Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 73 (1998)

$\mathbf{NQP}_{\mathbb{C}} = \mathbf{co\text{-}C_{=}P}$

Tomoyuki Yamakami* and Andrew C. Yao†

Department of Computer Science, Princeton University

Princeton, NJ 08544

Abstract

Adleman, DeMarrais, and Huang introduced the nondeterministic quantum polynomial-time complexity class **NQP** as an analogue of **NP**. Fortnow and Rogers implicitly showed that, when the amplitudes are rational numbers, **NQP** is contained in the complement of $\mathbf{C_{=}P}$. Fenner, Green, Homer, and Pruim improved this result by showing that, when the amplitudes are arbitrary algebraic numbers, **NQP** coincides with co-$\mathbf{C_{=}P}$. In this paper we prove that, even when the amplitudes are arbitrary complex numbers, **NQP** still remains identical to co-$\mathbf{C_{=}P}$. As an immediate corollary, **BQP** differs from **NQP** when the amplitudes are unrestricted.

key words: computational complexity, theory of computation

1 Introduction

In recent years, the possible use of the power of quantum interference and entanglement to perform computations much faster than classical computers has attracted attention from computer scientists and physicists (e.g., [6, 9, 13, 16, 18, 19]).

In 1985 Deutsch [7] proposed the fundamental concept of *quantum Turing machines* (see Bernstein and Vazirani [4] for further discussions). A quantum Turing machine is an extension of a classical probabilistic Turing machine so that all computation paths of the machine interfere with each other (similar to the phenomenon in physics known as *quantum interference*). This makes it potentially possible to carry out a large number of bit operations simultaneously. Subsequent studies have founded the structural analysis of quantum complexity classes. In particular, quantum versus classical counting computation has been a focal point in recent studies [1, 11, 14].

Adleman, DeMarrais, and Huang [1] introduced, as a quantum analogue of **NP**, the "nondeterministic" quantum polynomial-time complexity class $\mathbf{NQP}_K$, which is the set of decision problems accepted with positive probability by polynomial-time quantum Turing machines with amplitudes all drawn from set K. In their paper, they argued that $\mathbf{NQP}_{\mathbb{A}\cap\mathbb{R}}$ lies within **PP**, where $\mathbb{A}$ denotes the set of algebraic complex numbers.

*This work was supported by NSERC Postdoctoral Fellowship and DIMACS Fellowship.

†This work was supported in part by National Science Foundation and DARPA under grant CCR-9627819.

ISSN 1433-8092

In classical complexity theory, Wagner [22] defined the counting class $\mathbf{C}_{=}\mathbf{P}$ as the set of decision problems that determine whether the number of accepting computation paths (of nondeterministic computation) equals that of rejecting computation paths. Fortnow and Rogers [14] implicitly showed that $\mathbf{NQP}_{\mathbb{Q}} \subseteq \text{co-}\mathbf{C}_{=}\mathbf{P}$; in fact, as pointed out in [11], their proof technique proves $\mathbf{NQP}_{K} \subseteq \text{co-}\mathbf{C}_{=}\mathbf{P}$ so long as all members of K are products of rational numbers and the square root of natural numbers. Fenner, Green, Homer, and Pruim [11] further improved this result by showing $\mathbf{NQP}_{\mathbb{A}} = \text{co-}\mathbf{C}_{=}\mathbf{P}$, which gives a characterization of $\mathbf{NQP}$ in terms of classical counting computation when the amplitudes are restricted to algebraic numbers (in [11] $\mathbf{NQP}_{\mathbb{A}}$ is succinctly denoted as $\mathbf{NQP}$). Nevertheless, it has been unknown whether $\mathbf{NQP}_{\mathbb{C}}$ further collapses to $\text{co-}\mathbf{C}_{=}\mathbf{P}$.

In this paper we resolve this open question affirmatively as in Theorem 3.5: $\mathbf{NQP}_{K}$ collapses to $\text{co-}\mathbf{C}_{=}\mathbf{P}$ for any set K with $\mathbb{Q} \subseteq K \subseteq \mathbb{C}$. The proof of the theorem consists of two steps. First we must show that $\text{co-}\mathbf{C}_{=}\mathbf{P} \subseteq \mathbf{NQP}_{\mathbb{Q}}$ (actually $\text{co-}\mathbf{C}_{=}\mathbf{P} \subseteq \mathbf{NQP}_{\{0,\pm\frac{3}{5},\pm\frac{4}{5},\pm 1\}}$). This claim was already mentioned in [11] and its proof recently appeared in [12]. For completeness, we give its proof in Section 3. Then we must prove the claim $\mathbf{NQP}_{\mathbb{C}} \subseteq \text{co-}\mathbf{C}_{=}\mathbf{P}$ in Section 4 by a detailed algebraic analysis of transition amplitudes of quantum Turing machines.

Our result yields another important consequence about the relationship between $\mathbf{NQP}_{K}$ and $\mathbf{BQP}_{K}$, a quantum analogue of $\mathbf{BPP}$, which was introduced by Bernstein and Vazirani [4] as the set of decision problems recognized by polynomial-time quantum Turing machines with bounded-error with amplitudes from K. It is shown in [1] that $\mathbf{BQP}_{\mathbb{Q}} = \mathbf{BQP}_{\mathbb{A}}$ but $\mathbf{BQP}_{\mathbb{C}}$ has uncountable cardinality. Theorem 3.5 thus highlights a clear contrast between the power of nondeterministic quantum computation and that of bounded-error quantum computation: $\mathbf{BQP}_{\mathbb{C}} \neq \mathbf{NQP}_{\mathbb{C}}$. This extends the separation of the exact quantum computation from bounded-error quantum computation in the case when amplitudes are unrestricted [1].

The reader who needs more background on quantum computation may refer to recent survey papers [2, 5].

2 Basic Notions and Notation

Let $\mathbb{Z}$ be the set of all integers, $\mathbb{Q}$ the set of rational numbers, $\mathbb{C}$ the set of complex numbers, and $\mathbb{A}$ the set of all algebraic complex numbers. Moreover, let $\mathbb{Z}_{\geq 0}$ and $\mathbb{Z}_{>0}$ denote the sets of all nonnegative integers and of all positive integers, respectively. For any $d \in \mathbb{Z}_{>0}$ and $k \in \mathbb{Z}_{\geq 0}$, define $\mathbb{Z}_d = \{i \in \mathbb{Z} \mid 0 \leq i \leq d-1\}$ and $\mathbb{Z}_{[k]} = \{i \in \mathbb{Z} \mid -k \leq i \leq k\}$. By *polynomials* we mean elements in $\mathbb{Z}[x_1, x_2, \ldots, x_m]$ for some $m \in \mathbb{Z}_{\geq 0}$. For any finite sequence $\boldsymbol{k} \in \mathbb{Z}^m$, let $|\boldsymbol{k}| = \max_{1 \leq i \leq m}\{|k_i|\}$, where $\boldsymbol{k} = (k_1, k_2, \ldots, k_m)$. Furthermore, $\mathbf{0}^k$ denotes the k-tuple $(0, 0, \ldots, 0)$ for $k \in \mathbb{Z}_{>0}$.

Let $k \in \mathbb{Z}_{>0}$. A finite subset $\{\gamma_i\}_{1 \leq i \leq k}$ of $\mathbb{C}$ is *linearly independent* if $\sum_{i=1}^{k} a_i \gamma_i \neq 0$ for any k-tuple $(a_1, a_2, \ldots, a_k) \in \mathbb{Q}^k \setminus \{\mathbf{0}^k\}$. Similarly, $\{\gamma_i\}_{1 \leq i \leq k}$ is *algebraically independent* if there is no

q in $\mathbb{Q}[x_1, x_2, \ldots, x_k]$ such that q is not identical to 0 but $q(\gamma_1, \gamma_2, \ldots, \gamma_k) = 0$.

We assume the reader's familiarity with classical complexity theory and here we give only a brief description of quantum Turing machines [4]. A k-track *quantum Turing machine* (QTM) M is a triplet (Σ^k, Q, δ), where Σ is a finite alphabet with a distinguished blank symbol #, Q is a finite set of states with initial state q_0 and final state q_f, and δ is a multi-valued *quantum transition function* from $Q \times \Sigma^k$ to $\mathbb{C}^{Q\times\Sigma^k\times\{L,R\}}$. A QTM has k two-way infinite tracks of cells indexed by $\mathbb{Z}$ and a read/write head that moves along all the tracks. The expression $\delta(p, \boldsymbol{\sigma}, q, \boldsymbol{\tau}, d)$ denotes the (transition) amplitude in $\delta(p, \boldsymbol{\sigma})$ of $|q\rangle|\boldsymbol{\tau}\rangle|d\rangle$, where $\boldsymbol{\sigma}, \boldsymbol{\tau} \in \Sigma^k$ and $d \in \{L, R\}$.

A *superposition* of M is a finite complex linear combination of configurations of M with unit L_2-norm. The *time-evolution operator* of M is a map from each superposition of M to the superposition of M that results from a single application of the transition function δ. These time-evolution operators are naturally identified with matrices.

The *running time* of M on input x is defined to be the minimum integer T such that, at time T, all computation paths of M on input x have reached final configurations, and at any time less than T there are no final configurations, where a *final configuration* is a configuration with state q_f. We say that M on input x *halts in time* T if the running time of M on input x is T. The *final superposition* of M is the superposition that M reaches when it halts. A QTM M is called a *polynomial-time QTM* if there exists a polynomial p such that, on every input x, M halts in time $p(|x|)$.

A QTM is called *well-formed* if its time-evolution operator preserves the L_2-norm. A QTM is *stationary* if it halts on all inputs in a final superposition where each configuration has the head in the start cells and a QTM is in *normal form* if, for every k-tuple of track symbols $\boldsymbol{\sigma}$, $\delta(q_f, \boldsymbol{\sigma}) = |q_0\rangle|\boldsymbol{\sigma}\rangle|R\rangle$. For brevity, we say that a QTM is *conservative* if it is well-formed and stationary and in normal form. For any subset K of $\mathbb{C}$, we say that a QTM has K-*amplitudes* if its transition amplitudes are all drawn from K.

Let M be a multitrack, well-formed QTM whose last track, called the *output* track, has alphabet $\{0, 1, \#\}$. We say that M *accepts* x *with probability* p and also *rejects* x *with probability* $1 - p$ if M halts and p is the sum of the squared magnitudes of the amplitude of each final configuration in which the output track consists only of 1 as nonblank symbol in the start cell. For convenience, we call such a final configuration an *accepting configuration.*

For a more general model of quantum Turing machines, the reader may refer to [23].

3 Main Result

In this section we state the main theorem of this paper. First we give the formal definitions of the complexity classes $\mathbf{C_{=}P}$ [22] and $\mathbf{NQP}$ [1].

The counting class $\mathbf{C_{=}P}$ was originally introduced by Wagner [22]. For convenience, we begin with the definition of **GapP**-functions. For a nondeterministic Turing machine M, $Acc_M(x)$ denotes the number of accepting computation paths of M on input x. Similarly, we denote by $Rej_M(x)$ the number of rejecting computation paths of M on input x.

Definition 3.1 **[10]** A function from Σ^* to $\mathbb{Z}$ is in **GapP** if there exists a polynomial-time nondeterministic Turing machine M such that $f(x) = Acc_M(x) - Rej_M(x)$ for every string x.

Lemma 3.2 **[10]** *Let $f \in$ **GapP** and p a polynomial. Then, the following functions are also **GapP**-functions: f^2, $\lambda x. \sum_{y \in \Sigma^{p(|x|)}} f(x,y)$, and $\lambda x. \prod_{i=1}^{p(|x|)} f(x,1^i)$, where $f^2(x)$ means $(f(x))^2$ and the λ-notation $\lambda x.g(x)$ means the function g.*

Definition 3.3 **[22]** A set S is in $\mathbf{C_{=}P}$ if there exists a **GapP**-function f such that, for every x, $x \in S$ exactly when $f(x) = 0$.

Adleman, DeMarrais, and Huang [1] introduced the notion of "nondeterministic" quantum computation and defined the complexity class $\mathbf{NQP}_K$ as the collection of all sets that can be recognized by nondeterministic QTMs with K-amplitudes in polynomial time.

Definition 3.4 **[1]** Let K be a subset of $\mathbb{C}$. A set S is in $\mathbf{NQP}_K$ if there exists a polynomial-time, conservative QTM M with K-amplitudes such that, for every x, if $x \in S$ then M accepts x with positive probability and if $x \notin S$ then M rejects x with probability 1.

It immediately follows from Definition 3.4 that $\mathbf{NP} \subseteq \mathbf{NQP}_{\mathbb{Q}} \subseteq \mathbf{NQP}_{\mathbb{A}} \subseteq \mathbf{NQP}_{\mathbb{C}}$. Adleman et al.[1] first showed that $\mathbf{NQP}_{\mathbb{A}\cap\mathbb{R}}$ is a subset of **PP**. Based on the work of Fortnow and Rogers [14], Fenner, Green, Homer, and Pruim [11] later obtained the significant improvement: $\mathbf{NQP}_K =$ co-$\mathbf{C_{=}P}$ for any set K satisfying $\mathbb{Q} \subseteq K \subseteq \mathbb{A}$.

We expand their result and show as the main theorem that any class $\mathbf{NQP}_K$, $\mathbb{Q} \subseteq K \subseteq \mathbb{C}$, collapses to co-$\mathbf{C_{=}P}$.

Theorem 3.5 *For any set K with $\mathbb{Q} \subseteq K \subseteq \mathbb{C}$, $\mathbf{NQP}_K =$ co-$\mathbf{C_{=}P}$.*

Before giving the proof of Theorem 3.5, we state its immediate corollary. We need the notion of bounded-error quantum polynomial-time complexity class $\mathbf{BQP}_K$ given by Bernstein and Vazirani [4].

Definition 3.6 **[4]** A set S is in $\mathbf{BQP}_K$ if there exists a polynomial-time, conservative QTM M with K-amplitudes such that, for every x, if $x \in S$ then M accepts x with probability at least $\frac{2}{3}$ and if $x \notin S$ then M rejects x with probability at least $\frac{2}{3}$.

It is known from [1] that $\mathbf{BQP}_{\mathbb{C}}$ has uncountable cardinality. Theorem 3.5 thus implies that $\mathbf{BQP}_{\mathbb{C}}$ differs from $\mathbf{NQP}_{\mathbb{C}}$.

Corollary 3.7 $\mathbf{BQP}_{\mathbb{C}} \neq \mathbf{NQP}_{\mathbb{C}}$.

The proof of Theorem 3.5 consists of two parts: co-$\mathbf{C_=P} \subseteq \mathbf{NQP}_{\{0,\pm\frac{3}{5},\pm\frac{4}{5},\pm1\}}$ and $\mathbf{NQP}_{\mathbb{C}} \subseteq$ co-$\mathbf{C_=P}$. The proof of the first claim recently appeared in [12]. For completeness, however, we include a proof of the first claim below. The second claim needs an elaborate argument and will be proved in the next section.

Let S be any set in co-$\mathbf{C_=P}$. By definition, there exists a $\mathbf{GapP}$-function f such that, for every x, $x \in S$ if and only if $f(x) \neq 0$. Without loss of generality, we can assume that, for some polynomial p and some deterministic polynomial-time computable predicate[‡] R, $f(x) = |\{y \in \{0,1\}^{p(|x|)} \mid R(x,y)=1\}| - |\{y \in \{0,1\}^{p(|x|)} \mid R(x,y)=0\}|$ for all binary strings x.

We wish to design a quantum algorithm with $\{0,\pm\frac{3}{5},\pm\frac{4}{5},\pm1\}$-amplitude that produces, on input x, a particular configuration with amplitude $-\epsilon^{p(|x|)+1}f(x)$, where $\epsilon = 12/25$, so that we can observe this configuration with positive probability if and only if $x \in S$. This implies that S is in $\mathbf{NQP}_{\{0,\pm\frac{3}{5},\pm\frac{4}{5},\pm1\}}$. To simplify our argument, we make use of the four letter alphabet $\Sigma_4 = \{0,1,2,3\}$.

Let I be the identity transform and let $H[a,b|\delta]$ be the generalized Hadamard transform defined as $\sum_{y,u\in\{a,b\}}(-1)^{[y=u=b]}\delta^{[y=u]}(1-\delta)^{[y\neq u]}|u\rangle\langle y|$, where $a,b\in\Sigma_4$, $\delta\in\mathbb{C}$, and the square brackets mean the truth value.[§] Moreover, let $H = H[0,1|\frac{4}{5}]$, $J = H[0,1|\frac{3}{5}]$, and $K = H[0,2|\frac{3}{5}] + H[1,3|\frac{4}{5}]$. Notice that H, I, J, and K are unitary and their amplitudes are all in $\{0,\pm\frac{3}{5},\pm\frac{4}{5},\pm1\}$.

Let x be an input of length n. We start with the initial superposition $|\phi_0\rangle = |0^{p(n)}\rangle|0\rangle$. We apply the operations $H^{p(n)} \otimes I$ to $|\phi_0\rangle$. Next we change the content of the last track from $|0\rangle$ to $|R(x,y)\rangle$. This can be done reversibly in polynomial time since R is computable by a polynomial-time reversible Turing machine [3, 4]. Finally we apply the operations $J^{p(n)} \otimes JK$ to this last superposition and let $|\phi\rangle$ denote the consequence.

Let $|\phi_1\rangle$ denote the *observable* $|0^{p(n)}\rangle|1\rangle$. When we observe $|\phi\rangle$, we can find state $|\phi_1\rangle$ with amplitude $\langle\phi_1|\phi\rangle$, which is $\epsilon^{p(n)}\sum_{y\in\{0,1\}^{p(n)}}(-1)^{R(x,y)}\epsilon$ since $\langle 1|JK|R(x,y)\rangle = (-1)^{R(x,y)}\epsilon$. By the definition of f, $\langle\phi_1|\phi\rangle$ is equal to $-\epsilon^{p(n)+1}f(x)$.

4 Proof of the Main Theorem

This section completes the proof of Theorem 3.5 by proving $\mathbf{NQP}_{\mathbb{C}} \subseteq$ co-$\mathbf{C_=P}$. Assume that S is in $\mathbf{NQP}_{\mathbb{C}}$. By Definition 3.4, there exists an element $p \in \mathbb{Z}[x]$ and an ℓ-track conservative quantum Turing machine $M = (\Sigma, Q, \delta)$ with $\mathbb{C}$-amplitudes that recognizes S in time $p(n)$ on any input of

‡A predicate can be seen as a function from $\{0,1\}^* \times \{0,1\}^*$ to $\{0,1\}$.

§Conventionally we set TRUTH=1 and FALSE=0. For example, $[0=0]=1$ and $[0=1]=0$.

length n. Let D be the set of all transition amplitudes of δ; that is, $D = \{\delta(p', \boldsymbol{\sigma}, q', \boldsymbol{\tau}, d') \mid p', q' \in Q, \boldsymbol{\sigma}, \boldsymbol{\tau} \in \Sigma^{\ell}, d' \in \{L, R\}\}$. We must show that S is in co-$\mathbf{C_{=}P}$.

The key ingredient of our proof is, similar to Lemma 6.6 in [1], to show that, for some constant $u \in \mathbb{C}$, every amplitude of a configuration in a superposition generated by M at time t, when multiplied by the factor u^{2t-1}, is uniquely expressed as a linear combination of $O(poly(t))$ linearly independent monomials with integer coefficients. If each basic monomial is properly indexed, any transition amplitude can be encoded as a collection of pairs of such indices and their integer coefficients. This encoding enables us to carry out amplitude calculations on a classical Turing machine.

We first show that any number in D can be expressed in a certain canonical way. Let $A = \{\alpha_i\}_{1\leq i\leq m}$ be any maximal algebraically independent subset of D and define $F = \mathbb{Q}(A)$, i.e., the field generated by all elements in A over $\mathbb{Q}$. We further define G to be the field generated by all the elements in $\{1\} \cup (D \setminus A)$ over F. Let $B = \{\beta_i\}_{0\leq i<d}$ be a basis of G over F. For convenience, we assume $\beta_0 = 1$ so that, even in the special case $A = D$, $\{\beta_0\}$ becomes a basis of G over F. Let $D' = D \cup \{\beta_i\beta_j\}_{0\leq i,j<d}$.

For each element α in G, since B is a basis, α can be uniquely written as $\sum_{j=0}^{d-1} \lambda_j\beta_j$ for some $\lambda_j \in F$. Since the elements in A are all algebraically independent, each λ_j can be written as s_j/u_j, where each s_j and u_j is a finite sum of linearly independent monomials of the form $a_{\boldsymbol{k}_j}(\prod_{i=1}^{m} \alpha_i^{k_{ij}})$ for some $\boldsymbol{k}_j = (k_{1j}, k_{2j}, \ldots, k_{mj}) \in \mathbb{Z}^m$ and $a_{\boldsymbol{k}_j} \in \mathbb{Z}$. Unfortunately, this representation is in general not unique, since $s_j/u_j = (s_jr)/(u_jr)$ for any non-zero element r.

To give a standard form for all the elements in D', we need to "normalize" them by choosing an appropriate common denominator. Let $u \in G$ be any common denominator of all the elements α in D' such that $u\alpha$ is written as $\sum_{\boldsymbol{k}} a_{\boldsymbol{k}}(\prod_{i=1}^{m} \alpha_i^{k_i})\beta_k$, where $\boldsymbol{k} = (k, k_1, k_2, \ldots, k_m) \in \mathbb{Z}_d \times \mathbb{Z}^m$ and $a_{\boldsymbol{k}} \in \mathbb{Z}$. Notice that such a form is uniquely determined by a collection of pairs of $\boldsymbol{k}$ and $a_{\boldsymbol{k}}$. We call this unique form the *canonical form* of $u\alpha$. Fix u from now on. For the canonical form, we call $\boldsymbol{k}$ an *index* and $a_{\boldsymbol{k}}$ a *major sign* of $u\alpha$ with respect to index $\boldsymbol{k}$ (or a *major* $\boldsymbol{k}$*-sign*, for short). An index $\boldsymbol{k}$ is said to be *principal* if the major $\boldsymbol{k}$-sign is nonzero. For each $\alpha \in D'$, let $ind(u\alpha)$ be the maximum of $|\boldsymbol{k}|$ over all principal indices $\boldsymbol{k}$ of $u\alpha$. Moreover, let e be the maximum of d and of $ind(u\alpha)$ over all elements α in D'.

A crucial point of our proof relies on the following lemma.

Lemma 4.1 *The amplitude of each configuration of M on input x in any superposition at time t, $t > 0$, when multiplied by the factor u^{2t-1}, can be written in the canonical form $\sum_{\boldsymbol{k}} a_{\boldsymbol{k}}(\prod_{i=1}^{m} \alpha_i^{k_i})\beta_k$, where $\boldsymbol{k} = (k, k_1, k_2, \ldots, k_m)$ ranges over $\mathbb{Z}_d \times (\mathbb{Z}_{[2et]})^m$ and $a_{\boldsymbol{k}} \in \mathbb{Z}$.*

Proof. Let $\alpha_{C,t}$ denote the amplitude of configuration C of M on input x in a superposition at time t. When $t = 1$, the lemma is trivial. Let C' be any configuration in a superposition at time $t+1$.

Note that $u^{2t+1}\alpha_{C',t+1}$ is a sum of $u^2(u^{2t-1}\alpha_{C,t})\delta_{C,C'}$ over all configurations C, where $\delta_{C,C'}$ denotes the transition amplitude of δ that corresponds to the transition from C to C' in a single step. By the induction hypothesis, $u^{2t-1}\alpha_{C,t}$ has a canonical form as in the lemma. Hence, it suffices to show that, for each configuration C and each index $\boldsymbol{k} \in \mathbb{Z}_d \times (\mathbb{Z}_{[2et]})^m$, $\alpha'_{C,C',\boldsymbol{k}} \stackrel{\text{def}}{=} u^2(\prod_{i=1}^m \alpha_i^{k_i})\beta_k\delta_{C,C'}$ has a canonical form in which all the principle indices lie in $\mathbb{Z}_d \times (\mathbb{Z}_{[2e(t+1)]})^m$ since $u^{2t+1}\alpha_{C',t+1}$ is expressed as the sum of $a_{\boldsymbol{k}}\alpha'_{C,C',\boldsymbol{k}}$ over all C and $\boldsymbol{k}$.

Let $\boldsymbol{k} = (k, k_1, \ldots, k_m)$ be an index in $\mathbb{Z}_d \times (\mathbb{Z}_{[2et]})^m$, which corresponds to monomial $(\prod_{i=1}^m \alpha_i^{k_i})\beta_k$. We first show that $\alpha'_{C,C',\boldsymbol{k}}$ has a canonical form. Since $\delta_{CC'} \in D'$, we can assume that the canonical form of $u\delta_{C,C'}$ is $\sum_{\boldsymbol{j}} b_{\boldsymbol{j}}(\prod_{i=1}^m \alpha_i^{j_i})\beta_j$, where $\boldsymbol{j} = (j, j_1, \ldots, j_m)$ ranges over $\mathbb{Z}_d \times (\mathbb{Z}_{[e]})^m$ and $b_{\boldsymbol{j}} \in \mathbb{Z}$. Then, $\alpha'_{C,C',\boldsymbol{k}}$ is written as:

$$(*) \qquad \alpha'_{C,C',\boldsymbol{k}} = \sum_{\boldsymbol{j}} b_{\boldsymbol{j}} \left(\prod_{i=1}^m \alpha_i^{k_i+j_i}\right) u\beta_k\beta_j = \sum_{\boldsymbol{j}} \sum_{\boldsymbol{h}_j} b_{\boldsymbol{j}} c_{\boldsymbol{h}_j} \left(\prod_{i=1}^m \alpha_i^{k_i+j_i+h_{ij}}\right) \beta_{h_j},$$

provided that $u\beta_k\beta_j$ has a canonical form $\sum_{\boldsymbol{h}_j} c_{\boldsymbol{h}_j}(\prod_{i=1}^m \alpha_i^{h_{ij}})\beta_{h_j}$, where $\boldsymbol{h}_j = (h_j, h_{1j}, \ldots, h_{mj})$ ranges over $\mathbb{Z}_d \times (\mathbb{Z}_{[e]})^m$ and $c_{\boldsymbol{h}_j} \in \mathbb{Z}$. Since $b_{\boldsymbol{j}}c_{\boldsymbol{h}_j} \in \mathbb{Z}$, $\alpha'_{C,C',\boldsymbol{k}}$ must have a canonical form. For later use, let $h(x, C, \boldsymbol{k}, C', \boldsymbol{k'})$ be the major $\boldsymbol{k'}$-sign of $\alpha'_{C,C',\boldsymbol{k}}$ for any index $\boldsymbol{k'} = (k', k'_1, \ldots, k'_m)$.

We next show that $ind(\alpha'_{C,C',\boldsymbol{k}}) \leq 2e(t+1)$. By $(*)$ it follows that $ind(\alpha'_{C,C',\boldsymbol{k}})$ is bounded above by the maximum of $k_i + j_i + h_{ij}$, which is at most $|\boldsymbol{k}| + |\boldsymbol{j}| + |\boldsymbol{h}_j| \leq 2et + 2e = 2e(t+1)$; in other words, all the principal indices of $\alpha'_{C,C',\boldsymbol{k}}$ must lie in $\mathbb{Z}_d \times (\mathbb{Z}_{[2e(t+1)]})^m$. This also shows that $h(x, C, \boldsymbol{k}, C', \boldsymbol{k'})$ is computed[¶] deterministically in time polynomial in the length of C and C' and also in $|\boldsymbol{k}|$ and $|\boldsymbol{k'}|$ since $h(x, C, \boldsymbol{k}, C', \boldsymbol{k'})$ is the sum of $b_{\boldsymbol{j}}c_{\boldsymbol{h}_j}$ over all pairs of $\boldsymbol{j}$ and $\boldsymbol{h}_j$ such that $h_j = k'$ and $k_i + j_i + h_{ij} = k'_i$ for each i with $1 \leq i \leq m$. □

In what follows, we show how to simulate a quantum computation of M. First we define a function f as follows. Let x be a string of length n, C an accepting configuration of M on input x, and $\boldsymbol{k}$ an index. Let $f(x, C, \boldsymbol{k})$ be the major $\boldsymbol{k}$-sign of $u^{2p(n)-1}$ times the amplitude of C in the final superposition of M on input x. For convenience, we set $f(x, C, \boldsymbol{k}) = 0$ for any other set of inputs $(x, C, \boldsymbol{k})$.

Notice by Lemma 4.1 that M rejects x with certainty if and only if the amplitude of any accepting configuration of M on x, multiplied by $u^{2p(n)-1}$, has major sign 0 with respect to any principal index. The following lemma is thus immediate.

Lemma 4.2 *For every x, $x \notin S$ if and only if, for every accepting configuration C of M on input x and for every index $\boldsymbol{k} \in \mathbb{Z}_d \times (\mathbb{Z}_{[2ep(n)]})^m$, $f(x, C, \boldsymbol{k}) = 0$.*

We want to show that f is a **GapP**-function. Theorem 3.5 follows once this is proved. To see

[¶]We assume that C, C', $\boldsymbol{k}$, and $\boldsymbol{k'}$ are appropriately encoded into strings in Σ^*.

this, define

$$g(x) = \sum_{C} \sum_{\boldsymbol{k}} f^2(x, C, \boldsymbol{k}),$$

where C ranges over all accepting configurations of M on input x and $\boldsymbol{k}$ is drawn from $\mathbb{Z}_d \times (\mathbb{Z}_{[2ep(n)]})^m$. It follows from Lemma 3.2 that g is also in **GapP**, and by Lemma 4.2 $g(x) = 0$ if and only if $x \notin S$. This yields the desired conclusion that S is in co-$\mathbf{C_{=}P}$.

To show $f \in \mathbf{GapP}$, let $\boldsymbol{C} = \langle C_0, C_1, \ldots, C_{p(n)} \rangle$ be any "computation path" of M on input x of length n; that is, C_0 is the initial configuration of M on input x and δ transforms C_{i-1} into C_i in a single step. Also let $\boldsymbol{K} = \langle \boldsymbol{k}_0, \boldsymbol{k}_1, \ldots, \boldsymbol{k}_{p(n)} \rangle$ be any sequence of indices in $\mathbb{Z}_d \times (\mathbb{Z}_{[2ep(n)]})^m$ such that $\boldsymbol{k}_0 = \mathbf{0}^{m+1}$. We define $h'(x, \boldsymbol{C}, \boldsymbol{K})$ to be the product of $h(x, C_{i-1}, \boldsymbol{k}_{i-1}, C_i, \boldsymbol{k}_i)$ over all i, $1 \leq i \leq p(n)$, where h has been defined in the proof of Lemma 3.2. Notice that h' is polynomial-time computable since h is.

Note that the sum $\sum_{\boldsymbol{K}} h'(x, \boldsymbol{C}, \boldsymbol{K})$ relates to the major $\boldsymbol{k}$-sign of the amplitude, multiplied by $u^{2p(n)-1}$, of the computation path $\boldsymbol{C}$, where $\boldsymbol{K} = \langle \boldsymbol{k}_0, \boldsymbol{k}_1, \ldots, \boldsymbol{k}_{p(n)} \rangle$ ranges over $\mathbb{Z}_d \times (\mathbb{Z}_{[2ep(n)]})^m$ with $\boldsymbol{k}_0 = \mathbf{0}^{m+1}$ and $\boldsymbol{k}_{p(n)} = \boldsymbol{k}$. The following equation is thus straightforward.

$$f(x, C, \boldsymbol{k}) = \sum_{\boldsymbol{K}} \sum_{\boldsymbol{C}} h'(x, \boldsymbol{C}, \boldsymbol{K}),$$

where $\boldsymbol{K} = \langle \boldsymbol{k}_0, \boldsymbol{k}_1, \ldots, \boldsymbol{k}_{p(n)} \rangle$ ranges over $(\mathbb{Z}_d \times (\mathbb{Z}_{[2ep(n)]})^m)^{p(n)}$ and $\boldsymbol{C} = \langle C_0, C_1, \ldots, C_{p(n)} \rangle$ is a computation path of M on input x such that $\boldsymbol{k}_0 = \mathbf{0}^{m+1}$, $\boldsymbol{k}_{p(n)} = \boldsymbol{k}$, and $C_{p(n)} = C$.

Lemma 3.2 guarantees that f is indeed a **GapP**-function. This completes the proof of Theorem 3.5.

5 Discussion

We have extended earlier works of [1, 11, 14] to show that nondeterministic polynomial-time quantum computation with arbitrary amplitudes can be completely characterized by Wagner's polynomial-time counting computation. Our result thus makes it possible to define the class **NQP** independent of the choice of amplitudes, whereas $\mathbf{BQP}_{\mathbb{C}}$ is known to differ from $\mathbf{BQP}_{\mathbb{Q}}$ [1]. We also note that the proof of Theorem 3.5 can relativize to an arbitrary oracle A; namely, $\mathbf{NQP}_K^A$ = co-$\mathbf{C_{=}P}^A$ for any set K with $\mathbb{Q} \subseteq K \subseteq \mathbb{C}$. As a result, for instance, we have $\mathbf{NQP}^{\mathbf{NQP}}$ = co-$\mathbf{C_{=}P}^{\mathbf{C_{=}P}}$ and thus $\mathbf{NQP} \subseteq \mathbf{PP} \subseteq \mathbf{NQP}^{\mathbf{NQP}} \subseteq \mathbf{PP}^{\mathbf{PP}}$. This implies that the hierarchy built over **NQP**, analogous to the polynomial-time hierarchy, interweaves into Wagner's counting hierarchy [22] over **PP**.

At the end, we remind the reader that the fact **NQP** = co-$\mathbf{C_{=}P}$ yields further consequences based on the well-known results on the class $\mathbf{C_{=}P}$. For example, $\mathbf{PP}^{\mathbf{PH}} \subseteq \mathbf{NP}^{\mathbf{NQP}}$ follows directly from $\mathbf{PP}^{\mathbf{PH}} \subseteq \mathbf{P}^{\mathbf{PP}}$ [20] and $\mathbf{NP}^{\mathbf{PP}} = \mathbf{NP}^{\mathbf{C_{=}P}}$ [21] and it also follows from [17] that all sparse **NQP** sets are in **APP**. Moreover, **NQP** = co-**NQP** if and only if $\mathbf{PH}^{\mathbf{PP}} = \mathbf{NQP}$, which follows from a result in [15]. Note that these results also follow from [11].

References

[1] L. M. Adleman, J. DeMarrais, and M. A. Huang, Quantum computability, *SIAM J. Comput.*, **26** (1997), 1524–1540.

[2] D. Aharonov, Quantum computation, in *Annual Reviews of Computational Physics VI*, ed. Dietrich Stauffer, World Scientific, 1998.

[3] C. H. Bennett, Logical reversibility of computation, *IBM J. Res. Develop.*, **17** (1973), 525–532.

[4] E. Bernstein and U. Vazirani, Quantum complexity theory, *SIAM J. Comput.*, **26** (1997), 1411–1473.

[5] A. Berthiaume, Quantum computation, in *Complexity Theory Retrospective II*, eds. L.A. Hemaspaandra and A.L. Selman, pp.23–51, Springer, 1997.

[6] P. Benioff, Quantum mechanical Hamiltonian models of Turing machines, *J. Stat. Phys.* **29** (1982), 515–546.

[7] D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer, *Proc. Roy. Soc. London,* Ser.A, **400** (1985), 97–117.

[8] D. Deutsch, Quantum computational networks, *Proc. Roy. Soc. London,* Ser.A, **425** (1989), 73–90.

[9] D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, *Proc. Roy. Soc. London,* Ser.A, **439** (1992), 553–558.

[10] S. Fenner, L. Fortnow, and S. Kurtz, Gap-definable counting classes, *J. Comput. and System Sci.*, **48** (1994), 116–148.

[11] S. Fenner, F. Green, S. Homer, and R. Pruim, Quantum NP is hard for PH, in *Proc. 6th Italian Conference on Theoretical Computer Science*, World-Scientific, Singapore, pp.241–252, 1998.

[12] S. Fenner, F. Green, S. Homer, and R. Pruim, Determining acceptance possibility for a quantum computation is hard for the polynomial hierarchy, quant-ph/9812056, December 18, 1998.

[13] R. Feynman, Quantum mechanical computers, *Found. Phys.*, **16** (1986), 507–531.

[14] L. Fortnow and J. Rogers, Complexity limitations on quantum computation, *Proc. 13th IEEE Conference on Computational Complexity*, pp.202–209, 1998.

[15] F. Green, On the power of deterministic reductions to $C_{=}P$, *Math. Systems Theory,* **26** (1993), 215–233.

[16] L. K. Grover, A fast quantum mechanical algorithm for database search, *Proceedings of 28th ACM Symposium on Theory of Computing*, pp.212-219, 1996.

[17] L. Li, *On the counting functions*, Ph.D. dissertation, Department of Computer Science, University of Chicago, 1993.

[18] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.*, **26** (1997), 1484–1509.

[19] D. R. Simon, On the power of quantum computation, *SIAM J. Comput*, **26** (1997), 1474–1483.

[20] S. Toda, PP is as hard as the polynomial-time hierarchy, *SIAM J. Comput.*, **20** (1991), 865–877.

[21] J. Torán, Complexity classes defined by counting quantifiers, *J. ACM*, **38** (1991), 753–774.

[22] K. Wagner, The complexity of combinatorial problems with succinct input representation, *Acta Inf.* **23** (1986), 325–356.

[23] T. Yamakami, A foundation of programming a multi-tape quantum Turing machine, to appear in *Proc. 24th International Symposium on Mathematical Foundation of Computer Science*, Lecture Note in Computer Science, September, 1999.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'