frontiers in Physics

ORIGINAL RESEARCH
published: 10 November 2020
doi: 10.3389/fphy.2020.547370

A Quantum Finite Automata Approach to Modeling the Chemical Reactions

Amandeep Singh Bhatia[1] *and Shenggen Zheng*[2]*

[1]*Chitkara University Institute of Engineering & Technology, Chitkara University, Punjab, India* [2]*Peng Cheng Laboratory, Shenzhen, China*

In recent years, the modeling interest has increased significantly from molecular level to atomic and quantum levels. Computational chemistry plays a significant role in designing computational models for the operation and simulation of systems ranging from atoms and molecules to industrial processes. It is influenced by a tremendous increase in computing power and the efficiency of algorithms. The representation of chemical reactions using classical automata theory in thermodynamic terms had a great influence on computer science. The study of chemical information processing with quantum computational models is a natural goal. In this study, we have modeled chemical reactions using two-way quantum finite automata, which are halted in linear time. Additionally, classical pushdown automata can be designed for such chemical reactions with multiple stacks. It has been proven that computational versatility can be increased by combining chemical accept/reject signatures and quantum automata models.

Keywords: chemical reaction, two-way quantum finite automata, quantum finite automata, Belousov-Zhabotinsky reaction, pushdown automata, quantum chemistry

OPEN ACCESS

Edited by:
Jiangfeng Du,
University of Science and Technology of China, China

Reviewed by:
Son Tung Ngo,
Ton Duc Thang University, Vietnam
Xin Wang,
City University of Hong Kong, Hong Kong

****Correspondence:***
Shenggen Zheng
zhengshg@pcl.ac.cn

Specialty section:
This article was submitted to Physical Chemistry and Chemical Physics, a section of the journal Frontiers in Physics

Received: *31 March 2020*
Accepted: *27 August 2020*
Published: *10 November 2020*

Citation:
Bhatia AS and Zheng S (2020) A Quantum Finite Automata Approach to Modeling the Chemical Reactions. Front. Phys. 8:547370. doi: 10.3389/fphy.2020.547370

1 INTRODUCTION

Recently, the connection between complex reactions and their thermodynamics has received overwhelming response among research communities. Initially, in the 1970s, Conrad [1] processed the information of molecular systems and stated that complex biochemical systems cannot be analyzed in classical computers. Till now, artificial approaches use complex biomolecules or logic gates–based reaction-diffusion systems to solve the problems [2–4]. Classical systems are not robust and incapable to describe quantum systems. Some tasks that are impossible in classical systems can be realized in quantum systems. Quantum computation is concerned with computer technology based on the principles of quantum mechanics, which describes the behavior and nature of matter and energy in quantum level [5]. Quantum computation demonstrates the computation power and other properties of the computers based on the principles of quantum mechanics.

Models of finite automata are abstract computing devices, which play a crucial role to solve computational problems in theoretical computer science. Classical automata theory is closely associated with formal language theory, where automata are ranked from simplest to most powerful depending on their language recognition power [6]. Classical automata theory has been of significant importance due to its practical real-time applications in the development of several fields. Therefore, it is the natural goal to study quantum variants of classical automata models, which play an important role in quantum information processing.

The quantum automata theory has been developed using the principles of quantum mechanics and classical automata. Quantum computational models make it possible to examine the resources needed for computations. Soon after the brainstorm of Shor's factorization quantum algorithm [7],

the first models of quantum finite automata (QFAs) have been introduced. Initially, Kondacs and Watrous [8], and Moore and Crutchfield [9] proposed the concept of quantum automata separately. Since then, a variety of quantum automata models have been studied and demonstrated in various directions, such as QFAs, Latvian QFA, 1.5-way QFA, two-way QFA (2QFA), quantum sequential machine, quantum pushdown automata, quantum Turing machine, quantum multicounter machines, quantum queue automata [10], quantum multihead finite automata, QFAs with classical states (2QCFA) [11, 12], state succinctness of two-way probabilistic finite automata (2PFA), QFA, 2QFA, and 2QCFA [13–15], interactive proof systems with QFAs [16, 17], quantum finite state machines of matrix product state [18], promise problems recognition by QFA [19–22], quantum-omega automata [23] and semi-quantum two-way finite automata [24–26], time complexity advantages of QFA [27], nonuniform classes of polynomial size QFA [28, 29], QFA and linear temporal logic relationship [30], and many more since the past 2 decades [31–34]. These models are effective in determining the boundaries of various computational features and expressive power [35–37]. Quantum computers are more powerful than Turing machines and even probabilistic Turing machines. Thus, mathematical models of quantum computation can be viewed as generalizations of its physical models.

Computational biochemistry has been a rapidly evolving research area at the interface between biology, chemistry, computer science, and mathematics. It helps us to apply computational models to understand biochemical and chemical processes and their properties. A combination of chemistry and classical automata theory provides a constructive means of refining the number of objects allowed to understand the energetic cost of computation [38]. The research has been consistently grown in the field of chemical computing. There exist two ways to model complex chemical reactions: abstract devices and formal models based on multiset rewriting [39]. Complex chemical reaction networks carry out chemical processes that mimic the workings of classical automata models. Recently, Duenas-Diez and Perez-Mercader [38, 40] have designed chemical finite automata for regular languages and chemical automata with multiple stacks for context-free and context-sensitive languages. Furthermore, the thermodynamic interpretation of the acceptance/rejection of chemical automata is given. It is useful to understand the energetic cost of chemical computation. They have used the one-pot reactor (mixed container), where chemical reactions and molecular recognition takes place after several steps, without utilizing any auxiliary geometrical aid.

In classical automata theory, it is known that two-way deterministic finite automata (2DFA) can be designed for all regular languages. It has also been investigated that 2PFA can be designed for a nonregular language $L = \{a^n b^n | n \geq 1\}$ in an exponential time [38, 40]. The research has consistently evolved in the field of quantum computation and information processing. In quantum automata theory, it has been proved that 2QFA can be designed for L with one-sided bounded error and halted in linear time. Moreover, it has been demonstrated that 2QFA can be also designed for non–context-free language $L = \{a^n b^n c^n | n \geq 1\}$ [8]. Hence, 2QFA is strictly more powerful than its classical counterparts based on language recognition capability.

The field of chemistry and chemical computation plays a significant role in the evolution of computational models to mimic the behavior of systems at its atomic level. It is greatly influenced by the computing power of quantum computers. Motivated from the abovementioned facts, we have modeled chemical reactions in the form of formal languages and represented those using two-way QFAs. The main objective is to examine how chemical reactions perform chemical sequence identification equivalent to quantum automata models without involving biochemistry or any auxiliary device. The crucial advantage of this approach is that chemical reactions in the form of accept/reject signatures can be processed in linear time with one-sided bounded error (if the automata makes error only in one direction, i.e., either on "no" instances or on "yes" instances). This article is further designed as follows: Subsection is devoted to prior work. In **Section 2**, some preliminaries are given. The definition of two-way QFAs is given in **Section 3**. In **Section 4**, the chemical reactions are transcribed in formal languages and modeled using two-way QFAs approach. Summary of work is given in **Section 5**. Finally, **Section 6** is the conclusion.

1.1 Prior Work

The field of chemical computation has rich and interesting history. Various researchers have represented chemical computation using the concept of logic gates–based reaction-diffusion systems and artificial intelligence approaches. In early 1970s, Conrad [1] differentiated the information processing in molecules using digital computation. Nearly a decade later, Okamoto et al. [43] proposed the concept of a theoretical chemical diode in cyclic enzyme systems. It has been proved that it can be used to analyze the dynamic behavior of metabolic switching events in biocomputer. In 1991, Hjelmfelt et al. [44] designed neural networks and finite state machines using chemical diodes. It has been found that the execution of a universal Turing machine is possible using connecting chemical diodes. Hjelmfelt et al. constructed clocked finite state machines of binary adder, binary decoder, and stack memory and showed that finite state machines can be simulated by clocked neural networks.

In 1995, Tóth and Showalter [45] implemented AND and OR logic gates using reaction-diffusion systems, where the signals are programmed by chemical waves. It was the first empirical realization of chemical logical gates. In 1997, Magnasco [46] showed that logic gates can be constructed and executed in the chemical kinetics of homogeneous solutions. It has been proved that such constructions have computational power equivalent to Turing machine. Adamatzky and Lacy Costello [47] experimentally understood the Chemical XOR gate by following the same approach of Toth and Showlter in 2002. Further, Górecki et al. [48] constructed the chemical counters for information processing in the excitable reaction-diffusion systems.

It is one of the most promising new areas of research. Some difficulties can be caused by connecting several gates together for

advanced computation. Thus, recently, researchers started focusing on native chemical computation, that is, without reaction-diffusion systems. In 1994, Adleman [49] proposed the concept of DNA computation and solved the Hamiltonian path problem by changing DNA strands. In 2009, Benenson [2] reviewed biological measurement tools for new-generation biocomputers. Prohaska et al. [3] studied protein domain using chromatin computation and introduced chromatin as a powerful machine for chemical computation and information processing. In 2012, Bryant [4] proved chromatin computer as computationally universal by using it to solve an example of combinatorial problem.

The structures of DNA and RNA are represented using the concept of classical automata theory [50, 51]. Krasinski et al. [52] represented the restricted enzyme in DNA with pushdown automata in circular mode. Khrennikov and Yurova [53] modeled the behavior of protein structures using classical automata theory and investigated the resemblance between the quantum systems and modeling behavior of proteins. Bhatia and Kumar [54] modeled ribonucleic acid (RNA) secondary structures using two-way QFAs, which are halted in linear time. Duenas-Diez and Perez-Mercader designed molecular machines for chemical reactions. The native chemical computation has been implemented beyond the scope of logic gates, that is, with chemical automata [40]. It has been demonstrated that chemical reactions transcribed in formal languages can be recognized by Turing machine without using biochemistry [38]. Recently, Bhatia and Zheng [55] modeled hairpin loop, pseudoknot, and dumbbell RNA secondary structures using 2QCFA.

2 PRELIMINARIES

In this section, some preliminaries are given. We assume that the reader is familiar with the classical automata theory and the concept of quantum computation; otherwise, reader can refer to the theory of automata [6], quantum information, and computation [5, 56]. Linear algebra is inherited from quantum mechanics to describe the field of quantum computation. It is a crucial mathematical tool and allows us to represent the quantum operations and quantum states by matrices and vectors, respectively, that obey the rules of linear algebra. The following are the notions of linear algebra used in quantum computational theory:

- Vector space (V) [56]: A vector space (V) is defined over the field F of complex numbers $\mathbb{C}$ consisting of a nonempty set of vectors, satisfying the following operations:
- Addition: If two vectors $|a\rangle$ and $|b\rangle$ belong to V, then $|a\rangle + |b\rangle \in V$.
- Multiplication by a scalar: If $|a\rangle$ belongs to V, then $\lambda|a\rangle \in V$, where $\lambda \in \mathbb{C}$.
- Dirac notation [5]: In quantum mechanics, the Dirac notation is one of the most peculiarities of linear algebra. The combination of vertical and angle bars ($|\rangle$ $\langle|$) is used to unfold quantum states. It provides an inner product of any two vectors. The bra $\langle b|$ and ket $|a\rangle$ represent the row vector and column vector, respectively.

$$|a\rangle = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}, \langle b| = \begin{bmatrix} \beta_1^* & \beta_2^* & \beta_3^* \end{bmatrix}, |a\rangle\langle b| = \begin{bmatrix} \alpha_1\beta_1^* & \alpha_1\beta_2^* & \alpha_1\beta_3^* \\ \alpha_2\beta_1^* & \alpha_2\beta_2^* & \alpha_2\beta_3^* \\ \alpha_3\beta_1^* & \alpha_3\beta_2^* & \alpha_3\beta_3^* \end{bmatrix} \quad [1]$$

where β_i^* indicates the complex conjugate of complex number α_i.

- Quantum bit [34]: A quantum bit (qubit) is a unit vector defined over complex vector space $\mathbb{C}^2$. In general, it is represented as a superposition of two basis states labeled $|0\rangle$ and $|1\rangle$.

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \quad [2]$$

- The probability of state occurrence $|0\rangle$ is $|\alpha|^2$ and $|1\rangle$ is $|\beta|^2$. It satisfies that $|\alpha|^2 + |\beta|^2 = 1$. The two complex amplitudes (α and β) are represented by one qubit. Thus, 2^n complex amplitudes can be represented by n qubits.
- Quantum state [5]: A quantum state $|\psi\rangle$ is defined as a superposition of classical states

$$|\psi\rangle = \alpha_1|w_1\rangle + \alpha_2|w_2\rangle + \ldots + \alpha_n|w_n\rangle \quad [3]$$

where α_i s are complex amplitudes and $|w_i\rangle$s are classical states for $1 \le i \le n$. Therefore, a quantum state $|\psi\rangle$ can be represented as n-dimensional column vector.

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \ldots \\ \alpha_n \end{bmatrix} \quad [4]$$

- Unitary transformation: In quantum mechanics, the transformation between the quantum systems must be unitary. Consider a state $|\psi\rangle$ of quantum system at time t: $|\psi\rangle = \alpha_1|w_1\rangle + \alpha_2|w_2\rangle + \ldots + \alpha_n|w_n\rangle$ transformed into state $|\psi'\rangle$ at time t': $|\psi'\rangle = \alpha_1'|w_1\rangle + \alpha_2'|w_2\rangle + \ldots + \alpha_n'|w_n\rangle$, where complex amplitudes are associated by $|\psi'(t')\rangle = U(t'-t)|\psi(t)\rangle$, where U denotes a time reliant unitary operator, which satisfies that $(U(t'-t))^* U(t'-t) = 1$ and $\sum_{i=1}^{n}|\alpha_i|^2 = |\alpha_i'|^2 = 1$ [5].
- Hilbert space: A physical system is described by a complex vector space called Hilbert space $\mathcal{H}$ [56]. It allows us to describe the basis of the quantum system. The direct sum $|x|y\rangle : \mathcal{H}\oplus\mathcal{H} \to \mathbb{C}$ or inner product $|x|v\rangle : \mathcal{H}\otimes\mathcal{H} \to \mathbb{C}$ of two subspaces satisfies the following properties for any vectors:
- Linearity: $(\alpha\langle x| + \beta\langle y|)|z\rangle = \alpha|x|z\rangle + \beta|y|z\rangle$.
- Symmetric property: $|x|y\rangle = |y|x\rangle$.
- Positivity: $|x|x\rangle \ge 0$ and $|x|x\rangle = 0$ iff $x = 0$, where $x \in \mathcal{H}$.
- where $x, y, z \in \mathcal{H}$ and $\alpha, \beta \in \mathbb{C}$.

- Quantum finite automaton (QFA) [57]: It is defined as a quintuple $(Q, \Sigma, s_{int}, P_{acc}, U_\sigma)$, where
- Q is a set of states,
- Σ is an input alphabet,
- Hilbert space $\mathcal{H}$ and $s_{init} \in \mathcal{H}$ is an initial vector such that $|s_{init}|^2 = 1$,
- $\mathcal{H}_{acc} \subset \mathcal{H}$ and P_{acc} is an acceptance projection operator on $\mathcal{H}_{acc}$,
- U_σ denotes a unitary transition matrix for each input symbol ($\sigma \in \Sigma$).

The computation procedure of QFA consists of an input string $w = \sigma_n \sigma_2 \ldots \sigma_n$. The automaton works by reading each input symbol, and their respective unitary matrices are applied on the current state, starting with an initial state. The quantum language accepted by QFA is represented as a function $f_{QFA}(w) = |s_{init} U_w P_{acc}|^2$, where $U_w = U_{\sigma_1} U_{\sigma_2} \ldots U_{\sigma_n}$. The tape head is allowed to move only in the right direction. Finally, the probability of QFA in an acceptance state is observed: that is, indicating whether the input string is accepted or rejected by QFA. It is also called a real-time quantum finite automaton.

Based on the movement of tape head, QFA is classified as one-way QFA, 1.5-way QFA, and 2QFA. In 1.5-way QFA, the tape head is permitted to move only in the right direction or can be stationary, but it cannot move toward the left direction. It has been proved that it can be designed for non–context-free languages, if the input tape is circular [58]. In this study, we focused on the 2QFA model due to the high computational power than its classical counterparts.

3 TWO-WAY QUANTUM FINITE AUTOMATA

A quantum finite automaton (QFA) is a quantum variant of a classical finite automaton. In QFA, quantum transitions are applied by reading the input symbols from the tape [9]. Two-way quantum finite automaton (2QFA) is a quantum counterpart of a two-way deterministic finite automaton (2DFA). In 2QFA, the tape head is allowed to move either in the left direction or right direction or can be stationary. The illustration of 2DFA is shown in **Figure 1**.

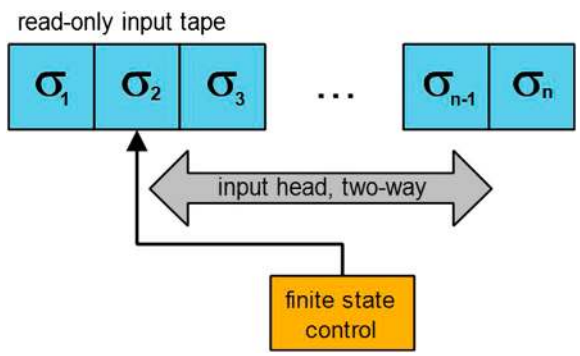

FIGURE 1 | Representation of two-way deterministic finite automata.

1. [9] A two-way quantum finite automaton is represented as sextuple $(Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$, where

- *Q is a finite set of states.*
- *Σ is an input alphabet.*
- *Transition function δ is defined by $\delta : Q \times \Gamma \times Q \times D \to \mathbb{C}$, where $\mathbb{C}$ is a complex number, $\Gamma = \Sigma \cup \{\#$, and $D = \{-1, 0, +1\}$ represent the left, stationary and right direction of tape head.*
- $Q = Q_{acc} \cup Q_{rej} \cup Q_{non}$, where Q_{non}, Q_{acc}, and Q_{rej} represent the set of nonhalting, accepting, and rejecting states, respectively. The transition function must satisfy the following conditions:
- (i) Local probability and orthogonality condition:

$$\sum_{\substack{\forall (q_1,\sigma_1),(q_2,\sigma_2) \in Q\times\Gamma \\ (q',d) \in Q\times D}} \overline{\delta(q_1,\sigma,q',d)}\,\delta(q_2,\sigma,q',d) = \begin{Bmatrix} 1 & q_1 = q_2 \\ 0 & q_1 \neq q_2 \end{Bmatrix}$$

(ii) First separability condition:

$$\begin{aligned} &\sum_{\forall (q_1,\sigma_1),(q_2,\sigma_2) \in Q\times\Gamma,\ q' \in Q} \overline{\delta(q_1,\sigma_1,q',+1)}\,\delta(q_2,\sigma_2,q',0) \\ &+ \overline{\delta(q_1,\sigma_1,q',0)}\,\delta(q_2,\sigma_2,q',-1) \\ &= 0 \end{aligned}$$

(iii) Second separability condition:

$$\sum_{\forall (q_1,\sigma_1),(q_2,\sigma_2) \in Q\times\Gamma,\ q' \in Q} \overline{\delta(q_1,\sigma_1,q',+1)}\,\delta(q_2,\sigma_2,q',-1) = 0$$

For each $\sigma \in \Gamma$, a 2QFA is said to be simplified, if there exists a unitary linear operator V_σ on the inner product space such that $L_2\{Q\} \to L_2\{Q\}$. The transition function is represented as

$$\delta(q,\sigma,q',d) = \left\{ \begin{matrix} q' V_\sigma q \\ 0 \end{matrix} \middle| \begin{matrix} \text{if } D(q') = d \\ \text{else} \end{matrix} \right\}, \qquad [5]$$

where $q' V_\sigma q$ is a coefficient of $|q'\rangle$ in $V_\sigma |q\rangle$.

Consider an input string w, written on the input tape enclosed with both end markers such as $\#w$. The computation of 2QFA is as follows. The tape head is above the input symbol σ, and the automaton is in any state q. Then, the state of 2QFA is changed to q' with an amplitude $\delta(q,\sigma,q',d)$ and moves the tape head one cell toward right, stationary, and left direction according to $\in \{-1, 0, +1\}$. It corresponds to a unitary evolution in the innerproduct space $\mathscr{H}_n$.

A computation of a 2QFA is a chain of superpositions $c_0, c_1, c_2, \ldots$, where c_0 denotes an initial configuration. For any c_i, when the automaton is observed in a superposition state with an amplitude α_c, it has the form $U_\delta |c_i\rangle \sum_{c \in C_n} \alpha_c |c_i\rangle$,

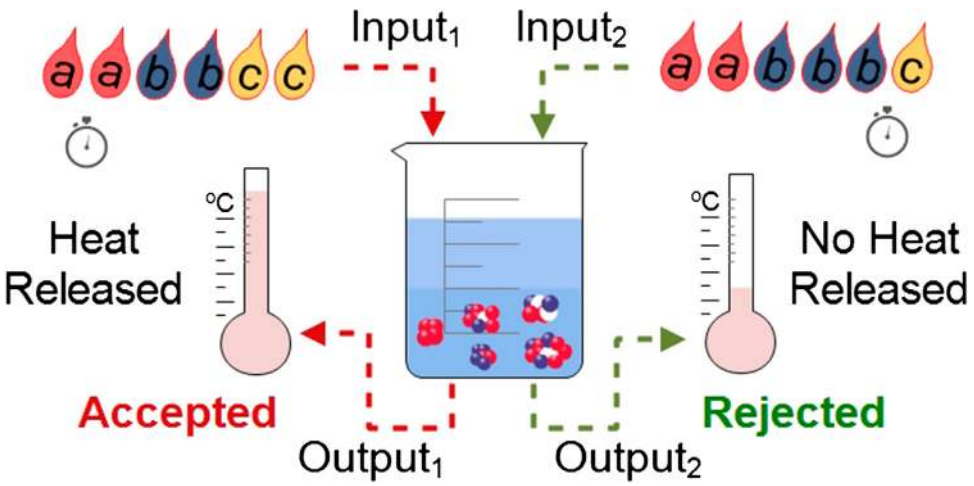

FIGURE 2 | Representation of language recognition by chemical computation. It is reproduced from [40] under the Creative Commons CCBY license.

where C_n represents the set of configurations. The probability associated with a configuration is calculated by absolute squares of amplitude. Superposition is said to be valid if the sum of the squared moduli of their probability amplitudes is unitary. In quantum theory, the time evolution is specified by unitary transformations. Each transition function δ prompts a transformation operator over the Hilbert space $\mathscr{H}_n$ in linear time.

$$U_\delta^w \left| q,j\right\rangle = \sum_{(q',d)\,\in\, Q\times D} \delta\left(q, w(j), q', d\right) \left| q', j + d \ mod |w|\right\rangle$$

for each $(q,j) \in C_{|w|}$, where $q \in Q, j \in Z_{|w|}$ and extended to $\mathscr{H}_n$ by linearity [9, 59].

4 MODELING OF CHEMICAL REACTIONS

Before we recognize the chemical reactions using two-way QFAs model, it is important to show how computational chemistry works. **Figure 2** shows the illustration of language recognition by the chemical computation model. It consists of three parts: i) a mixed container where the computation process occurs, ii) an input translator that translates the chemical aliquots into input symbols and gives those consecutively depending upon the processing time, iii) a system to monitor the response of an automaton as a chemical criterion. Finally, the chemical computation produces well-defined chemical accept/reject signatures for the input. For instance, if the number of *a*s and *b*s are equal in the input, then the chemical computation produces heat, that is, an input is said to be accepted. Otherwise, if no heat is released at the end of computation, then the input is said to be rejected by the system. The following are the construction of two-way quantum finite state machines of chemical reactions.

THEOREM 1. Two-way QFAs can recognize all regular languages.

PROOF. The proof has been shown in Ref. 9.

4.1 Chemical Reaction-1 Consisting Regular Language

For an illustrative and visual implementation, we can choose a precipitation reaction in an aqueous medium such as

$$KIO_3 + AgNO_3 \rightarrow AgIO_3\,(s) + KNO_3 \qquad [6]$$

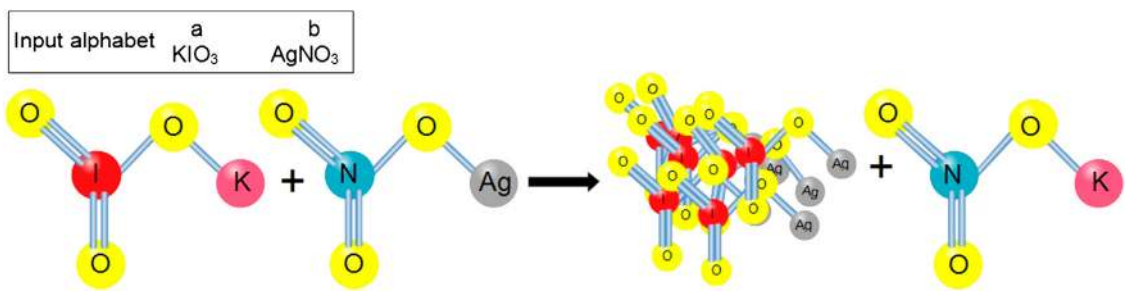

FIGURE 3 | Illustration of acid/base reaction of L_1.

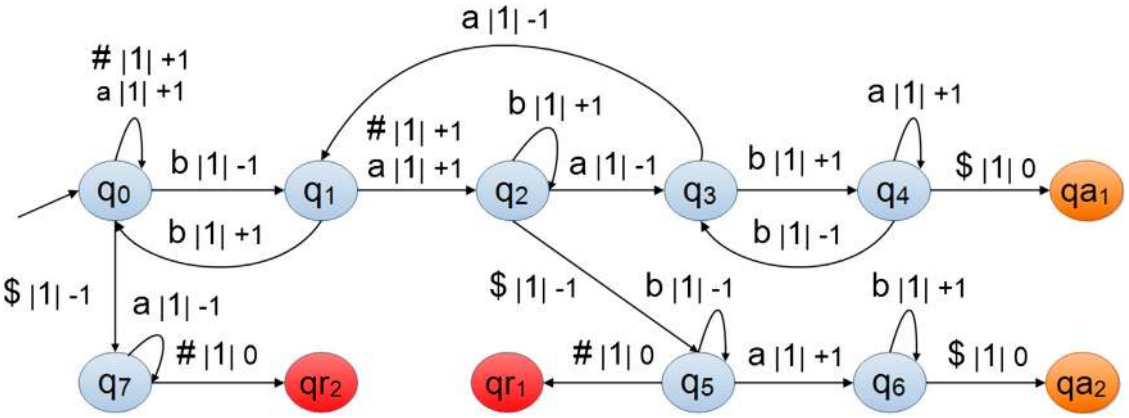

FIGURE 4 | State transition diagram of L_1.

During computation, if a white precipitate of silver iodate is observed, then the input string is said to be accepted; if the solution is clear from precipitate, the string has been rejected because there was no reaction. Therefore, we have chosen the recipes of alphabet symbols a for potassium iodate (KIO_3) and b for silver nitrate ($AgNO_3$) quantitatively. **Figure 3** shows the chemical representation of symbols a and b, the bimolecular precipitation reaction [38]. If the precipitate $AgIO_3$ is not presented in the solution, then the computation is said to be rejected. For example, the input string $w = aaab$ is said to be accepted due to the presence of precipitate or, equally, heat has been determined during computation. But, the input $w = aa$ is said to be rejected due to the absence of precipitate or, precisely, heat has not been observed. The Kleene star (Σ^*) operator is a set of infinite strings of all lengths over input alphabet as well as empty string (ϵ). The language "$(a+b)^*$" means the string containing any number of "a"s or "b"s in any order or empty string. The language "$(ab)^*$" means the string containing any number of "ab"s or string of length zero. **Figure 4** shows the corresponding theoretical 2QFA state transition graph to recognize L_1.

THEOREM 2. A language $L_1 = \{(a+b)^* a (a+b)^* b (a+b)^* aa^* bb^*\}$ representing precipitation reaction in ***Eq. 6*** can be recognized by 2QFA.

PROOF. The idea of this proof is as follows. The initial state q_0 reads a right-end marker # and moves the head toward the right direction. If there is no occurrence of symbol b, then it shows no precipitate, and the input is said to be rejected by the 2QFA. Similarly, on reading the symbol b, the state q_0 is changed into q_1. If there is no occurrence of symbol a, then the state is transformed into rejecting state q_{r_1}. If the input string $w \in L_1$ contains at least one a and one b, then silver iodate is present during computation, and it is said to be recognized by 2QFA. A 2QFA for L_1 is defined as follows: $M_{2QFA} = (Q, \Sigma, q_0, Q_{acc}, Q_{rej}, \delta)$,

where $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_{a_1}, q_{a_2}, q_{r_1}, q_{r_2}\}$, where q_0 and q_2 are used to move the head toward the \$ on reading as and bs, respectively. The states q_1 and q_3 are used to confirm that the last symbol read by head is a and b, respectively.

$\Sigma = \{a, b\}$, q_0 is an initial state, $Q_{acc} = \{q_{a_1}, q_{a_2}\}$ and $Q_{rej} = \{q_{r_1}, q_{r_2}\}$.

The specification of transition functions are given in **Table 1**.

It can be noted that in 2QFA, transition matrices consist of 0 and 1, i.e., basically a two-way reversible finite automata (2RFA). Therefore, 2QFA can be designed for all the languages accepted by 2RFA. In transition matrix, each column and row have exactly only one entry 1. Hence, the dot product of any two rows is equal to zero. It is known that the language recognition power of 2RFA is an equivalent to 2DFA.

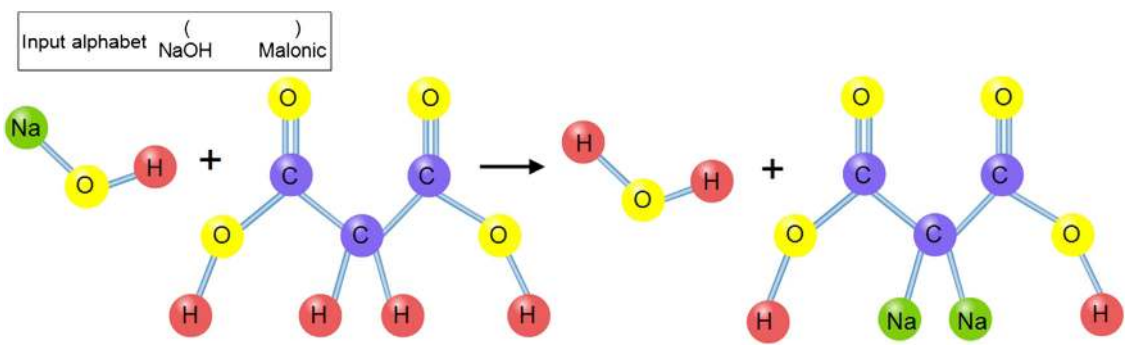

FIGURE 5 | Illustration of the acid/base reaction of L_2.

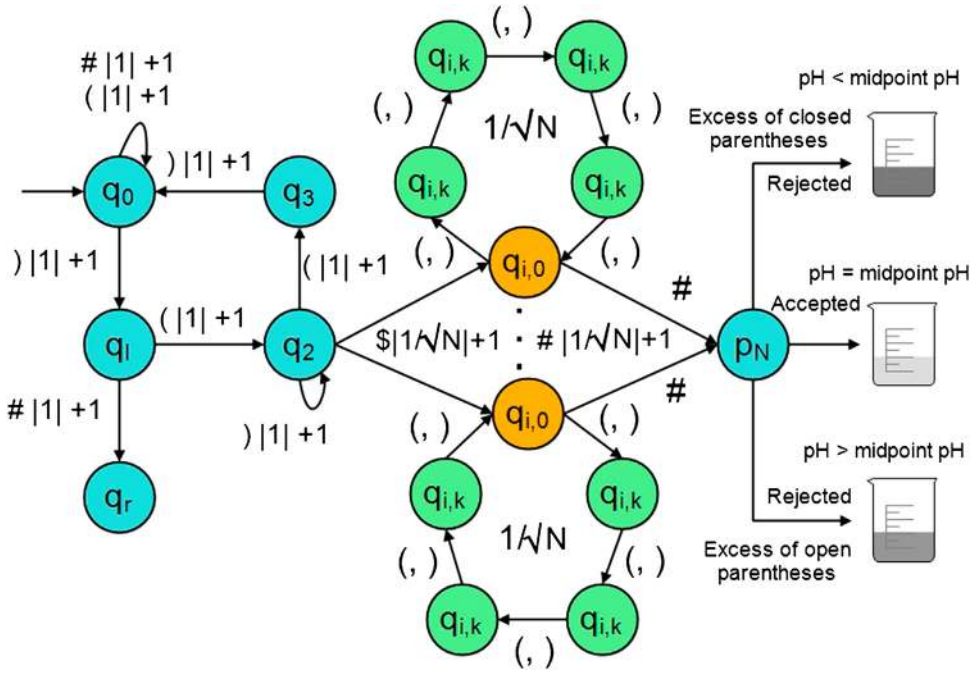

FIGURE 6 | State transition diagram of L_2.

4.2 Chemical Reaction-2 Consisting Context-Free Language

Next, we have considered the context-free language from Chomsky hierarchy satisfying the balanced chemical reaction between NaOH and malonic acid as follows:

$$H_2C_3H_2O_4 + 2NaOH \rightarrow Na_2C_3H_2O_4 + 2H_2O \qquad [7]$$

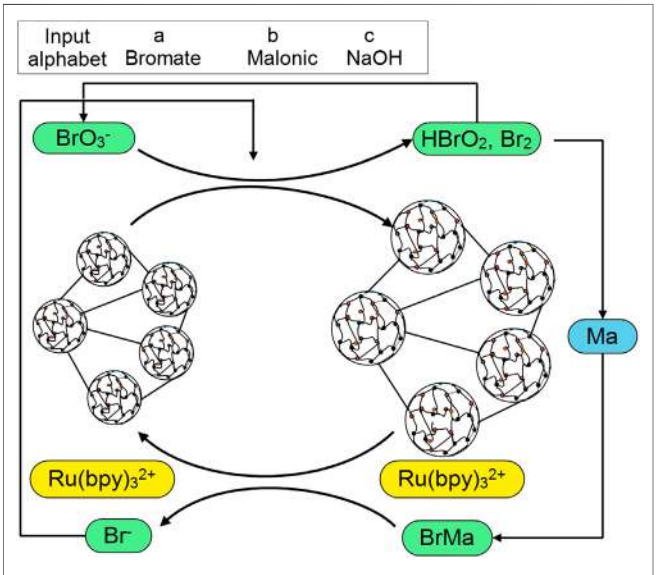

FIGURE 7 | Illustration of the acid/base reaction of L_3.

The language generated by the abovementioned chemical reaction is L_2 consisting Dyck language of all words with balanced parentheses. Figure 5 shows the acid/base reaction of L2. 2QFA is designed for L_2 as follows:

THEOREM 3. A language L_2 consisting of Dyck language of all words with balanced parentheses can be recognized by 2QFA with probability 1, otherwise rejected with probability at least $1 - \frac{1}{N}$, where N is any positive number.

PROOF. The idea of this proof is as follows. It consists of three phases. First, the initial state q_0 reads a first symbol and both heads start moving toward the right-end marker $. If the input string starts with closed parentheses, then it is said to be rejected. On reading the left-end marker #, the computation is split into N paths, denoted by $q_{1,0}, q_{2,0}, \ldots, q_{N,0}$. Each path possesses equal amplitude $\frac{1}{\sqrt{N}}$. Along the N different paths, each path moves deterministically to the right-end marker . Each computational path keeps track of the open parentheses with respect to the closed parentheses. At the end of computation, if the excess of open parentheses is observed, then it is said to be rejected. It means pH value is greater than midpoint pH, and intermediate gray tone is observed. Secondly, if there is an excess of closed parentheses, then the darkest gray tone is observed, that is, pH value is less than midpoint pH. It is said to be rejected by 2QFA with probability $1 - \frac{1}{N}$. If there is a balanced occurrence of open and closed parentheses, the input string is said to be accepted with probability 1. Hence, pH value is equal to midpoint pH, and the lightest gray tone is observed at the end of computation. A 2QFA for L_2 is defined as follows: $M_{2QFA} = (Q, \Sigma, q_0, Q_{acc}, Q_{rej}, \delta)$,where $Q = \{q_0, q_1, q_2, q_3\} \cup \{q_{i,j} | 1 \le i \le N, 0 \le j \le \max(i, N - i + 1)\} \cup \{p_k | 1 \le k \le N\} \cup \{s_{i,0}, w_{i,0}, r_{i,0} | 1 \le i \le N\}$

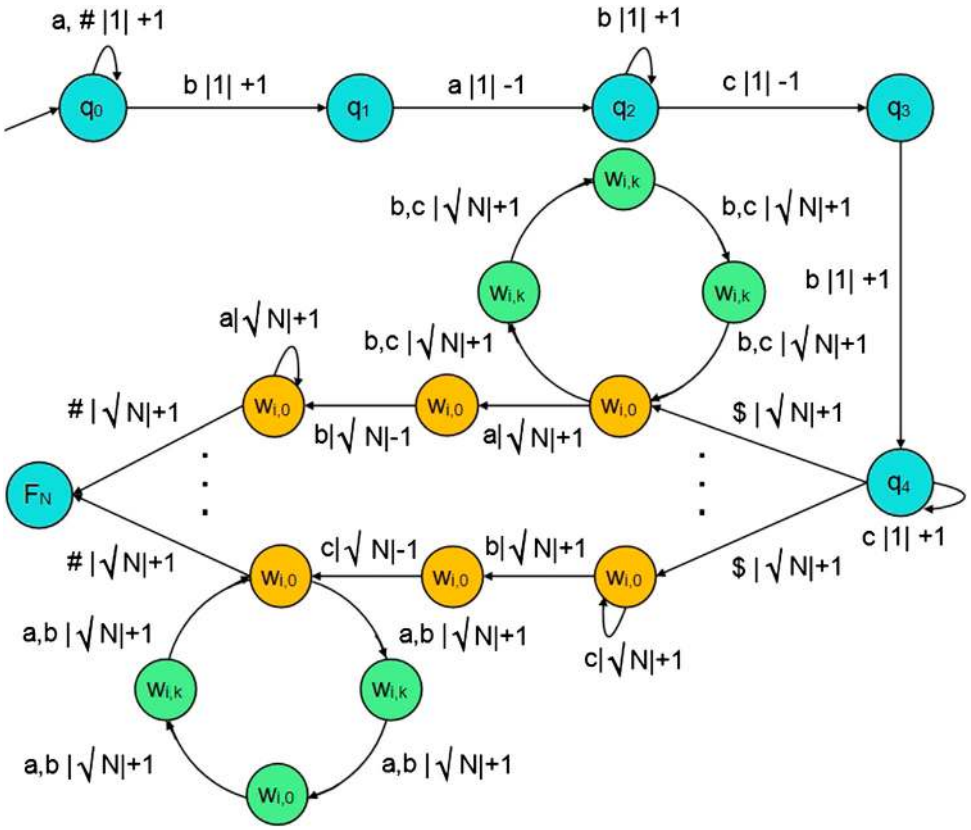

FIGURE 8 | State transition diagram of L_3.

$\cup \{q_{acc}, q_{rej}\}$, where q_1 is used to check whether the first symbol is an open parentheses or not, and q_2 and q_3 are used to traverse the input string. Figure 6 shows the state transition diagram for L2.

$\Sigma = \{(,)\}, q_0$ is an initial state, $Q_{acc} = \{p_N\}$ and $Q_{rej} = \{q_r\} \cup \{p_k \big| 1 \le k < N\} \cup \{r_{i,0} \big| 1 \le i \le N\}$.

The specification of transition functions is given in **Table 2**.

4.3 Chemical Reaction-3 Consisting Context-Sensitive Language

To implement a chemical 2QFA for context-sensitive language, we have used Belousov-Zhabotinsky (BZ) reaction network for the nonlinear oscillatory chemistry [38], which consists of temporal oscillation in the sodium bromate and malonic acid system [60] as Figure 7 shows the acid/base reaction of L3. Figure 8 shows the state transition diagram for L3.

$$3BrO_3^- + 5CH_2(COOH)_2 + 3H^+ \rightarrow 3BrCH(COOH)_2 + 4CO_2 + 2HCOOH + 5H_2O \quad [8]$$

In 2019, Duenas-Diez and Perez-Mercader [38] designed chemical Turing machine for BZ reaction network. The chemical reaction is fed sequentially to the reactor as $\{(BrO_3^-)^n (MA)^n (NaOH)^n\}$, where $n > 0$. It is transcribed in formal language as $L_3 = \{a^n b^n c^n | n > 0\}$. The symbol a is interpreted as a fraction of sodium bromate, b is used for malonic acid and symbol, c is transcribed as a quantity of NaOH. It is known that L_3 is a context-sensitive language and cannot be recognized by finite automata or pushdown automata with a stack. Although it can be recognized by two-stack PDA, we have shown that L_3 can be recognized by 2QFA without using any external aid.

THEOREM 4. A language $L_3 = \{a^n b^n c^n | n > 0\}$ can be recognized by 2QFA in linear time. For a language, $L_3 = \{a^n b^n c^n | n > 0\}$, and for arbitrary N-computational paths, there exists a 2QFA such that for $w \in L_3$; it accepts w with bounded error ϵ and rejects $w \notin L_3$ with probability at least $1 - \frac{1}{N}$.

TABLE 1 | Details of the transition functions and head function for L_1.

$V_{\#}\|q_0\rangle = \|q_0\rangle$	$V_a\|q_0\rangle = \|q_0\rangle$	$V_b\|q_0\rangle = \|q_1\rangle$	$V_{\#}\|q_1\rangle = \|q_2\rangle$
V	$V_a\|q_1\rangle = \|q_2\rangle$	$V_b\|q_1\rangle = \|q_2\rangle$	$V_b\|q_2\rangle = \|q_2\rangle$
V	$V_a\|q_7\rangle = \|q_7\rangle$	$V_a\|q_2\rangle = \|q_3\rangle$	$V_a\|q_3\rangle = \|q_1\rangle$
V	$V_a\|q_4\rangle = \|q_4\rangle$	$V_b\|q_3\rangle = \|q_4\rangle$	$V_b\|q_4\rangle = \|q_3\rangle$
$V_{\#}\|q_5\rangle = \|q_{r_1}\rangle$	$V_a\|q_5\rangle = \|q_6\rangle$	$V_b\|q_5\rangle = \|q_5\rangle$	$V_b\|q_6\rangle = \|q_6\rangle$
V	$V_a\|q_6\rangle = \|q_{a_2}\rangle$		

Head functions:

$$D(q_0) = (+1), D(q_1) = (-1), D(q_2) = (+1), D(q_3) = (-1),$$
$$D(q_4) = (+1), D(q_5) = (-1), D(q_6) = (+1), D(q_7) = (-1),$$
$$D(q_{a_1}) = D(q_{a_2}) = (0), D(q_{r_1}) = D(q_{r_2}) = (0)$$

TABLE 2 | Details of the transition functions and head function for L_2.

$$V_{\#}|q_0\rangle = |q_0\rangle, V_{(}|q_0\rangle = |q_0\rangle, V_{)}|q_0\rangle = |q_1\rangle, V_{\#}|q_1\rangle = |q_r\rangle$$
$$V_{(}|q_1\rangle = |q_2\rangle, V_{)}|q_2\rangle = |q_2\rangle, V_{(}|q_2\rangle = |q_3\rangle, V_{)}|q_3\rangle = |q_0\rangle$$
$$V_{\#}|q_2\rangle = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}|q_{i,0}\rangle$$
$$V_{(}|q_{i,0}\rangle = |q_{i,i}\rangle, V_{)}|q_{i,0}\rangle = |q_{i,N-i+1}\rangle, \text{for } 1 \le i \le N, V_{(}|q_{i,j}\rangle = |q_{i,j-1}\rangle, \text{for } 1 \le j \le i$$
$$V_{)}|q_{i,j}\rangle = |q_{i,j-1}\rangle, \text{for } 1 \le j \le N-i+1, 1 \le i \le N$$
$$V_{\$}|q_{i,0}\rangle = |s_{i,0}\rangle, V_{)}|s_{i,0}\rangle = |w_{i,0}\rangle, V_{(}|s_{i,0}\rangle = |r_{i,0}\rangle, \text{for } 1 \le i \le N$$

$$V_{\$}|w_{i,0}\rangle = \frac{1}{\sqrt{N}}\sum_{k=1}^{N} exp\left(\frac{2\pi i}{N}ki\right)\Big|p_k\rangle, \text{for } 1 \le i \le N$$

Head functions:

$$D(q_0) = (+1), D(q_1) = (-1), D(q_2) = (+1), D(q_3) = (-1), D(q_{i,0}) = (+1), \text{for } 1 \le i \le N$$
$$D(q_{i,j}) = (0), \text{for } 1 \le i \le N, j \ne 0, D(q_r) = (0), D(p_k) = (0), \text{for } 1 \le k \le N$$
$$D(s_{i,0}) = (-1), D(w_{i,0}) = (+1), D(r_{i,0}) = (0), \text{for } 1 \le i \le N$$

PROOF. The design of proof for BZ reaction network is as follows. It consists of two phases. First, the 2QFA traverse the input to check the form $a^+b^+c^+$. On reading the right-end marker \$, the computation is split into N paths such that $w_{1,0}, w_{2,0}, \ldots, w_{N,0}$. Second, the first path is used to check if the number of bs and cs are equal or not. The second path is used to check the initial part of an input string to identify if it is in $\{a^n b^n | n > 0\}$. On reading the right-end marker \$, the both paths are split into N different paths with an equal amplitude $\frac{1}{\sqrt{N}}$. Finally, upon reading the right-end marker #, if the number of as and bs and the number of bs and cs are equal in respective computational paths, then all paths come into N-way quantum Fourier transform (QFT) and either one acceptance state or rejection states are observed. Suppose, if the input string is not in the corrected form, then all computation paths read the # at different times. Thus, their amplitudes do not cancel each other, and the input string is said to be rejected with probability $1 - \frac{1}{N}$. Otherwise, the input string is said to be recognized by 2QFA with probability 1.

5 SUMMARY

In summary, 2QFA model can be efficiently designed for balanced chemical reaction and BZ reaction network with one-sided error bound, which are halted in linear time. **Table 3** shows the language recognition ability of different computational models. The classical 2DFA and 2PFA are known to be equal in computational power to one-way deterministic finite automata (1DFA) [24, 61]. It has been proved that 2PFAs can be designed for nonregular languages in expected polynomial time. Additionally, it has been demonstrated that the chemical PDA can be designed for aforementioned chemical reactions with multiple stacks. The recognition of languages by native chemical automata can be found in Refs. 39–41. But, we have shown that 2QFA can recognize such chemical reactions without any external aid. It has been proved that 2QFA is more powerful than classical variants because it follows the quantum superposition principle to be in more than one state at a time on the input tape. For execution, it needs at least $O(log\ n)$ quantum states to store the position of tape head, where n denotes the length of an input string.

TABLE 3 | Comparison of computational power of models.

Languages	Class	2DFA/ 2PFA	Chemical FA/PDA	2QFA
$L_1 = \{(a+b)^*a(a+b)^*b(a+b)^*aa^*bb^*\}$	RL	✓	✓	✓
A language L_2 consisting Dyck language of balanced parentheses	CFL	✗	✓ (with 1-stack PDA)	✓
$L_3 = \{a^n b^n c^n \| n > 0\}$	CSL	✗	✓ (with 2-stack PDA)	✓

RL, CFL, and CSL stand for regular languages, context-free languages, and context-sensitive languages, respectively.

6 CONCLUSION

The enhancement in many existing computational approaches provides momentum to molecular and quantum simulations at the electronic level. It helps to test new abstract approaches for considering molecules and matter. Previous attempts to model the aforementioned chemical reactions used finite automata and pushdown automata with multiple stacks. In this study, we focused on well-known languages of Chomsky hierarchy and modeled those using two-way QFAs. The crucial advantage of the quantum approach is that these chemical reactions transcribed in formal languages can be parsed in linear time, without using any external aid. We have shown that two-way quantum automata are more superior to its classical variants by using quantum transitions. To the best of our knowledge, no such modeling of chemical reactions is performed using quantum automata theory so far. For the future purpose, we will try to represent complex chemical reactions in formal languages and model those using other quantum computational models.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

AB and SZ modeled the chemical reactions. AB wrote the manuscript under guidance of SZ.

ACKNOWLEDGMENTS

SZ acknowledges support in part from the National Natural Science Foundation of China (No. 61602532), the Natural Science Foundation of Guangdong Province of China (No. 2017A030313378), and the Science and Technology Program of Guangzhou City of China (No. 201707010194). This work was also supported in part by NSF-China (61772570), Pearl River S&T Nova Program of Guangzhou (201806010056), and Guangdong Natural Science Funds for Distinguished Young Scholar (2018B030306025).

REFERENCES

1. Conrad M. Information processing in molecular systems. *BioSystems* (1972) **5**: 1–14. doi:10.1016/0303-2647(72)90011-1
2. Benenson Y. Biocomputers: from test tubes to live cells. *Mol Biosyst.* (2009) **5**: 675–685. doi:10.1039/b902484k
3. Prohaska SJ, Stadler PF, Krakauer DC. Innovation in gene regulation: the case of chromatin computation. *J Theor Biol.* (2010) **265**:27–44. doi:10.1016/j.jtbi.2010.03.011
4. Bryant B. Chromatin computation. *PLoS One.* (2012) **7**:e35703. doi:10.1371/journal.pone.0035703
5. Nielsen MA, Chuang I. *Quantum computation and quantum information*. New York: AAPT (2002).
6. Hopcroft JE, Motwani R, Ullman JD. *Automata theory, languages, and computation*. **Vol. 24**. New York: Pearson addison wesley (2006). 19 p.
7. Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* (1999) **41**:303–332. doi:10.1137/s0036144598347011
8. Kondacs A, Watrous J. On the power of quantum finite state automata. In: Proceedings 38th annual symposium on foundations of computer science; 1997 Oct 20–22; Miami Beach, FL. IEEE (1997) p. 66–75.
9. Moore C, Crutchfield JP. Quantum automata and quantum grammars. *Theor Comput Sci.* (2000) **237**:275–306. doi:10.1016/s0304-3975(98)00191-1
10. Bhatia AS, Kumar A. On the power of quantum queue automata in real-time. arXiv preprint arXiv:1810.12095 (2018).
11. Ambainis A, Watrous J. Two-way finite automata with quantum and classical states. *Theor Comput Sci.* (2002) **287**:299–311. doi:10.1016/s0304-3975(02)00138-x
12. Zheng S, Qiu D, Li L, Gruska J. *One-way finite automata with quantum and classical states. Languages alive*. Berlin, Heidelberg: Springer (2012) 273–290 p.
13. Mereghetti C, Palano B, Pighizzini G. Note on the succinctness of deterministic, nondeterministic, probabilistic and quantum finite automata. *RAIRO Theor Inf Appl.* (2001) **35**:477–490. doi:10.1051/ita:2001106
14. Zheng S, Qiu D, Gruska J, Li L, Mateus P. State succinctness of two-way finite automata with quantum and classical states. *Theor Comput Sci.* (2013) **499**: 98–112. doi:10.1016/j.tcs.2013.06.005
15. Yakaryilmaz A, Say A. *Succinctness of two-way probabilistic and quantum finite automata*. arXiv preprint arXiv:0903.0050 (2009).
16. Nishimura H, Yamakami T. An application of quantum finite automata to interactive proof systems. *J Comput Syst Sci.* (2009) **75**:255–269. doi:10.1016/j.jcss.2008.12.001
17. Nishimura H, Yamakami T. Interactive proofs with quantum finite automata. *Theor Comput Sci.* (2015) **568**:1–18. doi:10.1016/j.tcs.2014.11.030
18. Bhatia AS, Kumar A. Quantifying matrix product state. *Quant Inf Process.* (2018) **17**:41. doi:10.1007/s11128-017-1761-1
19. Zheng S, Li L, Qiu D, Gruska J. Promise problems solved by quantum and classical finite automata. *Theor Comput Sci.* (2017) **666**:48–64. doi:10.1016/j.tcs.2016.12.025
20. Gruska J, Qiu D, Zheng S. Generalizations of the distributed deutsch–jozsa promise problem. *Math Struct Comput Sci.* (2017) **27**:311–331. doi:10.1017/s0960129515000158
21. Gruska J, Qiu D, Zheng S. Potential of quantum finite automata with exact acceptance. *Int J Found Comput Sci.* (2015) **26**:381–398. doi:10.1142/s0129054115500215
22. Gainutdinova A, Yakaryılmaz A. Unary probabilistic and quantum automata on promise problems. *Quant Inf Process.* (2018) **17**:28. doi:10.1007/s11128-017-1799-0
23. Bhatia AS, Kumar A. Quantum ω-automata over infinite words and their relationships. *Int J Theor Phys.* (2019) **58**:878–889. doi:10.1007/s10773-018-3983-0
24. Shepherdson JC. The reduction of two-way automata to one-way automata. *IBM J Res Dev.* (1959) **3**:198–200. doi:rd.32.0198/rd.32.0198
25. Zheng S, Gruska J, Qiu D. On the state complexity of semi-quantum finite automata. *RAIRO Theor Inf Appl.* (2014) **48**:187–207. doi:10.1051/ita/2014003
26. Li L, Qiu D. Lower bounds on the size of semi-quantum finite automata. *Theor Comput Sci.* (2016) **623**:75–82. doi:10.1016/j.tcs.2015.09.031
27. Zheng S, Qiu D, Gruska J. Time-space complexity advantages for quantum computing. In: International conference on theory and practice of natural computing; 2017 Dec 18-20, 2017; Prague, Czech Republic Springer (2017). p. 305–317. doi:10.1007/978-3-319-71069-3_24
28. Yamakami T. Relativizations of nonuniform quantum finite automata families. In: International conference on unconventional computation and natural computation; Springer (2019). p. 257–271. doi:10.1007/978-3-030-19311-9_20
29. Yamakami T. Nonuniform families of polynomial-size quantum finite automata and quantum logarithmic-space computation with polynomial-size advice. In: International conference on language and automata theory and applications; 2019 Jun 3-7; Tokyo Springer (2019). p. 134–145. doi:10.1007/978-3-030-13435-8_10
30. Bhatia AS, Kumar A. On relation between linear temporal logic and quantum finite automata. *J Logic Lang Inf.* (2019) **29**:109–120. doi:10.1007/s10849-019-09302-6
31. Bertoni A, Mereghetti C, Palano B. Quantum computing: 1-way quantum automata. In: International conference on developments in language theory; Springer (2003). p. 1–20. doi:10.1007/3-540-45007-6_1
32. Qiu D, Li L. An overview of quantum computation models: quantum automata. *Front Comput Sci China.* (2008) **2**:193–207. doi:10.1007/s11704-008-0022-y
33. Ambainis A, Yakaryılmaz A. Automata and quantum computing. arXiv preprint:1507.01988 (2015).
34. Bhatia AS, Kumar A. *Quantum finite automata: survey, status and research directions*. arXiv preprint arXiv:1901.07992 (2019).
35. Qiu D, Li L, Mateus P, Sernadas A. Exponentially more concise quantum recognition of non-rmm regular languages. *J Comput Syst Sci.* (2015) **81**: 359–375. doi:10.1016/j.jcss.2014.06.008
36. Singh Bhatia A. *On some aspects of quantum computational models*. [PhD thesis]. Patiala (India): Thapar Institute of Engineering and Technology (2020).
37. Say AC, Yakaryılmaz A. Quantum finite automata: a modern introduction. *Computing with new resources*. New York: Springer (2014). p. 208–222.
38. Duenas-Diez M, Perez-Mercader J. Native chemical automata and the thermodynamic interpretation of their experimental accept/reject responses. arXiv preprint:1903.03827 (2019).
39. Okubo F, Yokomori T. The computing power of determinism and reversibility in chemical reaction automata. *Reversibility and universality*. Springer (2018). p. 279–298.
40. Dueñas-Díez M, Pérez-Mercader J. How chemistry computes: language recognition by non-biochemical chemical automata. from finite automata to turing machines. *iScience* (2019) **19**:514–526. doi:10.1016/j.isci.2019.08.007
41. Freivalds R. Probabilistic two-way machines. In: International symposium on mathematical foundations of computer science; 1981 Aug 31 - Sep 4; Štrbské Pleso, Czechoslovakia Springer (1981). p. 33–45. doi:10.1007/3-540-10856-4_72
42. Dwork C., Stockmeyer L. A time complexity gap for two-way probabilistic finite-state automata. *SIAM J Comput.* (1990) **19**:1011–1023. doi:10.1137/0219069
43. Okamoto M., Sakai T., Hayashi K. Switching mechanism of a cyclic enzyme system: role as a "chemical diode". *Biosystems* (1987) **21**:1–11. doi:10.1016/0303-2647(87)90002-5
44. Hjelmfelt A, Weinberger ED, Ross J. Chemical implementation of neural networks and Turing machines. *Proc Natl Acad Sci USA*. (1991) **88**: 10983–10987. doi:10.1073/pnas.88.24.10983
45. Tóth Á, Showalter K. Logic gates in excitable media. *J Chem Phys.* (1995) **103**; 2058–2066. doi:10.1063/1.469732
46. Magnasco MO. Chemical kinetics is turing universal. *Phys Rev Lett.* (1997) **78**: 1190. doi:10.1103/physrevlett.78.1190
47. Adamatzky A, Costello BDL. Experimental logical gates in a reaction-diffusion medium: the xor gate and beyond. *Phys Rev.* (2002) **66**:046112. doi:10.1103/physreve.66.046112
48. Gorecki J, Yoshikawa K, Igarashi Y. On chemical reactors that can count. *J Phys Chem.* (2003) **107**:1664–1669. doi:10.1021/jp021041f
49. Adleman LM. Molecular computation of solutions to combinatorial problems. *Science* (1994) **266**:1021–1024. doi:10.1126/science.7973651

50. Kuppusamy L, Mahendran A. Modelling DNA and RNA secondary structures using matrix insertion–deletion systems. *Int J Appl Math Comput Sci.* (2016) **26**:245–258. doi:10.1515/amcs-2016-0017
51. Kuppusamy L, Mahendran A, Krishna SN. Matrix insertion-deletion systems for bio-molecular structures. In: International conference on distributed computing and internet technology; Feb 9-12; Bhubaneshwar, India Springer (2011). p. 301–312. doi:10.1007/978-3-642-19056-8_23
52. Krasiński T, Sakowski S, Poplawski T. Towards an autonomous multistate biomolecular devices built on DNA. In: 2014 Sixth World Congress on Nature and Biologically Inspired Computing (NaBIC 2014); 2014 Jul 30 - Aug 01; Porto, Portugal IEEE (2014). p. 23–28.
53. Khrennikov A, Yurova E. Automaton model of protein: dynamics of conformational and functional states. *Prog Biophys Mol Biol.* (2017) **130**: 2–14. doi:10.1016/j.pbiomolbio.2017.02.003
54. Bhatia AS, Kumar A. Modeling of RNA secondary structures using two-way quantum finite automata. *Chaos Solit Fractals.* (2018) **116**:332–339. doi:10.1016/j.chaos.2018.09.035
55. Bhatia AS, Zheng S. RNA-2QCFA: evolving two-way quantum finite Automata with classical states for RNA secondary structures. arXiv preprint arXiv:2007.06273 (2020).
56. Wang J. *Handbook of finite state based models and applications.* Boca Raton, FL: CRC Press (2012).
57. Dzelme-Bērziņa I. *Quantum finite automata and logic.* [PhD thesis]. Riga (Latvia):University of Latvia (2010).
58. Amano M, Iwama K. Undecidability on quantum finite automata. In: STOC '99: proceedings of the 31st annual ACM symposium on theory of computing; 1999 May 1-4; Atlanta, GA. Georgia: ACM (1999). p. 368–375.
59. Bhatia AS, Kumar A. On the power of two-way multihead quantum finite automata. *RAIRO Theor Inf Appl.* (2019) **53**:19–35. doi:10.1051/ita/2018020
60. Field RJ, Koros E, Noyes RM. Oscillations in chemical systems. ii. thorough analysis of temporal oscillation in the bromate-cerium-malonic acid system. *J Am Chem Soc.* (1972) **94**:8649–8664. doi:10.1021/ja00780a001
61. Rabin MO, Scott D. Finite automata and their decision problems. *IBM J Res Dev.* (1959) **3**:114–125. doi:10.1147/rd.32.0114

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be constructed as a potential conflict of interest.