Relationships Between Quantum and Classical Space-Bounded Complexity Classes

John Watrous[*]
Computer Sciences Department
University of Wisconsin
Madison, Wisconsin 53706
watrous@cs.wisc.edu

Abstract

This paper investigates the relative power of space-bounded quantum and classical (probabilistic) computational models. The following relationships are proved.

1. *Any probabilistic Turing machine (PTM) which runs in space s and which halts absolutely (i.e. halts with certainty after a finite number of steps) can be simulated in space $O(s)$ by a quantum Turing machine (QTM). 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 unbounded error case, the QTM may be taken to halt absolutely.*

2. *Any QTM running in space s can be simulated by an unbounded error PTM running in space $O(s)$. No assumptions on the probability of error or running time for the QTM are required, but it is assumed that all transition amplitudes of the quantum machine are rational.*

It follows that unbounded error, space $O(s)$ bounded quantum Turing machines and probabilistic Turing machines are equivalent in power. This implies that any space s QTM can be simulated deterministically in space $O(s^2)$, and further that any (unbounded-error) QTM running in log-space can be simulated in NC^2.

We also consider quantum analogues of nondeterministic and one-sided error probabilistic space-bounded classes, and prove some simple relationships 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 terms of time-bounded computation, than classical (probabilistic) computers (see [6, 7, 12, 16, 25, 26, 27], for instance). In this paper, we consider the relative power of quantum and classical machines when space, rather than time, is the resource of primary concern. In particular, we define quantum complexity classes which are analogous to classes traditionally studied in the context of space-bounded probabilistic computation, and prove various relationships among these quantum and classical classes.

The model for quantum computation which we use in this paper is the quantum Turing machine (QTM) model, first formally defined by Deutsch [11] (see also [6, 31]). We use a multitape version of the QTM model; in addition to having a read-only input tape, our QTMs also have an output tape which is assumed to be observed after each and every computation step. Such a model is better 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. In this paper, we restrict our attention to QTMs which have rational transition amplitudes.

QTMs can perform exactly those deterministic computations which are reversible, and consequently previous work on reversible computation is quite relevant to our discussion. It was proved by Bennett [3] that any deterministic Turing machine (DTM) computation can be simulated by a reversible Turing machine (RTM) (which we may define to be a deterministic Turing machine for which each configuration has at most one predecessor). 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 [4]. This implies the relationship DSPACE(s) $\subseteq$ RevSPACE(s^2),

[*] Supported in part by NSF grant CCR-95-10244

where RevSPACE(s) denotes, for a given space bound s, the class of languages recognizable in space $O(s)$ by a RTM. It was later proved [9] that nondeterministic Turing machines can also be simulated reversibly with the same increase in space, i.e. NSPACE(s) $\subseteq$ RevSPACE(s^2). Quite recently, Lange, McKenzie and Tapp [22] 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. DSPACE(s) = RevSPACE(s).

Given that DSPACE(s) = RevSPACE(s), we may deduce various relationships among probabilistic and quantum space-bounded classes by considering deterministic simulations of probabilistic machines. Independently, Jung [18] and Borodin, Cook and Pippenger [8] showed that any probabilistic Turing machine (PTM), even in the case of unbounded error and without restriction on running time, can be simulated deterministically with at most a quadratic increase in space, i.e. PrSPACE(s) $\subseteq$ DSPACE(s^2). (The result proved by Borodin, Cook and Pippenger [8] is somewhat stronger than this, implying that PrSPACE($\log n$) $\subseteq$ NC^2.) 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 [24] proved that any bounded error PTM which runs in space s and halts absolutely (i.e. halts with certainty after some finite number of steps) 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. In the context of time-bounded computation, it is known that QTMs can simulate PTMs without significant increase in running time (following from the fact that QTMs can simulate coin-flips, along with Bennett's simulation). It is not clear, however, that a similar technique can be applied in the space-bounded case, as simulating coin-flips requires space which cannot be reused in any obvious way.

Using a method not directly based on simulating coin-flips, we show that any bounded error PTM which 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. We also 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 a space s nondeterministic Turing machine can be simulated by a QTM, running in space $O(s)$, with respect to this notion of acceptance.

In the other direction, we consider probabilistic simulations of space-bounded quantum machines. We show that any (unbounded error) QTM running in space s can be simulated by an unbounded error PTM running in space $O(s)$, from which it follows that unbounded error space-bounded PTMs and QTMs are equivalent in power. From this we conclude that any (unbounded error) QTM running in space s can be simulated deterministically in space $O(s^2)$, and in the case $s(n) = O(\log n)$ the simulation can be performed in NC^2. Further, we have that unbounded error, space-bounded QTMs do not lose power if required to halt absolutely; a result which is analogous to one proved by Jung [20] for the probabilistic case (see also [2]). Our proofs of these relationships use a technique similar to one often used in the probabilistic case; the problem of determining if quantum machines accept with probability exceeding 1/2 is reduced to the problem of comparing determinants of integer matrices.

The remainder of this paper has the following organization. In Section 2 we define the multitape quantum Turing machine model and space-bounded quantum complexity classes which are used throughout the paper. In Section 3 we discuss some of the relationships which hold among these quantum classes, and in Sections 4 and 5 we show how quantum and probabilistic space-bounded classes compare by considering quantum simulations of probabilistic machines and probabilistic simulations of quantum machines, respectively. Finally, Section 6 contains some concluding remarks and mentions some open questions. Proofs of various claims from Sections 3 – 5 appear in an appendix following Section 6.

2 Preliminaries

2.1 Notation

We begin by mentioning some of the notation used in this paper. As usual, $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{C}$ denote the natural numbers (excluding 0), integers and complex numbers, respectively, and $\mathbb{Z}^+ = \mathbb{N} \cup \{0\}$. The empty string over any given alphabet will be denoted ε. For any finite or countable set S, $\ell_2(S)$ will denote the Hilbert space whose elements are mappings from S to $\mathbb{C}$. 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 by $|\psi\rangle$, $|\phi\rangle$, 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 rather than the second).

2.2 Multitape quantum Turing machines

We now define a multitape version of the quantum Turing machine model which is better suited to the study of space-bounded classes than the more usual single-tape model. Specifically, our QTMs will have three tapes: a read-only input tape with a two-way tape head, a work tape with a two-way tape head, and a write-only output tape with a one-way tape head. 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, Q and Σ will denote the set of internal states and alphabet of M, respectively. It is assumed that Q contains an initial state q_0 and Σ contains at least the two symbols # (blank) and 1. Input strings are assumed to be strings over some alphabet $\Gamma \subseteq \Sigma \backslash \{\#\}$.

Although we will not include the entire contents of the output tape or the position of the output tape head when measuring the space used by a QTM, we will include this information in the definition of a configuration; 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. We denote the set of 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 zero, 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 $0, \ldots, |x|-1$, and all remaining squares on the input tape contain blanks.

At any instant, the state of a QTM may be described by a superposition of configurations. Formally, a superposition of a QTM M is a unit vector in the Hilbert space $\ell_2(\mathcal{C})$. For a 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 $c \in \mathcal{C}$ will also be referred to as classical states.

In general, the transition function of a QTM M is a mapping of the form

$$\mu : Q \times \Sigma \times \Sigma \times Q \times \{-1,0,1\} \\ \times \Sigma \times \{-1,0,1\} \times (\Sigma \cup \{\varepsilon\}) \to \mathbb{C}.$$

Each number $\mu(q, \sigma_i, \sigma_w, q', d_i, \sigma'_w, d_w, \tau)$ may be interpreted as the amplitude with which M, currently in state q, reading symbol σ_i on its input tape and reading σ_w on its work tape, will change its internal state to q', move its input tape head in direction d_i, write σ'_w on its work tape, move its work tape head in direction d_w and, if $\tau \neq \varepsilon$, write τ on its output tape and move its output tape head to the right. (If $\tau = \varepsilon$, then nothing is written to the output tape and the output tape head remains stationary.) We place the following restriction on allowable transition functions: we assume that for each $\sigma \in \Sigma$ there exists a unitary (i.e. norm preserving and invertible) mapping $V_\sigma : \ell_2(Q \times \Sigma) \to \ell_2(Q \times \Sigma)$, and for each $q \in Q$ there exist $D_i(q)$, $D_w(q) \in \{-1,0,1\}$ and $Z(q) \in \Sigma \cup \{\varepsilon\}$, such that

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

This restriction is analogous to unidirectionality for the single-tape QTM model, discussed in [6]; therein it is shown that this restriction does not decrease the power of QTMs. Similarly, the RTMs considered in [3, 4, 22] obey this restriction (where each V_σ is a permutation in this case). In the interest of simplicity, we prefer to include this restriction as part of the definition of multi-tape QTMs. In short, the restriction requires that the output and movement of tape heads be determined by whatever internal state the machine enters on the step in question. Each V_σ must be unitary in order to insure that the machine is *well-formed* (see below).

It is known that the power of QTMs depends greatly upon the values which the transition function μ may take; in the absence of any restrictions, it is possible to encode a great deal of information in these values. For example, it is shown in [1] that QTMs can recognize non-recursive sets in polynomial time, logarithmic space and with bounded probability of error if allowed to have arbitrary transcendental transition amplitudes. Thus we must place some restriction on these values in order to avoid this problem, and so we will insist that all transition functions of QTMs take only rational values. Although some quantum algorithms use algebraic transition amplitudes, it is shown in [1] that, for the case of bounded error polynomial time, machines with algebraic amplitudes are equivalent in power to ones with rational amplitudes. It is an open question not addressed in this paper whether QTMs with algebraic amplitudes are equivalent in power to ones with rational amplitudes in the case of space-bounded classes.

For given input x and any pair of configurations c and c', μ specifies some amplitude, which we will denote by $\alpha(c \vdash c')$, associated with performing the transition $c \vdash c'$ in the manner described above (if c' is not reachable from c in a single transition, then $\alpha(c \vdash c') = 0$). The *time evolution operator* of M on input x may now be defined as

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

so that if machine M on input x is in superposition $|\psi\rangle$ at some instant, and is allowed to evolve (unobserved) for one step, its new superposition will be $U_x |\psi\rangle$.

A QTM M is said to be well-formed whenever U_x is a unitary operator for every input x. It can be shown that any QTM obeying the restriction on transition functions mentioned above will necessarily be well-formed (following from the fact that each V_σ is unitary). Unitary operators preserve length, and hence we have $\|U_x |\psi\rangle\| = \| |\psi\rangle \| = 1$ for any superposition $|\psi\rangle$ – a property will be important in regard to observations of QTMs, which will now be discussed.

In order for a QTM to reveal any information about its computation, we must assume that it is observed. The information revealed by a particular observation is described by an *observable.* Formally, an observable is any finite or countable collection $\{(P_j, r_j)\}$, where each P_j is a projection operator on $\ell_2(\mathcal{C})$ and each r_j is a *result*, which we will take to be some element of Σ^*. 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 new superposition immediately after the observation will also be of unit norm.

The particular observable which we will be interested in corresponds to simply observing the output tape. As the output tape head moves right one square exactly when a symbol is written to the output tape, the contents of the output tape and the position of the output tape head can be identified with a unique string in Σ^*. For each $w \in \Sigma^*$, 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 any QTM M on input x will 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 according to U_x, then the output tape of the machine is observed as described above. The computation continues until it has been observed that some symbol has been written to the output tape (the output tape head has moved right); if the observed symbol is "1", then the result of the computation is *accept*, and for any other symbol the result is *reject.* For a given QTM M, input x, $k \in \mathbb{N}$ and $\sigma \in \Sigma$, let $p_{x,k,\sigma}$ denote the probability that, if M on input x is run as described above, each observation at time $k' < k$ yields ε and the observation at time k yields σ. The probability that M accepts x is thus $\sum_k p_{x,k,1}$, and the probability that M rejects x is $\sum_k \sum_{\sigma \neq 1} p_{x,k,\sigma}$. If for every input x there exists an N such that $\sum_{k \leq N} \sum_\sigma p_{x,k,\sigma} = 1$, i.e. M halts with certainty after some finite number of steps, then we say that M *halts absolutely.* A straightforward proof by induction shows that

$$p_{x,k,\sigma} = \left\| P_\sigma (U_x P_\varepsilon)^k |c_0\rangle \right\|^2 . \qquad (1)$$

2.3 Space-bounded quantum classes

We will measure the space used by (quantum and classical) Turing machines 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. 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 first symbol (if any) written to the output tape. 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, 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. 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 which has 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 $\langle c \,|\, (U_x P_\varepsilon)^k \,|\, c_0 \rangle \neq 0$ requires space at most l. (Note that the behavior of M on steps subsequent to observing any non-empty string written on the output tape is ignored; the computation has ended once such output is observed.) 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 which 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 which, on input x, outputs $t = 2^{O(s)}$ in binary and runs in space $O(s)$.

Now we may define various complexity classes based on space-bounded QTMs.

Definition 2.1 For $X \in \{\mathrm{EQ}, \mathrm{RQ}, \mathrm{BQ}, \mathrm{NQ}, \mathrm{PrQ}\}$, a given language L is said to be in the class XSPACE(s) if there exists a QTM M which runs in space $O(s)$ and which 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 absolutely, then L is in the class X_HSPACE(s).

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 [23], to which the reader is referred for further information regarding the probabilistic versions of these classes.

3 Relations among quantum classes

In this section, we discuss relationships among the space-bounded quantum classes defined in the previous section. In the two sections which follow, we will examine relationships between these quantum classes and their probabilistic counterparts.

Naturally, for each $X \in \{\mathrm{EQ}, \mathrm{RQ}, \mathrm{BQ}, \mathrm{NQ}, \mathrm{PrQ}\}$ we have X_HSPACE$(s) \subseteq X$SPACE(s). The following containments also follow immediately from the definitions:

$$\begin{aligned}\mathrm{RevSPACE}(s) &\subseteq \mathrm{EQSPACE}(s) \subseteq \mathrm{RQSPACE}(s)\\ &\subseteq \mathrm{BQSPACE}(s) \subseteq \mathrm{PrQSPACE}(s),\end{aligned}$$

and RQSPACE$(s) \subseteq$ NQSPACE(s) (and similarly for the halting classes).

The following lemma will be useful in establishing further relationships between space-bounded quantum classes. The lemma is somewhat more general than will be required in this section, but it will be useful to refer back to it in subsequent sections.

Lemma 3.1 *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. *After precisely t' steps, M' accepts with probability $p_{acc}(x)$ and rejects with probability $\alpha\, p_{rej}(x)$.*
2. *After precisely $t' + 1$ steps, M' accepts with probability β and rejects with probability $1 - \beta$.*

Informally, this lemma states that there exists a QTM which will simulate a given QTM for a given number of steps, possibly suppressing output and acting in the described manner once the simulation is complete. Note that the QTM M' halts absolutely, having a 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. (The values α and β could be taken to be any rational numbers in the range $[0,1]$, but the values above are sufficient for our needs.)

A proof of Lemma 3.1 can be found in the appendix.

An immediate consequence of Lemma 3.1 is that we have $\mathrm{NQ}_H\mathrm{SPACE}(s) \subseteq \mathrm{PrQ}_H\mathrm{SPACE}(s)$ (take $\alpha = 1$, $\beta = \frac{1}{2}$). However, this containment will follow trivially from results proved below. Another simple relation is as follows.

Proposition 3.2 *$NQ_H SPACE(s) = NQSPACE(s)$.*

Proof. For the nontrivial containment, take M to be a QTM running in space s, and for given input x let $|\psi_0\rangle = |c_0\rangle$ and let $|\psi_{k+1}\rangle = U_x P_\varepsilon |\psi_k\rangle$ 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 which 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 $k \leq 2^s$. 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 $k \leq 2^s$. Now apply Lemma 3.1 with $t = 2^s$ and $\beta = 0$. ■

It is known that NSPACE(s) = RSPACE(s), since 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 one). 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 well-formed QTM. However, the following lemma shows that a well-formed quantum machine can perform a process which has a similar outcome. As for the previous lemma, this lemma will also be useful in later sections.

Lemma 3.3 *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 otherwise M' accepts and rejects with probability 0.*

The proof appears in the appendix.

We now have the following proposition, which follows readily from the above lemma.

Proposition 3.4

$$EQSPACE(s) = RQSPACE(s) = NQSPACE(s).$$

It will be demonstrated in the two sections which follow that the class NQSPACE(s) corresponds to the counting class co-$\mathrm{C}_=$SPACE(s), defined analogously to co-$\mathrm{C}_=$L for $s(n) = \log n$ (see [2], and also see Section A.4 in the appendix for an equivalent definition).

4 Quantum simulations of probabilistic machines

In this section, we discuss quantum simulations of probabilistic machines. Given a PTM which runs in space s and which halts absolutely, we show that there exists a QTM running in space $O(s)$ and recognizing the same language. The quantum machine constructed has the property of having bounded error when the same is true of the PTM, but in this case the simulation is quite inefficient in terms of time; the QTM constructed will not halt absolutely and may have expected running time which is doubly exponential in s. In the unbounded error case the QTM may be taken to halt absolutely, having running time $2^{O(s)}$.

The following lemma provides the basis for these relationships.

Lemma 4.1 *Let M be a PTM running in space s and satisfying the properties (1) each non-halting configuration of M has either 1 or 2 successors, (2) for each input x, there is at most one accepting and one rejecting configuration reachable from the initial configuration, and (3) there exists $t : \mathbb{Z}^+ \to \mathbb{N}$ computable in space $O(s)$ such that, on each input x, M halts after precisely t steps on all computation paths. 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 $O(s)$ 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.*

The proof may be found in the appendix. In essence, the quantum machine constructed follows the computation paths of the probabilistic machine with positive amplitudes proportional to the probabilities for each path. As suggested by the probabilities of acceptance and rejection for the quantum machine, the constant of proportionality is very small.

For a given PTM M, we may apply Lemma 3.1 (with $\alpha = 1$ and $\beta = 1/2$) to the QTM M' resulting from Lemma 4.1, and we see that there exists a QTM M'' which halts absolutely and which has probability of acceptance greater than 1/2 if and only if $p_{acc}(x) > p_{rej}(x)$, yielding the following.

Proposition 4.2 *PrSPACE*$(s) \subseteq$ *PrQ$_H$SPACE*(s).

(Here we are relying on the fact that PrSPACE(s) = Pr$_H$SPACE(s), proved in [20] (see also [2]).) In the case that M has probability of error bounded away from 1/2, we may apply Lemmas 3.1 and 3.3 to M' to obtain a QTM which 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.

Proposition 4.3 *BP$_H$SPACE*$(s) \subseteq$ *BQSPACE*(s).

We note that the QTM M' constructed in the proof of Lemma 4.1 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 the following stronger result.

Proposition 4.4 *co-C$_=$SPACE*$(s) \subseteq$ *NQSPACE*(s).

For a proof of this proposition and definition of co-C$_=$SPACE(s), see the appendix.

5 Probabilistic simulations of quantum machines

In this section, we prove that space-bounded quantum machines can be efficiently simulated by space-bounded probabilistic machines in the unbounded error case. Combining these relationships with results of the previous section, we conclude that unbounded error space-bounded quantum and probabilistic machines are equivalent in power. Using similar arguments, the class NQSPACE(s) is shown to correspond directly to the class co-C$_=$SPACE(s).

The following lemma, proved in [2] (see also [10, 28, 29, 30]), is central to our argument.

Lemma 5.1 *Define*

$$L = \left\{ (A,B) \;\middle|\; \begin{array}{l} A, B \textit{ are integer matrices} \\ \textit{satisfying } det(A) > det(B) \end{array} \right\},$$

where pairs of matrices (A,B) *are encoded as binary strings. Then* $L \in$ *PrSPACE*$(\log n)$.

We now relate the probability of acceptance of space-bounded QTMs to the problem of comparing determinants of integer matrices, by means of the following lemma.

Lemma 5.2 *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 and B, where entries of A and B are integers of length $2^{O(s)}$, such that the following properties are satisfied.*

1. *There exists a DTM which, on input x and with integer $k = 2^{O(s)}$ initially written on its work tape, computes the kth bit of the encoding (A,B) in space $O(s)$.*

2. *$det(A) > det(B)$ if and only if M accepts x with probability exceeding $\frac{1}{2}$.*

Once again, the proof may be found in the appendix.

We are now ready for the main result of this section.

Proposition 5.3 *PrQSPACE*$(s) \subseteq$ *PrSPACE*(s).

Proof. Let M be a QTM running in space s, and let matrices A and B be as stated in Lemma 5.2 for each input x. As A and B 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,B) is at most $2^{O(s)}$.

By Lemma 5.1 there exists a PTM M_1 running in log-space and accepting exactly those strings in the language L with probability exceeding 1/2. Now, define a PTM M_2 which, on input x, simulates M_1 on (A,B) 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,B) corresponding to the position of M_2's input tape head, then simulates the action of M_1 given this input symbol. By item 1 of Lemma 5.2, this input symbol can be computed in space $O(s)$. As M_1 runs in space which is logarithmic in the length of (A,B), i.e. in space $O(s)$, it follows that M_2 runs in space $O(s)$ as well.

By the definition of L, along with item 2 of Lemma 5.2, it is clear that M_2 accepts with probability exceeding 1/2 exactly when the same is true of M, which completes the proof. ■

By Propositions 4.2 and 5.3, we have the following corollaries.

Corollary 5.4

$$PrQ_H SPACE(s) = PrQSPACE(s) = PrSPACE(s).$$

Corollary 5.5 *PrQSPACE*$(s) \subseteq$ *DSPACE*(s^2).

In fact, it is shown in [8] that PrSPACE$(\log n) \subseteq$ NC^2, from which it follows that any log-space quantum Turing machine, without restriction on error-probability or running time, can be simulated in NC^2.

Finally, we note the following relationship.

Proposition 5.6 *NQSPACE*$(s) \subseteq$ *co-C*$_=$*SPACE*(s).

Proof. It follows from the proof of Lemma 5.2 that the matrix A has zero determinant if and only if M accepts x with probability zero. As noted in [2], singularity of integer matrices is in C$_=$L, from which it follows that testing non-singularity of A can be performed in co-C$_=$SPACE(s). ■

From Propositions 4.4 and 5.6, we have

Corollary 5.7 *NQSPACE*$(s) =$ *co-C*$_=$*SPACE*(s).

This may be viewed as the space-bounded analogue of the result QNP = co-C$_=$P [13].

6 Conclusion and open problems

Figure 1 is a diagram which summarizes the relationships between some of the quantum and classical space-bounded classes we have discussed in this paper.

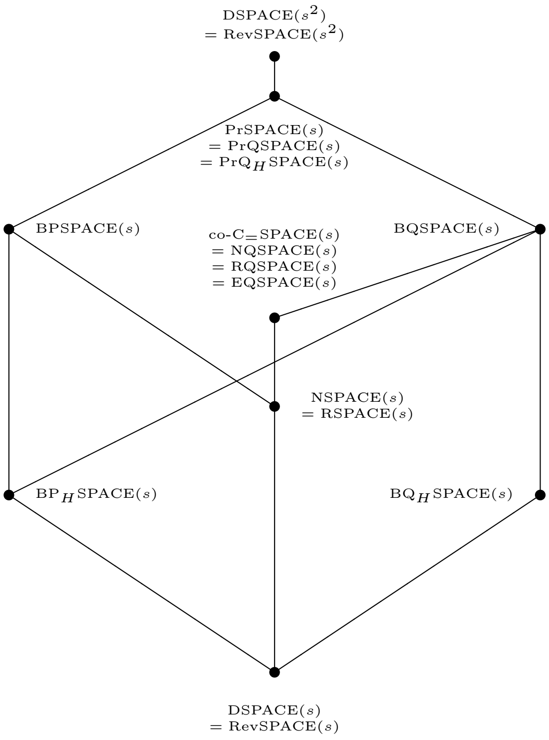

Figure 1: Relationships between quantum and classical space-bounded classes.

Many interesting questions have been left open by this paper. In particular, can probabilistic machines which halt absolutely be efficiently simulated by quantum machines which also halts absolutely for the case of bounded or one-sided error, e.g. do either of the relationships BP$_H$SPACE$(s) \subseteq$ BQ$_H$SPACE(s) or R$_H$SPACE$(s) \subseteq$ RQ$_H$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 RQ$_H$SPACE(s) or BQ$_H$SPACE(s) contained in, say, BPSPACE(s)?

A number of other classical space-bounded classes (e.g. symmetric space, probabilistic classes allowing multiple access to random bits, etc.) have not been mentioned in this paper. Are there natural quantum analogues of these classes, and how do these classes relate to those discussed in this paper?

Finally, we have restricted our attention to space-bounds which 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 [21]. What more can be said about sub-logarithmic space-bounds?

A Appendix

A.1 Proof of Lemma 3.1

Let Q, Σ and μ denote the state set, alphabet and transition function of QTM M running in space s, where we assume μ can be specified by V_σ, $D_i(q)$, $D_w(q)$, and $Z(q)$ for each $\sigma \in \Sigma$ and $q \in Q$, as described in Section 2. We will define a QTM M' which satisfies the requirements in the statement of the lemma.

Each internal state of M' will be of the form (q, τ, r), where $q \in Q$, $\tau \in \Sigma$ and r is one of a collection of states allowing M' to behave in the manner described below. The initial state of M' is assumed to be of the form $(q_0, \#, r_0)$ for some r_0. The work tape of M' will consist of six tracks, which will be used as follows.

Track 1: Records the position of the input tape head of M.
Track 2: Records the position of the work tape head of M.
Track 3: Represents the contents of the work tape of M.
Track 4: Records the number of steps of M which have thus far been simulated.
Track 5: Records the time at which M halts (or 0 if M has not halted).
Track 6: Records whether M has accepted, rejected or neither.

The integers stored on tracks 1, 2, 4 and 5 are to be encoded as binary strings beginning in square 0 in such a way that (1) the empty string represents 0, and (2) the integers and representations are in one-to-one correspondence (see [14], for example, for such an encoding). By this assumption, these tracks all initially

1. Execute the following loop with starting/stopping condition "track 4 contains 0":

   i. Increment the number on track 4 modulo $t(|x|)$.

   ii. If track 5 contains 0 (M has not yet halted), simulate M for 1 step. Specifically:

      a. Swap the tape symbol currently represented in the internal state of M' with the symbol on track 3 at the position encoded on track 2.

      b. Perform the transformation V_σ on the state/symbol pair currently represented in the internal state of M', where σ is the symbol on the input tape at the location recorded on track 1.

      c. Again swap the tape symbol currently represented in the internal state of M' with the symbol on track 3 at the position encoded on track 2.

      d. Letting q denote the state of M currently represented by M', add $D_i(q)$ to the number on track 1 and add $D_w(q)$ to the number on track 2.

   iii. 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, and if track 6 is empty (i.e. square 0 on track 6 contains #), then add the number on track 4 to the number on track 5 (recording the halting time).

   iv. If the numbers on tracks 4 and 5 are equal (M is halting on this step), perform a reversible transformation on square 0 of track 6, mapping $\# \mapsto 1$ if $Z(q) = 1$ and $\# \mapsto 0$ if $Z(q) \in \Sigma \backslash \{1\}$ (otherwise, leave track 6 unchanged).

2. If track 6 contains the symbol 1, output 1 (accept). In the case that $\alpha = 1$, output 0 (reject) if track 6 contains the symbol 0.

3. In case $\beta = 0$, output 0 (reject). Otherwise, simulate a fair coin-flip and output 1 (accept) or 0 (reject) accordingly.

Figure 2: Description of M' for Lemma 3.1.

encode the number 0, as $(\#, \ldots, \#)$ is taken to be the blank symbol of M'. Track 6 will represent acceptance, rejection, or neither depending on whether square 0 contains a 1, 0, or #, respectively.

The manner in which M' functions is described in Figure 2. It is assumed that after each step is performed, the input and work tape heads of M' return to the tape squares indexed by 0.

In order for M' to correctly simulate M, we must take care to insure that computation paths of M having equal length have corresponding paths in the computation of M' which also have equal length; otherwise M' may not produce the same "interference patterns" as M. For this reason, we will insist that each step in Figure 2 require an amount of time which is invariant over all computation paths of M consisting of configurations reachable with nonzero amplitude. Furthermore, in order to insure that M' is well-formed, each step (save the "quantum" steps b. and 3., which are commented on below) must be performed reversibly. We claim that each step can be performed in space $O(s)$ in a manner which is in compliance with these requirements, although this will not be shown in detail. For steps a, c and d, we will necessarily need to rely on the fact that M runs in space s to do this (in step a, for example, M' may move its tape head over each square in the range $\{-s, \ldots, s\}$, and perform the required swap only when some particular control bit is set, say). (Note that space constructibility of s, along with [22], implies that M' can compute $s(|x|)$ in space $O(s)$, etc.) For the remaining deterministic steps, this is straightforward.

Next we note that the loop (step 1) can be performed reversibly. Since this is a construct which is used in later proofs, we will discuss this situation in more generality. A loop consisting of a sequence of reversible (or quantum) actions can be performed reversibly given that there is a single starting/stopping condition for that loop: in this paper the condition will always be that a particular track on the work tape encodes 0 (i.e. square 0 on that track contains a blank). Suppose that the body of the loop corresponds to a sequence of actions beginning with state r_1 and ending with state r_1', and that the state of the machine immediately prior to (possibly) executing the loop is r_0 and immediately after executing the loop is r_0'. By defining the transition function of the machine so that the following reversible transformation on states is induced, the loop will be executed as required.

$$r_0 \mapsto \begin{cases} r_1 & \text{symbol being read on given track is \#} \\ r_0' & \text{symbol being read on given track is not \#} \end{cases}$$

$$r_1' \mapsto \begin{cases} r_0' & \text{symbol being read on given track is \#} \\ r_1 & \text{symbol being read on given track is not \#} \end{cases}$$

Note that we may substitute collections of states, in the present case states of the form (q,τ,r_0), (q,τ,r_0'), etc., for states r_0, r_0', etc. in the above transformations, allowing M' to store q and τ internally while still performing the loop reversibly.

Finally, we note that the two "quantum" steps (step b and, when $\beta = \frac{1}{2}$, step 3) are easily performed by M' which is well-formed. For step b this is obvious; each V_σ is unitary, so this step requires a single quantum step of M', and involves only the internal state of M', once the appropriate symbol on the input tape has been located. In order to simulate the coin-flip in step 3, we may use the 4-dimensional Hadamard transform H_4 on states $\{0,1,2,3\}$:

$$H_4 : |a\rangle \rightarrow \frac{1}{2}\sum_{b=0}^{3}(-1)^{(a,b)}\,|b\rangle \qquad (2)$$

where (a,b) denotes the number of 1's in the bitwise-and of a and b written in binary. The result of the coin-flip may be defined to be "heads" when the transformation (applied to an initially 0 state) results in 0 or 1, and " tails" otherwise. (Here we choose the 4-dimensional Hadamard transform rather than the usual 2-dimensional one since we are restricting our attention to machines with rational amplitudes.)

It is straightforward to see that M' mimics the behavior of M. In the case that M yields some accepting or rejecting configuration during the t steps simulated, M' records the step at which this occurs as well as the particular accepting or rejecting configuration reached (since the configuration represented does not change after some particular halting time has been recorded). It follows that the probabilities with which M' accepts and rejects on step 2 are $p_{acc}(x)$ and $p_{rej}(x)$, respectively. Defining $t' = t'(|x|)$ to be the number of steps required for M' to complete step 2, we have that t' is clearly computable in space $O(s)$, which completes the proof. ■

A.2 Proof of Lemma 3.3

M' will simulate M for t steps in a manner similar to the machine constructed in the proof of Lemma 3.1. In this case, however, the simulation will be repeated ad infinitum so as to to amplify the probabilities of acceptance and rejection accordingly. The problem, of course, is that we cannot simply restart the simulation after t steps have passed, since deleting any left over information from the previous simulation constitutes an irreversible action, resulting in a machine which is not well-formed. This problem can be eliminated in the manner described below.

The work tape of M' will consist of six tracks, precisely as in the proof of Lemma 3.1. Similarly the internal states of M' will be of the same form as in that proof. The behavior of M' is described in Figure 3.

Under the assumption that M runs in space s, the work tape head of M never leaves the region $\{-s,\ldots,s\}$ with nonzero amplitude when started from the initial configuration. However, there is no guarantee that the same is true when the simulation is inverted in step 3; output may occur in step 2, introducing the possibility that space-expensive paths may be followed with nonzero amplitude during step 3. To remedy this, we substitute the transformation

$$k \mapsto [\,(k + D_w(q) + s) \bmod (2\,s+1)\,] - s,$$

for adding $D_w(q)$ to k (denoting the position of the work tape head of M stored on track 2 of M') in the simulation. Since this transformation is the same as adding $D_w(q)$ to k in the forward direction of the simulation, the correct probabilities of acceptance and rejection are preserved. In the reverse direction, the invariant $k \in \{-s,\ldots,s\}$ is preserved, so that only space $O(s)$ configurations of M are reached by M'. (The inverse of the simulation may result in nonlocal behavior, e.g. the work tape head may move from from square $-s$ to square s in a single step, but this is irrelevant.) It follows that the entire simulation can be performed in space $O(s)$.

Now we show that the claimed probabilities of acceptance and rejection are obtained. For fixed input x, let c_0' denote the initial configuration of M', and

1. Execute the following loop with starting/stopping condition "track 4 contains 0":

   i. Increment the number on track 4 modulo $t(|x|)$.

   ii. If track 5 contains 0 (M has not yet halted), then simulate M for 1 step. This will be done exactly as in the proof of Lemma 3.1, except that instead of simply adding $D_w(q)$ to the number represented on track 2 (here denoted k), the following transformation is performed:
   $$k \mapsto [\,(k + D_w(q) + s) \mod (2\,s + 1)\,] - s,$$
   whenever $k \in \{-s, \ldots, s\}$ (and $k \mapsto k$ otherwise).

   iii − iv. Same as in the proof of Lemma 3.1.

2. If track 6 contains the symbol 1, output 1 (accept). If track 6 contains the symbol 0, output 0 (reject). (Otherwise, do not produce any output.)

3. Perform the inverse of step 1.

4. If the current configuration of M represented is not the initial configuration (i.e. track 1 or 2 does not contain 0, squares $-s, \ldots s$ of track 3 do not all contain blanks, or the internal state represented by M' is not q_0), then multiply the current amplitude by -1.

5. Goto step 1.

Figure 3: Description of M' for Lemma 3.3.

let F be the operator which corresponds to performing step 1, i.e. simulating M for t steps. Since M' does not produce output during step 1, F is unitary. Let $|\psi\rangle = F\,|c_0'\rangle$, and 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 square 0 of track 6 contains 1, 0 or #, respectively. We have $\|\,|\psi_1\rangle\,\|^2 = p_{acc} = p_{acc}(x)$ and $\|\,|\psi_0\rangle\,\|^2 = p_{rej} = p_{rej}(x)$. During step 2, M' outputs 1, 0 or ε (no output) accordingly, and hence accepts with probability p_{acc} and rejects with probability p_{rej}. Otherwise, the superposition collapses to $|\psi_\varepsilon\rangle$ (renormalized) 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 the third component r of the internal state of M' is different, reflecting the fact that we are now at step 4 rather than step 1, etc.). 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

$$F^{-1}\,|\psi_\varepsilon\rangle = (1 - p_{acc} - p_{rej})\,|c_0'\rangle - |\xi_1\rangle - |\xi_0\rangle\,,$$

and $\langle\, c_0' \,|\, \xi_1\rangle = \langle\, c_0' \,|\, \xi_0\rangle = 0$. Thus, after applying step 4 and returning to step 1, the state of the machine is $(1-p_{acc}-p_{rej})\,|c_0'\rangle + |\xi_1\rangle + |\xi_0\rangle$. After again performing the simulation in step 1, the new superposition of M' will be

$$\begin{aligned}(1 - p_{acc} - p_{rej})F\,|c_0'\rangle + F\,|\xi_1\rangle + F\,|\xi_0\rangle \\ = \quad & (2 - 2p_{acc} - 2p_{rej})\,|\psi_1\rangle \\ & + (2 - 2p_{acc} - 2p_{rej})\,|\psi_0\rangle \\ & + (1 - 2p_{acc} - 2p_{rej})\,|\psi_\varepsilon\rangle\,.\end{aligned}$$

After k iterations of the loop ($k \geq 2$), the superposition of M' will therefore be

$$\begin{aligned}& (1 - 2p_{acc} - 2p_{rej})^{k-2}(2 - 2p_{acc} - 2p_{rej})\,|\psi_1\rangle \\ + \quad & (1 - 2p_{acc} - 2p_{rej})^{k-2}(2 - 2p_{acc} - 2p_{rej})\,|\psi_0\rangle \\ + \quad & (1 - 2p_{acc} - 2p_{rej})^{k-1}\,|\psi_\varepsilon\rangle\,.\end{aligned}$$

Based on equation (1), the probability that M' accepts may now be calculated as

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

and similarly for the probability that M' rejects. ■

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. Execute the following loop with starting/stopping condition "track 4 contains 0":
    - i. Increment the number on track 4 modulo $t(|x|)$.
    - 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' which is a successor of c, then increment the number on track 3.
    - 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, increment the number on track 3.
3. If track 3 contains 0 and track 1 encodes an accepting configuration, then output 1 (accept). If track 3 contains 0 and track 1 encodes a rejecting configuration, then output 0 (reject).
4. Output 0 (reject).

Figure 4: Description of M' for Lemma 4.1

A.3 Proof of Lemma 4.1

Given a PTM M as in the statement of the Lemma, we construct a QTM M' which follows the computation paths of M, each with positive amplitude proportional to the corresponding probability for that path in the computation of M. In order to do this with a well-formed machine, the constant of proportionality will decrease as the simulation proceeds, being on the order of 2^{-sk} during the kth step of the simulation.

The work tape of M' will consist of four tracks: tracks 1 and 2 will be used to encode configurations of M, track 3 will contain a counter which is described below, and track 4 will record the number of steps for which M has been simulated. The tape symbols used on tracks 1 and 2 are elements of the set $\{\#, 0, 1, 2, 3\}$. Integers on tracks 3 and 4 are encoded as in the proof of Lemma 3.1. Included in the internal state of M' is a variable a which may take values in $\{0, 1, 2, 3\}$, and has initial value 0.

The behavior of M' is described in Figure 4. The transformation H_4 is defined in (2).

Throughout the computation of M', the integers encoded on tracks 3 and 4 must be at most $2t$ and $t-1$, respectively, and will therefore have length $O(s)$ encodings. After step 1 is performed, tracks 1 and 2 will always contain length s strings, and consequently the contents of each track of M' will have length $O(s)$. Furthermore, each step is readily seen to be performable by a well-formed QTM within the required $O(s)$ space bound.

Again we insist that the time required for each step does not depend on the configurations of M represented by M' during those particular steps. It follows that there exists t', as in the statement of the lemma, which is the number of steps required for M' to complete step 3 along every computation path.

Now we will determine the probability with which M' accepts and rejects after t' steps. The counter on track 3 acts as a flag; whenever the number represented on track 3 is nonzero, the simulation has failed (a counter is used so that this can be done reversibly). We will say that any configuration of M' is *good* whenever track 3 contains 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 configura-

1 – 2. Same as in the proof of Lemma 4.1 (see Figure 4).

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 contains all zeroes, add 1 to the number written on track 3.

5. Perform H_4 on a. If track 3 contains 0 and now $a = 1$, then output 1 (accept), otherwise output 0 (reject).

Figure 5: Description of M' for Proposition 4.4.

tion 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 which 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 mapped to 0). 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 satisfies the assumptions in the statement of the lemma, 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 be $(2^{-2st})p_{acc}(x)$ and $(2^{-2st})p_{rej}(x)$, respectively. Hence, 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 claimed. ■

A.4 Proof of Proposition 4.4

In the proof of Proposition 4.4 we will use the following definition for $\mathrm{C_{=}SPACE}(s)$.

Definition A.1 The class $\mathrm{C_{=}SPACE}(s)$ consists of all languages L for which there exists a PTM M, running in space $O(s)$ and satisfying conditions (1) - (3) of Lemma 4.1, such that $x \in L$ if and only if M accepts x with probability exactly $\frac{1}{2}$.

We refer the reader to [2] for further discussion regarding $\mathrm{C_{=}L}$; the log-space representative of these classes.

Proof of Proposition 4.4. Let $L \in \mathrm{C_{=}SPACE}(s)$, and let M be a PTM for L in the sense of Definition A.1. Define a QTM M' in a similar manner to the machine constructed in the proof of Lemma 4.1, but modified as described in Figure 5.

We now determine the probability with which M' accepts each input x. Recall the definition of a good configuration from the proof of Lemma 4.1. Since M must be in the unique accepting or unique rejecting configuration after t steps, there are 2 good configurations which 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 two configurations are equal, since tracks 1 and 2 may contain only zeroes at this point). The amplitudes associated with these two configurations are $2^{-s(2t+1)}p_{acc}$ and $2^{-s(2t+1)}p_{rej}$, 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 good configuration for which $a = 1$ if and only if $p_{acc} \neq p_{rej}$. Hence M' accepts with nonzero probability if and only if $x \notin L$. ■

A.5 Proof of Lemma 5.2

The proof of Lemma 5.2 will follow from Lemmas A.1 and A.2, stated and proved below.

Lemma A.1 *Let M be a QTM running in space s. Then for each input x, there exists a $(2^{2s}+2) \times (2^{2s}+2)$ matrix E such that for every $k \geq 0$, $E^{k+2}[2^{2s}+2, 1]$ is the probability that M accepts x after precisely k steps (and $E[2^{2s}+2, 1] = 0$). Furthermore, the following properties are satisfied.*

1. *Entries of E may be written in the form $\frac{a_{ij}}{m}$, where $a_{ij} \in \{-m, \ldots, m\}$ and m is the square of the least common denominator of the values taken by M's transition function.*

2. *There exists a DTM M_0 which, on input x and with indices $i, j \in \{1, \ldots, 2^{2s}+2\}$ initially written on it's work tape,computes the value $m \cdot E[i,j]$ in space $O(s)$.*

3. *All eigenvalues of E are bounded in absolute value by 1.*

Proof. Given QTM M, let us fix input x and let U_x, P_1 and P_ε be as defined in Section 2.2. Define a $2^s \times 2^s$ matrix D as follows.

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

where we identify integers in the range $\{0, \ldots, 2^s - 1\}$ with their encodings as length s binary strings in the obvious way. D may be viewed as the matrix representation of $U_x P_\varepsilon$, restricted to space s configurations. Next, define 2^{2s}-dimensional vectors y_{init} and y_{acc} as follows.

$$y_{init}[i] \;=\; \begin{cases} 1 & i = i_0 + (i_0 - 1)2^s, \text{ where} \\ & (i_0 - 1) \in \{0, \ldots, 2^s - 1\} \\ & \text{encodes } c_0 \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{acc}[i] \;=\; \begin{cases} 1 & i = i_0 + (i_0 - 1)2^s, \text{ where} \\ & (i_0 - 1) \in \{0, \ldots, 2^s - 1\} \\ & \text{encodes } c \in \mathcal{C}_{acc} \\ 0 & \text{otherwise.} \end{cases}$$

Here we write $\mathcal{C}_{acc}$ to denote the set of accepting configurations of M (i.e. the first square on the output tape contains the symbol 1). Finally, define E as follows.

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

Note that E is of the following form:

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

where $\otimes$ denotes the Kronecker product. For $k \in \mathbb{Z}^+$, E^{k+2} has the form

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

It follows that, for $(j_{init} - 1) \in \{0, \ldots, 2^s - 1\}$ encoding the initial configuration of M on x, we have

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

given that M runs in space s. Hence we have that $E^{k+2}[2^{2s} + 2, 1]$ is the probability that M accepts x after precisely k steps.

It remains to show that items $1 - 3$ in the statement of the lemma are satisfied. Item 1 is obvious from the definition of E, and item 2 is straightforward given reasonable assumptions on our encoding of configurations. For item 3, note that D may be viewed as a unitary operator composed with various projection operators (corresponding to P_ε and also corresponding to the fact that only space s configurations are present). It follows that D cannot possibly increase length, and consequently all of its eigenvalues are bounded in absolute value by 1. Thus the same is true for $D \otimes D$, and hence for E (since any nonzero eigenvalue of E must also be an eigenvalue of $D \otimes D$). ■

Lemma A.2 *For each positive integer m, there exist polynomials $p, q : \mathbb{Z}^+ \to \mathbb{Z}^+$ satisfying the following. For 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) $\left(1 + 2^{-p(N)}\right)\left(I - \left(1 - 2^{-q(N)}\right) E\right)$ is invertible, and (ii) for all values of i, j for which $\lim_{z \uparrow 1} (I - z\,E)^{-1}[i,j]$ exists,*

$$\lim_{z \uparrow 1} (I - z\,E)^{-1}[i,j] \;>\; \frac{1}{2}$$

if and only if

$$\left(\left(1 + 2^{-p(N)}\right) \left(I - \left(1 - 2^{-q(N)}\right) E\right) \right)^{-1} [i,j] \;>\; \frac{1}{2}.$$

The proof of Lemma A.2 will be made simpler by the lemmas which follow. The following notation will be used in these lemmas: 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 A.3 *For any two polynomials f and g, we have*

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

Proof. Straightforward. ■

Lemma A.4 *Let $Q(z)$ be an $N \times N$ matrix having entries which are polynomials in z with degree at most 1 and height bounded by m. Then*

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

Proof. The product of any N entries of $Q(z)$ must have height bounded by $m^N 2^{N-1}$, which follows from $N-1$ applications of Lemma A.3. 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 $N!\, m^N 2^{N-1}$. ■

Lemma A.5 *Let f and g be integer polynomials satisfying $(1-z)^k f(z) = g(z)$. Then $\|f\| \le \|g\| \binom{\deg(g)}{k}$.*

Proof. Given integer polynomial $g(z) = \sum_{j=0}^{n} a_j z^j$ which 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 which follows from the power series expansion $1/(1-z)^k = \sum_{j \ge 0} \binom{j+k-1}{j} z^j$. Consequently

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

as claimed. ■

Lemma A.6 *Let f be a polynomial and let $\epsilon \in (0,1)$. Then*

$$|f(1) - f(1-\epsilon)| \;\le\; \epsilon\, \|f\| \binom{\deg(f)+1}{2}.$$

Proof. First, we note that for $j \ge 1$,

$$\frac{1-(1-\epsilon)^j}{\epsilon} \;=\; \frac{(1-\epsilon)^j - 1}{(1-\epsilon)-1} \;=\; \sum_{i=0}^{j-1} (1-\epsilon)^i \;<\; j.$$

Now, write $f(z) = \sum_{j=0}^{k} a_j z^j$. We have

$$\begin{aligned} |f(1) - f(1-\epsilon)| \;&=\; \left| \sum_{j=1}^{k} a_j (1-(1-\epsilon)^j) \right| \\ &\le\; \epsilon\, \|f\| \sum_{j=1}^{k} \left(\frac{1-(1-\epsilon)^j}{\epsilon} \right) \\ &\le\; \epsilon\, \|f\| \binom{k+1}{2}, \end{aligned}$$

as required. ■

Lemma A.7 *Let f and g be integer polynomials satisfying $\|f\|, \|g\| \le K$, $\deg(f), \deg(g) \le N$, and $g(1) \ne 0$, and suppose that $0 < \epsilon < \left(2K\binom{N+1}{2}\right)^{-1}$. Then*

$$\left| \frac{f(1)}{g(1)} - \frac{f(1-\epsilon)}{g(1-\epsilon)} \right| \;<\; 4\epsilon\,(N+1)^3 K^2.$$

Proof. By Lemma A.6, we have

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

implying that $|g(1-\epsilon)| > 1/2$ since $g(1)$ is a nonzero integer. Now, again by Lemma A.6, we have

$$\begin{aligned} &\left| \frac{f(1)}{g(1)} - \frac{f(1-\epsilon)}{g(1-\epsilon)} \right| \\ &\quad\le\; \left| \frac{f(1)g(1-\epsilon) - f(1)g(1)}{g(1)g(1-\epsilon)} \right| \\ &\quad\quad +\; \left| \frac{f(1)g(1) - f(1-\epsilon)g(1)}{g(1)g(1-\epsilon)} \right| \\ &\quad<\; 2|f(1)|\,|g(1) - g(1-\epsilon)| \\ &\quad\quad + 2|g(1)|\,|f(1) - f(1-\epsilon)| \\ &\quad<\; 4\epsilon\,(N+1)^3 K^2 \end{aligned}$$

as claimed. ■

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

Now, for fixed i and j, define

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

where $(I - z\,E)_{j,i}$ denotes $(I - z\,E)$ with row j and column i removed. 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

$$\begin{aligned} u(z) \;&=\; (1-z)^k f(z), \\ v(z) \;&=\; (1-z)^k g(z), \end{aligned}$$

for integer polynomials f and g, $g(1) \ne 0$, so that $\frac{f(z)}{g(z)} = (I - z\,E)^{-1}[i,j]$ for $|z| < 1$, and $\frac{f(1)}{g(1)} = \lim_{z\uparrow 1}(1 - z\,E)^{-1}[i,j]$.

By Lemma A.4 we have $\|u\|, \|v\| \leq N!\, m^N 2^{N-1}$, and thus $\|f\|, \|g\| \leq N!\, m^N 2^{2N-1}$ by Lemma A.5. Consequently, if $\frac{f(1)}{g(1)} > \frac{1}{2}$, then

$$\frac{f(1)}{g(1)} - \frac{1}{2} \;\geq\; \left|\frac{1}{2\,g(1)}\right| \;\geq\; \left(N!\, m^N 2^{2N}\right)^{-1}. \quad (3)$$

Furthermore, by Lemma A.7 we have

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

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

Now define $p(N) = 4mN^2$ and $q(N) = 16mN^2$. We clearly have that $2^{-p(N)} < (N!\, m^N 2^{2N})^{-1}$ and $2^{-q(N)} < \frac{1}{2}\, 2^{-p(N)}\, ((N+1)^3 (N!)^2 m^{2N} 2^{4N})^{-1}$. It follows from (4) that

$$\begin{aligned}\left|\frac{f(1)}{g(1)} - \frac{f(1-2^{-q(N)})}{g(1-2^{-q(N)})}\right| \;&<\; \frac{1}{2}\, 2^{-p(N)} \\ &<\; \frac{1}{2}\,(N!\, m^N 2^{2N})^{-1}. \quad (5)\end{aligned}$$

Now, in the case that $\frac{f(1)}{g(1)} \leq \frac{1}{2}$ we see that

$$\begin{aligned}&\frac{f(1-2^{-q(N)})}{(1+2^{-p(N)})g(1-2^{-q(N)})} \\ &\quad<\; \frac{1}{1+2^{-p(N)}}\left(\frac{1}{2} + \frac{1}{2}\,2^{-p(N)}\right) \;=\; \frac{1}{2},\end{aligned}$$

and in the case that $\frac{f(1)}{g(1)} > \frac{1}{2}$ we have

$$\begin{aligned}&\frac{f(1-2^{-q(N)})}{(1+2^{-p(N)})g(1-2^{-q(N)})} \\ &\quad>\; \frac{1}{1+2^{-p(N)}}\left(\frac{1}{2} + \frac{1}{2}\,(N!\, m^N 2^{2N})^{-1}\right) \\ &\quad>\; \frac{1}{1+2^{-p(N)}}\left(\frac{1}{2} + \frac{1}{2}\,2^{-p(N)}\right) \;=\; \frac{1}{2}\end{aligned}$$

by (3) and (5). Since

$$\begin{aligned}&\left(\left(1+2^{-p(N)}\right)\left(I - \left(1-2^{-q(N)}\right)E\right)\right)^{-1}[i,j] \\ &\qquad=\; \frac{f(1-2^{-q(N)})}{(1+2^{-p(N)})g(1-2^{-q(N)})},\end{aligned}$$

the lemma follows. ■

Proof of Lemma 5.2. For fixed input x, let E and m be as in Lemma A.1, write $N = 2^{2s} + 2$, and let p and q be as in Lemma A.2. Define F and G to be $N \times N$ integer matrices as follows.

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

and

$$F = \begin{bmatrix} G_{1,N} & 0 \\ 0 & -2\,m\,2^{p(N)+q(N)} \end{bmatrix}.$$

We have

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

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

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

since E has eigenvalues bounded in absolute value by 1 (see [17], p. 54 for example) from which we conclude

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

by Lemma A.1. It follows that $\frac{\det(F)}{\det(G)} > 1$ if and only if M accepts x with probability greater than $\frac{1}{2}$. We do not know what the signs of $\det(F)$ and $\det(G)$ are, and so we define

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

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

The length in binary of each entry of A and B can be seen to be $2^{O(s)}$, so (A, B) can be encoded as a binary string of length $2^{O(s)}$. It is straightforward to see that, for each i, j, the kth bit of both $A[i,j]$ and $B[i,j]$ can be computed in space $O(s)$ as required. This completes the proof. ■

Acknowledgments

I would like to thank Eric Bach and Anne Condon for their helpful comments and advice.

References

[1] L. Adleman, J. Demarrais, and M. Huang. Quantum computability. *SIAM Journal on Computing*, 26(5): 1524–1540, 1997.

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

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

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

[5] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

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

[7] A. Berthiaume and G. Brassard. The quantum challenge to structural complexity theory. In *Proceedings of the 7th Annual IEEE Conference on Structure in Complexity*, pages 132–137, 1992.

[8] 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.

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

[10] C. Damm. $DET = L^{\#L}$? Informatik-Preprint 8, Fachbereich Informatik der Humboldt-Universität zu Berlin, 1991.

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

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

[13] S. Fenner, F. Green, S. Homer, and R. Pruim. Determining acceptance possibility for a quantum computation is hard for PH. Preprint, 1997.

[14] L. Fortnow. Counting complexity. In L. Hemaspaandra and A. Selman, editors, *Complexity Theory Retrospective II*, pages 81–107. Springer, 1997.

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

[16] L. Grover. A fast quantum mechanical algorithm for database search. In Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, pages 212–219, 1996.

[17] A. Householder. *The Theory of Matrices in Numerical Analysis*. Blaisdell Publishing Company, New York, 1965.

[18] 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.

[19] H. Jung. On probabilistic tape complexity and fast circuits for matrix inversion problems. In *Proceedings of the 11th International Colloquium on Automata, Languages and Programming*, volume 172 of *Lecture Notes in Computer Science*, pages 281–291, 1984.

[20] 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*, 310–317, 1985.

[21] 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.

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

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

[24] 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.

[25] P. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

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

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

[28] S. Toda. Counting problems computationally equivalent to the determinant. Technical Report CSIM 91-07, University of Electro-Communications, Tokyo, 1991.

[29] L. Valiant. Why is Boolean complexity theory difficult? In *Boolean Function Complexity*, volume 169 of *London Mathematical Society Lecture Notes Series*, pages 84–94, 1992.

[30] V. Vinay. Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *Proceedings of the 6th Annual Structure in Complexity Theory Conference*, pages 270–284, 1991.

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