Space-Bounded Quantum Complexity*

John Watrous†
Computer Sciences Department
University of Wisconsin
Madison, Wisconsin 53706

* Appears in *Journal of Computer and System Sciences*, 59(2):281–326, 1999. A preliminary version of this paper appeared in the *Proceedings of the 13th Annual IEEE Conference on Computational Complexity*, pages 210–227, 1998.

† Supported in part by NSF grant CCR-95-10244.

Abstract

This paper investigates the computational power of space-bounded quantum Turing machines. The following facts are proved for space-constructible space bounds s satisfying $s(n) = \Omega(\log n)$.

1. Any quantum Turing machine (QTM) running in space s can be simulated by an unbounded error probabilistic Turing machine (PTM) running in space $O(s)$. No assumptions on the probability of error or running time for the QTM are required, although it is assumed that all transition amplitudes of the QTM are rational.
2. Any PTM that runs in space s and halts absolutely (i.e., has finite worst-case running time) can be simulated by a QTM running in space $O(s)$. If the PTM operates with bounded error, then the QTM may be taken to operate with bounded error as well, although the QTM may not halt absolutely in this case. In the case of unbounded error, the QTM may be taken to halt absolutely.

We therefore have that unbounded error, space $O(s)$ bounded quantum Turing machines and probabilistic Turing machines are equivalent in power, and furthermore that any QTM running in space s can be simulated deterministically in $\mathrm{NC}^2(2^s) \subseteq \mathrm{DSPACE}(s^2) \cap \mathrm{DTIME}\left(2^{O(s)}\right)$.

We also consider quantum analogues of nondeterministic and one-sided error probabilistic space-bounded classes, and prove some simple facts regarding these classes.

1 Introduction

Within the past several years, a number of researchers have provided compelling evidence suggesting that quantum computers may be considerably more powerful, in the context of time-bounded computation, than classical (probabilistic) computers (see [4, 5, 10, 13, 23, 24], for instance). In this paper, we investigate the computational power of quantum computers when space, rather than time, is the resource of primary concern. In particular, we define various quantum complexity classes analogous to traditionally studied space-bounded probabilistic classes, and prove a number of relationships among these quantum and classical classes.

The model for quantum computation we use is the quantum Turing machine (QTM), first formally defined by Deutsch [9] (see also [4, 27]). Specifically, we use a multitape version of this model; in addition to having a read-only input tape and read/write work tape, QTMs in this paper also have an output tape that is assumed to be observed after each and every computation step. This variant of the QTM model is well-suited to the study of space-bounded computation since we may consider not only machines with sublinear space bounds, but also machines with rather weak conditions on halting times. We restrict our attention to QTMs having rational transition amplitudes—some of our proofs rely on this restriction, leaving open a number of interesting questions regarding QTMs with irrational transition amplitudes.

We first consider probabilistic simulations of space-bounded quantum machines. It is proved that any unbounded error QTM running in space s, for $s(n) = \Omega(\log n)$ space-constructible, can be simulated by an unbounded error PTM running in space $O(s)$. Our proof of this fact is based on a technique previously used in the probabilistic case (e.g., in [1, 16]); the problem of determining if a given quantum machine accepts with probability exceeding $1/2$ is reduced to the problem of comparing determinants of integer matrices. From this fact, we conclude that any QTM running in space s, even in the case of unbounded error and with no restrictions on running time, can be simulated deterministically in $\mathrm{NC}^2(2^s) \subseteq \mathrm{DSPACE}(s^2) \cap \mathrm{DTIME}\left(2^{O(s)}\right)$ by a result of Borodin, Cook and Pippenger [7] (see below).

Next, we consider quantum simulations of space-bounded probabilistic machines. The most straightforward technique by which quantum machines can simulate probabilistic machines (presented in [4], for example), involves direct simulation of the probabilistic machine's coin-flips by appropriately defined quantum transformations (e.g., Hadamard transforms). Although the resulting simulation is quite efficient in the time-bounded setting, it is terribly inefficient in the space-bounded case. Indeed, since a PTM running in space s may require a number of coin-flips exponential in s (or even doubly exponential in case the PTM does not halt absolutely), and since there is no obvious way to reuse the space required for each simulated coin-flip, this technique may result in an exponential increase in space. A considerably more efficient technique is to simply derandomize the probabilistic computation and to simulate the resulting deterministic computation with a quantum machine. As QTMs can perform exactly those deterministic computations that are reversible, it is appropriate to refer to previous work on reversible computation at this point.

A reversible Turing machine (RTM) is a deterministic Turing machine (DTM) for which every configuration has at most one immediate predecessor. It was proved by Bennett [2] that any DTM computation can be simulated by an RTM. Although Bennett's simulation incurred only a constant factor increase in running time, in the worst case the space required for the simulation was exponential in the space required by the original machine. Bennett later improved the space-efficiency of this simulation so that it required at most a quadratic increase in space, at the cost of only a slight increase in running time [3]. This implies $\mathrm{DSPACE}(s) \subseteq \mathrm{RevSPACE}(s^2)$, where $\mathrm{RevSPACE}(s)$ denotes, for a

given space bound s, the class of languages recognizable in space $O(s)$ by an RTM. It was later proved [8] that nondeterministic Turing machines can also be simulated reversibly with the same increase in space, i.e., $\mathrm{NSPACE}(s) \subseteq \mathrm{RevSPACE}(s^2)$. Recently, Lange, McKenzie and Tapp [18] proved that, at the cost of a possibly exponential increase in running time, DTMs can be simulated by RTMs with only a constant factor increase in space, i.e., $\mathrm{DSPACE}(s) = \mathrm{RevSPACE}(s)$.

Various relationships regarding quantum simulations of probabilistic machines follow from derandomization, given that $\mathrm{DSPACE}(s) = \mathrm{RevSPACE}(s)$. Independently, Jung [15] and Borodin, Cook and Pippenger [7] showed that any (unbounded error) PTM can be simulated deterministically with at most a quadratic increase in space, i.e., $\mathrm{PrSPACE}(s) \subseteq \mathrm{DSPACE}(s^2)$. (In fact, Borodin, Cook and Pippenger prove the somewhat stronger relationship $\mathrm{PrSPACE}(s) \subseteq \mathrm{NC}^2(2^s) \subseteq \mathrm{DSPACE}(s^2) \cap \mathrm{DTIME}\left(2^{O(s)}\right)$.) This implies that RTMs, and hence QTMs, can also simulate PTMs with at most a quadratic increase in space. Along similar lines, Saks and Zhou [21] proved that any bounded error PTM that runs in space s and halts absolutely (i.e., has finite worst case running time) can be simulated deterministically (and hence by a QTM) in space $O(s^{3/2})$.

A natural question to ask is if it is possible for QTMs to simulate PTMs in a more space-efficient manner than implied by these deterministic simulations. We prove in this paper that any bounded error PTM that runs in space s and halts absolutely can be simulated by a bounded error QTM running in space $O(s)$ (but which does not necessarily halt absolutely). A similar result is shown to hold for the cases of one-sided error and unbounded error, and in the case of unbounded error it may be assumed that the quantum machine does halt absolutely. It follows from these simulations that unbounded error, space-bounded PTMs and QTMs are equivalent in power. Furthermore, we have that unbounded error, space-bounded QTMs do not lose power if required to halt absolutely; a result analogous to one proved by Jung [16] for the probabilistic case (see also [1]).

Finally, we define quantum analogues of nondeterministic space-bounded classes by considering whether or not input strings are accepted with zero or nonzero probability. It is shown that the class of languages for which there exists a space s QTM accepting precisely those strings in the given language with nonzero probability corresponds to the counting class co-$\mathrm{C}_{=}\mathrm{SPACE}(s)$, and hence contains $\mathrm{NSPACE}(s)$. This characterization may be viewed as the space-bounded analogue of a recent result of Fenner, Green, Homer, and Pruim [11] that equates "quantum NP" and co-$\mathrm{C}_{=}\mathrm{P}$. Simple relationships between co-$\mathrm{C}_{=}\mathrm{SPACE}(s)$ and one-sided error space-bounded quantum classes are examined as well.

The remainder of this paper has the following organization. In Section 2 we define the quantum Turing machine model and space-bounded quantum complexity classes studied throughout the paper. Section 3 examines complexity-theoretic relationships following from classical simulations of quantum Turing machine computations, and Section 4 examines relationships following from quantum simulations of both classical and quantum Turing machine computations. Results of Section 4 rely on a number of lemmas regarding QTM constructions that are proved in Section 5. We conclude with Section 6, which mentions a number of open questions pertaining to space-bounded quantum computation.

2 Definitions

We begin by mentioning some of the notation used in this paper. As usual, $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Q}$ denote the natural numbers (excluding 0), integers, and rational numbers, respectively, and $\mathbb{Z}^+ = \mathbb{N} \cup \{0\}$. The empty string over any given alphabet is denoted by ε. For any finite or countable set S, $\ell_2(S)$ denotes the Hilbert space whose elements are mappings from S to the complex numbers. Elements of such spaces will be expressed using the Dirac notation: for each $s \in S$, $|s\rangle$ denotes the elementary unit

vector taking value 1 at s and 0 elsewhere, and arbitrary elements of $\ell_2(S)$ (generally denoted $|\psi\rangle$, $|\phi\rangle$, etc.) may be written as linear combinations of these elementary vectors. For $|\phi\rangle \in \ell_2(S)$, $\langle\phi|$ denotes the linear functional mapping each $|\psi\rangle \in \ell_2(S)$ to the inner product $\langle \phi \,|\, \psi\rangle$ (conjugate-linear in the first coordinate). For a matrix A, we denote the i, j entry of A by $A[i, j]$, while $A_{i,j}$ denotes the matrix obtained by removing the ith row and jth column of A. All logarithms in this paper are to the base 2.

2.1 Quantum Turing machines

A quantum Turing machine (QTM) consists of the following components: a read-only input tape with a two-way tape head, a read/write work tape with a two-way tape head, a write-only output tape with a one-way tape head, and a finite state control. The input and work tapes are assumed to be two-way infinite and indexed by $\mathbb{Z}$, while the output tape is one-way infinite and indexed by $\mathbb{Z}^+$. For a given QTM M, we let Q, Σ and Γ denote the set of internal states, input tape alphabet, and work tape alphabet of M, respectively. It is assumed that Q contains an initial state q_0 and that Σ and Γ each contain a distinguished blank symbol denoted by #. The output tape alphabet will always be assumed to be $\{0, 1, \#\}$. All input strings are assumed to be elements of $(\Sigma\backslash\{\#\})^*$, i.e., containing no embedded blank symbols.

A configuration of a QTM includes (1) the internal state of the machine, (2) the position of the input tape head, (3) the contents of the work tape and the position of the work tape head, and (4) the contents of the output tape and the position of the output tape head. It is assumed that only finitely many tape squares contain non-blank symbols in any configuration, and further that any non-blank symbol on the output tape must be written in a tape square having index smaller than the current output tape head position. We denote the set of such configurations of a QTM M by $\mathcal{C}(M)$ (or just $\mathcal{C}$ if M is understood from context). The initial configuration, denoted c_0, is that configuration in which the internal state is q_0, all tape heads are positioned over the tape squares indexed by 0, and all tape squares on the work tape and output tape contain blanks.

Throughout the computation of a given machine on input x, it is assumed that x is written on the input tape in squares $1, \ldots, |x|$, and all remaining squares on the input tape contain blanks.

At a given instant, it may not be the case that a QTM is in a single configuration, but rather the machine may be in a *superposition* of configurations. A superposition of a QTM M is a unit vector in the Hilbert space $\ell_2(\mathcal{C}(M))$. For a given superposition $|\psi\rangle = \sum_{c\in\mathcal{C}} \alpha_c \,|c\rangle$, each α_c is called the amplitude associated with configuration c. Superpositions of the form $|c\rangle$ for $c \in \mathcal{C}$ are called classical states, and correspond to the machine being in configuration c.

The manner in which a QTM M evolves from one superposition to the next is specified by a transition function μ having the form

$$\mu : Q \times \Sigma \times \Gamma \times Q \times \{-1, 0, 1\} \times \Gamma \times \{-1, 0, 1\} \times \{0, 1, \varepsilon\} \to \mathbb{Q}.$$

Each number $\mu(q, \sigma, \tau, q', d_i, \tau', d_w, \omega)$ may be interpreted as follows. Suppose M is currently in a classical state $|c\rangle$ for which the internal state is q and the symbols σ and τ are currently being scanned on the input tape and work tape, respectively. Then in one step, M will be in a superposition for which $\mu(q, \sigma, \tau, q', d_i, \tau', d_w, \omega)$, is the amplitude associated with that configuration resulting from c by (1) changing the internal state to q', (2) moving the input tape head in direction d_i, (3) replacing τ with τ' on the work tape and moving the work tape head in direction d_w, and (4) writing ω on the output tape and moving the output tape head one square to the right (or, if $\omega = \varepsilon$, writing nothing on the output tape and leaving the output tape head stationary), for each q', d_i, τ', d_w and ω.

The behavior of M on superpositions is determined by linearity. For a given input x and any pair of configurations c and c', let $\alpha_x(c \vdash c')$ denote the amplitude associated with the transition $c \vdash c'$, as specified by μ in the manner described above. (For c' not reachable from c in a single transition, we define $\alpha(c \vdash c') = 0$.) The *time evolution operator* of M on input x may now be defined as

$$U_x = \sum_{c,c' \in \mathcal{C}} \alpha_x \left(c \vdash c'\right) |c'\rangle \langle c| \,;$$

if M on input x is in superposition $|\psi\rangle$ and evolves for one step, it will then be in superposition $U_x |\psi\rangle$.

The following restriction is placed on all transition functions for the remainder of this paper. It is assumed that for each $\sigma \in \Sigma$ there exists a unitary (i.e., norm preserving and invertible) mapping $V_\sigma : \ell_2(Q \times \Gamma) \to \ell_2(Q \times \Gamma)$, and for each $q \in Q$ there exist $D_i(q)$, $D_w(q) \in \{-1, 0, 1\}$ and $Z(q) \in \{0, 1, \varepsilon\}$, such that

$$\mu(q, \sigma, \tau, q', d_i, \tau', d_w, \omega) = \begin{cases} \langle q', \tau' \,|\, V_\sigma \,|\, q, \tau \rangle & \text{if } d_i = D_i(q'),\ d_w = D_w(q'), \text{ and } \omega = Z(q') \\ 0 & \text{otherwise} \end{cases}$$

for every choice of arguments to μ. This restriction, which is analogous to unidirectionality for the single-tape QTM model discussed in [4], essentially requires that the output and movement of tape heads of a QTM depend only on the internal state the machine enters on the step in question. It is proved in [4] that unidirectionality does not decrease the power of single-tape QTMs, and the proof extends readily to the multitape case. In the interest of simplicity, we prefer to include this restriction as part of the definition of QTMs. Note that the restriction implies that U_x is necessarily a norm-preserving operator for every input x, following from the fact that each V_σ is unitary.

We define reversible Turing machines (RTMs) to be QTMs having transition functions that take only the values 0 and 1. The RTMs considered in [2, 3, 18] have transition functions that may be expressed as QTMs in this way, in accordance with the abovementioned restriction.

In order for a QTM to reveal any information about its computation, it must be observed; the information revealed by a particular observation is described by an *observable*. For the purposes of this paper, it is sufficient to define observables as finite or countable collections $\{(P_j, r_j)\}$, where each P_j is a projection operator on $\ell_2(\mathcal{C})$ and each r_j is a *result*, which we take to be some element of $\{0,1\}^*$. This collection of pairs must satisfy (1) $P_j P_k = 0$ for $j \neq k$, (2) $\sum_j P_j = I$, and (3) $r_j \neq r_k$ for $j \neq k$. If a machine M in superposition $|\psi\rangle$ is observed with observable $\{(P_j, r_j)\}$, then the following occurs:

1. Each result r_j will be selected with probability $\|P_j |\psi\rangle\|^2$.
2. For whichever result r_j was selected, the superposition of M will "collapse" to $\frac{1}{\|P_j|\psi\rangle\|} P_j |\psi\rangle$.

As superpositions are of unit norm, it follows that the probabilities in item 1 sum to 1. Item 2 implies that the superposition of M immediately after the observation will also be of unit norm.

The particular observable that we will restrict our attention to corresponds to simply observing the contents of the output tape. As the output tape head moves right one square exactly when one of the symbols in $\{0,1\}$ is written to the output tape, the contents of this tape together with the position of the tape head are in one-to-one correspondence with strings in $\{0,1\}^*$. For each $w \in \{0,1\}^*$, let P_w be the projection from $\ell_2(\mathcal{C})$ onto the space spanned by classical states for which the output tape contents and tape head position are described by w. Now $\{(P_w, w)\}_{w \in \Sigma^*}$ is a formal description of our observable.

The computation of a given QTM M on input x is to proceed as follows. We assume that M begins in the classical state $|c_0\rangle$ with x written on its input tape. Each step of the computation consists of

two phases: first the machine evolves for one step according to U_x, and second the output tape of the machine is observed as described above. The computation continues in this way until it has been observed that some symbol has been written to the output tape; if the observed symbol is "1", the result of the computation is *accept*, and if the symbol observed is "0", the result is *reject*.

Since the results of the observations occurring during a given computation are random, we may view a given computation as being a random process. For a given QTM M, input x, $j \in \mathbb{N}$ and $\omega \in \{0,1\}$, let $p_{x,j,\omega}$ denote the probability that, if M on input x is run as described above, each observation at time $j' < j$ yields ε and the observation at time j yields ω. The probability that M accepts x is thus $\sum_j p_{x,j,1}$, and the probability that M rejects x is $\sum_j p_{x,j,0}$. A straightforward proof by induction shows

$$p_{x,j,\omega} = \left\| P_\omega (U_x P_\varepsilon)^j \left| c_0 \right\rangle \right\|^2 . \tag{1}$$

If, for a given input x, $\sum_j (p_{x,j,1} + p_{x,j,0}) = 1$, i.e., M halts on x with probability 1, we say M *halts almost surely* on x. If there exists $k = k(x)$ such that $\sum_{j \leq k} (p_{x,j,1} + p_{x,j,0}) = 1$, i.e., M on x halts with certainty after k steps, then we say that M *halts absolutely* on input x. If M halts almost surely (halts absolutely) on every input x, then we simply say M halts almost surely (halts absolutely, respectively).

2.2 Space-bounded quantum classes

The space used by (quantum and classical) Turing machines will be measured in terms of the number of bits required to encode certain information regarding configurations of these machines, relative to some reasonable encoding scheme. We note that this notion of space will differ from the more standard notion by at most a constant factor for the space bounds we consider. Specifically, the following information regarding each configuration is to be encoded: (1) the internal state of the machine, (2) the position of the input tape head, (3) the position of the work tape head and the contents of the work tape, and (4) the symbol contained in the output tape square indexed by 0. It is assumed that the length of the encoding of any configuration is logarithmic in the distance of the input tape head from square 0 (for fixed work tape contents and work tape head position), and is linear in both the maximum distance of any non-blank work tape square from square 0 and in the distance of the work tape head from square 0 (for fixed input tape head position). We further assume that each encoding begins with 1, and each configuration has a unique encoding. Now, we say that the space required for a given configuration is the length of the binary string encoding the above information about this configuration. It follows that the number of configurations with space bounded by l is at most 2^l, and each such configuration can be written uniquely as a binary string of length l (padding the beginning of the string with zeroes as necessary).

Next, we say that the space required for a superposition is the maximum space required for any configuration having nonzero amplitude in that superposition, and we say that a QTM M on input x runs in space l if each superposition obtained during an execution of M on x requires space at most l. More precisely, M on x runs in space l if, for every $k \geq 0$, we have that each configuration c for which $\left\langle c \left| (U_x P_\varepsilon)^k \right| c_0 \right\rangle \neq 0$ requires space at most l. (Note that the fact that a given QTM runs within a particular space bound may depend on the machine being observed after each step.) Similarly, we say that a PTM on input x runs in space l if each configuration reachable with nonzero probability requires space at most l.

Finally, we say that a QTM or PTM M runs in space s (where s will always denote a function of the form $s : \mathbb{Z}^+ \to \mathbb{N}$) if, for every input x, M on input x runs in space $s(|x|)$. Throughout this paper, whenever we refer to a space bound s, we assume that $s(n) = \Omega(\log n)$ and that s is space constructible.

Frequently we will write s to mean $s(|x|)$, and similarly for any function $t : \mathbb{Z}^+ \to \mathbb{N}$ denoting some number of time steps that is a function of $|x|$.

When we say that a time bound $t : \mathbb{Z}^+ \to \mathbb{N}$ is computable in space $O(s)$, we mean the following: there exists a DTM running in space $O(s)$ that, on input x, writes t in binary on its work tape and then halts (so it is implicit that $t = 2^{O(s)}$).

We now define various complexity classes based on space-bounded QTMs. For each prefix $X \in \{\mathrm{EQ}, \mathrm{RQ}, \mathrm{BQ}, \mathrm{NQ}, \mathrm{PrQ}\}$, a given language L is said to be in the class $X\mathrm{SPACE}(s)$ if there exists a QTM M that runs in space $O(s)$ and satisfies the appropriate condition below:

EQSPACE(s):
For $x \in L$, M accepts x with probability 1, and for $x \notin L$, M accepts x with probability 0.

RQSPACE(s):
There exists an $\varepsilon > 0$ such that for $x \in L$, M accepts x with probability greater than $\frac{1}{2} + \varepsilon$, and for $x \notin L$, M accepts x with probability 0.

BQSPACE(s):
There exists an $\varepsilon > 0$ such that for $x \in L$, M accepts x with probability greater than $\frac{1}{2} + \varepsilon$, and for $x \notin L$, M accepts x with probability less than $\frac{1}{2} - \varepsilon$.

NQSPACE(s):
For $x \in L$, M accepts x with probability greater than 0, and for $x \notin L$, M accepts x with probability 0.

PrQSPACE(s):
For $x \in L$, M accepts x with probability strictly greater than $\frac{1}{2}$, and for $x \notin L$, M accepts x with probability less than or equal to $\frac{1}{2}$.

If in addition M halts almost surely, then L is in the class $X_{AS}\mathrm{SPACE}(s)$, and if M halts absolutely, then L is in the class $X_H\mathrm{SPACE}(s)$.

Naturally, we have

$$X_H\mathrm{SPACE}(s) \subseteq X_{AS}\mathrm{SPACE}(s) \subseteq X\mathrm{SPACE}(s)$$

for each $X \in \{\mathrm{EQ}, \mathrm{RQ}, \mathrm{BQ}, \mathrm{NQ}, \mathrm{PrQ}\}$. Furthermore,

$$\mathrm{RevSPACE}(s) \subseteq \mathrm{EQSPACE}(s) \subseteq \mathrm{RQSPACE}(s) \subseteq \mathrm{BQSPACE}(s) \subseteq \mathrm{PrQSPACE}(s),$$
$$\mathrm{RQSPACE}(s) \subseteq \mathrm{NQSPACE}(s),$$

and similarly for the halting almost surely and halting absolutely versions of these classes.

The prefixes RQ, BQ, NQ and PrQ may be replaced by R, BP, N and Pr, respectively, to obtain the analogously defined probabilistic classes. Here we have adopted the notation of [20], to which the reader is referred for further information regarding the probabilistic versions of these classes.

3 Classical Simulations of Quantum Machines

We consider in this section probabilistic simulations of quantum Turing machine computations. It is proved that probabilistic Turing machines can simulate quantum Turing machines with at most a

constant factor increase in space in the unbounded error case, i.e., PrQSPACE(s) $\subseteq$ PrSPACE(s). Using similar arguments, we conclude NQSPACE(s) $\subseteq$ co-C$_=$SPACE(s) as well.

The class C$_=$SPACE(s) is a straightforward generalization of the counting class C$_=$L defined in [1]. There are a number of equivalent definitions of C$_=$SPACE(s); we will use the following definition. A language L is in C$_=$SPACE(s) if there exists a PTM that runs in space $O(s)$, halts absolutely, and accepts each input x with probability precisely equal to $1/2$ if and only if $x \in L$.

Our proofs of the above relationships rely on the two lemmas that follow.

Lemma 3.1 *Define*

$$\begin{aligned} L_1 &= \{(A,B) : A \text{ and } B \text{ are integer matrices satisfying } det(A) > det(B)\}, \\ L_2 &= \{A : A \text{ is an integer matrix satisfying } det(A) \neq 0\}. \end{aligned}$$

Then $L_1 \in$ PrSPACE($\log n$) and $L_2 \in$ co-C$_=$SPACE($\log n$), where n denotes the length of the encoding of (A,B) or A appropriately.

The two facts comprising Lemma 3.1 are noted by Allender and Ogihara [1], to which the reader is referred for a proof.

Lemma 3.2 *Let M be a QTM running in space s. Then for each input x there exist $2^{O(s)} \times 2^{O(s)}$ matrices $A(x)$ and $B(x)$, where entries of $A(x)$ and $B(x)$ are integers of length $2^{O(s)}$, such that the following properties are satisfied.*

1. *$det(A(x)) > det(B(x))$ if and only if M accepts x with probability exceeding $\frac{1}{2}$.*
2. *$det(A(x)) = 0$ if and only if M accepts x with probability 0.*
3. *There exists a DTM that, on input x and with integer $k = 2^{O(s)}$ initially written on its work tape, computes the kth bit of the encoding $(A(x), B(x))$ in space $O(s)$.*

Lemma 3.2 is proved below. First, however, let us state and prove the main results of this section, which rely on the above two lemmas.

Theorem 3.3 *PrQSPACE(s) $\subseteq$ PrSPACE(s).*

Proof. Let M be a QTM running in space s, and let matrices $A(x)$ and $B(x)$ be as stated in Lemma 3.2 for each input x. As $A(x)$ and $B(x)$ are of dimension $2^{O(s)} \times 2^{O(s)}$ and have entries with length at most $2^{O(s)}$, we may assume that the length of the encoding of $(A(x), B(x))$ is at most $2^{O(s)}$.

By Lemma 3.1 there exists a PTM M_1 that runs in space s' for $s'(n) = O(\log n)$ and accepts with probability exceeding $1/2$ exactly those strings in the language L_1. Define a PTM M_2 that, on input x, simulates M_1 on $(A(x), B(x))$ as follows. M_2 will record the position of M_1's tape head, which requires space at most $O(s)$. During each step of M_1, M_2 computes the symbol in the encoding of $(A(x), B(x))$ corresponding to the position of M_2's input tape head, then simulates the action of M_1 given this input symbol. By item 3 of Lemma 3.2, this input symbol can be computed in space $O(s)$. As M_1 runs in space logarithmic in the length of $(A(x), B(x))$, i.e., in space $O(s)$, it follows that M_2 runs in space $O(s)$ as well. By definition of L_1, along with item 1 of Lemma 3.2, we have that M_2 accepts with probability exceeding $1/2$ exactly when the same is true of M. ■

Corollary 3.4 *PrQSPACE(s) $\subseteq$ NC$^2(2^s) \subseteq$ DSPACE(s^2) $\cap$ DTIME($2^{O(s)}$).*

This corollary follows from the well-known result PrSPACE(s) $\subseteq$ NC$^2(2^s)$ due to Borodin, Cook, and Pippenger [7].

Next, we relate NQSPACE(s) to co-C$_=$SPACE(s).

Theorem 3.5 *NQSPACE(s) $\subseteq$ co-C$_=$SPACE(s).*

Proof. Let M be a QTM running in space s. We show that there exists a PTM M_2 running in space $O(s)$ that accepts each input x with probability precisely $\frac{1}{2}$ if and only if M accepts x with probability 0.

By Lemma 3.1, we have that

$$L_2 = \{A : A \text{ is an integer matrix satisfying } \det(A) \neq 0\}$$

is in co-C$_=$SPACE($\log n$), from which it follows that there exists a log-space PTM M_1 that takes as input an encoding of an integer matrix A and accepts with probability precisely $\frac{1}{2}$ if and only if $\det(A) = 0$. Similar to the proof of Theorem 3.3, we may construct a PTM M_2 that, on input x, simulates the action of M_1 on the matrix $A(x)$, for $A(x)$ as in Lemma 3.2. By item 2 of Lemma 3.2, $\det(A(x)) = 0$ if and only if M accepts with probability 0, from which the theorem follows. ■

The remainder of this section is devoted to the proof of Lemma 3.2. The proof will utilize the lemmas that follow.

Lemma 3.6 *Let M be a QTM running in space s. Then for each input x, there exists an $N \times N$ matrix $E(x)$, where $N = 2^{2s} + 2$, such that the following properties are satisfied.*

1. *For each $k \geq 0$, $E(x)^{k+2}[N, 1]$ is the probability that M accepts x after precisely k steps (and $E(x)[N, 1] = 0$).*
2. *Each entry of $E(x)$ may be written in the form $\frac{a_{ij}}{m}$, where m is the square of the least common denominator of the values taken by the transition function of M and $a_{ij} \in \{-m, \ldots, m\}$.*
3. *There exists a DTM M_0 that, on input x and with indices $i, j \in \{1, \ldots, N\}$ initially written on its work tape, computes the value $m \cdot E(x)[i, j]$ in space $O(s)$.*
4. *All eigenvalues of $E(x)$ are bounded in absolute value by 1.*

Proof. For given QTM M and input x, recall the definitions of U_x, P_1 and P_ε from Section 2.1.

First, define a $2^s \times 2^s$ matrix $D(x)$ as follows.

$$D(x)[i, j] = \begin{cases} \langle c' \,|\, U_x P_\varepsilon \,|\, c \rangle & \text{if } (i-1), (j-1) \text{ encode } c', c \in \mathcal{C}, \text{ respectively} \\ 0 & \text{otherwise.} \end{cases}$$

Here we identify integers in the range $\{0, \ldots, 2^s - 1\}$ with length s binary strings in the most straightforward way. Next, define 2^{2s}-dimensional vectors $y_{init}(x)$ and $y_{acc}(x)$ as follows.

$$y_{init}(x)[i] = \begin{cases} 1 & \text{if } i = (i_0 - 1)2^s + i_0 \text{ with } (i_0 - 1) \in \{0, \ldots, 2^s - 1\} \text{ encoding} \\ & \text{the initial configuration of } M \text{ on } x \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{acc}(x)[i] = \begin{cases} 1 & \text{if } i = (i_0 - 1)2^s + i_0 \text{ with } (i_0 - 1) \in \{0, \ldots, 2^s - 1\} \text{ encoding} \\ & \text{an accepting configuration of } M \text{ on } x \\ 0 & \text{otherwise.} \end{cases}$$

Finally, define $E(x)$ as follows.

$$E(x)[i,j] = \begin{cases} D(x)[i_0,j_0]D(x)[i_1,j_1] & \text{if } i = (i_0-1)2^s + i_1 + 1, \\ & j = (j_0-1)2^s + j_1 + 1, \\ & i_0, i_1, j_0, j_1 \in \{1, \ldots, 2^s\} \\ y_{init}(x)[i-1] & \text{if } i \in \{2, \ldots, 2^s+1\},\ j = 1 \\ y_{acc}(x)[j-1] & \text{if } i = 2^{2s}+2,\ j \in \{2, \ldots, 2^{2s}+1\} \\ 0 & \text{otherwise.} \end{cases}$$

Note that $E(x)$ takes the form

$$E(x) = \begin{bmatrix} 0 & 0 & 0 \\ y_{init}(x) & D(x) \otimes D(x) & 0 \\ 0 & y_{acc}^T(x) & 0 \end{bmatrix},$$

where $\otimes$ denotes the Kronecker product, and $E(x)^{k+2}$ takes the form

$$E(x)^{k+2} = \begin{bmatrix} 0 & 0 & 0 \\ (D(x) \otimes D(x))^{k+1} y_{init}(x) & (D(x) \otimes D(x))^{k+2} & 0 \\ y_{acc}^T(x)(D(x) \otimes D(x))^k y_{init}(x) & y_{acc}^T(x)(D(x) \otimes D(x))^{k+1} & 0 \end{bmatrix}$$

for each $k \geq 0$. Consequently, we have

$$\begin{aligned} E(x)^{k+2}[2^{2s}+2, 1] &= y_{acc}^T \left(D(x)^k \otimes D(x)^k \right) y_{init}(x) \\ &= \sum_{\substack{i \in \{1,\ldots,2^s\} \\ (i-1) \text{ encodes } c \in \mathcal{C}_{acc}}} \left(D(x)^k[i, j_{init}] \right)^2 \\ &= \left\| P_1 (U_x P_\varepsilon)^k \left| c_0 \right\rangle \right\|^2, \end{aligned}$$

where we assume $(j_{init} - 1) \in \{0, \ldots, 2^s - 1\}$ encodes the initial configuration of M on input x, and we write $\mathcal{C}_{acc}$ to denote the set of accepting configurations of M. Note that the last equality in the above equation follows from the fact that M runs in space s. By (1), we therefore have that $E(x)^{k+2}[2^{2s}+2, 1]$ is the probability that M accepts x after precisely k steps as required.

It remains to show that items 2 – 4 in the statement of the lemma are satisfied. Item 2 is obvious from the definition of each $E(x)$, and item 3 is straightforward given reasonable assumptions on our encoding of configurations. To prove item 4, we first note that since U_x is norm-preserving and P_ε is a projection, multiplication by $D(x)$ cannot increase the length of any vector. Thus, all eigenvalues of $D(x)$ are bounded in absolute value by 1, and hence the same is true of $D(x) \otimes D(x)$. Now item 4 follows by noting that any nonzero eigenvalue of $E(x)$ must also be an eigenvalue of $D(x) \otimes D(x)$. ■

The following notation is used in the lemmas that follow: for a given polynomial $f(z) = \sum_{j=0}^n a_j z^j$, define the *height* of f, denoted $\|f\|$, as $\|f\| = \max\{|a_j| : j = 0, \ldots, n\}$.

Lemma 3.7 *Let $Q(z)$ be an $N \times N$ matrix having entries that are polynomials in z with degree at most 1 and height bounded by m. Then*

$$\|\det(Q(z))\| \leq N!\, m^N 2^{N-1}.$$

Proof. For any two polynomials f and g, it is straightforward to show

$$\|fg\| \;\leq\; \|f\|\,\|g\|\,(\deg(f)+1).$$

Consequently, the product of any N entries of $Q(z)$ must have height bounded by $m^N 2^{N-1}$, and hence

$$\det(Q(z)) \;=\; \sum_{\sigma\in S_N} \operatorname{sign}(\sigma) \prod_{i=1}^{N} Q(z)[i,\sigma(i)]$$

has height bounded by $N!\,m^N 2^{N-1}$ as required. ■

Lemma 3.8 *Let f and g be integer polynomials satisfying $(1-z)^k f(z) = g(z)$. Then*

$$\|f\| \;\leq\; \|g\| \binom{\deg(g)}{k}.$$

Proof. Given an integer polynomial $g(z) = \sum_{j=0}^{n} a_j z^j$ that is divisible by $(1-z)^k$, we may write $f(z) = g(z)/(1-z)^k$ explicitly as

$$f(z) \;=\; \sum_{i=0}^{n-k} \left(\sum_{j=0}^{i} \binom{j+k-1}{j} a_{i-j} \right) z^i;$$

a fact that follows from the power series expansion $(1-z)^{-k} = \sum_{j\geq 0} \binom{j+k-1}{j} z^j$. Consequently

$$\|f\| \;\leq\; \|g\| \sum_{j=0}^{n-k} \binom{j+k-1}{j} \;=\; \|g\| \binom{n}{k}$$

as claimed. ■

Lemma 3.9 *Let f and g be integer polynomials satisfying $\|f\|, \|g\| \leq K$, $\deg(f), \deg(g) \leq N$, and $g(1) \neq 0$, and suppose that $0 < \epsilon < \frac{1}{N(N+1)K}$. Then*

$$\left| \frac{f(1)}{g(1)} - \frac{f(1-\epsilon)}{g(1-\epsilon)} \right| \;\leq\; 2\,\epsilon\,N\,(N+1)^2\,K^2.$$

Proof. For any polynomial h we have

$$|h(1) - h(1-\epsilon)| \;\leq\; \epsilon\,\|h\| \binom{\deg(h)+1}{2}$$

by the Mean Value Theorem. Thus,

$$|g(1) - g(1-\epsilon)| \;\leq\; \epsilon\,\|g\| \binom{\deg(g)+1}{2} \;<\; \frac{1}{2},$$

implying that $|g(1-\epsilon)| > 1/2$ since $g(1)$ is a nonzero integer. We may now conclude

$$\begin{aligned}\left|\frac{f(1)}{g(1)} - \frac{f(1-\epsilon)}{g(1-\epsilon)}\right| &\leq \left|\frac{f(1)g(1-\epsilon) - f(1)g(1)}{g(1)g(1-\epsilon)}\right| + \left|\frac{f(1)g(1) - f(1-\epsilon)g(1)}{g(1)g(1-\epsilon)}\right| \\ &\leq 2(N+1)K\left(|g(1)-g(1-\epsilon)| + |f(1)-f(1-\epsilon)|\right) \\ &\leq 2\,\epsilon\,N(N+1)^2K^2,\end{aligned}$$

as claimed. ■

Lemma 3.10 *For every positive integer m, there exists a polynomial p, where $p(n) > 0$ for $n > 0$, satisfying the following. Given any $N \times N$ matrix E for which each entry $E[i,j]$ takes the form $\frac{a_{ij}}{m}$ for $a_{ij} \in \{-m, \ldots, m\}$, and for which all eigenvalues are bounded in absolute value by 1, we have (i) $\left(I - \left(1 - 2^{-p(N)}\right)E\right)$ is invertible, and (ii) for any value of i, j for which $E^l[i,j] \geq 0$ for every $l \geq 0$, and for which $\lim_{z\uparrow 1}(I - z\,E)^{-1}[i,j]$ exists, we have*

$$\lim_{z\uparrow 1}(I - z\,E)^{-1}[i,j] \;>\; \frac{1}{2} \qquad \textit{if and only if} \qquad \left(I - \left(1 - 2^{-p(N)}\right)E\right)^{-1}[i,j] \;>\; \frac{1}{2},$$

and

$$\lim_{z\uparrow 1}(I - z\,E)^{-1}[i,j] \;=\; 0 \qquad \textit{if and only if} \qquad \left(I - \left(1 - 2^{-p(N)}\right)E\right)^{-1}[i,j] \;=\; 0.$$

Proof. Under the assumption that all eigenvalues of E are bounded in absolute value by 1, we have that $(I - z\,E)$ is invertible for $|z| < 1$, and hence $I - \left(1 - 2^{-p(N)}\right)E$ is invertible.

For fixed i and j, define

$$\begin{aligned}u(z) &= (-1)^{i+j}m^N \det((I - z\,E)_{j,i}), \\ v(z) &= m^N \det(I - z\,E).\end{aligned}$$

We have that u and v are integer polynomials in z and $\frac{u(z)}{v(z)} = (I - z\,E)^{-1}[i,j]$ for $|z| < 1$. Under the assumption that $\lim_{z\uparrow 1}(I - z\,E)^{-1}[i,j]$ exists, we may write $u(z) = (1-z)^k f(z)$ and $v(z) = (1-z)^k g(z)$ for some $k \geq 0$, where f and g are integer polynomials with $g(1) \neq 0$, so that $\frac{f(z)}{g(z)} = (I - z\,E)^{-1}[i,j]$ for $|z| < 1$, and $\frac{f(1)}{g(1)} = \lim_{z\uparrow 1}(1 - z\,E)^{-1}[i,j]$. Note that since

$$\frac{f(z)}{g(z)} \;=\; (I - z\,E)^{-1}[i,j] \;=\; \sum_{l\geq 0} z^l E^l[i,j]$$

for $|z| < 1$ (see [14], p. 54 for example), the assumption $E^l[i,j] \geq 0$ for $l \geq 0$ implies $\frac{f(z)}{g(z)}$ is nondecreasing and nonnegative on the interval $[0,1]$.

By Lemma 3.7 we have $\|u\|, \|v\| \leq N!\,m^N 2^{N-1}$, and thus $\|f\|, \|g\| \leq N!\,m^N 2^{2N-1}$ by Lemma 3.8. Consequently, if $\frac{f(1)}{g(1)} > \frac{1}{2}$, then

$$\frac{f(1)}{g(1)} - \frac{1}{2} \;\geq\; \left|\frac{1}{2\,g(1)}\right| \;\geq\; \frac{1}{(N+1)!\,m^N 2^{2N}}. \tag{2}$$

Furthermore, by Lemma 3.9 we have

$$\frac{f(1)}{g(1)} - \frac{f(1-\epsilon)}{g(1-\epsilon)} \;<\; \epsilon N((N+1)!)^2 m^{2N} 2^{4N-1}, \tag{3}$$

for $\epsilon < \left(N(N+1)!m^N 2^{2N-1}\right)^{-1}$.

Now take p to be any polynomial satisfying

$$p(N) > \log\left(N((N+1)!)^3 m^{3N} 2^{\,6N-1}\right)$$

for every $N \geq 1$ (e.g., $p(N) = 3N^2 + (3\log m + 7)N$).

It remains to prove $\frac{f(1)}{g(1)} > \frac{1}{2}$ if and only if $\frac{f\left(1-2^{-p(N)}\right)}{g\left(1-2^{-p(N)}\right)} > \frac{1}{2}$, and $\frac{f(1)}{g(1)} = 0$ if and only if $\frac{f\left(1-2^{-p(N)}\right)}{g\left(1-2^{-p(N)}\right)} = 0$. First, note that $\frac{f(1)}{g(1)} \leq \frac{1}{2}$ implies $\frac{f\left(1-2^{-p(N)}\right)}{g\left(1-2^{-p(N)}\right)} \leq \frac{1}{2}$, and $\frac{f(1)}{g(1)} = 0$ implies $\frac{f\left(1-2^{-p(N)}\right)}{g\left(1-2^{-p(N)}\right)} = 0$, as $\frac{f(z)}{g(z)}$ is nondecreasing and nonnegative on $[0,1]$. Now assume $\frac{f(1)}{g(1)} > \frac{1}{2}$. Since

$$2^{-p(N)} N((N+1)!)^2 m^{2N} 2^{4N-1} < \frac{1}{(N+1)!\, m^N 2^{2N}},$$

we have

$$\frac{f(1)}{g(1)} - \frac{f\left(1-2^{-p(N)}\right)}{g\left(1-2^{-p(N)}\right)} \;<\; \frac{1}{(N+1)!\, m^N 2^{2N}}$$

by (3). Thus

$$\frac{f(1)}{g(1)} - \frac{f\left(1-2^{-p(N)}\right)}{g\left(1-2^{-p(N)}\right)} \;<\; \frac{f(1)}{g(1)} - \frac{1}{2}$$

by (2), and hence $\frac{f\left(1-2^{-p(N)}\right)}{g\left(1-2^{-p(N)}\right)} > \frac{1}{2}$. Finally, suppose $\frac{f\left(1-2^{-p(N)}\right)}{g\left(1-2^{-p(N)}\right)} = 0$. Since $\frac{f(z)}{g(z)}$ is nondecreasing and nonnegative on $[0,1]$, we have that $\frac{f(z_0)}{g(z_0)} = 0$ for every $z_0 \in \left[0, 1-2^{-p(N)}\right]$. Since $\frac{f(z)}{g(z)}$ is a rational function and $1 - 2^{-p(N)} > 0$, this implies $\frac{f(z)}{g(z)}$ is identically 0, and hence $\frac{f(1)}{g(1)} = 0$ as required. ∎

Proof of Lemma 3.2. For each input x, let $E(x)$ and m be as in Lemma 3.6, write $N = 2^{2s} + 2$, and let p be as in Lemma 3.10. Define $F(x)$ and $G(x)$ to be $N \times N$ integer matrices as follows.

$$G(x) = m\, 2^{p(N)} I - \left(2^{p(N)} - 1\right) m\, E(x),$$

and

$$F(x) = \begin{bmatrix} G(x)_{1,N} & 0 \\ 0 & -2\, m\, 2^{p(N)} \end{bmatrix}.$$

We have

$$\begin{aligned} \frac{\det(F(x))}{\det(G(x))} &= -2\, m\, 2^{p(N)} \frac{\det(G(x)_{1,N})}{\det(G(x))} \\ &= -2 \frac{\det\left(\left(I - \left(1-2^{-p(N)}\right) E(x)\right)_{1,N}\right)}{\det\left(I - \left(1-2^{-p(N)}\right) E(x)\right)} \\ &= 2\left(I - \left(1-2^{-p(N)}\right) E(x)\right)^{-1}[N,1]. \end{aligned}$$

Hence, by Lemma 3.10, $\frac{\det(F(x))}{\det(G(x))} > 1$ if and only if $\lim_{z\uparrow 1}(I - zE(x))^{-1}[N,1] > \frac{1}{2}$, and $\det(F(x)) = 0$ if and only if $\lim_{z\uparrow 1}(I - zE(x))^{-1}[N,1] = 0$. For $|z| < 1$, we have

$$(I - zE(x))^{-1}[N,1] \;=\; \sum_{k\geq 0} z^k E(x)^k[N,1],$$

since $E(x)$ has eigenvalues bounded in absolute value by 1, from which we conclude

$$\lim_{z\uparrow 1}(I - zE(x))^{-1}[N,1] \;=\; \text{Prob}[M \text{ accepts } x]$$

by Lemma 3.6. (Note here that

$$\lim_{z\uparrow 1}\sum_{k\geq 0} z^k E(x)^k[N,1] \;=\; \sum_{k\geq 0} E(x)^k[N,1],$$

as $\sum_{k\geq 0} E(x)^k[N,1]$ is clearly a convergent series.) It follows that $\frac{\det(F(x))}{\det(G(x))} > 1$ if and only if M accepts x with probability greater than $\frac{1}{2}$. We do not know what the signs of $\det(F(x))$ and $\det(G(x))$ are, and so we define

$$A(x) = \begin{bmatrix} F(x) & 0 \\ 0 & F(x) \end{bmatrix}, \quad B = \begin{bmatrix} G(x) & 0 \\ 0 & G(x) \end{bmatrix}.$$

Now we have $\det(A(x)) > \det(B(x))$ if and only if M accepts x with probability exceeding $\frac{1}{2}$, and $\det(A(x)) = 0$ if and only if M accepts x with probability 0.

It remains to show item 3 is satisfied. First, note that if we have integers a and b of length $2^{O(s)}$, and each bit of a and b is computable in space $O(s)$, then each bit in the sum and difference of a and b is computable in space $O(s)$ as well; this follows from the fact that addition and subtraction have log-space uniform Boolean circuits of logarithmic depth (see [22]) along with the well-known results of [6] relating circuit depth to deterministic Turing machine space.

Now, for given integers i, j and l, we may assume the lth bit of the (i,j)-entry of $m\,E(x)$ can be computed in space $O(s)$ by Lemma 3.6. Naturally, the same may be said of $m\,2^{g(N)}E(x)$ and $m\,2^{g(N)}I$, since $g(N)$ is computable in space $O(s)$. As a result, we have that the lth bit in the (i,j)-entry of $G(x)$ can be computed in space $O(s)$, given the above facts concerning addition and subtraction. Similarly, this holds for $F(x)$. Finally, computation of each bit in the encoding of $(A(x), B(x))$ follows in straightforward fashion, as all remaining arithmetic can clearly be performed in $O(s)$ space. This completes the proof of Lemma 3.2. ■

4 Relationships Following from Quantum Simulations

In this section, we prove a number of facts concerning space-bounded quantum complexity classes that follow from QTM simulations of other QTMs and of PTMs. We state applicable facts regarding the various simulations we consider as lemmas in the present section; proofs of these lemmas, which include technical details regarding the simulations themselves, may be found in Section 5.

4.1 Basic Quantum Simulations

We begin by considering two lemmas that describe rather basic types of simulations, the first regarding QTMs that simulate the computation of a given QTM for some specified number of steps and then halt in some predetermined manner, and the second regarding QTMs that repeatedly simulate some specified number of steps of the computation of a given QTM in order to amplify its probabilities of accepting and rejecting. The primary use of these lemmas will be in conjunction with other lemmas presented later in this section.

Lemma 4.1 *Let M be a QTM running in space s and let $t : \mathbb{Z}^+ \to \mathbb{N}$ be computable in space $O(s)$. Let $p_{acc}(x)$ and $p_{rej}(x)$ denote the cumulative probabilities that M accepts and rejects, respectively, input x after t steps have passed. Then for any choice of $\alpha \in \{0, 1\}$ and $\beta \in \{0, \frac{1}{2}\}$ there exists a QTM M' running in space $O(s)$ and $t' : \mathbb{Z}^+ \to \mathbb{N}$ computable in space $O(s)$ such that the following hold for each input x.*

1. *The probability that M' halts within the first $t' - 1$ steps of its computation is 0.*
2. *After precisely t' steps, M' accepts x with probability $p_{acc}(x)$ and rejects x with probability $\alpha\, p_{rej}(x)$.*
3. *After precisely $t' + 1$ steps, M' accepts x with probability β and rejects x with probability $1 - \beta$ (conditioned upon M' not halting within the first t' steps of its computation).*

Note that, for given M, the machine M' resulting from this lemma halts absolutely, having cumulative probability of $\beta + (1 - \beta)\, p_{acc}(x) - \alpha\, \beta\, p_{rej}(x)$ for acceptance and $(1 - \beta) - (1 - \beta)\, p_{acc}(x) + \alpha\, \beta\, p_{rej}(x)$ for rejection.

As indicated above, Lemma 4.1 will be most useful later in this section when combined with other simulation techniques. We may, however, deduce two very simple relationships from this lemma. The first,

$$\mathrm{NQ}_H\mathrm{SPACE}(s) \subseteq \mathrm{PrQ}_H\mathrm{SPACE}(s),$$

results by taking $\alpha = 1$, $\beta = \frac{1}{2}$. (This containment also follows from results proved below.) The second relationship is as follows.

Proposition 4.2 *$\mathrm{NQ}_H\mathrm{SPACE}(s) = \mathrm{NQSPACE}(s)$.*

Proof. For the nontrivial containment, take M to be a QTM running in space s, and for given input x let $|\psi_k\rangle = (U_x P_\varepsilon)^k |c_0\rangle$ for each $k \geq 0$. Under the assumption that M runs in space s, there exists a subspace of $\ell_2(\mathcal{C})$ of dimension 2^s that contains every $|\psi_k\rangle$. Hence, if k is the largest number such that $|\psi_k\rangle \notin \mathrm{span}\{|\psi_0\rangle, \ldots, |\psi_{k-1}\rangle\}$, then $k < 2^s$. It follows that if $P_1 |\psi_k\rangle \neq 0$ for any $k \geq 0$, then $P_1 |\psi_k\rangle \neq 0$ for some $k < 2^s$. Now apply Lemma 4.1 with $t = 2^s$ and $\beta = 0$. ■

In the classical case, it is known that $\mathrm{NSPACE}(s) = \mathrm{RSPACE}(s)$ [12]; a space-bounded probabilistic machine can simulate a nondeterministic machine by repeatedly choosing random computation paths until it inevitably picks an accepting path (if there is such a path). It is not immediately clear that a similar result holds in the quantum case, since restarting a quantum machine likely constitutes an irreversible action not performable by a QTM. The second lemma of this section states that a process having a similar outcome can be performed by a QTM.

Lemma 4.3 *Let M be a QTM running in space s and let $t : \mathbb{Z}^+ \to \mathbb{N}$ be computable in space $O(s)$. Let $p_{acc}(x)$ and $p_{rej}(x)$ denote the cumulative probabilities that M accepts and rejects, respectively, input x after t steps have passed. Then there exists a QTM M' running in space $O(s)$ such that for each input x, if $p_{acc}(x) + p_{rej}(x) > 0$ then M' accepts with probability $\frac{p_{acc}(x)}{p_{acc}(x)+p_{rej}(x)}$ and rejects with probability $\frac{p_{rej}(x)}{p_{acc}(x)+p_{rej}(x)}$, and if $p_{acc}(x) + p_{rej}(x) = 0$ then M' accepts and rejects with probability 0.*

We now have the following theorem, based on Lemmas 4.1 and 4.3.

Theorem 4.4 *EQSPACE(s) = RQSPACE(s) = NQSPACE(s).*

Proof. As EQSPACE$(s) \subseteq$ RQSPACE$(s) \subseteq$ NQSPACE(s) follows by definition, it suffices to prove NQSPACE$(s) \subseteq$ EQSPACE(s).

Assume $L \in$ NQSPACE(s). Let M be a QTM running in space $O(s)$ accepting inputs in L with nonzero probability and accepting inputs not in L with zero probability. Define $t = 2^{a \cdot s}$ for a sufficiently large constant a, and let $p_{acc}(x)$ denote the probability M accepts input x after t steps. As is shown in the proof of Proposition 4.2, $p_{acc}(x) > 0$ if and only if $x \in L$. By Lemma 4.1, there exists a QTM M' and t' computable in space $O(s)$ such that M' accepts each x with probability $p_{acc}(x)$ and rejects each input x with probability 0 after t' steps. Applying Lemma 4.3 to M' together with t', we see that there exists a QTM M'' running in space $O(s)$ that accepts inputs in L with probability 1 and accepts inputs not in L with probability 0. Hence $L \in$ EQSPACE(s). ■

4.2 QTMs Halting Almost Surely

The non-halting space-bounded classes EQSPACE(s), RQSPACE(s), BQSPACE(s), and PrQSPACE(s) are defined in terms of quantum Turing machines having no restrictions on the probabilities with which various input strings are rejected. Consequently, a particular language may be in one of these classes by virtue of a machine that does not necessarily halt almost surely on some or all inputs. Here, we prove that the classes RQSPACE(s), BQSPACE(s), and PrQSPACE(s) do not change if the quantum Turing machines defining these classes are required to halt almost surely on all inputs, i.e.,

$$\begin{aligned} \text{PrQSPACE}(s) &= \text{PrQ}_{AS}\text{SPACE}(s), \\ \text{BQSPACE}(s) &= \text{BQ}_{AS}\text{SPACE}(s), \\ \text{RQSPACE}(s) &= \text{RQ}_{AS}\text{SPACE}(s). \end{aligned}$$

The method used to prove these relations is similar to one used in [19] for the classical versions of these results; in the present case, the notion of a *probabilistic clock* is replaced by a suitable quantum analogue. As a corollary, we have that BQSPACE(s) is closed under complementation.

These results follow from the next two lemmas.

Lemma 4.5 *Let M be a quantum Turing machine running in space s. Then there exists a polynomial h satisfying the following. For each input x, if M accepts x with probability exceeding $1/2$, then M accepts x with probability exceeding*

$$\frac{1}{2} + 2^{-h(2^s)}.$$

Proof. For each input x, let $E(x)$ be as in Lemma 3.6 for M on input x, and define polynomials f and g for $E(x)$ as in the proof of Lemma 3.10. Recall that $N = 2^{2s} + 2$ and m is fixed for given M. As in (2), we have

$$\text{Prob}[M \text{ accepts } x] - \frac{1}{2} \;=\; \frac{f(1)}{g(1)} - \frac{1}{2} \;\geq\; \left|\frac{1}{2\,g(1)}\right| \;\geq\; \frac{1}{(N+1)!\,m^N 2^{2N}}$$

whenever $\text{Prob}[M \text{ accepts } x] > \frac{1}{2}$. Let h be a polynomial satisfying $h(2^s) > \log\left((N+1)!\,m^N 2^{2N}\right)$ for every input x (e.g., $h(n) = n^4 + (17 \log m)\, n^2$). Then

$$\text{Prob}[M \text{ accepts } x] \;>\; \frac{1}{2} + 2^{-h(2^s)}$$

whenever $\text{Prob}[M \text{ accepts } x] > \frac{1}{2}$, as required. ■

Lemma 4.6 *Let M be a QTM running in space s, and for each input x let $p_{acc}(x)$ denote the probability that M accepts x. Then for any polynomial h there exists a QTM M' running in space $O(s)$ such that the following hold.*

1. M' accepts each input x with probability $p'_{acc}(x)$ satisfying $p_{acc}(x) - 2^{-h(2^s)} \;\leq\; p'_{acc}(x) \;\leq\; p_{acc}(x)$.

2. M' halts almost surely.

From this lemma, we may conclude the following.

Theorem 4.7 *The following equalities hold.*

$$\begin{aligned} PrQSPACE(s) &= PrQ_{AS}SPACE(s), \\ BQSPACE(s) &= BQ_{AS}SPACE(s), \\ RQSPACE(s) &= RQ_{AS}SPACE(s). \end{aligned}$$

Corollary 4.8 *$BQSPACE(s)$ is closed under complementation.*

We note that closure of PrQSPACE(s) under complementation may be proved in a similar way (assuming we take care when dealing with the case that a given QTM accepts with probability precisely 1/2). However, closure of PrQSPACE(s) under complementation will follow more easily from facts proved below.

4.3 Quantum Simulations of PTMs

Let us say that a probabilistic Turing machine M is *well-behaved* if the following hold.

1. For every input, there is at most one accepting and one rejecting configuration of M reachable with nonzero probability from the initial configuration.
2. There exists $t : \mathbb{Z}^+ \to \mathbb{N}$ such that, on each input x, M halts after precisely $t(|x|)$ steps on all computation paths.

Note that any well-behaved probabilistic Turing machine necessarily halts absolutely. Furthermore, if a given well-behaved probabilistic Turing machine runs in space $O(s)$, the function t in item 2 is computable in space $O(s)$ as well. Finally, note that the classes PrSPACE(s), BP$_H$SPACE(s), and co-C$_=$SPACE(s) remain unchanged if the underlying machine is required to be well-behaved (following from [1, 16] in the case of PrSPACE(s)).

Lemma 4.9 *Let M be a well-behaved PTM running in space s, and let $p_{acc}(x)$ and $p_{rej}(x)$ denote the probabilities that M accepts x and rejects x, respectively. Then there exist a QTM M' running in space $O(s)$ and $t' : \mathbb{Z}^+ \to \mathbb{N}$ computable in space $O(s)$ such that for each input x, M' accepts x with probability $\left(2^{-2st}\, p_{acc}(x)\right)^2$ and rejects x with probability $\left(2^{-2st}\, p_{rej}(x)\right)^2$ after t' steps.*

For a given PTM M, we may apply Lemma 4.1 (with $\alpha = 1$ and $\beta = 1/2$) to the QTM M' resulting from Lemma 4.9, and we see that there exists a QTM M'' that halts absolutely and has probability of acceptance greater than $1/2$ if and only if $p_{acc}(x) > p_{rej}(x)$, yielding the following.

Theorem 4.10 *PrSPACE(s) $\subseteq$ PrQ$_H$SPACE(s).*

By Theorems 3.3 and 4.10, we have the following corollary.

Corollary 4.11 *PrSPACE(s) $=$ PrQSPACE(s) $=$ PrQ$_H$SPACE(s).*

In the case that M has probability of error bounded away from $1/2$, we may apply Lemmas 4.1 and 4.3 to M' to obtain a QTM that accepts with probability $\frac{p_{acc}(x)^2}{p_{acc}(x)^2+p_{rej}(x)^2}$ and rejects with probability $\frac{p_{acc}(x)^2}{p_{acc}(x)^2+p_{rej}(x)^2}$. These probabilities are bounded at least as far from $1/2$ as $p_{acc}(x)$ and $p_{rej}(x)$, and consequently the following relationship holds.

Theorem 4.12 *BP$_H$SPACE(s) $\subseteq$ BQSPACE(s).*

We note that the QTM M' constructed in the proof of Lemma 4.9 accepts with nonzero probability if and only if the same is true of the PTM M, and hence the containment NSPACE(s) $\subseteq$ NQSPACE(s) immediately follows. However, it is possible to obtain a stronger result by means of the following lemma.

Lemma 4.13 *Let M be a well-behaved PTM running in space s and let $p_{acc}(x)$ and $p_{rej}(x)$ denote the probabilities that M accepts x and rejects x, respectively. Then there exists a QTM M' running in space $O(s)$ such that, for each input x, M' accepts x with probability 0 if and only if $p_{acc}(x) = p_{rej}(x)$.*

Theorem 4.14 *co-C$_=$SPACE(s) $\subseteq$ NQSPACE(s).*

By Theorems 3.5, 4.4 and 4.14, we have the following.

Corollary 4.15 *EQSPACE(s) $=$ RQSPACE(s) $=$ NQSPACE(s) $=$ co-C$_=$SPACE(s).*

5 Quantum Simulations

The purpose of the current section is to prove the lemmas regarding QTM simulations stated in the previous section. The proofs appear in Section 5.3 below. Since it would not be justifiable at this point for us to describe space-bounded QTMs in the high-level manner that is typical in the classical case

without first providing a formal basis for such descriptions, Sections 5.1 and 5.2 are devoted to providing such a basis. In Section 5.1, we define two types of primitive operations—reversible transformations and quantum transformations—that will be the building blocks for more complex QTMs, and discuss how these transformations may be composed. In Section 5.2 we prove a fact concerning reversible transformations that will simplify greatly the task of analyzing the behavior of the QTMs presented in Section 5.3.

5.1 Specification of Quantum Turing Machines

In order to construct QTMs performing various simulations, we will compose simpler QTMs performing certain primitive operations that we call *transformations*. The two types of transformations we consider are *reversible transformations* and *quantum transformations*.

Before discussing transformations, let us first introduce some additional notation used throughout the present section, and also make explicit certain assumptions we will make.

The contents of the work tape of a given QTM having work tape alphabet Γ may be described by a mapping of the form $y : \mathbb{Z} \to \Gamma$ for which $y(i) = \#$ for all but finitely many $i \in \mathbb{Z}$. We define $W(\Gamma)$ to be the set of all such mappings for a given alphabet Γ, and for any nonnegative integer m we let $W_m(\Gamma)$ denote the subset of $W(\Gamma)$ consisting of all mappings y for which $y(i) = \#$ whenever $|i| > m$. Let $\widehat{\#}$ be the mapping satisfying $\widehat{\#}(i) = \#$ for all $i \in \mathbb{Z}$.

Frequently it will be useful to consider machines whose work tapes have multiple tracks. A QTM having k tracks on its work tape is assumed to have a work tape alphabet of the form $\Gamma = \Gamma_1 \times \cdots \times \Gamma_k$. If, for given $y \in W(\Gamma)$, we have $y(i) = (y_1(i), \ldots, y_k(i))$ for each $i \in \mathbb{Z}$, then $y_j \in W(\Gamma_j)$ is a mapping that specifies the contents of the jth track. For brevity we write $y = (y_1; y_2; \ldots; y_k)$ in this situation.

All of the quantum simulations we present require storage and manipulation of integers. We assume that the following scheme is used to encode integers.

1. The empty string ε represents 0.
2. Each string of the form $0w$ represents the positive integer with absolute value having binary representation $1w$.
3. Each string of the form $1w$ represents the negative integer with absolute value having binary representation $1w$.

This encoding induces a one-to-one correspondence between the integers and the set $\{0,1\}^*$. When we say that a particular track on the work tape of some machine encodes an integer, we assume that the corresponding binary string is written on that track beginning in square 0 (and all other squares on that track contain blanks). Note that a track containing only blanks therefore encodes 0.

Finally, it will be useful to define an addition operation on finite sets and alphabets, for the purpose of describing certain machines more succinctly. For a given set A, we define addition on A by simply identifying elements of A with elements of the additive group of integers modulo $|A|$ (via some arbitrary one-to-one correspondence). The element of A corresponding to 0 is thus the unique identity element of A with respect to this operation. Addition on alphabets is defined similarly, and in this case we always assume that the blank symbol # corresponds to 0. For a given alphabet Γ, addition on Γ is extended to $W(\Gamma)$ pointwise.

We now discuss the first type of primitive operation: the reversible transformation. Formally, a reversible transformation is a one-to-one and onto mapping of the form $\Phi : A \times W_m(\Gamma) \to A \times W_m(\Gamma)$ for some finite set A, alphabet Γ, and integer $m \geq 0$. Below, the sets A and $W_m(\Gamma)$ will be represented

in the internal state and on the work tape of a given Turing machine (since it will be convenient to allow a reversible transformation to modify both the internal state and work tape contents).

Let us now define what it means for a reversible transformation to be performed by a particular Turing machine. For a given finite set A and alphabet Γ, consider a DTM M having internal state set $G \times A$, for some set G containing distinct elements g_0 and g_f, and work tape alphabet $\Gamma' \supseteq \Gamma$. (We will indulge in a slight abuse of terminology and say that g_0 and g_f are the initial and final states of M.) Configurations of M will be expressed in the form $((g,a),h_i,y,h_w)$ for $(g,a) \in G \times A$ denoting the internal state of M, $h_i \in \mathbb{Z}$ denoting the input tape head position of M, $y \in W(\Gamma')$ specifying the contents of M's work tape, and $h_w \in \mathbb{Z}$ denoting the work tape head position of M. Unless otherwise noted, the output tape will be assumed to contain only blanks. Now, for a given input x, we say that M on input x performs the reversible transformation Φ on $A \times W_m(\Gamma)$ if the following holds. For every $(a,y) \in A \times W_m(\Gamma)$, there exists an integer $k = k(a,y)$ such that, if M is placed in configuration $((g_0,a),0,y,0)$ and is run for precisely k steps, it will then be in configuration $((g_f,a'),0,y',0)$, for $(a',y') = \Phi(a,y)$. Furthermore, for each $j \in \{1,\ldots,k-1\}$, the configuration reached after running M for j steps starting on configuration $((g_0,a),0,y,0)$ must be a non-halting configuration having internal state in the set $(G \backslash \{g_0,g_f\}) \times A$. (Thus, M does not produce output during this computation).

Naturally, we say that k is the number of steps required for M on x to perform transformation Φ on argument (a,y). The space required for M on x to perform Φ on argument (a,y) is defined in the same way as for an ordinary computation.

Next, let us consider RTMs that perform reversible transformations. Recall that we define RTMs to be QTMs having amplitudes restricted to the set $\{0,1\}$, so that the transition function of a given RTM may be specified by a collection of operators $\{V_\sigma : \sigma \in \Sigma\}$, together with functions D_i, D_w, and Z, as in the case of more general QTMs. We restrict our attention to machines that satisfy the following properties (in addition to having state set of the form $G \times A$ with $g_0, g_f \in G$ and work tape alphabet $\Gamma' \supseteq \Gamma$, for given A and Γ).

1. We have $Z((g,a)) = \varepsilon$ for every $(g,a) \in G \times A$, i.e., no output is produced by the machine.
2. For every $a \in A$ and $\tau \in \Gamma'$, we have $V_\sigma \,|(g_f,a),\tau\rangle = |(g_0,a),\tau\rangle$.
3. For every $a \in A$ we have $D_i((g_0,a)) = D_w((g_0,a)) = D_i((g_f,a)) = D_w((g_f,a)) = 0$.

An RTM satisfying these properties will be called a *reversible transformation machine*. These are essentially technical properties that will be helpful shortly.

Now, we say that a reversible transformation machine M on input x performs the reversible transformation Φ on $A \times W_m(\Gamma)$ if for every $(a,y) \in A \times W_m(\Gamma)$ there exists a positive integer $k = k(a,y)$ such that

$$U_x^k \,|(g_0,a),0,y,0\rangle \;=\; \big|(g_f,a'),0,y',0\big\rangle\,,$$

where $(a',y') = \Phi(a,y)$ and U_x denotes the time-evolution operator of M on input x. By item 1 above, we have that $U_x^j \,|(g_0,a),0,y,0\rangle$ must be a classical state corresponding to a non-halting configuration of M for every j.

The following lemma is a restatement of the main result of [18], phrased in terms of reversible transformations.

Lemma 5.1 (Lange, McKenzie & Tapp) *Let A be a finite set, let Σ and Γ be finite alphabets, and let s be any space bound. For each $x \in \Sigma^*$, let Φ_x be a reversible transformation on $A \times W_{s(|x|)}(\Gamma)$, and*

suppose that there exists a DTM M_0 that performs Φ_x on input x and requires space $O(s)$. Then there exists a reversible transformation machine M that, on input x, performs Φ_x in space $O(s)$.

It is difficult to overstate the importance of this result to the current section—whenever it is necessary to specify a reversible transformation machine for a particular reversible transformation, we may essentially ignore the reversibility constraint, specify an ordinary DTM for the task at hand, and apply this lemma. There is, however, one problem arising from the machines resulting from Lemma 5.1 in regard to the applications we have in mind, which is that little can be said about the number of steps these machines require to perform their transformations for different arguments. Since this information will generally be needed for the analysis of the quantum machines we construct, we will also make use the following lemma; this lemma will essentially allow us to view reversible transformations as requiring unit time when analyzing machines later in this section.

Lemma 5.2 *Let A be a finite set, let Σ and Γ be finite alphabets, and let s be a space bound. For each input $x \in \Sigma^*$, let Φ_x be a reversible transformation on $A \times W_{s(|x|)}(\Gamma)$, and assume that there exists a reversible transformation machine M_0 that, on input x, performs Φ_x in space $O(s)$. Then there exists a reversible transformation machine M that, on input x, performs Φ_x in space $O(s)$ and requires a number of steps that is independent of the argument of Φ_x.*

Section 5.2 is devoted to the proof of Lemma 5.2. It should be noted that our proof of Lemma 5.2 relies on the fact that the space bound s is space-constructible and satisfies $s(n) = \Omega(\log n)$, while these restrictions are not required for Lemma 5.1.

Next, we discuss quantum transformations. Unlike reversible transformations, quantum transformations consist of a single step and do not involve the contents of either the input tape or work tape of the machine in question. Also unlike reversible transformations, quantum transformations may involve writing symbols to the output tape.

A quantum transformation is a unitary operator

$$\Lambda : \ell_2(A) \to \ell_2(A), \tag{4}$$

where A is some finite set (which again will be represented by the internal state of a Turing machine). A QTM M is said to perform transformation Λ if the following hold.

1. The state set of M is $\{g_0, g_f\} \times A$.
2. For every $\sigma \in \Sigma$, $a \in A$, and $\tau \in \Gamma$ we have

$$\begin{aligned} V_\sigma \left|(g_0, a), \tau\right\rangle &= \sum_{a'} \left\langle a' \,|\, \Lambda \,|\, a \right\rangle \left|(g_f, a'), \tau\right\rangle, \\ V_\sigma \left|(g_f, a), \tau\right\rangle &= \left|(g_0, a), \tau\right\rangle, \end{aligned}$$

and $D_i((g_0, a)) = D_i((g_f, a)) = D_w((g_0, a)) = D_w((g_f, a)) = 0$.

A QTM satisfying these properties will be called a *quantum transformation machine*. Any quantum transformation of the form (4) can be performed by a quantum transformation machine provided $\langle a' \,|\, \Lambda \,|\, a \rangle$ is rational for each a and a'.

Finally, we describe how reversible transformations and quantum transformations may be composed to form more complex QTMs. Fix a space bound s, input and work tape alphabets Σ and Γ, and a finite

set A, and suppose we have a collection of reversible/quantum transformation machines $M^{(1)}, \ldots, M^{(k)}$, each having input alphabet Σ and performing, on given input x, either a reversible transformation on $A \times W_{s(|x|)}(\Gamma)$ or a quantum transformation on $\ell_2(A)$. We wish to compose these machines to form a single QTM M in some way.

Assuming the state set of each $M^{(j)}$ is denoted $G^{(j)} \times A$, let the state set of M be $G \times A$ for

$$G = \bigcup_{j=1}^{k} G^{(j)}.$$

Here, we view the $G^{(j)}$ state sets as being disjoint. The input tape and work tape alphabets of M are taken to be Σ and some suitable superset of Γ, respectively. Now, the transition function of M may be specified by $\{V_\sigma : \sigma \in \Sigma\}$, D_i, D_w, and Z in any manner consistent with the following. Under the assumption that the transition function of each $M^{(j)}$ is specified by $\left\{V_\sigma^{(j)} : \sigma \in \Sigma\right\}$, $D_i^{(j)}$, $D_w^{(j)}$, and $Z^{(j)}$, for each $j = 1, \ldots, m$ we define

$$\begin{aligned} V_\sigma \left|(g,a),\tau\right\rangle &= V_\sigma^{(j)} \left|(g,a),\tau\right\rangle, \\ D_i((g,a)) &= D_i^{(j)}((g,a)), \\ D_w((g,a)) &= D_w^{(j)}((g,a)), \end{aligned}$$

for every $\sigma \in \Sigma$, $g \in G^{(j)} \setminus \left\{g_f^{(j)}\right\}$, and $a \in A$; $Z((g,a)) = Z^{(j)}((g,a))$ for every $g \in G^{(j)}$ and $a \in A$; and $D_i((g,a)) = D_w((g,a)) = 0$ for each $g \in \left\{g_f^{(j)} : j = 1, \ldots, m\right\}$ and $a \in A$. Finally, for any chosen collection $\{\Delta_{a,\tau} : a \in A, \tau \in \Gamma\}$ of one-to-one and onto mappings of the form

$$\Delta_{a,\tau} : \left\{g_f^{(j)} : j = 1, \ldots, m\right\} \to \left\{g_0^{(j)} : j = 1, \ldots, m\right\},$$

define

$$V_\sigma \left|\left(g_f^{(j)}, a\right), \tau\right\rangle = \left|\left(\Delta_{a,\tau}\left(g_f^{(j)}\right), a\right), \tau\right\rangle$$

for every $j = 1, \ldots, m$ and $\sigma \in \Sigma$.

The resulting QTM M mimics the behavior of each machine $M^{(j)}$ when in states $\left(G^{(j)} \setminus \left\{g_f^{(j)}\right\}\right) \times A$; for the remaining states, state transitions are performed according to the mappings $\Delta_{a,\tau}$ (while the input and work tape heads remain stationary during such transitions). Given that each $M^{(j)}$ is either a valid reversible transformation machine running in space $O(s)$ or a quantum transformation machine, it is straightforward to verify that M is a valid QTM (i.e., each V_σ is unitary) also running in space $O(s)$.

The one-to-one and onto mappings $\Delta_{a,\tau}$ determine the "flow" of M's computation between transformations corresponding to $M^{(1)}, \ldots, M^{(k)}$. There are two basic constructs that we use in the following subsections that may be induced by such mappings; the first is simply a collection of transformations applied in sequence, and the second is a loop with a single *starting/stopping condition*. (I.e., there is a single condition that is tested immediately before each iteration of the loop. If the condition is met, the machine either enters or exits the loop, depending upon whether the machine was previously outside or inside the loop, respectively, while if the condition is not met, the machine remains inside or outside the loop as it was previous to testing the condition. In general, this type of loop is reversible.)

It is perhaps easiest to describe how these constructs may be implemented by giving a simple example; this example can be generalized to compose transformations as needed. Suppose we have reversible/quantum transformation machines $M^{(1)}, \ldots, M^{(4)}$ that we wish to compose in the manner described in Figure 1. To simplify this example, suppose that the transformations $M^{(1)}, \ldots, M^{(4)}$

Repeat ad infinitum:

1. Loop with starting/stopping condition "work tape encodes 0":
   Perform transformations corresponding to $M^{(1)}, \ldots, M^{(3)}$ in sequence.
2. Perform the transformation corresponding to $M^{(4)}$.

Figure 1: Example composition of transformations.

preserve the property that their work tape encodes an integer, given that this is initially true (so the condition that the work tape encodes 0 is equivalent to square 0 on the work tape containing #). First, let $M^{(0)}$ be a reversible transformation machine that does nothing (i.e., performs the identity transformation). Now define

$$\begin{aligned}
&\Delta_{a,\#}\left(g_f^{(0)}\right) = g_0^{(1)}, \qquad \Delta_{a,\#}\left(g_f^{(1)}\right) = g_0^{(2)}, \qquad \Delta_{a,\#}\left(g_f^{(2)}\right) = g_0^{(3)}, \\
&\Delta_{a,\#}\left(g_f^{(3)}\right) = g_0^{(4)}, \qquad \Delta_{a,\#}\left(g_f^{(4)}\right) = g_0^{(0)},
\end{aligned}$$

for every $a \in A$, and

$$\begin{aligned}
&\Delta_{a,\tau}\left(g_f^{(0)}\right) = g_0^{(4)}, \qquad \Delta_{a,\tau}\left(g_f^{(1)}\right) = g_0^{(2)}, \qquad \Delta_{a,\tau}\left(g_f^{(2)}\right) = g_0^{(3)}, \\
&\Delta_{a,\tau}\left(g_f^{(3)}\right) = g_0^{(1)}, \qquad \Delta_{a,\tau}\left(g_f^{(4)}\right) = g_0^{(0)},
\end{aligned}$$

for every $\tau \neq \#$ and $a \in A$. Setting the initial state of M to be $\left(g_0^{(0)}, a\right)$ for chosen $a \in A$, the reader may verify that the transformations are performed as described in Figure 1.

The proof of Lemma 5.2 found in the next subsection further illustrates how transformations may be composed.

5.2 Proof of Lemma 5.2

This subsection is devoted to a proof of the following lemma from the previous subsection.

Lemma 5.2 *Let A be a finite set, let Σ and Γ be finite alphabets, and let s be a space bound. For each input $x \in \Sigma^*$, let Φ_x be a reversible transformation on $A \times W_{s(|x|)}(\Gamma)$, and assume that there exists a reversible transformation machine M_0 that, on input x, performs Φ_x in space $O(s)$. Then there exists a reversible transformation machine M that, on input x, performs Φ_x in space $O(s)$ and requires a number of steps that is independent of the argument of Φ_x.*

The proof relies on a number of simple lemmas, which we now state and prove.

Lemma 5.3 *For any reversible transformation machine M, there exists a reversible transformation machine M^{-1} such that the following holds. For each input x, if M on input x performs transformation Φ_x on $A \times W_m(\Gamma)$, then M^{-1} performs transformation Φ_x^{-1} on $A \times W_m(\Gamma)$. Furthermore, for each $(a, y) \in A \times W_{s(|x|)}(\Gamma)$, the time and space required by M to perform Φ_x on (a, y) is precisely the same as the time and space required by M^{-1} to perform Φ_x^{-1} on $\Phi_x(a, y)$.*

Proof. Let the transition function of M be specified by V_σ, $D_i(q)$ and $D_w(q)$ for each $\sigma \in \Sigma$ and $q \in G \times A$. M^{-1} will have the same input alphabet, work tape alphabet, and state set as M, with $g_0^{(-1)} = g_f$ and $g_f^{(-1)} = g_0$ denoting the initial and final state of M^{-1}, respectively. The transition function of M^{-1} is be specified by $V_\sigma^{(-1)}$, $D_i^{(-1)}(q)$ and $D_w^{(-1)}(q)$, for each $\sigma \in \Sigma$ and $q \in G \times A$, as follows: $V_\sigma^{(-1)} = V_\sigma^{-1}$, $D_i^{(-1)}(q) = -D_i(q)$, and $D_w^{(-1)}(q) = -D_w(q)$. Given that M is a reversible transformation machine, we have $D_i(g_0, a) = D_w(g_0, a) = D_i(g_f, a) = D_w(g_f, a) = 0$ for every $a \in A$. From this, it is straightforward to verify that M^{-1} inverts the transformation performed by M. ■

Lemma 5.4 *Let A be a finite set, let Γ be a finite alphabet, and let s be a space bound. Then there exists a reversible transformation machine M such that the following holds. For each $n \geq 0$, there exists an enumeration $(a_1, y_1), \ldots, (a_l, y_l)$ of $A \times W_{s(n)}(\Gamma)$ such that M performs the reversible transformation*

$$\Phi_n(a_j, y_j) = \begin{cases} (a_{j+1}, y_{j+1}) & \text{if } 1 \leq j \leq l-1 \\ (a_1, y_1) & \text{if } j = l \end{cases}$$

on any input x for which $|x| = n$, and furthermore M performs this transformation in space $O(s)$.

Proof. Follows immediately from Lemma 5.1 along with a straightforward deterministic Turing machine specification. ■

Lemma 5.5 *Let A be a finite set, let Γ be a finite alphabet, and let s be a space bound. Define a reversible transformation Φ_n on $(A \times A \times \{0,1\}) \times W_{s(n)}(\Gamma \times \Gamma)$, for each $n \geq 0$, as follows.*

$$\Phi_n((a, a', b), (y; y')) = \begin{cases} ((a, a', 1-b), (y; y')) & \text{if } a = a' \text{ and } y = y' \\ ((a, a', b), (y; y')) & \text{otherwise.} \end{cases}$$

Then there exists a reversible transformation machine M that, on each input x, performs transformation $\Phi_{|x|}$ in space $O(s)$. Furthermore, the number of steps required for M on x to perform this transformation is independent of its argument.

Proof. We describe informally the machine M; as formal specification of M is straightforward, we omit the details.

M will have two auxiliary tracks on its work tape in addition to the two tracks containing y and y'. We may view $\Gamma \times \Gamma$ as being a subset of this new alphabet by identifying elements of $\Gamma \times \Gamma$ with those symbols having blanks on the auxiliary tracks.

The computation of M consists of a number of phases. In the first phase, M marks the squares indexed by $-s(|x|)$, 0, and $s(|x|)$ on the first auxiliary track. By Lemma 5.1, along with space-constructibility of s, such a transformation can be performed reversibly in space $O(s)$. The time for this transformation may be assumed to be independent of the argument of Φ_n, as the contents of tracks

1 and 2 may be ignored during this phase. We assume the input and work tape heads of M return to the squares indexed by 0 upon completion of this transformation, in accordance with Lemma 5.1.

In the next phase, the input tape head remains stationary and the work tape head simply moves left to the square indexed by $-s(|x|)$, sweeps through the contents of the work tape to the square indexed by $s(|x|)$, then returns to square 0. As these squares are marked on the first auxiliary track, this movement can be performed reversibly. We imagine that M has a pebble at the beginning of this phase, which it may pick up or put down on the second auxiliary track. As the tape head is sweeping from square $-s(|x|)$ to square $s(|x|)$, the pebble is put down or picked up whenever the contents of tracks 1 and 2 disagree. (In fact, the pebble is never picked up given the particular movement of the tape head we have described—the process is described in this way in order to illustrate that it is a reversible process.)

The third phase requires a single reversible step involving only the internal state of M: b is replaced with $1 - b$ whenever $a = a'$ and the pebble has not been placed anywhere on the tape.

Finally, the second and first phases are inverted, returning the auxiliary tracks to their initial blank state. The result is that M performs the required transformation in the manner claimed. ■

Lemma 5.6 *Let A be a finite set, let Γ be a finite alphabet, let s be a space bound, and let addition be defined on A and Γ as described in Section 5.1. For $n \geq 0$, define a reversible transformation Φ_n on $(A \times A \times \{0,1\}) \times W_{s(n)}(\Gamma \times \Gamma)$ as follows.*

$$\Phi_n((a, a', b), (y; y')) = \begin{cases} ((a, a + a', b), (y; y + y')) & \text{if } b = 1 \\ ((a, a', b), (y; y')) & \text{if } b = 0, \end{cases}$$

Then there exists a reversible transformation machine M that, on each input x, performs transformation $\Phi_{|x|}$ in space $O(s)$. Furthermore, the number of steps required for M on x to perform this transformation is independent of its argument.

Proof. As in the proof of Lemma 5.5 we describe M informally, as formal specification is straightforward. In addition to the two tracks initially containing y and y', the work tape of M will have one auxiliary track; $\Gamma \times \Gamma$ may be identified with a subset of the resulting tape alphabet as in the proof of Lemma 5.5.

The operation of M consists of three phases. The first phase is identical to the machine described in the proof of Lemma 5.5, i.e., squares $-s(|x|)$, 0, and $s(|x|)$ are marked on the auxiliary track. The tape head movement in the second phase is identical to the machine in the proof of Lemma 5.5 as well. However, rather than performing the operation with the pebble, M now performs a *controlled add* (where b is the control bit) from each symbol on track 1 onto the corresponding symbol on track 2 as squares $-s(|x|), \ldots, s(|x|)$ are visited. At any chosen time during this phase, a is added to a' in case $b = 1$. The final phase inverts the first phase, returning the auxiliary track to its initial state. ■

Lemma 5.7 *Let A be a finite set, let Γ be a finite alphabet, and let s be a space bound. For each $n \geq 0$, define a reversible transformation Φ_n on $(A \times A) \times W_{s(n)}(\Gamma \times \Gamma)$ as follows.*

$$\Phi_n((a, a'), (y; y')) = ((a', a), (y'; y)).$$

Then there exists a reversible transformation machine M that, on each input x, performs transformation $\Phi_{|x|}$ in space $O(s)$. Furthermore, the number of steps required for M on x to perform this transformation is independent of its argument.

1. Repeat the following for each possible value of $(a_2, y_2) \in A \times W_{s(|x|)}(\Gamma)$.
   - i. If $(a_1, y_1) = (a_2, y_2)$, set $b = 1 - b$.
   - ii. Perform Φ_x on (a_2, y_2).
   - iii. In case $b = 1$, add (a_2, y_2) to (a_3, y_3).
   - iv. Perform Φ_x^{-1} on (a_2, y_2).
   - v. If $(a_1, y_1) = (a_2, y_2)$, set $b = 1 - b$.
2. Swap (a_1, y_1) and (a_3, y_3).
3. Perform the inverse of step 1, substituting transformation Φ_x^{-1} for Φ_x, and Φ_x for Φ_x^{-1}.

Figure 2: Description of reversible transformation machine M' for Lemma 5.2.

Proof. Similar to the proof of Lemma 5.6, replacing the *controlled add* transitions with appropriately defined *swap* transitions. ■

Proof of Lemma 5.2. We first construct a reversible transformation machine M' that performs a transformation similar to Φ_x on each input x, then modify M' slightly to yield M satisfying the statement of the lemma.

Define $A' = A \times A \times A \times \{0, 1\}$ and $\Gamma' = \Gamma \times \Gamma \times \Gamma$. The transformation performed by M' on input x, denoted Φ'_x, will be a transformation on $A' \times W_{s(|x|)}(\Gamma')$ satisfying

$$\Phi'_x((a, 0, 0, 0), (y; \widehat{\#}; \widehat{\#})) \;=\; ((a', 0, 0, 0), (y'; \widehat{\#}; \widehat{\#})), \tag{5}$$

where $(a', y') = \Phi_x(a, y)$ and 0 denotes the identity element of A with respect to addition on A, as described in Section 5.1.

We denote the state set of M' by $G' \times A'$, where g'_0 and g'_f denote the initial and final states of M', respectively, and G' is specified below. The work tape of M' will consist of three tracks, each capable of storing symbols in Γ as well as any auxiliary symbols that may be needed to implement the various transformations described below. In general, elements of $A' \times W_{s(|x|)}(\Gamma')$ may be written in the form $((a_1, a_2, a_3, b), (y_1; y_2; y_3))$ for $a_1, a_2, a_3 \in A$, $b \in \{0, 1\}$, and $y_1, y_2, y_3 \in W_{s(n)}(\Gamma)$; whenever we refer to one of a_1, a_2, a_3, b, y_1, y_2 or y_3, it is assumed that we are referring to the particular value of a_1, a_2, etc., represented by M' at the instant in question.

The behavior of M' is described in Figure 2. In essence, M' performs the transformation Φ_x on each possible value of (a_2, y_2) in $A \times W_{s(|x|)}(\Gamma)$, and copies the resulting value to (a_3, y_3) in the case that (a_2, y_2) matched the argument (a, y) (stored in (a_1, y_1)) immediately before Φ_x was performed. This corresponds to step 1. During step 2, M' swaps (a_1, y_1) and (a_3, y_3), and during step 3, M' performs a process similar to step 1 in order to "erase" (a, y) from the third track and from the a_3 internal state component. Assuming the argument of M' is of the form in (5), the transformation performed by M' requires a number of steps independent of the argument (a, y). This is because Φ_x is applied to every possible argument, and the required procedures for copying, swapping, etc., can be performed in a number of steps independent of the original argument.

We now formalize this construction. For each input $x \in \Sigma^*$, we first define transformations $\Psi_x^{(1)}, \ldots, \Psi_x^{(19)}$ on $A' \times W_{s(|x|)}(\Gamma')$. Note that these transformations are not all distinct, but are simply numbered in a manner that will be convenient below.

1. Define

$$\Psi_x^{(1)}((a_1, a_2, a_3, b), (y_1; y_2; y_3)) = \begin{cases} ((a_1, a_2, a_3, 1-b), (y_1; y_2; y_3)) & \text{if } a_2 = 0 \text{ and } y_2 = \widehat{\#} \\ (a_1, a_2, a_3, b), (y_1; y_2; y_3)) & \text{otherwise,} \end{cases}$$

   and let $\Psi_x^{(2)} = \Psi_x^{(9)} = \Psi_x^{(11)} = \Psi_x^{(12)} = \Psi_x^{(19)} = \Psi_x^{(1)}$.

2. Define

$$\Psi_x^{(3)}((a_1, a_2, a_3, b), (y_1; y_2; y_3)) = \begin{cases} ((a_1, a_2, a_3, 1-b), (y_1; y_2; y_3)) & \text{if } a_1 = a_2 \text{ and } y_1 = y_2 \\ (a_1, a_2, a_3, b), (y_1; y_2; y_3)) & \text{otherwise,} \end{cases}$$

   and let $\Psi_x^{(7)} = \Psi_x^{(13)} = \Psi_x^{(17)} = \Psi_x^{(3)}$.

3. Define

$$\Psi_x^{(4)}((a_1, a_2, a_3, b), (y_1; y_2; y_3)) = ((a_1, a_2', a_3, b), (y_1; y_2'; y_3)),$$

   where $(a_2', y_2') = \Phi_x(a_2, y_2)$, and let $\Psi_x^{(16)} = \Psi_x^{(4)}$. Also let $\Psi_x^{(6)} = \Psi_x^{(14)} = \left(\Psi_x^{(4)}\right)^{-1}$.

4. Define

$$\Psi_x^{(5)}((a_1, a_2, a_3, b), (y_1; y_2; y_3)) = \begin{cases} ((a_1, a_2, a_2 + a_3, b), (y_1; y_2; y_2 + y_3)) & \text{if } b = 1 \\ ((a_1, a_2, a_3, b), (y_1; y_2; y_3)) & \text{if } b = 0, \end{cases}$$

   and let $\Psi_x^{(15)} = \left(\Psi_x^{(5)}\right)^{-1}$.

5. Define

$$\Psi_x^{(8)}((a_1, a_2, a_3, b), (y_1; y_2; y_3)) = ((a_1, a_2', a_3, b), (y_1; y_2'; y_3)),$$

   where (a_2', y_2') denotes the successor of (a, y) according to the enumeration of $A \times W_{s(|x|)}(\Gamma)$ described in Lemma 5.4. Also define $\Psi_x^{(18)} = \Psi_x^{(8)}$.

6. Define

$$\Psi_x^{(10)}((a_1, a_2, a_3, b), (y_1; y_2; y_3)) = ((a_3, a_2, a_1, b), (y_3; y_2; y_1)).$$

By making straightforward modifications of the machines constructed in the proofs of Lemmas 5.3 – 5.7, reversible transformation machines $M^{(1)}, \ldots, M^{(19)}$ may be constructed that, on input x, perform transformations $\Psi_x^{(1)}, \ldots, \Psi_x^{(19)}$, respectively, on $A' \times W_{s(|x|)}(\Gamma')$, each requiring space $O(s)$. Furthermore, we may assume that each of these machines, save $M^{(4)}$, $M^{(6)}$, $M^{(8)}$, $M^{(14)}$, $M^{(16)}$, and $M^{(18)}$, performs its respective transformation in a number of steps independent of its argument.

For each $M^{(j)}$, let $G^{(j)} \times A'$ denote the state set of $M^{(j)}$, and let $g_0^{(j)}$ and $g_f^{(j)}$ denote the initial and final states of $M^{(j)}$. We may now define $G' = \{g_0', g_f'\} \cup G^{(1)} \cup \cdots \cup G^{(19)}$. Here, we view elements of $G^{(i)}$ and $G^{(j)}$ as being distinct for $i \neq j$.

It remains to define the transition function of M', which we specify by operators $\{V'_\sigma : \sigma \in \Sigma\}$ and functions D'_i and D'_w in the usual way. (Z will always take the value ε as M' is a reversible transformation machine, and hence produces no output.) First, we define $V'_\sigma|(g'_f, a'), \tau\rangle = |(g'_0, a'), \tau\rangle$ and $D'_i((g'_0, a')) = D'_w((g'_0, a')) = D'_i((g'_f, a')) = D'_w((g'_f, a')) = 0$ for every σ, a' and τ, in accordance with the restrictions we have placed on reversible transformation machines. Next, for each $j = 1, \ldots, 19$, $\sigma \in \Sigma$, $g \in G^{(j)} \backslash \{g_f^{(j)}\}$ and $a' \in A'$, define

$$\begin{aligned} V'_\sigma \left|(g, a'), \tau\right\rangle &= V_\sigma^{(j)} \left|(g, a'), \tau\right\rangle, \\ D'_i((g, a')) &= D_i^{(j)}((g, a')), \\ D'_w((g, a')) &= D_w^{(j)}((g, a')), \end{aligned}$$

so that whenever M' is in an internal state of the form (g, a') for $g \in G^{(j)} \backslash \{g_f^{(j)}\}$, M' simply mimics the behavior of $M^{(j)}$. For each of the remaining states (g, a') with $g \in \{g_f^{(j)} : 1 \leq j \leq 19\}$, define $D_i((g, a')) = D_w((g, a')) = 0$; the tape heads of M' do not move between the various transitions $\Psi_x^{(1)}, \ldots, \Psi_x^{(19)}$. Finally we must specify $V_\sigma \left|(g, a'), \tau\right\rangle$ for $g \in \{g'_0\} \cup \{g_f^{(j)} : 1 \leq j \leq 19\}$ and $\sigma \in \Sigma$, in order to determine the "flow" of M' between transformations $\Psi_x^{(1)}, \ldots, \Psi_x^{(19)}$. The action of M' will in fact depend only on g and on the b components of the internal state in these situations. Similar to the method described in Section 5.1, we define one-to-one mappings

$$\Delta_b : \{g'_0\} \cup \{g_f^{(j)} : 1 \leq j \leq 19\} \to \{g'_f\} \cup \{g_0^{(j)} : 1 \leq j \leq 19\},$$

for $b = 0, 1$, from which each V_σ is defined as $V_\sigma \left|(g, (a_1, a_2, a_3, b)), \tau\right\rangle = |(\Delta_b(g), (a_1, a_2, a_3, b)), \tau\rangle$ for each σ, τ, a_1, a_2, and a_3. (We could of course define mappings $\Delta_{a',\tau}$ as described in Section 5.1, but this is more general than what is required in the present situation.) The mappings Δ_0 and Δ_1 are to take the following values.

$$\begin{array}{llll} \Delta_0(g'_0) = g_0^{(1)}, & \Delta_0(g_f^{(5)}) = g_0^{(6)}, & \Delta_0(g_f^{(10)}) = g_0^{(11)}, & \Delta_0(g_f^{(15)}) = g_0^{(16)}, \\ \Delta_0(g_f^{(1)}) = g_0^{(9)}, & \Delta_0(g_f^{(6)}) = g_0^{(7)}, & \Delta_0(g_f^{(11)}) = g_0^{(19)}, & \Delta_0(g_f^{(16)}) = g_0^{(17)}, \\ \Delta_0(g_f^{(2)}) = g_0^{(3)}, & \Delta_0(g_f^{(7)}) = g_0^{(8)}, & \Delta_0(g_f^{(12)}) = g_0^{(13)}, & \Delta_0(g_f^{(17)}) = g_0^{(18)}, \\ \Delta_0(g_f^{(3)}) = g_0^{(4)}, & \Delta_0(g_f^{(8)}) = g_0^{(2)}, & \Delta_0(g_f^{(13)}) = g_0^{(14)}, & \Delta_0(g_f^{(18)}) = g_0^{(12)}, \\ \Delta_0(g_f^{(4)}) = g_0^{(5)}, & \Delta_0(g_f^{(9)}) = g_0^{(10)}, & \Delta_0(g_f^{(14)}) = g_0^{(15)}, & \Delta_0(g_f^{(19)}) = g'_f, \end{array}$$

$$\begin{array}{lll} \Delta_1(g_f^{(1)}) = g_0^{(2)}, & \Delta_0(g_f^{(5)}) = g_0^{(6)}, & \Delta_1(g_f^{(13)}) = g_0^{(14)}, \\ \Delta_1(g_f^{(2)}) = g_0^{(9)}, & \Delta_1(g_f^{(6)}) = g_0^{(7)}, & \Delta_1(g_f^{(14)}) = g_0^{(15)}, \\ \Delta_1(g_f^{(3)}) = g_0^{(4)}, & \Delta_1(g_f^{(11)}) = g_0^{(12)}, & \Delta_1(g_f^{(15)}) = g_0^{(16)}, \\ \Delta_1(g_f^{(4)}) = g_0^{(5)}, & \Delta_1(g_f^{(12)}) = g_0^{(19)}, & \Delta_1(g_f^{(16)}) = g_0^{(17)}. \end{array}$$

For all remaining elements of $\{g'_0\} \cup \{g_f^{(j)} : 1 \leq j \leq 19\}$, Δ_0 and Δ_1 may take arbitrary values, so long as each remains one-to-one.

Now, given that $M^{(1)}, \ldots, M^{(19)}$ are reversible transformation machines, it is routine to check that M' is a reversible transformation machine as well. Furthermore, since $\Psi_x^{(1)}, \ldots, \Psi_x^{(19)}$ are each reversible

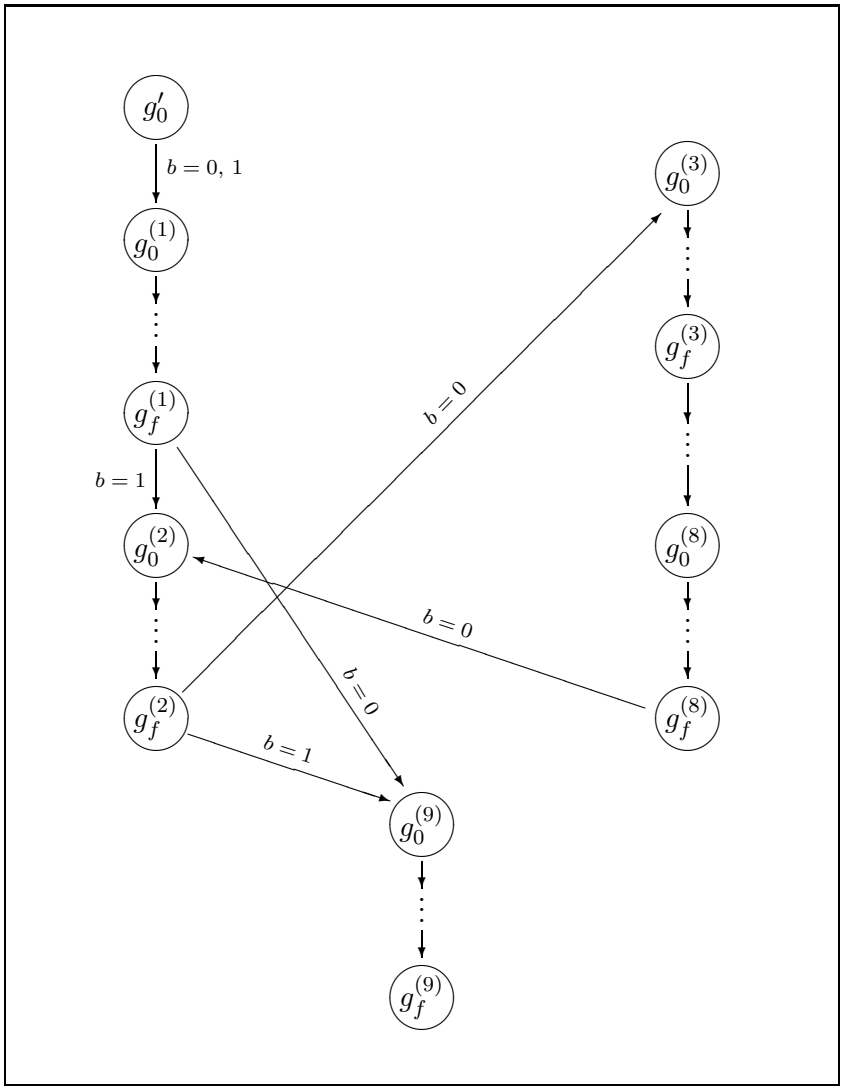

Figure 3: Diagram of state-transitions for machine M' in Lemma 5.2.

transformations on $A' \times W_{s(|x|)}(\Gamma')$ that require space at most $O(s)$, and since M' only modifies its work tape or moves its tape heads when performing one of these transformations, it follows that M' performs its transformation in space $O(s)$ as well.

Finally, we argue that M' performs the required transformation, and does so in a number of steps independent of its argument.

Transformations $\Psi_x^{(1)}, \ldots, \Psi_x^{(9)}$ perform the loop described in step 1 of Figure 2. This loop has starting/stopping condition $(a_2, y_2) = (0, \widehat{\#})$. (Note that the incrementing of (a_2, y_2) is considered to be included in the body of the loop). Figure 3 illustrates how control flows between the g components of the internal state of M' as this loop is performed. We now describe this computation. Supposing M' is initially in configuration $((g'_0, (a, 0, 0, 0)), (y; \widehat{\#}; \widehat{\#}))$, in one step the configuration becomes $((g_0^{(1)}, (a, 0, 0, 0)), (y; \widehat{\#}; \widehat{\#}))$. Transformation $\Psi_x^{(1)}$ is now performed, which flips b when-

ever the starting/stopping condition is met. As this condition is met at this point, configuration $((g_f^{(1)}, (a, 0, 0, 1)), (y; \widehat{\#}; \widehat{\#}))$ results. Since we now have $b = 1$, the state of M' evolves to a $g_0^{(2)}$ state, i.e., to $((g_0^{(2)}, (a, 0, 0, 1)), (y; \widehat{\#}; \widehat{\#}))$, causing transformation $\Psi_x^{(2)}$ (identical to $\Psi_x^{(1)}$) to then be performed. The configuration that results is $((g_f^{(2)}, (a, 0, 0, 0)), (y; \widehat{\#}; \widehat{\#}))$, as $\Psi_x^{(2)}$ causes b to again be flipped. Now, since $b = 0$, the configuration evolves to $((g_0^{(3)}, (a, 0, 0, 0)), (y; \widehat{\#}; \widehat{\#}))$. Transformations $\Psi_x^{(3)}, \ldots, \Psi_x^{(8)}$ are then applied in sequence, corresponding to steps i – v in Figure 2 and to the incrementing of (a_2, y_2) in the sense of Lemma 5.4. Thus, one iteration of the loop has been executed and control is transfered back to the $g_0^{(2)}$ state (since the value of b is invariant under transformations $\Psi_x^{(3)}, \ldots, \Psi_x^{(8)}$ applied in sequence). Transformation $\Psi_x^{(2)}$ is again applied, but this time b is not flipped because the starting/stopping condition is not met (as (a_2, y_2) has been incremented). Thus, b remains in its zero state and the loop is again executed precisely as above. This process continues until eventually $\Psi_x^{(2)}$ is performed when the starting/stopping condition is again met. When this happens (after the loop has been executed for every possible $(a_2, y_2) \in A \times W_{s(|x|)}(\Gamma)$), control is transfered to the $g_0^{(9)}$ state, which applies $\Psi_x^{(9)}$ (identical to $\Psi_x^{(2)}$) to return b to the zero value. Thus, the loop has been executed as required. During the loop, transformations $\Psi_x^{(3)}$, $\Psi_x^{(5)}$, and $\Psi_x^{(7)}$ serve to copy $\Phi_x(a, y)$ to (a_3, y_3) when it is the case that the initial value of (a_2, y_2) on that iteration of the loop matches $(a_1, y_1) = (a, y)$. It follows that the configuration of M' after executing the loop is $((g_f^{(9)}, (a, 0, a', 0)), (y; \widehat{\#}; y'))$ for $(a', y') = \Phi_x(a, y)$.

Each iteration of the loop in step 1 requires time depending only on the value of (a_2, y_2) at the start of that iteration, following from the properties of the $M^{(j)}$ that were assumed based on Lemmas 5.3 – 5.7. As the loop is executed once for each possible value of (a_2, y_2), the total time required for the loop is thus independent of the initial value of (a_1, y_1).

Now, after $\Psi_x^{(9)}$ is performed, control is transfered to the $g_0^{(10)}$ state. This induces transformation $\Psi_x^{(10)}$, causing (a_1, y_1) and (a_3, y_3) to be swapped (Step 2 in Figure 2), yielding configuration $((g_f^{(10)}, (a', 0, a, 0)), (y'; \widehat{\#}; y))$. The number of steps required for this transformation is independent of (a, y) and (a', y') by our assumptions on $M^{(10)}$.

Finally, control is transfered to the $g_0^{(11)}$ state. Transformations $\Psi_x^{(11)}, \ldots, \Psi_x^{(19)}$ induce Step 3 in Figure 2, which is a loop similar to the loop described above. Here, however, Φ_x^{-1} is applied to each argument (a_2, y_2) rather than Φ_x, and transformation $\Psi_x^{(15)}$ inverts the transformation performed by $\Psi_x^{(5)}$ (i.e., inverts the copying transformation). This has the effect of erasing (a, y) from the third track and from the a_3 component in the internal state of M', yielding configuration $((g_f^{(19)}, (a', 0, 0, 0)), (y'; \widehat{\#}; \widehat{\#}))$ when the loop is completed. This configuration evolves to $((g'_f, (a', 0, 0, 0)), (y'; \widehat{\#}; \widehat{\#}))$ in one step, and the transformation is complete. Similar to the first loop, the loop performed by transformations $\Psi_x^{(11)}, \ldots, \Psi_x^{(19)}$ requires a number of steps independent of (a, y).

It follows that M' performs the transformation (5) in a number of steps independent of its argument (for arguments of the form $((a, 0, 0, 0), (y; \widehat{\#}; \widehat{\#}))$). We may now modify M' in order to define M satisfying the statement of the lemma. The state set of M is to be $G \times A$ for $G = G' \times A \times A \times \{0, 1\}$. We identify each state $(g, (a_1, a_2, a_3, b))$ of M' with the state $((g, a_2, a_3, b), a_1)$ of M, and take the initial and final states of M to be $g_0 = (g'_0, 0, 0, 0)$ and $g_f = (g'_f, 0, 0, 0)$, respectively. The work tape alphabet and transition function of M are the same as for M', modulo the above identifications. Finally, we may view Γ as being a subset of the tape alphabet of M by identifying Γ with those tape symbols having blanks on the second and third tracks. It follows that M performs the transformation Φ_x in a number of steps independent of its argument as required. This completes the proof of Lemma 5.2. ■

5.3 Proofs of Simulation Lemmas

Now we are prepared to prove the simulation lemmas from Section 4—each lemma is restated and proved below.

Lemma 4.1 *Let M be a QTM running in space s and let $t : \mathbb{Z}^+ \to \mathbb{N}$ be computable in space $O(s)$. Let $p_{acc}(x)$ and $p_{rej}(x)$ denote the cumulative probabilities that M accepts and rejects, respectively, input x after t steps have passed. Then for any choice of $\alpha \in \{0, 1\}$ and $\beta \in \{0, \frac{1}{2}\}$ there exists a QTM M' running in space $O(s)$ and $t' : \mathbb{Z}^+ \to \mathbb{N}$ computable in space $O(s)$ such that the following hold for each input x.*

1. *The probability that M' halts within the first $t' - 1$ steps of its computation is 0.*
2. *After precisely t' steps, M' accepts x with probability $p_{acc}(x)$ and rejects x with probability $\alpha\, p_{rej}(x)$.*
3. *After precisely $t' + 1$ steps, M' accepts x with probability β and rejects x with probability $1 - \beta$ (conditioned upon M' not halting within the first t' steps of its computation).*

Proof. Let Q, Σ, and Γ denote the state set, input tape alphabet, and work tape alphabet of M, respectively, and assume the transition function of M is specified by $\{V_\sigma : \sigma \in \Sigma\}$, D_i, D_w, and Z as usual. As t is computable in space $O(s)$ for $s(n) = \Omega(\log n)$ space-constructible, we may assume there exists space-constructible $s' = \Theta(s)$ such that $s(n) \leq s'(n)$ and $\log t(n) \leq s'(n)$ for every $n \in \mathbb{Z}^+$.

Define M' as follows. Let the internal state set of M' be $G \times A$, where $A = Q \times \Sigma \times \Gamma \times \{0, 1\}$, and G is a set allowing M' to function as described below. Any particular internal state of M' may be written as $(g, (q, \sigma, \tau, b))$, and we refer to components of arbitrary internal states of M' by g, q, σ, τ, and b as necessary. For the initial state of M', we have $q = q_0$, $\sigma = \tau = \#$, and $b = 0$. Let the input tape alphabet of M' be identical to that of M, i.e., Σ, and let the work tape alphabet of M' include $\Gamma' = \Gamma \times \{0, 1, \#\}^4$, as well as any auxiliary symbols needed to perform the transformations described below. We view the work tape of M' as consisting of five tracks to be used as follows.

Track 1:	Represents the contents of the work tape of M.
Track 2:	Records the position of the input tape head of M.
Track 3:	Records the position of the work tape head of M.
Track 4:	Records the number of steps of M that have been simulated.
Track 5:	Records the time at which M halts (or 0 if M has not halted).

The integers recorded on tracks 2, 3, 4 and 5 are to be encoded as described in Section 5.1.

The manner in which M' functions is described in Figure 4.

Steps i – iii and v – viii are reversible transformations, assumed to be defined on $A \times W_{s'(|x|)}(\Gamma')$. Whenever one of these transformations refers to an integer encoded on track 2, 3, 4, or 5, we assume that this transformation is defined to be the identity when squares $-s', \ldots, s'$ do not correspond to the encoding of an integer for the track or tracks in question. For each of these transformations, it is straightforward to see that the transformation is in fact invertible, maps $A \times W_{s'(|x|)}(\Gamma')$ onto itself, and is performable by a deterministic Turing machine in space $O(s')$. By Lemmas 5.1 and 5.2, we may therefore assume each of these transformations can be performed reversibly in space $O(s)$, requiring a number of steps that depends only on x and not on the particular configuration of M' at the time the transformation is performed.

Steps iv, 2, and 3 are quantum transformations. For steps iv and 2 (and 3 in case $\beta = 0$), it is straightforward to define transformations that produce the described effects. To implement the coin-flip

1. Loop with starting/stopping condition "track 4 contains 0":

   i. If track 4 encodes a number in the range $\{0, \ldots, t-1\}$, then increment this number modulo t.

   ii. If track 5 encodes 0 (M has not yet halted), set $b = 1 - b$.

   iii. Letting y denote the contents of track 1, h_i the number encoded on track 2, and h_w the number encoded on track 3, do the following in case $b = 1$. If $h_i \in \{1, \ldots, |x|\}$ then add $x(h_i)$ to σ, and if $h_w \in \{-s, \ldots, s\}$ then swap $y(h_w)$ with τ.

   iv. If $b = 1$, perform transformation V_σ on the pair (q, τ).

   v. Invert step iii.

   vi. Again letting h_i and h_w denote the numbers encoded on tracks 2 and 3, respectively, perform the following transformations in case $b = 1$.

$$h_i \mapsto \begin{cases} (h_i + D_i(q)) \mod (|x| + 2) & \text{if } h_i \in \{0, \ldots, |x| + 1\} \\ h_i & \text{otherwise} \end{cases}$$

$$h_w \mapsto \begin{cases} [(h_w + D_w(q) + s) \mod (2s + 1)] - s & \text{if } h_w \in \{-s, \ldots, s\} \\ h_w & \text{otherwise} \end{cases}$$

   vii. Invert step ii.

   viii. Letting q denote the state of M currently represented by M', check if $Z(q) \neq \varepsilon$ (i.e., M produces output in this state); if this is the case, do the following. If track 5 encodes 0, copy the contents of track 4 onto track 5. If tracks 4 and 5 encode the same number, set track 5 to encode 0.

2. If $Z(q) = 1$, then accept. If $\alpha = 1$ and $Z(q) = 0$, then reject.

3. In case $\beta = 0$, reject. Otherwise, simulate a fair coin-flip and accept or reject accordingly.

Figure 4: Description of quantum Turing machine M' for Lemma 4.1.

in step 3, we may apply the quantum transformation H_4, defined on $\{0, 1, 2, 3\}$ as

$$H_4 : |a\rangle \rightarrow \frac{1}{2} \sum_{a'=0}^{3} (-1)^{(a,a')} \left|a'\right\rangle, \tag{6}$$

to a suitable collection of internal states. Here, (a, a') denotes the number of 1's in the bitwise-and of a and a' written in binary.

The above reversible and quantum transformations can be composed in the manner described in Figure 4 as discussed in Section 5.1. It follows that M' operates in space $O(s)$, and furthermore each step in Figure 4 requires a number of steps that is invariant over all computation paths of M' for fixed input x.

Now we argue that M' behaves as claimed. The main loop (step 1) is iterated $t(|x|)$ times; during each iteration, one step in the computation of M is simulated. Step i of the loop simply increments

Repeat ad infinitum:

1. Same as step 1 in the proof of Lemma 4.1 (see Figure 4).
2. If $Z(q) = 1$, then accept. If $Z(q) = 0$, then reject.
3. Perform the inverse of step 1.
4. If the current configuration of M represented is not the initial configuration, then multiply the current amplitude by -1.

Figure 5: Description of quantum Turing machine M' for Lemma 4.3.

the number on track 4, which clocks the simulation. Steps iii – vi of the loop correspond to one step in the evolution of M, and are to be executed only along computation paths of M' corresponding to computation paths of M along which M has not yet halted. In order to insure this, in step ii the value of b is set to 1 in case M has not halted (track 5 contains 0), and b is returned to its initial value of 0 in step vii: if $b = 0$, steps iii – vi do not modify the configurations of M represented by M', and if $b = 1$, steps iii – vi cause M' to mimic one step in the evolution of M. Finally, in step viii the halting time is recorded on track 5 along those computation paths for which M has just entered a halting configuration. In effect, this simulates the observation that occurs after each step of M's computation, as M' will not cause halting configurations of M to evolve on subsequent iterations of the loop.

We note that under the assumption that M is a valid machine running in space s, step vi simply corresponds to adding $D_i(q)$ and $D_w(q)$ to h_i and h_w, respectively. The reason the transformation given in Figure 4 is used is that this transformation has domain and range contained in $A \times W_{s'(|x|)}(\Gamma')$, allowing us to apply Lemmas 5.1 and 5.2.

In steps 2 and 3, M' produces output depending on the state of M represented by M', α and β. Defining $t'(|x|)$ to be the number of steps required for M' to complete step 2, we see that M' accepts and rejects at times $t'(|x|)$ and $t'(|x|) + 1$ as claimed. As t' is clearly computable in space $O(s)$, this completes the proof. ■

Lemma 4.3 *Let M be a QTM running in space s and let $t : \mathbb{Z}^+ \to \mathbb{N}$ be computable in space $O(s)$. Let $p_{acc}(x)$ and $p_{rej}(x)$ denote the cumulative probabilities that M accepts and rejects, respectively, input x after t steps have passed. Then there exists a QTM M' running in space $O(s)$ such that for each input x, if $p_{acc}(x) + p_{rej}(x) > 0$ then M' accepts with probability $\frac{p_{acc}(x)}{p_{acc}(x)+p_{rej}(x)}$ and rejects with probability $\frac{p_{rej}(x)}{p_{acc}(x)+p_{rej}(x)}$, and if $p_{acc}(x) + p_{rej}(x) = 0$ then M' accepts and rejects with probability 0.*

Proof. The manner in which M' simulates M is similar to the machine constructed in the proof of Lemma 4.1. In the present case the simulation will be repeated ad infinitum, in the manner described below, so as to to amplify the probabilities of acceptance and rejection accordingly.

The work tape of M' will consist of five tracks, used precisely as in the proof of Lemma 4.1. Similarly the internal states of M' will be of the same form as in that proof. The execution of M' is described in Figure 5. We note that step 3 can be performed by composing the inverses of the various reversible and quantum transformations that comprise step 1 in an appropriate manner. Step 4 may be effected

by a "controlled phase flip" as follows: (i) flip some initially zero bit b in the internal state of M' when the configuration currently represented by M' is not the initial configuration, (ii) perform the quantum transformation $|b\rangle \mapsto (-1)^b |b\rangle$ on b, and (iii) again apply the transformation described in (i). Finally, the process of repeating steps 1 – 4 ad infinitum may be accomplished as in the example in Section 5.1. Similar to the machine constructed in the proof of Lemma 4.1, M' operates in space $O(s)$ as each of the abovementioned transitions is performed in this space bound.

Now we show that the claimed probabilities of acceptance and rejection are obtained. In accordance with (1), we may calculate the unconditional probabilities with which M' accepts and rejects at various times by not renormalizing superpositions after each observation.

Let c'_0 denote the initial configuration of M', and let F be the operator that corresponds to performing step 1, i.e., simulating M for t steps. Let $|\psi\rangle = F|c'_0\rangle$, write $|\psi\rangle = |\psi_1\rangle + |\psi_0\rangle + |\psi_\varepsilon\rangle$, where $|\psi_1\rangle$, $|\psi_0\rangle$ and $|\psi_\varepsilon\rangle$ represent the projections of $|\psi\rangle$ onto those subspaces spanned by classical states for which $Z(q)$ is 1, 0, or ε for q denoting the internal state of M represented by M', and note that we have $\| |\psi_1\rangle \|^2 = p_{acc}$ and $\| |\psi_0\rangle \|^2 = p_{rej}$. During step 2, M' accepts, rejects, or does not halt accordingly, and hence accepts with probability p_{acc} and rejects with probability p_{rej}. Otherwise, the superposition collapses to $|\psi_\varepsilon\rangle$ and the computation continues. Next, the inverse of step 1 is performed, which maps $|\psi_\varepsilon\rangle$ to a state of the form $F^{-1}|\psi_\varepsilon\rangle = |c'_0\rangle - F^{-1}|\psi_1\rangle - F^{-1}|\psi_0\rangle$ (except that some component of the internal state of M' is different, reflecting the fact that we are at step 4 rather than step 1). Writing $|\xi_1\rangle = F^{-1}|\psi_1\rangle - p_{acc}|c'_0\rangle$ and $|\xi_0\rangle = F^{-1}|\psi_0\rangle - p_{rej}|c'_0\rangle$, we have

$$F^{-1}|\psi_\varepsilon\rangle \;=\; (1 - p_{acc} - p_{rej})\,|c'_0\rangle - |\xi_1\rangle - |\xi_0\rangle\,.$$

Furthermore

$$\langle c'_0 \,|\, \xi_1\rangle \;=\; \langle c'_0 \,|\, F^{-1} \,|\, \psi_1\rangle - p_{acc} \;=\; \langle \psi_1 \,|\, F \,|\, c'_0\rangle - p_{acc} \;=\; 0,$$

and similarly $\langle c'_0 \,|\, \xi_0\rangle = 0$. After applying step 4 and returning to step 1, the state of the machine is thus $(1 - p_{acc} - p_{rej})\,|c'_0\rangle + |\xi_1\rangle + |\xi_0\rangle$, and after again performing the simulation in step 1, the superposition of M' is

$$\begin{aligned}(1 - p_{acc} - p_{rej})F\,|c'_0\rangle + F\,|\xi_1\rangle + F\,|\xi_0\rangle \\ \quad = \quad (2 - 2p_{acc} - 2p_{rej})\,|\psi_1\rangle + (2 - 2p_{acc} - 2p_{rej})\,|\psi_0\rangle + (1 - 2p_{acc} - 2p_{rej})\,|\psi_\varepsilon\rangle\,.\end{aligned}$$

From this, we may determine that the probability that M' accepts after the $(k+2)$-st execution of step 2, for $k \geq 0$, is $\big((1 - 2p_{acc} - 2p_{rej})^k(2 - 2p_{acc} - 2p_{rej})\big)^2\, p_{acc}$. (Note that since we have not re-normalized superpositions, the above expression represents an unconditional probability.) Now the total probability that M' accepts may be calculated as

$$\sum_{k=0}^{\infty}((1 - 2p_{acc} - 2p_{rej})^k(2 - 2p_{acc} - 2p_{rej}))^2 p_{acc} + p_{acc} = \begin{cases} \frac{p_{acc}}{p_{acc}+p_{rej}} & \text{if } p_{acc} > 0 \\ 0 & \text{if } p_{acc} = 0, \end{cases}$$

and the probability that M' rejects may be determined similarly. ∎

Lemma 4.6 *Let M be a QTM running in space s, and let $p_{acc}(x)$ denote the probability that M accepts input x. Then for any polynomial h there exists a QTM M' running in space $O(s)$ such that the following hold.*

1. M' accepts each input x with probability $p'_{acc}(x)$ satisfying $p_{acc}(x) - 2^{-h(2^s)} \leq p'_{acc}(x) \leq p_{acc}(x)$.

2. M' halts almost surely for every input x.

Proof. Let Q, Σ and Γ denote the state set, input alphabet and work tape alphabet of M, and assume that the transition function of M is specified by $\{V_\sigma : \sigma \in \Sigma\}$, D_i, D_w, and Z as usual.

The internal state set of M' will be of the form $G \times A$, where

$$A \;=\; \{0,1\} \times Q \times \Sigma \times \Gamma \times \{0,1,2,3\},$$

and G is a set allowing M' to function as described below. Internal states of M' may be written in the form $(g,(b,q,\sigma,\tau,a))$, and we refer to particular components of arbitrary internal states by g, b, q, σ, τ, and a as necessary. For the initial state of M', we have $b = 0$, $q = q_0$, $\sigma = \tau = \#$, and $a = 0$.

The work tape of M' will consist of five tracks, which will be used as follows.

Track 1:	Represents the contents of the work tape of M.
Track 2:	Records the position of the input tape head of M.
Track 3:	Records the position of the work tape head of M.
Tracks 4/5:	Store integers as described below.

The work tape alphabet of M', which we denote Γ', may be defined appropriately. We assume that the integers recorded on tracks 2, 3, 4 and 5 are encoded as discussed in Section 5.1, so that these tracks each initially encode 0.

The manner in which M' functions is described in Figure 6. We let $t = h(2^s) + h_0(2^s)$, where h_0 is a polynomial depending on M defined below. Consequently we have $t = 2^{O(s)}$, and space-constructibility of s implies that $t(|x|)$ can be computed in space $O(s)$. Recall the definition of H_4 from (6).

Each step in Figure 6 corresponds to a reversible transformation on $A \times W_{s'(|x|)}(\Gamma')$ for appropriately defined s', a quantum transformation, or a composition of such transformations. It follows that M' is a valid quantum Turing machine operating in space $O(s)$, and furthermore that each step in Figure 6 requires a number of steps that is invariant over all computation paths of M' for fixed input x.

Now we calculate the probabilities with which M' accepts and rejects. For each input x, let $E(x)$ be an $N \times N$ rational matrix as in Lemma 3.6 for M on input x, and define polynomials f and g for $E(x)$ as in the proof of Lemma 3.10. Recall that $E(x)^{k+2}[N,1]$ is the probability that M accepts x after precisely k steps,

$$\frac{f(z)}{g(z)} \;=\; \sum_{k \geq 0} z^k E(x)^k [N,1]$$

for every $z \in [0,1]$, and we have $\deg(f)$, $\deg(g) \leq N$, $\|f\|, \|g\| \leq N!\, m^N 2^{2N-1}$. As noted in the proof of Lemma 3.10, $\frac{f(z)}{g(z)}$ is nondecreasing and nonnegative on the interval $[0,1]$.

At this point we may define h_0 to be any polynomial satisfying

$$h_0(2^s) > \frac{1}{2} \log \left(2\,(N+2)(N+3)^2 (N!\, m^N 2^{2N})^2 \right)$$

(e.g., $h_0(n) = n^4 + (8 \log m)\, n^2 + 18$).

Now, let us consider the effects of steps 1 – 4 on tracks 4 and 5 and on the a component of the internal state of M'; for convenience, let us refer to the numbers encoded on tracks 4 and 5 as i and j,

Repeat ad infinitum:

1. Loop with starting/stopping condition "track 4 contains 0":
   i. If track 4 encodes a number in the range $\{0, \ldots, t-1\}$, then increment this number modulo t.
   ii. Perform H_4 on a.
   iii. If track 5 encodes a number in the range $\{0, \ldots, 3t\}$, then add a to this number modulo $3t(|x|)+1$.
2. If track 5 encodes 0 and $a = 0$, multiply the current amplitude by -1.
3. Perform the inverse of step 1.
4. Reject if track 5 does not contain 0, or if $a \neq 0$.
5. Simulate one step of the computation of M:
   i. Letting y, h_i, and h_w denote the contents of track 1, the number encoded on track 2, and the number encoded on track 3, respectively, add $x(h_i)$ to σ if $h_i \in \{1, \ldots, |x|\}$ and swap $y(h_w)$ with τ if $h_w \in \{-s, \ldots, s\}$.
   ii. Perform transformation V_σ on the pair (q, τ) and produce output according to $Z(q)$.
   iii. Perform the inverse of step i.
   vi. Again letting h_i and h_w denote the numbers encoded on tracks 2 and 3, respectively, perform the following transformations.

$$h_i \mapsto \begin{cases} (h_i + D_i(q)) \mod (|x|+2) & \text{if } h_i \in \{0, \ldots, |x|+1\} \\ h_i & \text{otherwise} \end{cases}$$

$$h_w \mapsto \begin{cases} [(h_w + D_w(q) + s) \mod (2s+1)] - s & \text{if } h_w \in \{-s, \ldots, s\} \\ h_w & \text{otherwise} \end{cases}$$

Figure 6: Description of quantum Turing machine M' for Lemma 4.6.

respectively. Suppose we have $a = i = j = 0$ initially. If we consider the quantum state of a, i and j as an element of $\ell_2(\{0,1,2,3\} \times \mathbb{Z} \times \mathbb{Z})$, this initial state is $|0,0,0\rangle$. After k iterations of the loop in step 1, this state is transformed as follows.

$$\begin{aligned}
k = 1: &\quad \frac{1}{2} \sum_{a_1} |a_1, 1, a_1\rangle \,, \\
k = 2: &\quad \frac{1}{4} \sum_{a_1, a_2} (-1)^{(a_1, a_2)} |a_2, 2, a_1 + a_2\rangle \,, \\
k = 3: &\quad \frac{1}{8} \sum_{a_1, a_2, a_3} (-1)^{(a_1, a_2) + (a_2, a_3)} |a_3, 3, a_1 + a_2 + a_3\rangle \,, \\
&\vdots \\
k = t: &\quad 2^{-t} \sum_{a_1, \ldots, a_t} (-1)^{(a_1, a_2) + \cdots + (a_{t-1}, a_t)} |a_t, 0, a_1 + \cdots + a_t\rangle \,,
\end{aligned}$$

with each a_j summed over $\{0,1,2,3\}$. Letting F denote the unitary operator on $\ell_2(\{0,1,2,3\} \times \mathbb{Z} \times \mathbb{Z})$ corresponding to performing step 1 on a, i and j, we have $\langle 0,0,0 \,|\, F \,|\, 0,0,0 \rangle = 2^{-t}$, and hence we may write $F\,|0,0,0\rangle = 2^{-t}\,|0,0,0\rangle + |\xi\rangle$ for $|\xi\rangle$ satisfying $\langle 0,0,0 \,|\, \xi \rangle = 0$. Now step 2 is performed, which maps this state to

$$-2^{-t}\,|0,0,0\rangle + |\xi\rangle \;=\; F\,|0,0,0\rangle - 2^{-t+1}\,|0,0,0\rangle\,.$$

Step 3 inverts step 1, which has the effect of applying F^{-1} to $F\,|0,0,0\rangle - 2^{-t+1}\,|0,0,0\rangle$, yielding

$$\left(1 - 2^{-2t+1}\right)|0,0,0\rangle - 2^{-t+1}\left|\xi'\right\rangle,$$

for $\langle 0,0,0 \,|\, \xi' \rangle = 0$. Now step 4 is performed, which causes M' to reject whenever one of a, i or j is nonzero, and hence with probability $1 - \left(1 - 2^{-2t+1}\right)^2 = 2^{-2t+2}\left(1 - 2^{-2t}\right)$. Otherwise the state of a, i and j collapses to $|0,0,0\rangle$. Thus, for $\epsilon = 2^{-2t+2}\left(1 - 2^{-2t}\right)$, steps 1 – 4 cause M' to reject with probability ϵ, and otherwise leave tracks 4 and 5 and the a component of the internal state of M' in their initially zero states.

As step 5 simply corresponds to simulating successive steps in the computation of M, M' accepts on the kth iteration of steps 1 – 5 with probability $E(x)^{k+2}[N,1]$, conditioned on the fact that M' has not rejected during any iteration of steps 1 – 4 thus far. Hence, the unconditional probability that M' accepts on the kth iteration of the main loop is $(1-\epsilon)^k E(x)^{k+2}[N,1]$. Consequently, the total probability that M' accepts is

$$p'_{acc}(x) \;=\; \sum_{k \geq 1} (1-\epsilon)^k E(x)^{k+2}[N,1] \;=\; \frac{f(1-\epsilon)}{(1-\epsilon)^2 g(1-\epsilon)}.$$

We may now apply Lemma 3.9 to obtain

$$\begin{aligned}
|p'_{acc}(x) - p_{acc}(x)| &= \left| \frac{f(1)}{g(1)} - \frac{f(1-\epsilon)}{(1-\epsilon)^2 g(1-\epsilon)} \right| \\
&\leq 2\epsilon(N+2)(N+3)^2 \left(N!\, m^N 2^{2N-1}\right)^2 \\
&\leq 2^{-2h(2^s)} \left(2^{-2h_0(2^s)+2} \cdot 2\,(N+2)(N+3)^2 (N!\, m^N 2^{2N-1})^2\right) \\
&< 2^{-h(2^s)}
\end{aligned}$$

which proves item 1.

Item 2 follows from the fact that M' halts with some fixed, positive probability ϵ on each iteration of the main loop. ■

Lemma 4.9 *Let M be a well-behaved PTM running in space s, and let $p_{acc}(x)$ and $p_{rej}(x)$ denote the probabilities that M accepts x and rejects x, respectively. Then there exist a QTM M' running in space $O(s)$ and $t' : \mathbb{Z}^+ \to \mathbb{N}$ computable in space $O(s)$ such that for each input x, M' accepts x with probability $\left(2^{-2st}\, p_{acc}(x)\right)^2$ and rejects x with probability $\left(2^{-2st}\, p_{rej}(x)\right)^2$ after t' steps.*

Proof. The state set of M' will be of the form $G \times A$, where $A = \{0,1,2,3\} \times \{0,1\}$, and G is a set allowing M' to function as described below. States of M' may be written as $(g,(a,b))$, so we refer to

1. Compute the length s binary encoding of c_0 and write this encoding on track 1. Also mark off s zeroes on track 2.
2. Loop with starting/stopping condition "track 4 contains 0":
   i. If track 4 encodes a number in the range $\{0, \ldots, t-1\}$, then increment this number modulo t.
   ii. Perform H_4 on each digit on track 2.
   iii. If track 1 encodes a configuration with 2 successors, perform H_4 on a.
   iv. If any of the symbols on track 2 are in the set $\{2, 3\}$, or if $a \neq 0$, or if the contents of track 2 do not encode a configuration c' that is a successor of c, then increment the number on track 3 modulo $2t+1$ in case track 3 encodes a number in the range $\{0, \ldots, 2t\}$.
   v. Swap the contents of tracks 1 and 2.
   vi. Perform H_4 on each digit on track 2.
   vii. If track 2 contains any nonzero digit and track 3 encodes a number in the range $\{0, \ldots, 2t\}$, then increment the number on track 3 modulo $2t+1$.
3. If track 3 contains 0 and track 1 encodes an accepting configuration, then accept. If track 3 contains 0 and track 1 encodes a rejecting configuration, then reject.

Figure 7: Description of quantum Turing machine M' for Lemma 4.9

particular components in such states as g, a, and b as in previous proofs in this section. For the initial state of M' we have $a = b = 0$. The input tape alphabet of M' is identical to that of M, and the work tape alphabet of M' will be a superset of $\Gamma' = \{0, 1, 2, 3, \#\}^2 \times \{0, 1, \#\}^2$. The work tape of M' is viewed as consisting of four tracks: tracks 1 and 2 will be used to encode configurations of M, track 3 will contain a counter described below, and track 4 will record the number of steps for which M has been simulated.

The behavior of M' is described in Figure 7. The transformation H_4 referred to in Figure 6 is defined in (6). Each step in Figure 7 can be seen to correspond to a reversible transformation on $A \times W_{s'(|x|)}(\Gamma')$ for appropriately defined $s' = O(s)$, a quantum transformation, or a composition of such transformations. By arguments similar to those in previous proofs in this section, it follows that the resulting machine M' is a valid quantum Turing machine, operates in space $O(s)$, and furthermore that each step in Figure 7 requires a number of steps that is invariant over all computation paths of M' for fixed input x.

We now determine the probabilities with which M' accepts and rejects. The counter on track 3 acts as a flag; whenever this number is nonzero, the simulation has failed (a counter is used so that this can be done reversibly). Note that this counter is incremented at most $2t$ times, so incrementing the counter modulo $2t+1$ is equivalent to simply incrementing it. We will say that any configuration of M' is *good* whenever track 3 encodes the number 0.

Suppose that M' is in a good configuration in which track 1 encodes $c \in \mathcal{C}(M)$, track 2 contains all zeroes, and $a = 0$, and let a single iteration of the loop in step 2 be executed. If c has exactly one

successor c', then we see that the amplitude with which M' evolves into another good configuration with c replaced by c' (and the number on track 4 incremented) is 2^{-2s} (for each of $2s$ digits, there is exactly one new digit that must result from application of H_4, for which the corresponding amplitude will necessarily be $1/2$). Similarly, if c has two successors c' and c'', then the amplitudes in this case are each $\frac{1}{2}2^{-2s}$ (since now a must be transformed to 0 as well). All other good configurations are yielded with amplitude 0.

In this way, the amplitudes of the transitions between good configurations mimic the probabilities of the corresponding transitions of M, except that a factor of 2^{-2s} is introduced during each iteration of the loop in step 2. Given that M is well-behaved, we have that the amplitudes associated with the good configurations of M' encoding the single accepting and single rejecting configuration of M after t iterations of the loop will therefore be $(2^{-2st})p_{acc}(x)$ and $(2^{-2st})p_{rej}(x)$, respectively. Defining t' to be the number of steps required for M' to complete step 3, we have that M' accepts and rejects with probability $\left((2^{-2st})p_{acc}(x)\right)^2$ and $\left((2^{-2st})p_{rej}(x)\right)^2$, respectively, after t' steps. As t' is clearly computable in space $O(s)$, this completes the proof. ■

Lemma 4.13 *Let M be a well-behaved PTM running in space s and let $p_{acc}(x)$ and $p_{rej}(x)$ denote the probabilities that M accepts x and rejects x, respectively. Then there exists a QTM M' running in space $O(s)$ such that, for each input x, M' accepts x with probability 0 if and only if $p_{acc}(x) = p_{rej}(x)$.*

Proof. We define a quantum Turing machine M' in a similar manner to the machine constructed in the proof of Lemma 4.9, but modified as described in Figure 8.

1 – 2. Same as in the proof of Lemma 4.9 (see Figure 7).

3. If track 1 encodes an accepting configuration, then add 1 to a (otherwise leave a unchanged).

4. Perform H_4 on every digit on track 1. If track 1 does not now contain all zeroes, add 1 to the number encoded on track 3 (modulo $2t + 2$, etc.).

5. Perform H_4 on a. If track 3 encodes 0, and if $a = 1$, then accept, otherwise reject.

Figure 8: Description of quantum Turing machine M' for Lemma 4.13.

Recall the definition of a good configuration from the proof of Lemma 4.9. Since M must be in the unique accepting or unique rejecting configuration after t steps, there are only 2 good configurations that M' can be in after performing step 4: one in which $a = 1$ and the other in which $a = 0$ (all other aspects of these configurations being equal). The amplitudes associated with these two configurations are $2^{-s(2t+1)}p_{acc}$ and $2^{-s(2t+1)}p_{rej}$ (for $a = 1$ and $a = 0$, respectively). Since $\langle 1 \,|\, H_4 \,|\, 0 \rangle = -\langle 1 \,|\, H_4 \,|\, 1 \rangle$, we see that after performing H_4 on a we will have a nonzero amplitude associated with $a = 1$ if and only if $p_{acc}(x) \neq p_{rej}(x)$. Hence, M' accepts with nonzero probability if and only if $p_{acc}(x) \neq p_{rej}(x)$, as required. ■

6 Conclusion and open problems

Figure 9 is a diagram that summarizes the relationships among some of the quantum and classical space-bounded classes we have discussed in this paper.

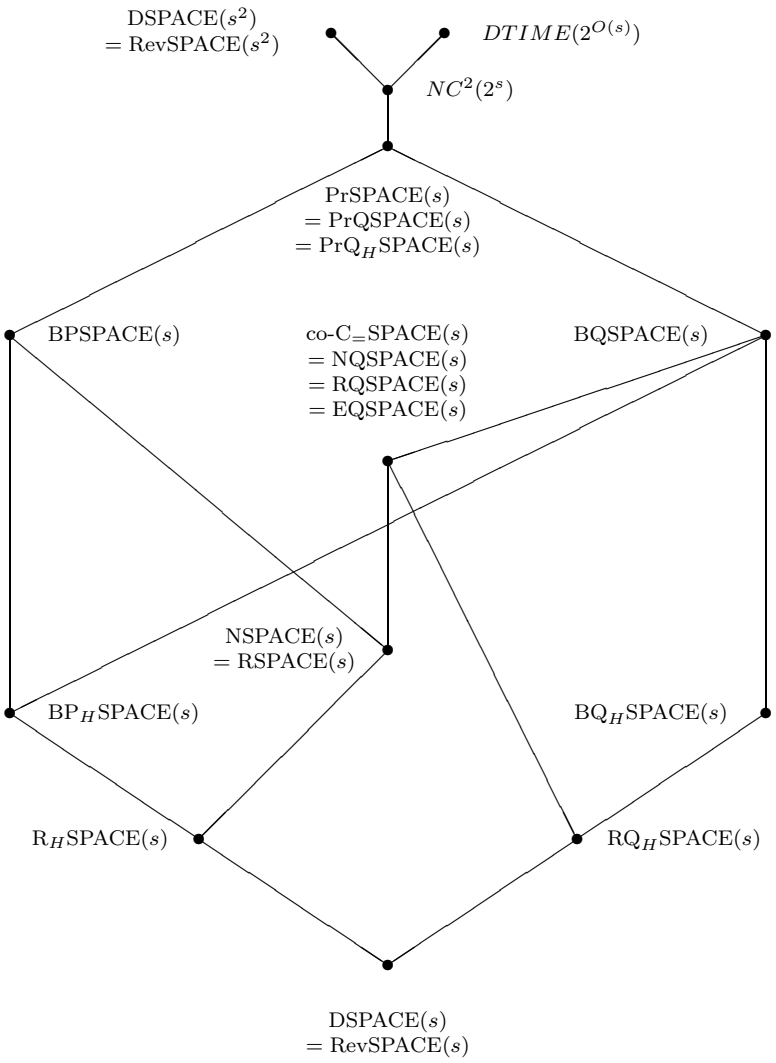

Figure 9: Relationships among quantum and classical space-bounded classes.

A number of interesting questions have been left open by this paper. In particular, can bounded-error or one-sided error probabilistic machines that halt absolutely be simulated by quantum machines that also halt absolutely and have bounded or one-sided error, e.g., do either of the relationships $\mathrm{BP}_H\mathrm{SPACE}(s) \subseteq \mathrm{BQ}_H\mathrm{SPACE}(s)$ or $\mathrm{R}_H\mathrm{SPACE}(s) \subseteq \mathrm{RQ}_H\mathrm{SPACE}(s)$ hold? Similarly, can probabilistic simulations of space-bounded quantum machines be performed with bounded error, e.g., are either of the quantum classes $\mathrm{RQ}_H\mathrm{SPACE}(s)$ or $\mathrm{BQ}_H\mathrm{SPACE}(s)$ contained in, say, $\mathrm{BPSPACE}(s)$?

We have not mentioned a number of other classical space-bounded classes (e.g., symmetric space, probabilistic classes allowing multiple access to random bits). Are there natural quantum analogues of

these classes, and how do they relate to the classes discussed in this paper? Similarly, oracle quantum Turing machines have not been mentioned, and we are not aware of any work in this direction in the space-bounded case. What can be said regarding relativized results for space-bounded quantum classes?

Finally, we have restricted our attention to space bounds that are at least logarithmic in the input size. In the case of constant space bounds, polynomial time QTMs are strictly more powerful than polynomial time PTMs [17]. What else can be said about sub-logarithmic space bounds?

References

[1] E. Allender and M. Ogihara. Relationships among PL, #L, and the determinant. *RAIRO - Theoretical Informatics and Applications*, 30:1–21, 1996. A preliminary version appeared in *Proceedings of the 9th Annual Structure in Complexity Theory Conference*, pages 267–278, 1994.

[2] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17:525–532, 1973.

[3] C. H. Bennett. Time/space trade-offs for reversible computation. *SIAM Journal of Computing*, 18(4):766–776, 1989.

[4] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

[5] A. Berthiaume and G. Brassard. Oracle quantum computing. *Journal of Modern Optics*, 41(12):2521–2535, 1994.

[6] A. Borodin. On relating time and space to size and depth. *SIAM Journal on Computing*, 6:733–744, 1977.

[7] A. Borodin, S. Cook, and N. Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. *Information and Control*, 58:113–136, 1983.

[8] P. Crescenzi and C. Papadimitriou. Reversible simulation of space-bounded computations. *Theoretical Computer Science*, 143:159–165, 1995.

[9] D. Deutsch. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London*, A400:97–117, 1985.

[10] D. Deutsch and R. Jozsa. Rapid solutions of problems by quantum computation. *Proceedings of the Royal Society of London*, A439:553–558, 1992.

[11] S. Fenner, F. Green, S. Homer, and R. Pruim. Determining acceptance possibility for a quantum computation is hard for PH. Technical Report 98-008, Computer Science Department, Boston University, April 1998.

[12] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.

[13] L. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(2):325–328, 1997.

[14] A. Householder. *The Theory of Matrices in Numerical Analysis*. Blaisdell Publishing Company, second edition, 1965.

[15] H. Jung. Relationships between probabilistic and deterministic tape complexity. In *10th Symposium on Mathematical Foundations of Computer Science*, volume 118 of *Lecture Notes in Computer Science*, pages 339–346, 1981.

[16] H. Jung. On probabilistic time and space. In *Proceedings of the 12th International Colloquium on Automata, Languages and Programming*, volume 194 of *Lecture Notes in Computer Science*, pages 310–317. Springer-Verlag, 1985.

[17] A. Kondacs and J. Watrous. On the power of quantum finite state automata. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 66–75, 1997.

[18] K. Lange, P. McKenzie, and A. Tapp. Reversible space equals deterministic space (extended abstract). In *Proceedings of the 12th Annual IEEE Conference on Computational Complexity*, pages 45–50, 1997.

[19] W. Ruzzo, J. Simon, and M. Tompa. Space-bounded hierarchies and probabilistic computations. *Journal of Computer and System Sciences*, 28:216–230, 1984.

[20] M. Saks. Randomization and derandomization in space-bounded computation. In *Proceedings of the 11th Annual IEEE Conference on Computational Complexity*, pages 128–149, 1996.

[21] M. Saks and S. Zhou. $RSPACE(s) \subseteq DSPACE(s^{3/2})$. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 344–353, 1995.

[22] J. Savage. *The Complexity of Computing*. Wiley, 1976.

[23] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[24] D. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997.

[25] J. Watrous. Relationships between quantum and classical space-bounded complexity classes. In *Proceedings of the 13th Annual IEEE Conference on Computational Complexity*, pages 210–227, 1998.

[26] J. Watrous. *Space-Bounded Quantum Computation*. PhD thesis, University of Wisconsin – Madison, 1998.

[27] A. Yao. Quantum circuit complexity. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, pages 352–361, 1993.