Relationships Between Quantum and Classical Space-Bounded Complexity Classes

John Watrous

Technical Report #1357

December 1997

Relationships between quantum and classical space-bounded complexity classes

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

December 5, 1997

Abstract

In this paper, we discuss the relative power of quantum and classical (probabilistic) machines when the limiting resource is space rather than time. In particular, quantum simulations of probabilistic machines and probabilistic simulations of quantum machines are presented which imply the following relationships.

(1) Any probabilistic Turing machine (PTM) which runs in space s and which halts absolutely (i.e. halts with certainty after some finite number of steps) can be simulated in space $O(s)$ by a quantum Turing machine (QTM). If the PTM has probability of error bounded away from $1/2$, then the same is true of the QTM. In the case of unbounded error, the quantum machine may be taken to halt absolutely, but for the bounded error case the QTM will not necessarily 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, and that any space s QTM can be simulated deterministically in space $O(s^2)$. 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, 8, 13, 17, 25, 26, 27], for instance). In this paper, we consider the relative power of quantum and classical machines when the limiting resource is space rather than time. In particular, we define quantum complexity classes which are analogous to classes traditionally studied in the context of space-bounded probabilistic computation, and prove that various relationships hold amongst these classical and quantum classes. By studying quantum versions of space-bounded complexity classes, we hope not only to better understand the strengths and limitations of quantum computational models, but also to possibly shed new light on the classical versions of these classes.

The model for quantum computation which we use in this paper is the quantum Turing machine (QTM) model, first formally defined by Deutsch [12], and also discussed in [5, 6, 31]. Specifically, 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 situation in which space is the limiting resource, 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 will restrict our attention to QTMs which have rational transition amplitudes.

Although we are not aware of any previous work on space-bounded quantum complexity classes, previous work on reversible computation is quite relevant to our discussion. The reason for this is that reversible computation represent the overlap between classical and quantum computation; a deterministic Turing machine (DTM) directly corresponds to a QTM, and vice versa, exactly when the machine in question is a reversible Turing machine (RTM) (i.e. a DTM for which each configuration may have only one predecessor). It was proved by Bennett in [3] 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 [4]. We may state this result as $\mathrm{DSPACE}(s) \subseteq \mathrm{RevSPACE}(s^2)$, where $\mathrm{RevSPACE}(s)$ denotes the class of languages recognizable in space $O(s)$ by some RTM. It was subsequently proved in [10] that nondeterministic Turing machines can also be simulated reversibly with the same increase in space, i.e. $\mathrm{NSPACE}(s) \subseteq \mathrm{RevSPACE}(s^2)$. 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. $\mathrm{DSPACE}(s) = \mathrm{RevSPACE}(s)$.

Given that $\mathrm{DSPACE}(s) = \mathrm{RevSPACE}(s)$, we may deduce various relationships between classical and quantum space-bounded classes by considering deterministic simulations of probabilistic machines. Independently, Jung [18] and Borodin, Cook and Pippenger [9] 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. $\mathrm{PrSPACE}(s) \subseteq \mathrm{DSPACE}(s^2)$. This implies that RTMs, and hence QTMs, can also simulate PTMs with at most a quadratic increase in space. Along similar lines, it was proved in [24] that any bounded error PTM which 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 whether 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 (which follows 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 without requiring the QTM to have multiple access to its (simulated) coin-flips. Since a PTM can perform useful work even after a number of steps which is doubly exponential in its space-bound [16], a QTM storing the values of the required number of coin-flips would, in the worst case, require considerably more space than the $O(s^2)$ deterministic simulation mentioned above.

Using a different 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 $O(s)$. As corollaries, we have that unbounded error space-bounded PTMs and QTMs are equivalent in power, and that any QTM running in space s can be simulated deterministically in space $O(s^2)$. Further, it follows 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 proof of these relationships uses a technique similar to the one used in the probabilistic case; the problem of determining whether or not 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 multi-tape quantum Turing machine model and space-bounded quantum complexity classes which will be used throughout the paper. Next, in Section 3 we discuss some of the relationships which hold amongst 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 will appear in an appendix following Section 6.

Notation

We will mention here some of the notation which is used in this paper. $\mathbb{N}$, $\mathbb{Z}$ 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 of vectors indexed by the set S. 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 $\langle \phi \,|\, \psi \rangle$ (skew-linear in the first coordinate rather than the second).

2 Preliminaries

Multitape quantum Turing machines

In this section, we 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 elements over some alphabet Γ, where $\Gamma \subseteq \Sigma \backslash \{\#\}$.

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 all configurations of a QTM M by $\mathcal{C}(M)$ (or just $\mathcal{C}$ if M is understood from context). The initial configuration of a machine M, denoted c_0, is that configuration in which the internal state of M 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 is a mapping of the form

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

The interpretation is as follows: $\mu(q, \sigma_i, \sigma_w, q', d_i, \sigma'_w, d_w, \tau)$ is the amplitude with which a machine, currently in state q, reading symbol σ_i on its input tape and reading σ_w on its work tape, will move the input tape head in direction d_i, write σ'_w on the work tape, move the work tape head in direction d_w and, if $\tau \neq \varepsilon$, write τ on the output tape and move the 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 will 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 $D_i(q)$ and $D_w(q)$ 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 condition says that the output and movement of tape heads is 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 may take only rational values. Although many 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

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

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 $\ell_2(C)$ 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 previously mentioned, superpositions are of unit norm, so it follows that the probabilities in (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 contents of the output tape. Since 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*.[1] With a given QTM M, input x, $k \in \mathbb{N}$ and $\sigma \in \Sigma$, we may therefore associate a probability $p_{M,x,k,\sigma}$, which is 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 $\sum_k p_{M,x,k,1}$, and the probability that M rejects x is $\sum_k \sum_{\sigma \neq 1} p_{M,x,k,\sigma}$. If for all $x \in \Sigma^*$ there exists an N such that $\sum_{k \leq N} \sum_\sigma p_{M,x,k,\sigma} = 1$, i.e. M halts with certainty after some finite number of steps, then we say that M *halts absolutely*[2].

Space-bounds and quantum space-bounded 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 bit 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 bit 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

[1] If we are interested in QTMs which output strings, we may define the output of a computation of M on x to be, say, the longest string in $(\Sigma \backslash \{\#\})^*$ observed during the computation (i.e. the computation ends when it is observed that the machine writes a blank to the output tape, and otherwise the machine does not halt). We will restrict our attention in this paper to machines for language membership, however.

[2] Unlike the probabilistic case, if a QTM M halts absolutely it may not be the case that M halts along every nonzero amplitude computation path, since interference may cause such paths may cancel each other out.

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 nondecreasing and 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 $|x|$.

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

Definition 2.1 For each $X \in \{\mathrm{EQ}, \mathrm{RQ}, \mathrm{BQ}, \mathrm{NQ}, \mathrm{PrQ}\}$, a given language L is said to be in the class $X\mathrm{SPACE}(s)$ if there exists a QTM M 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 $X_H\mathrm{SPACE}(s)$.

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 Relationships between quantum classes

In this section, we discuss relationships amongst 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, each of the halting classes is contained in its corresponding non-halting class, i.e. $X_H\mathrm{SPACE}(s) \subseteq X\mathrm{SPACE}(s)$ for $X \in \{\mathrm{EQ}, \mathrm{RQ}, BQ, NQ, PrQ\}$. The following containments also follow immediately from the definitions:

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

and $\mathrm{RQSPACE}(s) \subseteq \mathrm{NQSPACE}(s)$, 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.*

(1) After precisely $t'(|x|)$ steps, M' accepts with probability $p_{acc}(x)$ and rejects with probability $\alpha\, p_{rej}(x)$, for each input x.

(2) After precisely $t'(|x|) + 1$ steps, M' accepts with probability β and rejects with probability $1 - \beta$ (thus halting absolutely), resulting in 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.

Informally, the 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. 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 this lemma 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 *$NQ_HSPACE(s) = NQSPACE(s)$.*

Proof. For the nontrivial direction, 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 $t(|x|) = 2^{s(|x|)}$ and $\beta = 0$. ∎

It is known that $\mathrm{NSPACE}(s) = \mathrm{RSPACE}(s)$, 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.

Proposition 3.4 *$NQSPACE(s) \subseteq EQSPACE(s)$.*

Corollary 3.5 *$EQSPACE(s) = RQSPACE(s) = NQSPACE(s)$.*

It will be demonstrated in the two sections which follow that the class NQSPACE(s) corresponds to the counting class co-C$_=$SPACE(s), defined analogously to co-C$_=$L for $s(n) = \log n$ (see [2], and also see Section A.4 in the appendix for an equivalent definition.) It is not known whether C$_=$SPACE(s), and hence NQSPACE(s), is closed under complementation.

4 Quantum simulations of classical classes

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 will 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 $\left(2^{-2st} p_{acc}\right)^2$ and rejects x with probability $\left(2^{-2st} p_{rej}\right)^2$ 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 indicated 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 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 $p_{acc}(x) > p_{rej}(x)$.

Proposition 4.2 *$PrSPACE(s) \subseteq PrQ_HSPACE(s)$.*

(Here we rely 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 classes

In this section, we discuss probabilistic simulations of space-bounded quantum machines.

In [1] it is shown that unbounded error, polynomial time probabilistic machines are capable of simulating polynomial time quantum machines. In the context of space-bounded classes, it is possible to modify this simulation to obtain the result PrQ$_H$SPACE(s) $\subseteq$ PrSPACE(s). However, we are able to obtain the somewhat stronger result PrQSPACE(s) $\subseteq$ PrSPACE(s) by adopting techniques which have been used to prove Pr$_H$SPACE(s) = PrSPACE(s) (see [2, 20]).

Lemma 5.1 *Let M be a QTM running in space s. Then for each input x there exist $2^{2s+1} \times 2^{2s+1}$ matrices A and B, where entries of A and B are l bit integers with l computable in space $O(s)$, such that the following properties are satisfied.*

(1) For each $i, j \leq 2^{2s+1}$ and $k \leq l$, the kth bit of the i,j entries of A and B are computable in space $O(s)$.

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

It is known that the problem of deciding whether or not one given integer matrix has a larger determinant than a second given matrix can be solved by an unbounded error PTM running in space $O(\log n)$ ([2], see also [11, 28, 29, 30]). From this it follows that for A and B as above, there exists a space $O(s)$ PTM which accepts with probability greater than 1/2 if and only if $\det(A) > \det(B)$, and hence if and only if the given QTM M halts with probability greater than 1/2. Thus, we have the following proposition.

Proposition 5.2 *$PrQSPACE(s) \subseteq PrSPACE(s)$.*

By Propositions 4.2 and 5.2, we have

Corollary 5.3 *$PrQ_H SPACE(s) = PrQSPACE(s) = PrSPACE(s)$.*

Corollary 5.4 *$PrQSPACE(s) \subseteq DSPACE(s^2)$.*

Finally, we note the following relationship.

Proposition 5.5 *$NQSPACE(s) \subseteq$ co-$C_=SPACE(s)$.*

Proof. It follows from the proof of Lemma 5.1 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 5.5, we have

Corollary 5.6 *$NQSPACE(s) =$ co-$C_=SPACE(s)$.*

This may be viewed as the space-bounded analogue of the result QNP = co-C$_=$P [14].

6 Conclusion and open problems

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

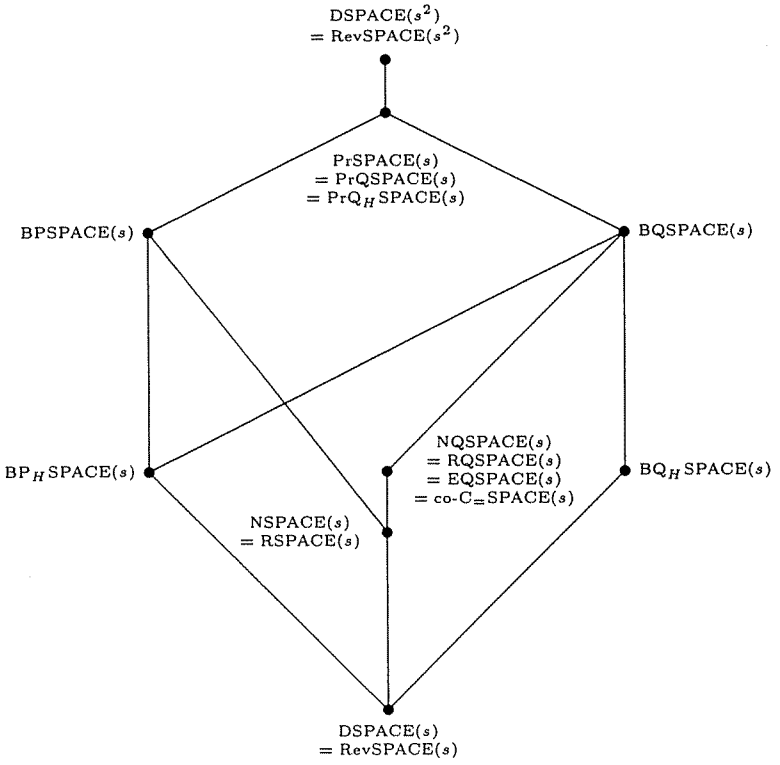

Figure 1: Relationships between space-bounded quantum and probabilistic classes.

A number of questions have been left open by this paper. In particular, can a probabilistic machine which halts absolutely be simulated efficiently by a quantum machine which also halts absolutely for the case of bounded error? For example, 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? Also, can probabilistic simulations of space-bounded quantum machines be performed in such a way that bounded error probability is achieved? For example, are any of the quantum classes $\mathrm{EQ}_H\mathrm{SPACE}(s)$, $\mathrm{RQ}_H\mathrm{SPACE}(s)$, $\mathrm{BQ}_H\mathrm{SPACE}(s)$, or $\mathrm{NQSPACE}(s)$ contained in $\mathrm{BPSPACE}(s)$, say?

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 (see [23]). Are there natural quantum analogues of these classes, and how do these classes relate to those discussed in this paper?

We have restricted our attention in this paper to space-bounds which are at least logarithmic in the input size. In the case of constant space-bounds, polynomial time QTMs are more powerful than polynomial time PTMs [21]. What else 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 M, 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.

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. For the initial state of M', it is assumed that $q = q_0$ (the initial state of M), and $\tau = \#$. 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.

We will assume that integers are efficiently encoded as strings over $\{0, 1\}$ in such a way that (1) the empty string represents 0, and (2) the integers and representations are in one-to-one correspondence. (Such an encoding appears in [15]: letting $0x$ represent the positive integer with binary representation $1x$ and letting $1x$ represent the negative integer having absolute value with binary representation $1x$, we have a suitable encoding.)

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.

First we note that the loop (step 1) can be performed reversibly. Let us suppose that the body of the loop corresponds to some sequence of actions beginning with states of the form (q, τ, r_1) and ending with states of the form (q, τ, r_1'), and that step 2 begins with states of the form (q, τ, r_2). Since track 5 contains 0 if and only if square 0 on track 5 contain a blank (given our encoding of integers as described previously), the transition function of M' can induce the following reversible transformation on its internal state in order to perform the loop as required.

$$(q, \tau, r_0) \mapsto \begin{cases} (q, \tau, r_1) & \text{symbol being read on track 5 is } \# \\ (q, \tau, r_2) & \text{symbol being read on track 5 is not } \# \end{cases}$$

$$(q, \tau, r_1') \mapsto \begin{cases} (q, \tau, r_1) & \text{symbol being read on track 5 is not } \# \\ (q, \tau, r_2) & \text{symbol being read on track 5 is } \# \end{cases}$$

Next, note that the coin-flip in step 3 could easily be simulated by performing the Hadamard transform on some initially 0 bit (in the internal state of M', say). Since we require that all amplitudes of our machines be rational, however, we may use the 4-dimensional Hadamard transform H_4 instead:

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

where (a, b) denotes the number of 1's in the bitwise-and of a and b written in binary.

Each remaining step described in Figure 2 is clearly reversible (save the remaining "quantum" step which, once σ has been located on the input tape, is performed in a single step and involves

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

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

      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.

      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.

      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.

      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.

   ii. 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).

   iii. If the numbers on tracks 4 and 5 are equal, 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).

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

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.

only the internal memory of M'). Furthermore, we claim that each step can be performed reversibly using $O(s)$ space and in such a way that the amount of time required for each step is independent of whatever configuration of M is represented by M' at that time. (This is routine to show for each step, given that M runs in space s.)

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 probabilities of acceptance and rejection are as claimed. The number of steps t' required for M' to complete step 2 is readily seen to be computable in space $O(s)$. ∎

A.2 Proof of Lemma 3.3

M' will simulate M for t steps in a manner similar to the machine contructed 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 would constitute an irreversible action. 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 of the same form as in that proof. The execution of M' is described in Figure 3.

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

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

   whenever $k \in \{-s, \ldots, s\}$ (and $k \mapsto k$ otherwise).

   ii. – 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 or track 3 does not contain all blanks), then multiply the current amplitude by -1.
5. Goto step 1.

Figure 3: Description of M' for Lemma 3.3.

Under the assumption that M runs in space s, the work tape head of M never leaves the region $\{-s, \ldots, s\}$ when started from the initial configuration. However, there is no guarantee that the same is true when the simulation is inverted in step 3. (This is because output may have occurred in step 2, possibly resulting in space-expensive paths being followed with nonzero amplitude, even if these paths had zero amplitude in the forward direction.) However, because we substitute the transformation

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

for adding $D_w(q)$ to k (the position of the work tape head of M stored on track 2 of M'), we guarantee that the inverse simulation in step 3 can always be performed in space $O(s)$, as $k \in \{-s, \ldots, s\}$ is an invariant throughout the simulation (step 3 may not correspond to simulating t steps of a QTM for this reason, but this is irrelevant). It follows that the entire simulation can be performed in space $O(s)$, as only space $O(s)$ configurations of M are reached. Of course this transformation is the same as adding $D_w(q)$ to k in the forward direction of the simulation, and so the correct probabilities of acceptance and rejection are preserved.

Now we will show that the claimed probabilities of acceptance and rejection are obtained. Let c'_0 denote the initial configuration of M', and 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 that $\| |\psi_1\rangle \|^2 = p_{acc}$ and $\| |\psi_0\rangle \|^2 = p_{rej}$. 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 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

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

The probability that M' accepts may now be calculated as

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

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

Note that we did not renormalize superpositions after each step in performing the above calculation; this is a shortcut for calculating unconditional probabilities. For example, if we renormalized $|\psi_\varepsilon\rangle$ after the first time step 3 was performed, we could have obtained the probabilities with which M' accepts and rejects on the second iteration of the loop, given that no output was observed during the first iteration. Adjusting these answers to find the unconditional probabilities with which M' accepts and rejects on the second iteration of the loop would have been equivalent to not renormalizing $|\psi_\varepsilon\rangle$ in the first place.

A.3 Proof of Lemma 4.1

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 will be described below, and track 4 will record the number of steps for which M has been simulated. The tape symbols which will be used on tracks 1 and 2 will be elements of the set $\{\#, 0, 1, 2, 3\}$. 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.

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. If track 3 contains 0, perform H_4 on each digit on track 2. If, in addition, track 1 encodes a configuration with 2 successors, perform H_4 on a.
   ii. 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.
   iii. Swap the contents of tracks 1 and 2.
   iv. Perform H_4 on each digit on track 2. If track 2 contains any nonzero digit, increment the number on track 3.
   v. Increment the number on track 4 modulo $t(|x|)$.
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: Quantum simulation of a PTM

Clearly the contents of each track of M' has length $O(s)$, and further each step can be performed by a QTM within $O(s)$ space. (Note that space constructibility of s, along with [22], implies that M' can compute $s(|x|)$, mark off $s(|x|)$ squares, etc., in space $O(s)$.) Again we assume that the time required for each step does not depend on the particular configurations of M represented by M' at that particular step. Thus, there exists t', as in the statement of the lemma, which is the number of steps required for M' to complete step 3 along all computation paths.

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 configuration with c replaced by c' (and the number on track 4 incremented) is 2^{-2s} (each of $2s$ bits is mapped to the correct bit with amplitude $1/2$). 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

The class C$_=$SPACE(s) may be defined in an analogous way to the class C$_=$L, defined in [2]. However, we will use the following equivalent definition.

Definition A.1 The class 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 $\frac{1}{2}$.

See [2] for further discussion on this class.

Proof of Proposition 4.4. Let $L \in$ 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.

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.

We will 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. There are only 2 good configurations which

M' can be in after performing step 4: one in which $a = 1$ and the other in which $a = 0$ (and all other aspects of these two configurations are 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 $p_{acc} \neq p_{rej}$. Hence M' accepts with nonzero probability if and only if $x \notin L$. ∎

A.5 Proof of Lemma 5.1

For given input x, define a $2^s \times 2^s$ matrix D, indexed by the set $\{0, \ldots, 2^s - 1\}$, as follows.

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

(where we identify numbers in $\{0, \ldots, 2^s - 1\}$ and their binary representations as length s bit strings). Next, define a $(2^{2s} + 1) \times (2^{2s} + 1)$ matrix E, indexed by the set $\{0, \ldots, 2^{2s}\}$, as follows. For each $i, j < 2^{2s}$, write $i = i_0 + i_1 2^s$ and $j = j_0 + j_1 2^s$, where $i_0, i_1, j_0, j_1 \in \{0, \ldots, 2^s - 1\}$, and let

$$E[i,j] = D[i_0, j_0]\, D[i_1, j_1].$$

Also let

$$E[2^{2s}, j] = \begin{cases} 1 & j = j_0 + j_0 2^s,\ j_0 \text{ encoding an accepting configuration of } M \\ 0 & \text{otherwise,} \end{cases}$$

and let $E[i, 2^{2s}] = 0$ for every i. Taking y_{acc}^T be the vector consisting of the first 2^{2s} entries of the last row of E, we see that the matrix E has the form

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

where $D \otimes D$ denotes the Kronecker product of D with itself.

Lemma A.1 *For any space s bounded QTM M and input x, let E be as defined above. Let $j_{init} = j_0 + j_0 2^s$, where j_0 encodes the initial configuration of M. Then for each $k \in \mathbb{N}$, the probability that M accepts x on the kth step is $E^{k+1}[2^{2s}, j_{init}]$.*

Proof. Let y_{init} be the 2^{2s} dimensional vector indexed by $\{0, \ldots, 2^{2s} - 1\}$ with $y_{init}[j_{init}] = 1$ and all other entries zero. We see that

$$E^{k+1}[2^{2s}, j_{init}] = y_{acc}^T \left(D \otimes D\right)^k y_{init} = y_{acc}^T \left(D^k \otimes D^k\right) y_{init} = \left\| P_1 (U_x P_\varepsilon)^k \,|c_0\rangle \right\|^2.$$

Under the assumption that $(U_x P_\varepsilon)^k \,|c_0\rangle \neq 0$, the conditional probability that M accepts on the kth step, given that M has not previously halted, is

$$\frac{\left\| P_1 (U_x P_\varepsilon)^k \,|c_0\rangle \right\|^2}{\left\| (U_x P_\varepsilon)^k \,|c_0\rangle \right\|^2}$$

(and otherwise the probability is zero). It is simple to show by induction that the probability that M does not halt prior to the kth step is $\left\| (P_\varepsilon U_x)^{k-1} \,|c_0\rangle \right\|^2 = \left\| (U_x P_\varepsilon)^k \,|c_0\rangle \right\|^2$, and hence we have that $E^{k+1}\left[2^{2s}, j_{init}\right]$ is the (unconditional) probability that M accepts x on the kth step. ∎

Lemma A.2 *Let M be a QTM running in space s. Then there exists a polynomial f such that, for any input x, (1) $\left(I - \left(1 - 2^{-f(2^s)}\right) E\right)$ is invertible, and (2) M accepts x with probability exceeding $1/2$ if and only if*

$$\left(I - \left(1 - 2^{-f(2^s)}\right) E\right)^{-1} [2^{2s}, j_{init}] > \frac{1}{2},$$

where E is as defined above.

In order to prove Lemma A.2 we will first prove the following simple lemmas regarding bounds for integer polynomials and rational functions.

Lemma A.3 *Let $p \in \mathbb{Z}[X]$ where $deg(p) \leq N$ and the coefficients of p are bounded in absolute value by L. Then for $0 < \epsilon \leq 1$, we have*

$$|p(1) - p(1-\epsilon)| \leq \epsilon L \binom{N+1}{2}.$$

Proof. Write $p(X) = \sum_{j=0}^{N} a_j X^j$. Then

$$|p(1) - p(1-\epsilon)| = \left|\sum_{j=1}^{N} a_j \sum_{i=1}^{j} \binom{j}{i} (-\epsilon)^i\right| \leq \epsilon L \sum_{j=1}^{N} \left|\sum_{i=1}^{j} \binom{j}{i} (-\epsilon)^{i-1}\right| \leq \epsilon L \binom{N+1}{2}.$$

■

Lemma A.4 *Let $p, q \in \mathbb{Z}[X]$ where $deg(p), deg(q) \leq N$, coefficients of p and q are bounded in absolute value by L, and $q(1) \neq 0$. Then for $0 < \epsilon < \frac{1}{2L(N+1)^2}$, we have*

$$\left|\frac{p(1)}{q(1)} - \frac{p(1-\epsilon)}{q(1-\epsilon)}\right| < 4\epsilon L^2 (N+1)^3.$$

Proof. Note that $|q(1-\epsilon)| > 1/2$ follows from the assumption $0 < \epsilon < \frac{1}{2L(N+1)^2}$, along with Lemma A.3. We have

$$\begin{aligned}
\left|\frac{p(1)}{q(1)} - \frac{p(1-\epsilon)}{q(1-\epsilon)}\right| &= \left|\frac{p(1)q(1-\epsilon) - p(1-\epsilon)q(1)}{q(1)q(1-\epsilon)}\right| \\
&\leq \left|\frac{p(1)q(1-\epsilon) - p(1)q(1)}{q(1)q(1-\epsilon)}\right| + \left|\frac{p(1)q(1) - p(1-\epsilon)q(1)}{q(1)q(1-\epsilon)}\right| \\
&< 4\epsilon L^2 (N+1)^3.
\end{aligned}$$

■

Proof of Lemma A.2. 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 eigenvalue of E must also be an eigenvalue of $D \otimes D$). This implies that $(I - z\,E)$ is invertible for $0 < z < 1$, and that we may write

$$(I - z\,E)^{-1} = \sum_{k \geq 0} z^k E^k.$$

Now, it follows from Lemma A.1 that we have $0 \leq (I - z\,E)^{-1}[2^{2s}, j_{init}] \leq p_{acc}$, where p_{acc} is the probability that M accepts input x. Furthermore $\lim_{z \uparrow 1} (I - z\,E)^{-1}[2^{2s}, j_{init}]$ exists and is equal to p_{acc}. Thus, it remains to show that if

$$\lim_{z \uparrow 1} (I - z\,E)^{-1}[2^{2s}, j_{init}] > \frac{1}{2},$$

then

$$\left(I - \left(1 - 2^{-f(2^s)}\right) E\right)^{-1} [2^{2s}, j_{init}] > \frac{1}{2}$$

for some polynomial f (which is independent of x).

For the remainder of the proof let $N = 2^{2s} + 1$, let d denote the least common denominator of the entries in E, and for brevity write $i = 2^{2s}$, $j = j_{init}$. We have

$$(I - z\,E)^{-1}[i,j] = (-1)^{i+j} \frac{\det((I - z\,E)_{ji})}{\det(I - z\,E)} = (-1)^{i+j} d \frac{\det((d\,I - z\,d\,E)_{ji})}{\det(d\,I - z\,d\,E)} = \frac{u(z)}{v(z)},$$

where $u, v \in \mathbb{Z}[z]$ may be taken to have degree at most N and coefficients which are bounded in absolute value by $N!(2d)^N$. (Here, $(d\,I - z\,d\,E)_{ji}$ denotes the $(N-1) \times (N-1)$ matrix obtained by removing row j and column i from $(d\,I - z\,d\,E)$.) Under the assumption that $\lim_{z \uparrow 1} (I - zE)^{-1}[i,j]$ exists, we may write

$$\frac{u(z)}{v(z)} = \frac{(z-1)^k\, u_0(z)}{(z-1)^k\, v_0(z)},$$

where $(z-1) \nmid v_0(z)$, so that $(I - zE)^{-1}[i,j] = \frac{u_0(z)}{v_0(z)}$ for $0 < z < 1$ and $\lim_{z \uparrow 1} (I - zE)^{-1}[i,j] = \frac{u_0(1)}{v_0(1)}$. The coefficients of u_0 and v_0 may be bounded in absolute value by $N^k N!(2d)^N < (d\,N)^{2N}$. It follows that if $\lim_{z \uparrow 1} (I - zE)^{-1}[i,j] > \frac{1}{2}$, then we must have

$$\lim_{z \uparrow 1} (I - zE)^{-1}[i,j] - \frac{1}{2} = \frac{u_0(1)}{v_0(1)} - \frac{1}{2} \geq \frac{1}{2\,v_0(1)} \geq \frac{1}{2(N+1)(d\,N)^{2N}}.$$

Choose $\epsilon < \frac{1}{8\,(N+1)^4 (dN)^{6N}}$. By Lemma A.4 we have

$$\left| \frac{u_0(1)}{v_0(1)} - \frac{u_0(1-\epsilon)}{v_0(1-\epsilon)} \right| \leq 4\,\epsilon\,(dN)^{4N} (N+1)^3 < \frac{1}{2(N+1)(dN)^{2N}},$$

from which it follows that $(I - (1-\epsilon)E)^{-1}[i,j] > \frac{1}{2}$. Now f may be chosen appropriately. ∎

Now, we are almost ready to define the matrices A and B for the main lemma. First, let F and G be $(2^{2s}+1) \times (2^{2s}+1)$ integer matrices, indexed by the set $\{0, \ldots, 2^{2s}\}$, defined as follows.

$$G = d\,2^{f(2^s)}\,I - (2^{f(2^s)} - 1)(dE),$$

where d denotes the least common denominator of the entries in E, and

$$F = \begin{bmatrix} G_{j_{init},2^{2s}} & 0 \\ 0 & 2\,(-1)^{j_{init}}\, d\, 2^{f(2^s)} \end{bmatrix}$$

where $G_{j_{init},2^{2s}}$ denotes G with row j_{init} and column 2^{2s} removed. We now have

$$\begin{aligned} \frac{\det(F)}{\det(G)} &= 2\,(-1)^{j_{init}}\, d\, 2^{f(2^s)} \frac{\det(G_{j_{init},2^{2s}})}{\det(G)} \\ &= 2\,(-1)^{j_{init}} \frac{\det\left(\left(I - \left(1 - 2^{-f(2^s)}\right) E\right)_{j_{init},2^{2s}}\right)}{\det\left(I - \left(1 - 2^{-f(2^s)}\right) E\right)} \\ &= 2\left(I - \left(1 - 2^{-f(2^s)}\right) E\right)^{-1} \left[2^{2s}, j_{init}\right]. \end{aligned}$$

Hence, by Lemma A.2, $\frac{\det(F)}{\det(G)} > 1$ 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

$$A = \begin{bmatrix} F_{0,0} & 0 \\ 0 & F_{0,0} \end{bmatrix}, \quad B = \begin{bmatrix} G_{0,0} & 0 \\ 0 & G_{0,0} \end{bmatrix},$$

$F_{0,0}$ and $G_{0,0}$ being F and G with row and column 0 removed, respectively (recall that all encodings of configurations begin with 1, so removing row and column 0 from F and G will not affect the ratio $\frac{\det(F)}{\det(G)}$). 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 is $2^{O(s)}$, and an appropriate bound l on this length can be computed in space $O(s)$. 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 $O(s)$. This completes the proof. ∎

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. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17:525–532, 1973.

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

[5] C. 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. Preliminary version appeared in *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 11-20, 1993.

[7] A. Berthiaume. Quantum computation. In L. Hemaspaandra and A. Selman, editors, *Complexity Theory Retrospective II*, pages 23–50. Springer, 1997.

[8] 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.

[9] 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.

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

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

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

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

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

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

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

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

[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 Colloquiun on Automata, Languages and Programming*, volume 172 of *Lecture Notes in Computer Science*, pages 281–291. Springer-Verlag, 1984.

[20] H. Jung. On probabilistic time and space. In *Proceedings of the 12th International Colloquiun on Automata, Languages and Programming*, volume 194 of *Lecture Notes in Computer Science*, pages 310–317. Springer-Verlag, 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 (extended abstract). In *Proceedings of the 12th 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. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 116–123, 1994.

[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 M. S. Paterson, editor, *Boolean Function Complexity*, volume 169 of *London Mathematical Society Lecture Notes Series*, pages 84–94. Cambridge University Press, 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.