we have $U\mathcal{O}_x|\omega\rangle = |\omega\rangle$ so since $\mathcal{O}_x|\omega\rangle = |\omega\rangle$, $U|\omega\rangle = |\omega\rangle$. Thus

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

884 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, $\mathcal{O}_x|\omega\rangle = (I - 2\Pi_x)|\omega\rangle = |\omega\rangle$, and since $\langle \omega | A\Pi_{H_{\text{true}}} = \langle \omega | (I - U^\dagger)$ must be 0, $U|\omega\rangle = |\omega\rangle$. Thus $|\omega\rangle$ is a 1-eigenvector of $U\mathcal{O}_x$, so

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

885 We complete the proof by noticing that since $\langle \omega | A\Pi_{H_{\text{true}}} = 0$, we have $\langle \omega | A = \langle \omega | \langle 1|$, and
886 $w_-(x) = \|\langle \omega | A\|^2 = \|\omega\|^2$. \blacktriangleleft

887 Next we find approximate positive witnesses.

888 **► Lemma 49.** *For any $\Theta \geq 0$, the span program $P_{\mathcal{A}}$ has approximate positive witnesses for
889 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:

$$|v\rangle = (I - (U\mathcal{O}_x)^\dagger)^+(I - \Pi_\Theta(x))|\psi_0\rangle.$$

890 Note that $I - (U\mathcal{O}_x)^\dagger$ is supported everywhere except the $(+1)$ -eigenvectors of $(U\mathcal{O}_x)^\dagger$,
891 which are exactly the $(+1)$ -eigenvectors of $U\mathcal{O}_x$. Thus, $(I - \Pi_\Theta(x))|\psi_0\rangle$ is contained in this
892 support.

893 Next we define $|w\rangle = (|\psi_0\rangle - (I - U^\dagger)|v\rangle) |1\rangle + |v\rangle$. Then we have:

$$894 \quad A|w\rangle = |\psi_0\rangle - (I - U^\dagger)|v\rangle + (I - U^\dagger)|v\rangle = |\psi_0\rangle = |\tau\rangle.$$

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

$$897 \quad \begin{aligned} \|\Pi_{H(x)^\perp}|w\rangle\|^2 &= \|\Pi_{\bar{x}}(|\psi_0\rangle - (I - U^\dagger)|v\rangle)\|^2 \\ 898 \quad &= \|\Pi_{\bar{x}}|\psi_0\rangle - \Pi_{\bar{x}}(I - U^\dagger)(I - (U\mathcal{O}_x)^\dagger)^+(I - \Pi_\Theta(x))|\psi_0\rangle\|^2. \end{aligned}$$

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

$$\Pi_{\bar{x}}(I - \mathcal{O}_x U^\dagger) = \Pi_{\bar{x}}(\Pi_{\bar{x}} - (\Pi_{\bar{x}} - \Pi_x)U^\dagger) = \Pi_{\bar{x}}(I - U^\dagger).$$

900 Thus, continuing from above, we have:

$$901 \quad \begin{aligned} \|\Pi_{H(x)^\perp}|w\rangle\|^2 &= \|\Pi_{\bar{x}}|\psi_0\rangle - \Pi_{\bar{x}}(I - \mathcal{O}_x U^\dagger)(I - \mathcal{O}_x U^\dagger)^+(I - \Pi_\Theta(x))|\psi_0\rangle\|^2 \\ 902 \quad &= \|\Pi_{\bar{x}}|\psi_0\rangle - \Pi_{\bar{x}}(I - \Pi_\Theta(x))|\psi_0\rangle\|^2 = \|\Pi_{\bar{x}}\Pi_\Theta(x)|\psi_0\rangle\|^2 \leq \|\Pi_\Theta(x)|\psi_0\rangle\|^2. \end{aligned}$$

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

$$906 \quad (I - (U\mathcal{O}_x)^\dagger)^+ = \sum_{j:\theta_j \neq 0} \frac{1}{1 - e^{-i\theta_j}} |\lambda_j\rangle\langle \lambda_j| \quad \text{and} \quad I - \Pi_\Theta(x) = \sum_{j:|\theta_j| > \Theta} |\lambda_j\rangle\langle \lambda_j|.$$

907

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

$$\begin{aligned}
 909 \quad \|v\|^2 &= \|(I - (U\mathcal{O}_x)^\dagger)^+(I - \Pi_\Theta(x))|\psi_0\rangle\|^2 = \left\| \sum_{j:|\theta_j|>\Theta} \frac{1}{1 - e^{-i\theta_j}} \langle \lambda_j | \psi_0 \rangle | \lambda_j \rangle \right\|^2 \\
 910 \quad &= \sum_{j:|\theta_j|>\Theta} \frac{1}{4 \sin^2 \frac{\theta_j}{2}} |\langle \lambda_j | \psi_0 \rangle|^2 \leq \frac{\pi^2}{4\Theta^2}. \\
 911
 \end{aligned}$$

912 Next, using $\mathcal{O}_x + 2\Pi_x = I - 2\Pi_x + 2\Pi_x = I$, we compute:

$$\begin{aligned}
 913 \quad &\| |\psi_0\rangle - (I - U^\dagger)v \|^2 \\
 914 \quad &= \| |\psi_0\rangle - (I - \mathcal{O}_x U^\dagger - 2\Pi_x U^\dagger)(I - \mathcal{O}_x U^\dagger)^+(I - \Pi_\Theta(x))|\psi_0\rangle \|^2 \\
 915 \quad &= \| |\psi_0\rangle - (I - \Pi_\Theta(x))|\psi_0\rangle + 2\Pi_x U^\dagger (I - (U\mathcal{O}_x)^\dagger)^+(I - \Pi_\Theta(x))|\psi_0\rangle \|^2 \\
 916 \quad &\leq \left(\| \Pi_\Theta(x)|\psi_0\rangle \| + 2 \left\| \Pi_x U^\dagger \sum_{j:|\theta_j|>\Theta} \frac{1}{1 - e^{-i\theta_j}} \langle \lambda_j | \psi_0 \rangle | \lambda_j \rangle \right\| \right)^2 \\
 917 \quad &\leq \left(\| \Pi_\Theta(x)|\psi_0\rangle \| + 2 \sqrt{\sum_{j:|\theta_j|>\Theta} \frac{1}{4 \sin^2 \frac{\theta_j}{2}} |\langle \lambda_j | \psi_0 \rangle|^2} \right)^2 \\
 918 \quad &\leq \left(\| \Pi_\Theta(x)|\psi_0\rangle \| + \frac{\pi}{\Theta} \| (I - \Pi_\Theta(x))|\psi_0\rangle \| \right)^2 \leq \frac{\pi^2}{\Theta^2}. \\
 919
 \end{aligned}$$

920 Then we have the complexity of $|w\rangle$:

$$921 \quad \|w\|^2 = \| |\psi_0\rangle - (I - U^\dagger)v \|^2 + \|v\|^2 \leq \frac{\pi^2}{\Theta^2} + \frac{\pi^2}{4\Theta^2} = \frac{5\pi^2}{4\Theta^2}. \quad \blacktriangleleft$$

923 We conclude with the following two corollaries, whose combination gives [Theorem 43](#).

924 **► Corollary 50.** *Let $\mathcal{A} = (U, |\psi_0\rangle, 0, T, M)$ be a monotone phase estimation algorithm for*
 925 *f with space complexity $S = \log \dim \mathcal{H} + \log T + \log M + 1$ and query complexity $O(TM)$.*
 926 *Then there is a monotone span program that decides f (exactly) whose size is $2 \dim \mathcal{H} \leq 2^S$*
 927 *and whose complexity is $O(TM)$.*

928 **Proof.** If $f(x) = 0$, then by [Lemma 46](#), we have $\| \Pi_0(x)|\psi_0\rangle \|^2 \geq \frac{1}{M^2}$, so by [Lemma 48](#),
 929 $w_-(x) \leq M^2$. Thus $W_- \leq M^2$.

930 If $f(x) = 1$, then by [Lemma 46](#), we have $\| \Pi_{2/T}(x)|\psi_0\rangle \|^2 = 0$, so by [Lemma 49](#), there's
 931 an exact positive witness for x with complexity $O(T^2)$. Thus $W_+ \leq O(T^2)$, and so the
 932 span program $P_{\mathcal{A}}$ from (2) has complexity $O(TM)$. The size of the span program $P_{\mathcal{A}}$ is
 933 $\dim H = 2 \dim \mathcal{H}$. ◀

934 **► Corollary 51.** *Let $\mathcal{A} = (U, |\psi_0\rangle, \delta, T, M)$ be a monotone phase estimation algorithm for*
 935 *f with space complexity $S = \log \dim \mathcal{H} + \log T + \log M + 1$ and query complexity $O(TM)$.*
 936 *Then there is a constant $\kappa \in (0, 1)$ such that there exists a monotone span program that*
 937 *κ -approximates f whose size is $2 \dim \mathcal{H} \leq 2^S$ and whose complexity is $O(TM)$.*

938 **Proof.** If $f(x) = 0$, then by [Lemma 47](#), we have $\| \Pi_0(x)|\psi_0\rangle \|^2 > \delta(1+c)$ for some constant
 939 $c > 0$. Thus, by [Lemma 48](#), $W_- \leq \frac{1}{(1+c)\delta}$.

If $f(x) = 1$, then by Lemma 49, setting $\Theta = d\pi/T$ for $d = \frac{2}{\pi}\sqrt{\frac{c}{1+c}}$, (where c is the constant from above), by Lemma 49 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 47, we have

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

940 Thus, letting $\kappa = \frac{1}{1+c/2} < 1$, we have that $P_{\mathcal{A}}$ κ -approximates f . Since the positive witness
 941 complexity is $O(T^2)$, and by Lemma 47, we also have $W_- \leq O(M^2)$, the complexity of $P_{\mathcal{A}}$
 942 is $O(TM)$. The size of $P_{\mathcal{A}}$ is $\dim H = 2 \dim \mathcal{H}$. ◀

943 Acknowledgements

944 I am grateful to Tsuyoshi Ito for discussions that led to the construction of approximate span
 945 programs from two-sided error quantum algorithms presented in Section 3.2, and to Alex
 946 B. Grilo and Mario Szegedy for insightful comments. I thank Robin Kothari for pointing out
 947 the improved separation between certificate complexity and approximate degree in [5], which
 948 led to an improvement in from $(\log n)^{7/6}$ (using [1]) to $(\log n)^{2-o(1)}$ in Theorem 32.

949 — References —

- 950 1 S. Aaronson, S. Ben-David, and R. Kothari. Separations in query complexity using cheat
 951 sheets. In *Proceedings of the forty-eighth annual ACM Symposium on Theory of Computing*
 952 (*STOC 2016*), pages 863–876, 2016. [arXiv:1511.01937](#).
- 953 2 S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University
 954 Press, 2009.
- 955 3 L. Babai, A. Gál, and A. Wigderson. Superpolynomial lower bounds for monotone span
 956 programs. *Combinatorica*, 19:301–319, 1999.
- 957 4 G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and
 958 estimation. In S. J. Lomonaca and H. E. Brandt, editors, *Quantum Computation and Quantum*
 959 *Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*
 960 *Millennium Volume*, pages 53–74. AMS, 2002. [arXiv:quant-ph/0005055v1](#).
- 961 5 M. Bun and J. Thaler. A nearly optimal lower bound on the approximate degree of AC^0 . In
 962 *Proceedings of the IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS*
 963 *2017)*, 2017. [arXiv:1703.05784](#).
- 964 6 B. Fefferman and C. Lin. A complete characterization of unitary quantum space. In *Proceedings*
 965 *of the 2018 ACM Conference on Innovations in Theoretical Computer Science (ITCS 2018)*,
 966 pages 4:1–4:21, 2018. [arXiv:1604.01384](#).
- 967 7 A. Gál. A characterization of span program size and improved lower bounds for monotone
 968 span programs. *Computational Complexity*, 10(4):277–296, 2001.
- 969 8 T. Ito and S. Jeffery. Approximate span programs. *Algorithmica*, 81(6):2158–2195, 2019.
 970 [arXiv:1507.00432](#).
- 971 9 R. Jozsa, B. Kraus, A. Miyake, and J. Watrous. Matchgate and space-bounded quantum
 972 computations are equivalent. *Proceedings of the Royal Society A*, 466(2115), 2009. [doi:http://doi.org/10.1098/rspa.2009.0433](http://doi.org/10.1098/rspa.2009.0433).
- 974 10 M. Karchmer and A. Wigderson. On span programs. In *Proceedings of the IEEE 8th Annual*
 975 *Conference on Structure in Complexity Theory*, pages 102–111, 1993.
- 976 11 A. Kitaev. Quantum measurements and the Abelian stabilizer problem, 1995.
 977 [arXiv:quant-ph/9511026](#).

- 978 12 T. Lee, R. Mittal, B. Reichardt, R. Špalek, and M. Szegedy. Quantum query complexity of
 979 state conversion. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of*
 980 *Computer Science (FOCS 2011)*, pages 344–353, 2011.
- 981 13 S. V. Lokam. *Complexity Lower Bounds using Linear Algebra*. Now Publishers Inc., Hanover,
 982 MA, USA, 2009. doi:10.1561/0400000011.
- 983 14 T. Pitassi and R. Robere. Strongly exponential lower bounds for monotone computation. In
 984 *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC*
 985 *2017)*, pages 1246–1255, 2017.
- 986 15 A. A. Razborov. Applications of matrix methods to the theory of lower bounds in computational
 987 complexity. *Combinatorica*, 10(1):810093, 1990.
- 988 16 A. A. Razborov. On submodular complexity measures. In *Pocceedings of the London Mathem-*
 989 *atical Society symposium on Boolean function complexity*, pages 76 – 83, 1992.
- 990 17 B. Reichardt. Span programs and quantum query complexity: The general adversary bound is
 991 nearly tight for every Boolean function. In *Proceedings of the 50th IEEE Symposium on Found-*
 992 *ations of Computer Science (FOCS 2009)*, pages 544–551, 2009. arXiv:quant-ph/0904.2759.
- 993 18 B. Reichardt and R. Špalek. Span-program-based quantum algorithm for evaluating formulas.
 994 *Theory of Computing*, 8(13):291–319, 2012.
- 995 19 R. Robere, T. Pitassi, B. Rossman, and S. A. Cook. Exponential lower bounds for monotone
 996 span programs. In *Proceedings of the 57th IEEE Symposium on Foundations of Computer*
 997 *Science (FOCS 2016)*, pages 406–415, 2016. doi:10.1109/FOCS.2016.51.
- 998 20 Alexander A. Sherstov. The pattern matrix method. *SIAM Journal on Computing*,
 999 40(6):1969–2000, 2009.
- 1000 21 J. Watrous. Space-bounded quantum complexity. *Journal of Computer and System Sciences*,
 1001 59(2):281–326, 1999.

1002 A Proof of Claim 18

1003 In this section, we prove Claim 18, restated below:

1004 ▷ **Claim 18.** Let P be a span program that κ -approximates $f : D \rightarrow \{0, 1\}$ for some constant
 1005 κ . For any constant $\kappa' \leq \kappa$, there exists a span program P' that κ' -approximates f with
 1006 $s(P') = (s(P) + 2)^{\frac{\log \frac{\kappa'}{\kappa}}{\log \frac{1}{\kappa}}}$, and $C_{\kappa'}(P', D) \leq O(C_{\kappa}(P, D))$.

1007 Let $|w_0\rangle = A^+|\tau\rangle$. We say a span program is *normalized* if $\| |w_0\rangle \| = 1$. A span program
 1008 can easily be normalized by scaling $|\tau\rangle$, which also scales all positive witnesses and inverse
 1009 scales all negative witnesses. However, we sometimes want to normalize a span program,
 1010 while also keeping all negative witness sizes bounded by a constant. We can accomplish this
 1011 using the following construction, from [8].

► **Theorem 52.** 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 :

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

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

1012 Then we have the following:

- 1013 ■ $\| (A^\beta)^+ |\tau^\beta\rangle \| = 1$;
- 1014 ■ for all $x \in P_1$, $w_+(x, P^\beta) = \frac{1}{\beta^2} w_+(x, P) + 2$;
- 1015 ■ for all $x \in P_0$, $w_-(x, P^\beta) = \beta^2 w_-(x, P) + 1$.

4:32 Span Programs and Quantum Space Complexity

1016 ► **Corollary 53.** Let P be a span program on $\{0,1\}^n$, and P^β be defined as above for
 1017 $\beta = \frac{1}{\sqrt{W_-(P)}}$. If P κ -approximates f , then P^β $\sqrt{\kappa}$ -approximates f , with $W_-(P^\beta) \leq 2$,
 1018 $\widehat{W}_+(P^\beta) \leq W_-(P)\widehat{W}_+(P) + 2$ and $s(P^\beta) \leq s(P) + 2$.

Proof. First note that by [Theorem 52](#), $W_-(P^\beta) \leq 2$. Let $|w\rangle$ be an approximate positive witness for x in P , with $\|\Pi_{H(x)^\perp}|w\rangle\|^2 \leq \frac{\kappa}{W_-(P)}$ and $\| |w\rangle \|^2 \leq \widehat{W}_+(P)$. Define

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

1019 One can check that $A^\beta|w'\rangle = |\tau^\beta\rangle$.

$$\begin{aligned} 1020 \quad \|\Pi_{H^\beta(x)^\perp}|w'\rangle\|^2 &= \frac{1}{\beta^2(1+\kappa)^2} \|\Pi_{H(x)^\perp}|w\rangle\|^2 + \frac{\kappa^2}{(1+\kappa)^2} \\ 1021 \quad &\leq \frac{1}{\beta^2(1+\kappa)^2} \frac{\kappa}{W_-(P)} + \frac{\kappa^2}{(1+\kappa)^2} \\ 1022 \quad &= \frac{\kappa + \kappa^2}{(1+\kappa)^2} \leq \frac{2\kappa(1+\kappa)}{W_-(P^\beta)(1+\kappa)^2} = \frac{1}{W_-(P^\beta)} \frac{2\kappa}{1+\kappa} \leq \frac{\sqrt{\kappa}}{W_-(P^\beta)}, \\ 1023 \end{aligned}$$

1024 where we have used $W_-(P^\beta) \leq 2$. We upper bound $\widehat{W}_+(P^\beta)$ by noting that:

$$1025 \quad \||w'\rangle\|^2 \leq \frac{1}{\beta^2(1+\kappa)^2} \widehat{W}_+(P) + \frac{\beta^2}{\beta^2+N} + \frac{\kappa^2}{(1+\kappa)^2} \leq W_-(P)\widehat{W}_+(P) + 2.$$

1027 Finally, $s(P^\beta) = s(P) + 2$ because of the two extra degrees of freedom $|\hat{0}\rangle$ and $|\hat{1}\rangle$. ◀

1028 **Proof of Claim 18.** We will first show how, given a span program P such that $\| |w_0\rangle \|^2 \leq 1$,
 1029 and P κ -approximates f , we can get a span program P' such that $\| |w'_0\rangle \|^2 \leq 1$, $W_-(P') \leq$
 1030 $W_-(P)^2$, 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 $|u\rangle, |v\rangle \in H$:

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

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

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

1031 Note that $A'(|w_0\rangle|w_0\rangle) = |\tau'\rangle$, so $\| |A'^+|\tau'\rangle \| \leq \| |w_0\rangle|w_0\rangle \| \leq 1$.

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

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

1032 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

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

1033 We have

$$\begin{aligned}
 1034 \quad A'|w'\rangle &= A|w\rangle A|w\rangle - \frac{1}{2} (A\Pi_{H(x)}|w\rangle \otimes A\Pi_{\ker(A)}|w\rangle + A\Pi_{\ker(A)}|w\rangle \otimes A\Pi_{H(x)}|w\rangle) \\
 1035 \quad &= |\tau\rangle|\tau\rangle = |\tau'\rangle. \\
 1036
 \end{aligned}$$

1037 We can bound the error as:

$$\begin{aligned}
 1038 \quad \|\Pi_{H'(x)^\perp}|w'\rangle\|^2 &= \|(\Pi_{H(x)^\perp} \otimes I)|w'\rangle\|^2 = \|\Pi_{H(x)^\perp}|w\rangle|w\rangle - \Pi_{H(x)^\perp}|w\rangle\Pi_{H(x)}|w\rangle\|^2 \\
 1039 \quad &= \|\Pi_{H(x)^\perp}|w\rangle\Pi_{H(x)^\perp}|w\rangle\|^2 \leq \frac{\kappa^2}{W_-(P)^2} \leq \frac{\kappa^2}{W_-(P')}. \\
 1040
 \end{aligned}$$

1041 Next, observe that

$$\begin{aligned}
 1042 \quad &(\Pi_{H(x)} + \Pi_{H(x)^\perp}) \otimes (\Pi_{H(x)} + \Pi_{H(x)^\perp}) - \Pi_{H(x)^\perp} \otimes \Pi_{H(x)} + \Pi_{H(x)} \otimes \Pi_{H(x)^\perp} \\
 1043 \quad &= \Pi_{H(x)} \otimes \Pi_{H(x)} + \Pi_{H(x)} \otimes \Pi_{H(x)^\perp} + \Pi_{H(x)^\perp} \otimes \Pi_{H(x)} + \Pi_{H(x)^\perp} \otimes \Pi_{H(x)^\perp} \\
 1044 \quad &= \Pi_{H(x)} \otimes I + I \otimes \Pi_{H(x)^\perp}
 \end{aligned}$$

$$1045 \quad \text{so } |w'\rangle = \Pi_{H(x)}|w\rangle \otimes |w\rangle + |w\rangle \otimes \Pi_{H(x)^\perp}|w\rangle - \Pi_{H(x)}|w\rangle \otimes \Pi_{\ker(A)}|w\rangle.$$

1047 Thus, using the assumption $\|w_0\| \leq 1$, and the fact that $\Pi_{\text{row}(A)}|w\rangle = |w_0\rangle$:

$$\begin{aligned}
 1048 \quad \||w'\rangle\|^2 &= \|\Pi_{H(x)}|w\rangle|w\rangle + |w\rangle\Pi_{H(x)^\perp}|w\rangle - \Pi_{H(x)}|w\rangle\Pi_{\ker(A)}|w\rangle\|^2 \\
 1049 \quad &= \|\Pi_{H(x)}|w\rangle\Pi_{\text{row}(A)}|w\rangle + |w\rangle\Pi_{H(x)^\perp}|w\rangle\|^2 \\
 1050 \quad &= \|\Pi_{H(x)}|w\rangle|w_0\rangle\|^2 + \||w\rangle\Pi_{H(x)^\perp}|w\rangle\|^2 + 2\|\Pi_{H(x)}|w\rangle\|^2 \langle w_0|\Pi_{H(x)^\perp}|w\rangle \\
 1051 \quad &\leq \widehat{W}_+(P) + \widehat{W}_+(P) \frac{\kappa}{W_-(P)} + 2\widehat{W}_+(P) \sqrt{\frac{\kappa}{W_-(P)}} \leq (1 + \kappa + 2\sqrt{\kappa})\widehat{W}_+(P). \\
 1052
 \end{aligned}$$

1053 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 52](#) and [Corollary 53](#), we can get a span program, P_0 , with $\|w_0\| = 1$, $W_-(P_0) \leq 2$, $\widehat{W}_+(P_0) \leq C(P)^2 + 2$, 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

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

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

1054 Setting $d = \log\left(\frac{\log \frac{1}{\kappa'}}{\log \frac{1}{\kappa}}\right) + 1$ gives the desired κ' . ◀

1055 **B** Proofs of [Lemma 46](#) and [Lemma 47](#)

1056 We will prove the lemmas as a collection of claims. Fix $T' \geq T$ and $M' \geq M$ with which to
 1057 run the algorithm. Suppose $\Phi(x)$ outputs $|\psi(x)\rangle = \sqrt{p_x}|0\rangle_A |\Phi_0(x)\rangle + \sqrt{1-p_x}|1\rangle_A |\Phi_1(x)\rangle$,
 1058 and let \tilde{p} denote the estimate output by the algorithm. We will let $U\mathcal{O}_x = \sum_j e^{i\sigma_j(x)} |\lambda_j^x\rangle \langle \lambda_j^x|$
 1059 be an eigenvalue decomposition.

1060 ▷ **Claim 54.** If $f(x) = 0$ then $\|\Pi_0(x)|\psi_0\rangle\|^2 \geq \frac{1}{M^2}$.

4:34 Span Programs and Quantum Space Complexity

1061 **Proof.** Since the algorithm computes f with bounded error, the probability of accepting x is
 1062 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 $p_x = \sin^2 \theta_x$, $\theta_x \in [0, \pi/2)$ [4]. 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

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

1063 If $M'\theta_x \leq \frac{\pi}{2}$, then this would give $\frac{1}{3} \geq \frac{4}{\pi^2}$, which is a contradiction. Thus, we have:

$$1064 \quad M'\theta_x > \frac{\pi}{2} \quad \Rightarrow \quad \frac{2\theta_x}{\pi} > \frac{1}{M'} \quad \Rightarrow \quad \sin \theta_x > \frac{1}{M'} \quad \Rightarrow \quad \sqrt{p_x} > \frac{1}{M'}.$$

1065
 1066 Since $\Phi(x)$ is the result of running phase estimation, we have

$$1067 \quad p_x = \sum_j |\langle \lambda_j^x | \psi_0 \rangle|^2 \frac{\sin^2(T'\sigma_j(x)/2)}{T'^2 \sin^2(\sigma_j(x)/2)} \leq \|\Pi_\Theta(x)|\psi_0\rangle\|^2 + \frac{\pi^2}{T'^2 \Theta^2},$$

1068

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

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

1069 This is true for any choices $T' \geq T$ and $M' \geq M$, so we must have:

$$1070 \quad \frac{1}{M^2} \leq \|\Pi_0(x)|\psi_0\rangle\|^2. \quad \blacktriangleleft$$

1071

1072 \triangleright **Claim 55.** If $f(x) = 1$ and $\delta = 0$, then for any $d < \frac{\sqrt{8}}{\pi}$, $\|\Pi_{d\pi/T}(x)|\psi_0\rangle\|^2 = 0$.

1073 **Proof.** Suppose towards a contradiction that $\|\Pi_{d\pi/T}(x)|\psi_0\rangle\|^2 > 0$. Then $p_x > 0$, and some
 1074 sufficiently large $M' \geq M$ would detect this and cause the algorithm to output 0, so we must
 1075 actually have $\|\Pi_{d\pi/T}(x)|\psi_0\rangle\|^2 = 0$. In fact, in order to sure that no large enough value M'
 1076 detects amplitude > 0 on $|0\rangle_A$, we must have $p_x = 0$ whenever $f(x) = 1$. That means that
 1077 when $f(x) = 1$, the algorithm never outputs 0, so the algorithm has one-sided error. \blacktriangleleft

1078 \triangleright **Claim 56.** There is some constant c such that if $f(x) = 0$ and $\delta > 0$ then $\|\Pi_0(x)|\psi_0\rangle\|^2 >$
 1079 $\delta(1+c)$.

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 $\lfloor \frac{3\tau M'}{\pi} \rfloor = 0 \pmod 3$. 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 $\ell > 0$,

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

1080 so there must be one $\ell \in \{0, \dots, 6\}$ such that $\lfloor \frac{3\tau}{\pi} M_\ell \rfloor = 0 \pmod 3$. In particular, there is
 1081 some choice M_ℓ satisfying the condition such that (using some $M' \leq \frac{1}{\sqrt{\delta}}$):

$$1082 \quad \sqrt{\delta} M_\ell \leq \sqrt{\delta} \left(\frac{1}{\sqrt{\delta}} + 6 \frac{\pi}{6\tau} \right) = 1 + \frac{\pi \sin \tau}{\tau} \leq 1 + \pi. \quad (3)$$

1083

1084 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:

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

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

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

1085 Thus:

$$\begin{aligned} 1086 \quad & \frac{(d+1/3)\pi}{M'} + \frac{2\pi}{3M'} \leq \theta_x + \frac{\pi}{2M'} \\ 1087 \quad & \sin\left(\frac{(d+1/3)\pi}{M'} + \frac{\pi}{6M'}\right) \leq \sin \theta_x \\ 1088 \quad & \sin\left(\frac{(d+1/3)\pi}{M'}\right) \cos \frac{\pi}{6M'} + \cos\left(\frac{(d+1/3)\pi}{M'}\right) \sin \frac{\pi}{6M'} \leq \sqrt{p_x} \\ 1089 \quad & \sqrt{\delta} \sqrt{1 - \sin^2 \frac{\pi}{6M'}} + \sqrt{1 - \delta} \sin \frac{\pi}{6M'} \leq \sqrt{p_x} \end{aligned}$$

1090
1091

1092 When $\sin^2 \frac{\pi}{6M'} \leq 1 - \delta$, which we can assume, the above expression is minimized when
1093 $\sin^2 \frac{\pi}{6M'}$ is as small as possible. We have, using $M' \leq \frac{1+\pi}{\sqrt{\delta}}$, from (3):

$$1094 \quad \sin^2 \frac{\pi}{6M'} \geq \frac{4}{36M'^2} \geq \frac{\delta}{9(1+\pi)^2}.$$

1096 Thus, continuing from above, letting $k = \frac{1}{9(1+\pi)^2}$, we have:

$$\begin{aligned} 1097 \quad & \sqrt{\delta} \sqrt{1 - k\delta} + \sqrt{1 - \delta} \sqrt{k\delta} \leq \sqrt{p_x} \\ 1098 \quad & \delta(1 - k\delta) + (1 - \delta)k\delta + 2\delta \sqrt{k(1 - \delta)(1 - k\delta)} \leq p_x \end{aligned}$$

1099
1100

1101 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,
1102 using $k < 1$ and $\delta \leq 1/2$:

$$\begin{aligned} 1103 \quad & \delta(1 + k(1 - 2\delta) + 2\sqrt{k}\sqrt{(1 - k/2)(1 - 1/2)}) \leq p_x \\ 1104 \quad & \delta(1 + 0 + \sqrt{k}) \leq p_x. \end{aligned}$$

1106 Since $\Phi(x)$ is the result of running phase estimation of $U\mathcal{O}_x$ for $T' \geq T$ steps, we have:

$$1107 \quad p_x = \sum_j |\langle \lambda_j^x | \psi_0 \rangle|^2 \frac{\sin^2(\frac{T'\sigma_j(x)}{2})}{(T')^2 \sin^2(\frac{\sigma_j(x)}{2})},$$

1109 so in particular, for any $\Theta \in [0, \pi)$, we have

$$\begin{aligned} 1110 \quad & p_x \leq \|\Pi_{\Theta}(x)|\psi_0\rangle\|^2 + \sum_{j:|\sigma_j(x)|>\Theta} |\langle \lambda_j^x | \psi_0 \rangle|^2 \frac{1}{(T')^2 \sin^2(\frac{\Theta}{2})} \\ 1111 \quad & \leq \|\Pi_{\Theta}(x)|\psi_0\rangle\|^2 + \|(I - \Pi_{\Theta}(x))|\psi_0\rangle\|^2 \frac{\pi^2}{(T')^2 \Theta^2}. \end{aligned}$$

1112

4:36 Span Programs and Quantum Space Complexity

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

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

1113 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
 1114 by letting $c = \sqrt{k}$. ◀

1115 **Claim 57.** If $f(x) = 1$ and $\delta > 0$ then $\|\Pi_{d\pi/T}(x)|\psi_0\rangle\|^2 (1 - d^2\pi^2/8) \leq \delta$.

1116 **Proof.** If $|\lambda\rangle$ is an $e^{i\theta}$ -eigenvector of $U\mathcal{O}_x$ for some $|\theta| \leq d\pi/T < \sqrt{8}/T$, then the probability
 1117 of measuring 0 in the phase register upon performing T steps of phase estimation is:

$$1118 \quad p_x(\theta) := \frac{1}{T^2} \left| \sum_{t=0}^{T-1} e^{it\theta} \right|^2 = \frac{\sin^2 \frac{T\theta}{2}}{T^2 \sin^2 \frac{\theta}{2}}.$$

1119

1120 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
 1121 $\varepsilon(x) \in [0, 1]$ for any x . So we have:

$$1122 \quad p_x(\theta) \geq \frac{(T\theta/2)^2(1 - \varepsilon(T\theta/2))}{T^2(\theta/2)^2(1 - \varepsilon(\theta/2))} \geq 1 - \varepsilon(T\theta/2) \geq 1 - \frac{T^2\theta^2}{8}.$$

1123

1124 Thus, we conclude that

$$1125 \quad p_x \geq \|\Pi_{d\pi/T}(x)|\psi_0\rangle\|^2 \left(1 - \frac{T^2 d^2\pi^2}{8}\right) = \|\Pi_{d\pi/T}(x)|\psi_0\rangle\|^2 \left(1 - \frac{d^2\pi^2}{8}\right).$$

1126

1127 If this is $> \delta$, then with some sufficiently large $M' \geq M$, amplitude estimation would detect
 1128 this and cause the algorithm to output 0 with high probability. ◀