How Does Adiabatic Quantum Computation Fit into Quantum Automata Theory?

Tomoyuki Yamakami

How Does Adiabatic Quantum Computation Fit into Quantum Automata Theory?

Tomoyuki Yamakami

Faculty of Engineering, University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan
TomoyukiYamakami@gmail.com

Abstract. Quantum computation has emerged as a powerful computational medium of our time, having demonstrated the efficiency in solving the integer factoring and searching a database faster than any currently known classical computer algorithm. Adiabatic evolution of quantum systems have been studied as a potential means that physically realizes quantum computation. Up to now, all the research on adiabatic quantum systems has dealt with polynomial time-bounded computation and little attention has been paid to, for example, adiabatic quantum systems consuming only constant memory space. Such quantum systems can be modeled in a form similar to quantum finite automata. This exposition dares to ask a bold question of how to make adiabatic quantum computation fit into the rapidly progressing framework of quantum automata theory. As our answer to this eminent but profound question, we first lay out a basic framework of adiabatic evolutionary quantum systems (AEQSs) with limited computational resources and then establish their close connection to quantum finite automata. We also explore fundamental structural properties of languages solved quickly by such adiabatic evolutionary quantum systems.

Keywords: adiabatic quantum computation, quantum finite automata, Hamiltonian, Schrödinger equation

1 Motivations and a Quick Overview

1.1 Adiabatic Quantum Computation

Quantum computation has gained large popularity over the past few decades. There are several important milestones to remember in our time. Shor proposed a polynomial-time quantum algorithm of factoring a positive integer and computing discrete logarithms whereas Grover presented a quantum way to locate a key in a unstructured database quadratically faster than traditional search. Basis of such quantum computation has been modeled typically by quantum Turing machines and quantum circuits. See, e.g., [8, 10, 14].

In a given quantum system, its quantum state $|\psi(t)\rangle$ evolves according to the Schrödinger equation $\imath\hbar\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle$ using a specified time-dependent *Hamiltonian* $H(t)$ (which is simply a Hermitian matrix), where $\hbar$ is the reduced *Planck constant*. In early 2000's, Farhi, Goldstone, Gutmann, and Sipser [6]

and Farhi, Goldstone, Gutmann, Lapan, Lundgren, and Preda [5] developed a quantum algorithm based on a variant of quantum annealing, called *adiabatic quantum computation*, in which an initial quantum system whose ground state is easily prepared gradually evolves to find a solution represented by a ground state of a final quantum system. A crucial point is how fast this evolution takes place. Unfortunately, it turns out that the algorithm of Farhi et al. [5] requires exponential time to execute [4].

Adiabatic quantum computation is dictated by a quadruple $(n, \varepsilon, H_{ini}, H_{fin})$ of two Hamiltonians H_{ini} and H_{fin} of dimension 2^n and a closeness bound ε such that H_{ini}'s ground state (i.e., an eigenvector of the smallest eigenvalue) is easily prepared and the outcome of the system becomes the ground state of H_{fin} and this ground state is ε-close to the desired solution. Such a quantum system starts with the ground state $|\psi_g(0)\rangle$ of $H(0) = H_{ini}$ at time $t = 0$. If $H(t)$ changes sufficiently slowly, the evolving quantum state $|\psi(t)\rangle$ stays close to the ground state $|\psi_g(t)\rangle$ of $H(t)$. For the efficiency of adiabatic quantum computation, we have concerned with the *evolution time* of the underlying quantum system and the *structural complexity* of two Hamiltonians used in the system. The running time of the system is determined roughly by the evolution time of the system and is basically proportional to the reciprocal of the *spectral gap* of H_{ini} and H_{fin} according to the well-known *adiabatic theorem* [9, 11]. See Section 2.3.

Later, van Dam, Mosca, and Vazirani [4] gave a detailed analysis of adiabatic quantum computation and presented how to simulate adiabatic quantum computation on quantum circuits. In addition, Aharonov, van Dam, Kemp, Landau, Lloyd, and Regev [1] demonstrated how to simulate quantum circuit computation by adiabatic quantum computation and thus established the (polynomial) equivalence between adiabatic quantum computation and standard quantum computation.

Although adiabatic quantum computation is no more powerful than standard quantum computation, it seems to remain as significant potentials to realize restricted variants of quantum computation. With the current technology, it still seems to be difficult to build a large-scale adiabatic quantum computing device since making local evolution in a large system is quite sensitive to *decoherence*. It is rather better to make global evolution in a small system. It thus remains more realistic to prepare Hamiltonians of adiabatic quantum computation using a memory-restricted quantum device. We may wonder what would happen if we restrict our attention onto a constant-memory model of Turing machine, which is conceptually realized by *finite(-state) automata*. To seek for yet-unearthed potentials of adiabatic quantum computation in such a realistic setting, this exposition intends to make a new, bold step by taking an automata-theoretic approach toward adiabatic quantum computability. Since adiabatic quantum computation follows the gradual evolution between two Hamiltonians H_{ini} and H_{fin}, the key to the realization of such computation relies on how easily we can build these Hamiltonians. This fact motivates us to consider the circumstances where the matrices are "generated" by memory-restricted device, namely, quantum finite automata. This exposition reports an initial result of our bold attempt

to deal with adiabatic quantum computation from a viewpoint of quantum finite automata.

At first glance, since our desired quantum algorithm to solve a given decision problem requires outcomes of only one-bit solutions, it seems sufficient for us to prepare either $|1\rangle$ or $|0\rangle$ as a unique ground state of a final Hamiltonian H_{fin}. For such a preparation, however, we should know the solution well ahead of quantum computation. Without knowing any solution of the problem, we may not in general prepare H_{fin} prior to the start of the computation. Farhi et al. [5]'s idea of designing a final Hamiltonian H_{fin} to solve Search-2SAT (i.e., a search version of the satisfiability problem for 2CNF formulas) is to encode a solution of the problem directly into H_{fin} without knowing the solution. In a similar fashion, we want to find a way to prepare H_{fin} without apparently computing any solution to the target problem. This exposition proposes the use of quantum finite automata as a mechanical tool to directly generate Hamiltonians as the automata read input symbols one by one. In this way, we can prepare Hamiltonians even without knowing a solution to a given decision problem.

1.2 Quantum Automata Theory

A quantum finite automaton takes an input string given onto its read-only input tape and, as reading the input symbol one by one, it changes inner states in a quantum manner until it finally terminates. This process can be described as quantum transitions of inner states according to input symbols. The theory of quantum finite automata has been developed significantly since the first installment of quantum finite automata in 1990s (see, e.g., a survey [3]).

Since a paradigm of adiabatic quantum computation looks quite different from a standard framework of quantum finite automata, we certainly face the following challenging question. Is it possible for us to make adiabatic quantum computation fit into the framework of quantum automata theory?

1.3 An Overview of Main Contributions

This exposition attempts to answer the aforementioned question of how adiabatic quantum computation fits into quantum automata theory.

We need to define a scaled-down model for adiabatic quantum computation, aiming at capturing an essence of such computation in terms of quantum finite automata. For this purpose, we introduce an *adiabatic evolutionary quantum system* (AEQS, pronounced as "eeh-ks") consisting of an input alphabet, a size parameter, a closeness bound, two Hamiltonians for each input, and acceptance/rejection criteria for each input size (whose precise definition will be given in Section 2.4).

Lemma 1. *For any language L over an alphabet Σ, there is a series of AEQSs $\{\mathcal{S}_x\}_{x\in\check{\Sigma}}$ of finite size such that, for any x, $\mathcal{S}_x$ computes $L(x)$.*

Although this lemma demonstrates the power of AEQSs, it does not provide us with a constructive mechanism of generating AEQSs.

In this exposition, we are focused on how to prepare two Hamiltonians of AEQSs using variants of quantum finite automata. The use of algorithmic construction of Hamiltonians also provides the *uniformity condition* to adiabatic quantum computation. After giving the basic notions and notation in Section 2, we will demonstrate how to design (or program) AEQSs for several languages (of Proposition 2) in Section 3.

Concerning the computational complexity of AEQSs, we wish to limit the behaviors of AEQSs. To describe the families of decision problems (or equivalently, languages) associated with AEQSs under a set $\mathcal{F}$ of certain "natural" conditions on key ingredients, two Hamiltonians, we use the notation AEQS($\mathcal{F}$). In general, the complexity class AEQS($\mathcal{F}$) is composed of all decision problems (or languages), each of which is solved (or recognized) with "high-accuracy" by a certain AEQS whose Hamiltonians satisfy the conditions specified by $\mathcal{F}$. This helps us discuss various types of conditions, which play essential roles in determining the computational complexity of AEQSs. Of all possible types of conditions, we are focused on the following 4 condition types.

Firstly, we are interested in how to "generate" two Hamiltonians of AEQSs. In particular, we consider the case where those Hamiltonians are generated by certain *one-way quantum quasi-automata* (abbreviated as 1qqa's), each of which acts as a means to produce a series of *Kraus operators* according to each input symbol so that the product of its adjoint and itself matches the target Hamiltonians (whose precise definition will be given in Section 2.2). We use the notation $\mathcal{F}$ ="1qqa" to denote the use of 1qqa's to generate Hamiltonians of AEQSs. We also define another condition set $\mathcal{F}$ ="2cqqa" using *2-way classical-head quantum quasi-automata* (or 2cqqa's) in place of 1qqa's. In a similar manner, we define $\mathcal{F}$ ="1qpdqa" to mean the use of *one-way quantum pushdown quasi-automata* (abbreviated as 1qpdqa's) induced from a quantum version of pushdown automata.

Secondly, we are concerned with the *(system) size* of AEQSs, where the (system) size of an AEQS is the logarithm of the dimension of its Hamiltonians. We write $\mathcal{F}$ ="constsize" (constant size), $\mathcal{F}$ ="logsize" (logarithmic size), and $\mathcal{F}$ ="linsize" (linear size) to express the sizes of target AEQSs.

Thirdly, we pay attention to the value of the *spectral gap* of a final Hamiltonian of an AEQS, where the spectral gap is the difference between the first and the second smallest eigenvalues. This value provides an upper bound of the runtime of the AEQS. For instance, if the final Hamiltonian of an AEQS have their spectral gap inverse-polynomially large, then the adiabatic evolution of the AEQS takes only polynomially many steps. Notice that, even if Hamiltonians are generated by 1qqa's, there seems a chance that the spectral gap is exponentially small. From this fact, we introduce the notation $\mathcal{F}$ ="polygap" to mean that the spectral gap is at least $1/n^{O(1)}$ (i.e., inverse-polynomially large). Similarly, $\mathcal{F}$ ="constgap" indicates that the spectral gap is at least $1/O(1)$ (i.e., constantly large).

Fourthly, we look into the *ground energy level* of a final Hamiltonian of an AEQS. In certain cases [5, 6], it is possible to set the ground energy of a final Hamiltonian to be 0. This motivates us to define $\mathcal{F}$ ="0-energy" for the situation where the ground energy of a final Hamiltonian is 0 for every input x.

Proposition 2. *Let $\mathbb{N}$ denote the set of all natural numbers. The following statements hold.*

1. *For each fixed string $a \in \{0,1\}^*$, the regular language $L_a = \{ax \mid x \in \{0,1\}^*\}$ is in* AEQS(1qqa, constgap).
2. *The reversible unary language L_{even} defined as $L_{even} = \{a^{2n} \mid n \in \mathbb{N}\}$ is in* AEQS(1qqa, constgap, 0-energy).
3. *The deterministic context-free language $L_{eq} = \{0^n 1^n \mid n \in \mathbb{N}\}$ is in* AEQS(2cqqa, logsize, polygap).
4. *The language TRI consists of all strings of the form $w_1\#w_2\#w_3$ such that $w_1, w_2, w_3 \in \{0,1\}^*$, $|w_1| = |w_2| = |w_3|$, $(w_1 = w_2^R)$ XOR $(w_1 = w_3^R)$. This language is in* AEQS(1qpdqa, linsize, polygap).

We show a more general theorem below. 1MOQFA is a language family characterized by bounded-error 1-way measure-once quantum finite automata [12].

Theorem 3. 1MOQFA $\subseteq$ AEQS(1qqa, constsize, constgap, 0-energy).

Next, we target the class REG of all regular languages. If we use 1qqa's to generate Hamiltonians, then we obtain the following relations. Here, we expand 1qqa's by allowing them to have $O(n)$ inner states. To express such machines, we use the term "linsize-1qqa's."

Theorem 4. REG $\subseteq$ AEQS(linsize-1qqa, constgap, 0-energy).

We further explore structural properties of AEQS($\mathcal{F}$). A language family $\mathcal{L}$ is said to be *closed under* a binary operation $\circ$ if, for any two languages $L_1, L_2 \in \mathcal{L}$, $L_1 \circ L_2$ also belongs to $\mathcal{L}$. We say that $\mathcal{F}$ *allows a swap of acceptance/rejection criteria* if the new AEQS obtained from any given AEQS satisfying $\mathcal{F}$ by exchanging its $S_{acc}^{(n)}$ and $S_{rej}^{(n)}$ also satisfies $\mathcal{F}$, where $S_{acc}^{(n)}$ and $S_{rej}^{(n)}$ respectively denote sets of indices providing criteria for acceptance and rejection of ground states. Additionally, we say that $\mathcal{F}$ *allows the amplification of accuracy* if, for any AEQS satisfying $\mathcal{F}$ and any constant $c > 0$, there always exists another computationally-equivalent AEQS with $\mathcal{F}$ whose accuracy is at least c times as high as the original AEQS's.

Proposition 5. *Let $\mathcal{F}$ be any nonempty set of conditions. Each of the following statements holds.*

1. AEQS($\mathcal{F}$) *is closed under complementation if $\mathcal{F}$ allows a swap of acceptance/rejection criteria.*
2. AEQS($\mathcal{F}$) *is closed under XOR if $\mathcal{F}$ allows the amplification of accuracy.*

This exposition is merely the initial attempt to expand the scope of adiabatic quantum computability and to relate it to quantum finite automata using the new notion of AEQSs. We expect that this exposition marks the beginning of a series of exciting research works, aiming at the deeper understanding of adiabatic quantum computation.

2 Preparations: Notions and Notation

2.1 Numbers, Vectors, and Matrices

The notation $\mathbb{N}$ expresses the set of all *natural numbers* (that is, nonnegative integers) and we set $\mathbb{N}^+ = \mathbb{N} - \{0\}$. Given two integers m, n with $m \leq n$, the *integer interval* $[m, n]_{\mathbb{Z}}$ is the set $\{m, m+1, m+2, \ldots, n\}$, which is compared to a real interval $[a, b]$. Let $\mathbb{C}$ denote the set of all *complex numbers*; in particular, we set $\imath = \sqrt{-1}$. All *polynomials* are assumed to have nonnegative integer coefficients and all *logarithms* are taken to the base 2.

We deal with finite-dimensional Hilbert spaces. To express (column) vectors of such a space, we use Dirac's notation $|\cdot\rangle$. A *density operator* (or a density matrix) expresses a mixed quantum state. Given a complex matrix A, $A^\dagger$ indicates the complex conjugate transpose of A. For any matrix A and its index pair (q, r), the notation $A[q, r]$ indicates the (q, r)-entry of A. Similarly, for a vector v, $v[i]$ denotes the ith entry of v. Given any square complex matrix A, the notation e^A expresses a *matrix exponential* defined by $e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$ (where $0! = 1$ and $A^0 = I$) and the *spectral norm* $\|A\|$ is defined by $\|A\| = \max_{|\phi\rangle \neq 0} \{ \frac{\|A|\phi\rangle\|_2}{\||\phi\rangle\|_2} \}$, where $\|\cdot\|_2$ indicates the ℓ_2-norm. For a number $\varepsilon \in [0, 1]$ and for two vectors v_1 and v_2 in the same Hilbert space, we say that v_1 is *ε-close to v_2* if $\|v_1 - v_2\|_2 \leq \varepsilon$. The *commutator* $[A, B]$ of square matrices A and B is defined as $AB - BA$.

We use the notation W for the *Walsh-Hadamard transform*. For convenience, we write $|\hat{a}\rangle = W|a\rangle$ for any $a \in \{0, 1\}$. The *Hadamard basis* is $\{|\hat{0}\rangle, |\hat{1}\rangle\}$ and the *computational basis* is $\{|0\rangle, |1\rangle\}$.

A *Hamiltonian* is a complex Hermitian matrix. For any Hamiltonian H, we set $\Delta(H)$ to be the *spectral gap* of H, which is the difference between the lowest eigenvalue and the second lowest eigenvalue of H. The lowest eigenvalue is called the *ground energy* of H and its eigenvector is called the *ground state* of H.

2.2 Languages and Quantum Quasi-Automata

An *alphabet* is a finite nonempty set of "symbols" or "letters." A *string* over an alphabet Σ is a finite sequence of symbols in Σ. The *length* of a string x is the total number of symbols in x and is denoted by $|x|$. In particular, the *empty string* has length 0 and is denoted by λ. The notation Σ^* stands for the set of all strings over Σ. A *language* over Σ is a subset of Σ^*. Hereafter, we freely identify a decision problem with its associated language.

A *one-way quantum finite automaton*[1] (abbreviated as a 1qfa) *with mixed states and quantum operations* M is a septuple $(Q, \Sigma, \{\text{¢}, \$\}, \{A_\sigma\}_{\sigma \in \check{\Sigma}}, q_0, Q_{acc}, Q_{rej})$, where Q is a finite set of inner states, Σ is an (input) alphabet, ¢ and \$ are respectively the *left endmarker* and the *right endmarker*, $\check{\Sigma} = \Sigma \cup \{\text{¢}, \$\}$, and each A_σ is a *quantum operation*[2] acting on the Hilbert space of linear operators on the *configuration space*

[1] This model is called general quantum finite automata in a survey [3].

[2] This is a completely positive, trace preserving map and is also called a *superoperator*.

spanned by the basis vectors $\{|q\rangle \mid q \in Q\}$ [2, 7, 15]. Such a quantum operation A_σ has a Kraus representation with *Kraus operators* (or operation elements) $\{A_{\sigma,j}\}_{j\in[k]}$ for a certain constant $k \in \mathbb{N}^+$. More precisely, A_σ takes the form $A_\sigma(H) = \sum_{j=1}^{k} A_{\sigma,j} H A_{\sigma,j}^\dagger$ for any linear operator H and satisfies the completeness relation $\sum_{j=1}^{k} A_{\sigma,j}^\dagger A_{\sigma,j} = I$. In particular, when $k = 1$, we identify $A_{\sigma,1}$ with A_σ and then obtain $A_\sigma(H) = A_\sigma H A_\sigma^\dagger$. Given strings $y_1, y_2, \cdots, y_k$ in $\check{\Sigma}$, we abbreviate a matrix multiplication $A_{y_k} \cdot A_{y_{k-1}} \cdots A_{y_2} \cdot A_{y_1}$ as $A_{y_1 y_2 \cdots y_{k-1} y_k}$. Given a language L, M *recognizes* with error probability at most ε if, for any $x \in L$, $\mathrm{tr}(P_{acc} A_{¢x\$}(\rho_0)) \geq 1 - \varepsilon$ and, for any $x \notin L$, $\mathrm{tr}(P_{rej} A_{¢x\$}(\rho_0)) \geq 1 - \varepsilon$, where $\rho_0 = |q_0\rangle\langle q_0|$ and P_{acc} and P_{rej} are projections onto the spaces spanned by $\{|q\rangle \mid q \in Q_{acc}\}$ and by $\{|q\rangle \mid q \in Q_{rej}\}$, respectively. This model is in essence equivalent to a *garbage-tape model* used in [16]. In contrast, a *one-way measure-once quantum finite automaton* (or a 1moqfa) applies only unitary operators until it reads \$. The *state complexity* of a finite automaton is the total number of inner states of the automaton. We write 1MOQFA and $1\mathrm{QFA}_{mix}$ to denote the collections of all languages recognized respectively by bounded-error 1moqfa's and bounded-error 1qfa's with mixed states and quantum operators.

We attempt to run finite automata to produce Hamiltonians, which are necessary to carry out adiabatic quantum computation. For this purpose, we need to modify the aforementioned model of 1qfa's so that they can produce "matrices." A *one-way quantum quasi-automaton* (or a 1qqa, for short) is a 1qfa equipped with mixed states and quantum operators with no use of initial state and final state. More formally, a 1qqa M is a quadruple $(Q, \Sigma, \{A_\sigma\}_{\sigma\in\check{\Sigma}}, \Lambda_0)$, where each A_σ is a quantum operation on the Hilbert space of linear operators on the configuration space and Λ_0 is a Hermitian operator acting on the same space. This machine M can produce a matrix $A_{¢x\$}$ for any given input $x \in \Sigma^*$. Notice that $A_{¢x\$}(B)$ is a Hermitian matrix for any Hermitian B. From this fact, we say that a Hamiltonian H is *generated by a 1qqa* M if H coincides with $A_{¢x\$}(\Lambda_0)$.

As a natural extension of 1qqa's, in a model of *2-way classical-head quantum quasi-automaton* (or 2cqqa), firstly we apply either a quantum operation A_σ to a finite quantum register or a projection measurement on the quantum register and, secondly we use a deterministic procedure to apply A_σ when a two-way tape head reads an input symbol σ; in the case of a measurement, we apply $A_{\sigma,\tau}$ instead, where τ is a result of the measurement.

Similarly to [13], a *1-way quantum pushdown quasi-automaton* (or a 1qpdqa) is a 1-way quantum finite automaton equipped with a stack in which we can push and pop stack symbols as an input tape head reads input symbols.

2.3 Adiabatic Evolution of a Quantum System

Loosely following [6], we briefly discuss how a quantum system evolves according to the Schrödinger equation of the following general form: $\imath\hbar\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle$ for a time-dependent Hamiltonian $H(t)$ and a time-dependent quantum state $|\psi(t)\rangle$. To carry out adiabatic quantum computation, we prepare two Hamiltonians H_{ini} and H_{fin} acting on the same Hilbert space and, for a suffi-

ciently large constant $T > 0$, we define $H(t) = \left(1 - \frac{t}{T}\right) H_{ini} + \frac{t}{T} H_{fin}$ for a time parameter $t \in [0, T]$, provided that $[H_{ini}, H_{fin}] \neq 0$. To ensure $[H_{ini}, H_{fin}] \neq 0$, we often use the *Hadamard basis* for H_{ini} and the *computational basis* for H_{fin} [5].

At time $t = 0$, we assume that the quantum system is initialized to be the ground state $|\psi_g(0)\rangle$ of H_{ini}. We allow the system to gradually evolve by applying $H(t)$ *discretely* from time $t = 0$ to $t = T$. Let $|\psi(t)\rangle$ denote the quantum state at time $t \in [0, T]$. This evolutionary process is referred to as an *adiabatic evolution according to H for T steps*. We take the smallest value T for which $|\psi(T)\rangle$ is ε-close to the ground state of H_{fin}. For convenience, we call this T the *minimum evolution time* of the system. The *runtime* of the system is then defined to be $T \cdot \max_{t\in[0,T]} \|H(t)\|$ and the *outcome* of the system is the quantum state $|\psi(T)\rangle$. The *adiabatic theorem* [9, 11] gives a lower bound on T. The following assertion is taken from [1]. For any constant $\delta > 0$, if $T \geq \Omega\left(\frac{\|H_{fin} - H_{ini}\|^{1+\delta}}{\varepsilon^{\delta} \min_{t\in[0,T]}\{\Delta(H(t))^{2+\delta}\}}\right)$, then $|\psi(T)\rangle$ (with an appropriately chosen *global phase*) is ε-close to the ground state of H_{fin}, provided that $H(t)$ has a unique ground state for each value $t \in [0, T]$.

For a practical simulation of the system, it is useful to consider a refinement of the time intervals. Let R be a fixed number satisfying $T \ll R$ and consider a refined time interval of $[\frac{jT}{R}, \frac{(j+1)T}{R}]$ for each index $j \in [0, R-1]_{\mathbb{Z}}$.

Lemma 6. *Given a quantum system $(n, \varepsilon, H_{ini}, H_{fin})$ of adiabatic evolution, let T be the minimum evolution time. Let U_T denote a unitary matrix satisfying $|\psi(T)\rangle = U_T|\psi(0)\rangle$. Let R be a number with $T \ll R$. Let $\alpha_j = \frac{1}{\hbar}\frac{T}{R}\left(1 - \frac{2j+1}{2R}\right)$ and $\beta_j = \frac{1}{\hbar}\frac{T}{R}\frac{2j+1}{2R}$ for each index $j \in [0, R-1]_{\mathbb{Z}}$. It then follows that U_T can be approximated by the matrix $V_R = \left(e^{-\imath\alpha_R H_{ini}} \cdot e^{-\imath\beta_R H_{fin}}\right) \cdots \left(e^{-\imath\alpha_1 H_{ini}} \cdot e^{-\imath\beta_1 H_{fin}}\right)$.*

2.4 Adiabatic Evolutionary Quantum Systems

Our major target of this exposition is decision problems (or equivalently, languages). Instead of searching solutions as in [5], we are asked to determine "acceptance" (yes) or "rejection" (no) of any given input string. Let us define our quantum systems that evolve adiabatically. Since adiabatic quantum computation is dictated by the ground states of an initial Hamiltonian and a final Hamiltonian, we thus need to specify these Hamiltonians for each given input.

We loosely adapt the definition of Aharonov et al. [1] but modify it significantly to match our purpose. First of all, we wish to realize adiabatic quantum computation by a new notion of an *adiabatic evolutionary quantum system* (or an AEQS, pronounced as "eeh-ks"). An AEQS $\mathcal{S}$ is a septuple $(m, \Sigma, \varepsilon, \{H_{ini}^{(x)}\}_{x\in\Sigma^*}, \{H_{fin}^{(x)}\}_{x\in\Sigma^*}, \{S_{acc}^{(n)}\}_{n\in\mathbb{N}}, \{S_{rej}^{(n)}\}_{n\in\mathbb{N}})$, where $m : \Sigma^* \to \mathbb{N}$ is a size function, Σ is an (input) alphabet, ε is an accuracy bound in $[0, 1]$, both $H_{ini}^{(x)}$ and $H_{fin}^{(x)}$ are Hamiltonians acting on the same Hilbert space of $2^{m(x)}$ dimension (where this space is referred to as the system's *evolution space*), and $S_{acc}^{(n)}$ and $S_{rej}^{(n)}$ are sets of indices representing acceptance and rejection (where each pair $(S_{acc}^{(n)}, S_{rej}^{(n)})$ is called an *(acceptance/rejection) criteria pair*). The func-

tion $m(x)$ is called the *(system) size* of $\mathcal{S}$, expressing how large the evolution space is.

An evolution process of an AEQS can be described as in Section 2.3. Letting T_x indicate the minimum evolution time of this system, we define $H^{(x)}(t)$ to be $\left(1-\frac{t}{T_x}\right)H_{ini}^{(x)}+\frac{t}{T_x}H_{fin}^{(x)}$ for any real number $t\in[0,T_x]$. At time $t=0$, the AEQS is initialized to be the ground state $|\psi_g(0)\rangle$ of $H^{(x)}(0)$ $(=H_{ini}^{(x)})$. The system slowly evolves by applying $H^{(x)}(t)$ discretely from time $t=0$ to $t=T_x$. The AEQS is considered to take the runtime of $T_x\cdot\max_{t\in[0,T_x]}\|H^{(x)}(t)\|$.

To solve a mathematical problem using the adiabatic evolution of a quantum system, following [5], we may assume that $H_{ini}^{(x)}=\sum_{u\in S_0^{(x)}}h(u)|u\rangle\langle u|$, where $S_0^{(x)}$ is the set of all indices for which $|u\rangle$ are eigenvectors of $H_{ini}^{(x)}$ and $h(u)$ is a real value associated with $|u\rangle$. The ground state $|\psi_g(0)\rangle$ of $H_{ini}^{(x)}$ is $|u_0\rangle$ if u_0 satisfies that $u_0\in S_0^{(x)}$ and $h(u_0)=\min\{h(u)\mid u\in S_0^{(x)}\}$.

To work on decision problems, in particular, we need to specify *accepting and rejecting quantum states* in the evolution space on which $H^{(x)}(t)$ acts. This can be done by incorporating the two sets $S_{acc}^{(n)}$ and $S_{rej}^{(n)}$. We define $QS_{acc}^{(n)}$ and $QS_{rej}^{(n)}$ respectively to be the Hilbert spaces spanned by $\{|u\rangle\mid u\in S_{acc}^{(n)}\}$ and $\{|u\rangle\mid u\in S_{rej}^{(n)}\}$. We call $QS_{acc}^{(n)}$ and $QS_{rej}^{(n)}$ the *accepting space* and the *rejecting space*, respectively.

To determine the outcome of an AEQS, we want to design the AEQS to make the ground state of $H_{fin}^{(x)}$ sufficiently "close" to a certain accepting or rejecting quantum state, which belongs to $QS_{acc}^{(m(x))}\cup QS_{rej}^{(m(x))}$. The closeness of the ground state of $H_{fin}^{(x)}$ to such a quantum state corresponds to the *accuracy* of the AEQS to the desired solution of a decision problem on each input x.

Definition 7. *Given a decision problem L and any constant $\varepsilon\in[0,1/2)$, we say that an AEQS $\mathcal{S}=(m,\Sigma,\varepsilon,\{H_{ini}^{(x)}\}_{x\in\Sigma^*},\{H_{fin}^{(x)}\}_{x\in\Sigma^*},\{S_{acc}^{(n)}\}_{n\in\mathbb{N}},\{S_{rej}^{(n)}\}_{n\in\mathbb{N}})$* solves *(or* recognizes*)* L with accuracy at least ε *if (i) for each input $x\in\Sigma^*$, there exists a unique ground state $|\psi_g(0)\rangle$ of $H_{ini}^{(x)}$, (ii) for any string $x\in L$, the ground state of $H_{fin}^{(x)}$ is $(1-\varepsilon)$-close to a certain quantum state in $QS_{acc}^{(m(x))}$, and (iii) for any string $x\in\Sigma^*-L$, the ground state of $H_{fin}^{(x)}$ is $(1-\varepsilon)$-close to a certain quantum state in $QS_{rej}^{(m(x))}$. The* adiabatic quantum size complexity *of L is $m(x)$, where "x" expresses a "symbolic" input.*

Proof Sketch of Lemma 1. Let Σ be any alphabet and let L be any language over Σ. For each fixed string $x\in\Sigma^*$, we define an AEQS $\mathcal{S}_x$ as follows: $H_{ini}^{(x)}=|-\rangle\langle-|$ and $H_{fin}^{(x)}=|\overline{L}(x)\rangle\langle\overline{L}(x)|$, where $|-\rangle=\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$ and $\overline{L}(x)=1-L(x)$ for every input $x\in\Sigma^*$ and $L(x)$ is the characteristic function of L. Note that the ground state of $H_{ini}^{(x)}$ is $|+\rangle=\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$ and that of $H_{fin}^{(x)}$ is $|L(x)\rangle$. It

then follows that $x \in L$ iff $\mathcal{S}_x$ outputs $L(x)$. The accuracy of $\mathcal{S}_x$ thus turns out to be 1. □

We consider a complexity class of decision problems solved by certain AEQSs.

Definition 8. *Let $\mathcal{F}$ indicate a set of conditions imposed on Hamiltonians. The complexity class,* highly-accurate AEQS($\mathcal{F}$)*, is the collection of all languages, each of which is recognized by a certain AEQS on each input with accuracy at least an absolute constant $\varepsilon \in (1/2, 1]$ and Hamiltonians of the AEQS satisfy the conditions specified by $\mathcal{F}$. Since we discuss mostly highly-accurate AEQS's in the subsequent sections, we often drop the prefix "highly-accurate" and simply call them AEQS's.*

As an example, we may use $\mathcal{F}$ to specify a type of quantum quasi-automata, such as 1qqa's. We may also use $\mathcal{F}$ to refer to the condition that the spectral gaps of Hamiltonians are at most the reciprocal of a certain polynomial. By the adiabatic theorem, this condition implies that the corresponding AEQSs run for polynomially many steps.

One of the difficulties that we face in constructing an AEQS is how to prepare its Hamiltonians, in particular, $H_{fin}^{(x)}$ and to define (acceptance/rejection) criteria pairs $(S_{acc}^{(n)}, S_{rej}^{(n)})$. In this exposition, we aim at relating AEQSs to quantum finite automata. Since AEQSs are dictated by Hamiltonians, we say that an AEQS $\mathcal{S}$ is *generated by 1qqa's* if (1) there exist two 1qqa's M_0 and M_1 such that $H_{ini}^{(x)}$ and $H_{fin}^{(x)}$ are generated respectively by M_0 and M_1 for each input $x \in \Sigma^*$, and (2) $S_{acc}^{(i)} = S_{acc}^{(j)}$ and $S_{rej}^{(i)} = S_{rej}^{(j)}$ for any pair $i, j \in \mathbb{N}$. We can expand this definition to 2cqqa's and 1qpdqa's. Possibly, we can relax Condition 92) by requiring a one-way reversible finite automaton (or 1rfa) to determine whether or not "q is in $S_{acc}^{(n)}$" (as well as "q is in $S_{rej}^{(n)}$") from inputs of the form $(1^n, q)$.

In the subsequent section, we will demonstrate how to design (or program) AEQSs for the simple languages given in Proposition 2.

3 How to Program AEQSs: Proof of Proposition 2

In what follows, we will demonstrate only (1)–(2) of Proposition 2.

(1) We want to construct the desired AEQS for the language $L_a = \{ax \mid x \in \{0,1\}^*\}$ for each fixed string $a \in \{0,1\}^*$. Here, we consider only the simplest case where $a = 0$. Write Σ for $\{0,1\}$ for simplicity.

Our goal is to construct $H_{ini}^{(x)} = W_3 \mathrm{diag}(0,1,1) W_3^\dagger$, $H_{fin}^{(0y)} = \mathrm{diag}(1, 3/4, 1/4)$, and $H_{fin}^{(1y)} = \mathrm{diag}(1, 1/4, 3/4)$ using 1qqa's, where W_3 is an appropriate 3×3 unitary matrix forcing the condition $[H_{ini}^{(x)}, H_{fin}^{(x)}] \neq 0$. Since the ground energy of $H_{fin}^{(x)}$ is $1/4$, it follows that the spectral gap is $\frac{3}{4} - \frac{1}{4} = \frac{1}{2}$.

To generate the Hamiltonians $H_{ini}^{(x)}$ and $H_{fin}^{(x)}$, we need to define an appropriate 1qqa. Let $m(x) = |x|$, $Q = \{q_\ell, q_0, q_1\}$, $S_{acc}^{(|x|)} = \{q_1\}$ and $S_{rej}^{(|x|)} =$

$\{q_0\}$. We choose $\Lambda_0 = |q_0\rangle\langle q_0| + |q_1\rangle\langle q_1|$. We further define Kraus operators $\{A_{\sigma,i}\}_{\sigma\in\check{\Sigma},i\in[5]}$ as follows. Let $A_{\mathfrak{c},1}|q_{\mathfrak{c}}\rangle = |q_{\mathfrak{c}}\rangle$, $A_{\mathfrak{c},2}|q_0\rangle = \frac{1}{\sqrt{2}}|q_{\mathfrak{c}}\rangle$, $A_{\mathfrak{c},3}|q_0\rangle = \frac{1}{\sqrt{2}}|q_0\rangle$, $A_{\mathfrak{c},4}|q_1\rangle = \frac{1}{\sqrt{2}}|q_{\mathfrak{c}}\rangle$, $A_{\mathfrak{c},5}|q_1\rangle = \frac{1}{\sqrt{2}}|q_1\rangle$, and $A_{\$,i} = A_{\mathfrak{c},i}$. Moreover, let $A_{0,1}|q_{\mathfrak{c}}\rangle = |q_0\rangle$, $A_{0,2}|q_0\rangle = |q_0\rangle$, $A_{0,3}|q_1\rangle = |q_1\rangle$, $A_{1,1}|q_{\mathfrak{c}}\rangle = |q_1\rangle$, $A_{1,2} = A_{0,2}$, and $A_{1,3} = A_{0,3}$. For all other pairs $(\sigma,i) \in \check{\Sigma}\times[5]$, let $A_{\sigma,i}|q\rangle = 0$. It then follows that $\sum_{i=1}^{5} A_{\sigma,i}^{\dagger}A_{\sigma,i} = I$ for any $\sigma\in\check{\Sigma}$. It is not difficult to show that $A_{\mathfrak{c}0y\$}(\Lambda_0) = H_{fin}^{(0y)}$ and $A_{\mathfrak{c}1y\$}(\Lambda_0) = H_{fin}^{(1y)}$.

(2) Next, we are focused on the reversible unary language $L_{even} = \{a^{2n} \mid n\in\mathbb{N}\}$. Let $m(x) = |x|$, $Q = \{q_0,q_1,q_2\}$, $S_{acc}^{(|x|)} = \{q_1\}$, and $S_{rej}^{(|x|)} = \{q_2\}$. Let $U_{\mathfrak{c}}|q_0\rangle = |q_1\rangle$, $U_{\mathfrak{c}}|q_1\rangle = |q_0\rangle$, $U_{\mathfrak{c}}|q_2\rangle = |q_2\rangle$, and $U_{\$} = I$. Moreover, let $U_a|q_0\rangle = |q_0\rangle$, $U_a|q_1\rangle = |q_2\rangle$, and $U_a|q_2\rangle = |q_1\rangle$. It then follows that, for any input $x\in\{a\}^*$, $U_{\mathfrak{c}x\$}|q_0\rangle = |q_1\rangle$ if $x\in L_{even}$ and $U_{\mathfrak{c}x\$}|q_0\rangle = |q_2\rangle$ if $x\notin L_{even}$. Let us define $H_{fin}^{(x)} = U_{\mathfrak{c}x\$}\Lambda_0 U_{\mathfrak{c}x\$}^{\dagger}$, where $\Lambda_0 = \sum_{q\in Q-\{q_0\}}|q\rangle\langle q|$. Since $U_{\mathfrak{c}x\$}|q_1\rangle = |q_0\rangle$ and $U_{\mathfrak{c}x\$}|q_2\rangle = |q_2\rangle$ if $x\in L_{even}$, and $|q_1\rangle$ otherwise, it follows that $H_{fin}^{(x)} = |q_0\rangle\langle q_0| + \overline{L}_{even}(x)|q_1\rangle\langle q_1| + L_{even}(x)|q_2\rangle\langle q_2|$, where $L_{even}(x)$ is the characteristic function of L_{even} and $\overline{L}_{even}(x) = 1 - L_{even}(x)$. The ground state of $H_{fin}^{(x)}$ thus becomes $|q_1\rangle$ if $x\in L_{even}$, and $|q_2\rangle$ otherwise. Therefore, we obtain $\Delta(H_{fin}^{(x)}) = 1$. Note that the ground energy is 0.

4 Basic Simulations between QFAs and AEQSs

4.1 Proof of Theorem 3

Aharonov et al. [1] demonstrated how to simulate quantum circuits by adiabatic quantum computation with polynomial overhead. Our concern here is the AEQSs of constant size, and thus our situation significantly differs from their's.

Proof Sketch of Theorem 3. We intend to show that 1MOQFA $\subseteq$ AEQS(1qqa, constsize, copnstgap, 0-energy). Let $L\in$ 1MOQFA and choose a 1moqfa $M = (Q,\Sigma,\{\mathfrak{c},\$\},\{U_\sigma\}_{\sigma\in\check{\Sigma}},q_0,Q_{acc},Q_{rej})$ for L with error probability at most ε for a certain constant $\varepsilon\in[0,1/2)$. For simplicity, we assume that $|Q|$ is of the form 2^{k_0} for a certain constant $k_0\in\mathbb{N}^+$ and that all elements in Q are expressed as k_0-bit strings. Let $\rho_0 = |q_0\rangle\langle q_0|$ and $\rho_{i+1} = U_{x_i}\rho_i U_{x_i}^{\dagger}$ for each $i\in[0,n+1]_{\mathbb{Z}}$, where x_i denotes the $(i+1)$th symbol of $\mathfrak{c}x\$$. Let P_{acc} and P_{rej} be two projections onto the Hilbert spaces H_{acc} and H_{rej} spanned by $\{|q\rangle \mid q\in Q_{acc}\}$ and $\{|q\rangle \mid q\in Q_{rej}\}$, respectively. Note that, for any $x\in L$, $\mathrm{tr}(P_{acc}\rho_{n+2})\geq 1-\varepsilon$ and, for any $x\notin L$, $\mathrm{tr}(P_{rej}\rho_{n+2})\geq 1-\varepsilon$.

Our goal is to show how to simulate M by a suitable AEQS. We define $H_{ini}^{(x)}$ and $H_{fin}^{(x)}$ of the desired AEQS as $H_{ini}^{(x)} = W^{\otimes k_0}\Lambda_0 W^{\otimes k_0}$ and $H_{fin}^{(x)} = U_{\mathfrak{c}x\$}\Lambda_0 U_{\mathfrak{c}x\$}^{\dagger}$, where $\Lambda_0 = \sum_{q\in Q-\{q_0\}}|q\rangle\langle q|$.

For each $x\in\Sigma^*$, let $|\phi_x\rangle = U_{\mathfrak{c}x\$}|q_0\rangle$. By the definition of $H_{fin}^{(x)}$, $|\phi_x\rangle$ is its ground state because $H_{fin}^{(x)}|\phi_x\rangle = U_{\mathfrak{c}x\$}\Lambda U_{\mathfrak{c}x\$}^{\dagger}U_{\mathfrak{c}x\$}|q_0\rangle = U_{\mathfrak{c}x\$}\Lambda|q_0\rangle = 0$. Note

that, if $x \in L$, then $\|P_{acc}|\phi_x\rangle\|^2 \geq 1-\varepsilon$. This implies that there is a vector $|\phi_{acc}\rangle \in H_{acc}$ such that $|\langle\phi_{acc}|\phi_x\rangle|^2 \geq 1-\varepsilon$. We then obtain $\||\phi_x\rangle - |\phi_{acc}\rangle\|^2 \leq 1-|\langle\phi_{acc}|\phi_x\rangle|^2 \leq \varepsilon$. Hence, $|\phi_x\rangle$ is ε-close to $|\phi_{acc}\rangle$. A similar argument handles the case of $x \notin L$.

Next, we consider nonzero eigenvalues of $H_{fin}^{(x)}$. Let $|\psi_q\rangle = U_{\mathfrak{c}x\$}|q\rangle$ for each $q \in Q$. It then follows that $U_{\mathfrak{c}x\$}\Lambda_0 U_{\mathfrak{c}x\$}^\dagger = \sum_{q\in Q-\{q_0\}}(U_{\mathfrak{c}x\$}|q\rangle\langle q|U_{\mathfrak{c}x\$}^\dagger) = \sum_{q\in Q-\{q_0\}}|\psi_q\rangle\langle\psi_q|$. Since $U_{\mathfrak{c}x\$}$ is unitary, $\{|\psi_q\rangle\}_{q\in Q-\{q_0\}}$ consists of all nonzero eigenvectors and they must be 1. Therefore, the spectral gap of $H_{fin}^{(x)}$ is 1. □

4.2 Proof of Theorem 4

Finally, we are ready to describe the proof of Theorem 4. This proof is composed of two critical simulations between AEQSs and 1qfa's with mixed states and quantum operators. Since REG = 1QFA_{mix}, it suffices to simulate such a 1qfa by an appropriately chosen AEQS.

Lemma 9. *Any 1qfa M with mixed states and quantum operators can be exactly simulated by a certain AEQS whose Hamiltonians are generated by certain linear-size 1qqa's with the spectral gap of* 1 *and the ground energy of* 0.

Proof of Theorem 4. Let L be any regular language. Take a 1qfa M with mixed states and quantum operators that recognizes L with bounded-error probability. By Lemma 9, there exists an AEQS $\mathcal{S}$ that simulates M. □

References

1. Aharonov, D., van Dam, W., Kemp, J., Landau, Z., Lloyd, S., Regev, O.: Adiabatic quantum computation is equivalent to standard quantum computation. SIAM J. Comput. **37**, 166–14 (2007)
2. Ambainis, A., Beaudry, M., Golovkins, M., Ķikusts, A., Mercer, M., Thérien, D.: Algebraic results on quantum automata. Theory Comput. Syst. **39**, 165–188 (2006)
3. Ambainis, A., Yakaryilmaz, A.S.: Automata and quantum computing. Manuscript (2015), https://arxiv.org/abs/1507.01988
4. van Dam, W., Mosca, M., Vazirani, U.: How powerful is adiabatic quantum computation? In: Proc. of FOCS 2001. pp. 279–287 (2001)
5. Farhi, E., Goldstone, J., Gutmann, S., Lapan, J., Lundgren, A., Preda, D.: A quantum evolution algorithm applied to random instances of an NP-complete problem. Science **292**, 472–476 (2001)
6. Farhi, E., Goldstone, J., Gutmann, S., Sipser., M.: Quantum computation by adiabatic evolution. Manuscript (2000), https://arxiv.org/abs/quant-ph/0001106
7. Freivalds, R., Ozols, M., Mančinska, L.: Improved constructions of mixed state quantum automata. Theoret. Comput. Sci. **410**, 1923–1931 (2009)
8. Gruska, J.: Quantum Computing. McGraw-Hill (2000)

9. Kato, T.: On the adiabatic theorem of quantum mechanics. J. Phys. Soc. Jap. **5**, 435–439 (1951)
10. Kitaev, A., Shen, A., Vyalyi, M.: Classical and Quantum Computation. AMS, Providence, RI (2002)
11. Messiah, A.: Quantum Mechanics. John Wiley & Sons, New York (1958)
12. Moore, C., Crutchfield, J.P.: Quantum automata and quantum grammars. Theor. Comp. Sci. **237**, 275–306 (2000)
13. Nakanishi, M.: Quantum pushdown automata with garbage tape. Int. J. Found. Comput. Sci. **29**, 425–446 (2018)
14. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
15. Yakaryilmaz, A., Say, A.C.C.: Unbounded-error quantum computation with small space bounds. Inf. Comput. **279**, 873–892 (2011)
16. Yamakami, T.: Nonuniform families of polynomial-size quantum finite automata and quantum logarithmic-space computation with polynomial-size advice. In: Martn-Vide C., Okhotin A., S.D. (ed.) LATA 2019. LNCS, vol. 11417, pp. 134–145. Springer (2018), https://doi.org/10.1007/978-3-030-13435-8_10