

# Nonuniform Families of Polynomial-Size Quantum Finite Automata and Quantum Logarithmic-Space Computation with Polynomial-Size Advice<sup>1</sup>

TOMOYUKI YAMAKAMI<sup>2</sup>

## Abstract

The state complexity of a finite(-state) automaton intuitively measures the size of the description of the automaton. Sakoda and Sipser [STOC 1972, pp. 275–286] were concerned with nonuniform families of finite automata and they discussed the behaviors of the nonuniform complexity classes defined by such families of finite automata having polynomial-size state complexity. In a similar fashion, we introduce nonuniform state complexity classes using nonuniform families of quantum finite automata empowered by the flexible use of garbage tapes. We first present general inclusion and separation relationships among nonuniform state complexity classes of various one-way finite automata, including deterministic, nondeterministic, probabilistic, and quantum finite automata having polynomially many inner states. For two-way quantum finite automata equipped with flexible garbage tapes, we show a close relationship between the nonuniform state complexity of the family of such polynomial-size quantum finite automata and the parameterized complexity class induced by logarithmic-space quantum computation assisted by polynomial-size advice. We further establish a direct connection between space-bounded quantum computation with quantum advice and quantum finite automata whose transitions are dictated by superpositions of transition tables.

**keywords:** nonuniform state complexity, quantum finite automata, flexible garbage tape, quantum Turing machine, quantum advice, super quantum finite automata

## 1 Prelude: Quick Overview

This exposition reports a collection of fundamental results obtained by our initial study on the state complexity of nonuniform families of various finite automata. Such a complexity measure is briefly referred to as the *nonuniform state complexity* throughout this exposition.

### 1.1 Nonuniform State Complexity of Finite Automata Families

Each finite(-state) automaton is completely described in terms of a set of transitions of its inner states because there is no memory device, such as a stack, a work tape, etc., other than a read-only input tape. The number of such inner states is thus crucial to measure the “descriptive size” of the automaton in question and it works as a general complexity measure, known as the *state complexity* of the automaton. This state complexity therefore naturally serves as a clear indicator to scale the computational power of the automaton. Instead of taking a single automaton, in this exposition, we consider a “family” of finite automata in a way similar to taking a family of Boolean circuits. Such a family of finite automata may or may not be generated individually by a certain fixed deterministic procedure in a certain uniform manner. Unlike Boolean circuits, nevertheless, inputs of automata are *not limited* to fixed lengths and this situation provides an additional consideration to the simulation of automata in general. In this exposition, for clarity, we intend to use the term “uniform state complexity” to refer to the state complexity of a uniform family of finite automata. Opposed to this uniform state complexity, we attempt to study its “nonuniform” counterpart, which we intend to call *nonuniform state complexity*. This nonuniform complexity measure has turned out to be closely related to a nonuniform model of Turing-machine computation.

In the past literature, nonuniform state complexity has played various roles in automata theory. An early discussion that attempted to relate certain state complexity issues to the collapses of known space-bounded complexity classes dates back to late 1970s. Sakoda and Sipser [32], following Berman and Lingas [5], argued the state complexity of transforming one family of 2-way nondeterministic finite automata (or

<sup>1</sup>A preliminary version appeared in the Proceedings of the 13th International Conference on Language and Automata Theory and Applications (LATA 2019), Saint Petersburg, Russia, March 26–29, 2019, Lecture Notes in Computer Science, Springer, vol. 11417, pp. 134–145, 2019. This current paper corrects and extends the preliminary version.

<sup>2</sup>Present Affiliation: Faculty of Engineering, University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan

2nfa's, for short) into another family of 2-way deterministic finite automata (or 2dfa's). From their work, we have come to know that the state complexity of a family of automata is closely related to the work-tape space usage of a Turing machine. In this line of study, after a long recess, Kapoutsis [21] and Kapoutsis and Pighizzini [22] lately revitalized a discussion on the relationships between logarithmic-space (or log-space, for short) complexity classes and state complexity classes in connection to the  $L = NL$  question (in fact, the  $NL \subseteq L/\text{poly}$  question, where  $L/\text{poly}$  is the nonuniform analogue of  $L$ ).

Taking a complexity-theoretic approach, Kapoutsis [19, 20] earlier studied relationships among the nonuniform state complexity classes 1D (one-way deterministic), 1N (one-way nondeterministic), 2D (two-way deterministic), and 2N (two-way nondeterministic). These classes constitute families of “promise” decision problems, each family of which is solved by an appropriately chosen nonuniform family of deterministic or nondeterministic finite automata of polynomially many inner states. In contrast, two more nonuniform state complexity classes  $2^{1D}$  and  $2^{1N}$  were introduced by the use of finite automata having exponentially many inner states (see Section 2.6 for their precise definitions). There are only a few known separations on those nonuniform state complexity classes. For instance, Kapoutsis [19, 20] demonstrated that  $1D \subsetneq 1N \subsetneq 2^{1D}$ ,  $1N \neq \text{co-1N}$ , and  $1D \subsetneq 2D \subseteq 2N \subsetneq 2^{1D}$ . Along the same line of study, Yamakami [52] recently gave a precise characterization of the polynomial-time sub-linear-space parameterized complexity class PsubLIN and an NL-complete problem 3DSTCON parameterized by the number of vertices of an input graph (which is generally referred to as a *size parameter*) in terms of the uniform state complexities of restricted 2nfa's and narrow 2-way alternating finite automata (abbreviated as 2afa's).

An important discovery of [52], nevertheless, is the fact that a nonuniform family of promise decision problems is more closely related to parameterized decision problems than to “standard” decision problems (whose complexities are measured by the “binary” encoding size of inputs). A decision problem (identified freely with a language)  $L$  over alphabet  $\Sigma$  together with a reasonable size parameter  $m$ , which is a map from  $\Sigma^*$  to the set  $\mathbb{N}$  of all natural numbers, forms a *parameterized decision problem*  $(L, m)$  [50]. We can naturally translate such a parameterized decision problem  $(L, m)$  into a uniform family  $\{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems satisfying  $L_n^{(+)} \cup L_n^{(-)} = \Sigma_n$ , where  $\Sigma_n$  is a set of input strings having “size”  $n$ , and we can also translate  $\{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  back into another parameterized decision problem  $(K, m')$ , which is “almost” the same as  $(L, m)$ . These translations between parameterized decision problems and families of promise decision problems play an essential role in this exposition. For notational readability, following [53], we use the special prefix “para-” and write, for example, para-L and para-NL to denote respectively the parameterized analogues of L and NL. See Section 2.5 for the precise definitions and descriptions of the aforementioned concepts.

After the study of state complexity classes was initiated in [32], a further elaboration has been long anticipated; however, there has been little research on how to expand the scope of these classes. Our primary purpose of this exposition is to enrich the world of nonuniform state complexity classes and to lead our study to a whole new direction.

## 1.2 An Extension to Quantum Finite Automata

A surge of papers has already dealt with “standard” state complexity issues in an emerging field of *quantum finite automata* [2, 8, 40, 57, 55]. We intend to further expand the scope of *nonuniform state complexity theory* to this new field. The behaviors of quantum finite automata, viewed as a natural extension of probabilistic finite automata, are fundamentally governed by *quantum physics*. Moore and Crutchfield [28] and Kondacs and Watrous [27] modeled the quantization of finite automata in two quite different ways. Lately, these definitions have been considered insufficient for implementation and advantages over classical finite automata and, for this reason, numerous generalizations have been proposed (see, e.g., a survey [4] for references therein). Here, we intend to use two distinct models: *measure-many 1-way<sup>3</sup> quantum finite automata with garbage tapes* (or 1qfa's, for short) and *measure-many 2-way quantum finite automata with garbage tapes* (or 2qfa's), where garbage tapes are write-once<sup>4</sup> tapes used to discard unwanted information, which is thought to be released into an external environment surrounding the target quantum finite automata. For early use of extra tape tracks to discard the unnecessary

<sup>3</sup>We use this term “1-way” *in a strict sense* that a tape head always moves to the right and is not allowed at any time to stay still on the same cell. This term is also called “real time” in certain literature.

<sup>4</sup>A tape is *write-once* if its tape head never moves to the left and, whenever it writes a non-blank symbol, it must move to the right.



paths of these machines. From a concern for execution efficiency, it is reasonable to concentrate on 2qfa's that run in *expected polynomial time* (or *average polynomial time*) rather than 2qfa's with no time bound. We say that a family  $\{M_n\}_{n \in \mathbb{N}}$  of 2qfa's runs in expected polynomial time if the average runtime of  $M_n$  is restricted to a certain fixed polynomial in  $(n, |x|)$ , where  $x$  is any input given to  $M_n$ . To emphasize the expected polynomial runtime, we append the prefix "ptime-" as in ptime-2BQ and ptime-2BQ/poly.

To introduce the nonuniformity notion into a model of quantum Turing machine (or QTM, for short), we equip such a QTM with the Karp-Lipton style *advice* as supplemental external information to empower the QTM (see, e.g., [30]). We use the notation BQL/poly to express the complexity class composed of decision problems that are solvable with bounded-error probability by QTMs using (worst-case) logarithmic space. For restricted QTMs running in expected polynomial time, we also emphasize this runtime bound with the prefix "ptime-" and write ptime-BPL and ptime-BQL instead of BPL and BQL.

We wish to present in Theorem 1.1 a close connection between advised complexity classes and nonuniform state complexity classes. By our convention of the runtime of (non)deterministic automata and Turing machines, ptime-2D, ptime-2N, ptime-L, and ptime-NL appearing in the theorem should be understood respectively as 2D, 2N, L, and NL. Concerning quantum and probabilistic computations, for technical reason, we slightly restrict 2BQ and 2BP and write  $2BQ^\dagger$  and  $2BP^\dagger$  for these restrictions (see Section 5 for the reasoning).

**Theorem 1.1** *Let  $(\mathcal{A}, \mathcal{B}) \in \{(N, D), (N, BP^\dagger), (N, BQ^\dagger), (BQ^\dagger, BP^\dagger)\}$ . It then follows that  $\text{ptime-2}\mathcal{A}/\text{poly} \subseteq \text{ptime-2}\mathcal{B}$  iff  $\text{ptime-}\mathcal{A}L \subseteq \text{ptime-}\mathcal{B}L/\text{poly}$ , where, when  $\mathcal{A} = D$ , "DL" is understood as "L".*

**Corollary 1.2** 1.  $2N/\text{poly} \subseteq \text{ptime-2}BQ^\dagger$  iff  $NL \subseteq \text{ptime-BQL}/\text{poly}$ .  
2.  $\text{ptime-2}BQ^\dagger/\text{poly} \subseteq \text{ptime-2}BP^\dagger$  iff  $\text{ptime-BQL} \subseteq \text{ptime-BPL}/\text{poly}$ .

Theorem 1.1 follows from the exact characterizations (Proposition 5.1) of parameterized complexity classes in terms of nonuniform state complexity classes, and vice versa. This proposition helps us translate nonuniform state complexity classes, such as 2D/poly, 2N/poly, ptime-2BP/poly, and ptime-2BQ/poly, respectively into their corresponding advised parameterized complexity classes, para-L/poly, para-NL/poly, para-ptime-BPL/poly, and para-ptime-BQL/poly, where the last class para-ptime-BQL/poly, for example, is the collection of parameterized decision problems  $(L, m)$  solvable by bounded-error QTMs in expected polynomial time in  $|x|m(x)$  using work tapes of space logarithmic in  $m(x)$  with (deterministic) advice of size polynomial in  $|x|m(x)$  (see Section 2.5 for their precise definitions).

The third main issue concerns quantum advice and an extension of 2qfa's. The notion of *quantum advice* was introduced by Nishimura and Yamakami [30] to enhance the computational capability of polynomial-time QTMs. Quantum advice manifests a quantization of *deterministic advice* and *randomized advice* (see, e.g., [46, 48]). To emphasize the use of quantum advice, we write BQL/Qpoly in accordance with [30].

In parallel to the change of deterministic advice to quantum advice, we also modify our basic model of 2qfa's as follows. Firstly, we express a quantum transition function as the form of a matrix or a table, which must be easily encoded into a string over a certain alphabet. For readability, we use the term "transition table" to address this encoded string. See Section 2.3 for the precise definition. This encoding further makes it possible to consider a superposition of transition tables. Generally, we call by a *super quantum finite automaton* a quantum finite automaton obtained by substituting a superposition of transition tables for the original quantum transition function.

For convenience, we use the notation 2sBQ to express the nonuniform state complexity class obtained from 2BQ by substituting super 2qfa's for ordinary 2qfa's and write  $2sBQ^\dagger$  for its restricted version similarly to  $2BQ^\dagger$ .

**Theorem 1.3**  $\text{ptime-2s}BQ^\dagger/\text{poly} \subseteq \text{ptime-2}BQ^\dagger$  iff  $\text{ptime-BQL}/\text{poly} = \text{ptime-BQL}/\text{Qpoly}$ .

A further study on relativizations (or Turing reducibility) was lately conducted in [53, 54].

## 2 Preparation: Basic Notions and Notation

This section will provide basic notions and notation necessary to read through this exposition.

### 2.1 Numbers, Languages, and Hilbert Spaces

Let  $\mathbb{N}$ ,  $\mathbb{Z}$ , and  $\mathbb{C}$  denote respectively the sets of all *natural numbers* (i.e., nonnegative integers), of all *integers*, and of all *complex numbers*. In particular, we set  $\mathbb{N}^+ = \mathbb{N} - \{0\}$ , which is the set of all positive integers. Given two integers  $m$  and  $n$  with  $n \geq m$ ,  $[m, n]_{\mathbb{Z}}$  denotes the *integer interval*, which is the set  $\{m, m+1, m+2, \dots, n\}$ . For a given number  $n \in \mathbb{N}^+$ , we abbreviate  $[1, n]_{\mathbb{Z}}$  as  $[n]$ . For a finite set  $S$ ,  $|S|$  indicates the *cardinality* of  $S$ . All *polynomials* in this exposition are assumed to have nonnegative integer coefficients, and thus the polynomials are all nondecreasing. Assume also that all *logarithms* are taken to the base 2. For our convenience, we set  $\log 0$  to be 0. A *logarithmic function* is a mapping from  $\mathbb{N}$  to  $\mathbb{N}$  for which there exist two constants  $c, d \geq 0$  satisfying  $f(n) \leq c \log n + d$  for any  $n \in \mathbb{N}$ .

Let  $\Sigma$  be any *alphabet*, which is a finite nonempty set of “symbols” or “letters.” A *string* over  $\Sigma$  is a finite sequence of symbols in  $\Sigma$ . The *length* of any string  $x$  is denoted  $|x|$ . In particular, we use the notation  $\lambda$  to denote the *empty string* of length 0. Given a number  $n \in \mathbb{N}$ ,  $\Sigma^n$  (resp.,  $\Sigma^{\leq n}$ ) indicates the set of all strings of length exactly  $n$  (resp., of length at most  $n$ ) over  $\Sigma$ . Let  $\Sigma^* = \bigcup_{n \in \mathbb{N}} \Sigma^n$ . For a string  $w$  and a number  $i \in [|x|]$ ,  $w_{(i)}$  denotes the  $i$ th symbol of  $w$ ; for example,  $0110_{(1)} = 0$  and  $0110_{(3)} = 1$ . A subset of  $\Sigma^*$  is called a *language* over  $\Sigma$ . The *complement* of  $L$  is the set  $\Sigma^* - L$  and is succinctly denoted  $\bar{L}$ . We freely identify a language  $L$  with its *characteristic function*; that is,  $L(x) = 1$  for all  $x \in L$  and  $L(x) = 0$  for all  $x \notin L$ . Given a size-bounding function  $t : \mathbb{N} \rightarrow \mathbb{N}$ , a function  $h : \mathbb{N} \rightarrow \Sigma^*$  is called  $t(n)$ -*bounded* if  $|h(n)| \leq t(n)$  holds for all  $n \in \mathbb{N}$ .

To describe quantum computation, we use various Hilbert spaces. A *Hilbert space* is a complex inner product space (i.e., a vector space with an inner product), which is also a complete metric space. To express each element of such a space, we use Dirac’s notation of  $|\cdot\rangle$ . The *dual* of  $|\phi\rangle$  is denoted  $\langle\phi|$ . The *norm* of  $|\phi\rangle$  is denoted  $\| |\phi\rangle \|$ , which equals  $\sqrt{|\langle\phi|\phi\rangle|}$ , where  $\langle\phi|\psi\rangle$  is the *inner product* of  $|\phi\rangle$  and  $|\psi\rangle$ .

### 2.2 Computational Models of Finite Automata

Our finite automata are always equipped with *read-only input tapes*, which use two endmarkers,  $\text{\textcircled{c}}$  (left endmarker) and  $\text{\textcircled{d}}$  (right endmarker), where a given input string is written initially between the two endmarkers. A *stationary move* refers to a finite automaton’s special move by which an input-tape head reads an input symbol but never moves to a neighboring cell. For clarity reason, we use the term “one way” to refer to the condition of a tape head that always moves to the right without stopping, and therefore there is no stationary move. On the contrary, if we allow a machine to make “stationary moves,” we instead use the term “1.5 way” to emphasize its difference from “one way” machines. For brevity, we generally abbreviate “1-way deterministic finite automaton” as “1dfa”, “1-way nondeterministic finite automaton” as “1nfa”, and “1-way probabilistic finite automaton” as “1pfa”. Analogously, for models of “two way” tape head moves, we use similar abbreviations, such as 2dfa, 2nfa, and 2pfa.

We assume the reader’s familiarity with the basics of quantum computation (see, e.g., [17, 29]). Since Kondacs and Watrous’s model of 1qfa’s [27] is strictly weaker in power than 1dfa’s, there have been numerous generalizations proposed in the past literature (see, e.g., a survey [4]). We here empower their 1qfa’s by simply equipping each of them with a *write-once garbage tape* in which a machine drops any symbol (called a *garbage symbol*) but never accesses any non-blank symbol written already on the garbage tape again. An early idea of 1qfa’s discarding unwanted information down to a portion of a read-once input tape was materialized in [48, Section 5.2] and such a mechanism was shown to enhance the computational power of 1qfa’s. Yakaryilmaz, Freivalds, Say, and Agadzanyan [39] also discussed a similar concept in terms of write-only memory. The use of a garbage tape makes 1qfa’s simulate all 1dfa’s simply by discarding the current inner state of the 1dfa’s onto the garbage tape. The garbage tape can be viewed as a surrounding environment that exists “externally,” separated from the essential part of a machine’s computation. Observing the inner state and also the content of the garbage tape at every step produces a mixed state of  $M$ ’s configurations, and therefore, the use of the garbage tape can simulate the behavior of superoperators acting on mixed states. Our model thus turns out to be at least as powerful as the model of *generalized one-way quantum finite automata* given in [18, 42], which allow 1qfa’s to use mixed states and superoperators. Recall that input tapes of finite automata always

have the left endmarker  $\clubsuit$  and the right endmarker  $\$$ . All tape cells are indexed by numbers in  $\mathbb{N}$ ; in particular,  $\clubsuit$  is always placed in cell 0 (which is also called the *start cell*). For the sake of convenience, we use *circular input tapes*, where a circular tape is a standard tape whose ends are both glued together so that the right-side of the right endmarker  $\$$  is the left endmarker  $\clubsuit$ .

Formally, a *1-way quantum finite automaton with a garbage tape* (where we hereafter use the term “1qfa” to indicate this particular model unless otherwise stated)  $M$  is an octuple<sup>5</sup>  $(Q, \Sigma, \{\clubsuit, \$\}, \Xi, \delta, q_0, Q_{acc}, Q_{rej})$ , where  $Q$  is a finite set of inner states,  $\Sigma$  is an input alphabet,  $\Xi$  is a *garbage alphabet*,  $\delta$  is a (quantum) transition function mapping  $Q \times \tilde{\Sigma} \times Q \times \Xi_\lambda$  to  $\mathbb{C}$ ,  $q_0 (\in Q)$  is the initial (inner) state, and  $Q_{acc}$  is a set of *accepting states* with  $Q_{acc} \subseteq Q$ , and  $Q_{rej}$  is a set of *rejecting states* with  $Q_{rej} \subseteq Q$ , where  $\tilde{\Sigma} = \Sigma \cup \{\clubsuit, \$\}$  and  $\Xi_\lambda = \Xi \cup \{\lambda\}$ . For clarity reason, we write  $\delta(q, \sigma|p, \xi)$  instead of  $\delta(q, \sigma, p, \xi)$ . A transition  $\delta(q, \sigma|p, \xi) = \alpha$  indicates that, assuming that  $M$  is in inner state  $q$  scanning input symbol  $\sigma$ , in a single step with *transition amplitude*  $\alpha$ ,  $M$  changes its inner state to  $p$ , moves the input-tape head to the right, writes down garbage symbol  $\xi$  onto the garbage tape (where  $\lambda$  means “no writing”), and moves the garbage-tape head to the right if  $\xi \neq \lambda$ . At any step,  $M$  freely chooses whether or not it dumps garbage symbols onto its garbage tape and then moves its tape head to the right. When  $M$  does not write any garbage symbol, the garbage-tape head must stay still. For later convenience, this property is referred to as the *flexible use of a garbage tape* or, more succinctly, a *flexible garbage tape*. Let  $Q_{halt} = Q_{acc} \cup Q_{rej}$ . All inner states in  $Q_{halt}$  are called *halting states* and the rest of inner states are *non-halting states*.

A *configuration* of a 1qfa  $M$  on input  $x$  is a quadruple  $(q, x, i, z)$ , where  $q \in Q$ ,  $x \in \Sigma^*$ ,  $i \in [0, |x| + 1]_{\mathbb{Z}}$ , and  $z \in \Xi^*$ . This describes a “snapshot” of the machine’s internal status at a certain moment, where  $M$  is in inner state  $q$ , its input-tape head is located at the  $i$ th tape cell, and the garbage tape contains  $z$ . The *initial configuration* of  $M$  on  $x$  is  $(q_0, x, 0, \lambda)$  and an *accepting configuration* (resp., a *rejecting configuration*) of  $M$  on  $x$  is of the form  $(q, x, i, z)$  with  $q \in Q_{acc}$  (resp.,  $q \in Q_{rej}$ ). A *halting configuration* is either an accepting configuration or a rejecting configuration. Let  $\mathcal{H}_{conf}$  denote the Hilbert space spanned by all configurations of  $M$ . Similarly,  $\mathcal{H}_{acc}$  and  $\mathcal{H}_{rej}$  are Hilbert spaces spanned respectively by all accepting configurations and all rejecting configurations of  $M$ . For convenience, two more Hilbert spaces,  $\mathcal{H}_{halt}$  and  $\mathcal{H}_{non}$ , are defined respectively in terms of halting configurations and non-halting configurations. The (quantum) transition function  $\delta$  naturally induces the *time-evolution operator*  $U_\delta$  of  $M$ , which is a linear operator acting on the space  $\mathcal{H}_{conf}$ , defined as:

$$U_\delta |q, x, i, z\rangle = \sum_{(p, \xi)} \delta(q, x_{(i)}|p, \xi) |p, x, i + 1 \pmod{|x| + 2}, z\xi\rangle,$$

where  $(p, \xi)$  ranges over  $Q \times \Xi_\lambda$ , each  $x_{(i)}$  is the  $(i + 1)$ th symbol of  $\clubsuit x \$$  with  $x_{(0)} = \clubsuit$  and  $x_{(|x|+1)} = \$$ .

For any subscript  $\tau \in \{acc, rej, halt, non\}$ ,  $\Pi_\tau$  denotes the *projective measurement* onto the Hilbert space  $\mathcal{H}_\tau$ . At each step, we first apply  $U_\delta$  and then perform a projective measurement by applying  $\Pi_{acc} \oplus \Pi_{rej}$ . A *computation* of  $M$  on input  $x$  is a series  $(|\phi_0\rangle, |\phi_1\rangle, \dots, |\phi_{|x|+2}\rangle)$  of superpositions of configurations of  $M$  on  $x$  defined as  $|\phi_0\rangle = |q_0, x, 0, \lambda\rangle$  and  $|\phi_{i+1}\rangle = \Pi_{non} U_\delta |\phi_i\rangle$  for each index  $i \in [0, |x| + 1]_{\mathbb{Z}}$ . At Step  $i \in [1, |x| + 2]_{\mathbb{Z}}$ , we say that  $M$  *accepts* (resp., *rejects*)  $x$  with probability  $p_{acc}(x, i) = \|\Pi_{acc} U_\delta |\phi_{i-1}\rangle\|^2$  (resp.,  $p_{rej}(x, i) = \|\Pi_{rej} U_\delta |\phi_{i-1}\rangle\|^2$ ). The *(cumulative) acceptance probability*  $p_{acc}(x)$  of  $M$  on  $x$  is the sum  $\sum_{i=1}^{|x|+2} p_{acc}(x, i)$  and the *(cumulative) rejection probability* of  $M$  on  $x$  is  $\sum_{i=1}^{|x|+2} p_{rej}(x, i)$ .

When  $x$  is fixed through our analysis of quantum computation, we often remove  $x$  from the configurations and instead consider *surface configurations*  $(q, i, z)$ . In place of  $U_\delta$ , we use the notation  $U_\delta^{(x)}$ , which acts on the Hilbert space spanned by all surface configurations.

As a natural expansion of 1qfa’s, we define a *2-way quantum finite automaton with a garbage tape* (or a 2qfa, for short) by allowing a tape head to move in both directions as well as to stay still. To be more formal, a 2qfa  $M$  is of the form  $(Q, \Sigma, \{\clubsuit, \$\}, \Xi, \delta, q_0, Q_{acc}, Q_{rej})$  with a (quantum) transition function  $\delta : Q \times \tilde{\Sigma} \times Q \times D \times \Xi_\lambda \rightarrow \mathbb{C}$  with  $D = \{-1, 0, +1\}$ , where “ $-1$ ” and “ $+1$ ” indicate that an input-tape head moves respectively to the left and to the right, and “ $0$ ” means that the tape head makes a stationary move. Assuming that  $M$  is in inner state  $q$ , scanning input symbol  $\sigma$ , if we make a transition  $\delta(q, \sigma|p, d, \xi) = \alpha$ , then  $M$  changes its inner state to  $p$ , moves its input-tape head in direction  $d$ , and writes down garbage symbol  $\xi$  by moving its tape head to the right whenever  $\xi \neq \lambda$ . Similarly to 1qfa’s, the use of garbage tapes provides sufficiently high computational power to 2qfa’s.

<sup>5</sup>Introducing 1qfa’s in this form aims at handling both 1qfa’s and 2qfa’s in the same fashion.

The notion of configurations for 2qfa's is the same as that for 1qfa's; however, the time-evolution operator  $U_\delta$  is in the form:

$$U_\delta|q, x, i, z\rangle = \sum_{(p, d, \xi)} \delta(q, x_{(i)}|p, d, \xi)|p, x, i + d \pmod{|x| + 2}, z\xi\rangle,$$

where  $(p, d, \xi)$  ranges over  $Q \times D \times \Xi_\lambda$ . When  $x$  is fixed, we write  $U_\delta^{(x)}$  for the associated time-evolution operator on the Hilbert space spanned by surface configurations. Unlike the case of 1qfa's, no time bound is in general imposed on each 2qfa, and thus some of the computation paths may possibly be infinitely long. To scale the runtime of such a 2qfa, we thus need to take the expectation of the lengths of computation paths; that is,  $\sum_{i=1}^\infty i \cdot \|\Pi_{halt} U_\delta |\phi_{i-1}\rangle\|^2$ , where  $|\phi_i\rangle$  is the superposition of configurations obtained after the  $i$ th step of the 2qfa. We call this value the *expected runtime* of  $M$  on  $x$ .

We say that  $M$  is *well-formed* if  $U_\delta^{(x)}$  is a unitary matrix (i.e.,  $U_\delta^{(x)}(U_\delta^{(x)})^\dagger = I$ ) for all  $x \in \Sigma^*$ . In the rest of this exposition, we always assume that all 1qfa's as well as 2qfa's are well-formed.

## 2.3 Transition Tables

The behavior of a finite automaton is completely dictated by its transition function  $\delta$ . However, it is sometimes convenient to use “transition tables” instead of transition functions. A *transition table* is intuitively a matrix form of the “description” of  $\delta$ , which is further encoded into a classical string. Each row of a transition table is indexed by the elements  $(q, \sigma)$  in  $Q \times \tilde{\Sigma}$ , each column is indexed by the elements  $(p, d)$  in  $Q \times D$ , and the  $((q, \sigma), (p, d))$ -entry of the table contains 1 if  $\delta$  maps  $(q, \sigma)$  to  $(p, d)$ ; 0 otherwise. Although this definition is valid for deterministic/nondeterministic finite automata, we cannot use the same one for 2qfa's because we need to deal with a set of (quantum) transition amplitudes, which are generally arbitrary complex numbers. Hence, we need to find an appropriate way of fitting such transition amplitudes into transition tables.

Let  $M$  denote a 2qfa of the form  $(Q, \Sigma, \{\$, \#\}, \Xi, \delta, q_0, Q_{acc}, Q_{rej})$  with  $\Sigma$  and  $\Xi$  of constant sizes. Since  $M$ 's transition amplitudes are complex numbers, we want to use a *quantum circuit* to generate (or approximate) those amplitudes and we then encode this quantum circuit into a transition table, where a quantum circuit is made up of finitely many quantum gates taken from a certain *universal set*. In the rest of this exposition, we fix  $\{CNOT, H, T_{\pi/8}\}$  as such a universal set, where *CNOT* is the Controlled-NOT, *H* is the Hadamard transform, and  $T_{\pi/8}$  is the  $\pi/8$ -rotation around the  $z$  axis (see, e.g., [29] for their definitions and properties). If necessary, we also use the *identity map*  $I$  as a special quantum gate.

Formally, we express inner states of  $M$  as strings in  $\{0, 1\}^{r_1}$ , symbols in  $\Xi_\lambda$  as strings in  $\{0, 1\}^{r_2}$  for two appropriate numbers  $r_1$  and  $r_2$  in  $\mathbb{N}^+$  (thus,  $|Q| = 2^{r_1}$  and  $|\Xi_\lambda| = 2^{r_2}$ ), and head directions  $\{-1, 0, +1\}$  as elements  $\{10, 11, 01\}$ , respectively. For convenience, we write  $V_{q, \sigma}$  for a  $2^{r_1+r_2+2} \times 2^{r_1+r_2+2}$  unitary matrix that, on input  $|\phi_0\rangle = |0^{r_1}\rangle|00\rangle|0^{r_2}\rangle$ , produces a quantum state  $\sum_{(p, d, \xi)} \delta(q, \sigma|p, d, \xi)|p, d, \xi\rangle$ . A *transition table*  $T$  of  $M$  on input  $x$  is a matrix, each row of which is indexed by  $(q, \sigma)$  in  $Q \times \tilde{\Sigma}$ , whose  $(q, \sigma)$ -row contains a “description” of  $V_{q, \sigma}$ . Given any parameter  $n$ , any input length  $l$ , and any pair  $(q, \sigma) \in Q \times \tilde{\Sigma}$ , we intend to define a quantum circuit  $C_{q, \sigma}^{(n, l)}$  so as to approximate  $V_{q, \sigma}$ .

Assume that  $M$  on input  $x$  runs in expected  $p(n, |x|)$  time for a certain function  $p$  and errors with probability at most a certain constant  $\varepsilon \in [0, 1/2)$ . It suffices to consider the first  $cp(n, |x|)$  steps of  $M$  for an appropriately chosen absolute constant  $c \geq 1$  to guarantee that the error probability obtained during these steps is still at most another constant  $\varepsilon' = \frac{1}{2}(\varepsilon + \frac{1}{2})$ , where  $\varepsilon < \varepsilon' < \frac{1}{2}$ . Under this circumstance, letting  $\alpha = \frac{1}{2}(\frac{1}{2} - \varepsilon)$ , we want to approximate  $V_{q, \sigma}$  with *inaccuracy* of  $\alpha 2^{-(r_1+r_2+2)}/cp(n, |x|)$  by a certain quantum circuit  $C_{q, \sigma}^{(n, l)}$  made up of the aforementioned universal quantum gates; namely,  $\|V_{q, \sigma}|\phi_0\rangle - C_{q, \sigma}^{(n, |x|)}|\phi_0\rangle\| \leq \alpha 2^{-(r_1+r_2+2)}/cp(n, |x|)$ . For convenience, we write  $\tilde{M}$  for the machine obtained from  $M$  by replacing each  $V_{q, \sigma}$  with  $C_{q, \sigma}^{(n, l)}$ .

Let  $s = s(n, l)$  denote the total number of the quantum gates used in  $C_{q, \sigma}^{(n, l)}$ . A reasonable upper bound of  $s(n, l)$  is given by the following lemma.

**Lemma 2.1** *Any  $2^r \times 2^r$  unitary matrix  $V$  can be approximated by a certain  $r$ -qubit quantum circuit  $C$  of  $O(n^2 r^4 2^{2r})$  universal gates satisfying  $\|V|0^r\rangle - C|0^r\rangle\| \leq 2^{-n}$ .*

**Proof.** Given a unitary matrix  $V$ , as in the same way described in [29, Section 4.5.1], we can take a number  $k \leq 2^{r-1}(2^r - 1)$  and  $k$  2-level unitary matrices  $U_1, U_2, \dots, U_k$  yielding  $V = U_1 U_2 \dots U_k$ . In

the same way as in [29, Section 4.5.2], we then decompose each 2-level unitary matrix  $U_i$  into  $O(r^2)$  1-qubit and CNOT gates, where 1-qubit gates may not be limited to  $\{H, T_{\pi/8}\}$ . Combining them, we can realize  $V$  by a certain quantum circuit of  $O(r^2 2^{2r})$  1-qubit and CNOT gates. Let  $s$  denote the number of used quantum gates. We set  $\varepsilon = 2^{-n}/s$ . The Solovay-Kitaev theorem (e.g., [25, 29]) then guarantees the existence of a constant  $e$  in the real interval  $[1, 2]$  such that each 1-qubit gate can be approximated by  $O(\log^e(1/\varepsilon))$  universal gates from  $\{CNOT, H, T_{\pi/8}\}$  to within  $\varepsilon$ . To approximate  $V$ , we need  $O(r^2 2^{2r}) \times O(\log^e(r^2 2^{n+2r})) \subseteq O(n^2 r^4 2^{2r})$  universal gates because of  $1 \leq e \leq 2$ ,  $\log(r^2) \leq 2r$ , and  $(n + 2r)^e \leq n^e (2r)^e \leq 4n^2 r^2$ .  $\square$

Since  $V_{q,\sigma}$  is a  $2^{r_1+r_2+2} \times 2^{r_1+r_2+2}$  unitary matrix, Lemma 2.1 implies that  $s(n, l)$  is bounded by  $O((r_1 + r_2 + 2)^6 2^{2(r_1+r_2+2)} \log^2 p(n, l)) \subseteq O(|Q|^2 |\Xi|^2 \log^6 |Q| |\Xi| \cdot \log^2 p(n, l)) \subseteq O(|Q|^8 |\Xi|^8 \log^2 p(n, l))$  because of  $\log |Q| |\Xi| \leq |Q| |\Xi|$ . Thus, we can express the quantum circuit  $C_{q,\sigma}^{(n,l)}$  as an appropriate series  $(I^{k_1} \otimes G_1 \otimes I^{k_1})(I^{k_2} \otimes G_2 \otimes I^{k_2}) \dots (I^{k_s} \otimes G_s \otimes I^{k_s})$ , where  $s = s(n, l)$ ,  $k_i \in [|Q| + 2]$ , and  $G_i \in \{I, CNOT, H, T_{\pi/8}\}$  for any index  $i \in [s]$ . From this series,  $C_{q,\sigma}^{(n,l)}$  is completely specified by the series  $(k_1, G_1), (k_2, G_2), \dots, (k_s, G_s)$ , and thus  $C_{q,\sigma}^{(n,l)}$  can be encoded into a string  $\langle C_{q,\sigma}^{(n,l)} \rangle$  of the form  $1^{k_1} \# \langle G_1 \rangle \# 2^{k_2} \# \langle G_2 \rangle \# 2 \dots \# 2^{k_s} \# \langle G_s \rangle$ , where  $\langle I \rangle = 1$ ,  $\langle CNOT \rangle = 2$ ,  $\langle H \rangle = 3$ , and  $\langle T_{\pi/2} \rangle = 4$ . Note that, if this string  $\langle C_{q,\sigma}^{(n,l)} \rangle$  is given to a tape, then it is possible to generate the quantum state  $C_{q,\sigma}^{(n,l)} |0^{r_1}\rangle |00\rangle |0^{r_2}\rangle$  by “sweeping” the tape once from left to right and applying operators  $I^{k_i} \otimes G_i \otimes I^{k'_i}$  one by one.

We then expand our encoding to all quantum transitions. Let  $\{C_{a,b}^{(n,l)}\}_{(a,b) \in Q \times \tilde{\Sigma}}$  be a set of all approximated quantum circuits corresponding to  $\delta$ . We first enumerate all elements in  $Q \times \tilde{\Sigma}$  as  $\{(a_1, b_1), (a_2, b_2), \dots, (a_k, b_k)\}$ , where  $k = |Q \times \tilde{\Sigma}|$ , and, according to this enumeration, we set  $\langle T \rangle$  as  $\langle C_{a_1, b_1}^{(n,l)} \rangle \# 3 \langle C_{a_2, b_2}^{(n,l)} \rangle \# 3 \dots \# 3 \langle C_{a_k, b_k}^{(n,l)} \rangle$ . The *length* of the string  $\langle T \rangle$  is bounded from above by  $k \cdot O(s) \subseteq O(|Q|^9 |\Xi|^8 \log^2 p(n, l))$ .

Since  $M$  preforms at most  $cp(n, |x|) 2^{r_1+r_2+2}$  applications of the matrices  $V_{q,\sigma}$ , the quantum state produced by  $M$  after  $cp(n, |x|)$  steps can be approximated to within  $\alpha$  by the quantum state produced by  $\tilde{M}$  in  $cp(n, |x|)$  steps. This implies that the difference between the acceptance (resp., rejecting) probabilities between  $M$  and  $\tilde{M}$  is at most  $\alpha$  (see, e.g., [6, 44]). Therefore, if  $M$  accepts (resp., rejects)  $x$  with probability at least  $1 - \varepsilon$ , then  $\tilde{M}$  accepts (resp., rejects)  $x$  with probability at least  $1 - \varepsilon - \alpha$ , which equals  $1 - \varepsilon'$ .

## 2.4 Quantum Turing Machines with Advice

Each Turing machine is equipped with a read-only input tape with the two endmarkers,  $\clubsuit$  and  $\$$ , as well as a *rewritable work tape*. Occasionally, we further equip a Turing machine with a *read-only advice tape*, which holds a given advice string surrounded by the same endmarkers. It is important to note that no machine modifies an advice string during its computation.

In accordance with the aforementioned garbage-tape quantum finite automata, we also supply quantum Turing machines with *(flexible) garbage tapes*. Since we discuss only such machines in later sections, *quantum Turing machines equipped with (flexible) garbage tapes* are simply referred to as QTMs. later in Section 3, we will introduce another notion of “rigid” garbage tape. A QTM also has a work tape and a work alphabet  $\Gamma$  (including a unique *blank symbol*  $B$ ) and both the work and the garbage tapes are initially blank. We further provide a piece of useful information, known as “advice.” An *advice function*  $h$  is a function from  $\mathbb{N}$  to  $\Theta^*$  for a certain advice alphabet  $\Theta$ , and each value  $h(n)$  is called an *advice string*. Since we need to handle such advice, we further furnish the QTM with a distinguished *advice tape*. Each advice string is initially written on the advice tape, surrounded by the two endmarkers,  $\clubsuit$  and  $\$$ . For convenience, we call a QTM with an advice tape by an *advised QTM*. Formally, an advised QTM is a decuple  $(Q, \Sigma, \{\clubsuit, \$\}, \Gamma, \Theta, \Xi, \delta, q_0, Q_{acc}, Q_{rej})$ , including a work alphabet  $\Gamma$ , an advice alphabet  $\Theta$ , a garbage alphabet  $\Xi$ , and  $\delta$  is a (quantum) transition function from  $Q \times \tilde{\Sigma} \times \Gamma \times \check{\Theta} \times Q \times \Gamma \times \Xi_\lambda \times D_1 \times D_2 \times D_3$  to  $\mathbb{C}$ , where  $\check{\Theta} = \Theta \cup \{\clubsuit, \$\}$ ,  $\Xi_\lambda = \Xi \cup \{\lambda\}$ ,  $D_1 = D_2 = \{-1, 0, +1\}$  (for the input-tape and the work-tape heads), and  $D_3 = \{+1\}$  (for the garbage-tape head). Notice that our QTM can take arbitrary complex amplitudes, not limited to polynomial-time approximable amplitudes as in most literature.

A *configuration* of  $M$  is a tuple  $(q, x, t_1, y, t_2, w, t_3, z)$ , where  $q \in Q$ ,  $t_1, t_2, t_3 \in \mathbb{Z}$ ,  $x \in \Sigma^*$ ,  $y \in \Gamma^*$ ,  $w \in \Theta^*$ , and  $z \in \Xi^*$ . This configuration expresses a circumstance in which  $M$  is in inner state  $q$ , scanning the  $t_1$ th cell of the input tape, the  $t_2$ th cell of the work tape containing  $y$ , the  $t_3$ th cell of the advice tape



containing  $w$ , and the garbage tape containing  $z$ . There is no need to include the head position of the garbage tape similarly to the 2qfa case. When  $x$  and  $h(|x|)$  are fixed throughout computation, we use a *surface configuration* of the form  $(q, t_1, y, t_2, t_3)$  instead. In a way similar to 2qfa's, we define  $\mathcal{H}_{acc}$ ,  $\mathcal{H}_{rej}$ ,  $\mathcal{H}_{halt}$ , and  $\mathcal{H}_{non}$  as well as  $\Pi_{acc}$ ,  $\Pi_{rej}$ ,  $\Pi_{halt}$ , and  $\Pi_{non}$ . The *configuration space* of  $M$  is denoted  $\mathcal{H}_{conf}$ . The *time-evolution operator* of an advised QTM acting on  $\mathcal{H}_{conf}$  is defined in a similar way as 2qfa's; that is,

$$\begin{aligned} U_\delta |q, x, t_1, y, t_2, w, t_3, z\rangle \\ = \sum_{(p, \tau, d_1, d_2, d_3, \xi)} \zeta_\delta |p, x, t_1 + d_1 \bmod (|x| + 2), \tilde{y}_\xi, t_2 + d_2, w, t_3 + d_3, z\xi\rangle, \end{aligned}$$

where  $\zeta_\delta = \delta(q, x_{(t_1)}, y_{(t_2)}, w_{(t_3)}, p, \tau, d_1, d_2, d_3, \xi)$ ,  $\tilde{y}_\xi = y_{(1)}y_{(2)} \cdots y_{(t_2-1)}\xi y_{(t_2+1)} \cdots y_{(|y|)}$ , and  $(p, \tau, d_1, d_2, d_3, \xi)$  ranges over  $Q \times \tilde{\Theta} \times D_1 \times D_2 \times D_3 \times \Xi_\lambda$ . We also use  $U_\delta^{(x, h(|x|))}$  instead of  $U_\delta$  when  $x$  and  $h(|x|)$  are fixed throughout computation. We demand that the time-evolution operator  $U_\delta^{(x, h(|x|))}$  of our QTM should be *unitary* for any input  $x \in \Sigma^*$ . A *computation* of  $M$  on input  $x$  is defined, similarly to 2qfa's, as a series of superpositions of configurations generated by  $U_\delta$  and  $\Pi_{non}$ . The *expected runtime* of  $M$  with  $h$  on  $x$ , denoted by  $\text{Time}_{M, h}(x)$ , is the expectation of the lengths of the computation paths of  $M$  on  $x$ . We say that  $M$  with  $h$  runs in *expected polynomial time* if there exists a polynomial  $p$  satisfying  $\text{Time}_{M, h}(x) \leq p(|x|)$  for all  $x$ . The *space usage* of  $M$  on  $x$ , denoted by  $\text{Space}_{M, h}(x)$ , is the maximal cell index that  $M$ 's work-tape head visits during  $M$ 's computation on  $x$  with  $h$ . We say that  $M$  with  $h$  uses *logarithmic space* (or log space) if there exist two constants  $c, d \geq 0$  satisfying  $\text{Space}_{M, h}(x) \leq c \log |x| + d$  for all  $x$ .

Given a language  $L$  over  $\Sigma$ , we say in general that a machine  $M$  *recognizes  $L$  with bounded-error probability* if there exists a constant  $\varepsilon \in [0, 1/2)$  such that, for any  $x \in L$ ,  $M$  accepts  $x$  with probability at least  $1 - \varepsilon$  and, for any  $x \in \Sigma^* - L$ ,  $M$  rejects  $x$  with probability at least  $1 - \varepsilon$ . In contrast,  $M$  is said to *recognize  $L$  with unbounded-error probability* if, for any  $x \in L$ ,  $M$  accepts  $x$  with probability more than  $1/2$  and, for any  $x \in \Sigma^* - L$ ,  $M$  rejects  $x$  with probability at least  $1/2$ .

The advised quantum complexity class BQL/poly consists of all languages, each of which is recognized with bounded-error probability by a certain QTM equipped with an advice tape and a polynomially-bounded advice function using only logarithmic space. In a similar manner, we define BPL/poly and NL/poly respectively using probabilistic Turing machines (PTMs) and nondeterministic Turing machines (NTMs). When underlying QTMs and PTMs are further restricted to run in expected polynomial time, we obtain ptime-BQL/poly and ptime-BPL/poly from BQL/poly and BPL/poly, respectively.

In Section 2.5, we will further transform those complexity classes to their “parameterized” versions.

## 2.5 Promise Decision Problems and Parameterized Decision Problems

A *promise decision problem* in general has the form  $(A, B)$  over a given input alphabet  $\Sigma$  satisfying both  $A, B \subseteq \Sigma^*$  and  $A \cap B = \emptyset$ . In the past literature [3, 9, 56], promise problems have been studied also in connection to quantum finite automata. As stated in Section 1.1, we particularly deal with a “family”  $\{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems over a certain fixed alphabet  $\Sigma$  (not depending on  $n$ ), where  $L_n^{(+)}$  consists of “positive” instances and  $L_n^{(-)}$  consists of “negative” instances. Notice that, for each index  $n \in \mathbb{N}$ ,  $L_n^{(+)} \cap L_n^{(-)} = \emptyset$  and  $L_n^{(+)} \cup L_n^{(-)} \subseteq \Sigma^*$ . However, we usually do not demand  $(L_n^{(+)} \cup L_n^{(-)}) \cap (L_m^{(+)} \cup L_m^{(-)}) = \emptyset$  for any distinct pair  $(m, n)$ . For any family  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems, we use the notation  $\text{co-}\mathcal{L}$  to denote the family  $\{(L_n^{(-)}, L_n^{(+)})\}_{n \in \mathbb{N}}$  obtained by exchanging between  $L_n^{(+)}$  and  $L_n^{(-)}$ .

Let  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  denote any family of promise decision problems. Given a family  $\mathcal{M} = \{M_n\}_{n \in \mathbb{N}}$  of certain “specified” machines that satisfy certain “predetermined” criteria for acceptance and rejection, we generally say that  $M_n$  *recognizes (solves or computes)* the promise decision problem  $(L_n^{(+)}, L_n^{(-)})$  if (1) for any positive instance  $x \in L_n^{(+)}$ ,  $M_n$  accepts  $x$  and, (2) for any negative instance  $x \in L_n^{(-)}$ ,  $M_n$  rejects  $x$ . There is no requirement for the behavior of  $M_n$  on any string  $x$  outside of  $L_n^{(+)} \cup L_n^{(-)}$  and  $M_n$  may possibly *neither accept nor reject* such a string  $x$ . Conveniently, we set  $\Sigma_n = L_n^{(+)} \cup L_n^{(-)}$  for each index  $n \in \mathbb{N}$ . Any input in  $\Sigma_n$  is said to be a *valid input*. We also say that  $x$  is *promised* if  $x$  is a valid input. Note that we *do not* force  $\{\Sigma_n\}_{n \in \mathbb{N}}$  to satisfy  $\Sigma_n \cap \Sigma_{n'} = \emptyset$  for all distinct pairs  $n, n' \in \mathbb{N}$ . The family  $\mathcal{M}$  of machines is said to *solve  $\mathcal{L}$  with bounded-error probability* if

there exists a constant  $\varepsilon \in [0, 1/2)$  such that, for any index  $n \in \mathbb{N}$ , (i) for all  $x \in L_n^{(+)}$ ,  $p_{acc, M_n}(x) \geq 1 - \varepsilon$  and (ii) for all  $x \in L_n^{(-)}$ ,  $p_{rej, M_n}(x) \geq 1 - \varepsilon$ . In comparison,  $\mathcal{M}$  solves  $\mathcal{L}$  with unbounded-error probability if, for any  $n \in \mathbb{N}$ , (i') for all  $x \in L_n^{(+)}$ ,  $p_{acc, M_n}(x) > \frac{1}{2}$  and (ii') for all  $x \in L_n^{(-)}$ ,  $p_{rej, M_n}(x) \geq \frac{1}{2}$ .

For two families  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  and  $\hat{\mathcal{L}} = \{(\hat{L}_n^{(+)}, \hat{L}_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems, we say that  $\hat{\mathcal{L}}$  is an *extension* of  $\mathcal{L}$  if  $L_n^{(+)} \subseteq \hat{L}_n^{(+)}$  and  $L_n^{(-)} \subseteq \hat{L}_n^{(-)}$  hold for any index  $n \in \mathbb{N}$ . If the set  $\{1^n \# x \mid n \in \mathbb{N}, x \in L_n^{(+)} \cup L_n^{(-)}\}$  belongs to  $L$ , then  $\mathcal{L}$  is called *L-good*. Moreover, a collection  $\mathcal{F}$  of families of promise decision problems is *L-good* if every element in  $\mathcal{F}$  has an L-good extension of it in  $\mathcal{F}$ .

A *size parameter* is a function from  $\Sigma^*$  to  $\mathbb{N}$  for a certain alphabet  $\Sigma$ . As typical examples,  $m_{bin}(x)$  denotes the binary size  $|x|$  of input  $x$  and  $m_{ver}(G)$  indicates the number of vertices in a given graph  $G$ . A *parameterized decision problem* over  $\Sigma$  is a pair  $(L, m)$  with a language (equivalently, a decision problem)  $L$  over  $\Sigma$  and a size parameter  $m : \Sigma^* \rightarrow \mathbb{N}$ .

Let us define a useful translation between a parameterized decision problem and a family of promise decision problems. Given a parameterized decision problem  $(L, m)$  over an alphabet  $\Sigma$ , a family  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems is said to be *induced from*  $(L, m)$  if, for each index  $n \in \mathbb{N}$ ,  $L_n^{(+)} = L \cap \Sigma_n$  and  $L_n^{(-)} = \bar{L} \cap \Sigma_n$ , where  $\Sigma_n = \{x \in \Sigma^* \mid m(x) = n\}$ .

On the contrary, let  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  be an L-good family of promise decision problems over a fixed alphabet  $\Sigma$ . We set  $L_{all} = \bigcup_{n \in \mathbb{N}} (L_n^{(+)} \cup L_n^{(-)})$ , which is obviously included in  $\Sigma^*$  but may not equal  $\Sigma^*$ . With the use of a distinguished separator  $\#$ , we write  $\Sigma_{\#}$  for the set  $\Sigma \cup \{\#\}$ . For each index  $n \in \mathbb{N}$ , we define  $K_n^{(+)} = \{1^n \# x \mid x \in L_n^{(+)}\}$  and  $K_n^{(-)} = \{1^n \# x \mid x \in L_n^{(-)}\} \cup \{z \# x \mid z \in \Sigma^n - \{1^n\}, x \in \Sigma_{\#}^* \} \cup \{z \mid z \in \Sigma^n\}$ . Furthermore, we set  $K = \bigcup_{n \in \mathbb{N}} K_n^{(+)}$  and  $\bar{K} = \bigcup_{n \in \mathbb{N}} K_n^{(-)}$ . It follows that  $K \cap \bar{K} = \emptyset$  and  $K \cup \bar{K} = \Sigma_{\#}^*$ . We define a new size parameter  $m' : \Sigma_{\#}^* \rightarrow \mathbb{N}$  by setting, for any  $w \in \Sigma_{\#}^*$ ,  $m'(w) = n$  if  $w = 1^n \# x$  for a certain  $x \in L_n^{(+)} \cup L_n^{(-)}$ , and  $m'(w) = |w|$  otherwise. Since  $\mathcal{L}$  is L-good,  $m'$  must be computed using logarithmic space. The pair  $(K, m')$  therefore turns out to be a parameterized decision problem over  $\Sigma_{\#}$  and  $(K, m')$  is said to be *induced from*  $\mathcal{L}$ .

A size parameter  $m : \Sigma^* \rightarrow \mathbb{N}$  is said to be *polynomially bounded* if there exists a polynomial  $p$  witnessing  $m(x) \leq p(|x|)$  for all  $x \in \Sigma^*$ ; in contrast,  $m$  is *polynomially honest* if a certain fixed polynomial  $q$  satisfies  $|x| \leq q(m(x))$  for any  $x \in \Sigma^*$ . For the size parameter  $m'$  defined above, since  $m'(w) \leq |w|$  holds for all  $w$ ,  $m'$  must be polynomially bounded. We use the notation PHSP to denote the set of all parameterized decision problems  $(L, m)$  for which  $m$  is polynomially honest.

We say that  $m$  is a *log-space size parameter* if there exists a deterministic Turing machine (or a DTM, for short)  $M$  such that, for any string  $x$ ,  $M$  takes  $x$  as an input and produces  $1^{m(x)}$  (i.e.,  $m(x)$  in unary) on its write-once output tape using  $O(\log |x|)$  work space [50]. Notice that the function  $f(x) = 1^{m(x)}$  is polynomially bounded because, otherwise, a log-space machine computing  $f$  must stay in an infinite loop. Hence,  $m$  is polynomially bounded.

As noted in Section 1.1, the prefix “para-” is used to describe parameterized complexity classes. We define para-BQL as the class of parameterized decision problems  $(L, m)$  solvable by bounded-error QTMs using  $O(\log m(x))$  space, where  $m$  is any log-space size parameter. If the expected runtime of each underlying QTM is further limited to  $p(m(x), |x|)$  for a fixed polynomial  $p$ , we use the notation of para-ptime-BQL. The probabilistic counterparts of para-BQL and para-ptime-BQL are respectively denoted by para-BPL and para-ptime-BPL. With the use of deterministic and nondeterministic Turing machines instead, their runtime is customarily set to be the length of the shortest accepting path if it exists, and the length of the shortest rejecting path otherwise. Following [50], we similarly define para-L and para-NL as the parameterizations of L and NL, respectively. Moreover, we write para-NL/poly to denote the parameterization of NL/poly, which is obtained by replacing languages  $L$  in NL with their associated parameterized decision problems  $(L, m)$  and also by taking advice functions of advice size at most polynomials in  $m(x)|x|$ . Similarly, we obtain para-BQL/poly, para-ptime-BQL/poly, etc. Refer to [50, 52, 53] for the relevant notions.

## 2.6 Nonuniform State Complexity

Our purpose is to introduce nonuniform complexity classes defined by state complexities of quantum finite automata families. Related to these classes, we also consider classes based on deterministic, non-deterministic, and probabilistic finite automata.

The *state complexity* generally refers to the number of inner states used to describe the behavior of a given automaton. However, since we use a (uniform or nonuniform) family  $\mathcal{M} = \{M_n\}_{n \in \mathbb{N}}$  of finite automata, the state complexity of such a family becomes a function in  $n$ . More formally, the *state complexity*  $sc(n)$  (or  $sc(M_n)$ ) of a family  $\mathcal{M}$  of finite automata  $M_n$  with a set  $Q_n$  of inner states is a function defined by  $sc(n) = |Q_n|$  for all indices  $n \in \mathbb{N}$  [36]. In later sections, nonetheless, we use nonuniform families  $\{M_n\}_{n \in \mathbb{N}}$  of finite automata, in which each  $M_n$  may possibly be chosen independently, and therefore we emphatically call  $sc(n)$  the *nonuniform state complexity function*. Customarily, “uniformity” is thought as a special case of “nonuniformity.”

The nonuniform state complexity class 1D is the collection of all nonuniform families  $\{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems, each family of which is based on a certain fixed alphabet  $\Sigma$  (not depending on  $n$ ) and satisfies the following condition: there exist a polynomial  $p$  and a nonuniform family  $\{M_n\}_{n \in \mathbb{N}}$  of 1dfa’s such that, for each index  $n \in \mathbb{N}$ , (i)  $M_n$  has at most  $p(n)$  inner states and (ii)  $M_n$  solves  $(L_n^{(+)}, L_n^{(-)})$  on all inputs. In a similar way, we can define 1N using 1nfa’s instead of 1dfa’s.

The use of  $n^{O(1)}$  inner states endows underlying 1dfa’s with enormous computational power. As a quick example, let us consider the family  $\{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  with  $L_n^{(+)} = \{a^i b^{i^2} \in \Sigma_n \mid i \leq n\}$ , and  $L_n^{(-)} = \Sigma_n - L_n^{(+)}$  for any index  $n \in \mathbb{N}$ , where  $\Sigma_n = \{a^i b^j \mid i, j \in \mathbb{N}, i, j < 2^n\}$ . No standard pushdown automaton recognizes the language  $L' = \bigcup_{n \in \mathbb{N}} L_n^{(+)}$ ; however, the following 1dfa  $M_n$  with  $O(n)$  inner states easily solves  $(L_n^{(+)}, L_n^{(-)})$ : on input  $x \in \Sigma_n$ , find a number  $i \leq n$  for which  $x = a^i b^j$  and, using this value  $i$  repeatedly, check whether  $j = i^2$ .

The notation  $2^{1D}$  indicates the collection of all nonuniform families of promise decision problems satisfying the following: for each  $\{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of such families, every promise decision problem  $(L_n^{(+)}, L_n^{(-)})$  is recognized by a certain 1dfa of at most  $2^{p(n)}$  inner states for a certain fixed polynomial  $p$  independent of  $n$ . If nonuniform families of 2dfa’s that have polynomially many inner states are used instead, we obtain 2D from 1D. In a similar manner, the use of nondeterministic finite automata introduces 1N,  $2^{1N}$ , and 2N as well.

In what follows, we present a useful lemma, which directly follows from [52, Lemma 3.3]. This lemma will be used in later sections.

**Lemma 2.2** [52] *Let  $m$  be a log-space size parameter over alphabet  $\Sigma$ . If  $m$  is polynomially bounded and polynomially honest, there is a nonuniform family  $\{M_n\}_{n \in \mathbb{N}}$  of 1dfa’s equipped with write-once output tapes such that each  $M_n$  has  $n^{O(1)}$  inner states and  $M_n$  produces  $1^{m(x)}$  on its output tape from each input  $x \in \Sigma^*$ .*

We also consider finite automata whose input-tape heads either move to the right or make stationary moves. Such automata are briefly called *1.5 way*. If we replace 1dfa’s in the definition of 1D by 1.5dfa’s, then we obtain 1.5D. Clearly,  $1D \subseteq 1.5D \subseteq 2D$  follows. As is shown in the lemma below, 1.5D actually coincides with 1D. Similarly, we can consider *1.5-way quantum finite automata with flexible garbage tapes* (or 1.5qfa’s). Nevertheless, as we will show in Lemma 3.7, the same equality does not hold for quantum finite automata, exhibiting a clear difference between 1.5dfa’s and 1.5qfa’s.

**Lemma 2.3**  $1D = 1.5D \neq 2D$ .

**Proof.** Clearly, 1D is included in 1.5D. For the converse inclusion, let  $\{M_n\}_{n \in \mathbb{N}}$  be a nonuniform family of 1.5dfa’s. We want to simulate each  $M_n$  by a certain 1dfa, say,  $N_n$  of  $O(n^2)$  states. The desired 1dfa  $N_n$  works as follows. On input  $x$ , if  $M_n$  moves its tape head to the right, then  $N_n$  makes the same move. Consider the case where  $M_n$  is in inner state  $q$  and makes its tape head stay still. Let  $\delta_n$  denote the transition function of  $M_n$ . Assume that there are a number  $k \geq 1$  and a series  $p_1, p_2, \dots, p_k$  of inner states for which  $\delta_n(q, \sigma) = (p_1, 0)$ ,  $\delta_n(p_i, \lambda) = (p_{i+1}, 0)$  for any  $i \in [k-1]$ , and  $\delta_n(p_k, \lambda) = (p, 1)$ . Since  $M_n$  must halt for an arbitrary input  $x$ , it follows that  $k < |Q|$ . We define a transition function  $\delta'_n$  of  $N_n$  as  $\delta'_n(q, \sigma) = p$ . The obtained  $N_n$  is clearly 1-way and simulates  $M_n$  on all inputs. Note that  $N_n$  uses at most  $|Q|$  states. Therefore, we conclude that  $1.5D \subseteq 1D$ .

The separation  $1.5D \neq 2D$  follows instantly from the known result of  $1D \neq 2D$  [19].  $\square$

A family  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems over a fixed alphabet  $\Sigma$  is said to have a *polynomial ceiling* if there exists a polynomial  $r$  satisfying  $L_n^{(+)} \cup L_n^{(-)} \subseteq \Sigma^{\leq r(n)}$  for all  $n \in \mathbb{N}$ . The nonuniform state complexity class 2D/poly is the restriction of 2D, whose families  $\mathcal{L}$  all have appropriate

polynomial ceilings. Similarly, we can obtain  $2N/\text{poly}$  from  $2N$ .

Given any nonuniform state complexity class  $\mathcal{F}$ ,  $\text{co-}\mathcal{F}$  consists of all families  $\mathcal{L}$  of promise decision problems for which  $\text{co-}\mathcal{L}$  belongs to  $\mathcal{F}$ . Using this notion, we obtain, e.g.,  $\text{co-1N}$  from  $1N$ .

### 3 One-Way Quantum Finite Automata Families

One-way finite automata are often used to model *data processing* where streaming input data are processed instantly with little memory space, since their tape heads read input strings from left to right without stopping. Notice that, by our definition of one-wayness, 1-way finite automata halt exactly in  $|x| + 2$  steps for any given input  $x$ . For this reason, in many cases, it is possible to demonstrate class separations among nonuniform state complexity classes.

Formally, the notation  $1BQ$  denotes the collection of all nonuniform families  $\{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems over fixed input alphabets  $\Sigma$  (not depending on  $n$  in each family) such that there exist a nonuniform family  $\{M_n\}_{n \in \mathbb{N}}$  of 1qfa's, two polynomials  $p$  and  $r$ , and a constant  $\varepsilon \in [0, 1/2)$  satisfying the following: for each index  $n \in \mathbb{N}$ , (1) for any string  $x \in L_n^{(+)}$ ,  $M_n$  accepts  $x$  with probability at least  $1 - \varepsilon$  and, for any string  $x \in L_n^{(-)}$ ,  $M_n$  rejects  $x$  with probability at least  $1 - \varepsilon$ , (2)  $M_n$  uses at most  $p(n)$  inner states, and (3)  $M_n$ 's garbage alphabet has size at most  $r(n)$ . When  $M_n$  satisfies Condition (1), we simply say that  $M_n$  *solves* (or *recognizes*)  $(L_n^{(+)}, L_n^{(-)})$  with error probability at most  $\varepsilon$ . In this case,  $M_n$  is said to *make bounded errors*. In contrast, we obtain  $1Q$  from  $1BQ$  if we change Condition (1) to the following new condition for each index  $n \in \mathbb{N}$ : (1') for every string  $x \in L_n^{(+)}$ ,  $M_n$  accepts  $x$  with probability more than  $1/2$  and, for every string  $x \in L_n^{(-)}$ ,  $M_n$  rejects  $x$  with probability at least  $1/2$ . In this case, we say that  $M_n$  *makes unbounded errors*. Notice that these unbounded-error criteria supersede the aforementioned bounded-error criteria, resulting in the inclusion  $1BQ \subseteq 1Q$ . In addition, we also obtain  $1NQ$  from  $1BQ$  if, instead of Condition (1), we use the following new condition for each  $n \in \mathbb{N}$ : (1'') for any string  $x \in L_n^{(+)}$ ,  $M_n$  accepts  $x$  with positive probability, and for any string  $x \in L_n^{(-)}$ ,  $M_n$  rejects  $x$  with certainty. Furthermore, we obtain  $1.5BQ$  from  $1BQ$  by substituting 1.5qfa's for 1qfa's in the definition of  $1BQ$ .

In a way similar to introducing  $1Q$ , we define  $1P$  using *one-way probabilistic finite automata* (or 1pfa's, for short), whose transition probabilities are *arbitrary numbers* in the real interval  $[0, 1]$ , which make unbounded-error probability. With the use of the bounded-error criteria instead, we further obtain  $1BP$  from  $1P$ . Analogously to  $1BQ \subseteq 1Q$ , we can obtain  $1BP \subseteq 1P$ . By allowing underlying 1dfa's to use exponentially many inner states, we have already defined  $2^{1D}$  in Section 2.6. Similarly, we define  $2^{1BQ}$ ,  $2^{1Q}$ ,  $2^{1BP}$ , and  $2^{1P}$ , respectively from  $1BQ$ ,  $1Q$ ,  $1BP$ , and  $1P$ .

There are known inclusions and separations:  $1D \subsetneq 1N \subsetneq 2^{1D}$ ,  $1D = \text{co-1D}$ , and  $1N \neq \text{co-1N}$  [19, 20]. To obtain Figure 1 as our first major contribution, we intend to prove the following collapse and separation relationships among the nonuniform state complexity classes listed in the figure.

**Theorem 3.1** 1.  $1D \subsetneq 1BP \subsetneq 1BQ \subseteq 2^{1D}$ .

2.  $1BQ = \text{co-1BQ}$  and  $1P = \text{co-1P}$ .

3.  $1BQ \subsetneq 1.5BQ$ .

4.  $1N \subsetneq 1P \subseteq 1Q$  and  $2^{1D} \not\subseteq 1P$ .

5.  $1BP \subsetneq 2^{1BP}$  and  $1P \subsetneq 2^{1P}$ .

6.  $1N \subsetneq 1NQ \subseteq 1Q$  and  $1NQ \not\subseteq 2^{1D}$ .

The proof of Theorem 3.1 is split into Lemmas 3.2–3.7 that follow shortly.

Concerning the state complexity of 1qfa's, unfortunately, the past literature has only a few results. Bianchi, Mereghetti, and Palano [8], for example, demonstrated an exponential gap between the state complexities of 1dfa's and 1qfa's. Their result leads to the class separation  $1D \neq 1BQ$ , which is further strengthened in Lemma 3.2(1). Let us present the following five relationships, which are partly due to [1, 2, 16, 27, 42].

**Lemma 3.2** (1)  $1D \subsetneq 1BP$ . (2)  $1BP \subsetneq 1BQ$ . (3)  $1BQ \subseteq 2^{1D}$ .

**Proof.** (1) Since  $1D \subseteq 1BP$  is obvious, we here show that  $1D \neq 1BP$ . Following [16], we

define  $\mathcal{TRIO} = \{(TRIO_n^{(+)}, TRIO_n^{(-)})\}_{n \in \mathbb{N}}$  with  $TRIO_n^{(+)} = \{\#x_1^2 y_1 \# x_2^2 y_2 \# \dots \# x_n^2 y_n \mid \forall i \in [n^2] (x_i, y_i \in \{0, 1\}^n), (x_1, x_2, \dots, x_n) \prec (y_1, y_2, \dots, y_n)\}$  and  $TRIO_n^{(-)} = \{\#x_1^2 y_1 \# x_2^2 y_2 \# \dots \# x_n^2 y_n \mid \forall i \in [n^2] (x_i, y_i \in \{0, 1\}^n), (x_1, x_2, \dots, x_n) \succ (y_1, y_2, \dots, y_n)\}$ , where  $(x_1, \dots, x_r) \succ (y_1, \dots, y_r)$  iff, for any index  $i \in [r]$ , there exists an index  $j \in [n]$  satisfying  $(x_i)_{(j)} > (y_i)_{(j)}$ . It was shown in [16, Theorems 5.1–5.2] that  $\mathcal{TRIO}$  can be solved by a certain family of  $(4n+3)$ -state 1pfa's with error probability at most  $(1 - \frac{1}{n})^{n^2}$  and that any family of 1dfa's solving  $\mathcal{TRIO}$  requires at least  $2^n$  inner states. These results assert that  $\mathcal{TRIO} \in 1BP$  but  $\mathcal{TRIO} \notin 1D$ .

(2) With the use of a flexible garbage tape, any 1dfa can be simulated by an appropriately chosen 1qfa in such a way that, at every step, the 1qfa discards onto its garbage tape the information on the current inner state of the 1dfa. In a similar way, any 1pfa can be simulated with the same error probability by an appropriate 1qfa. This instantly implies that  $1BP \subseteq 1BQ$ .

The separation  $1BP \neq 1BQ$  is shown as follows. Given two positive integers  $m$  and  $r$ , the notation  $m|r$  expresses that  $m$  is *divisible* by  $r$  (i.e.,  $r$  divides  $m$ ). Let  $e(n) = 2^n$  for all  $n \in \mathbb{N}$ . Given any index  $n \in \mathbb{N}^+$ , we define  $\Sigma_n = \{a^j b^m \mid j, m \in \mathbb{N}^+ \wedge j, m < 2^{e(n)}\}$ . Let us consider  $2MOD_n^{(+)} = \{a^j b^m \in \Sigma_n : j|pr(e(n))\}$  and  $2MOD_n^{(-)} = \Sigma_n - 2MOD_n^{(+)}$ , where  $pr(m)$  is the largest prime number  $p$  satisfying  $p \leq m$  if  $m \geq 2$ , and  $pr(1)$  is undefined. We write  $2MOD$  for the family  $\{(2MOD_n^{(+)}, 2MOD_n^{(-)})\}_{n \in \mathbb{N}}$ . By modifying a 1qfa construction in [1, 2], we can build a 1qfa that solves  $(2MOD_n^{(+)}, 2MOD_n^{(-)})$  using  $O(\log pr(e(n)))$  inner states with bounded-error probability. Since  $2^{n/c} \leq pr(e(n)) \leq 2^n - 1$  for a certain absolute constant  $c \geq 1$ , we conclude that  $\Theta(\log pr(e(n))) = \Theta(n)$ , and thus  $2MOD$  belongs to  $1BQ$ .

Next, we want to verify that, for any integer  $n \geq 1$ , no 1pfa with less than  $pr(e(n))$  inner states solves  $(2MOD_n^{(+)}, 2MOD_n^{(-)})$  with bounded-error probability. In comparison, given each prime number  $p$ , let  $MOD_p$  denote the unary language  $\{a^j : j \in \mathbb{N}^+, j|p\}$ . It is known in [1, 2] that (\*) for any prime number  $p$ , any 1pfa needs at least  $p$  inner states in order to recognize  $MOD_p$ . If there is a 1pfa  $M$  with  $k$  states ( $k < pr(e(n))$ ) that solves  $(2MOD_n^{(+)}, 2MOD_n^{(-)})$  with bounded-error probability, then we can convert  $M$  to another 1pfa that can recognize  $MOD_{pr(e(n))}$  with bounded-error probability by setting  $m = 0$  in the definition of  $2MOD_n^{(+)}$ . This is a contradiction against the above statement (\*). Therefore,  $2MOD$  cannot belong to  $1BP$ .

(3) We start with the following general claim. Kondacs and Watrous [27] proved a similar claim for 1qfa's with no garbage tape. In the claim, we need to deal with flexible garbage tapes as well.

**Claim 1** *Any  $n$ -state bounded-error 1qfa with a garbage alphabet of arbitrary size can be simulated by a certain  $2^{O(n^2)}$ -state 1dfa.*

**Proof.** Let  $\Sigma$  be any alphabet and let  $\varepsilon \in [0, 1/2)$  be any error bound. Take an  $n$ -state 1qfa  $M$  with a garbage alphabet  $\Xi$  of size  $r \in \mathbb{N}^+$  with error probability at most  $\varepsilon$ . Following an argument similar to [27] (or [4]), we want to simulate  $M$  classically. Let  $Q$  denote the set of all inner states of  $M$  with  $n = |Q|$  and take two Hilbert spaces  $\mathcal{H}_Q = \text{span}\{|q\rangle \mid q \in Q\}$  and  $\mathcal{H}_\Xi = \text{span}\{|z\rangle \mid z \in \Xi^{\leq |x|+2}\}$ . Let  $x$  be any input in  $\Sigma^*$ . We need to express the content of the garbage tape as a string in  $\Xi^{\leq |x|+2}$ . For convenience, we write  $A$  for the set  $\Xi^{\leq |x|+2}$ . We then define the unit sphere  $\mathcal{S}_n$  to be  $\{|\phi\rangle \in \mathcal{H}_Q \otimes \mathcal{H}_\Xi \mid \|\phi\| = 1\}$ .

Let us consider any superposition  $|\phi\rangle$  of surface configurations of  $M$  on the input  $x$ . Assume that  $|\phi\rangle$  is of the form  $\sum_{z \in A} \sum_{q \in Q} \alpha_{q,z} |q\rangle |z\rangle$  for appropriate amplitudes  $\alpha_{q,z}$ 's. For each fixed string  $z \in A$ , let  $|\phi_z\rangle = (1/p_z) \sum_{q \in Q} \alpha_{q,z} |q\rangle$ , where  $p_z$  is a normalizing positive constant. We are focused on the density operator  $\rho = \sum_{z \in A} p_z |\phi_z\rangle \langle \phi_z|$ . Since  $\rho$  is of dimension  $n$ , we can choose  $k$  vectors  $\{|\phi'_i\rangle\}_{i \in [k]}$  and  $k$  numbers  $\{p'_i\}_{i \in [k]}$  for which  $\rho$  is expressed as  $\sum_{i \in [k]} p'_i |\phi'_i\rangle \langle \phi'_i|$ .

A set  $S$  of vectors in  $\mathcal{H}_Q$  is called an  $\varepsilon$ -net if, for any vector  $|\phi\rangle$  in  $\mathcal{H}_Q$ , there always exists another vector  $|\psi\rangle$  in  $S$  satisfying  $\|\phi\rangle - |\psi\rangle\| \leq \varepsilon$ . The Solovay-Kitaev theorem (e.g., [25, 29]) then ensures the existence of an  $\varepsilon$ -net of  $2^{O(n)}$  vectors. For the Hilbert space  $\mathcal{H}_Q^{\otimes n}$ , there is an  $\varepsilon$ -net, say,  $S$  of  $(2^{O(n)})^n (= 2^{O(n^2)})$  vectors. Treating each vector in  $S$  as a new inner state, we can construct a new 1dfa that recognizes  $L$ . Note that the state complexity of this 1dfa is at most  $|S|$ . The lemma follows from the fact that  $|S|$  is bounded by  $2^{O(n^2)}$ .  $\square$

The desired inclusion  $1BQ \subseteq 2^{1D}$  comes directly from Claim 1. This completes the proof.  $\square$

**Lemma 3.3** (1)  $1P \subseteq 1Q$ . (2)  $1NQ \subseteq 1Q$ .

**Proof.** (1) Similarly to the proof of Lemma 3.2(2) for  $1BP \subseteq 1BQ$ , the use of a garbage tape makes it possible to simulate any unbounded-error 1pfa by an appropriately designed unbounded-error 1qfa, and therefore  $1P \subseteq 1Q$  follows.

(2) To verify  $1NQ \subseteq 1Q$ , for any given 1qfa  $M$ , it suffices to construct another 1qfa  $N$  so that  $N$  splits its first move into two parts: (i) with probability  $1/2$ ,  $N$  starts simulating  $M$  on  $x$  and, with probability  $1/2$ ,  $N$  immediately accepts  $x$ . Let us consider the case where  $N$  tries to simulate  $M$ . If  $M$  accepts  $x$  with positive probability  $\alpha$ , then  $N$  also accepts  $x$  with probability  $\frac{1}{2} + \frac{\alpha}{2}$ . On the contrary, if  $M$  rejects  $x$  with certainty, then  $N$  rejects  $x$  with probability  $\frac{1}{2}$ . Thus,  $N$  correctly simulates  $M$  with unbounded-error probability.  $\square$

**Lemma 3.4** (1)  $1BQ = \text{co-}1BQ$ . (2)  $1BP \subsetneq 2^{1BP}$ .

**Proof.** (1) This is shown simply by exchanging the roles of accepting states and rejecting states of underlying 1qfa's.

(2) Obviously,  $1BP \subseteq 2^{1BP}$  follows. If  $1BP = 2^{1BP}$  holds, then  $2^{1BP} \subseteq 1P$  follows instantly because of  $1BP \subseteq 1P$ . Since  $1D \subseteq 1BP$ , we conclude from  $2^{1BP} \subseteq 1P$  that  $2^{1D} \subseteq 1P$ . However, this contradicts Lemma 3.5(1). Therefore, we obtain  $1BP \neq 2^{1BP}$ .  $\square$

**Lemma 3.5** (1)  $2^{1D} \not\subseteq 1P$ . (2)  $1P = \text{co-}1P$ . (3)  $1N \subsetneq 1P$ . (4)  $1P \subsetneq 2^{1P}$ .

**Proof.** (1) Let us consider  $L_{NH} = \{0^x 10^{y_1} 1 \dots 10^{y_k} 1 \mid x, y_1, \dots, y_k \in \mathbb{N}^+, \exists l \in [k] (x = \sum_{i \in [l]} y_i)\}$ . Associated with  $L_{NH}$ , we define the family  $\mathcal{L}_{NH} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  by setting  $L_n^{(+)} = \{z \in L_{NH} \mid |z| \leq n^{\log n}\}$  and  $L_n^{(-)} = \{z \notin L_{NH} \mid |z| \leq n^{\log n}\}$ . We want to show that  $\mathcal{L}_{NH} \in 2^{1D} - 1P$ . As demonstrated by Freivalds and Karpinski [14], no one-way PTM recognizes  $L_{NH}$  using  $o(\log n)$  space with unbounded-error probability. This means that no 1pfa with  $o(n^{\log n})$  inner states can recognize  $(L_n^{(+)}, L_n^{(-)})$  for almost all  $n \in \mathbb{N}$ . We thus conclude that  $\mathcal{L}_{NH}$  does not belong to  $1P$ . However, as shown in [14],  $\mathcal{L}_{NH}$  can be solved deterministically using  $O(n^{\log n})$  ( $\subseteq O(2^n)$ ) inner states, and thus  $\mathcal{L}_{NH}$  falls in  $2^{1D}$ .

(2) Firstly, we modify the acceptance/rejection criteria of unbounded-error 1pfa's.

**Claim 2** Given a 1pfa  $M$ , there exists another 1pfa  $N$  such that, for any  $x$ , if  $M$  accepts  $x$  with probability more than  $1/2$  (resp., rejects  $x$  with probability at least  $1/2$ ), then  $N$  accepts (resp., rejects)  $x$  with probability more than  $1/2$ .

Let us show how the desired result follows from this claim. By the claim, we first pick up a nonuniform family  $\{M_n\}_{n \in \mathbb{N}}$  of polynomial-size unbounded-error 1pfa's that always either accept or reject with probability more than  $1/2$ . We then simply exchange the roles of accepting and rejecting states. This concludes that  $1P = \text{co-}1P$ .

To show the claim, let  $\alpha_0$  be the minimum nonzero transition probability of  $M$ . Without loss of generality, we assume that  $\alpha_0 \leq \frac{1}{2}$ . Note that the acceptance probability of  $M$  on  $x$  (if any) is more than  $1/2 + \alpha_0^{|x|+2}$  since  $M$  halts within  $|x| + 2$  steps. We design a new 2pfa  $N$  to behave as follows. On input  $x$ , in scanning  $\phi$ , with probability  $1 - \alpha_0/2$ , we mimic the behavior of  $M$ . With probability  $\alpha_0/2$ , from the initial inner state  $q_0$ , we enter a new inner state, say,  $\tilde{q}_0$ . Whenever we read each tape symbol, from the current inner state  $\tilde{q}_0$ , we re-enter  $\tilde{q}_0$  with probability  $\alpha_0$ . Furthermore, with equal probability  $\frac{1}{2}(1 - \alpha_0)$ , we enter both an accepting state and a rejecting state. When we read  $\$,$  from  $\tilde{q}_0$ , we enter a rejecting state with certainty. It thus follows that either the accepting probability of  $N$  is more than  $\frac{1}{2} + \frac{5}{8}\alpha_0^{|x|+2} > \frac{1}{2}$  or the rejecting probability of  $N$  is more than  $\frac{1}{2} + \frac{1}{4}\alpha_0^{|x|+2} > \frac{1}{2}$ .

(3) The inclusion  $1N \subseteq 1P$  can be shown by modifying underlying 1nfa's as follows: choose all nondeterministic transitions "probabilistically" and enter both an accepting state and a rejecting state with equal probability only along each rejecting path. The separation  $1N \neq 1P$  instantly follows because  $1P = \text{co-}1P$  by (2) and  $1N \neq \text{co-}1N$  by [19, 20].

(4) Clearly, it follows that  $1P \subseteq 2^{1P}$ . If  $1P = 2^{1P}$ , then we obtain  $2^{1D} \subseteq 2^{1P} = 1P$  since  $1D \subseteq 1P$ . As a consequence,  $2^{1D} \subseteq 1P$  follows. This is a clear contradiction against Lemma 3.5(1). Therefore,  $2^{1P}$  is different from  $1P$ .  $\square$

**Lemma 3.6**  $1N \subsetneq 1NQ \not\subseteq 2^{1D}$ .

**Proof.** It is easy to show that  $1N \subseteq 1NQ$  with the use of the amplitude set  $\{0, \pm 1/2, \pm 1\}$ . To lead to  $1N \neq 1NQ$ , it suffices to prove that  $1NQ \not\subseteq 2^{1D}$  because, if  $1N = 1NQ$ , then we obtain  $1NQ \subseteq 2^{1D}$  from  $1N \subseteq 2^{1D}$  [19], leading to a contradiction.

Our goal is now to prove that  $1NQ \not\subseteq 2^{1D}$ . Let  $\Sigma = \{0, 1\}$  and set  $e(n) = 2^n$  for any  $n \in \mathbb{N}$ . Consider the family  $\mathcal{NEQ} = \{(NEQ_n^{(+)}, NEQ_n^{(-)})\}_{n \in \mathbb{N}}$ , where  $NEQ_n^{(+)} = \{w \in \Sigma^{2^{e(n)}} \mid \#_0(w) \neq \#_1(w)\}$  and  $NEQ_n^{(-)} = \Sigma^{2^{e(n)}} - NEQ_n^{(+)}$  for each index  $n \in \mathbb{N}$ , where  $\#_b(x)$  expresses the total number of symbol  $b$  in string  $x$ . Since  $\mathcal{NEQ}$  belongs to  $1NQ$  [7, 10], we want to claim that  $\mathcal{NEQ} \notin 2^{1D}$ . Assume otherwise. Note that  $\text{co-}\mathcal{NEQ}$  also belongs to  $2^{1D}$ . By a communication-complexity argument (e.g., [26]) or a swapping lemma [45], it is not difficult to show that  $N_n$  requires at least  $o(2^{e(n)})$  inner states to recognize  $NEQ_n^{(-)}$  in the worst case. This is a clear contradiction against our assumption of  $\mathcal{NEQ} \in 2^{1D}$ .  $\square$

To finish the proof of Theorem 3.1, we still need to examine the computational power of 1.5-way finite automata. With the use of 1.5qfa's, we can define the associated nonuniform state complexity class 1.5BQ. We have already seen in Lemma 2.3 that 1.5D coincides with 1D. However, as shown in Lemma 3.7, the same coincidence does not occur between 1.5BQ and 1BQ.

**Lemma 3.7**  $1BQ \neq 1.5BQ$ .

**Proof.** To lead to a contradiction, we assume that  $1BQ = 1.5BQ$ . Let us consider the family  $\mathcal{EQ} = \{(EQ_n^{(+)}, EQ_n^{(-)})\}_{n \in \mathbb{N}}$ , each element  $(EQ_n^{(+)}, EQ_n^{(-)})$  of which is defined as  $EQ_n^{(+)} = \{a^{\bar{n}}b^{\bar{n}} \mid \bar{n} = 2^{e(n)}\}$  and  $EQ_n^{(-)} = \{a^i b^j \mid i, j \geq 0, i \neq j, i+j = 2^{e(n)+1}\}$  for each index  $n \in \mathbb{N}$ , where  $e(n) = 2^n$ . Since we need to deal only with *valid* inputs given to  $\mathcal{EQ}$ , for each promise decision problem  $(EQ_n^{(+)}, EQ_n^{(-)})$ , we do not need to check that any given input  $x$  has length exactly  $e(n)+1$ . As noted in [4], each  $(EQ_n^{(+)}, EQ_n^{(-)})$  can be solved by an appropriately designed bounded-error 1.5qfa with a constant number of inner states, and thus  $\mathcal{EQ}$  belongs to 1BQ by our assumption. Since  $1BQ \subseteq 2^{1D}$  by Lemma 3.2(3), there are a polynomial  $p$  and a nonuniform family of 1dfa's that solves  $\mathcal{EQ}$  using at most  $2^{p(n)}$  inner states. However, any 1dfa solving  $(EQ_n^{(+)}, EQ_n^{(-)})$  requires at least  $2^{e(n)}$  inner states. Since  $p(n) < e(n)$  for almost all  $n \in \mathbb{N}$ , we obtain a clear contradiction. Therefore, 1.5BQ differs from 1BQ.  $\square$

Recall that our 1qfa uses a *flexible garbage tape*. When both an input-tape head and a garbage-tape head synchronize in their movement (thus, the garbage-tape head makes no stationary move), we can prove another close relationship between 1pfa's and 1qfa's in Lemma 3.8. Such a restrictive use of the garbage tape is referred to as a *rigid garbage tape* in contrast to the flexible garbage tape. A similar model was already seen in [48, Section 5.2], in which a rewritable track of an input tape is used as our rigid garbage tape. To emphasize the use of rigid garbage tapes, we use the special notation  $1Q^{(+)}$  opposed to 1Q when underlying 1qfa's use rigid garbage tapes.

**Lemma 3.8** (1)  $1Q^{(+)} = 1P$ . (2)  $2^{1Q^{(+)}} = 2^{1P}$ .

**Proof.** (1) The proof of Lemma 3.3(2) has actually proven that  $1P \subseteq 1Q^{(+)}$ .

Our proof of  $1Q^{(+)} \subseteq 1P$  essentially follows an argument used in [33] together with [44]. In [33, Lemma 8.1], for any linear-time one-tape QTM with rational amplitudes, its acceptance probability on each input  $x$  is calculated in terms of two acceptance probabilities of linear-time one-tape PTMs. In [44, Theorem 7.1], a language of the form  $\{x \mid p_{acc, M_1}(x) > p_{acc, M_2}(x)\}$  for two polynomial-time QTMs  $M_1$  and  $M_2$  belongs to PP. It is also possible to restrict all transition amplitudes of any QTM on real numbers. This restriction simplifies our construction of the desired 1pfa. A major deviation from [33], however, is that we allow 1qfa's to take arbitrary real transition amplitudes in the real interval  $[0, 1]$ .

Let us consider an arbitrary family  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems in  $1Q^{(+)}$  and also a nonuniform family  $\{M_n\}_{n \in \mathbb{N}}$  of  $r(n)$ -state 1qfa's with rigid garbage tapes that recognizes  $\mathcal{L}$  in *exactly*  $|x| + 2$  steps with bounded-error probability, where  $r$  is a suitable polynomial. Let  $\alpha_n$  denote the smallest absolute value of any transition amplitude used by  $M_n$ .

In a natural way, we express each computation path of  $M_n$  on input  $x$  as strings of length  $|x| + 2$  over an appropriate alphabet, say,  $\Theta$ . To such a computation path  $y$ , we assign the product of all transition amplitudes taken by  $M_n$  along the path  $y$ . This product is briefly called the *amplitude associated with y*. Let us denote by  $amp(z, y, z)$  the amplitudes associated with a computation path  $y$  of  $M_n$  on  $x$  ending with a final surface configuration  $z$ .

Firstly, we wish to define two functions. The first function  $f_+(x, z)$  is set to be  $\sum_y^{(+)} |\text{amp}(x, y, z)|$ , where  $y$  ranges over all accepting computation paths of  $M_n$  on  $x$  ending with  $z$  for which  $\text{amp}(x, y, z)$  is a positive real number. We also define  $f_-(x, z) = \sum_y^{(-)} |\text{amp}(x, y, z)|$  similarly except that we collect  $\text{amp}(x, y, z)$ 's having negative values. The acceptance probability  $p_{n, \text{acc}}(x)$  of  $M_n$  on  $x$  is therefore expressed as  $\sum_z (f_+(x, z) - f_-(x, z))^2$ . To compute this value  $p_{n, \text{acc}}(x)$ , we need to compute two additional values  $p_n^{(+)}(x) = \frac{1}{2} \sum_z (f_+^2(x, z) + f_-^2(x, z))$  and  $p_n^{(-)}(x) = \sum_z f_+(x, z) f_-(x, z)$ . Since  $M_n$  makes bounded-error probability, for an appropriate constant  $\varepsilon \in [0, 1/2)$ , it follows that (i)  $p_{n, \text{acc}}(x) \geq 1 - \varepsilon$  iff  $p_n^{(+)}(x) > p_n^{(-)}(x)$  and (ii)  $p_{n, \text{acc}}(x) \leq \varepsilon$  iff  $p_n^{(+)}(x) < p_n^{(-)}(x)$ .

Secondly, we want to produce in a probabilistic manner the values  $p_n^{(+)}(x)$  and  $p_n^{(-)}(x)$  multiplied by  $\zeta_{n, x}$ , where  $\zeta_{n, x} = \alpha_0^{2(|x|+2)} \cdot |\Theta|^{-2(|x|+2)}$ . Since the computation paths of  $M_n$  on  $x$  may interfere with one another, we need to ignore such an interference and classically generate these computation paths with certain probabilities. Concerning  $p_n^{(-)}(x)$ , we want to produce  $\zeta_{n, x} \sum_{y_1} \sum_{y_2} |\text{amp}(x, y_1, z)| |\text{amp}(x, y_2, z)|$ , provided that  $y_1$  makes  $\text{amp}(x, y_1, z)$  positive and  $y_2$  makes  $\text{amp}(x, y_2, z)$  negative. For this purpose, we first generate all pairs  $(y_1, y_2)$  in  $\Theta^{|x|+2} \times \Theta^{|x|+2}$  at random, simulate  $M_n$  along two computation paths  $y_1$  and  $y_2$  simultaneously, check that the same final configuration is reached, check that  $y_1$  and  $y_2$  satisfy the aforementioned condition, and finally accept with probability  $\zeta_{n, x} |\text{amp}(x, y_1, z)| |\text{amp}(x, y_2, z)|$ . Similarly in essence, we can also estimate  $p_n^{(+)}(x)$ . The simultaneous simulation of the two computation paths  $y_1$  and  $y_2$  is possible because the garbage-tape head of  $M_n$  always moves to the right.

In a way similar to [44, Theorem 7.1], we can construct an  $n^{O(1)}$ -state unbounded-error 1pfa whose acceptance probability on the input  $x$  equals  $\zeta_{n, x} p_{n, \text{acc}}^{(+)}(x)$  and also another 1pfa whose acceptance probability is  $\zeta_{n, x} p_{n, \text{acc}}^{(-)}(x)$ . Finally, we combine those two 2pfa's to a new 1pfa  $D_n$  so that (i')  $p_n^{(+)}(x) > p_n^{(-)}(x)$  iff  $D_n$  accepts  $x$  with probability more than  $1/2$  and (ii')  $p_n^{(+)}(x) < p_n^{(-)}(x)$  iff  $D_n$  accepts  $x$  with probability less than  $1/2$ . Hence,  $D_n$  solves  $(L_n^{(+)}, L_n^{(-)})$  with unbounded-error probability. This concludes that  $\mathcal{L}$  belongs to 1P.

(2) An argument similar to (1) leads to the collapse  $2^{\text{1Q}^{(+)}} = 2^{\text{1P}}$ .  $\square$

## 4 Two-Way Quantum Finite Automata Families

We have already introduced 1BQ and 1Q using 1qfa's in Section 3. Furthermore, let us define 2BQ to be the collection of all nonuniform families  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems such that, for each family  $\mathcal{L}$ , there exist a nonuniform family  $\{M_n\}_{n \in \mathbb{N}}$  of 2qfa's, two polynomials  $p$  and  $r$ , and an error bound  $\varepsilon \in [0, 1/2)$  satisfying the following: for each index  $n \in \mathbb{N}$ , (1)  $M_n$  makes error probability at most  $\varepsilon$  on all inputs in  $\Sigma_n = L_n^{(+)} \cup L_n^{(-)}$ , (2)  $M_n$  uses at most  $p(n)$  inner states, and (3)  $M_n$ 's garbage alphabet has size at most  $r(n)$ . We further define 2Q similarly to 2BQ using unbounded-error probability instead of bounded-error probability.

In a similar fashion, we define 2BP in terms of bounded-error 2pfa's having polynomially many states. The unbounded-error analogue of 2BP is denoted by 2P.<sup>6</sup>

Notice that, for any suffix  $\mathcal{A} \in \{\text{D}, \text{N}, \text{BP}, \text{BQ}\}$ , we obtain an inclusion  $1\mathcal{A} \subseteq 2\mathcal{A}$  since "1-way" machines can be thought as a special case of "2-way" machines. Unlike the one-way case, there are few known separations among 2-way nonuniform state complexity classes, except that  $1\text{D} \subsetneq 2\text{D} \subseteq 2\text{N} \subsetneq 2^{\text{1D}}$  [19]. Even for the basic classes 2D and 2N, unfortunately, we do not know that  $2\text{N} \neq \text{co-}2\text{N}$  as well as  $1\text{N} \not\subseteq 2\text{D}$ . For the probabilistic counterpart, nonetheless, we can draw the following relationships from an early work of Dwork and Stockmeyer [11].

**Lemma 4.1** (1)  $2\text{BP} \subseteq 2^{\text{1D}}$ . (2)  $2\text{BP} \not\subseteq 2\text{N}$ .

**Proof.** (1) From [11, Theorem 6.1], we can conclude that any  $n$ -state 2pfa with error probability  $\varepsilon \in [0, 1/2)$  running in expected  $O(n^k)$  time for any fixed constant  $k > 0$  can be precisely simulated by an appropriately chosen 1dfa of  $2^{O(n^2 \log n)}$  states. This implies the first inclusion.

<sup>6</sup>In [19], the polynomial-time 2BP was considered under the name of  $2\text{P}_2$  and the polynomial-time 2P was studied under the name of 2P but they are restricted to so-called "regular" language families. Here, we demand neither the polynomial time-bound nor the regular family requirement.



(2) Let us define an example family  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems as follows: for any index  $n \in \mathbb{N}$ , let  $L_n^{(+)} = \{0^{e(n)}\}$  and  $L_n^{(-)} = \{0^i \mid 0 \leq i \leq e(n)^2, i \neq e(n)\}$ , where  $e(n) = 2^n$ . It was proven in [11, Theorem 6.2] that (1) for any  $n \in \mathbb{N}$ ,  $(L_n^{(+)}, L_n^{(-)})$  is solved with bounded-error probability by a certain 2pfa of  $O(\log^2 e(n)/\log \log e(n))$  inner states in expected  $O(|x|^2)$  time, where  $x$  is a “symbolic” input, and (2) any 2nfa requires at least  $e(n)$  inner states to solve  $(L_n^{(+)}, L_n^{(-)})$ . These results (1)–(2) help us conclude that  $2BP \not\subseteq 2N$ .  $\square$

Unlike 1qfa’s, it is possible to reduce the garbage alphabets of 2qfa’s to constant size. This reduction in size makes it easier to estimate the computational complexity of 2qfa’s.

**Lemma 4.2** *Given any  $n$ -state 2qfa  $M$  with a garbage alphabet of size  $k$ , there exists another  $O(nk \log k)$ -state 2qfa  $N$  with a constant-size garbage alphabet that can simulate  $M$  with the same error probability.*

**Proof.** Let  $M = (Q, \Sigma, \{\$, \#\}, \Xi, \delta, q_0, Q_{acc}, Q_{rej})$  denote any given  $n$ -state 2qfa. For simplicity, let  $r = \lceil \log k \rceil$  and assume that  $\Xi = \{0, 1\}^r$  for a certain  $r \in \mathbb{N}^+$ . For each symbol  $\xi$  in  $\Xi$ , we express it as  $\xi_1 \xi_2 \cdots \xi_r$  using  $r$  binary symbols. We define a new 2qfa  $N$  with  $Q' = Q \times \Xi \times [r]$  and  $\Xi' = \{0, 1\}$  as follows. The initial state of  $N$  is set to be  $(q_0, \lambda, 0)$ . Assume that  $N$  is in inner state  $(q, \lambda, 0)$ . If  $M$  applies a quantum transition of the form  $\delta(q, \sigma | p, d, \xi) = \alpha$  with  $\alpha \neq 0$ , then  $N$  first writes  $\xi_1$  on a garbage tape, enters the next inner state  $(p, \xi, 1)$  with amplitude  $\alpha$ , and moves its tape head in direction  $d$ . For each index  $i \in [r - 1]$ , with no tape-head move,  $N$  enters  $(p, \xi, i)$  from  $(p, \xi, i - 1)$  by dumping  $\xi_i$  onto the garbage tape. Finally,  $N$  enters  $(p, \lambda, 0)$  from  $(p, \xi, r)$  with no tape-head move. It is not difficult to show that  $N$  correctly simulates  $M$  with the same error probability.  $\square$

As a consequence of Lemma 4.2, in the rest of this exposition, we consider only 2qfa’s that have garbage alphabets of *constant size* unless otherwise stated.

To complete Figure 1, we further need to show the following relationships.

**Lemma 4.3** (1)  $2D \subsetneq 2BP$ . (2)  $2BP \subseteq 2BQ$ . (3)  $2BQ \subseteq 2Q$ . (4)  $2N \subseteq 2P$ . (5)  $2P \subseteq 2Q$ .

**Proof.** (1) Since every 2dfa can be viewed as a special case of bounded-error 2pfa, we obtain  $2D \subseteq 2BP$ . The separation  $2D \neq 2BP$  comes from  $2BP \not\subseteq 2N$  of Lemma 4.1(2).

(2) Any 2pfa can be simulated by an appropriately designed 2qfa, which discards all information on the previously taken inner states to a garbage tape. The inclusion  $2BP \subseteq 2BQ$  follows immediately.

(3) This is trivial by the difference between the error bound criteria of  $2BQ$  and  $2Q$ .

(4) In a way similar to Lemma 3.5(3), we can show that  $2N \subseteq 2Q$ .

(5) We have already proven  $1P \subseteq 1Q$  in Lemma 3.2(4). A similar argument proves  $2P \subseteq 2Q$  as well.

$\square$

For two-way head moves, the behaviors of two-way finite automata vary significantly depending on their machine types. For deterministic and nondeterministic computations, as discussed in Section 1.3, the length of “accepting” computation paths of 2dfa’s and 2nfa’s are always upper-bounded *linearly* in input size. For probabilistic computation, nonetheless, this is not always true. As demonstrated by Freivalds [12], bounded-error 2pfa’s with no runtime bound in general have more computational power than bounded-error expected-polynomial-time 2pfa’s. Since we are interested in two-way machines running in *expected polynomial time*, we introduce the nonuniform state complexity class ptime-2BQ from 2BQ by further requiring each family  $\{M_n\}_{n \in \mathbb{N}}$  of underlying 2qfa’s  $M_n$  to run in expected  $p(n, |x|)$  time on all inputs  $x$  for an appropriate polynomial  $p$  (depending only on the choice of the family  $\{M_n\}_{n \in \mathbb{N}}$ ). Similarly, we obtain ptime-2Q, ptime-2BP, and ptime-2P from 2Q, 2BP, and 2P, respectively. A similar statement to Lemma 4.3 also holds for ptime-2BP, ptime-2P, ptime-2BQ, and ptime-2Q.

## 5 Advised QTMs and Nonuniform Families of Quantum Finite Automata

The nonuniform state complexity classes  $2D/\text{poly}$  and  $2N/\text{poly}$  are composed of only families of promise decision problems having *polynomial ceilings*. In a similar way, we define two more classes  $2BP/\text{poly}$  and  $2BQ/\text{poly}$  using the notion of polynomial ceiling. From these complexity classes, we can obtain

ptime-2BP/poly and ptime-2BQ/poly by forcing underlying machines to halt in expected polynomial time.

Concerning the acceptance/rejection criteria for families of promise decision problems, quantum and probabilistic computations are quite different from deterministic and nondeterministic computations. Recall from Section 2.5 the notion of *L-good* collection  $\mathcal{F}$  of families of promise decision problems. It is not difficult to verify that 2D and 2N are L-good because 2dfa's and 2nfa's can be easily forced to either accept or reject on all possible inputs. On the contrary, 2BQ and 2BP may not be L-good. For this very reason, we introduce the special notation  $2BQ^\dagger$  (resp.,  $2BP^\dagger$ ) to denote the largest subclass of 2BQ (resp., 2BP) that is *closed downward under L-good extensions* (i.e., if  $\mathcal{L}_1$  is an L-good extension of  $\mathcal{L}_2$  and  $\mathcal{L}_2 \in 2BQ^\dagger$ , then  $\mathcal{L}_1$  belongs to  $2BQ^\dagger$ ).

The sole purpose of this section is to verify Theorem 1.1, from which Corollary 1.2 follows immediately. For the proof of the theorem, as a supportive statement inspired by [52], we first establish a close connection between nonuniform state complexity classes and parameterized complexity classes. The actual proof of Theorem 1.1 will be given in Section 5.2.

## 5.1 The Roles of Advice and the Honesty Condition

When we discuss quantum and probabilistic finite automata with two-way head moves, the runtime bounds of 2qfa's and 2pfa's are quite essential because, as Freivalds [12] demonstrated, expected-polynomial-time bounded-error 2pfa's are in general less powerful than, say, expected-exponential-time bounded-error 2pfa's.

Let us begin with the precise statement of Proposition 5.1, in which we give a nice characterization of parameterized decision problems solvable by advised QTMs in expected polynomial time using logarithmic space in terms of expected-polynomial-time 2qfa's having polynomially many inner states. Firstly, we recall that PHSP consists of all parameterized decision problems with polynomially-honest size parameters.

**Proposition 5.1** *Let  $(\mathcal{A}, \mathcal{B}) \in \{(D, L), (N, NL), (BP, BPL), (BQ, BQL)\}$  and let  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  denote any family of promise decision problems.*

1. *Given each parameterized decision problem  $(L, m)$ , if  $\mathcal{L}$  is induced from  $(L, m)$ , then it follows that  $(L, m) \in \text{para-ptime-}\mathcal{B}/\text{poly} \cap \text{PHSP}$  iff  $\mathcal{L} \in \text{ptime-2}\mathcal{A}/\text{poly}$ .*
2. *If  $\mathcal{L}$  is L-good and  $(K, m)$  is the parameterized decision problem induced from  $\mathcal{L}$ , then it follows that  $(K, m) \in \text{para-ptime-}\mathcal{B}/\text{poly} \cap \text{PHSP}$  iff  $\mathcal{L} \in \text{ptime-2}\mathcal{A}/\text{poly}$ .*

Hereafter, we intend to verify Proposition 5.1 only for the case of  $\mathcal{A} = \text{BQ}$  and  $\mathcal{B} = \text{BQL}$  since the other cases can be proven in a similar way with only necessary modification. For readability, the proof of Proposition 5.1(1) is split into two lemmas, Lemmas 5.2 and 5.3. Lemma 5.2, which is in fact more general than what the proposition actually needs, states that we can simulate an advised QTM by a certain nonuniform family of 2qfa's of appropriate state complexity.

**Lemma 5.2** *Let  $m$  be a size parameter and let  $M$  be an advised QTM running with an advice function  $h$ . Let  $r$  be a function on  $\mathbb{N}$  satisfying  $|h(n)| \leq r(n)$  for all  $n \in \mathbb{N}$ . For two functions  $p$  and  $\ell$ , there always exists a nonuniform family  $\{N_{n,l}\}_{n,l \in \mathbb{N}}$  of 2qfa's with  $O(r(l)) \cdot 2^{O(\ell(n,l))}$  inner states such that, for any input  $x$ , if  $M$  accepts (resp., rejects)  $(x, h(|x|))$  with bounded-error probability in expected  $p(m(x), |x|)$  time using at most  $\ell(m(x), |x|)$  space, then  $N_{m(x), |x|}$  accepts (resp., rejects)  $x$  with bounded-error probability in expected  $p(m(x), |x|)$  time.*

**Proof.** Let  $p$ ,  $\ell$ , and  $r$  denote respectively a time-bounding function, a space-bounding function, and an advice-bounding function. Let  $m$  be a given size parameter over an input alphabet  $\Sigma$  and let  $h$  be an advice function from  $\mathbb{N}$  to  $\Theta^*$  for a certain advice alphabet  $\Theta$  satisfying  $|h(n)| \leq r(n)$  for all  $n \in \mathbb{N}$ . We take an arbitrary advised QTM  $M = (Q, \Sigma, \{\epsilon, \$\}, \Gamma, \Theta, \Xi, \delta, q_0, Q_{acc}, Q_{rej})$  equipped with a work alphabet  $\Gamma$  and a garbage alphabet  $\Xi$ . Assume that, with the use of  $h$ ,  $M$  runs in expected  $p(m(x), |x|)$  time using at most  $\ell(m(x), |x|)$  space on any input  $x \in \Sigma^*$ . In what follows, we fix  $x$  and set  $a = h(|x|)$ . Note that, since  $p(m(x), |x|)$  is an expected runtime bound of  $M$  on  $x$ , for an appropriately chosen absolute constant  $c \geq 1$ , it is enough to consider only the first  $cp(m(x), |x|)$  steps of any computation path of  $M$  on  $x$  without losing  $M$ 's bounded-error probability criteria. Remember that the contents of

the input tape and the advice tape do not change during any computation. A *surface configuration* of  $M$  on  $x$  is of the form  $(q, j, k, y, t, z)$ , which indicates that  $M$  with garbage-tape content  $z$  is in inner state  $q$ , scanning the  $j$ th cell of an input tape, the  $k$ th cell of a work tape containing  $y$ , and the  $t$ th cell of an advice tape.

A basic idea is to maintain the partial information on the current surface configuration of  $M$  on  $x$  in the form of an inner state of the desired 2qfa. The desired nonuniform family  $\mathcal{N} = \{N_{n,l}\}_{n,l \in \mathbb{N}}$  of 2qfa's thus has the following form. Given an instance  $x$ , let  $n = m(x)$  and set  $N_{n,|x|} = (Q', \Sigma, \{\$, \$, \Xi, \delta', q'_0, Q'_{qcc}, Q'_{rej}\}$ . The set  $Q'$  consists of all inner states of the form  $(q, k, y, t)$ , where  $q \in Q$ ,  $k \in [0, \ell(n, |x|) + 1]_{\mathbb{Z}}$ ,  $y \in \Gamma^*$ , and  $t \in [0, |a| + 1]_{\mathbb{Z}}$ . It then follows that  $|Q'| \leq |Q|(\ell(n, |x|) + 1)(r(|x|) + 2)|\Gamma|^{\ell(n, |x|) + 1}$  since  $a$  is fixed and  $|a| \leq r(|x|)$ . In general, the state complexity of  $N_{n,l}$  is bounded by  $O(r(l)) \cdot 2^{O(\ell(n, l))}$ .

Let us define  $\delta'$  by way of describing how the 2qfa  $N_{n,|x|}$  operates. Notice that  $\delta'$  is a map from  $Q' \times \Sigma \times Q' \times D \times \Xi_{\lambda}$  to  $\mathbb{C}$ . Since  $M$  is supposed to use at most  $\ell(m(x), |x|)$  work tape cells, we use a series  $y_0 y_1 y_2 \cdots y_{\ell(m(x), |x|)}$  of  $\ell(n, |x|) + 1$  tape symbols to express the content  $y$  of  $M$ 's work tape starting with  $y_0 = \$$  (left endmarker). Let  $B$  indicate a blank of a tape cell. For each index  $n \in \mathbb{N}$ , we write  $\Sigma_n$  for the set  $\{x \in \Sigma^* \mid m(x) = n\}$ . The 2qfa  $N_{n,|x|}$  behaves as follows. Let  $y = y_0 y_1 \cdots y'_{\ell(m(x), |x|)}$  and  $y' = y'_0 y'_1 \cdots y'_{\ell(m(x), |x|)}$ .

On input  $x \in \Sigma_n$ ,  $N_{n,|x|}$  starts with the initial inner state  $(q_0, 0, \$, B^{\ell(n, |x|)}, 0)$ . Inductively, assume that  $M$  changes its surface configuration from  $(q, j, k, y, t, z)$  to  $(p, j', k', y', t', z\xi)$  by moving its input-tape head in direction  $d_1$ , its work-tape head in direction  $d_2$ , and its advice-tape head in direction  $d_3$ , and also by writing  $y'_k$  over  $y_k$  and changing  $z$  to  $z\xi$  on a garbage tape. Corresponding to this step,  $N_{n,|x|}$  moves its own input-tape head similarly, changes its inner state from  $(q, k, y, t)$  to  $(p, k', y', t')$ , and modifies  $z$  to  $z\xi$ . More formally, we define a (quantum) transition function  $\delta'$  of  $N_{n,|x|}$  by setting  $\delta'((q, k, y, t), x_j | (p, k', y', t'), d_1, \xi) = \delta(q, x_j, y_k, a_t | p, y'_k, d_1, d_2, d_3, \xi)$ , where  $k' = k + d_2$ ,  $t' = t + d_3$ , and  $rest_k(y) = rest_k(y')$ .

Notice that the above construction of  $N_{n,|x|}$  heavily depends on the fixed advice string  $a$ .

By the definition of  $N_{n,|x|}$ , on input  $x$ ,  $N_{n,|x|}$  correctly simulates all steps of  $M$  on  $(x, a)$  one by one and reaches the same outcome of  $M$ . Therefore, for any fixed constant  $\varepsilon \in [0, 1/2)$ , if  $M$  accepts (resp., rejects)  $(x, h(|x|))$  within  $p(n, |x|)$  time using at most  $\ell(n, |x|)$  space with error probability at most  $\varepsilon$ , then  $N_{n,|x|}$  also accepts (resp., rejects)  $x$  within  $p(n, |x|)$  time with error probability at most  $\varepsilon$ .

Since  $N_{n,|x|}$  simulates  $M$  with  $h$  precisely, if  $M$ 's configurations quantumly interfere with one another, then their associated  $N_{n,|x|}$ 's configurations interfere as well. Hence,  $N_{n,|x|}$  is indeed the desired 2qfa and the lemma then follows instantly.  $\square$

The converse of Lemma 5.2, Lemma 5.3, will be shown by giving a simulation of a nonuniform family of 2qfa's by appropriately chosen advised QTMs. In the proof of Lemma 5.3, nonetheless, in order to make a quantum interference take place correctly, we further need to avoid any time discrepancy caused by the different simulation speed along different computation paths and to adjust the timing of reaching the same configurations. For this purpose, we need to control the movement of an input-tape head of a 2qfa. We say that an automaton  $M$  *sweeps* a tape or  $M$  is a *sweeping* automaton if  $M$ 's tape head always moves rightward along a circular input tapes in one direction from  $\$$  to  $\$$ , and further to  $\$$ .

Another crucial point of the proof of Lemma 5.3 is how to encode all quantum transitions of a 2qfa into a single advice string for the purpose of performing such transitions easily from the information retrieved from this advice string.

**Lemma 5.3** *Let  $b$ ,  $r$ , and  $p$  be functions and let  $m$  be a size parameter satisfying  $m(x) \leq b(|x|)$  for all  $x$ . Let  $\{N_{n,l}\}_{n,l \in \mathbb{N}}$  denote a nonuniform family of  $r(n, l)$ -state 2qfa's over a fixed input alphabet  $\Sigma$  and a constant-size garbage alphabet. There exist an advised QTM  $M$  and an  $O(b(|x|)r(m(x), |x|)^9 \log^2 p(m(x), |x|))$ -bounded advice function  $h$  such that, for any input  $x$ , if  $N_{m(x), |x|}$  accepts (resp., rejects)  $x$  with bounded-error probability in expected  $p(m(x), |x|)$  time, then  $M$  accepts (resp., rejects)  $(x, h(|x|))$  with bounded-error probability within expected  $O(b(|x|)\tilde{r}(m(x), |x|)\tilde{p}(m(x), |x|))$  time using  $O(\log r(m(x), |x|))$  space, where  $\tilde{r}(n, l) = r(n, l)^9 \log r(n, l)$  and  $\tilde{p}(n, l) = p(n, l) \log^2 p(n, l)$ .*

**Proof.** Let  $b$ ,  $r$ ,  $p$ ,  $m$ ,  $\Sigma$ , and  $\{N_{n,l}\}_{n,l \in \mathbb{N}}$  be given as in the premise of the lemma. For each index  $n \in \mathbb{N}$ , we set  $\Sigma_n = \{x \in \Sigma^* \mid m(x) = n\}$ . Take an arbitrary input  $x \in \Sigma^*$  and let  $n = m(x)$  for brevity.

Let  $N_{n,|x|}$  have the form  $(Q_n, \Sigma, \{\$, \#\}, \Xi, \delta_n, q_{n,0}, Q_{n,qcc}, Q_{n,rej})$ . We want to define the desired advised QTM  $M$ , together with an appropriate advice function  $h$ , which can simulate  $N_{n,|x|}$  on all inputs  $x$  in  $\Sigma_n$  for any index  $n \in \mathbb{N}$ . For our convenience, we identify  $Q_n$  with the set  $\{0,1\}^{r_1(n,|x|)}$  and  $\Xi_n$  with  $\{0,1\}^{r_2}$  for a certain function  $r_1$  and a certain constant  $r_2 \in \mathbb{N}^+$ ; thus, we obtain  $r_1(n,|x|) = \log |Q_n|$  ( $= \log r(n,|x|)$ ) and  $r_2 = \log |\Xi_n|$ , where  $\Xi_n = \Xi \cup \{\lambda\}$ .

The transition function  $\delta_n$  of  $N_{n,l}$  can be viewed as a *transition table*  $T_{n,l}$ , as explained in Section 2.3, which is a matrix in which each  $(q, \sigma)$ -row contains a “description” of a quantum circuit  $C_{q,\sigma}^{(n,l)}$  that takes  $|\phi_0\rangle = |0^{r_1(n,|x|)}\rangle|00\rangle|0^{r_2}\rangle$  as an input and approximates to within  $2^{-cp(n,l)}$  a quantum state  $\sum_{(p,d,\xi)} \delta_n(q, \sigma|p, d, \xi)|p, d, \xi\rangle$ . Let us recall from Section 2.3 the encoding scheme of such a transition table into an appropriate binary string. Using this encoding scheme, we encode  $T_{n,l}$  into an advice string  $\langle T_{n,l} \rangle$  of  $O(|Q|^9 |\Xi_n|^8 \log^2 p(n, l)) \subseteq O(r(n, l)^9 \log^2 p(n, l))$  symbols, from which we can easily retrieve the necessary information on any transition  $\delta_n(q, \sigma|p, d, \xi) = \alpha$ .

For each number  $l$  in  $\mathbb{N}$ , we define the desired advice function  $h$  as  $h(l) = 1^1 \# \langle T_{1,l} \rangle \# 1^2 \# \langle T_{2,l} \rangle \# 1^3 \# \langle T_{3,l} \rangle \# \dots \# 1^{\bar{n}_l} \# \langle T_{\bar{n}_l,l} \rangle$ , where  $\bar{n}_l = \max\{m(z) \mid z \in \Sigma^*, |z| = l\}$ . Since  $\bar{n}_l \leq b(l)$ , the length of  $h(l)$  is bounded by  $O(b(l)r(n, l)^9 \log^2 p(n, l))$ .

The following procedure briefly describes the behavior of  $M$  with  $h$ .

On input  $(x, h(|x|))$ ,  $M$  first computes  $n = m(x)$ , writes down the string  $\$q_{n,0}\$$  on its work tape by sweeping the tape, and tries to simulate  $N_{n,|x|}$  on  $x$  as follows. Assume that currently  $N_{n,|x|}$  is in inner state  $q$  scanning the  $j$ th input tape cell, changes  $q$  to  $p$ , moves its input-tape head in direction  $d$ , and dumps  $\xi$  onto a garbage tape with amplitude  $\delta_n(q, x_j|p, d, \xi)$ . By sweeping the work tape from  $\$$  to  $\$$ ,  $M$  searches for the  $(q, x_j)$ -row, and reads its entry  $\langle C_{q,x_j}^{(n,|x|)} \rangle$  symbol by symbol by performing each quantum gate constituting  $C_{q,x_j}^{(n,|x|)}$  to generate a quantum state  $\sum_{(p',d',\xi')} \delta(q, x_j|p', d', \xi')|p', d', \xi'\rangle$ . If an entry  $(p, d, \xi)$  appears in this quantum state, then  $M$  overwrites the work tape by  $p$ , moves the input-tape head in direction  $d$ , and writes down  $\xi$  onto the garbage tape. To simulate one step of  $N_{n,|x|}$ ,  $M$  needs to sweep the advice tape once and sweep the work tape at most  $|h(|x|)|$  times. Note that  $M$  keeps its input-tape head at a standstill during each sweeping process.

A quick analysis of the above description shows that the space usage of  $M$ 's work tape is bounded by  $O(r_1(m(x), |x|)) \subseteq O(\log r(m(x), |x|))$  and  $M$ 's expected runtime is bounded by  $O(p(m(x), |x|)r_1(m(x), |x|)|h(|x|)|) \subseteq O(b(|x|)r(m(x), |x|) \log r(m(x), |x|)p(m(x), |x|) \log^2 p(m(x), |x|))$ . Moreover, it is not difficult to verify that, if  $N_{n,|x|}$  accepts (resp., rejects)  $x$  with bounded-error probability, then  $M$  accepts (resp., rejects)  $(x, h(|x|))$  with bounded-error probability.  $\square$

The following lemma relates to the polynomial honesty condition of size parameters. This condition will become quite essential in the proof of Proposition 5.1.

**Lemma 5.4** *Let  $(L, m)$  and  $(K, m')$  be two parameterized decision problems and let  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  be any family of promise decision problems. Let  $\mathcal{C}$  denote an arbitrary nonempty nonuniform state complexity class.*

1. *If  $\mathcal{L}$  is induced from  $(L, m)$  and  $\mathcal{L} \in \mathcal{C}/\text{poly}$ , then  $m$  is polynomially honest.*
2. *If  $(K, m')$  is induced from  $\mathcal{L}$  and  $\mathcal{L} \in \mathcal{C}/\text{poly}$ , then  $m'$  is polynomially honest.*

**Proof.** (1) Assume that  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  is induced from  $(L, m)$  and that  $\mathcal{L}$  is in  $\mathcal{C}/\text{poly}$ . Since  $\mathcal{L}$  has a polynomial ceiling, there exists a polynomial  $r$  such that, for any index  $n \in \mathbb{N}$  and for any input  $x$  in  $L_n^{(+)} \cup L_n^{(-)}$ ,  $|x| \leq r(n)$  holds. In the case of  $x \in L_n^{(+)} \cup L_n^{(-)}$ , since  $m(x) = n$  by the definition of  $(L_n^{(+)}, L_n^{(-)})$ , we obtain  $|x| \leq r(m(x))$ . Thus,  $m$  is polynomially honest.

(2) Similarly to (1), from  $\mathcal{L} \in \mathcal{C}/\text{poly}$ , we obtain a polynomial  $r$  satisfying  $|x| \leq r(n)$  for all  $n \in \mathbb{N}$  and all  $x \in L_n^{(+)} \cup L_n^{(-)}$ . Recall from Section 2.5 that  $K_n^{(+)} = \{1^n \# x \mid x \in L_n^{(+)}\}$  and  $K_n^{(-)} = \{1^n \# x \mid x \in L_n^{(-)}\} \cup \{z \# x \mid z \in \Sigma^n - \{1^n\}, x \in \Sigma_n^*\} \cup \Sigma^n$ , where  $\Sigma$  is a fixed input alphabet. We define  $s(n) = n + r(n) + 1$  for each number  $n \in \mathbb{N}$ . Consider the case where  $w$  is of the form  $1^n \# x$  for  $x \in L_n^{(+)} \cup L_n^{(-)}$ . It then follows that  $|w| = |1^n \# x| \leq n + |x| + 1 \leq n + r(n) + 1 \leq s(m'(w))$  because of

$m'(w) = n$ . In any other case, we obtain  $|w| = m'(w)$  by the definition of  $m'$ , and thus  $|w| \leq s(m'(w))$  follows. Therefore,  $m'$  is polynomially honest.  $\square$

With the help of the supporting lemmas, Lemmas 5.2–5.4, we provide the proof of Proposition 5.1(1).

**Proof of Proposition 5.1(1).** Hereafter, we intend to present the desired proof only for the case where  $\mathcal{A} = \text{BQ}$  and  $\mathcal{B} = \text{BQL}$ . Let  $(L, m)$  be any parameterized decision problem and let  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  be the family of promise decision problems induced from  $(L, m)$ . For convenience, we abbreviate  $L_n^{(+)} \cup L_n^{(-)}$  as  $\Sigma_n$  for each index  $n \in \mathbb{N}$ . By the definition of  $\mathcal{L}$ ,  $\Sigma_n$  equals the set  $\{x \in \Sigma^* \mid m(x) = n\}$ .

(If-part) Assuming that  $\mathcal{L} \in \text{ptime-2BQ/poly}$ , we take polynomials  $p$  and  $r$ , and a family  $\mathcal{N} = \{N_n\}_{n \in \mathbb{N}}$  of 2qfa's having  $r(n)$  inner states such that, for every index  $n \in \mathbb{N}$ ,  $N_n$  solves  $(L_n^{(+)}, L_n^{(-)})$  with bounded-error probability in expected  $p(n, |x|)$  time on all inputs  $x$  in  $\Sigma_n$ . We define a new 2qfa  $N'_{n,l}$  simply as  $N_n$  restricted to all inputs  $x$  of length  $l$ . Take a polynomial  $b$  satisfying  $m(x) \leq b(|x|)$  for all  $x$ . Since  $m$  is polynomially honest by Lemma 5.4(1), there is a polynomial  $q$  for which  $|x| \leq q(|x|)$  holds for all  $x$ . therefore,  $(L, m)$  belongs to PHSP.

From Lemma 5.3, it follows that there exist an  $O(b(|x|)r(m(x))^9 \log p(m(x), |x|))$ -bounded advice function  $h$  and an advised QTM  $M$  such that, for an arbitrary string  $x$ ,  $M$  on the input  $(x, h(|x|))$  simulates  $N'_{m(x), |x|}$  on  $x$  using space  $O(\log r(m(x))) \subseteq O(\log m(x))$ . In other words,  $M(x, h(|x|))$  computes  $L_{m(x)}(x)$  with bounded-error probability. Since  $L_{m(x)} = L_n$  for all  $x \in \Sigma_n$ ,  $M(x, h(|x|))$  actually computes  $L(x)$  for all strings  $x$  in  $\bigcup_{n \in \mathbb{N}} \Sigma_n$ . Since  $\Sigma^* = \bigcup_{n \in \mathbb{N}} \Sigma_n$ ,  $(L, m)$  belongs to para-BQL/poly.

(Only if-part) Assume that  $(L, m)$  is in para-ptime-BQL/poly  $\cap$  PHSP. Since  $(L, m) \in \text{PHSP}$ ,  $m$  is polynomially honest. There exist an advised QTM  $M$  and an  $r(|x|)$ -bounded advice function  $h$  satisfying the following: for all inputs  $x$ ,  $M(x, h(|x|))$  computes  $L(x)$  with bounded-error probability in expected  $p(m(x), |x|)$  time using  $O(\log m(x))$  work space. Since  $m$  is polynomially honest, we take a constant  $c > 0$  satisfying  $|x| \leq m(x)^c + c$  for all  $x$ . With this value  $c$ , given any index  $n \in \mathbb{N}$ , we write  $\tilde{\Sigma}_n$  for the set  $\{x \in \Sigma^* \mid |x| \leq n^c + c\}$ .

Lemma 5.2 then ensures the existence of a nonuniform family  $\mathcal{N} = \{N_{n,l}\}_{n \in \mathbb{N}}$  of 2qfa's satisfying the following condition: for any given string  $x \in \tilde{\Sigma}_n$ ,  $N_{m(x), |x|}$  on  $x$  simulates  $M$  on  $(x, h(|x|))$  and the state complexity of  $N_{m(x), |x|}$  is at most  $O(r(|x|)) \cdot 2^{O(\log m(x))}$ , which is upper-bounded by  $|x|^{O(1)} \cdot 2^{O(\log m(x))} \subseteq m(x)^{O(1)}$  because of  $|x| \leq m(x)^c + c$ . For an arbitrary index  $n \in \mathbb{N}$ , a new machine  $N'_n$  is defined to behave as follows. Given any input  $x$ , we first set  $l = |x|$  if  $x \in \tilde{\Sigma}_n$  and  $l = n$  otherwise. This can be done by sweeping an input tape once using additional  $n^{O(1)}$  inner states. We then run  $N_{n,l}$  on  $x$ . In particular, when  $m(x) = n$ ,  $N'_n$  can simulate  $N_{n, |x|}$  on every input  $x$  in  $\tilde{\Sigma}_n$ . Since  $\mathcal{N}$  simulates  $M$  with  $h$ ,  $N'_n$  correctly computes  $L(x)$  on all inputs  $x$  in  $\tilde{\Sigma}_n$  in expected  $O(p(n, |x|))$  time. Since  $N'_n$  uses  $n^{O(1)}$  states, we conclude that  $\mathcal{L}$  falls in ptime-2BQ/poly.  $\square$

Next, we wish to prove Proposition 5.1(2), which deals with the parameterized decision problem  $(K, m)$  induced from each L-good family  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems. Notice that, by the definition of  $K$ , for any  $n$  and  $x$ ,  $1^n \# x \in K$  (resp.,  $1^n \# x \in \bar{K}$ ) iff  $x \in L_n^{(+)}$  (resp.,  $x \in L_n^{(-)}$ ). For the proof of the proposition, we need two more lemmas, Lemmas 5.5–5.6, which look analogous to Lemmas 5.2–5.3.

**Lemma 5.5** *Let  $M$  be an advised QTM, let  $r$  and  $\ell$  be two functions, and let  $h$  be an advice function with  $|h(n)| \leq r(n)$  for all  $n \in \mathbb{N}$ . There exists a nonuniform family  $\{N_{n,l}\}_{n,l \in \mathbb{N}}$  of  $O(r(l)) \cdot 2^{O(\ell(n,l))}$ -state 2qfa's such that, for any  $n$  and  $x$ , if  $M$  accepts (resp., rejects)  $(1^n \# x, h(|x|))$  with bounded-error probability in expected  $p(n, |x|)$  time using at most  $\ell(n, |x|)$  space, then  $N_{n, |x|}$  accepts (resp., rejects)  $x$  with bounded-error probability in expected  $O(p(n, |x|))$  time.*

**Proof.** Let  $r$ ,  $h$ ,  $p$ , and  $M$  be given as in the premise of the lemma. Let  $\Sigma$  denote an input alphabet used by  $M$ . Slightly different from the proof of Lemma 5.2, we construct the desired family  $\{N_{n,l}\}_{n,l \in \mathbb{N}}$  of 2qfa's to work as follows. Given an index  $n \in \mathbb{N}$  and an input string  $x \in \Sigma^*$ ,  $N_{n, |x|}$  generates both  $1^n \# x$  and  $h(|x|)$  and then runs  $M$  on  $(1^n \# x, h(|x|))$  to produce an outcome. This behavior of  $N_{n, |x|}$  is possible because  $n$  and  $|x|$  are fixed for  $N_{n, |x|}$  and we can store  $h(|x|)$  in the form of inner states. Clearly,  $N_{n, |x|}$  correctly outputs  $M(1^n \# x, h(|x|))$  on all inputs  $x$  in expected  $O(p(n, |x|))$  time since  $M$  halts on  $(1^n \# x, h(|x|))$  in expected  $p(n, |x|)$  time.  $\square$

**Lemma 5.6** *Let  $r$  be any function and let  $\{N_{n,l}\}_{n,l \in \mathbb{N}}$  denote any nonuniform family of 2qfa's with  $r(n,l)$  inner states. There is an advised QTM  $M$  and an  $O(nr(n, |x|)^9 \log^2 p(n, |x|))$ -bounded advice function  $h$  such that, for any  $x$ , if  $N_{n,|x|}$  accepts (resp., rejects)  $x$  with bounded-error probability in expected  $p(n, |x|)$  time, then  $M$  accepts (resp., rejects)  $(1^n \# x, h(|x|))$  with bounded-error probability in expected  $O(nr(n, |x|)^9 \tilde{p}(n, |x|))$  time using  $O(\log r(n, |x|))$  space, where  $\tilde{p}(n, l) = p(n, l) \log^2 p(n, l)$ .*

**Proof.** Let  $r$  and  $\{N_{n,l}\}_{n,l \in \mathbb{N}}$  be given as in the premise of the lemma. With a construction similar to the proof of Lemma 5.3, we define the desired advised QTM  $M$  that behaves as follows in the presence of an appropriate advice function  $h$ . On input  $w$ , we first check whether  $w$  is of the form  $1^n \# x$  for certain  $n$  and  $x$ . If not, then we reject  $w$  instantly. Assuming that  $w = 1^n \# x$ , we compute  $|x|$ , retrieve the description of  $N_{n,|x|}$  from  $h(|x|)$  and run  $N_{n,|x|}$  on  $x$ . It is not difficult to see that  $M$  accepts (resp., rejects)  $(1^n \# x, h(|x|))$  with bounded-error probability iff  $N_{n,|x|}$  accepts (resp., rejects)  $x$  with bounded-error probability. The expected runtime and the space usage of  $M$  can be estimated similarly to the proof of Lemma 5.3.  $\square$

The proof of Proposition 5.1(2) is in essence similar to the proof of Proposition 5.1(1) except for the use of  $(K, m)$  that is induced from a given family  $\mathcal{L}$  of promise decision problems.

**Proof of Proposition 5.1(2).** Let  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  be any L-good family of promise decision problems and let  $(K, m)$  denote the parameterized decision problem induced from  $\mathcal{L}$ . For convenience, we write  $\Theta$  for the set  $\{1^n \# x \mid n \in \mathbb{N}, x \in L_n^{(+)} \cup L_n^{(-)}\}$ . Since  $\Theta$  is in L, by the definition of  $(K, m)$ ,  $m$  is a log-space size parameter, and thus it is polynomially bounded. The following proof is meant for the case of  $\mathcal{A} = \text{BQ}$  and  $\mathcal{B} = \text{BQL}$  in accordance with the proof of Proposition 5.1(1).

(If-part) Assume that  $\mathcal{L}$  is in ptime-2BQ/poly. There exist two polynomials  $p$  and  $r$  as well as a nonuniform family  $\mathcal{N} = \{N_n\}_{n \in \mathbb{N}}$  of  $O(r(n))$ -state 2qfa's solving  $\mathcal{L}$  in expected  $O(p(n, |x|))$  time with bounded-error probability. By Lemma 5.4(2),  $m$  is polynomially honest, and thus the problem  $(K, m)$  belongs to PHSP.

For the 2qfa family  $\mathcal{N}$ , Lemma 5.6 guarantees the existence of an advised QTM  $M$  and an  $O(nr(n)^9 \log^2 p(n, |x|))$ -bounded advice function  $h$  such that  $M$  on input  $(1^n \# x, h(|x|))$  simulates  $N_{n,|x|}$  on  $x$  in expected  $O(nr(n, |x|)^9 \log^2 p(n, |x|))$  time using  $O(\log r(n, |x|))$  space. We further modify  $M$  so that it first checks whether an input  $w$  of the form  $1^n \# x$  belongs to  $\Theta$  using logarithmic space; if not, it immediately rejects the input. This modified machine makes bounded-error probability on all inputs  $w$  and thus recognizes  $K$  with only additional  $O(\log |w|)$  space. Since  $O(\log r(n, |x|) + \log |w|) \subseteq O(\log m(w))$ ,  $(K, m)$  belongs to para-ptime-BQL/poly.

(Only if-part) Assuming that  $(K, m) \in \text{para-ptime-BQL/poly} \cap \text{PHSP}$ , we take an advised QTM  $M$  and an  $r(|w|)$ -bounded advice function  $h$  for which  $M$  solves  $(K, m)$  using  $h$  with bounded-error probability in expected  $p(m(w), |w|)$  time using at most  $\ell(m(w))$  space for any input  $w$ , where  $p$  and  $\ell$  are respectively a polynomial and a logarithmic function. Additionally, we assume that  $\ell$  is nondecreasing. The polynomial honesty of  $m$  comes from the assumption of  $(K, m) \in \text{PHSP}$ . Let us take a polynomial  $q$  satisfying  $|w| \leq q(m(w))$  for any  $w$ . Note that, for any  $x \in L_n^{(+)} \cup L_n^{(-)}$ ,  $|x| \leq |1^n \# x| \leq q(m(1^n \# x)) = q(n)$  since  $m(1^n \# x) = n$ . Moreover, if  $w = 1^n \# x$ , then it follows that  $p(m(w), |w|) \leq p(n, q(n))$  and  $\ell(m(w), |w|) \leq \ell(n, q(n))$ .

Lemma 5.5 then provides a nonuniform family  $\mathcal{N} = \{N_{n,l}\}_{n,l \in \mathbb{N}}$  of 2qfa's having  $O(r(|x|)) \cdot 2^{O(\ell(n, q(n)))}$  inner states for which each  $N_{n,|x|}$  on input  $x$  simulates  $M$  on  $(1^n \# x, h(|x|))$  correctly in expected  $O(p(n, q(n)))$  time. Thus,  $\mathcal{N}$  solves  $\mathcal{L}$  with bounded-error probability. We note that  $O(p(n, q(n))) \subseteq n^{O(1)}$  and  $2^{O(\ell(n, q(n)))} \subseteq n^{O(1)}$ . This immediately concludes that  $\mathcal{L}$  belongs to ptime-2BQ/poly.  $\square$

## 5.2 Proof of Theorem 1.1

Finally, we are ready to describe the proof of Theorem 1.1. We have already proven a key claim, Proposition 5.1, in Section 5.1. For the intended proof of the theorem, however, we still need two more supporting claims regarding polynomial-size advice.

**Lemma 5.7** *Let  $(\mathcal{A}, \mathcal{B}) \in \{(\text{NL}, \text{L}), (\text{NL}, \text{BPL}), (\text{NL}, \text{BQL}), (\text{BQL}, \text{BPL})\}$ . It then follows that  $\mathcal{A}/\text{poly} \subseteq \mathcal{B}/\text{poly}$  iff  $\mathcal{A} \subseteq \mathcal{B}/\text{poly}$ . The same holds even if the expected runtime of underlying Turing machines is*

limited to polynomials.

**Proof.** In what follows, we intend to show the lemma only for the case of  $\mathcal{A} = \text{NL}$  and  $\mathcal{B} = \text{BQL}$  since the other cases are similarly proven.

(Only If-part) The implication of  $\text{NL/poly} \subseteq \text{BQL/poly}$  to  $\text{NL} \subseteq \text{BQL/poly}$  is obviously true because  $\text{NL}$  is included in  $\text{NL/poly}$ .

(If-part) Assume that  $\text{NL} \subseteq \text{BQL/poly}$ . Our goal is to verify that  $\text{NL/poly} \subseteq \text{BQL/poly}$ . Let us focus on an arbitrary language  $L$  in  $\text{NL/poly}$  and take a polynomial  $p$  and a log-space NTM  $M$  satisfying the following two conditions for every input  $x$ : (i)  $|h(|x|)| \leq p(|x|)$  and (ii)  $N$  accepts  $(x, h(|x|))$  iff  $x$  is in  $L$ . We define a new language  $K$  as  $K = \{(x, s) \mid |s| \leq p(|x|), N \text{ accepts } (x, s)\}$ . Since  $K$  clearly belongs to  $\text{NL}$ , our assumption implies that  $K \in \text{BQL/poly}$ . We then take a log-space advised QTM  $M$  that recognizes  $K$  with bounded-error probability using an appropriate polynomially-bounded advice function  $g$ ; that is,  $M(x, s, g(|x|, |s|))$  computes  $K(x, s)$  with bounded-error probability for any pair  $(x, s)$ . Let us define a new advice function  $r$  by setting  $r(n) = \langle h(n), g(n, |h(n)|) \rangle$  and design a new QTM  $\tilde{N}$  so that it starts with  $(x, r(|x|))$  and simulates  $M$  on  $(x, h(|x|), g(|x|, |h(|x|)|))$ . Clearly,  $\tilde{N}$  with  $r$  recognizes  $L$  with bounded-error probability. Take a polynomial  $k$  satisfying  $|g(n, n')| \leq k(n, n')$  for any pair  $n, n' \in \mathbb{N}$ . Note that  $|r(n)|$  is  $O(|h(n)| + |g(n, |h(n)|)|)$ , which is included in  $O(p(n)k(n, p(n)))$ . Since  $\tilde{N}$  is a bounded-error advised QTM for  $L$  with the polynomially-bounded advice function  $r$ ,  $L$  belongs to  $\text{BQL/poly}$ . Therefore, we conclude that  $\text{NL/poly} \subseteq \text{BQL/poly}$ , as requested.  $\square$

Another claim stated below connects between parameterized complexity classes and standard complexity classes under the presence of advice.

**Lemma 5.8** *Let  $(\mathcal{A}, \mathcal{B}) \in \{(\text{NL}, \text{L}), (\text{NL}, \text{BPL}), (\text{NL}, \text{BQL}), (\text{BQL}, \text{BPL})\}$ . It then follows that  $\text{para-}\mathcal{A}/\text{poly} \cap \text{PHSP} \subseteq \text{para-}\mathcal{B}/\text{poly}$  iff  $\mathcal{A}/\text{poly} \subseteq \mathcal{B}/\text{poly}$ . The same holds even if the expected run-time of underlying Turing machines is limited to polynomials.*

Recall that, by our convention stated in Section 1.3,  $\text{para-ptime-NL/poly}$  and  $\text{ptime-2N/poly}$  mean  $\text{para-NL/poly}$  and  $2\text{N/poly}$ , respectively.

**Proof of Lemma 5.8.** Hereafter, we are focused only on the case where  $\mathcal{A} = \text{NL}$  and  $\mathcal{B} = \text{BQL}$  in accordance with the proof of Lemma 5.7.

(Only If-part) We begin with assuming that  $\text{para-NL/poly} \cap \text{PHSP} \subseteq \text{para-BQL/poly}$ . Let us take the binary size parameter  $m_{\text{bin}}(x) = |x|$  defined for all strings  $x$  and consider any parameterized decision problem  $(L, m_{\text{bin}})$  satisfying  $L \in \text{NL/poly}$ . Notice that  $m_{\text{bin}}$  is polynomially honest. Since  $(L, m_{\text{bin}}) \in \text{para-NL/poly} \cap \text{PHSP}$ , our assumption concludes that  $(L, m_{\text{bin}}) \in \text{para-BQL/poly}$ . This is logically equivalent to  $L \in \text{BQL/poly}$  by the definition of  $m_{\text{bin}}$ . Since  $L$  is arbitrary, we conclude that  $\text{NL/poly} \subseteq \text{BQL/poly}$ .

(If-part) On the contrary, assume that  $\text{NL/poly} \subseteq \text{BQL/poly}$ . Let  $(L, m)$  be any parameterized decision problem in  $\text{para-NL/poly} \cap \text{PHSP}$ . By the definition,  $m$  is a log-space size parameter, and thus we can deterministically compute  $1^{m(x)}$  from  $x$  using  $O(\log |x|)$  space. Since  $m$  is polynomially bounded, an appropriately chosen polynomial  $q$  can satisfy  $m(x) \leq q(|x|)$  for all strings  $x$ . The polynomial honesty of  $m$  also guarantees the existence of a constant  $c \geq 0$  satisfying  $|x| \leq m(x)^c + c$  for all  $x$ . Since  $(L, m) \in \text{para-NL/poly}$ , there exists an NTM  $N$  and a polynomially-bounded advice function  $k$  such that  $N$  with  $k$  solves  $L$  using  $O(\ell(m(x)))$  space for a certain logarithmic function  $\ell$ . Without loss of generality, we can assume that  $\ell$  is nondecreasing.

Let us define  $K' = \{(x, 1^t) \mid x \in L, t \in \mathbb{N}, m(x) \leq t\}$ . We wish to claim that  $K'$  belongs to  $\text{NL/poly}$ . To verify this claim, let us consider a new NTM that behaves as follows. On input  $w$  of the form  $(x, 1^t)$ , firstly check whether  $m(x) \leq t$ . If so, then output the value  $L(x)$ ; otherwise, immediately reject  $x$ . With the help of the advice string  $k(|x|)$ , we can check that “ $x \in L$ ?” using space  $O(\ell(m(x))) \subseteq O(\ell(q(|w|)))$  and also check that “ $m(x) \leq t$ ?” using space  $O(\log |x| + \log t) \subseteq O(\log |w|)$ . Hence, this NTM requires only  $O(\log |w|)$  space. From this result, we conclude that  $K'$  belongs to  $\text{NL/poly}$ .

Since  $K' \in \text{NL/poly}$ , our assumption further implies that  $K' \in \text{BQL/poly}$ . This provides us with a logarithmic function  $\ell'$ , a polynomially-bounded advice function  $h$ , and an advised QTM  $M$  that recognizes  $\{(x, h(|x|)) \mid x \in K'\}$  with bounded-error probability using at most  $\ell'(|x|)$  space. Finally, we design a new machine  $N$  that behaves as follows: on input  $x$ , compute  $n = m(x)$  and run  $M$  with  $h$  on

$(x, 1^n)$ . Note that  $N$  is also an advised QTM and runs using space  $O(\ell'(|x|) + \log |x|) \subseteq O(\log m(x))$  since  $|x| \leq m(x)^c + c$ . It thus follows that  $(L, m)$  is in para-BQL/poly.  $\square$

Let us present the proof of Theorem 1.1, which is now a relatively easy consequence of Lemmas 5.7–5.8 and Proposition 5.1.

**Proof of Theorem 1.1.** In what follows, we intend to show only the case of  $\mathcal{A} = \text{N}$  and  $\mathcal{B} = \text{BQ}^\dagger$  because the other cases can be proven in a similar manner.

(If-part) Firstly, we assume that  $\text{NL} \subseteq \text{ptime-BQL/poly}$ . Since this assumption is, by Lemma 5.7, logically equivalent to  $\text{NL/poly} \subseteq \text{ptime-BQL/poly}$ , Lemma 5.8 implies that  $\text{para-NL/poly} \cap \text{PHSP} \subseteq \text{para-ptime-BQL/poly}$ . Using this last inclusion, it suffices for us to verify that  $2\text{N/poly} \subseteq \text{ptime-2BQ}^\dagger$ .

For our purpose, let us consider an arbitrary family  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  of promise decision problems in  $2\text{N/poly}$ . Since  $2\text{N/poly}$  is L-good, we can take an L-good extension  $\hat{\mathcal{L}} = \{(\hat{L}_n^{(+)}, \hat{L}_n^{(-)})\}_{n \in \mathbb{N}}$  of  $\mathcal{L}$  in  $2\text{N/poly}$ . We write  $\Sigma_n$  for  $\hat{L}_n^{(+)} \cup \hat{L}_n^{(-)}$  for each index  $n \in \mathbb{N}$ . There exists a nonuniform family  $\{M_n\}_{n \in \mathbb{N}}$  of 2nfa's with polynomially many inner states such that, for any index  $n \in \mathbb{N}$ ,  $M_n$  solves  $(\hat{L}_n^{(+)}, \hat{L}_n^{(-)})$  on all inputs in  $\Sigma_n$  with bounded-error probability. Consider the parameterized decision problem  $(K, m)$  induced from  $\hat{\mathcal{L}}$ . Since  $\hat{\mathcal{L}} \in 2\text{N/poly}$ , by Lemma 5.4(2),  $m$  is polynomially honest. We apply Proposition 5.1(2) and then obtain  $(K, m) \in \text{para-NL/poly} \cap \text{PHSP}$ , from which our assumption further places  $(K, m)$  in  $\text{para-ptime-BQL/poly}$ . Proposition 5.1(2) then concludes that  $\hat{\mathcal{L}}$  belongs to  $\text{ptime-2BQ/poly}$ . Since  $\hat{\mathcal{L}}$  is an L-good extension of  $\mathcal{L}$ ,  $\mathcal{L}$  must belong to  $\text{ptime-2BQ}^\dagger/\text{poly}$ . Because  $\text{ptime-2BQ}^\dagger/\text{poly} \subseteq \text{ptime-2BQ}^\dagger$ , the desired inclusion  $2\text{N/poly} \subseteq \text{ptime-2BQ}^\dagger$  follows immediately.

(Only If-part) On the contrary, we assume that  $2\text{N/poly} \not\subseteq \text{ptime-2BQ}^\dagger$ . This implies that  $2\text{N/poly} \not\subseteq \text{ptime-2BQ}^\dagger/\text{poly}$ . It therefore suffices to prove that  $\text{para-NL/poly} \cap \text{PHSP} \subseteq \text{para-ptime-BQL/poly}$  because, once this is proven, Lemma 5.8 implies that  $\text{NL/poly} \subseteq \text{ptime-BQL/poly}$  and Lemma 5.7 further concludes that  $\text{NL} \subseteq \text{ptime-BQL/poly}$ .

Let us take any parameterized decision problem  $(L, m)$  in  $\text{para-NL/poly} \cap \text{PHSP}$ . Let  $\mathcal{L}$  denote the family of promise decision problems induced from  $(L, m)$ . Proposition 5.1(1) implies that  $\mathcal{L} \in 2\text{N/poly}$ . Our assumption then leads to the conclusion that  $\mathcal{L} \in \text{ptime-2BQ}^\dagger/\text{poly}$ . It then follows by Proposition 5.1(1) that  $(L, m)$  belongs to  $\text{para-ptime-BQL/poly}$ .  $\square$

From Theorem 1.1, Corollary 1.2 follows immediately. This theorem also leads to the main result of [21, Theorem 1.1] (shown here as Corollary 5.9) whose proof relies on the property of a particular NL-complete problem, called the two-way liveness problem [32]. Our key argument in the proof of Proposition 5.1, however, uses the parameterized complexity classes para-L and para-NL as well as their properties, which are not directly relying on any particular NL-complete problem.

**Corollary 5.9** [21]  $2\text{N/poly} \subseteq 2\text{D}$  iff  $\text{NL} \subseteq \text{L/poly}$ .

## 6 Quantum Advice and Quantum Transition Tables

Since Theorem 1.3 concerns *quantum advice*, for the proof of the theorem, we first examine the basic properties of quantum advice. It was shown in [30, Lemma 3.1] that a polynomial-time QTM with quantum advice can be translated into an “equivalent” polynomial-size quantum circuit family starting with additional quantum states. In the case of quantum finite automata, nevertheless, we rather intend to quantize transition tables into “superpositions” of such transition tables and feed them to quantum finite automata so that different transition tables may regulate different behaviors of quantum finite automata. Firstly, we need to clarify the notion of a *superposition of transition tables* or, more succinctly, a *quantum transition table*. As discussed in Section 2.3, a transition table  $T$  of a 2qfa can be encoded into a binary string  $\langle T \rangle$ , from which an appropriately designed 2qfa can generate from  $(q, \sigma)$  a quantum circuit  $C_{q, \sigma}^{(n, l)}$  representing a target transition and approximately execute this transition by simply running  $C_{q, \sigma}^{(n, l)}$  on the quantum state  $|q, \sigma\rangle$ . To handle various transition tables  $T_1, T_2, \dots, T_k$  at once, we can create a superposition  $|\phi\rangle$  of the form  $\sum_{i=1}^k \alpha_i |T_i\rangle$  with the use of appropriately chosen amplitudes  $\{\alpha_i\}_{i \in [k]}$  satisfying  $\sum_{i \in [k]} |\alpha_i|^2 = 1$ . For technical reason, we further “encode” such a superposition  $|\phi\rangle$  of transition tables into another superposition  $|\phi^{(code)}\rangle = \sum_{i \in [k]} \alpha_i |\langle T_i \rangle\rangle$ . For convenience, we define the *encoded length* of  $|\phi\rangle$  to be  $\max_{i \in [k]} |\langle T_i \rangle|$ .



To execute a quantum transition table, we also need to expand our underlying 2qfa's to "super" 2sqfa's. A *2-way super quantum finite automaton with a flexible garbage tape* (abbreviated as a 2sqfa) is an octuple  $(Q, \Sigma, \{\$, \$, \}, \Xi, |\phi\rangle, q_0, Q_{acc}, Q_{rej})$ , where  $|\phi\rangle$  is a quantum transition table and the rest is similar to a 2qfa. A 2sqfa takes an input and, starting with the inner state  $|q_0\rangle$ , makes transitions described by the quantum transition table  $|\phi\rangle$ . Similar to 2qfa's, at every step, the 2sqfa must observe its inner state to check whether or not it is in a halting state. We further consider a nonuniform family  $\{M_n\}_{n \in \mathbb{N}}$  of such 2sqfa's.

The notation ptime-2sBQ denotes the collection of all nonuniform families of promise decision problems, each family of which is solved with bounded-error probability in expected polynomial time by a certain nonuniform family of 2sqfa's having polynomially many inner states. By restricting our interest only on inputs having polynomial ceilings, we obtain ptime-2sBQ/poly from ptime-2sBQ in a way similar to getting ptime-2BQ/poly from ptime-2BQ.

Our purpose is to give the proof of Theorem 1.3; for this purpose, we wish to prepare two key statements, Proposition 6.1 and Lemma 6.2.

**Proposition 6.1** *Let  $(L, m)$  and  $(K, m')$  be two parameterized decision problems and let  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  denote a nonuniform family of promise decision problems.*

1. *Assume that  $\mathcal{L}$  is induced from  $(L, m)$ . It then follows that  $(L, m) \in \text{para-ptime-BQL/Qpoly} \cap \text{PHSP}$  iff  $\mathcal{L} \in \text{ptime-2sBQ/poly}$ .*
2. *Assume that  $\mathcal{L}$  is L-good and  $(K, m')$  is induced from  $\mathcal{L}$ . It then follows that  $(K, m') \in \text{para-ptime-BQL/Qpoly} \cap \text{PHSP}$  iff  $\mathcal{L} \in \text{ptime-2sBQ/poly}$ .*

For readability, we postpone the proof of Proposition 6.1 until the end of this section. Meanwhile, we demonstrate another useful lemma concerning classical and quantum advice.

**Lemma 6.2** *Given any  $\mathcal{A} \in \{\text{BQL}, \text{ptime-BQL}\}$ , it follows that  $\text{para-}\mathcal{A}/\text{poly} \cap \text{PHSP} = \text{para-}\mathcal{A}/\text{Qpoly} \cap \text{PHSP}$  iff  $\mathcal{A}/\text{poly} = \mathcal{A}/\text{Qpoly}$ .*

**Proof.** A key idea of the following proof is similar to that of Lemma 5.8. In what follows, we target the case of  $\mathcal{A} = \text{BQL}$  and skip the other case of  $\mathcal{A} = \text{ptime-BQL}$ .

(Only if-part) Assume that  $\text{para-BQL/poly} \cap \text{PHSP}$  equals  $\text{para-BQL/Qpoly} \cap \text{PHSP}$ . Since the inclusion  $\text{BQL/poly} \subseteq \text{BQL/Qpoly}$  obviously holds, it suffices to verify that  $\text{BQL/Qpoly} \subseteq \text{BQL/poly}$ . Let  $L$  be any language in  $\text{BQL/Qpoly}$  over an appropriate alphabet  $\Sigma$ . By setting  $m_{bin}(x) = |x|$  for all  $x \in \Sigma^*$ , we instantly obtain the membership of  $(L, m_{bin})$  to  $\text{para-BQL/Qpoly} \cap \text{PHSP}$ . By our assumption,  $(L, m_{bin})$  falls into  $\text{para-BQL/poly}$ . By the definition of  $m_{bin}$ ,  $L$  must belong to  $\text{BQL/poly}$ .

(If-part) We start with assuming  $\text{BQL/poly} = \text{BQL/Qpoly}$ . Let us consider any parameterized decision problem  $(L, m)$  in  $\text{para-BQL/Qpoly} \cap \text{PHSP}$ . We remark that  $m$  is polynomially honest. Take a polynomially-bounded quantum advice function  $h$  and an advised QTM  $M$  that solves  $L$  on inputs of the form  $(x, h(|x|))$  with bounded-error probability using  $O(\log m(x))$  space. Our goal is to demonstrate that  $(L, m)$  belongs to  $\text{para-BQL/poly}$ . Notice that, since  $m$  is a log-space size parameter,  $m$  must be polynomially bounded. This fact ensures the existence of a constant  $c > 0$  that forces  $m(x) \leq |x|^c + c$  to hold for all  $x \in \Sigma^*$ . Thus, the space usage of  $M$  is at most  $O(\log m(x)) \subseteq O(\log |x|)$  on all inputs of the form  $(x, h(|x|))$ . Since  $h$  is also polynomially bounded, we conclude that  $L$  belongs to  $\text{BQL/Qpoly}$ .

Our assumption then yields  $L \in \text{BQL/poly}$ . Take an advised QTM  $N$  and a polynomially-bounded classical-advice function  $k$  such that  $N$  solves  $L$  on all inputs of the form  $(x, k(|x|))$  with bounded-error probability using  $O(\log |x|)$  space. By the polynomial honesty of  $m$ , there exists another constant  $e > 0$  satisfying  $|x| \leq m(x)^e + e$  for all  $x$ . It then follows that  $O(\log |x|) \subseteq O(\log(m(x)^e + e)) \subseteq O(\log m(x))$ . Thus, the space usage of  $N$  is upper-bounded by  $O(\log m(x))$ . We therefore conclude that  $(L, m)$  is actually in  $\text{para-BQL/poly} \cap \text{PHSP}$ .  $\square$

Theorem 1.3 follows from Propositions 5.1 and 6.1 by an additional application of Lemma 6.2.

**Proof of Theorem 1.3.** (If-part) Assume that  $\text{ptime-BQL/poly} = \text{ptime-BQL/Qpoly}$ . Lemma 6.2 then implies that  $\text{para-ptime-BQL/poly} \cap \text{PHSP}$  coincides with  $\text{para-ptime-BQL/Qpoly} \cap \text{PHSP}$ . Let  $\mathcal{L}$  denote any family of promise decision problems in  $\text{ptime-2sBQ}^\dagger/\text{poly}$  and take its L-good extension  $\hat{\mathcal{L}}$  in  $\text{ptime-2sBQ}^\dagger/\text{poly}$ . Let  $(K, m)$  denote the parameterized decision problem induced from  $\hat{\mathcal{L}}$ . By

Proposition 6.1(2), it follows that  $(K, m)$  belongs to  $\text{para-ptime-BQL/Qpoly} \cap \text{PHSP}$ . By our assumption,  $(K, m)$  falls in  $\text{para-ptime-BQL/poly}$ . Proposition 5.1(2) then implies that  $\hat{\mathcal{L}} \in \text{ptime-2BQ/poly}$ . By the property of L-good extension, we obtain  $\mathcal{L} \in \text{ptime-2BQ}^\dagger/\text{poly}$ . Therefore, we conclude that  $\text{ptime-2sBQ}^\dagger/\text{poly} \subseteq \text{ptime-2BQ}^\dagger$ .

(Only If-part) Assume that  $\text{ptime-2sBQ}^\dagger/\text{poly} \subseteq \text{ptime-2BQ}^\dagger$ . Owing to Lemma 6.2, it suffices to verify that  $\text{para-ptime-BQL/poly} \cap \text{PHSP}$  equals  $\text{para-ptime-BQL/Qpoly} \cap \text{PHSP}$ . It is obvious that  $\text{para-ptime-BQL/poly} \subseteq \text{para-ptime-BQL/Qpoly}$ . To show the converse inclusion, let  $(L, m)$  denote any parameterized decision problem in  $\text{ptime-para-BQL/Qpoly} \cap \text{PHSP}$ . Additionally, take the family  $\mathcal{L}$  of promise decision problems induced from  $(L, m)$ . By Proposition 6.1(1),  $\mathcal{L}$  falls in  $\text{ptime-2sBQ/poly}$ . Since  $\mathcal{L}$  is L-good, it belongs to  $\text{ptime-2sBQ}^\dagger/\text{poly}$ . Our assumption further places  $\mathcal{L}$  in  $\text{ptime-2BQ}^\dagger/\text{poly}$ . Proposition 5.1(1) then concludes that  $(L, m) \in \text{para-ptime-BQL/poly} \cap \text{PHSP}$ , as requested.  $\square$

The proof of Proposition 6.1(1) requires one more crucial lemma, Lemma 6.3, which is analogous to Lemmas 5.2–5.3; however, due to the use of 2sqfa's, we need to deal with “transition tables” in Lemma 6.3 instead of “transition functions” used in Lemmas 5.2–5.3.

**Lemma 6.3** *Assume that  $m$  is a size parameter, which is polynomially honest. Let  $b, r, s, t$ , and  $\ell$  denote arbitrary functions.*

1. *Let  $M$  be an advised QTM  $M$  and, for each index  $l \in \mathbb{N}$ , define  $h(l)$  to be a superposition  $|\phi_l\rangle$  of advice strings in  $\Theta^{r(l)}$ , where  $\Theta$  denotes a fixed advice alphabet. There exist a nonuniform family  $\mathcal{N} = \{N_{n,l}\}_{n,l \in \mathbb{N}}$  of 2sqfa's with  $O(r(l)) \cdot 2^{O(\ell(n,l))}$  inner states and a family  $\{|\psi_{n,l}\rangle\}_{n,l \in \mathbb{N}}$  of superpositions  $|\psi_{n,l}\rangle$  of  $\mathcal{N}$ 's transition tables of encoded length  $O(r(n,l)^9 \log^2 t(n,l)) \cdot 2^{O(\ell(n,l))}$  satisfying the following: for any string  $x$ , if  $M$  accepts (resp., rejects)  $(x, h(|x|))$  with bounded-error probability in expected  $t(m(x), |x|)$  time using at most  $\ell(m(x), |x|)$  space, then  $N_{m(x), |x|}$  accepts (resp., rejects)  $x$  following the quantum transition table  $|\psi_{m(x), |x|}\rangle$  with bounded-error probability in expected  $O(t(m(x), |x|))$  time.*
2. *Let  $\mathcal{N} = \{N_{n,l}\}_{n,l \in \mathbb{N}}$  be a nonuniform family of  $r(n,l)$ -state 2sqfa's and let  $\{|\psi_{n,l}\rangle\}_{n,l \in \mathbb{N}}$  be a family of superpositions  $|\psi_{n,l}\rangle$  of  $\mathcal{N}$ 's transition tables of encoded length at most  $s(n,l)$ . Assume that  $m(x) \leq b(|x|)$  for all  $x$ . There exist an advised QTM  $M$  and an  $O(b(|x|)s(m(x), |x|))$ -bounded quantum advice function  $h$  such that, for any string  $x$ , if  $N_{m(x), |x|}$  accepts (resp., rejects)  $x$  following  $|\psi_{m(x), |x|}\rangle$  with bounded-error probability in expected  $t(m(x), |x|)$  time, then  $M$  accepts (resp., rejects)  $(x, 1^{m(x)}, h(|x|))$  with bounded-error probability in expected  $O(b(|x|)s(m(x), |x|)t(m(x), |x|) \log r(m(x), |x|))$  time using  $O(\log r(m(x), |x|))$  space.*

**Proof.** (1) Let  $m, r, t, \ell, M, h$ , and  $\Theta$  satisfy the premise of the lemma. Assume that  $M$  is of the form  $(Q, \Sigma, \{\$, \# \}, \Gamma, \Theta, \Xi, \delta, q_0, Q_{acc}, Q_{rej})$ . Similarly to the proof of Lemma 5.2, we want to construct from  $M$  a nonuniform family  $\mathcal{N} = \{N'_{n,l}\}_{n,l \in \mathbb{N}}$  of 2sqfa's. For each index  $l \in \mathbb{N}$ , we assume that  $h(l)$  has the form  $\sum_{a:|a|=r(l)} \alpha_a |a\rangle$  with  $\sum_{a:|a|=r(l)} |\alpha_a|^2 = 1$ , where  $a$  ranges over  $\Theta^{r(l)}$ .

Let  $a$  denote any advice string in  $\Theta^{r(l)}$ . We write  $M_a$  to denote the QTM  $M$  whose advice tape consists of  $a$ . We first construct a “2qfa”  $N_{n,|x|,a}$  from  $M_a$  in a way similar to the proof of Lemma 5.2 except that we translate its transition function  $\delta$  into a transition table  $T_{n,l,a}$  partly composed of the descriptions of quantum circuits, as discussed in Section 2.3. This new machine  $N_{n,l,a}$  has  $O(r(l)) \cdot 2^{O(\ell(n,l))}$  inner states and  $T_{n,l,a}$  can be expressed as its encoded string  $\langle T_{n,l,a} \rangle$  of length  $O(|Q|^9 |\Xi|^8 \log^2 t(n,l))$ , which is further bounded by  $O(r(l)^9 \log^2 t(n,l)) \cdot 2^{O(\ell(n,l))}$ . From such transition tables, we define  $|\psi_{n,l}\rangle$  to be the superposition  $\sum_{a:|a|=r(l)} \alpha_a |T_{n,l,a}\rangle$ , which is used as a quantum transition table for  $N_{n,l,a}$ .

Finally, we define a 2sqfa  $N'_{n,|x|}$  as follows. With the use of the quantum transition table  $|\psi_{n,|x|}\rangle$ , we start with an input  $|x\rangle$  and, for each transition table  $T_{n,|x|,a}$  in  $|\psi_{n,|x|}\rangle$ , we run  $N_{n,|x|,a}$  on  $x$  by following the transition instructions of  $T_{n,|x|,a}$ . This new machine  $N'_{n,|x|}$  is the desired 2sqfa.

(2) From the premise of the lemma, we take  $\{N_{n,l}\}_{n,l \in \mathbb{N}}$  and  $\{|\psi_{n,l}\rangle\}_{n,l \in \mathbb{N}}$ . For brevity, let  $e = \lceil \log |Q| |\Xi| \rceil + 2$ . Assume that each superposition  $|\psi_{n,l}\rangle$  has the form  $\sum_{a:|a|=r(n,l)} \alpha_a |T_{n,l,a}\rangle$ , where  $T_{n,l,a}$  is  $N_{n,l}$ 's transition table of encoded length at most  $s(n,l)$ . The encoding of  $|\psi_{n,l}\rangle$ , denoted by  $|\psi_{n,l}^{(code)}\rangle$ , is the superposition  $\sum_{a:|a|=r(n,l)} \alpha_a |\langle T_{n,l,a} \rangle\rangle$ , where  $\langle T_{n,l,a} \rangle$  is the encoded string of  $T_{n,l,a}$ .

Our goal is to construct the desired advised QTM  $M$  and the desired quantum advice function  $h$ . The function  $h$  is simply defined by setting  $h(l)$  to be  $|1^1\rangle |\# \rangle |\psi_{1,l}^{(code)}\rangle |\#^2\rangle |1^2\rangle |\# \rangle |\psi_{2,l}^{(code)}\rangle |\#^2\rangle \dots |1^{b(l)}\rangle |\# \rangle |\psi_{b(l),l}^{(code)}\rangle$  for any number  $l \in \mathbb{N}$ , where  $\#$  is a

new symbol used as a separator. Similarly to the proof of Lemma 5.3,  $M$  works roughly as follows. On input  $(x, 1^{m(x)}, h(|x|))$ ,  $M$  first sweeps its advice tape to locate  $|\psi_{m(x), |x|}^{(code)}\rangle$  and prepares the quantum state  $|0^e\rangle$  in the form of inner states. When  $N_{m(x), |x|}$  is in inner state  $q$  scanning  $\sigma$ ,  $M$  executes a quantum circuit  $C_{q, \sigma}^{(m(x), |x|)}$  described by a series of quantum gates specified inside  $|\psi_{m(x), |x|}\rangle$  and approximately generates a quantum state  $\sum_{(p, d, \xi)} \alpha_{(q, \sigma, p, d, \xi)} |p, d, \xi\rangle$  from  $|0^e\rangle$ , where  $\alpha_{q, \sigma, p, d, \xi}$  is an appropriate transition amplitude incurred by  $M$ . The expected runtime of  $M$  is bounded by  $O(t(m(x), |x|)h(|x|)\log r(m(x), |x|))$ , which is at most  $O(b(|x|)s(m(x), |x|)t(m(x), |x|)\log r(m(x), |x|))$ .  $\square$

Proposition 6.1(1) directly follows from Lemmas 5.4(1) and 6.3.

**Proof of Proposition 6.1(1).** Take any parameterized decision problem  $(L, m)$  and let  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  denote the family of promise decision problems induced from  $(L, m)$ .

(Only if-part) Let us assume that  $(L, m)$  is in para-ptime-BQL/Qpoly  $\cap$  PHSP. Take an advised QTM  $M$  together with an  $r(|x|)$ -bounded quantum advice function  $h$  for which  $M$  with  $h$  solves  $(L, m)$  with bounded-error probability in expected  $p(m(x), |x|)$  time using at most  $\ell(m(x))$  space, where  $p$  and  $r$  are fixed polynomials and  $\ell$  is a fixed logarithmic function. Since  $m$  is polynomially honest, there is an absolute constant  $c > 0$  such that, for any  $n$  and  $x$ ,  $x \in L_n^{(+)} \cup L_n^{(-)}$  implies  $|x| \leq n^c + c$ . This concludes that  $\mathcal{L}$  has a polynomial ceiling.

By Lemma 6.3(1), we can convert the pair  $(M, h)$  into a nonuniform family  $\{N_{n, l}\}_{n, l \in \mathbb{N}}$  of 2sqfa's with  $O(r(l)) \cdot 2^{O(\ell(n, l))} \subseteq (nl)^{O(1)}$  inner states together with a family  $\{|\psi_{n, l}\rangle\}_{n, l \in \mathbb{N}}$  of quantum transition tables of encoded length  $O(r(l)^9 \log^2 p(n, l)) \cdot 2^{O(\ell(n, l))} \subseteq (nl)^{O(1)}$  since  $2^{O(\ell(n, l))} \subseteq (nl)^{O(1)}$ . Since  $|x| \leq n^c + c$  for any input  $x \in L_n^{(+)} \cup L_n^{(-)}$ , it suffices to consider the case of  $l \leq n^c + c$ . Let us define  $N'_n$  that works as follows. We prepare a new quantum transition table  $|\psi'_n\rangle = |1^1\rangle|\# \rangle|\psi_{n, 1}\rangle|\#^2\rangle|1^2\rangle|\# \rangle|\psi_{n, 2}\rangle|\#^3\rangle \cdots |1^{n^c+c}\rangle|\# \rangle|\psi_{n, n^c+c}\rangle$ . On input  $x$ , using  $n^{O(1)}$  extra inner states,  $N'_n$  first sets  $l = |x|$  if  $|x| \leq n^c + c$  and  $l = n^c + c$  otherwise, and  $N'_n$  then simulates  $N_{n, l}$  on  $x$  following the quantum transition table  $|\psi_{n, l}\rangle$  chosen from  $|\psi'_n\rangle$ . The expected runtime of  $N'_n$  is bounded by  $(n|x|)^{O(1)}$ . Finally, we set  $\mathcal{N}' = \{N'_n\}_{n \in \mathbb{N}}$  and  $\Psi' = \{|\psi'_n\rangle\}_{n \in \mathbb{N}}$ . Note that  $\Psi'$  has encoded length of  $n^{O(1)}$ .

It thus follows that  $M$  accepts (resp., rejects)  $(x, h(|x|))$  with bounded-error probability iff  $N'_{m(x)}$  accepts (resp., rejects)  $x$  following  $|\psi'_{m(x)}\rangle$  with bounded-error probability. Since  $M$  with  $h$  solves  $(L, m)$  with bounded-error probability,  $\mathcal{N}'$  solves  $\mathcal{L}$  following  $\Psi'$  with bounded-error probability as well. This implies that  $\mathcal{L}$  is in ptime-2sBQ/poly.

(If-part) For the converse implication, we assume that  $\mathcal{L}$  is in ptime-2sBQ/poly. Note that  $|x| = n^{O(1)}$  holds for all  $x \in L_n^{(+)} \cup L_n^{(-)}$ . Take a family  $\Psi = \{|\psi_n\rangle\}_{n \in \mathbb{N}}$  of quantum transition tables of encoded length at most  $s(n)$  and a nonuniform family  $\mathcal{N} = \{N_n\}_{n \in \mathbb{N}}$  of  $r(n)$ -state 2sqfa's that solves  $\mathcal{L}$  in expected  $t(n, |x|)$  time with bounded-error probability by following the transition tables embedded in  $\Psi$ , where  $r$ ,  $s$ , and  $t$  are suitable polynomials. Since  $\{x \mid m(x) = n\} = L_n^{(+)} \cup L_n^{(-)}$ ,  $m$  is polynomially bounded. Let us take a polynomial  $b$  satisfying  $m(x) \leq b(|x|)$  for all  $x$ . Since  $\mathcal{L}$  is induced from  $(L, m)$ , Lemma 5.4(1) implies that  $m$  is polynomially honest; thus,  $(L, m)$  belongs to PHSP. By Lemma 6.3(2), there exist an advised QTM  $M$  and a polynomially-bounded quantum advice function  $h$  such that  $M$  with  $h$  simulates  $\mathcal{N}$  with  $\Psi$  in expected time  $O(b(|x|)s(m(x))t(m(x), |x|)\log r(m(x))) \subseteq (|x|m(x))^{O(1)}$  using space  $O(\log r(m(x))) \subseteq O(\log m(x))$ . We therefore conclude that  $(L, m)$  belongs to para-ptime-BQL/Qpoly.  $\square$

For the remaining proof of Proposition 6.1(2), we need to claim the second crucial lemma, Lemma 6.4, which can be shown in parallel to Lemmas 5.5–5.6.

**Lemma 6.4** *Let  $m$  denote a size parameter and let  $b, r, s, t$ , and  $\ell$  be arbitrary functions.*

1. *Take an advised QTM  $M$  and a quantum advice function  $h$  with an advice alphabet  $\Theta$ . Assume that  $h(l)$  produces a superposition of strings in  $\Theta^{r(l)}$ . There exist a nonuniform family  $\mathcal{N} = \{N_{n, l}\}_{n, l \in \mathbb{N}}$  of 2sqfa's with  $O(r(l)) \cdot 2^{O(\ell(n, l))}$  states and a family  $\Psi = \{|\psi_{n, l}\rangle\}_{n, l \in \mathbb{N}}$  of  $\mathcal{N}$ 's quantum transition tables of encoded length  $O(r(l)^9 \log^2 t(n, l)) \cdot 2^{O(\ell(n, l))}$  such that, for any string  $x$ , if  $M$  accepts (resp., rejects)  $(1^n \# x, h(|x|))$  with bounded-error probability in expected  $t(n, |x|)$  time using at most  $\ell(n, |x|)$  space, then  $N_{n, |x|}$  accepts (resp., rejects)  $x$  following the quantum transition table  $|\psi_{n, |x|}\rangle$  with bounded-error probability in expected  $O(t(n, |x|))$  time.*
2. *Let  $\mathcal{N} = \{N_{n, l}\}_{n, l \in \mathbb{N}}$  be a nonuniform family of  $r(n, l)$ -state 2sqfa's and a family  $\{|\psi_{n, l}\rangle\}_{n, l \in \mathbb{N}}$*

of superpositions of  $\mathcal{N}$ 's transition tables of encoded length  $s(n, l)$ . There exist an advised QTM  $M$  and an  $O(ns(n, |x|))$ -bounded quantum advice function  $h$  such that, for any string  $x$ , if  $N_{n, |x|}$  with the quantum transition table  $|\psi_{n, |x|}\rangle$  accepts (resp., rejects)  $x$  with bounded-error probability in expected  $t(n, |x|)$  time, then  $M$  accepts (resp., rejects)  $(1^n \# x, h(|x|))$  with bounded-error probability in expected  $O(ns(n, |x|)t(n, |x|) \log r(n, |x|))$  time using  $O(\log r(n, |x|))$  space.

**Proof.** A basic idea of the proof of the lemma is a combination of the proofs of Lemma 6.3 and Lemmas 5.5–5.6. Therefore, we give only the sketch of the proof.

(1) Take an advised QTM  $M$  and a quantum advice function  $h$  as in the premise of the lemma. Assume that  $h(l)$  has the form  $\sum_{a: |a|=r(l)} \alpha_a |a\rangle$  over an advice alphabet  $\Theta$ . For each advice string  $a \in \Theta^{r(l)}$ , we define  $M_a$  from  $M$  in a way similar to the proof of Lemma 6.3(1). By a construction similar to the one used in the proof of Lemma 5.5, we first build a family  $\{N_{n, l, a}\}_{n, l, a}$  of 2sqfa's and a family  $\{T_{n, l, a}\}_{n, l, a}$  of transition tables of encoded length  $O(r(l)^9 \log^2 p(n, l)) \cdot 2^{O(\ell(n, l))}$ . From these  $N_{n, l, a}$ 's and  $T_{n, l, a}$ 's, we define a quantum transition table  $|\psi_{n, l}\rangle$  as  $\sum_{a: |a|=r(l)} \alpha_a |T_{n, l, a}\rangle$  and a new 2sqfa  $N'_{n, l}$  so that, for all  $a \in \Theta^{r(|x|)}$ ,  $N'_{n, l}$  simulates  $N_{n, l, a}$  according to the quantum transition table  $|T_{n, l, a}\rangle$ .

(2) Starting with a given family  $\mathcal{N} = \{N_{n, l}\}_{n, l \in \mathbb{N}}$  of 2sqfa's and also a given family  $\Psi = \{|\psi_{n, l}\rangle\}_{n, l \in \mathbb{N}}$  of quantum transition tables, we construct the desired advised QTM  $M$  and the desired quantum advice function  $h$  in a way similar to the proof of Lemma 5.6 except that  $h(l)$  holds the information on the superpositions of encoded transition tables of  $N_{n, l}$  and that  $M$  first retrieves a transition table from  $h(|x|)$  and simulates  $N_{n, |x|}$  on  $x$ .  $\square$

With the help of Lemmas 5.4(2) and 6.4, we intend to verify Proposition 6.1(2).

**Proof of Proposition 6.1(2).** Let  $\mathcal{L} = \{(L_n^{(+)}, L_n^{(-)})\}_{n \in \mathbb{N}}$  be any L-good family of promise decision problems and let  $(K, m)$  denote the parameterized decision problem induced from  $\mathcal{L}$ .

(Only If-part) Assume that  $(K, m)$  is in para-ptime-BQL/Qpoly  $\cap$  PHSP. There exist an advised QTM  $M$  and a polynomially-bounded quantum advice function  $h$  for which  $M$  with  $h$  solves  $(K, m)$  with bounded-error probability in expected  $t(m(x), |x|)$  time using at most  $\ell(m(x))$  space, where  $p$  is a fixed polynomial and  $\ell$  is a fixed logarithmic function. Take a suitable polynomial  $r$  satisfying  $|h(l)| \leq r(l)$  for all  $l \in \mathbb{N}$ . By Lemma 6.4(1), we can take an appropriate nonuniform family  $\mathcal{N} = \{N_{n, l}\}_{n, l \in \mathbb{N}}$  of expected- $O(t(n, l))$ -time 2sqfa's having  $O(r(l)) \cdot 2^{O(\ell(n, l))}$  inner states for which  $\mathcal{N}$  with  $\Psi$  solves  $\mathcal{L}$  with bounded-error probability following a certain family  $\Psi = \{|\psi_{n, l}\rangle\}_{n, l \in \mathbb{N}}$  of quantum transition tables of encoded length  $O(r(l)^9 \log^2 t(n, l)) \cdot 2^{O(\ell(n, l))}$ . Since  $m$  is polynomially honest, there is a polynomial  $s$  satisfying  $|x| \leq s(m(x))$  for all  $x$ . This implies that  $\mathcal{L}$  has a polynomial ceiling.

We first define  $|\psi'_n\rangle$  as  $|1^1\rangle|\# \rangle|\psi_{n, 1}\rangle|\#^2\rangle|1^2\rangle|\# \rangle|\psi_{n, 2}\rangle|\#^2\rangle \cdots |1^{s(n)}\rangle|\# \rangle|\psi_{n, s(n)}\rangle$ . Note that  $|\psi'_n\rangle$  has encoded length  $n^{O(1)}$  by the choice of  $r$  and  $\ell$ . We then make  $N'_n$  simulate  $N_{n, |x|}$  as follows: on input  $x$ ,  $N'_n$  sets  $l = |x|$  if  $|x| \leq s(n)$  and  $l = s(n)$  otherwise, and follows the quantum transition table  $|\psi_{n, l}\rangle$  chosen from  $|\psi'_n\rangle$ . It then follows that  $N'_n$  uses  $n^{O(1)}$  inner states and solves  $(L_n^{(+)}, L_n^{(-)})$  in expected  $n^{O(1)}$  time. Therefore,  $\mathcal{L}$  belongs to ptime-2sBQ/poly.

(If-part) On the contrary, we assume that  $\mathcal{L}$  is in ptime-2sBQ/poly. Lemma 5.4(2) implies that  $m$  is polynomially honest; thus,  $(K, m)$  belongs to PHSP. Take a polynomial  $b$  satisfying  $m(x) \leq b(|x|)$  for all  $x$ . We thus aim at verifying that  $(K, m)$  belongs to para-ptime-BQL/poly. Take a nonuniform family  $\mathcal{N} = \{N_n\}_{n \in \mathbb{N}}$  of  $r(n)$ -state 2sqfa's solving  $\mathcal{L}$  with bounded-error probability in expected  $t(n, |x|)$  time following a certain family  $\Psi = \{|\psi_n\rangle\}_{n \in \mathbb{N}}$  of quantum transition tables of encoded length  $s(n, l)$ , where  $r$ ,  $s$ , and  $t$  are suitable polynomials. Since  $m(x) \leq b(|x|)$  for all  $x$ , Lemma 6.4(2) guarantees the existence of an advised QTM  $M$  and an  $O(b(|x|)s(m(x), |x|))$ -bounded quantum advice function  $h$  for which  $M$  with  $h$  simulates  $\mathcal{N}$  in expected time  $O(b(|x|)s(m(x), |x|)t(m(x), |x|) \log r(m(x))) \subseteq (|x|m(x))^{O(1)}$  using space  $O(\log r(m(x))) \subseteq O(\log m(x))$ . We therefore conclude that  $(K, m)$  indeed falls in para-ptime-BQL/Qpoly.  $\square$

## 7 Challenging Open Questions

*Finite automata* are generally regarded as one of the simplest models of computation because their behaviors are simple enough to describe and easy to execute. Throughout this exposition, we have aimed

at establishing bridges between such simple models and space-bounded advised computations by way of utilizing parameterized decision problems. Our effort in this exposition has earned a partial success in making a limited progress in the *theory of nonuniform state complexity*, which was initiated by Berman and Lingas [5] and Sakoda and Sipser [32] and lately revitalized by Kapoutsis [19, 20, 21], Kapoutsis and Pighizzini [22], and Yamakami [52].

To direct fruitful future research, we intend to provide a short list of challenging open questions to the avid reader.

1. Computational models of one-way finite automata are relatively easier to analyze than two-way head models; however, we have not determined all inclusion and separation relationships among the nonuniform state complexity classes appearing in Figure 1. An important task is to complete this figure by establishing all missing relationships in the figure; for instance, prove or disprove that  $1N \not\subseteq 2D$  and  $2^{1BP} \not\subseteq 1NQ$ .
2. In computational complexity theory, many intriguing complexity classes have been introduced and extensively studied over the years. In comparison, the *theory of nonuniform state complexity* is still in a cradle stage of its development. Develop that to a full fledged theory by, e.g., further introducing natural, intriguing state complexity classes and studying their structural properties. A simple example of such complexity classes is quantum interactive proof systems with quantum finite automata verifiers [31]. Even for structural properties of the existing classes, there still remain numerous open questions of whether or not, e.g.,  $2N$  is closed under complementation.
3. Since finite automata are simple in structure, they may be suitable to be used for numerous applications outside of automata theory. In this exposition, we have shown only a direct application of nonuniform state complexity to the precise characterizations of several main-stream advised space complexity classes. We need to discover more intriguing applications of nonuniform state complexity in other fields of computer science.
4. The choice of transition amplitudes and transition probabilities seems to endow underlying quantum and probabilistic finite automata with quite different computational power. For example, we experience a great difficulty in showing that  $2Q = \text{co-}2Q$  and  $2P = \text{co-}2P$  because of the lack of our understanding of complex transition amplitudes and real transition probabilities. A much better understanding of complex and real numbers are definitely needed in further analyses of various computations. See also [44, Section 5.3] for relevant subjects.
5. We have studied automata families of polynomially many inner states as well as exponentially many inner states. Since there is an enormous gap between them, it seems natural to study finite automata families of “sub-exponentially many” inner states. One of such sub-exponential state complexities is  $2^{\ell(n)}$  with  $\ell(n) = \Theta(\log^k n)$  for a fixed constant  $k \geq 2$ . The nonuniform state complexity class defined by automata families of such state complexity will naturally fill the gap between polynomial state complexity and exponential state complexity and will surely replenish our study of nonuniform state complexity.
6. Another important consideration is *unary languages* (or *tally sets*). In general, finite automata working on unary input strings are succinctly called *unary automata*. With the use of such unary automata, from any nonuniform state complexity class  $\mathcal{C}$  (such as  $1BQ$  and  $\text{ptime-}2BQ$ ), we can define  $\mathcal{C}/\text{unary}$  by restricting all underlying finite automata to be unary automata. For those restrictive classes, we need to prove various inclusion and separation relationships. As for the (non)deterministic cases, refer to, e.g., [19, 20, 21, 22].
7. Our main concern in this exposition has been *nonuniform* families of various types of finite automata. As another direction of the current research, we suggest to study a “uniform” variant of those families. A family  $\{M_n\}_{n \in \mathbb{N}}$  of automata is *uniform* if there exists a DTM  $N$  that, on input  $1^n$ , produces an appropriate encoding  $\langle M_n \rangle$  of  $M_n$  on its write-once output tape. Along this line of study, there are only a few interesting results found in [5, 21, 52]. We need to explore this subject further in connection to, e.g., main-stream complexity classes.

## References

- [1] A. Ambainis and R. Freivalds. 1-way quantum finite automata: strengths, weaknesses, and generalizations. In *Proc. of the 39th Annual Symposium on Foundations of Computer Science (FOCS'98)*, pp. 332–341 (1998)
- [2] A. Ambainis and N. Nashimovs. Improved constructions of quantum automata. *Theoretical Computer Science* 410, 1916–1922 (2009)
- [3] A. Ambainis and A. Yakaryilmaz. Superiority of exact quantum automata for promise problems. *Information Processing Letters* 112, 289–291 (2012)
- [4] A. Ambainis and A. Yakaryilmaz. Automata and quantum computing. Available on <https://arxiv.org/abs/1507.01988> (2015)
- [5] P. Berman and A. Lingas. On complexity of regular languages in terms of finite automata. Report 304, Institute of Computer Science, Polish Academy of Science, Warsaw (1977)
- [6] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing* 26, 1411–1473 (1997)
- [7] A. Bertoni and M. Carpentieri. Analogies and differences between quantum and stochastic automata. *Theoretical Computer Science* 262, 69–81 (2001)
- [8] M. P. Bianchi, C. Mereghetti, and B. Palano. Size lower bounds for quantum automata. *Theoretical Computer Science* 551, 102–115 (2014)
- [9] M. P. Bianchi, C. Mereghetti, and B. Palano. Complexity of promise problems on classical and quantum automata. *Gruska Festschrift, Lecture Notes in Computer Science* 8808, pp. 161–175 (2014)
- [10] A. Brodsky and N. Pippenger. Characterizations of 1-way quantum finite automata. *SIAM Journal on Computing* 31, 1456–1478 (2002)
- [11] C. Dwork and L. Stockmeyer. A time-complexity gap for two-way probabilistic finite state automata. *SIAM Journal on Computing* 19, 1011–1023 (1990)
- [12] R. Freivalds. Probabilistic two-way machines. In the *Proc. of the 10th International Symposium on Mathematical Foundations of Computer Science (MFCS'81)*, *Lecture Notes in Computer Science*, vol. 118, pp. 33–45 (1981)
- [13] R. Freivalds. Non-constructive methods for finite probabilistic automata. *International Journal of Foundations of Computer Science* 19, 565–580 (2008)
- [14] R. Freivalds and M. Karpinski. Lower space bounds for randomized computation. In the *Proc. of the 21st International Colloquium on Automata, Languages, and Programming (ICALP'94)*, *Lecture Notes in Computer Science*, vol. 820, pp. 580–592 (1994)
- [15] V. Geffert and G. Pighizzini. Two-way unary automata versus logarithmic space. *Information and Computation* 209, 1016–1025 (2011)
- [16] V. Geffert and A. Yakaryilmaz. Classical automata on promise problems. *Discrete Mathematics and Theoretical Computer Science* 17, 157–180 (2015)
- [17] J. Gruska. *Quantum Computing*, McGraw-Hill (2000)
- [18] M. Hirvensalo. Quantum automata with open time evolution. *International Journal of Natural Computing* 1, 70–85 (2010)
- [19] C. A. Kapoutsis. Size complexity of two-way finite automata. In the *Proc. of the 13th International Conference on Developments of Language Theory (DLT 2009)*, *Lecture Notes in Computer Science*, vol. 5583, pp. 47–66 (2009)
- [20] C. A. Kapoutsis. Minicomplexity. *Journal of Automata, Languages, and Combinatorics* 17, 205–224 (2012)
- [21] C. A. Kapoutsis. Two-way automata versus logarithmic space. *Theory of Computing Systems* 55, 421–447 (2014)
- [22] C. A. Kapoutsis and G. Pighizzini. Two-way automata characterizations of  $L/poly$  versus  $NL$ . *Theory of Computing Systems* 56, 662–685 (2015)
- [23] R. M. Karp and R. Lipton. Turing machines that take advice, *L'Enseignement Mathématique* 28, 191–209 (1982)
- [24] M. Karpinski and R. Verbeek. On the Monte Carlo space constructible functions and separation results for probabilistic complexity classes. *Information and Computation* 75, 178–189 (1987)
- [25] A. Kitaev, A. Shen, and M. Vyalı. *Classical and Quantum Computation*. AMS, Providence, RI (2002)
- [26] E. Kushilevitz and N. Nisan. *Communication Complexity*, Cambridge University Press, 2nd edition (2006)

- [27] A. Kondacs and J. Watrous. On the power of quantum finite state automata. In the Proc. of the 38th Annual Symposium on Foundations of Computer Science (FOCS'97), pp. 66–75 (1997)
- [28] C. Moore and J. P. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science* 237, 275–306 (2000)
- [29] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*, Cambridge University Press (2000)
- [30] H. Nishimura and T. Yamakami. Polynomial time quantum computation with advice. *Information Processing Letters* 90, 195–204 (2004)
- [31] H. Nishimura and T. Yamakami. An application of quantum finite automata to interactive proof systems. *Journal of Computer and System Sciences* 75, 255–269 (2009)
- [32] W. J. Sakoda and M. Sipser. Nondeterminism and the size of two-way finite automata. In the Proc. of the 10th annual ACM symposium on Theory of computing (STOC'78), pp. 275–286 (1978)
- [33] K. Tadaki, T. Yamakami, and J. C. H. Lin. Theory of one-tape linear-time Turing machines. *Theoretical Computer Science* 411, 22–43 (2010)
- [34] A. Ta-Shma. Inverting well conditioned matrices in quantum logspace. In the Proc. of the 45th annual ACM symposium on Theory of Computing (STOC 2013), pp. 881–890 (2013)
- [35] D. van Melkebeek and T. Watson. Time-space efficient simulations of quantum computations. *Theory of Computing* 8, 1–51 (2012)
- [36] M. Villagra and T. Yamakami. Quantum state complexity of formal languages. In the Proc. of the 17th International Workshop on Descriptive Complexity of Formal Systems (DCFS 2015), *Lecture Notes in Computer Science*, Springer, vol. 9118, pp. 280–291 (2015)
- [37] J. Watrous. Space-bounded quantum computations. *Journal of Computer and System Sciences* 59, 281–326 (1999)
- [38] J. Watrous. On the complexity of simulating space-bounded quantum computations. *Computational Complexity* 12, 48–84 (2003)
- [39] A. Yakaryilmaz, R. Freivalds, A. C. C. Say, and R. Agadzanyan. Quantum computation with write-only memory. *Natural Computing* 11, 81–94 (2012)
- [40] A. Yakaryilmaz and A. C. C. Say. Languages recognized by nondeterministic quantum finite automata. *Quantum Information and Computation* 10, 747–770 (2010)
- [41] A. Yakaryilmaz and A. C. C. Say. Succinctness of two-way probabilistic and quantum finite automata. *Discrete Mathematics and Theoretical Computer Science* 12, 19–40 (2010)
- [42] A. Yakaryilmaz and A. C. C. Say. Unbounded-error quantum computation with small space bounds. *Information and Computation* 209, 873–892 (2011)
- [43] T. Yamakami. A foundation of programming a multi-tape quantum Turing machine. In the Proc. of the 24th International Symposium on Mathematical Foundations of Computer Science (MFCS'99), *Lecture Notes in Computer Science*, Springer, vol. 1672, pp. 430–441 (1999)
- [44] T. Yamakami. Analysis of quantum functions. *International Journal of Foundations of Computer Science* 14, 815–852 (2003) A preliminary version appeared in the Proc. of the 19th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'99), *Lecture Notes in Computer Science*, Springer, vol.1738, pp.407–419, 1999.
- [45] T. Yamakami. Swapping lemmas for regular and context-free languages. Manuscript, available at arXiv:0808.4122 (2008)
- [46] T. Yamakami. The roles of advice to one-tape linear-time Turing machines and finite automata. *Journal of Foundations of Computer Science* 21, 941–962 (2010)
- [47] T. Yamakami. Oracle pushdown automata, nondeterministic reducibilities, and the hierarchy over the family of context-free languages. In the Proc. of the 40th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2014), *Lecture Notes in Computer Science*, vol. 8327, pp. 514–525 (2014). A complete and corrected version is available at arXiv:1303.1717.
- [48] T. Yamakami. One-way reversible and quantum finite automata with advice. *Information and Computation* 239, 122–148 (2014)
- [49] T. Yamakami. Complexity bounds of constant-space quantum computation (extended abstract). In the Proc. of the 19th International Conference on Developments in Language Theory (DLT 2015), *Lecture Notes in Computer Science*, vol. 9168, pp. 426–438 (2015). An extended and corrected version is available at arXiv:1606.08764 (2016).

- [50] T. Yamakami. The 2CNF Boolean formula satisfiability problem and the linear space hypothesis. In the Proc. of the 42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017), Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik 2017, vol. 83, 62:1–62:14 (2017). A complete and corrected version is available at [arXiv:1709.10453](https://arxiv.org/abs/1709.10453).
- [51] T. Yamakami. Parameterized graph connectivity and polynomial-time sub-linear-space short reductions (preliminary report). In the Proc. of the 11th International Workshop on Reachability Problems (RP 2017), Lecture Notes in Computer Science, vol. 10506, pp. 176–191 (2017)
- [52] T. Yamakami. State complexity characterizations of parameterized degree-bounded graph connectivity, sub-linear space computation, and the linear space hypothesis. *Theoretical Computer Science* 798, 2–22 (2019). A preliminary version appeared in the Proc. of the 20th DCFS 2018, Lecture Notes in Computer Science, vol. 10952, pp. 237–249 (2018)
- [53] T. Yamakami. Supportive oracles for parameterized polynomial-time sub-linear-space computations in relation to L, NL, and P. In the Proc. of the 5th Annual Conference on Theory and Applications of Models of Computation (TAMC 2019), Lecture Notes in Computer Science, vol. 11436, pp. 659–673 (2019)
- [54] T. Yamakami. Relativizations of nonuniform quantum finite automata families. In the Proc. of the 18th International Conference on Unconventional Computation and Natural Computation (UCNC 2019), Lecture Notes in Computer Science, vol. 11493, pp. 257–271, 2019.
- [55] S. Zheng, J. Gruska, and D. Qiu. On the state complexity of semi-quantum finite automata. *RAIRO-Theoretical Informatics and Applications* 48, 187–207 (2014)
- [56] S. Zheng, L. Z. Li, D. Qiu, J. Gruska. Promise problems solved by quantum and classical finite automata. *Theoretical Computer Science* 666, 48–64 (2017)
- [57] S. Zheng, D. Qiu, J. Gruska, L. Li, and P. Mateus. State succinctness of two-way finite automata with quantum and classical states. *Theoretical Computer Science* 499, 98–112 (2013)