A Schematic Definition of Quantum Polynomial Time Computability

Tomoyuki Yamakam ‡

Abstract

In the past four decades, the notion of quantum polynomial-time computability has been mathematically modeled by quantum Turing machines as well as quantum circuits. This paper seeks the third model, which is a quantum analogue of the schematic (inductive or constructive) definition of (primitive) recursive functions. For quantum functions mapping finite-dimensional Hilbert spaces to themselves, we present such a schematic definition, composed of a small set of initial quantum functions and a few construction rules that dictate how to build a new quantum function from the existing ones. We prove that our schematic definition precisely characterizes all functions that can be computable with high success probabilities on well-formed quantum Turing machines in polynomial time, or equivalently uniform families of polynomial-size quantum circuits. Our new, schematic definition is quite simple and intuitive and, more importantly, it avoids the cumbersome introduction of the well-formedness condition imposed on a quantum Turing machine model as well as of the uniformity condition necessary for a quantum circuit model. Our new approach can further open a door to the descriptional complexity of quantum functions, to the theory of higher-type quantum functionals, to the development of new first-order theories for quantum computing, and to the designing of programming languages for real-life quantum computers.

Key words: quantum computing, quantum function, quantum Turing machine, quantum circuit, schematic definition, descriptional complexity, polynomial-time computability, normal form theorem

1 Background, Motivation, and the Main Results

In early 1980s emerged a groundbreaking idea of exploiting quantum physics to build mechanical computing devices, dubbed as quantum computers, which have completely altered the way we used to envision "computers." Subsequent discoveries of more efficient quantum computations for factoring positive integers [30] and searching unstructured databases [14, 15] than classical computations prompted us to look for more mathematical and practical problems that can be solvable effectively on the quantum computers. Efficiency in quantum computing has since then rapidly become an important research subject of computer science as well as physics.

As a mathematical model to realize quantum computation, Deutsch 11 introduced a notion of quantum Turing machine (or QTM, for short), which was later discussed by Yao 40 and further refined by Bernstein and Vazirani [5]. This mechanical model largely expands the classical model of (probabilistic) Turing machine by allowing a physical phenomenon, called quantum interference, to take place on its computation. A different Hamiltonian formalism of Turing machines was also suggested by Benioff [3]. A QTM has an ability of computing a quantum function mapping a finite-dimensional Hilbert space to itself by evolving unitarily a superposition of (classical) configurations of the machine, starting with a given input string and an initial inner state. A more restrictive use of the term of "quantum function" is found in, e.g., [38], in which quantum functions take classical input strings and produce either classical output strings of QTMs or acceptance probabilities of QTMs. Throughout this paper, nevertheless, quantum functions refer only to functions acting on Hilbert spaces of arbitrary dimensions.

To ensure the unitary nature of quantum computation, a QTM requires its mechanism to meet the socalled well-formedness condition on a single-tape model of QTMs 5 and a multi-tape model 36, 38 as well as [27]. Refer to Section 2.2 for their precise definitions.

Bernstein and Vazirani further formulated a new complexity class, denoted by BQP, as the collection of all languages recognized by well-formed QTMs running in polynomial time with error probability bounded

from above by 1/3. Furthermore, QTMs equipped with output tapes can compute string-valued functions in place of languages, and those functions form a function class, called FBQP .

From a different viewpoint, Yao 40 expanded Deutsch's notion of quantum network 12 and formalized a notion of quantum circuit, which is a quantum analogue of classical Boolean circuit. Different from the classical Boolean circuit model, a quantum circuit is composed of quantum gates, each of which represents a unitary transformation acting on a Hilbert space of a small, fixed dimension. To act as a "programmable" unitary operator, a family of quantum circuits requires the so-called uniformity condition, which ensures that a blueprint of each quantum circuit is easily rendered. Yao further demonstrated that a uniform family of quantum circuits is powerful enough to simulate a well-formed quantum Turing machine. As Nishimura and Ozawa 26 pointed out, the

*This work was done while the author was at the University of Ottawa between 1999 and 2003, and it was financially supported by the Natural Sciences and Engineering Research Council of Canada.

† An extended abstract appeared under the title "A recursive definition of quantum polynomial time computability (extended abstract)" in the Proceedings of the 9th Workshop on Non-Classical Models of Automata and Applications (NCMA 2017), Prague, Czech Republic, August 17-18, 2017, Österreichische Computer Gesellschaft 2017, the Austrian Computer Society, pp.243-258, 2017. The current paper intends to correct erroneous descriptions in the extended abstract and provide more detailed explanations to all omitted proofs due to the page limit.

‡ Current Affiliation: Faculty of Engineering, University of Fukui, 3-9-1 Bunkyo, Fukui, 910-8507 Japan

uniformity condition of a quantum circuit family is necessary to precisely capture quantum polynomial-time computation. With this uniformity condition, BQP and FBQP are characterized exactly by uniform families of quantum circuits made up of polynomially many quantum gates.

This current paper boldly takes the third approach toward the characterization of quantum polynomialtime computability. Unlike the aforementioned mechanical device models, our approach is to extend the schematic (inductive or constructive) definition of (primitive) recursive functions on natural numbers. Such a schematic definition was thought in the 19th century by Peano [28], opposed to the definition given by Turing's machine model 32. This classical scheme comprises a small set of initial functions and a small set of rules, which dictate how to construct a new function from the existing ones. For instance, every primitive recursive function is built from the constant, successor, and projection functions by applying composition and primitive recursion finitely many times. In particular, the primitive recursion introduces a new function whose values are defined by induction. Recursive functions (in the form of μ-recursive functions [19, 20]) further require an additional scheme, known as the minimization (or the least number) operator. These functions coincide with the Herbrand-Gödel formalism of general recursive functions (see [10]). For a historical account of these notions, refer to, e.g., 29. Similar schematic approaches to capture classical polynomial-time computability have already been sought in the literature [7, 8, 9, 24, 35. Those approaches have led to quite different research subjects from what the Turing machine model provides.

Our purpose in this paper is to give a schematic definition of quantum functions to capture the notion of quantum polynomial-time computability and, more importantly, to make such a definition simpler and more intuitive for a practical merit of our own. Our schematic definition (Definition 3.1) includes a set of initial quantum functions, I (identity), NOT (negation of a qubit), $PHASE_\theta$ (phase shift by $e^{i\theta}$), ROT_θ (rotation around xy-axis by angle θ), $SWAP$ (swap between two qubits), and MEAS (partial projective measurement), as well as construction rules, composed of composition (Compo $[\cdot,\cdot]$), branching (Branch $[\cdot,\cdot]$), and multi-qubit quantum recursion ($kQRec[\cdot,\cdot \mid \cdot]$). Our choice of these initial quantum functions and construction rules stems mostly from a universal set of quantum gates in use in the past literature. Our quantum recursion, on the contrary, is quite different in nature from the primitive recursion used to build primitive recursive functions. Instead of using the successor function to count down the number of inductive iterations in the primitive recursion, the quantum recursion uses a divide-and-conquer strategy of reducing the number of accessible qubits needed for performing a specified quantum function. Within our new framework, we can implement typical unitary operators, such as the Walsh-Hadamard transform (WH), the controlled-NOT (CNOT), and the global phase shift (GPS).

An immediate merit of our schematic definition is that we can avoid the cumbersome introduction of the well-formedness condition imposed on the QTM model and the uniformity condition on the quantum circuit model. Another advantage of our schemata is that each scheme has its own inverse; namely,

for any quantum function g defined by one of the schemata, its inverse g^{-1} is also defined by the same kind of scheme. For instance, the inverses of the quantum functions ROT_θ and $kQRec_t\left[g, h, p \mid \{f_s\}_{s\in\{0,1\}^k}\right]$ introduced in Definition 3.1 are exactly $ROT_{-\theta}$ and $kQRec_t\left[g^{-1}, p^{-1}, h^{-1} \mid \left\{f_s^{-1}\right\}_{s\in\{0,1\}^k}\right]$, respectively (Proposition 3.5).

For a further explanation of our main contributions, it is time to introduce a succinct notation of $\square_1^{\mathrm{QP}}$ (where $\square$ is pronounced "square") to denote the set of all quantum functions built from the initial quantum functions and by a finite series of sequential applications of the construction rules. Since the partial measurement (MEAS) is not a unitary operator, we denote the class obtained from $\square_1^{\mathrm{QP}}$ without use of MEAS by $\widehat{\square_1^{\mathrm{QP}}}$. Briefly, let us discuss clear differences between our schematic definition and the aforementioned two formalisms of polynomial-time quantumly computable functions in terms of QTMs and quantum circuits. Two major differences are listed below.

1. While a single quantum circuit takes a fixed number of input qubits, our quantum function takes an "arbitrary" number of qubits as an input. This situation is similar to QTMs because a QTM has an infinite tape and uses an arbitrary number of tape cells during its computation as extra storage space. On the contrary to the QTMs, a $\widehat{\square_1^{\mathrm{QP}}}$-function is constructed using the same number of qubits as its original input
in such a way that a quantum circuit has the same number of input qubits and output qubits.

2. The two machine models exhort an algorithmic description to dictate the behavior of each machine; more specifically, a QTM uses a transition function, which algorithmically describes how each step of the machine acts on certain qubits, and a family of quantum circuits uses its uniformity condition to render the design of quantum gates in each quantum circuit. Unlike these two models, no $\square_1^{\mathrm{QP}}$-function has any mechanism to store information on the description of the function itself but the construction process itself specifies the behavior of the function.

As a consequence, the above mentioned differences help the $\square_1^{\mathrm{QP}}$-functions take a distinctive position among all possible computational models that characterize quantum polynomial-time computability, and therefore we expect them to play an important role in analyzing the features of quantum polynomial-time computation from a quite different perspective.

In Section 3.1, we will formally present our schematic definition of $\square_1^{\mathrm{QP}}$ -functions (as well as $\widehat{\square_1^{\mathrm{QP}}}$ functions) and show in Section 4.1 that $\square_1^{\mathrm{QP}}$ (also $\widehat{\square_1^{\mathrm{QP}}}$) can characterize all functions in FBQP. More precisely, we assert in the main theorem (Theorem 4.1) that any function from $\{0,1\}^*$ to $\{0,1\}^*$ in FBQP can be characterized by a certain polynomial p and a certain quantum function $g \in \square_1^{\mathrm{QP}}$ in such a way that, by using an appropriate coding scheme, in the final

quantum state of g on instances x and the runtime bound $p(|x|)$, we observe an output value $f(x)$ with high probability. This theorem will be split into two lemmas, Lemmas 4.2 and 4.3 The former lemma will be proven in Section 4.1 however, the proof of the latter lemma is so lengthy that it will be postponed until Section 5. In this proof, we will construct a $\Box_1^{\mathrm{QP}}$-function that can simulate the behavior of a given QTM.

Notice that, since BQP is a special case of FBQP, BQP is also characterized by our model. In our proof of the characterization theorem (Theorem 4.1), we will utilize a main result of Bernstein and Vazirani [5] and that of Yao 40 extensively. In Section 4.2, we will apply our characterization, in the help of a universal QTM [5, 26], to obtain a quantum version of Kleene's normal form theorem [19, 20, in which there is a universal pair of primitive recursive predicate and function that can describe the behavior of every recursive function.

Unlike classical computation on natural numbers (equivalently, strings over finite alphabets by appropriate coding schemata), quantum computation is a series of certain manipulations of a single vector in a finite-dimensional Hilbert space and we need only high precision to approximate each function in FBQP by such a vector. This fact allows us to choose a different set of schemata (initial quantum functions and construction rules) to capture the essence of quantum computation. In Section 6.1, we will discuss this issue using an example of a general form of the quantum Fourier transform (QFT). This transform may not be "exactly" computed in our current framework of $\Box_1^{\mathrm{QP}}$ but we can easily expand $\Box_1^{\mathrm{QP}}$ to compute the generalized QFT exactly if we include an additional initial quantum function, such as CROT (controlled rotation).

Concerning future research on the current subject, we will discuss in Section 6 four new directions of the subject. Our schematic definition provides not only a different way of describing languages and functions computable quantumly in polynomial time but also a simple way of measuring the "descriptional" complexity of a given language and a function restricted to instances of specified length. This new complexity measure will be useful to prove basic properties of $\widehat{\Box_1^{\mathrm{QP}}}$-functions in Section 3. Its future application will be briefly discussed in Section 6.2

Kleene 21, 22 defined recursive functionals of higher types by extending the aforementioned recursive functions on natural numbers. A more general study of higher-type functionals has been conducted in computational complexity theory for decades [8, 9, 24, 31, 35]. In a similar spirit, our schematic definition enables us to study higher-type quantum functionals. In Section 6.3, using oracle functions (function oracles or oracles), we will define type- 2 quantum functionals, which may guide us to a rich field of research in the future.

A schematic definition of how to construct a target $\Box_1^{\mathrm{QP}}$-function can be viewed as a "program" that describes a series of instructions on which schemata to use. Hence, our schematic formulation opens a door to a new, practical application to the designing of succinct programming languages to control the operations of real-life quantum computers. In Section 6.4, we will briefly argue on an application of the schematic definition to the future development of

"quantum programming languages." As a further application of our schematic definition, we can look into a new aspect of first-order theories and their subtheories. Earlier, a quantum analogue of NP (nondeterministic polynomial time class) and beyond were sought in [37] with the use of bounded quantifiers over quantum states in finite-dimensional Hilbert spaces. In a similar vein, we expect that $\Box_1^{\mathrm{QP}}$ will serve as a foundation to the introduction of first-order theories and their subtheories over quantum states in Hilbert spaces.

2 Fundamental Notions and Notation

We begin with explaining basic notions and notation necessary to read through the subsequent sections. Let us assume the reader's familiarity with classical Turing machines (see, e.g., [16]). For the foundation of quantum information and computation, in contrast, the reader refers to basic textbooks, e.g., [18, 25].

2.1 Numbers, Languages, and Qustrings

The notation $\mathbb{Z}$ indicates the set of all integers and $\mathbb{N}$ expresses the set of all natural numbers (that is, non-negative integers). For convenience, we set $\mathbb{N}^+ = \mathbb{N} - \{0\}$. Moreover, $\mathbb{Q}$ denotes the set of all rational numbers and $\mathbb{R}$ indicates the set of all real numbers. For two numbers $m, n \in \mathbb{Z}$ with $m \leq n$, the notation $[m, n]_{\mathbb{Z}}$ denotes an integer interval $\{m, m+1, m+2, \ldots, n\}$, compared to a real interval $[\alpha, \beta]$ for $\alpha, \beta \in \mathbb{R}$ with $\alpha \leq \beta$. In particular, $[n]$ is shorthand for $[1, n]_{\mathbb{Z}}$ for any $n \in \mathbb{N}^+$. By $\mathbb{C}$, we express the set of all complex numbers. Given $\alpha \in \mathbb{C}, \alpha^*$ expresses the complex conjugate of α. Polynomials are assumed to have natural numbers as their coefficients and they thus produce nonnegative values from nonnegative inputs. A real number α is called polynomial-time approximabl if there exists a multi-tape polynomial-time deterministic Turing machine M (equipped with a write-only output tape) that, on each input of the form 1^n for a natural number n, produces a finite binary fraction, $M(1^n)$, on its designated output tape with $|M(1^n) - \alpha| \leq 2^{-n}$. Let $\tilde{\mathbb{C}}$ be the set of complex numbers whose real and imaginary parts are both polynomial-time approximable. For a bit $a \in \{0, 1\}, \bar{a}$ indicates $1 - a$. Given a matrix A, A^T denotes its transpose and $A^\dagger$ denotes the transposed conjugate of A.

An alphabet is a finite nonempty set of "symbols" or "letters." Given such an alphabet Σ, a string over Σ is a finite series of symbols taken from Σ. The concatenation of two strings u and w is expressed as $u \cdot w$ or more simply uw. The length of a string x, denoted by $|x|$, is the number of all occurrences of symbols in x. In particular, the empty string has length 0 and is denoted λ. We write Σ^n for the subset of Σ^* consisting only of all strings of length n and we set $\Sigma^* = \bigcup_{n \in \mathbb{N}} \Sigma^n$ (the set of all strings over Σ). A language over Σ is a subset of Σ^*. Given a language S, its characteristic function is also expressed by S; that is, $S(x) = 1$ for all $x \in S$ and $S(x) = 0$ for all $x \notin S$. A function on Σ^* (i.e., from Σ^* to Σ^*) is polynomially bounded if there exists a polynomial p satisfying $|f(x)| \leq p(|x|)$ for all strings $x \in \Sigma^*$.

For each natural number $k \geq 1, \mathcal{H}_k$ expresses a Hilbert space of dimension k and each element of $\mathcal{H}_k$ is expressed as $|\phi\rangle$ using Dirac's "ket" notation. In this paper, we are interested only in the case where k is a power of 2 and we implicitly assume that k is of the form 2^n for a certain $n \in \mathbb{N}$. Any element of $\mathcal{H}_2$ that has the unit norm is called a quantum bit or a qubit. By choosing a standard computational basis $B_1 = \{|0\rangle, |1\rangle\}$, every qubit $|\phi\rangle$ can be expressed as $\alpha_0|0\rangle + \alpha_1|1\rangle$ for an appropriate choice of two values $\alpha_0, \alpha_1 \in \mathbb{C}$ (called amplitudes) satisfying $|\alpha_0|^2 + |\alpha_1|^2 = 1$. We also express $|\phi\rangle$ as a column vector of the form $\binom{\alpha_0}{\alpha_1}$; in particular, $|0\rangle = \binom{1}{0}$ and $|1\rangle = \binom{0}{1}$. In a more general case of $n \geq 1$, we use $B_n = \{|s\rangle \mid s \in \{0,1\}^n\}$ as a computational basis of $\mathcal{H}_{2^n}$ with $|B_n| = 2^n$. Given any number $n \in \mathbb{N}^+$, a qustring of length n is a vector $|\phi\rangle$ of $\mathcal{H}_{2^n}$ with unit norm; namely, it is of the form $\sum_{s\in\{0,1\}^n} \alpha_s|s\rangle$, where each amplitude α_s is in $\mathbb{C}$ with $\sum_{s\in\{0,1\}^n} |\alpha_s|^2 = 1$. Notice that a qubit is a qustring of length 1 . The exception is the null vector, denoted simply by $\mathbf{0}$, which has norm 0 . Although the null vector could be a qustring of "arbitrary" length n, we instead refer to it as the qustring of length 0 for convenience. We use the notation Φ_n for each $n \in \mathbb{N}$ to denote the collection of all qustrings of length n. Finally, we set $\Phi_\infty = \bigcup_{n\in\mathbb{N}} \Phi_n$ (the set of all qustrings).

When $s = s_1 s_2 \cdots s_n$ with $s_i \in \{0,1\}$ for any index $i \in [n]$, the qustring $|s\rangle$ coincides with $|s_1\rangle \otimes |s_2\rangle \otimes \cdots \otimes |s_n\rangle$, where $\otimes$ denotes the tensor product and is expressed as, for example, $|00\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^T$, $|01\rangle = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}^T$, and $|11\rangle = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}^T$. The transposed conjugate of $|s\rangle$ is denoted by $\langle s|$ (with the "bra" notation). For instance, if $|\phi\rangle = \alpha|01\rangle + \beta|10\rangle$, then $\langle\phi| = \alpha^*\langle 01| + \beta^*\langle 10|$. The inner product of $|\phi\rangle$ and $|\psi\rangle$ is expressed as $\langle\phi \mid \psi\rangle$ and the norm of $|\phi\rangle$ is thus $\sqrt{\langle\phi \mid \psi\rangle}$. When we observe or measure $|\phi\rangle$ in the computational basis B_n, we obtain each string $s \in \{0,1\}^n$ with probability $|\langle s \mid \phi\rangle|^2$.

Let $\mathcal{H}_\infty = \bigcup_{n\in\mathbb{N}^+} \mathcal{H}_{2^n}$. We extend the "length" notion to arbitrary quantum states in $\mathcal{H}_\infty$. Given each non-null vector $|\phi\rangle$ in $\mathcal{H}_\infty$, the length of $|\phi\rangle$, denoted by $\ell(|\phi\rangle)$, is the minimal number $n \in \mathbb{N}$ satisfying $|\phi\rangle \in \mathcal{H}_{2^n}$; in other words, $\ell(|\phi\rangle)$ is the logarithm of the dimension of the vector $|\phi\rangle$. Conventionally, we further set $\ell(\mathbf{0}) = \ell(\alpha) = 0$ for the null vector $\mathbf{0}$ and any scalar $\alpha \in \mathbb{C}$. By this convention, if $\ell(|\phi\rangle) = 0$ for a quantum state $|\phi\rangle$, then $|\phi\rangle$ must be the qustring of length 0 . A qustring $|\phi\rangle$ of length n is called basic if $|\phi\rangle = |s\rangle$ for a certain binary string s and we often identify such a basic qustring $|s\rangle$ with the corresponding classical binary string s for convenience. Since all basic qustrings in Φ_n form $B_n, \mathcal{H}_{2^n}$ is spanned by all elements in Φ_n.

The partial trace over a system B of a composite system AB, denoted by tr_B, is a quantum operator for which $\mathrm{tr}_B(|\phi\rangle\langle\phi|)$ is a vector obtained by tracing out B from the outer product $|\phi\rangle\langle\phi|$ of a quantum state $|\phi\rangle$. Regarding a quantum state $|\phi\rangle$ of n qubits, we use a handy notation $\mathrm{tr}_k(|\phi\rangle\langle\phi|)$ to mean the quantum state obtained from $|\phi\rangle$ by tracing out all qubits except for the first k qubits.

Ko and Friedman 23 first introduced this notion under the name of "polynomial-time computable." To avoid reader's confusion in this paper, we prefer to use the term "polynomial-time approximation."

For example, it follows for $\sigma_1, \sigma_2, \tau_1, \tau_2 \in \{0,1\}$ that $\mathrm{tr}_1(|\sigma_1\rangle\langle\sigma_2| \otimes |\tau_1\rangle\langle\tau_2|) = |\sigma_1\rangle\langle\sigma_2| \cdot \mathrm{tr}(|\tau_1\rangle\langle\tau_2|)$, where $\mathrm{tr}(B)$ denotes the trace of a matrix B. The trace norm $\|A\|_{\mathrm{tr}}$ of a square matrix A is defined by $\|A\|_{\mathrm{tr}} = \mathrm{tr}\left(\sqrt{AA^\dagger}\right)$. The total variation distance between two ensembles $p = \{p_i\}_{i\in A}$ and $q = \{q_i\}_{i\in A}$ of real numbers over a finite index set A is $\frac{1}{2}\|p-q\|_1 = \frac{1}{2}\sum_{i\in A}||p_i|-|q_i||$.

Throughout this paper, we take special conventions concerning three notations, $|\cdot\rangle, \otimes$, and $\|\cdot\|$, which respectively express quantum states, the tensor product, and the ℓ_2-norm. These conventions slightly deviate from the standard ones used in, e.g., 25, but they make our mathematical descriptions in later sections simpler and more succinct.

Notational Conventions: We freely abbreviate $|\phi\rangle \otimes |\psi\rangle$ as $|\phi\rangle|\psi\rangle$ for any two vectors $|\phi\rangle$ and $|\psi\rangle$. Given two binary strings s and t, $|st\rangle$ means $|s\rangle \otimes |t\rangle$ or $|s\rangle|t\rangle$. Let k and n be two integers with $0 < k < n$. Any qustring $|\phi\rangle$ of length n is expressed in general as $|\phi\rangle = \sum_{s:|s|=k} |s\rangle\,|\phi_s\rangle$, where each $|\phi_s\rangle$ is a qustring of length $n-k$. This qustring $|\phi_s\rangle$ can be viewed as a consequence of applying a partial projective measurement to the first k qubits of $|\phi\rangle$, and therefore it is possible to express $|\phi_s\rangle$ succinctly as $\langle s \mid \phi\rangle$. With this new, convenient notation, $|\phi\rangle$ coincides with $\sum_{s:|s|=k} |s\rangle \otimes \langle s \mid \phi\rangle$, which is simplified as $\sum_{s:|s|=k} |s\rangle\langle s \mid \phi\rangle$. Notice that, when $k = n$, $\langle s \mid \phi\rangle$ is a scalar, say, α in $\mathbb{C}$. Hence, $|s\rangle \otimes \langle s \mid \phi\rangle$ is $|s\rangle \otimes \alpha$ and it is treated as a column vector $\alpha|s\rangle$; similarly, we identify $\alpha \otimes |s\rangle$ with $\alpha|s\rangle$. In these cases, $\otimes$ is treated merely as the scalar multiplication. As a consequence, the equality $|\phi\rangle = \sum_{s:|s|=k} |s\rangle\langle s \mid \phi\rangle$ holds even when $k = n$. Concerning the null vector $\mathbf{0}$, we also take the following special treatment: for any vector $|\phi\rangle \in \mathcal{H}_\infty$, (i) $0 \otimes |\phi\rangle = |\phi\rangle \otimes 0 = \mathbf{0}$, (ii) $|\phi\rangle \otimes \mathbf{0} = \mathbf{0} \otimes |\phi\rangle = \mathbf{0}$, and (iii) when $|\psi\rangle$ is the null vector, $\langle \phi \mid \psi\rangle = \langle \psi \mid \phi\rangle = 0$. Associated with those conventions on the partial projective measurement $\langle \phi \mid \psi\rangle$, we also extend the use of the norm notation $\|\cdot\|$ to scalars. When $\ell(|\phi\rangle) = \ell(|\psi\rangle)$, $\|\langle \phi \mid \psi\rangle\|$ expresses the absolute value $|\langle \phi \mid \psi\rangle|$; more generally, for any number $\alpha \in \mathbb{C}$, $\|\alpha\|$ means $|\alpha|$. With these extra conventions, when $|\phi\rangle$ has the form $\sum_{s:|s|=k} |s\rangle\langle s \mid \phi\rangle$, the equation $\||\phi\rangle\|^2 = \sum_{s:|s|=k} \|\langle s \mid \phi\rangle\|^2$ always holds for any index $k \in [n]$.

2.2 Quantum Turing Machines

We assume the reader's fundamental knowledge on the notion of quantum Turing machine (or QTM) defined in 5. As was done in 36, we allow a QTM to equip multiple tapes and to move its multiple tape heads non-concurrently either to the right or to the left, or to make the tape heads stay still. Such a QTM was also discussed elsewhere (e.g., 27) and is known to polynomially equivalent to the model proposed in [5].

To compute functions from Σ^* to Σ^* over a given alphabet Σ, we generally introduce QTMs as machines equipped with output tapes on which output strings are written in a certain specified way by the time the machines halt. By identifying languages with their characteristic functions, such QTMs can be seen as language acceptors as well.

Formally, a k-tape quantum Turing machine (referred to as k-tape QTM), for $k \in \mathbb{N}^+$, is a sextuple $(Q, \Sigma, \Gamma_1 \times \cdots \times \Gamma_k, \delta, q_0, Q_f)$, where Q is a finite set of inner states including the initial state q_0 and a set Q_f of final states with $Q_f \subseteq Q$, each Γ_i is an alphabet used for tape i with a distinguished blank symbol $\#$ satisfying $\Sigma \subseteq \Gamma_1$, and δ is a quantum transition function from $Q \times \tilde{\Gamma}^{(k)} \times Q \times \tilde{\Gamma}^{(k)} \times \{L, N, R\}^k$ to $\mathbb{C}$, where $\tilde{\Gamma}^{(k)}$ stands for $\Gamma_1 \times \cdots \times \Gamma_k$. For convenience, we identify L, N, and R with $-1, 0$, and $+1$, respectively, and we set $D = \{0, \pm 1\}$. For more information, refer to [36].

All tape cells of each tape are indexed sequentially by integers. The cell indexed 0 on each tape is called the start cell. At the beginning of the computation, M is in inner state q_0, all the tapes except for the input tape are blank, and all tape heads are scanning the start cells. A given input string $x_1 x_2 \cdots x_n$ is initially written on the input tape in such a way that, for each index $i \in [n], x_i$ is in cell i (not cell $i - 1$). When M enters a final state, an output of M is the content of the string written on an output tape (if M has only a single tape, then an output tape is the same as the tape used to hold inputs) from the start cell, stretching to the right until the first blank symbol. A configuration of M is expressed as a triplet $\left(p, (h_i)_{i\in[k]}, (z_i)_{i\in[k]}\right)$, which indicates that M is currently in inner state p having k tape heads at cells indexed by $h_1, \ldots, h_k$ with tape contents $z_1, \ldots, z_k$, respectively. The notion of configuration will be slightly modified in Sections 45 to make the proof of our main theorem simpler. An initial configuration is of the form ($q_0, 0, x$) and a final configuration is a configuration having a final state. The configuration space is spanned by the basis vectors in $\left\{|q, h, z\rangle \mid q \in Q, h \in \mathbb{Z}^k, z \in \Gamma_1^* \times \cdots \Gamma_k^*\right\}$. For a nonempty string z_i and an index $h \in [|z_i|], z_i[h]$ denotes the h th symbol in z_i. For example, if $z_i = 01101$, then $z_i[1] = 0, z_i[2] = 1$, and $z_i[5] = 1$. The time-evolution operator U_δ of M acting on the configuration space is induced from δ as

$$U_\delta |p, h, z\rangle = \sum_{q,w,d} \delta\left(p, z_h, q, z'_h, d\right) |q, h_d, z'\rangle ,$$

where $p \in Q$, $h = (h_i)_{i\in[k]} \in \mathbb{Z}^k, z = (z_i)_{i\in[k]} \in \Gamma_1^* \times \cdots \times \Gamma_k^*, z_h = \left(z_i[h_i]\right)_{i\in[k]}, h_d = (h_i + d_i)_{i\in[k]}$, and $z' = (z'_i)_{i\in[k]}$, where each z'_i is the same as z except for the h_i-th symbol. Moreover, in the summation, variables $q, z'_h = \left(z'_i[h_i]\right)_{i\in[k]}$, and $d = (d_i)_{i\in[k]}$ respectively range over $Q, \tilde{\Gamma}^{(k)}$, and D^k. Any entry of U_δ is called an amplitude. Quantum mechanics demand the time-evolution operator U_δ of the QTM to be unitary.

Each step of M consists of two phases: first apply δ and then take a partial projective measurement, in which we check whether M is in a final state (i.e., an inner state in Q_f). Formally, a computation of M on input x is a series of superpositions of configurations produced by sequential applications of U_δ, starting from an initial configuration of M on x. If M enters a final state along a computation path, this computation path terminates; otherwise, its computation must continue.

A k-tape QTM $M = \left(Q, \Sigma, \tilde{\Gamma}^{(k)}, \delta, q_0, Q_f\right)$ is well-formed if δ satisfies three

local conditions: unit length, separability, and orthogonality. To explain these conditions, as presented in [36, Lemma 1], we first introduce the following notations. For our convenience, we set $E = \{0, \pm 1, \pm 2\}$ and $H = \{0, \pm 1, \natural\}$. Given elements $(p, \sigma, \tau) \in Q \times \left(\tilde{\Gamma}^{(k)}\right)^2, \epsilon = (\varepsilon_i)_{i\in[k]} \in E^k$, and $d = (d_i)_{i\in[k]} \in D^k$, we define $D_\epsilon = \left\{d \in D^k \mid \forall i \in [k] \left(|2d_i - \varepsilon_i| \leq 1\right)\right\}$ and $E_d = \left\{\varepsilon \in E^k \mid d \in D_\epsilon\right\}$. Moreover, let $h_{d,\epsilon} = \left(h_{d_i,\varepsilon_i}\right)_{i\in[k]}$, where $h_{d_i,\varepsilon_i} = 2d_i - \varepsilon_i$ if $\varepsilon_i \neq 0$ and $h_{d_i,\varepsilon_i} = \natural$ otherwise. Finally, we define $\delta(p, \sigma) = \sum_{q,\tau,d} \delta(p, \sigma, q, \tau, d)|q, \tau, d\rangle$ and $\delta[p, \sigma, \tau \mid \epsilon] = \sum_{q\in Q} \sum_{d\in D_\epsilon} \delta(p, \sigma, q, \tau, d) \left|E_d\right|^{-1/2} |q\rangle \left|h_{d,\epsilon}\right\rangle$, where $\sigma, \tau \in \tilde{\Gamma}^{(k)}$ and $d \in D^k$.

1. (unit length) $\|\delta(p, \sigma)\| = 1$ for all $(p, \sigma) \in Q \times \tilde{\Gamma}^{(k)}$.
2. (orthogonality) $\delta\left(p_1, \sigma_1\right) \cdot \delta\left(p_2, \sigma_2\right) = 0$ for any distinct pair $\left(p_1, \sigma_1\right), \left(p_2, \sigma_2\right) \in Q \times \tilde{\Gamma}^{(k)}$.
3. (separability) $\delta\left[p_1, \sigma_1, \tau_1 \mid \epsilon\right] \cdot \delta\left[p_2, \sigma_2, \tau_2 \mid \epsilon'\right] = 0$ for any distinct pair $\epsilon, \epsilon' \in E^k$ and for any pair $\left(p_1, \sigma_1, \tau_1\right), \left(p_2, \sigma_2, \tau_2\right) \in Q \times \left(\tilde{\Gamma}^{(k)}\right)^2$.
   The well-formedness of a QTM captures the unitarity of its time-evaluation operator.
   Lemma 2.1 (Well-Formedness Lemma of [36]) A k-tape QTM M with a transition function δ is wellformed iff the time-evolution operator of M preserves the ℓ_2-norm.

Given a nonempty subset K of $\mathbb{C}$, we say that a QTM is of K-amplitude if all values of its quantum transition function belong to K. It is of significant importance to limit the choice of amplitude within an appropriate set K of reasonable numbers. With the use of such a set K, we introduce two important complexity classes BQP_K and FBQP_K.

Definition 2.2 Let K be any nonempty subset of $\mathbb{C}$ and let Σ be any alphabet.

1. A subset S of Σ^* is in BQP_K if there exists a multi-tape, polynomial-time, well-formed QTM M with K-amplitudes such that, for every string x, M outputs $S(x)$ with probability at least 2/3[5].
2. A single-valued function f from Σ^* to Σ^* is called bounded-error quantum polynomial-time computable if there exists a multi-tape, polynomial-time, well-formed QTM M with K-amplitudes such that, on every input x, M outputs $f(x)$ with probability at least 2/3. Let FBQP_K denote the set of all such functions 38.

The use of arbitrary complex amplitudes turns out to make BQP_K quite powerful. As Adleman, DeMarrais, and Huang 1 demonstrated, $\mathrm{BQP}_{\mathbb{C}}$ contains all possible languages, and thus $\mathrm{BQP}_{\mathbb{C}}$ is no longer recursive. Therefore, we usually pay our attention only to polynomial-time approximable amplitudes and, for this reason, when $K = \tilde{\mathbb{C}}$, we always drop subscript K and briefly write BQP and FBQP instead of BQP_K and FBQP_K, respectively. It is also possible to further limit the amplitude set K to $\left\{0, \pm 1, \pm\frac{3}{5}, \pm\frac{4}{5}\right\}$ because $\mathrm{BQP} = \mathrm{BQP}_{\left\{0,\pm 1,\pm\frac{3}{5},\pm\frac{4}{5}\right\}}$ holds 1 .

2.3 Quantum Circuits

A k-qubit quantum gate, for $k \in \mathbb{N}^+$, is a unitary operator acting on a Hilbert space of dimension 2^k. Since any quantum state is a vector in a certain Hilbert space, each entry of such a quantum state is customarily called an amplitude. Unitary operators, such as the Walsh-Hadamard transform (WH) and the controlled-NOT
transform (CNOT) defined as

$$WH = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ and } CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

are typical quantum gates acting on 1 qubit and 2 qubits, respectively. If a quantum gate U acting on k qubits is applied to a k-qubit quantum state $|\phi\rangle$, then we obtain a new quantum state $U|\phi\rangle$. Notice that every quantum gate preserves the norm of any quantum state given as an input. A quantum circuit is a product of a finite number of layers, where each layer is a Kronecker product of allowed quantum gates. We often concentrate on a particular set of quantum gates to construct quantum circuits. Let us consider the specific quantum gates: the CNOT gate and three one-qubit gates of the form

$$Z_{1,\theta} = \begin{pmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{pmatrix}, \quad Z_{2,\theta} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}, \quad \text{and } R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

where θ is a real number with $0 \leq \theta \leq 2\pi$. Notice that WH equals $R_{\frac{\pi}{4}}$. Those gates form a universal set of quantum gates [2] since WH and $Z_{2,\frac{\pi}{4}}$ (called the $\pi/8$ gate) can approximate any single qubit unitary operator to arbitrary accuracy. For convenience, we call them elementary gates. The set of CNOT, WH, and $Z_{2,\frac{\pi}{4}}$ is also known to be universal 6.

Given an amplitude set K, a quantum circuit C is said to have K-amplitudes if all entries of each quantum gate used inside C are drawn from K. For any k-qubit quantum gate and any integer $n > k, G^{(n)}$ denotes $G \otimes I^{\otimes n-k}$, the n-qubit expansion of G. An n-qubit quantum circuit is formally defined as a finite sequence $(G_m, \pi_m), (G_{m-1}, \pi_{m-1}), \cdots, (G_1, \pi_1)$ such that each G_i is an n_i-qubit quantum gate with $n_i \leq n$ and π_i is a permutation on $\{1, 2, \ldots, n\}$. This quantum circuit represents the unitary operator $U = U_m U_{m-1} \cdots U_1$, where U_i is of the form $V_{\pi_i}^\dagger G_i^{(n)} V_{\pi_i}$ and $V_{\pi_i}(|x_1 \cdots x_n\rangle) = \left|x_{\pi_i(1)} \cdots x_{\pi_i(n)}\right\rangle$ for each $i \in [m]$. The size of a quantum circuit is the total number of quantum gates in it. Yao 40 and later Nishimura and Ozawa 26 showed that, for any k-tape QTM and a polynomial p, there exists a family of quantum circuits of size $O\left(p(n)^{k+1}\right)$ that exactly simulates M.

A family $\{C_n\}_{n\in\mathbb{N}}$ of a quantum circuit is said to be P -uniform if there exists a deterministic (classical) Turing machine that, on input 1^n, produces a code of C_n in time polynomial in the size of C_n, provided that we use a fixed, efficient coding scheme to describe each quantum circuit.

Proposition 2.3 [40 (see also [26]) For any language L over an alphabet $\{0,1\}, L$ is in BQP iff there exist a polynomial p and a P -uniform family $\{C_n\}_{n\in\mathbb{N}}$ of quantum circuits having $\tilde{\mathbb{C}}$-amplitudes such that $\left\|\langle L(x)|C_{|x|}\left|x10^{p(|x|)}\right\rangle\right\|^2 \geq \frac{2}{3}$ holds for all $x \in \{0,1\}^*$, where L is seen as the characteristic function of L.

Yao's inspiring proof 40 of Proposition 2.3 gives a foundation to our proof of Lemma 4.3 , which provides in Section 3.1 a simulation of a well-formed QTM by an appropriately chosen $\Box_1^{\mathrm{QP}}$ -function.

3 A New, Simple Schematic Definition

As noted in Section 1 , the "schematic" definition of recursive function means an inductive (or constructive) way of defining the set of computable functions and it involves a small set of so-called initial functions as well as a small set of construction rules, which are sequentially applied finitely many times to build more complex functions from certain functions that have been already constructed. A similar schematic characterization is known for polynomial-time computable functions (as well as languages) [7, 8, 9, 24, 35. Along this line of work, we wish to present a new, simple schematic definition composed of a small set of initial quantum functions and a small set of construction rules, intending to make this schematic definition appropriately capture polynomial-time computable quantum functions, where a quantum function is a function mapping $\mathcal{H}_\infty$ to $\mathcal{H}_\infty$. As remarked briefly in Section 1, it is important to note that our term of "quantum function" is quite different from the one used in, e.g., [38], in which "quantum function" refers to functions that take classical input strings and produce either classical output strings or acceptance probabilities of multi-tape polynomial-time well-formed QTMs and thus it maps Σ^* to either Σ^* or the real unit interval $[0,1]$, where Σ is an appropriate alphabet.