Chaos, Solitons and Fractals 116 (2018) 332–339

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

journal homepage: www.elsevier.com/locate/chaos

Modeling of RNA secondary structures using two-way quantum finite automata

Amandeep Singh Bhatia, Ajay Kumar*

Department of Computer Science, Thapar Institute of Engineering & Technology, India

A R T I C L E I N F O

Article history:
Received 3 September 2017
Revised 13 July 2018
Accepted 17 September 2018

Keywords:
Ribonucleic acid
Hairpin loop
Internal loop
Double helix loop
Quantum finite automata
Two-way quantum finite automata

A B S T R A C T

Quantum finite automata (QFA) play a crucial role in quantum information processing theory. The representation of ribonucleic acid (RNA) and deoxyribonucleic acid (DNA) structures using theory of automata had a great influence in the computer science. Investigation of the relationship between QFA and RNA structures is a natural goal. Two-way quantum finite automata (2QFA) is more dominant than its classical model in language recognition. Motivated by the concept of quantum finite automata, we have modeled RNA secondary structure loops such as, internal loop and double helix loop using two-way quantum finite automata. The major benefit of this approach is that these sequences can be parsed in linear time. In contrast, two-way deterministic finite automata (2DFA) cannot represent RNA secondary structure loops and two-way probabilistic finite automata (2PFA) can parse these sequences in exponential time. To the best of authors knowledge this is the first attempt to represent RNA secondary structure loops using two-way quantum finite automata.

1. Introduction

Bioinformatics [1] is a field of applied science in which computational and mathematical models are applied to biology. Quantum computing is an attractive field that deals with theoretical computational systems (i.e., quantum computers), combining visionary ideas of Computer Science, Physics, and Mathematics. It is based upon the quantum phenomena of entanglement and superposition to perform operations on quantum computers. Quantum mechanics with classical finite automata gives us quantum finite automata (QFA).

It is a theoretical model with finite memory for quantum computers, which plays a vital role in performing real-time computations. Quantum automata lay down the vision of quantum processors for performing quantum actions on reading inputs. It is defined as a quantum counterpart of a classical finite automaton. In QFA, quantum actions are performed by reading the symbols from the input tape.

The concept of quantum automata was first proposed by Moore and Crutchfield [2] and Kondacs and Watrous [3], independently. In 1997, Kondacs and Watrous [3] proposed a variant of quantum automata: measure-many one-way quantum finite automata (MM-1QFA) and two-way quantum finite automata (2QFA), which is a quantum variant of two-way finite automata. In the 2QFA model, the tape head can remain stationary or move either in the left or right direction. The 2QFA is more dominant than the classical model. In 2000, Moore and Crutchfield [2] proposed another variant of the quantum model: measure-once one-way quantum finite automata (MO-1QFA).

Nowadays, a major concern of bioinformatics is the analysis and representation of deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). The grammatical formalism of biological sequences, such as DNA, RNA, and proteins, can be used to solve bioinformatics problems efficiently, such as multiple alignment calculation, classification, and prediction of biological sequences with their primary and secondary structures and their functions. The RNA structure folds around itself to do base pairing, which leads to the formation of various secondary structure motifs (loops). In RNA secondary structure formation, some of the base pairs contain exact complement base pairs, whereas other base pairs will not perform pairing because of the lack of exact complement pairs, thereby tending to form loops. Loops can be of various types, such as the hairpin loop, bulge loop, internal loop, double helix, and pseudoknot, depending on their shape. Moreover, RNA and DNA are made by monomers known as nucleotides. Each nucleotide consists of a pentose carbon sugar, a phosphate group, and a nitrogenous base. If the sugar is ribose, then the polymer is RNA. If the sugar is deoxyribose, then the polymer is DNA. In addition, RNA is single-stranded structure, whereas DNA is double-stranded helical structure.

* Corresponding author.
E-mail addresses: aman8oct1988@gmail.com (A.S. Bhatia), ajaykumar@thapar.edu (A. Kumar).

https://doi.org/10.1016/j.chaos.2018.09.035
0960-0779/© 2018 Elsevier Ltd. All rights reserved.

RNA structure is classified into primary, secondary and tertiary structures. RNA is single-stranded structure i.e. the sequence of the bases (a, c, g, u) are in linear order in its primary structure form [4], where purines are classified into adenine (a) and guanine (g), while pyrimidine are classified into cytosine (c) and uracil (u). RNA structure folds around itself to do base pairing which leads to the formation of various secondary structure motifs (loops). In secondary structure formation, all the bases do not contain exact complementary base pairs, some of the bases lack correct complement base pair, which leads to the formation of loops such as hairpin loops, internal loops, bulge loops and multi-branch loops. At least three bases are required to form loop [5]. Watson-Crick pairing mostly occurs whereas Wobbleâpair rarely occurs in RNA secondary structure. The tertiary structure involves orientation of these secondary structure motifs with respect to each other. It forms certain structures such as pseudoknot which have number of functional features. RNA tends to form single-stranded 3D structures due to presence of extra2′-hydroxyl group present in ribose part of RNA .

The field of quantum computation and information processing has subsequently made a significant impact on the academic and research community alike. Recently, various applications of QFA models to interactive proof system [6,7], debate system [8]; temporal logical theory [9]; automata in molecular biology by quantum mechanics [10] has been investigated. There are various surveys papers on QFAs. Recently, [11] survey is comprehensive enough and provides a general framework for all possible models. Moreover, it has classified and survey results systematically by considering almost all possible research questions that have been examined until now e.g. simulation between classical and quantum models, state complexity, different language recognition models, decidability and undecidability results, interactive proofs, using advise, promise problems, etc.

Furthermore, various generalizations of quantum computational models such as quantum computation over infinite words [12]; QFA models on promise problems [13], multihead one-way quantum finite automata (1QFA(k)) [14]; Two-tape QFA models [15] has been introduced. Recently, Pan et al. [16] quantified the multiqubit states in Grover's searching algorithm by using geometric measure of entanglement (GME). Li et al. [17] studied the phase estimation by using distributed semi-computing model. They have proposed a semi-distributed algorithm for phase estimation which has achieved exponential acceleration and performs better than the classical algorithm.

In classical automata theory, Freivalds [18] shown that a non-regular language $L = \{a^m b^m \mid m \geq 1\}$ can be recognized by 2PFA with arbitrary small error. Later, Dwork and Stockmeyer [19] proved that 2PFA takes an exponential time to recognize the same language. It has been shown that non-regular language can be recognized by 2PFA with error probability below $\frac{1}{2}$, then there exists a constant $b > 0$, such that for infinite number of input n, the expected runtime of 2PFA must exceed 2^{n^b}, where n is the length of input. It is known that 2DFA can recognize only regular languages [20,21].

The research has consistently grown in the field of quantum finite automata theory. Kondacs and Watrous [3] shown that a non-regular language $L = \{a^m b^m \mid m \geq 1\}$ can be recognized by 2QFA with one-sided error in linear time. In this paper, we have shown that it is possible to define non-context free languages by using 2QFA. This paper is concerned with representing the secondary structure loops of RNA using 2QFA, as the secondary structure loops of RNA contain nested dependencies. But, it cannot be represented by 2DFA, and two-way probabilistic finite automata can parse these sequences in exponential time. Thus, it follows that 2QFA is more powerful than its classical variants in terms of language recognition.

1.1. Prior work

Various representations of DNA and RNA sequences using formal grammar and automata have been found in literature. For example, Kalra and Kumar [22] represented DNA and RNA biological sequences, such as inverted repeat, pseudoknot and tandem repeat using state grammar and deep pushdown automata. Sung [23] represented the secondary structure loops of RNA, such as the hairpin loop, internal loop, bulge loop, and double helix, using context-sensitive grammar. Various forms of context-free grammar are also used to represent RNA sequences [24,25].

In addition, cross-interaction grammar was used by Rivas and Eddy [26] to represent the secondary structure loops of RNA, including pseudoknots. Searls [27] used indexed grammar to represent DNA and RNA sequences, such as tandem repeat, inverted repeat, and pseudoknot. Searls [28] also represented DNA sequences using string variable grammar. Mizoguchi et al. [29] used stochastic multiple context-free grammar to represent various classes of pseudoknots. Parallel communicating grammar systems were used by Cai et al. [30] to represent the pseudoknot structure of RNA.

Various researchers have represented the structures of RNA and DNA using concept of grammar and automata theory. Kuppusamy et al. [31] introduced the concept of matrix-insertion deletion grammar and represented the commonly found structures of DNA and RNA which occur at intramolecular level such as pseudoknot, hairpin, stem and loop, cloverleaf, dumbbell and attenuator. Kuppusamy et al. [32] also represented the DNA and RNA biomolecules structures which occur at intermolecular level such as nick language, double strand language and linear hybridization (ligated) languages using matrix insertion-deletion grammar. Further, Kuppusamy and Mahendran [33] extended their own work of bio-molecular representation using matrix-insertion deletion grammar and represented the RNA and DNA bio-molecular structures of the intermolecular level, intramolecular level and RNA secondary level using matrix-insertion deletion grammar. Mahendran and Kuppusamy [34] also modeled commonly occurring of RNA pairing process using matrix insertion-deletion grammar. Various commonly occurring structures are represented as double bulge loop, extended internal loop and triple stem and loop. Fernau et al. [35] proposed universal matrix insertion grammars of small size as an extension of matrix insertion-deletion grammar to represent all DNA computations.

Anderson et al. [36,37] represented the RNA structure using stochastic context-free grammar. Rothemund [38] and Cavaliere et al. [39] represented the action of a restriction enzyme in DNA using a Turing machine and pushdown automata, respectively. Furthermore, Krasinski et al. [40] extended the Cavaliere et al. [39] approach and represented the action of a restriction enzyme in DNA using circular pushdown automata in which stack and input tape are on the same circular strand. Recently, Khrennikov and Yurova [10] have proposed a model of protein behavior by using theory of automata and also explored the similarities between the modeling of behavior of proteins and quantum systems.

1.2. Contributions

After introducing some preliminary concepts in Section 2, the following contributions are claimed:

- Two-way Quantum Finite Automata of Hairpin Loop (Section 3).
- Two-way Quantum Finite Automata of Internal Loop (Section 3).
- Two-way Quantum Finite Automata of Double Helix Loop (Section 3).

We will conclude in Section 4.

2. Preliminaries

In a classical computer, a bit is a smallest basic building block for storing information. Similarly, in a quantum computer, a qubit is a quantum analogue of a classical bit. A qubit allows superposition of both states at a time. It is a state vector having two basis states labeled $|0\rangle$ and $|1\rangle$. Therefore, any two or more distinct quantum states can be added together i.e. superposed, and its outcome will be some other quantum state. In general, consider a quantum state: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. The complex amplitude describes the behavior of the system at a given point of time in space. And, its actual probability is given by the product with its own complex conjugate. The state $|0\rangle$ occurs with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$. One qubit represents two complex amplitudes (α and β), similarly n qubits represent 2^n complex amplitudes. Following are the basic concepts of quantum mechanics and linear algebra used in quantum automata theory:

- Quantum state [41]: A quantum state $|\phi\rangle$ is a superposition of classical states,
$$|\phi\rangle = \alpha_1|x_1\rangle + \alpha_2|x_2\rangle + \cdots + \alpha_n|x_n\rangle \tag{1}$$
where $|x_i\rangle$'s are classical states for $1 \le i \le n$, α_i's are complex numbers called amplitudes and $|\alpha_1|^2 + |\alpha_2|^2 + \cdots + |\alpha_n|^2 = 1$, where $|\alpha_i|^2$ is the squared norm of the corresponding amplitude. Quantum state can also be seen as n-dimensional column vector.
$$\begin{bmatrix}\alpha_1\\ \alpha_2\\ \cdots\\ \alpha_n\end{bmatrix} \tag{2}$$
- Linear vector space [42]: It is defined as a set of elements, called vectors. It is closed under addition and multiplication by scalars. If two vector $|\psi\rangle$ and $|\varphi\rangle$ are a part of vector space, then $|\psi\rangle + |\varphi\rangle$ belongs to vector space. There is also an operation of multiplication by scalars such that if $|\psi\rangle$ is in vector space, then $\alpha|\psi\rangle$ is in the space, where α is a complex scalar.
- Bra-ket notation [9]: In quantum mechanics, the bra-ket notation is used for describing quantum states residing in a complex separable Hilbert space. It is composed of angle brackets and vertical bars to signify the operation of a linear functional on a vector or the scalar product of vectors in a complex vector space.
$$|u\rangle = \begin{bmatrix}a_1\\ a_2\\ a_3\end{bmatrix}, \langle v| = \begin{bmatrix}a_1^* & a_2^* & a_3^*\end{bmatrix} \tag{3}$$
$$|u\rangle\langle v| = \begin{bmatrix}a_1a_1^* & a_1a_2^* & a_1a_3^*\\ a_2a_1^* & a_2a_2^* & a_2a_3^*\\ a_3a_1^* & a_3a_2^* & a_3a_3^*\end{bmatrix} \tag{4}$$
a_i^* denotes the complex conjugate of the complex number a_i. The ket $|u\rangle$ is a column vector, and its conjugate transpose bra $\langle v|$ is a row vector. It is also known as Dirac notation.
- Orthogonal and orthonormal vectors [9]: Two vectors $|u\rangle$ and $|v\rangle$ are orthogonal, if they are perpendicular to each other i.e. the inner product of vectors $\langle u|v\rangle = 0$. It can be defined as a set of vectors $V = \{v_1, v_2, \ldots v_n\}$ are mutually orthogonal if every pair of vectors are orthogonal, i.e. $\langle v_i|v_j\rangle = 0$, for all $i \neq j$. A set of vectors V is orthonormal if every vector in V is a unit vector and the set of vectors are mutually orthogonal.
- Unitary evolution: In quantum systems, Markov matrices are replaced by matrices with complex number entries for the time evaluation of probabilistic systems, by maintaining the condition $\sum_{i=1}^{n}|\alpha_i|^2$. Therefore, consider a quantum system state at time t_0: $|\phi(t_0)\rangle = \alpha_1|x_1\rangle + \alpha_2|x_2\rangle + \cdots + \alpha_n|x_n\rangle$ change into the state at time t: $|\phi'(t)\rangle = \alpha'_1|x_1\rangle + \alpha'_2|x_2\rangle + \cdots + \alpha'_n|x_n\rangle$ where complex amplitudes $\alpha_1, \alpha_2, \ldots, \alpha_n$ and $\alpha'_1, \alpha'_2, \ldots, \alpha'_n$ are related by $|\phi'(t)\rangle = U(t-t_0)|\phi(t_0)\rangle$, where $U(t-t_0)$ is a time dependent unitary operator such that $(U(t-t_0))^*U(t-t_0) = 1$, α_{ij}'s are its entries for $1 \le i, j \le n$.
$$\begin{bmatrix}\alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n}\\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n}\\ \cdots & \cdots & \cdots & \\ \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nn}\end{bmatrix}\begin{bmatrix}\alpha_1\\ \alpha_2\\ \cdots\\ \alpha_n\end{bmatrix} = \begin{bmatrix}\alpha'_1\\ \alpha'_2\\ \cdots\\ \alpha'_n\end{bmatrix} \tag{5}$$
and $\sum_{i=1}^{n}|\alpha_i|^2 = \sum_{i=1}^{n}|\alpha'_i|^2 = 1$. Therefore, evaluation of a quantum system at any time must be unitary [9].

- Grammar [43]: It is a quadruple (V_N, V_T, P, S), where V_N is a set of non-terminals, V_T is a set of terminals, P is a set of production rules of the form $\{\alpha \to \beta | \alpha, \beta \in (V_N \cup V_T)^*\}$, where * is the Kleene star operator, S is a start symbol, and $S \in V_N$. Each production rule maps an input string of symbols to another (i.e. an empty string) and the first string consist at least one non-terminal in number of symbols.
$L(G)$ denotes the language generated by grammar G. Let $\Sigma = \{g, c, a, u\}$ be an alphabet of RNA. Pairing occurs between purines and pyrimidine. The complement of symbol a is represented by a'. Complement pairing commonly occurs in RNA is Watson–Crick pairing. In Watson–Crick base pairs, $a' = u, u' = a, g' = c$ and $c' = g$.
- Quantum grammar [2]: It is defined as quadruple (V, T, I, P), where V is finite set of variables, T is finite non empty set of terminals, I is an initial variable ($I \in V$), P is a finite set of productions such that $\alpha \to \beta$, where $\alpha \in V^*$ and $\beta \in (V \cup T)^*$. Each production in P has a set of complex amplitudes such that $c_k(\alpha \to \beta)$ for $1 \le k \le n$, where n is the dimension of the grammar.
Therefore, quantum grammar combines a complex amplitude with each production rule. In order to find the probability of quantum system for a transition from its initial state to final state, we can calculate the probability amplitude on applying the production rules for each path from initial state to final state of the quantum system. Basically, a derivation $\alpha \to \beta$ is a series of strings, where a substring is replaced with another on applying one of the productions at each step. The kth amplitude c_k of a derivation is defined as the product of the c_k's for each productions in the chain and $c_k(\alpha \Rightarrow \beta)$ as the sum of the derivations of β from α. The amplitudes of a word $w \in T^*$ are $c_k(w) = c_k(I \Rightarrow w)$ and the probability associated with w is the norm of its vector of amplitudes, $f(w) = \sum_{k=1}^{n}|c_k(w)|^2$.
- Quantum finite automata [3] is defined as real-time quantum automata (Σ, s_{init}, H_{accept}, P_{acc}, U_σ), where Σ is an input alphabet, Hilbert space H, an initial state vector $s_{init} \in H$ with $|s_{init}|^2 = 1$, A subspace $H_{accept} \subset H$ and an operator P_{acc} which project on it, a unitary transition matrix U_σ for each $\sigma \in \Sigma$.
The quantum language recognized by quantum finite automata as a function $f_{QFA}(w) = |s_{init}U_wP_{acc}|^2$, where $U_w = U_{w_1}U_{w_2}\cdots U_{w_{|w|}}$ and $w = w_1, w_2, \ldots,$ is an input string belong to any quantum language. The process of computation of input string w starts with the initial vector, apply the unitary matrix of each symbol in order and measure the probability by applying projection operator such that the resultant state is in subspace H_{accept}.
- Two-way quantum finite automaton [3]: A 2QFA is defined as a sextuple (Q, Σ, δ, q_0, Q_{acc}, Q_{rej}), where Q is a set of states. Moreover, $Q = Q_{acc} \cup Q_{rej} \cup Q_{non}$, where Q_{acc}, Q_{rej}, Q_{non} represent the set of accepting, rejecting and non-halting states respectively.

Σ is an input alphabet, Transition function δ is defined by δ: $Q \times \Gamma \times Q \times D \rightarrow C$, where $\Gamma = \Sigma \cup \{\#, \$\}$, where #,\$ represent the left-end and right-end markers respectively, $D = \{\leftarrow, \uparrow, \rightarrow\}$ represent the left, stationary and right direction of tape head, $\times$ signify the Cartesian product between two sets and $\rightarrow$ is a mapping. Transition function must satisfy the following conditions:

1. Local probability and orthogonality condition:

$$\sum_{(q',d)\in Q\times D}^{\forall(q_1,\sigma_1),(q_2,\sigma_2)\in Q\times\Gamma} \overline{\delta(q_1,\sigma,q',d)}\delta(q_2,\sigma,q',d) = \begin{Bmatrix} 1 & q_1 = q_2 \\ 0 & q_1 \neq q_2 \end{Bmatrix} \tag{6}$$

2. First separability condition:

$$\sum_{q'\times Q}^{\forall(q_1,\sigma_1),(q_2,\sigma_2)\in Q\times\Gamma} \overline{\delta(q_1,\sigma_1,q',\rightarrow)}\delta(q_2,\sigma_2,q',\uparrow) + \overline{\delta(q_1,\sigma_1,q',\uparrow)}\delta(q_2,\sigma_2,q',\leftarrow) = 0 \tag{7}$$

3. Second separability condition:

$$\sum_{q'\times Q}^{\forall(q_1,\sigma_1),(q_2,\sigma_2)\in Q\times\Gamma} \overline{\delta(q_1,\sigma_1,q',\rightarrow)}\delta(q_2,\sigma_2,q',\leftarrow) = 0 \tag{8}$$

A 2QFA is simplified, for each $\sigma \in \Gamma$, if there exists a unitary linear operator V_σ on the inner product space such that $L_2\{Q\} \rightarrow L_2\{Q\}$, where Q is the set of states and a function $D: Q \rightarrow \{\leftarrow, \uparrow, \rightarrow\}$. Define transition function as

$$\delta(q,\sigma,q',d) = \left\{\begin{array}{l|l} \langle q'|V_\sigma|q\rangle & \text{if } D(q') = d \\ 0 & \text{else} \end{array}\right\} \tag{9}$$

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

In order to process the input string by 2QFA, we assume that input string x is written on input with both end-markers such that #x \$. The automata is in any state q and head is above the symbol σ. Then, with the amplitude $\delta(q, \sigma, q', d)$ moves to state $q', d \in \{\leftarrow, \uparrow, \rightarrow\}$, moves the head one cell towards left, stationary and in right direction. The automata for processing an input x corresponds a unitary evolution in the inner-product space H_n.

A computation of a 2QFA is a sequence of superpositions $c_0, c_1, c_2, \ldots,$ where c_0 is an initial configuration. When the automata are observed in a superposition state, for any c_i, it has the form $U_\delta|c_i\rangle \sum_{c\in C_n} \alpha_c|c_i\rangle$ where defines the set of configurations, and the configuration c_i is associated with amplitude α_c. Superposition is valid; if the sum of the absolute squares of their probability amplitudes is unitary. The probability for a specified configuration is given by the absolute squares of amplitude associated with that configuration. Time evolution of quantum systems is given by unitary transformations. Each transition function δ induces a linear time evolution operator over the space H_n.

$$U_\delta^x|q,k\rangle = \sum_{(q',d)\in Q\times D} \delta(q, x(k), q', d)|q', k + d mod|x|\rangle$$

for each $(q, k) \in C_{|x|}$, where $q \in Q$, $k \in Z_{|x|}$ and extended to H_n by linearity [3].

3. Modeling of RNA secondary structure loops using 2-way quantum finite automata

In this section, we model the RNA secondary loops such as hairpin loop, internal loop and double helix using two-way quantum finite automata (2QFA).

3.1. Hairpin loop

A hairpin loop is formed when RNA stands folds to pair with another section of the same strand and ends to form unpaired loop [44].

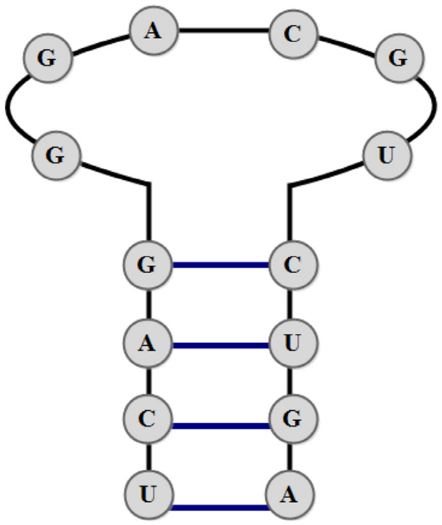

Fig. 1. Representation of Hairpin loop structure.

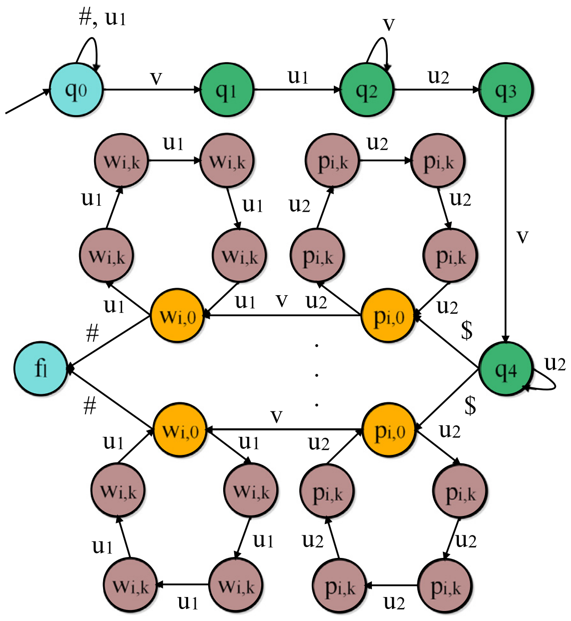

Fig. 2. State diagram of Hairpin loop for $L_1 = u_1^p v^m u_2^p$.

Fig. 1 describes the structure of hairpin loop. The structure of hairpin loop is as follows i.e. sequence 1 followed by hairpin loop followed by paired sequence 1 [23].

If we denote the sequence 1 by a, and hairpin loop by b and paired sequence by c, the language of above hairpin loop is $L_1 = u_1^p v^m u_2^p \mid p > 1, m \geq 3$. Here, u_1, v, $u_2 \in \{g, c, a, u\}$.

The quantum finite automata for language $L_1 = \{u_1^p v^m u_2^p \mid p > 1, m \geq 3\}$ is as follows:

Theorem 3.1. *For a language $L_1 = \{u_1^p v^m u_2^p \mid p > 1, m \geq 3\}$ and for an arbitrary fixed positive integer $n \geq 2$ such that $n \in \mathbb{N}$, there exists a 2QFA such that for $x \in L_1$, it accepts x with probability 1 and rejects $x \notin L_1$ with probability at least $1 - \frac{1}{n}$.*

Proof. We construct 2QFA model for the language:

$$L_1 = \{u_1^p v^m u_2^p \mid p > 1, m \geq 3\}$$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\} \cup \{p_{i,j}, w_{i,j} \mid 1 \leq i \leq n, 1 \leq j \leq \max(i, n - i + 1)\} \cup \{p_{i,0}, w_{i,0}, r_{i,0}, s_{i,0}, t_{i,0}, K_{i,0}, Y_{i,0} \mid 1 \leq i \leq n\} \cup \{f_i, R_i \mid 1 \leq i \leq n\}$, $\Sigma \in \{u_1, v, u_2\}$, $Q_{acc} = \{F_n\}$, $Q_{rej} = \{q_5\} \cup \{f_i \mid 1 \leq i < n\} \cup \{R_i \mid$

Table 1
Details of transition function and tape head function.

$V_{\#}\|q_0\rangle = \|q_0\rangle$	$V_v\|q_0\rangle = \|q_1\rangle$	$V_{u_1}\|q_1\rangle = \|q_2\rangle$
$V_{\#}\|q_1\rangle = \|q_5\rangle$	$V_v\|q_2\rangle = \|q_2\rangle$	$V_{u_1}\|q_2\rangle = \|q_5\rangle$
$V_{\$}\|q_2\rangle = \|q_5\rangle$	$V_v\|q_3\rangle = \|q_4\rangle$	$V_{u_1}\|q_4\rangle = \|q_5\rangle$
$V_{u_1}\|q_0\rangle = \|q_0\rangle$	$V_v\|q_4\rangle = \|q_5\rangle$	$V_{u_2}\|q_0\rangle = \|q_5\rangle$
		$V_{u_2}\|q_4\rangle = \|q_4\rangle$

$$V_{\$}|q_4\rangle = \frac{1}{\sqrt{n}} \textstyle\sum_{i=1}^{n} |p_{i,0}\rangle$$
$$V_{u_2}|p_{i,0}\rangle = |p_{i,n-i+1}\rangle \text{ for } 1 \leq i \leq n$$
$$V_{u_2}|p_{i,j}\rangle = |p_{i,j-1}\rangle \text{ for } 1 \leq i \leq n, 1 \leq j \leq n-i+1$$
$$V_v|p_{i,0}\rangle = |s_{i,0}\rangle, V_b|s_{i,0}\rangle = |r_{i,0}\rangle \qquad \text{If } v \neq 3$$
$$V_v|r_{i,0}\rangle = |t_{i,0}\rangle \text{ for } 1 \leq i \leq N \qquad V_{u_1}|s_{i,0}\rangle = |K_{i,0}\rangle$$
$$V_{u_1}|w_{i,0}\rangle = |w_{i,i}\rangle \text{ for } 1 \leq i \leq n \qquad V_{u_1}|K_{i,0}\rangle = |Y_{i,i}\rangle$$
$$V_{u_1}|w_{i,j}\rangle = |w_{i,j-1}\rangle \text{ for } 1 \leq i \leq n, 1 \leq j \leq n-i+1$$
$$V_{u_1}|Y_{i,j}\rangle = |Y_{i,j-1}\rangle \text{ for } 1 \leq i \leq n, 1 \leq j \leq n-i+1$$
$$V_{\#}|w_{i,0}\rangle = \frac{1}{\sqrt{n}} \sum_{l=1}^{n} e^{\frac{2\pi\iota}{n}il} |f_l\rangle \text{ for } 1 \leq i \leq n$$
$$V_{\#}|Y_{i,0}\rangle = \frac{1}{\sqrt{n}} \sum_{l=1}^{n} e^{\frac{2\pi\iota}{n}il} |R_l\rangle \text{ for } 1 \leq i \leq n$$

Head functions

$D(q_0) = \rightarrow$, $D(q_1) = \leftarrow$, $D(q_2) = \rightarrow$,
$D(q_3) = \leftarrow$, $D(q_4) = \rightarrow$, $D(q_5) = \uparrow$

$1 \leq i \leq n\}$. The state transition functions and head functions are given in Table 1 using equation (10). Each V_σ take the values as shown in Table 1 and also for each σ, the vectors formed V_σ are unitary on Hilbert space $L_2(Q)$ and D: $Q \rightarrow \{\leftarrow, \uparrow, \rightarrow\}$ represents head functions.

The process of designing 2QFA for language consists of three phases. In the first phase, if the input is not of the form $u_1^+ v^+ u_2^+$, where $^+$ represents the Kleene plus, then the computation will be rejected. Therefore, the first phase traverses the input string (i.e. scanning through the each symbol of input string). At the start of the second phase; the state q_4 reads the right-end marker \$. Furthermore, the computation is split in to the n different paths (state transformations occur in parallel). Each path possesses equal amplitude $\frac{1}{\sqrt{n}}$. Along the n different paths, each path moves deterministically to the left-end marker #. On reading symbol u_2, along the ith path, tape W head remain stationary for $n - i + 1$ times. On reading symbol u_1, head remain stationary for i times. On reading symbol v, it must satisfy the condition ($m \geq 3$); otherwise it leads to reject the input string at the end. If number of u_1's and u_2's are equal in input string, then different paths will reach the left-end marker # at the same time. Finally, in third phase on reading #, each computation path comes into n-way Quantum Fourier transform (QFT). It produces either one accepting state or rejecting state. Finally, 2QFA accepts the string $u_1^2 v^3 u_2^2$ with probability 1. If the input string is not in the desired form, then all computation paths reaches the left-end marker at different time and there is no cancellation of rejecting states. Thus, for all n computation paths, the conditional probability that an observation results in acceptance is $\frac{1}{n}$, such that some of the paths comes to halt. Therefore, the input string is said to be accepted with probability $\frac{1}{n}$. Correspondingly, the input string is said to be rejected with probability $1 - \frac{1}{n}$. □

3.2. Internal loop

Internal loops are formed when there is no match pair on the both sides of the double-stranded RNA i.e. formation of hairpin loop on both sides of double-stranded RNA [45].

Fig. 3 describes the structure of internal loop. The structure of internal loop is as follows i.e. Sequence 1 followed by hairpin loop followed by sequence 2, sequence, paired sequence 2, hairpin loop and paired sequence 1 [23].

Table 2
Details of transition function and tape head function.

$V_{\#}\|q_0\rangle = \|q_0\rangle$	$V_{u_3}\|q_2\rangle = \|q_3\rangle$	$V_{w_2}\|q_9\rangle = \|q_{10}\rangle$
$V_{\$}\|q_0\rangle = \|q_{14}\rangle$	$V_{u_3}\|q_4\rangle = \|q_4\rangle$	$V_{w_2}\|q_{11}\rangle = \|q_{11}\rangle$
$V_{\$}\|q_{11}\rangle = \|q_{14}\rangle$	$V_{u_3}\|q_5\rangle = \|q_6\rangle$	$V_{\$}\|q_{12}\rangle = \|q_{13}\rangle$
$V_{u_1}\|q_0\rangle = \|q_0\rangle$	$V_{u_3}\|q_8\rangle = \|q_9\rangle$	$V_{w_2}\|q_0\rangle = \|q_{14}\rangle$
$V_{u_1}\|q_1\rangle = \|q_2\rangle$	$V_{u_3}\|q_0\rangle = \|q_{14}\rangle$	$V_{w_2}\|q_2\rangle = \|q_{14}\rangle$
$V_{u_2}\|q_0\rangle = \|q_1\rangle$	$V_{w_1}\|q_2\rangle = \|q_{14}\rangle$	$V_{w_2}\|q_4\rangle = \|q_{14}\rangle$
$V_{u_2}\|q_1\rangle = \|q_2\rangle$	$V_{w_1}\|q_4\rangle = \|q_8\rangle$	$V_{w_3}\|q_{11}\rangle = \|q_{12}\rangle$
$V_{u_2}\|q_3\rangle = \|q_4\rangle$	$V_{w_1}\|q_6\rangle = \|q_7\rangle$	$V_{w_3}\|q_{13}\rangle = \|q_{13}\rangle$
$V_{u_2}\|q_0\rangle = \|q_1\rangle$	$V_{w_1}\|q_9\rangle = \|q_9\rangle$	$V_{w_3}\|q_0\rangle = \|q_{14}\rangle$
$V_v\|q_4\rangle = \|q_5\rangle$	$V_{w_1}\|q_{10}\rangle = \|q_{11}\rangle$	$V_{w_3}\|q_2\rangle = \|q_{14}\rangle$
$V_v\|q_6\rangle = \|q_6\rangle$	$V_{w_1}\|q_0\rangle = \|q_{14}\rangle$	$V_{w_3}\|q_4\rangle = \|q_{14}\rangle$
$V_v\|q_2\rangle = \|q_{14}\rangle$	$V_{u_2}\|q_2\rangle = \|q_2\rangle$	$V_{w_3}\|q_9\rangle = \|q_{11}\rangle$

Head functions

$D(q_0) = \rightarrow$, $D(q_1) = \rightarrow$, $D(q_2) = \rightarrow$, $D(q_3) = \leftarrow$
$D(q_4) = \rightarrow$, $D(q_5) = \leftarrow$, $D(q_6) = \rightarrow$, $D(q_7) = \leftarrow$
$D(q_8) = \leftarrow$, $D(q_9) = \rightarrow$, $D(q_{10}) = \leftarrow$, $D(q_{11}) = \rightarrow$
$D(q_{12}) = \leftarrow$, $D(q_{13}) = \rightarrow$, $D(q_{14}) = \uparrow$

The language generated by internal loop is

$$L_2 = \{u_1^r u_2^m u_3^q v^z w_1^q w_2^m w_3^r \mid r, q > 1, m \geq 3, z \leq 2\}$$

Here, $(u_1, u_2, u_3, v, w_1, w_2, w_3) \in \{g, c, a, u\}$ The quantum finite automata for language L_2 is as follows:

Theorem 3.2. *For a language $L_2 = \{u_1^r u_2^m u_3^q v^z w_1^q \ w_2^m w_3^r \mid r, q > 1, m \geq 3, z \leq 2$ and $r, q, m \in \mathbb{N}_{>0}, z \in N_{\geq 0}\}$ and for arbitrary computational paths n, M, P $\in \mathbb{N}$, there exists a 2QFA such that for $x \in L_2$, it accepts x with bounded error ϵ and rejects $x \notin L_2$ with probability at least $1 - \frac{1}{nMP}$.*

Proof. We construct 2QFA model for $L_2 = \{u_1^r u_2^m u_3^q \ v^z w_1^q w_2^m w_3^r \mid r, q > 1, m \geq 3, z \leq 2\}$.

Let $M_{2QFA} = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$ be a 2QFA $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}\} \cup \{p_{i,0}, r_{i,0}, u_{i,0}, s_{i,0}, t_{s,0}, v_{s,0} \mid 1 \leq i \leq n, 1 \leq s \leq n\} \cup \{q_{ij,0}, r_{ij,0}, g_{ij,0}, u_{ij,0}, v_{ij,0} \mid 1 \leq i \leq N, 1 \leq j \leq M\} \cup \{q_{ijl,0}, v_{ijl,0}, g_{ijl,0} \mid 1 \leq l \leq n, 1 \leq j \leq M, 1 \leq l \leq P\} \cup \{s_{i,k}, p_{i,k}, v_{ij,k}, g_{ijl,k} \mid 1 \leq i \leq n, 1 \leq j \leq M, 1 \leq l \leq \max(j, M-j+1)\}, \Sigma \in \{u_1, u_2, u_3, v, w_1, w_2, w_3\}, Q_{acc} = \{y_n\}$ and $Q_{rej} = \{q_{14}\}$. The state transition functions and head functions are given in Table 2 using equation (10). All vectors formed V_σ are unitary on Hilbert space $L_2(Q)$ and D represents head functions.

The process of designing a 2QFA for language L_2 consists of seven phases namely traverse the input (i.e. scanning through the each symbol of input string), splition (state transformations occur in parallel) of computation in to n different paths for having superposition of states on reading the right-marker \$ (i.e. quantum system is in more than one state at a time), then n different paths splits in to M different paths while reading leftmost w_3, further again M different paths splits into P different paths with equal probability on reading leftmost w_2, perform P-way QFT at the end of u_3, then perform M-way QFT at the end of u_2 and similarly perform n-way QFT to yield the result at the end.

In the first phase, if the input string is not of the form $u_1^+ u_2^+ u_3^+ v^+ w_1^+ w_2^+ w_3^+$ or $u_1^+ u_2^+ u_3^+ w_1^+ w_2^+ w_3^+$, where $^+$ represents the Kleene plus, then computation will be rejected. In Table 2, q_{13} reads the right-end marker \$ and the computation splits in to n different paths with equal probability $\frac{1}{\sqrt{n}}$. Along the all n paths, it checks whether the number of w_3's is greater than 1 or not. Further, we can define remaining transition functions such that V_σ must satisfy unitary property. Fig. 4 shows the detailed state transition diagram of L_2. In the third phase, upon reading the leftmost w_2, n paths splits in to M different paths with equal probability

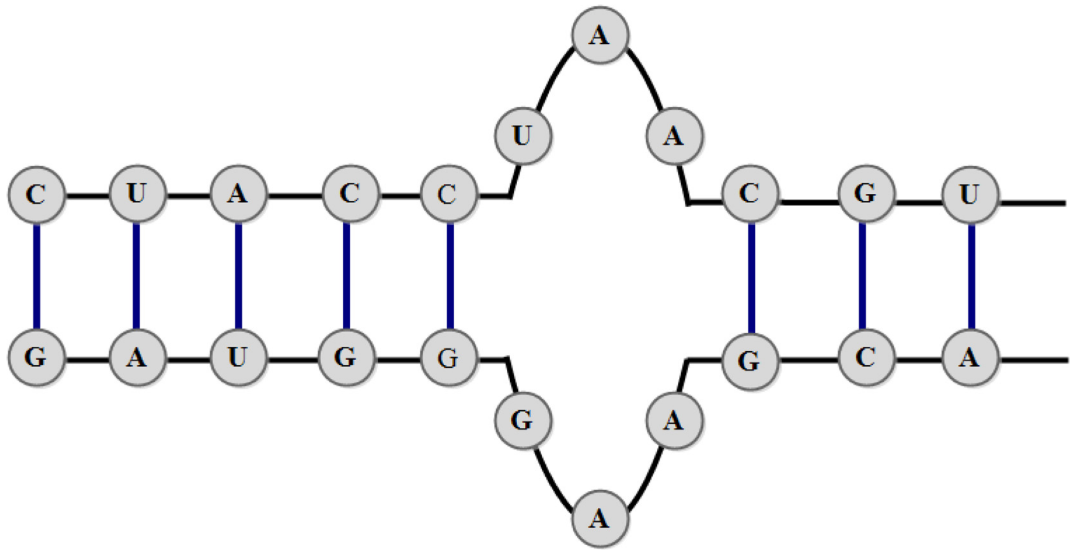

Fig. 3. Illustration of Internal loop structure.

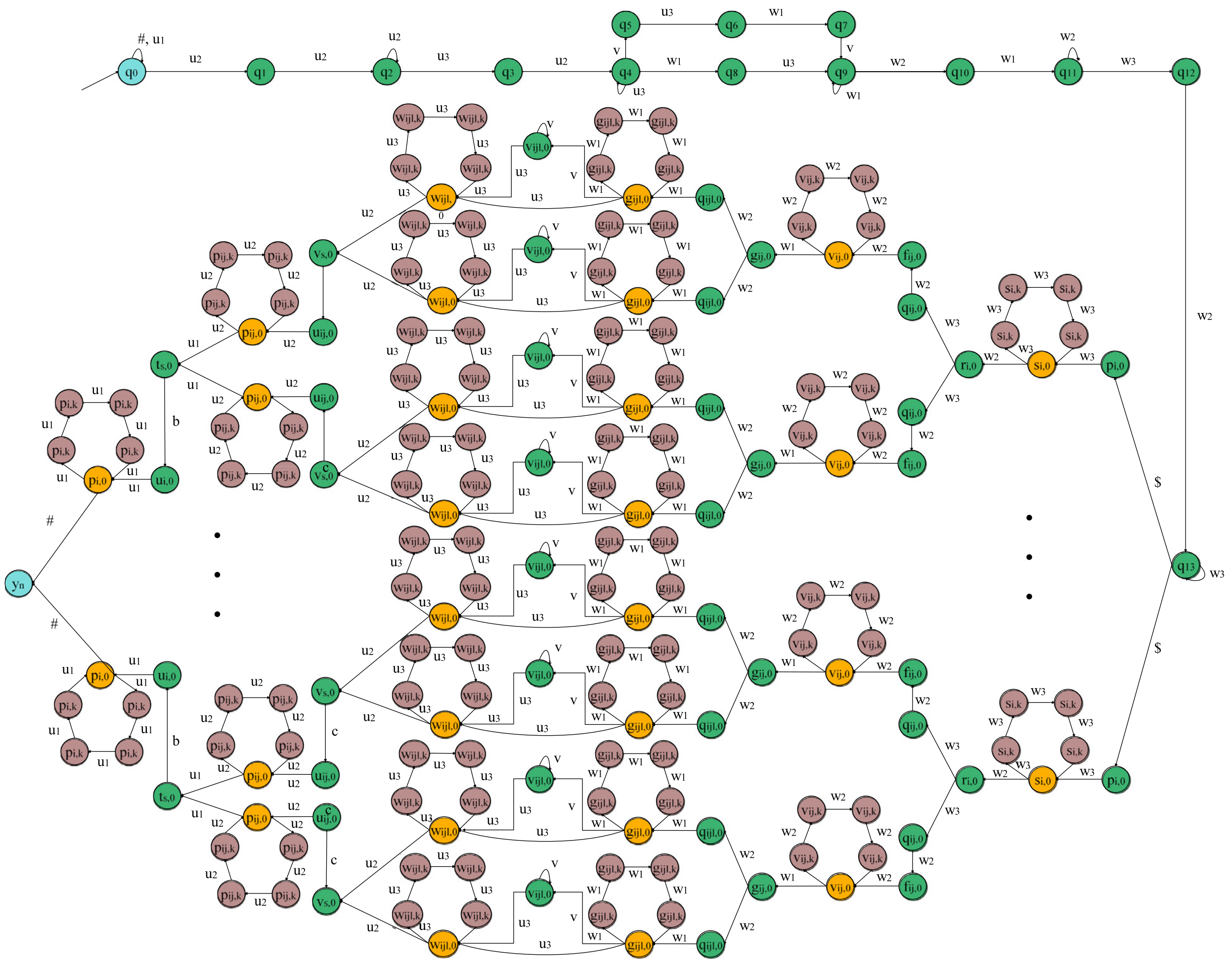

Fig. 4. State transition diagram of Internal loop for $L_2 = u_1^r u_2^m u_3^q v^z w_1^q w_2^m w_3^r$.

$\frac{1}{\sqrt{M}}$ and checks whether the number of w_2's satisfy the condition or not. Furthermore, in fourth phase, M paths splits in to P different paths with equal probability $\frac{1}{\sqrt{P}}$ and checks whether the number of w_1's is greater than 1 or not. In fifth phase, all P different paths reads v, u_3 or both. On reading the v, it must satisfy the condition $(z \leq 2)$ and all paths moves towards left. On reading the u_3's, each computation path enters P-way QFT. In the sixth, each computation path enters M-way QFT upon reading the left-most u_2. In the last phase, on reading the left-marker #, each computation path enters N-way QFT to yield the result. It results in single

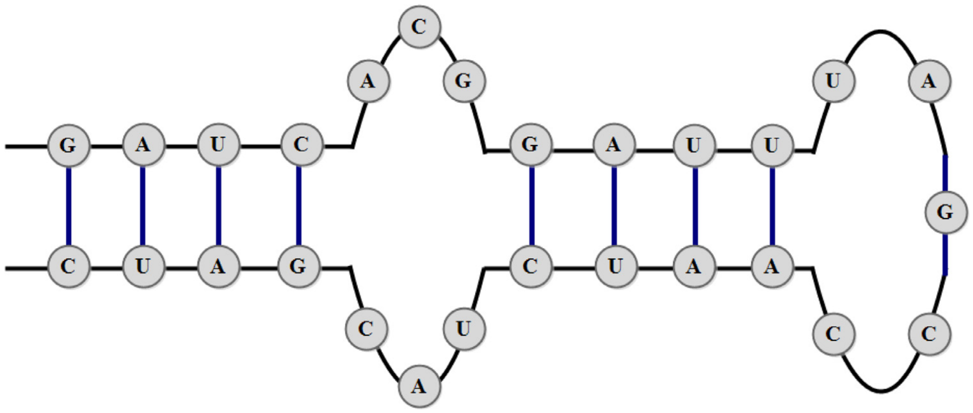

Fig. 5. Illustration of double helix loop structure.

accepting state and other rejecting states. If all nth paths read the left-end marker # at same time, then the input is accepted with probability 1. □

3.3. Double helix

Double helix loops are formed when there is no match pair on the both sides of the double-stranded RNA and also ends in a hairpin loop [23]. If in the language $L_2 = u_1^r u_2^m u_3^q v^z w_1^q w_2^m w_3^r \mid r, q > 1, m \geq 3 \mid z \leq 2$ of internal loop, value of z is $(z \geq 3)$, it leads to the formation of double helix.

Fig. 5 describes the structure of double helix loop. The language generated by grammar is of double helix loop i.e. Sequence 1 followed by loop segment 1 followed by sequence 2, followed by hairpin loop, followed by paired sequence 2 followed by loop segment 2 followed by paired sequence 1 [23]. The language generated by double helix is $L_3 = \{u_1^r u_2^m u_3^q v^z w_1^q \;\; w_2^m w_3^r \mid r, q > 1, m, z \geq 3\}$. Here, $(u_1, u_2, u_3, v, w_1, w_2, w_3) \in \{g, c, a, u\}$. The quantum finite automaton for language L_3 is as follows:

Theorem 3.3. *For a language $L_3 = \{u_1^r u_2^m u_3^q v^z w_1^q \;\; w_2^m w_3^r \mid r, q > 1, m, z \geq 3$ and $r, q, m, z \in \mathbb{N}_{>0}\}$ and for arbitrary computational paths $n, M, P \in \mathbb{N}$, there exists a 2QFA such that for $x \in L_3$, it accepts x with bounded error ϵ and otherwise rejects it $(x \notin L_3)$.*

Proof. It functions similar to above languages instead its process of designing consists of seven phases namely traverse the input, splition of computation in to n different paths for having superposition of states on reading the right-marker \$ and process w_3's, then n paths splits in to M different paths while reading leftmost w_3, further again M different paths splits into P different paths with equal probability on reading leftmost w_2, perform P-way QFT at the end of u_3, then perform M-way QFT at the end of u_2 and similarly perform n-way QFT to yield the result at the end. If all n^{th} paths read the left-end marker # at same time, then the input is accepted with probability 1. Therefore, it can be easily checked that input string $x \in L_3$, can be recognized by 2QFA with probability 1. □

4. Conclusion

RNA secondary structure loops contains nested dependencies, so it cannot be represented by regular grammar or finite automata. Previous attempts to represent these sequences use context-sensitive grammar or stochastic context-free grammar or tree adjoining grammar, which have higher time complexities. In quantum automata theory, a two-way quantum finite automaton is more dominant than its classical counterparts. In this contribution, we focused on RNA secondary structure hairpin loop; internal loop and double helix loop and represented them by using two-way quantum finite automata. The major benefit of this approach is that sequences can be parsed in linear time by one-sided bounded error. To the best of author's knowledge no such representation is done using two-way quantum finite automata so far. In the future, we will try to represent other secondary structure loops such as bulge loop and junctions using other quantum computational models.

Acknowledgement

Amandeep Singh Bhatia was supported by Maulana Azad National Fellowship (MANF), funded by Ministry of Minority Affairs, Government of India.

References

[1] García R. Prediction of rna pseudoknotted secondary structure using stochastic context free grammars (scfg). CLEI Electron J 2006;9(2):1–12.
[2] Moore C, Crutchfield JP. Quantum automata and quantum grammars. Theor Comput Sci 2000;237(1–2):275–306.
[3] Kondacs A, Watrous J. On the power of quantum finite state automata. In: Proceedings of the 38th annual symposium on foundations of computer science. IEEE; 1997. p. 66–75.
[4] Sippl MJ. Biological sequence analysis. probabilistic models of proteins and nucleic acids. Protein Sci 1999;8(3):695.
[5] Chen C, Ridzon DA, Broomer AJ, Zhou Z, Lee DH, Nguyen JT, et al. Real-time quantification of micrornas by stem–loop rt–pcr. Nucleic Acids Res 2005;33(20):e179.
[6] Nishimura H, Yamakami T. An application of quantum finite automata to interactive proof systems. J Comput Syst Sci 2009;75(4):255–69.
[7] Zheng S, Qiu D, Gruska J. Power of the interactive proof systems with verifiers modeled by semi-quantum two-way finite automata. Inf Comput 2015;241:197–214.
[8] Yakaryilmaz A, Say AC, Demirci HG. Debates with small transparent quantum verifiers. Int J Found Comput Sci 2016;27(02):283–300.
[9] Berzina ID. Quantum finite automata and logic. University of Latvia, Riga; 2010.
[10] Khrennikov A, Yurova E. Automaton model of protein: dynamics of conformational and functional states. Prog Biophys Mol Biol 2017:2–14.
[11] Ambainis A., Yakaryilmaz A.. Automata and quantum computing. arXiv:1507.01988 2015.
[12] Giannakis K, Papalitsas C, Andronikos T. Quantum automata for infinite periodic words. In: Proceedings of the 6th international conference on information, intelligence, systems and applications (IISA). IEEE; 2015. p. 1–6.
[13] Zheng S, Li L, Qiu D, Gruska J. Promise problems solved by quantum and classical finite automata. Theor Comput Sci 2017;666:48–64.
[14] Ganguly D, Chatterjee K, Ray KS. 1-Way multihead quantum finite state automata. Appl Math (Irvine) 2016;7(09):1005.
[15] Ganguly D., Ray K.S.. 2-Tape 1-way quantum finite state automata. arXiv:1607.00811 2016.
[16] Pan M, Qiu D, Zheng S. Global multipartite entanglement dynamics in groverâs search algorithm. Quantum Inf Process 2017;16(9):211.
[17] Li K, Qiu D, Li L, Zheng S, Rong Z. Application of distributed semi-quantum computing model in phase estimation. Inf Process Lett 2017;120:23–9.
[18] Freivalds R. Probabilistic two-way machines. In: Proceedings of the international Symposium on Mathematical Foundations of Computer Science. Springer; 1981. p. 33–45.
[19] Dwork C, Stockmeyer L. A time complexity gap for two-way probabilistic finite-state automata. SIAM J Comput 1990;19(6):1011–123.

[20] Shepherdson JC. The reduction of two-way automata to one-way automata. IBM J Res Dev 1959;3(2):198–200.
[21] Rabin MO, Scott D. Finite automata and their decision problems. IBM J Res Dev 1959;3(2):114–25.
[22] Kalra N, Kumar A. State grammar and deep pushdown automata for biological sequences of nucleic acids. Curr Bioinform 2016;11(4):470–9.
[23] Sung K-Y. The use of context-sensitive grammar for modeling RNA pseudoknots.. In: BIOCOMP; 2006. p. 338–44.
[24] Sakakibara Y, Brown M, Hughey R, Mian IS. The application of stochastic context-free grammars to folding, aligning and modeling homologous rna sequences. Technical report. University of California at Santa Cruz, CA, USA; 1993.
[25] Kobayashi S, Yokomori T. Modeling rna secondary structures using tree grammars. Genome Inform 1994;5:29–38.
[26] Rivas E, Eddy SR. The language of rna: a formal grammar that includes pseudoknots. Bioinformatics 2000;16(4):334–40.
[27] Searls DB. The computational linguistics of biological sequences. Artif Intell Mol Biol 1993;2:47–120.
[28] Searls DB. String variable grammar: a logic grammar formalism for the biological language of dna. J Logic Program 1995;24(1–2):73–102.
[29] Mizoguchi N, Kato Y, Seki H. A grammar-based approach to rna pseudoknotted structure prediction for aligned sequences. In: Proceedings of the 1st international conference on computational advances in bio and medical sciences (ICCABS). IEEE; 2011. p. 135–40.
[30] Cai L, Malmberg RL, Wu Y. Stochastic modeling of rna pseudoknotted structures: a grammatical approach. Bioinformatics 2003;19(suppl_1):i66–73.
[31] Kuppusamy L, Mahendran A, Krishna SN. Matrix insertion-deletion systems for bio-molecular structures. In: Proceedings of the international conference on distributed computing and internet technology. Springer; 2011. p. 301–12.
[32] Kuppusamy L, Mahendran A, de La Clergerie ÉV. Modelling intermolecular structures and defining ambiguity in gene sequences using matrix insertion-deletion systems, 228. Biology, Computation and Linguistics, IOS Press; 2011.
[33] Kuppusamy L, Mahendran A. Modelling dna and rna secondary structures using matrix insertion–deletion systems. Int J Appl Math Comput Sci 2016;26(1):245–58.
[34] Mahendran A, Kuppusamy L. Formal language representation and modelling structures underlying rna folding process. In: Proceedings of the international conference on theoretical computer science and discrete mathematics. Springer; 2016. p. 20-29.
[35] Fernau H, Kuppusamy L, Verlan S. Universal matrix insertion grammars with small size. In: Proceedings of the international conference on unconventional computation and natural computation. Springer; 2017. p. 182-93.
[36] Anderson JW, Tataru P, Staines J, Hein J, Lyngsø R. Evolving stochastic context-free grammars for rna secondary structure prediction. BMC Bioinform 2012;13(1):78.
[37] Anderson J. Stochastic context-free grammars and RNA secondary structure prediction. In: Poptsova Maria S, editor. Genome Analysis: Current Procedures and Applications. England: Caister Academic Press; 2014. p. 339–66.
[38] Rothemund PWK. A DNA and restriction enzyme implementation of turing machines.. DNA Based Comput 1995;27:75–119.
[39] Cavaliere M, Jonoska N, Yogev S, Piran R, Keinan E, Seeman NC. Biomolecular implementation of computing devices with unbounded memory. In: Proceedings of the international workshop on DNA-based computers. Springer; 2004. p. 35–49.
[40] Krasinski T, Sakowski S, Poplawski T. Autonomous push-down automaton built on dna. Inform Int J Comput Inform 2011;36(3):263–76.
[41] Peres A. Quantum theory: concepts and methods, 57. Springer Science & Business Media; 2006.
[42] Wang J. Handbook of finite state based models and applications. CRC press; 2012.
[43] Sakakibara Y, Brown M, Hughey R, Mian IS, Sjölander K, Underwood RC, et al. Stochastic context-free grammers for tRNA modeling. Nucleic Acids Res 1994;22(23):5112–20.
[44] Martin JC. Introduction to languages and the theory of computation, 4. McGraw-Hill NY, USA; 1991.
[45] Macke TJ, Ecker DJ, Gutell RR, Gautheret D, Case DA, Sampath R. Rnamotif, an RNA secondary structure definition and search algorithm. Nucleic Acids Res 2001;29(22):4724–35.