RAIRO-Theor. Inf. Appl. 53 (2019) 19–35
https://doi.org/10.1051/ita/2018020

RAIRO - Theoretical Informatics and Applications
www.rairo-ita.org

ON THE POWER OF TWO-WAY MULTIHEAD QUANTUM FINITE AUTOMATA

Amandeep Singh Bhatia* and Ajay Kumar

Abstract. This paper introduces a variant of two-way quantum finite automata named two-way multihead quantum finite automata. A two-way quantum finite automaton is more powerful than classical two-way finite automata. However, the generalizations of two-way quantum finite automata have not been defined so far as compared to one-way quantum finite automata model. We have investigated the newly introduced automata from two aspects: the language recognition capability and its comparison with classical and quantum counterparts. It has been proved that a language which cannot be recognized by any one-way and multi-letter quantum finite automata can be recognized by two-way quantum finite automata. Further, it has been shown that a language which cannot be recognized by two-way quantum finite automata can be recognized by two-way multihead quantum finite automata with two heads. Furthermore, it has been investigated that quantum variant of two-way deterministic multihead finite automata takes less number of heads to recognize a language containing of all words whose length is a prime number.

Mathematics Subject Classification. 81P68, 68Q05, 68Q10, 68Q12, 68Q45.

Received October 24, 2017. Accepted December 18, 2018.

1. Introduction and motivation

Quantum computing combines visionary ideas of Computer Science, Physics, and Mathematics to study theoretical computational models and designing quantum devices which can implement such models. It is based on the quantum phenomena of entanglement and superposition to perform operations [6]. Over the last few decades, interest in quantum computation and information has increased significantly. Shor [35] designed a quantum algorithm for calculating the factors of a large number n which can be applied for cracking the various cryptography algorithms. Soon after the discovery of Shor's algorithm, Grover [9] designed a quantum algorithm for searching an element in a unsorted database set of size n approximately in $\sqrt{n}$ operations.

A quantum finite automaton combines quantum mechanics with classical finite automata. It is a theoretical model with finite memory for quantum computers. Firstly, Moore and Crutchfield [24] and Kondacs and Watrous [15] conceptualized the quantum automata separately. Moore and Crutchfield [24] conceptualized measure-once one-way quantum finite automata (MO-1QFA). In 1997, Kondacs and Watrous [15] proposed another QFA model: measure-many one-way quantum finite automata (MM-1QFA). These models recognize a proper subset

Keywords and phrases: Two-way deterministic finite automata (2DFA), two-way reversible finite automata (2RFA), two-way deterministic multihead finite automata (DMFA), two-way reversible multihead finite automata (RMFA), two-way quantum finite automata (2QFA), two-way multihead quantum finite automata (2MQFA).

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

* Corresponding author: amandeepbhatia.singh@gmail.com

of regular languages. Amano and Iwama [1] proposed 1.5-way quantum finite automata (1.5QFA), in which the tape head is not permitted to move left of the input tape.

In theoretical computational theory, research on two-way multihead finite automata has a long history. In 1960s, Rabin and Scott [33] introduced a multi-tape or multihead finite state automata. Rosenberg [34] studied the basic properties of mutihead finite state automata and also examined their language recognition power. Kutrib and Malcher [17] introduced the variant of one-way reversible automata by introducing multiheads and investigated their properties. Furthermore, it has been proved that one-way two-head finite automata can recognize all uniletter regular languages. In 1970, Ibarra [14] introduced two-way multihead automata and proved that $(n+2)$-head deterministic two-way finite automata is more powerful than n-head deterministic two-way automata, where $n \geq 1$. Further, it has been shown that the class of languages recognized by n-head deterministic/non-deterministic two-way pushdown automata is a subset of the class of languages recognized by $(n+1)$-head deterministic/non-deterministic two-way pushdown automata.

In classical computational theory, it is known that two-way deterministic and nondeterministic multihead finite automata are distinguished by the complexity class L=DSPACE(log (n)) and NL=NSPACE(log(n)) respectively [13]. The class of languages recognized by $log_2\, n$ tape bounded Turing machine and two-way deterministic finite automata are equivalent. But, it is an open problem to determine that whether every context-free language belongs to this family [36]. Therefore, a two-way deterministic (and nondeterministic) multi-head finite automaton can be simulated by a deterministic (and nondeterministic) Turing machine using logarithmic space, and vice versa. Moreover, Sudborough [36] stated that a bounded language $L = \{a^n | n \text{ is a power of } 2\}$ can be recognized by two-way deterministic finite automata with two heads, which cannot be recognized by any one-way multihead finite automata. Hence, two-way multihead deterministic automata is more powerful than its one-way variants.

Lange *et al.* [18] proved that reversible machine can be designed for any deterministic machine by using same amount of space. Morita [25] introduced a two-way reversible multi-head finite automaton (RMFA) and studied its language recognizing capability. It has shown that it can be designed for various context-sensitive and context-free languages with less number of heads. After introducing the concept of two-way reversible multihead finite automata, Morita [26] investigated that deterministic two-way multihead finite automata (DMFA) can be simulated by its reversible model with the same number of heads. It has been proved that DMFA*(k)* can be converted into RMFA*(k)*. Further, it has been shown with an example that a language containing all words of prime number can be recognized by DMFA and RMFA with three heads. Thus, the class of languages recognized by two-way multihead deterministic automata can be characterized by reversible one logarithmic space.

In 1997, Kondacs and Watrous [15] defined the quantum analogue of 2-way deterministic finite state automata named *2-way quantum finite automata* (2QFA). They demonstrated that it is more powerful than the 2-way deterministic finite automata. In 2QFA model, the tape head can read the input tape bi-directionally, or it can be stationary. It is more dominant than the classical model because it allows quantum parallelism with the superposition of states on the input tape. Moreover, 2QFA recognizes non-context free languages in linear time, some context-free languages with one-sided error and regular languages. Ambainis and Watrous [2] presented 2QFA with classical states (2QCFA). In this model, the internal state can be mixed quantum state and the position of head is classical. It is an intermediate model between 1QFA and 2QFA.

The equivalence problems inherent to MO-1QFA, MM-1QFA, and QFA with control language (CL-1QFA) have been explored by various researchers. Mateus *et al.* [22] demonstrated that minimization of a given 1QFA with algebraic numbers is decidable and proposed an algorithm that takes automata as input and produce a minimal size equivalent automata. This algorithm runs in an exponential space (EXPSPACE). Li *et al.* [20] demonstrated the equivalence problem of two CL-1QFAs. It is noted that two CL-1QFSs are equivalent if and only if they are $(k_1 n_1^2 + k_2 n_2^2 - 1)$-equivalent, where n_1 and n_2 signify the minimal DFA states and k_1 and k_2 are the CL-1QFA states. As an application of MM-1QFA equivalence, they demonstrated that two MM-1QFAs were equivalent if and only if they are $(3_1 n_1^2 + 3_2 n_2^2 = 1)$-equivalent, where 3 is the number of states of DFA to recognize a regular language $\{L = g^* a(a|r|g)\}$. Using the algorithm proposed by Mateus *et al.* [22], the state minimization of MM-1QFA is also decidable in EXPSPACE.

Qiu *et al.* [31] studied the equivalence problem of multi-letter QFAs. There is a polynomial-time algorithm that decides the equivalence of two multi-letter QFAs. The state minimization problem of multi-letter QFAs is decidable in EXPSPACE. Li and Qiu [19] addressed the open problem imposed by Gudder [10] and proved the equivalence problem of two quantum sequential machines (QSMs). From the perspective of state complexity, Qiu *et al.* [32] studied one-way quantum finite automata with quantum and classical states (1QFAC) and stated that it is more concise than DFA exponentially. Further, Qiu *et al.* studied the equivalence problem of 1QFAC by a bilinearization technique. The quantum basis state minimization problem of 1QFAC can be solved in EXPSPACE. Further, there exists a polynomial-time $O((k_1n_1)^2 + (k_2n_2)^2 - 1)^4$ algorithm to determine their equivalence. Recently, Zheng *et al.* [39] explored the promise problems recognized by classical, quantum and semi-quantum finite automata.

Qiu and Ying [30] proposed q quantum finite automata (qQFA) and q quantum regular grammar (qQRG). They have verified the equivalence between MO-1QFAs and qQRGs. Furthermore, Qiu and Ying [30] defined q quantum pushdown automata (qQPDA) and investigated the equivalence with quantum pushdown automata (QPDA). It has been proved that the class of languages recognized by QPDA, can be recognized by qQPDA. Li *et al.* [21] studied the generalized variant of one-way quantum finite automata, named as one-way general quantum finite automata (1gQFA). It prompts trace-preserving quantum operation rather than unitary for each symbol in the input alphabet. It has been investigated that generalized variant of MO-1QFA can simulate probabilistic automata and classical DFA. Moreover, Li *et al.* [21] studied generalized variant of MM-1QFA on the basis of equivalence problem and its language recognition power. In 2000, Gudder [10] presented the quantum variant of stochastic sequential machines (SSMs) as sequential quantum machines (SQMs). Bhatia and Kumar [5] designed quantum finite state machine of matrix product state using unitary criteria. On year later, Qiu [28] presented the formal definition of quantum sequential machine (QSM), which is more analogues to SSM. It has been proved that QSM and SQM are equivalent on the basis of language recognition.

Zheng *et al.* [40] explored the power of Arthur-Merlin proof systems (AM) with verifier as 2QCFA. It has been proved that a language $L_{middle} = \{xay \mid x, y \in \{a, b\}^*, |x| = |y|\}$ can be recognized in polynomial expected running time with one-sided error ϵ by quantum Arthur-Merlin proof systems (QAM(2QCFA)). Further, it has been shown that a language $L_{mpal} = \{xax^R | x \in \{a, b\}^*\}$ can be recognized in an exponential expected running time by QAM(2QCFA). But, the languages L_{middle} and L_{mpal} cannot be recognized by two-way probabilistic finite automata (AM(2PFA)). In 2011, Zheng *et al.* [41] introduced two-way two-tape finite automata with the quantum and classical states (2TQCFA) model. They demonstrated efficient 2TFA (two-tape finite automata) algorithms to recognize languages which can be recognized by 2QCFA. They have also introduced k-tape automata with the quantum and classical states (kTQCFA) and proved that it can recognize $\{L = a^n b^{n^k} | n \in N\}$.

Zheng *et al.* [42] proved that $\{xcy | x, y \in \{a, b\}^*, c \in \Sigma, |x| = |y|\}$ can be accepted by 2QCFA in polynomial time with one-side error probability, but it can be accepted by 2PFA with bounded error in exponential time. Zheng *et al.* [43] explored the state complexity of semi-quantum finite automata and compared their space efficiency with classical variants. In communication complexity, they have addressed the strings equality problem using promise problem. It has been shown that a language $L_p = \{a^{kp} \mid k, p \in \mathbb{Z}^+\}$ can be accepted by 2QCFA in a linear expected running time $O(\frac{1}{\epsilon}p^2|w|)$ with one-sided error ϵ using two quantum basis states and a constant number of classical states, where $|w|$ is the length of input string w. Zheng *et al.* [44] proved a state conciseness outcome of 2QCFA by considering number of promise problems. Recently, Bhatia and Kumar [4] modeled ribonucleic acid (RNA) secondary structures namely hairpin loop, internal loop and double helix loop using 2QFA in linear time by one-sided bounded error.

Qiu and Yu [29] studied the concept of multi-letter QFA (QFA_*) proposed by Belovs *et al.* [3]. It has been shown that a language $L = ((a + b)^*b)$ can be recognized by 2-letter QFA with no error, which cannot be recognized by 1QFA. Qiu and Yu extended their work and proved that $(k + 1)$-letter QFA is more powerful than k-letter QFA in language recognition. It has been proved that multi-letter QFA recognize a proper subset of regular languages. There exists a language L containing all words that begin with any number of a and following first b (if there is one), there exists letter a in odd number. It cannot be recognized by neither MM-1QFA with bounded error nor any (QFA_*). As a result, there exits a minimal one-way deterministic finite automata (DFA) for L. Clearly, multi-letter QFA is not powerful than 1DFA.

However, it is known that any deterministic machine can be imitated by reversible machine using space equally. But, this is not in case of one-way reversible finite automata *e.g.* a language $L = \{a^*b^*\}$ can be recognized by 1DFA, but cannot be simulated by one-way reversible finite automata [12]. It is known that one-way reversible finite automata is less powerful than 1DFA. Kondacs and Watrous [15] proved that two-way reversible finite automata can be designed for any DFA (*i.e.* by 2QFA where transition matrices consists only 0s and 1s). Therefore, it has been shown that for every regular language, there exists a two-way reversible finite automata and consequently 2QFA which accepts it in linear time.

Till now, in quantum automata theory, multihead variants of various quantum automata models have been proposed and their language recognition capabilities have been examined. Recently, Ganguly *et al.* [8] proposed one-way multihead quantum finite automata (k-1QFA) and proved that it can recognize all unary languages. Further, it has been investigated that it is more powerful than 1-way reversible 2-head finite automata. Ganguly *et al.* [7] introduced two-tape one-way quantum finite automata (2T1QFA) and claimed that it can recognize all regular languages. It has been investigated that a language which cannot be recognized by any deterministic multihead finite automata can be recognized by 2T1QFA with two heads.

In one-way multihead finite automata, the language $L = \{w_1\$\ w_2\$\ldots\$w_{2n} \mid w_i \in \{a,b\}^*$ and $w_i = w_{2n+1-i}$ for $1 \leq i \leq n\}$ can be recognized with $(k+1)$-heads, which cannot be recognized with k-heads [12]. On the other hand, Monien [23] considered a language over unary alphabet and proved that $(k+1)$-heads are better than k-heads for two-way multihead finite automata. However, the same language can be recognized by 2MQFA with two-heads. In fact, it is possible to design only artificial languages by using diagonalization technique which cannot be recognized by two-way multihead finite automata. In this paper, we focused on the languages which cannot be recognized by 2QFA, but can be recognized by 2-2MQFA.

Although, the various generalizations of 2QFA such as 2QFA with more observables, multi-dimensional 2QFA, multihead 2QFA can be defined. In this paper, we started the investigation of 2QFA by introducing multiheads for refining the language recognition power. It has been proved that 2-2MQFA has more language recognizing power than 1-2QFA and 2QCFA. We have shown that 2-2MQFA can recognize context-free and context-sensitive languages such as $L = \{ww|w \in \{a,b\}^*\}, L = \{xay|x,y \in \{a,b\}^*, |x| = |y|\}$ and $L = \{www|w \in \{a,b\}^*\}$. Although, we find that in a 2-2MQFA where transition matrices consist 0 and 1, *i.e.* basically a two-way reversible multihead finite automata (RMFA). So, 2MQFA can recognize all the languages recognized by RMFA. Morita [26] claimed that RMFA takes few heads to recognize various non-regular and non-context free languages. We have proved that 2MQFA takes less number of heads as compared to RMFA to recognize a prime language by exploiting quantum parallelism.

1.1. Motivation

- 1QFA models are simpler and widely studied by various researchers. Its various variants have been proposed such as multi-letter QFA (QFA_*), multihead 1QFA, generalized 1QFA and many more.
- 2QFA models have not been extensively examined as compared to one-way models [22]. 2QFA is more dominant than classical 2DFA and two-way probabilistic finite automata (2PFA). A language is recognized by 2PFA in exponential time, which can be recognized by 2QFA with bounded error in linear time.
- By exploiting the superposition property of quantum automata, we have introduced the two-way multihead quantum finite automata and determined its language recognition power.

2. Preliminaries

We assume that the reader is intimate with the notations of quantum computational, quantum computing [27, 38] and classical automata theory [11] throughput the paper. Here, we give results and formal definitions of two-way deterministic finite automata, two-way reversible finite automata, two-way deterministic multihead finite automata, two-way quantum finite automata and two-way multihead quantum finite automata.

Definition 2.1. [37] A two-way deterministic finite automaton M_{2DFA} is defined as octuplet $(Q, \Sigma, \#, \$, s, t, r, \delta)$, where

- Q is a set of states,
- Σ is an input alphabet,
- $\#$ is a left-end marker ($\# \notin \Sigma$),
- $\$$ is a right-end marker ($\$ \notin \Sigma$),
- s is a starting state $s \in Q$,
- t is a set of accepting states such that $t \subset Q$,
- r is a set of rejecting states $r \subset Q$ such that $s \neq t$,
- Transition function δ is defined by $\delta : Q \times (\Sigma \cup \{\#, \$\}) \longrightarrow Q \times (\leftarrow, \rightarrow)$, where $(\leftarrow, \rightarrow)$ represents the head function for the left and right direction of tape head.

At any instance, 2DFA is in any state q and read some symbol a or an end-markers, based on q and current symbol, it will move its head one cell in direction $(\leftarrow, \rightarrow)$ and returns a new resultant state p. The tape head does not move beyond the end-markers.

Definition 2.2. [16] A two-way reversible finite automaton is defined as octuplet $(Q, \Sigma, \#, \$,\ s, t, r, \delta)$. The definition of 2RFA is same as 2DFA, the only difference lies in the pair of transitions (*i.e.* the transitions from two different states reading the same symbol cannot lead to single state such that $\delta(q_1, a) = (p, \rightarrow), \delta(q_2, a) = (p, \rightarrow)$ the states q_1 and q_2 must be the same.

It is a special case of two-way deterministic finite automaton, in which every step of computation is logically reversible, *i.e.*, every configuration prompts one-to-one partial mapping from the set of states into itself. It has been investigated that 2RFA is equivalent to 1DFA and 2DFA in language recognition power [15]. The transition matrices of reversible automata consists only 0 and 1. Each row and column has exactly only one entry 1. Therefore, the dot product of any two rows is zero.

Definition 2.3. [26] A two-way multihead finite automaton M_{2MFA} is defined as nontuplet $(Q, \Sigma, k, \#, \$, q_0, A, R, \delta)$, where

- Q is a non-empty finite set of states,
- Σ is an input alphabet,
- k is a number of heads ($k \in 1, 2, \ldots$),
- $\#$ is a left-end marker ($\# \notin \Sigma$),
- $\$$ is a right-end marker ($\$ \notin \Sigma$),
- q_0 is a starting state,
- A is a set of accepting states such that $A \subset Q$,
- R is a set of rejecting states $R \subset Q$ such that $A \cap R = \phi$,
- Transition function δ is a subset of $Q \times ((\Sigma \cup \{\#, \$\})^k \cup \{\leftarrow, \uparrow, \rightarrow\}^k \times Q)$, where $(\leftarrow, \uparrow, \rightarrow)$ represent the left, stationary and right direction of tape head.

The right and left-end markers are not included in Σ (*i.e.*$\{\#, \$\} \cup \Sigma = \phi$). To process the input string by M_{2MFA}, we assume that input string x is written on input tape with both end-markers such that $\#x\$$. The automaton is in any state q and for any $h \in \{0, 1, \ldots, \mid x \mid +1\}^k$. A triplet $\{\#x\$, q, h\}$ is known as configuration of 2MFA on input string x. It defines a function $s_x : \{0, 1, \ldots, \mid x \mid +1\}^k \to (\Sigma \cup \{\#, \$\})^k$. It generates a k-tuple of symbols in $\#x\$$ which is read by k-heads of 2MFA at the position h. Now, we define the notion of deterministic and reversibility for multihead finite automata.

A multihead finite automaton [26] M_{MFA} is said to be deterministic if it holds the following condition:

$$\forall r_1[q_1, s_1, d, q_1^{'}] \in \delta, \forall r_2[q_2, s_2, d, q_2^{'}] \in \delta((q_1 = q_2 \wedge s_1 = s_2) \implies (r_1 = r_2)) \tag{2.1}$$

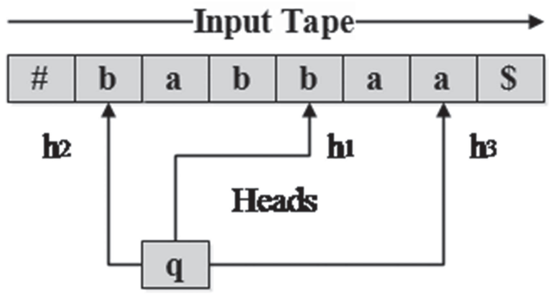

FIGURE 1. Two-way multihead finite automata.

The q_1 state of M_{MFA} moves the tape head according to the $d = (\leftarrow, \uparrow, \rightarrow)$ on reading the symbol s_1 and the resultant state is uniquely determined. Similarly, M_{MFA} is said to be reversible if the following condition holds:

$$\forall r_1[q_1, s_1, d, q_1^{'}] \in \delta, \forall r_2[q_2, s_2, d^{'}, q_2^{'}] \in \delta((q_1^{'} = q_2^{'} \wedge r_1 \neq r_2) \implies (d = d^{'} \wedge s_1 \neq s_2)) \tag{2.2}$$

The two different states on reading the different symbols results to same state and shifts the k-heads in same direction. Figure 1 represents the two-way finite automata with multiheads.

Theorem 2.4. *For any two-way deterministic multihead finite automata, we can design two-way reversible finite automata with same number of heads which recognizes the same language as the former.*

Proof. The proof has been shown in [26]. □

Theorem 2.5. *Two-way quantum finite automata can recognize all regular languages.*

Proof. The proof has been shown in [15]. □

Definition 2.6. [15] A Two-way quantum finite automaton is defined as sextuple $(Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$, 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 alphabet,
- Transition function δ is defined by $\delta : Q \times \Gamma \times Q \times D \to C$, where $\Gamma = \Sigma \cup \{\#, \$\}$ and $D = \{\leftarrow, \uparrow, \rightarrow\}$ represent the left, stationary and right direction of tape head. Transition function must satisfy the following conditions:
  - Local probability and orthogonality condition:

$$\sum_{(q',d)\in Q\times D}^{\forall(q_1,\sigma_1),(q_2,\sigma_2)\in Q\times\Gamma} \overline{\delta(q_1,\sigma,q',d)}\delta(q_2,\sigma,q',d) = \left\{ \begin{array}{l} 1 \; q_1 = q_2 \\ 0 \; q_1 \neq q_2 \end{array} \right\}$$

  - First separability condition:

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

- Second separability condition:

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

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

$$\delta(q, \sigma, q', d) = \left\{ \begin{array}{l|l} \langle q'|V_\sigma|q\rangle & \text{if } D(q') = d \\ 0 & \text{else} \end{array} \right\} \tag{2.3}$$

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 M_{2QFA}, we assume that input string x is written on input with both end-markers such that $\#x\$$. The automaton is in any state q and the tape 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 automaton for processing an input x corresponds a unitary evolution in the inner-product space H_n.

A computation of a 2QFA M_{2QFA} is a sequence of superpositions $c_0, c_1, c_2, \ldots$, where c_0 is an initial configuration. When the automaton is 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 prompts a linear time transformation 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 [15].

Theorem 2.7. *A language L containing of all words that begin with any number of a and following first b (if there is one), there exists letter a in odd number. It cannot be recognized by MM-1QFA and multi-letter QFA with bounded error, but it is recognizable by 2QFA.*

Proof. We construct 2QFA model for L such that $M_{2QFA} = (Q, \Sigma, \delta, q_0, Q_{acc}, Q_{rej})$ be a 2QFA $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{a_1}, q_{a_2}, q_{r_1}, q_{r_2}\}, \Sigma \in \{a, b\}, Q_{acc} = \{q_{a_1}, q_{a_2}\}$ and $Q_{rej} = \{q_{r_1}, q_{r_2}\}$. The state transition functions and head functions are given in Table 1. All vectors formed V_σ are unitary on Hilbert space $L_2(Q)$ and D represents head functions.

TABLE 1. Details of the transition functions and head functions.

$V_\# \|q_0\rangle = \|q_0\rangle$	$V_a \|q_0\rangle = \|q_0\rangle$	$V_b \|q_0\rangle = \|q_1\rangle$	$V_a \|q_1\rangle = \|q_2\rangle$
$V_b \|q_2\rangle = \|q_3\rangle$	$V_a \|q_3\rangle = \|q_4\rangle$	$V_a \|q_4\rangle = \|q_7\rangle$	$V_b \|q_4\rangle = \|q_5\rangle$
$V_b \|q_5\rangle = \|q_4\rangle$	$V_a \|q_5\rangle = \|q_6\rangle$	$V_a \|q_6\rangle = \|q_5\rangle$	$V_b \|q_6\rangle = \|q_7\rangle$
$V_a \|q_7\rangle = \|q_8\rangle$	$V_a \|q_8\rangle = \|q_3\rangle$	$V_b \|q_8\rangle = \|q_9\rangle$	$V_b \|q_9\rangle = \|q_2\rangle$
$V_a \|q_9\rangle = \|q_{10}\rangle$	$V_b \|q_{10}\rangle = \|q_0\rangle$	$V_\$ \|q_4\rangle = \|q_{a_1}\rangle$	$V_\$ \|q_5\rangle = \|q_{a_2}\rangle$
$V_\$ \|q_6\rangle = \|q_{r_1}\rangle$	$V_\$ \|q_7\rangle = \|q_{r_2}\rangle$	$V_\# \|q_1\rangle = \|q_{r_2}\rangle$	
$D(q_{r_1}) = (\uparrow), D(q_{a_1}) = (\uparrow), D(q_0) = (\rightarrow), D(q_1) = (\leftarrow), D(q_2) = (\rightarrow),$ $D(q_3) = (\rightarrow), D(q_4) = (\rightarrow), D(q_5) = (\rightarrow), D(q_6) = (\rightarrow), D(q_7) = (\rightarrow),$ $D(q_8) = (\leftarrow), D(q_9) = (\leftarrow), D(q_{10}) = (\leftarrow), D(q_{r_2}) = (\uparrow), D(q_{a_2}) = (\uparrow)$			

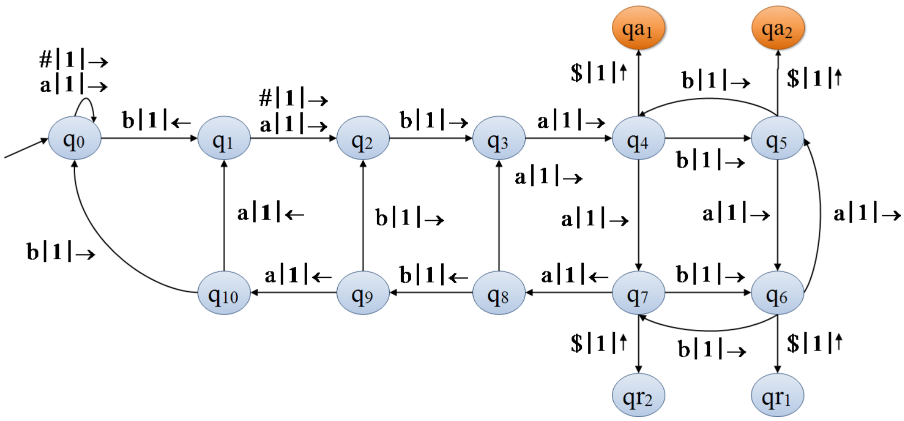

FIGURE 2. State transition diagram of L.

Figure 2 shows the state transition diagram for L. As a result, it can be easily said that a language L can be recognized by two-way reversible finite automata (and by 2QFA) in linear time, which cannot be recognized by any 1QFA and multi-letter QFA [29]. Thus, $L(MM-1QFA)\cup L(QFA_{*})$ is a proper subset of all regular languages after considering a language L. Consequently, now the relation among the 1QFA, multi-letter QFA and 2QFA on the basis of language recognition power is defined as $L(MO-1QFA)\subseteq L(MM-1QFA)\cap L(QFA_{*})\subset L(2QFA)$, where the acronyms of models are used to denote the classes of languages accepted by them, *e.g.* L(2QFA) depicts the class of languages accepted by 2QFA. □

Definition 2.8. A two-way multihead quantum finite automaton $M_{k-2MQFA}$ is a sextuple $(Q,\Sigma,q_0,Q_{acc},Q_{rej},\delta)$, 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 alphabet,
- Transition function δ is defined as $\delta: Q\times\Gamma^k\times Q\times D^k\rightarrow C$, where $\Gamma\in(\Sigma\cup\{\#,\$\})^k$, $D=\{\leftarrow,\uparrow,\rightarrow\}$ represent the left, stationary and right direction of tape head. The computation process of $M_{k-2MQFA}$ is as follows:

Consider a $M_{k-2MQFA}$, at instance, it is in any state q and with the amplitude $\delta(q,\sigma_1,\sigma_2,\ldots,\sigma_k,q^{'},d_1,d_2,\ldots,d_k)$ moves to state $q^{'}$ after reading the input symbols $(\sigma_1,\sigma_2,\ldots,\sigma_k)$ by k-heads and moves the heads according to $(d_1,d_2,\ldots,d_k)$ respectively. Similar to simplified 2QFA, for each $(\sigma_1,\sigma_2,\ldots,\sigma_k)\in\Gamma$, if there exists a unitary linear operator $V_{(\sigma_1,\sigma_2,\ldots,\sigma_k)}$ 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

$$\delta(q,\sigma_1,\sigma_2,\ldots,\sigma_k,q^{'},d_1,d_2,\ldots,d_k)=\left\{\begin{array}{l|l}\left\langle q^{'}\middle|V_{(\sigma_1,\sigma_2,\ldots,\sigma_k)}\middle|q\right\rangle & \text{if } D(q^{'})=(d_1,d_2,\ldots,d_k)\\ 0 & \text{else}\end{array}\right\} \tag{2.4}$$

where $\left\langle q^{'}\middle|V_{(\sigma_1,\sigma_2,\ldots,\sigma_k)}\middle|q\right\rangle$ is a coefficient of $\left|q^{'}\right\rangle$ in $V_{(\sigma_1,\sigma_2,\ldots,\sigma_k)}\left|q\right\rangle$ is said to be well-formed iff

$$\sum_{q^{'}\in Q}\left\langle q^{'}\middle|V_{(\sigma_1,\sigma_2,\ldots,\sigma_k)}\middle|q_1\right\rangle\left\langle q^{'}\middle|V_{(\sigma_1,\sigma_2,\ldots,\sigma_k)}\middle|q_2\right\rangle=\left\{\begin{array}{l}1\ q_1=q_2\\ 0\ q_1\neq q_2\end{array}\right\} \tag{2.5}$$

for each $V_{(\sigma_1,\sigma_2,\ldots,\sigma_k)}$. Equivalently, it is well-formed when every $V_{(\sigma_1,\sigma_2,\ldots,\sigma_k)}$ is unitary.

2.1. Language recognition

A two-way k-head quantum multihead finite automaton is defined as 2QFA in which k-heads reading the input string from single input tape and cannot move beyond the end-markers. We assume that 2MQFA has to be observed to produce information about its processing. Consider an observable O for finite-dimensional Hilbert space H_n, which is decomposed into subspaces such as E_a, E_r, E_n refers to the subspace of 'accept', 'reject' and 'non-halting' respectively. Each of these subspaces are traversed by configurations.

We assume that input string $x \in \Sigma^*$ with both end-markers such that $\#x\$$ is written on an input tape. The processing of input string starts with an initial state and heads pointing towards the left-end marker. Each computation step is composed of two linear operations. Firstly, the unitary transformation operator is performed to the present state and second one is a measurement. It computes several paths simultaneously (quantum parallelism). Then, the result is observed by an observable O. During computation, it is the branching and quantum interference that create parallelism and constructive/destructive superpositions of states result in amplify or destruct the effects of some computation.

Suppose if the automaton is in a superposition state $|\phi\rangle = \alpha_1 |x_1\rangle + \alpha_2 |x_2\rangle + \cdots + \alpha_n |x_n\rangle$ where $\alpha_i^{\prime}s$ are amplitudes and $|\alpha_1|^2 + |\alpha_2|^2 + \cdots + |\alpha_n|^2 = 1$, then the superposition is projected into above-mentioned subspaces $E_j, j \in \{a, r, n\}$. The result of each observation will be either 'accept' or 'reject' or 'non-halting'. The probability of each outcome is determined by the amplitudes of the associated configuration in the present superposition. However, these amplitudes are complex numbers, parallel computation paths can interfere with each other (if their amplitudes cancel each other *i.e.* destructive interference otherwise constructive). The computation is over if the observation results in accepting or rejecting. Sometimes, the computation paths may end in different perpendicular states during observation of some configuration. It results in constructive interference which holds $|\sum_{i=1}^{n} \alpha_i|^2 > \sum_{i=1}^{n} |\alpha_i|^2$. After every measurement, the superposition collapses to the measured subspaces. The outcome of the observation may results in a superposition of halting and non-halting states. Then, the computation continues and superposition collapse to non-halting state. If the amplitude of halting states in some superposition is zero, then the outcome of observation measured in non-halting subspace with probability 1. Therefore, superposition remains unchanged by the observation.

3. Computational power of 2MQFA

In this section, we have shown the language recognition power of 2MQFA for various languages. We have shown that by adding an extra head, 2-2MQFA becomes more powerful than 1-2QFA and 2QCFA. Therefore, a language which cannot be recognized by 2QFA can be recognized by 2MQFA with two heads. First of all, we have shown the superiority of 2-2MQFA over DMFA and RMFA on the basis of number of heads.

Theorem 3.1. $L_1 = \{w^n \mid n > 1, w \in \{a\}\}$ *where the length of w is a prime number, can be recognized by 3-DMFA and 3-RMFA [25, 26]. However, it can be recognized by two-way multihead quantum finite automata with 2-heads.*

Proof. An input string is said to be of prime number if its length is divisible by a number itself. During computation, we will divide the input string with the number $(2 \leq i \leq n)$ or $(2 \leq i \leq n-1)$ after checking its length of odd or even number respectively. The idea of the proof is as follows. The computation is divided into two paths with equal $\frac{1}{\sqrt{2}}$ amplitude. Consider an input string of length 7. Now, we will check its divisibility from number 7 to 2. The computation process is further divided into equal parts. On observation, the computation paths are in superposition of halting and non-halting states. Therefore, after checking divisibility with input string, the computation paths are halted with equal probability in acceptance and rejectance state. But, the computation remains continue with the non-halting states. The input string is said to be accepted (prime number), when it is not divided by any of the number except itself. At the end, second head of both computation paths read the left-end marker at same time and other head points towards their respective position of a number.

Thus, in case of a prime number, both computation paths leading to rejecting configuration interfere destructively, due to which their amplitude sums to 0. Hence, 2-2MQFA accepts the input string with probability 1. In case, if the left-end marker is read by the computation paths (both heads) at the same time, then the accepting configuration interferes destructively. It results in rejection of an input string with probability 1. A 2-2MQFA for L_1 is defined as follows:

$$M_{2-2MQFA} = (Q, \Sigma, q_0, Q_{acc}, Q_{rej}, \delta),$$

where

- Q=$\{q_0, Q_{acc}, Q_{rej}\} \cup \{X_{s,i}, Y_{s,i} \mid 2 \leq s \leq 3 \mid 2 \leq i \leq n+3\} \cup \{q_{xi,i}, q_{yi,i}, q_{zi,i}, q_{bi,i}, q_{i,i}, z_{i,i}, q_{ri,i}, q_{si,i}, q_{wi,i}, q_{ui,i}, q_{li,i}, q_{vi,i}, q_{ci,i}, q_{ti,i}, q_{gi,i}, f_{i,i}, g_{i,i}, q_{pi,k}, z_{i,k}, q_{ui,i} \mid 2 \leq i \leq n, 1 \leq k \leq max(i-1)\} \cup \{q_{pi,0}, q_{ui,0}, q_{ti,0}, x_{i,0}, z_{i,0} \mid 2 \leq i \leq n\}$,
- $\Sigma = \{a\}, q_0$ is an initial state, $Q_{acc} = \{q_{acc}\}$, and $Q_{rej} = \{q_{rej}\}$.
- Each $V_{(\sigma_1, \sigma_2, \ldots, \sigma_k)}$ is unitary by inspection, so $M_{2-2MQFA}$ for L_1 is well-formed. The specification of transition function and head functions are as follows:

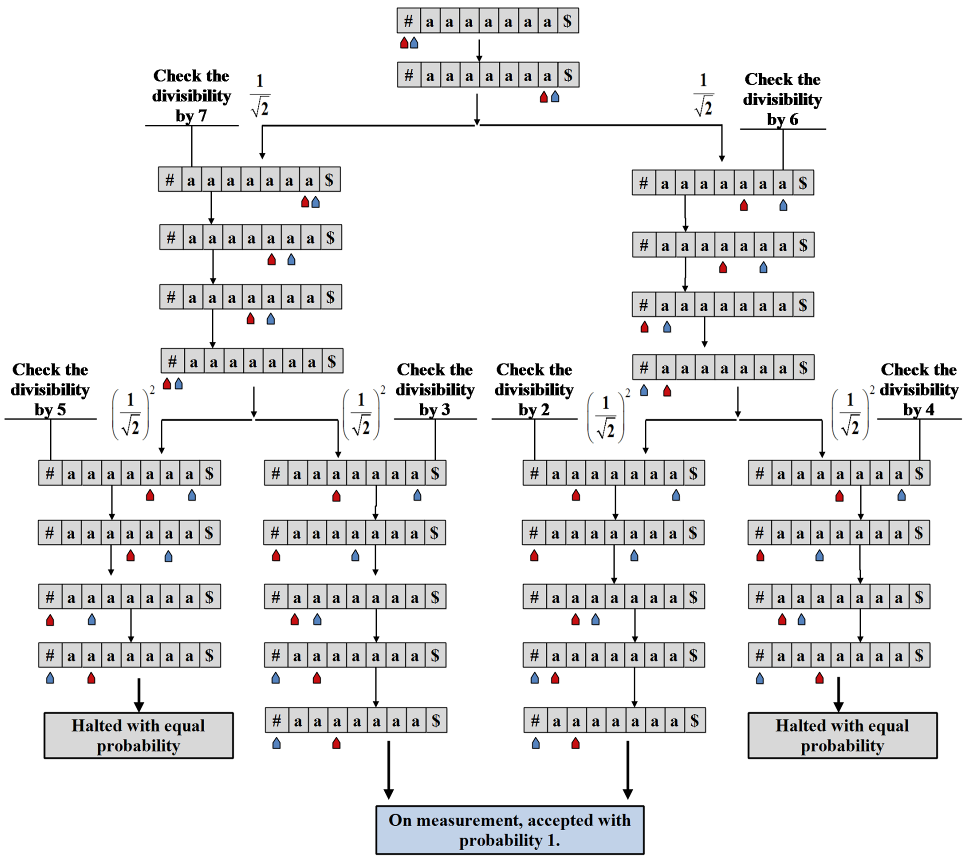

FIGURE 3. Computation process of language L_1 for input $x \in L_1$.

Similarly, we consider an input string x of length 8. The computation process remains same. But, the input string is divisible by more than one number. At the end, one head of computation reads the left-end marker # at same time; but other head of one of the path or both paths does not points towards their respective position of a number. It shows that input string is divisible by (2 or/and 3). It leads to an accepting configuration interfere destructively. Thus, the string is said to be rejected with probability 1. Figures 3 and 4 show the pictorial

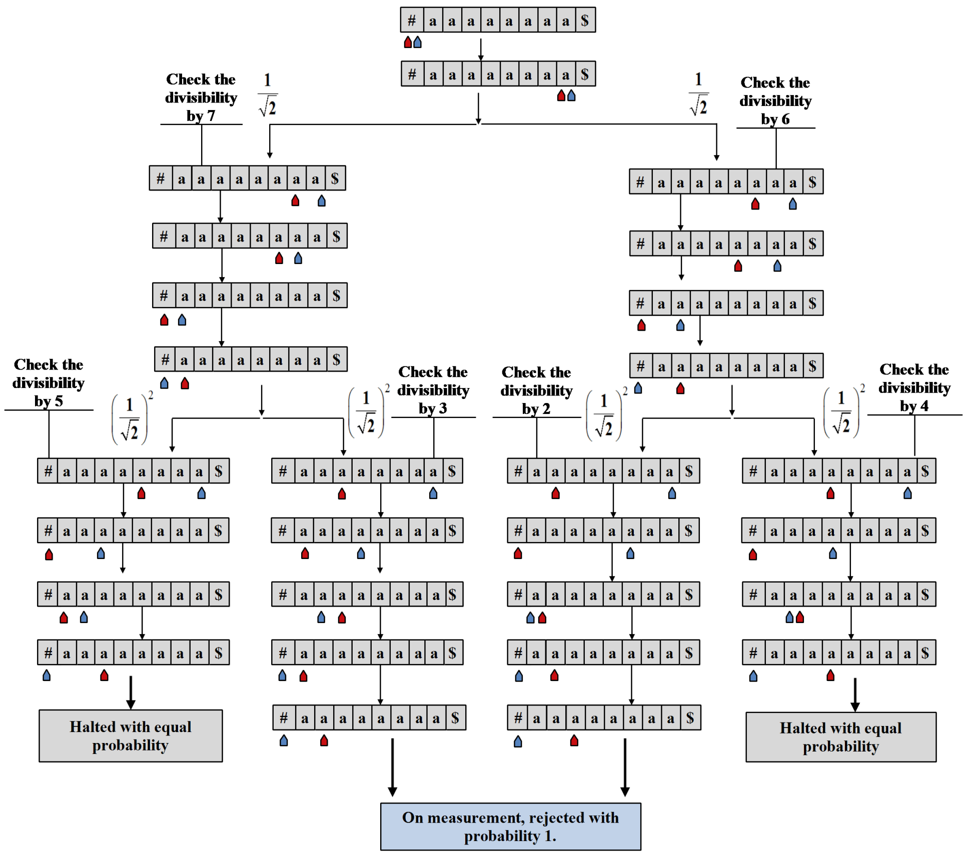

FIGURE 4. Computation process of language L_1 for input $x \notin L_1$.

representation of computation $x \in L_1$ and $x \notin L_1$. The state transition functions and head functions are given in Table 2. □

Theorem 3.2. *For k≥ 1, (k+1)-2MQFA is computationally more powerful than k-2QFA.*

Proof. Initially, we prove it for the case k=1. We claim that a language $L_2 = \{ww \mid w \in \{a,b\}^*\}$ is cannot be recognized by k-2QFA, but it can be recognized by (k+1)-2MQFA (proved in Thm. 3.3). We assume that a language L_2 can be recognized by k-2QFA. Consider an input string $x \in L_2$ *i.e.* written on the input tape enclosed with end-markers such that $\#x\$$. On reading the left-end marker #, the computation is split in two paths $V_\# |q_0\rangle = \frac{1}{\sqrt{2}} |q_1\rangle + \frac{1}{\sqrt{2}} |q_2\rangle$. Now, in first path along state q_1, the tape head moves after remain stationary one time on reading each symbol. Meantime, in other path, the head moves deterministically toward right-end marker \$. This process remains continue until the head does not reach the right-end marker in second path. Suppose, along the second path, the head points towards the right-end marker and simultaneously the head is somewhere in the middle of input string in first path. Now, in order to match the contents, the tape head should be at the beginning of second w in first path when the head is reading the right-end marker in second path. However, the tape head moves deterministically in both paths and first head does not know the position of head in other path. Regardless of the power of 2QFA, there are some languages which cannot be recognized by it. We get a contradiction that language L_2 is cannot be accepted by k-2QFA. But, it can be accepted by (k+1)-2MQFA (proved in Thm. 3.3).

TABLE 2. Details of the transition functions and head function for L_1.

$V_{\#,\#}\left\|q_0\right\rangle = \left\|X_{s,s}\right\rangle, V_{\#,a}\left\|X_{s,s}\right\rangle = \left\|Y_{s+1,s+1}\right\rangle, V_{a,a}\left\|Y_{s,k}\right\rangle = \left\|X_{s-1,k+1}\right\rangle,$
$V_{a,a}\left\|X_{s,l}\right\rangle = \left\|Y_{s,l+1}\right\rangle, V_{a,\$}\left\|Y_{s,l}\right\rangle = \left\|q_{s,l}\right\rangle, V_{a,\$}\left\|X_{s,l}\right\rangle = \left\|q_{s,l}\right\rangle$
$V_{a,a}\left\|q_{s,i}\right\rangle = \frac{1}{\sqrt{2}}\left\|q_{xi,i}\right\rangle + \frac{1}{\sqrt{2}}\left\|q_{yi,i}\right\rangle$, for $i = sk+1, k = 1,2,3,...$
$V_{a,a}\left\|q_{s,i}\right\rangle = \frac{1}{\sqrt{2}}\left\|q_{yi,i}\right\rangle + \frac{1}{\sqrt{2}}\left\|q_{zi,i}\right\rangle$, for $i = (s+1)k, k = 1,2,3,...$
$V_{a,a}\left\|q_{xi,i}\right\rangle = \left\|q_{pi,i}\right\rangle, V_{a,a}\left\|q_{pi,i-1}\right\rangle = \left\|q_{pi,k}\right\rangle, V_{a,a}\left\|q_{pi,k}\right\rangle = \left\|q_{pi,k-1}\right\rangle$
$V_{\#,\#}\left\|q_{pi,0}\right\rangle = \left\|z_{i,i+1}\right\rangle, V_{a,a}\left\|z_{i,k}\right\rangle = \left\|z_{i,k+1}\right\rangle, V_{a,a}\left\|z_{i,i}\right\rangle = \left\|f_{i,i}\right\rangle$
$V_{a,a}\left\|q_{yi,i}\right\rangle = \left\|x_{i,i}\right\rangle, V_{a,a}\left\|q_{y1,1}\right\rangle = \left\|q_{acc}\right\rangle, V_{a,a}\left\|q_{xi,i-1}\right\rangle = \left\|x_{i,k}\right\rangle, V_{a,a}\left\|q_{xi,k}\right\rangle = \left\|x_{i,k-1}\right\rangle$
$, V_{\#,a}\left\|x_{i,0}\right\rangle = \left\|z_{i,0}\right\rangle, V_{a,\#}\left\|z_{i,0}\right\rangle = \left\|z_{i,i+1}\right\rangle, V_{a,\#}\left\|z_{i,k}\right\rangle = \left\|z_{i,k+1}\right\rangle, V_{a,a}\left\|z_{i,k}\right\rangle = \left\|z_{i,k+1}\right\rangle$
$V_{a,a}\left\|z_{i,i}\right\rangle = \left\|g_{i,i}\right\rangle, V_{a,a}\left\|g_{i,i}\right\rangle = \left\|g_{i,i}\right\rangle, V_{a,\$}\left\|g_{i,i}\right\rangle = \left\|k_{i,i}\right\rangle, V_{a,a}\left\|k_{i,i}\right\rangle = \left\|f_{i,i}\right\rangle, V_{a,a}\left\|q_{zi,i}\right\rangle = \left\|q_{bi,i}\right\rangle, V_{a,a}\left\|q_{bi,i}\right\rangle = \left\|x_{i,i}\right\rangle$
$V_{a,a}\left\|f_{i,i}\right\rangle = \frac{1}{\sqrt{2}}\left\|q_{ri,i}\right\rangle + \frac{1}{\sqrt{2}}\left\|q_{si,i}\right\rangle$, for $i > 3$
$V_{a,\#}\left\|z_{i,k}\right\rangle = \frac{1}{\sqrt{2}}\left\|q_{acc}\right\rangle \pm \frac{1}{\sqrt{2}}\left\|q_{rej}\right\rangle, \forall k \neq i, 0 \leq k < 2$
$V_{a,\#}\left\|z_{i,k}\right\rangle = \pm\frac{1}{\sqrt{2}}\left\|q_{acc}\right\rangle + \frac{1}{\sqrt{2}}\left\|q_{rej}\right\rangle, \exists k = i, 2 \leq k \leq 3$
$V_{a,a}\left\|q_{ri,i}\right\rangle = \left\|q_{wi,i}\right\rangle, V_{a,a}\left\|q_{w(i-2),i-2}\right\rangle = \left\|q_{ui,i}\right\rangle, V_{a,a}\left\|q_{ui,i}\right\rangle = \left\|q_{ui,i-1}\right\rangle,$
$V_{a,a}\left\|q_{ui,k}\right\rangle = \left\|q_{ui,k-1}\right\rangle, V_{\#,a}\left\|q_{ui,0}\right\rangle = \left\|p_{i,i+1}\right\rangle, V_{a,a}\left\|p_{i,k}\right\rangle = \left\|p_{i,k+1}\right\rangle,$
$V_{a,a}\left\|p_{i,k}\right\rangle = \left\|q_{ui,i+1}\right\rangle$, for $k = i$,
$V_{a,\#}\left\|q_{ui,k}\right\rangle = \frac{1}{\sqrt{2}}\left\|q_{acc}\right\rangle + \frac{1}{\sqrt{2}}\left\|q_{rej}\right\rangle$, for $3 < i \leq n-2$
$V_{a,\#}\left\|q_{ui,k}\right\rangle = \frac{1}{\sqrt{2}}\left\|q_{acc}\right\rangle \pm \frac{1}{\sqrt{2}}\left\|q_{rej}\right\rangle$, for $2 \leq i \leq 3$
$V_{a,\#}\left\|p_{i,k}\right\rangle = \frac{1}{\sqrt{2}}\left\|q_{acc}\right\rangle + \frac{1}{\sqrt{2}}\left\|q_{rej}\right\rangle, V_{\#,\#}\left\|q_{ui,0}\right\rangle = \frac{1}{\sqrt{2}}\left\|q_{acc}\right\rangle \pm \frac{1}{\sqrt{2}}\left\|q_{rej}\right\rangle$
$V_{a,a}\left\|q_{si,i}\right\rangle = \left\|q_{li,i}\right\rangle, V_{a,a}\left\|q_{li,i}\right\rangle = \left\|q_{vi,i}\right\rangle, V_{a,a}\left\|q_{vi,i}\right\rangle = \left\|q_{ci,i}\right\rangle, V_{a,a}\left\|q_{ci,i}\right\rangle = \left\|q_{t(i-4),i-4}\right\rangle$
$V_{a,a}\left\|q_{ti,i}\right\rangle = \left\|q_{ti,i-1}\right\rangle, V_{a,a}\left\|q_{ti,k}\right\rangle = \left\|q_{ti,k-1}\right\rangle, V_{a,a}\left\|q_{ti,0}\right\rangle = \left\|q_{ri,k+1}\right\rangle,$
$V_{a,a}\left\|q_{ri,k}\right\rangle = \left\|q_{ri,i+1}\right\rangle, V_{a,\#}\left\|q_{ri,i}\right\rangle = \left\|q_{ti,i-1}\right\rangle, V_{a,a}\left\|q_{ri,k}\right\rangle = \left\|q_{yi,k+1}\right\rangle, 0 \leq k < i$
$V_{a,\#}\left\|q_{ti,k}\right\rangle = \left\|q_{yi,k+1}\right\rangle, k \neq 0,\ V_{\#,\#}\left\|q_{ti,0}\right\rangle = \left\|q_{yi,k+1}\right\rangle, V_{a,\#}\left\|q_{yi,k}\right\rangle = \left\|q_{yi,k+1}\right\rangle$
$V_{a,\#}\left\|q_{yi,i}\right\rangle = \left\|q_{fi,i}\right\rangle$, for $3 < i \leq n-2, V_{a,a}\left\|q_{fi,i}\right\rangle = \left\|q_{fi,i}\right\rangle, V_{a,\$}\left\|q_{fi,i}\right\rangle = \left\|q_{gi,i}\right\rangle, V_{a,a}\left\|q_{gi,i}\right\rangle = \left\|q_{fi,i}\right\rangle$
$V_{a,\#}\left\|q_{yi,k}\right\rangle = \frac{1}{\sqrt{2}}\left\|q_{acc}\right\rangle \pm \frac{1}{\sqrt{2}}\left\|q_{rej}\right\rangle, \forall k = i$, for $2 \leq i \leq 3$
$V_{a,\#}\left\|q_{yi,k}\right\rangle = \pm\frac{1}{\sqrt{2}}\left\|q_{acc}\right\rangle + \frac{1}{\sqrt{2}}\left\|q_{rej}\right\rangle, \exists k \neq i$, for $2 \leq i \leq 3$
Head Functions:
$D(q_0) = (\uparrow,\uparrow), D(X_{s,s}) = (\uparrow,\rightarrow)$, for $s = 1, D(Y_{s,k}) = (\rightarrow,\rightarrow), D(X_{s,l}) = (\rightarrow,\rightarrow)$, for $1 \leq l,k \leq n, D(q_{s,i}) = (\uparrow,\leftarrow)$, for $2 \leq i \leq n, D(q_{xi,i}) = (\uparrow,\uparrow), D(q_{yi,i}) = (\leftarrow,\uparrow), D(q_{zi,i}) = (\leftarrow,\uparrow), D(q_{bi,i}) = (\leftarrow,\uparrow), D(q_{pi,i}) = (\leftarrow,\leftarrow), D(q_{pi,k}) = (\leftarrow,\leftarrow), D(z_{i,0}) = (\uparrow,\uparrow), D(z_{i,k}) = (\rightarrow,\leftarrow)$, for $k \neq 0, D(f_{i,i}) = (\uparrow,\uparrow), D(x_{i,i}) = (\leftarrow,\leftarrow), D(x_{i,k}) = (\leftarrow,\leftarrow)$, for $k \neq 0, D(g_{i,i}) = (\uparrow,\rightarrow), D(k_{i,i}) = (\uparrow,\leftarrow), D(q_{ri,i}) = (\uparrow,\uparrow), D(q_{si,i}) = (\uparrow,\uparrow), D(q_{wi,i}) = (\leftarrow,\uparrow), D(q_{ui,i}) = (\leftarrow,\uparrow), D(q_{ui,k}) = (\leftarrow,\leftarrow), D(p_{i,k}) = (\rightarrow,\leftarrow), D(q_{li,i}) = (\leftarrow,\uparrow), D(q_{vi,i}) = (\leftarrow,\uparrow), D(q_{ci,i}) = (\leftarrow,\uparrow), D(q_{ti,i}) = (\leftarrow,\uparrow), D(q_{ti,k}) = (\leftarrow,\leftarrow)$, for $k \neq 0, D(q_{ti,0}) = (\uparrow,\uparrow), D(q_{ri,k}) = (\rightarrow,\leftarrow)$, for $k \neq i, D(q_{ri,i}) = (\uparrow,\uparrow), D(q_{yi,k}) = (\rightarrow,\uparrow), D(q_{fi,i}) = (\rightarrow,\rightarrow), D(q_{gi,i}) = (\uparrow,\leftarrow), D(q_{rej}) = (\uparrow,\uparrow), D(q_{acc}) = (\uparrow,\uparrow)$

Although, k-2QFA [15] with single head can recognize all regular languages, some non-regular languages $L = \{a^n b^n \mid n \geq 1\}$ and non-context free language $L = \{a^n b^n c^n \mid n \geq 1\}$. The power of 1-2QFA comes from the ability of their heads to be in superpositions of basis states corresponding to configurations with the heads in different positions, instead of one position as in the classical model. But, the relationship between 2QFA and its multihead variant is not known with respect to language recognition capability.

Thus, it has been proved that 2-2MQFA is computationally more powerful than 1-2QFA. As a result, in Theorem 3.3, 3.4 and 3.5, we considered the languages $L_2 = \{ww \mid w \in \{a,b\}^*\}, L_3 = \{xay \mid x, y \in \{a,b\}^*, |x| = |y|\}$ and $L_4 = \{www \mid w \in \{a,b\}^*\}$ which cannot be recognized by simple 2QFA and 2QCFA. However, it has been proved that these languages can be recognized by 2QFA with 2-heads. Each transition $V_{(\sigma_1,\sigma_2,\ldots,\sigma_k)}$ is unitary and $M_{2-2MQFA}$ is well-formed for these languages. However, it is still an open problem to find more languages like these that witness the superiority of (k+1)-2MQFA over k-2QFA for $k \geq 2$. □

Theorem 3.3. *Language $L_2 = \{ww \mid w \in \{a,b\}^*\}$ is recognizable by 2MQFA with 2 heads.*

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 starts moving towards the right-end marker \$. After reading a symbol, first head remains stationary at once and other moves forward. If the input string $x \in L_2$, then first head is at beginning of second w in the input string and second head points towards the right-end marker \$. In second phase, the symbols read by first head are getting matched with the symbols read by second head and moves towards the left-end marker #. If any of the symbol read by both heads does not get matched, then it leads to rejectance state q_{rej}. Otherwise, L_2 is said to be accepted if all the symbols read by first head gets matched with the symbols read by second head and first head is reading right-end marker # at the end. A 2-2MQFA for L_2 is defined as follows:

$$M_{2-2MQFA} = (Q, \Sigma, q_0, Q_{acc}, Q_{rej}, \delta),$$

where

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$, where q_1, q_2 and q_3 are used to move the second head towards the \$ and first head remains stationary at once on reading the symbol, q_4 is used to match the symbols of both heads.
- $\Sigma = \{a, b\}, q_0$ is an initial state, $Q_{acc} = \{q_{acc}\}$ and $Q_{rej} = \{q_{rej}\}$.
- In transition functions $r, s \in \{a,b\}^*$ and its specifications are defined in Table 3.

TABLE 3. Details of the transition functions and head function for L_2.

$V_{\#,\#} \|q_0\rangle = \|q_0\rangle$	$V_{\#,r} \|q_0\rangle = \|q_1\rangle$	$V_{r,s} \|q_1\rangle = \|q_2\rangle$	$V_{r,\$} \|q_1\rangle = \|q_{rej}\rangle$
$V_{r,s} \|q_2\rangle = \|q_1\rangle$	$V_{r,\$} \|q_2\rangle = \|q_3\rangle$	$V_{a,a} \|q_4\rangle = \|q_4\rangle$	$V_{b,b} \|q_4\rangle = \|q_4\rangle$
$V_{r,\$} \|q_3\rangle = \|q_4\rangle$	$V_{\#,r} \|q_4\rangle = \|q_{acc}\rangle$	$V_{b,a} \|q_4\rangle = \|q_{rej}\rangle$	$V_{a,b} \|q_4\rangle = \|q_{rej}\rangle$
Head Functions:			
$D(q_0) = (\uparrow, \rightarrow), D(q_1) = (\rightarrow, \rightarrow), D(q_2) = (\uparrow, \rightarrow), D(q_3) = (\rightarrow, \uparrow),$ $D(q_4) = (\leftarrow, \leftarrow), D(q_{acc}) = (\uparrow, \uparrow), D(q_{rej}) = (\uparrow, \uparrow)$			

Each $V_{(\sigma_1,\sigma_2,\ldots,\sigma_k)}$ is unitary, it obeying unitary time evolution defined by local interactions between its symbols which permitting being in a superposition of configurations for each $(\sigma_1, \sigma_2, \ldots, \sigma_k) \in \Gamma$. □

Theorem 3.4. *Language $L_3 = \{xay \mid x, y \in \{a,b\}^*, |x| = |y|\}$ is recognizable by 2MQFA with 2 heads.*

Proof. It consists of two phases. First we keep the first head stationary at once and other moves forward on reading symbols towards the left-end marker \$. At the end of first phase, if the middle symbol is a, then it proceed to second phase to check $|x| = |y$. Suppose, if the middle symbol is not a, then L_3 is said to be rejected. In second phase, we keep the first head stationary for once and other head keep moving towards left side. At the

TABLE 4. Details of the transition functions and head function for L_3.

$V_{\#,\#}\,\|q_0\rangle = \|q_0\rangle$	$V_{r,r}\,\|q_0\rangle = \|q_1\rangle$	$V_{r,s}\,\|q_1\rangle = \|q_2\rangle$
$V_{a,\$}\,\|q_1\rangle = \|q_3\rangle$	$V_{r,s}\,\|q_3\rangle = \|q_4\rangle$	$V_{r,s}\,\|q_4\rangle = \|q_3\rangle$
$V_{b,\$}\,\|q_1\rangle = \|q_{rej}\rangle$	$V_{r,s}\,\|q_2\rangle = \|q_1\rangle$	$V_{\#,\#}\,\|q_4\rangle = \|q_{acc}\rangle$
Head Functions:		
$D(q_0) = (\uparrow,\uparrow), D(q_1) = (\uparrow,\rightarrow), D(q_2) = (\rightarrow,\rightarrow), D(q_3) = (\uparrow,\leftarrow),$ $D(q_4) = (\leftarrow,\leftarrow), D(q_{rej}) = (\uparrow,\uparrow), D(q_{acc}) = (\uparrow,\uparrow)$		

end, if the both heads read the left-end marker at same time, it is said to be accepted ($w \in L_3$). A 2-2MQFA for L_3 is defined as follows:

$$M_{2-2MQFA} = (Q, \Sigma, q_0, Q_{acc}, Q_{rej}, \delta),$$

where

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$, where q_1 and q_2 are used to move the second head towards the \$ and first head remains stationary at once, q_3 and q_4 are used to move the both heads towards left side after checking the middle symbol a. On reading the left-hand marker by both heads at same time, the state q_4 is changed to q_{acc}.
- $\Sigma = \{a, b\}, q_0$ is an intial state, $Q_{acc} = \{q_{acc}\}$ and $Q_{rej} = \{q_{rej}\}$.
- In transition functions $r, s \in \{a, b\}^*$ and its specifications are defined in Table 4.

It is easy to verify that L_3 can be recognized by $M_{2-2MQFA}$. First phase checks whether the middle symbol is a or not. Finally, second phase confirms the length $|x| = |y|$. □

Theorem 3.5. *Language $L_4 = \{www \mid w \in \{a, b\}^*\}$ is recognizable by 2-2MQFA.*

Proof. The $M_{2-2MQFA}$ to accept this language is identical to the one accepting L_2. The idea of this proof is as follows. It consists of four phases. First, we keep the first head stationary two times and other moves forward. This process remains continue until the second head does not read right-end marker \$. If the input string $x \in L_4$, then first head is at beginning of second w in the input string and second head points towards the right-end marker \$. In second phase, the symbols read by first head are getting matched with the second head and moves towards left-end marker #. If any of the symbol read by both heads does not get matched, then it leads to rejectance state q_{rej}.

Otherwise, it confirms that the contents of first and third w are same. In third phase, we keep the second head stationary two times and other head moves forward towards \$. Finally, we repeat the second phase. The contents of all w's get matched with corresponding one respectively. If any of the symbols does not matched between both heads, then the input sting is said to be rejected. At the end, if the first read points the left-end marker #, which shows that the all the contents gets matched and string is accepted. A 2-2MQFA for L_4 is defined as follows:

$$M_{2-2MQFA} = (Q, \Sigma, q_0, Q_{acc}, Q_{rej}, \delta),$$

where

- $Q = \{q_0, q_1, q_2, q_3,\ q_4, q_5, q_6, q_7, q_8, q_{acc}, q_{rej}\}$, where q_1, q_2, q_3 and q_8 are used to keep the first head stationary two times and second head to move towards the \$, states q_4 and q_7 are used to match the symbols between both heads, q_5, q_6 used in third phase to move the first head towards the right side and second head remains stationary at once, q_4 and q_7 are changed to q_{rej} on reading the different symbols by both heads.
- $\Sigma = \{a, b\}, q_0$ is an initial state, $Q_{acc} = \{q_{acc}\}$ and $Q_{rej} = \{q_{rej}\}$.
- The specification of transition function and head function are as follows, where $r, s \in \{a, b\}^*$.

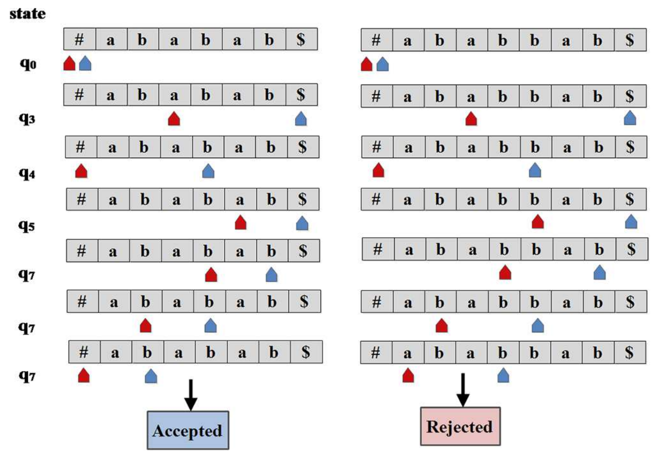

FIGURE 5. Computation process of language L_4 for inputs $ababab \in L_4$ (left) and $ababbab \notin L_4$ (right). In second case, $M_{2-2MQFA}$ halts with the two heads leaving at the positions when the symbols do not match.

TABLE 5. Details of the transition functions and head function for L_4.

$V_{\#,\#}\|q_0\rangle = \|q_0\rangle$	$V_{\#,r}\|q_0\rangle = \|q_1\rangle$	$V_{r,s}\|q_1\rangle = \|q_2\rangle$	$V_{r,s}\|q_2\rangle = \|q_3\rangle$
$V_{r,s}\|q_3\rangle = \|q_1\rangle$	$V_{r,\$}\|q_3\rangle = \|q_8\rangle$	$V_{r,\$}\|q_8\rangle = \|q_4\rangle$	$V_{a,a}\|q_4\rangle = \|q_4\rangle$
$V_{a,b}\|q_4\rangle = \|q_{rej}\rangle$	$V_{b,b}\|q_4\rangle = \|q_4\rangle$	$V_{\#,r}\|q_4\rangle = \|q_5\rangle$	$V_{r,s}\|q_5\rangle = \|q_6\rangle$
$V_{r,s}\|q_6\rangle = \|q_5\rangle$	$V_{b,a}\|q_4\rangle = \|q_{rej}\rangle$	$V_{a,a}\|q_7\rangle = \|q_7\rangle$	$V_{b,b}\|q_7\rangle = \|q_7\rangle$
$V_{a,b}\|q_7\rangle = \|q_{rej}\rangle$	$V_{r,\$}\|q_5\rangle = \|q_7\rangle$	$V_{\#,r}\|q_7\rangle = \|q_{acc}\rangle$	$V_{r,\$}\|q_1\rangle = \|q_{rej}\rangle$
$V_{r,\$}\|q_2\rangle = \|q_{rej}\rangle$	$V_{b,a}\|q_7\rangle = \|q_{rej}\rangle$		
Head Functions:			
$D(q_0) = (\uparrow, \rightarrow), D(q_1) = (\rightarrow, \rightarrow), D(q_2) = (\uparrow, \rightarrow), D(q_3) = (\uparrow, \rightarrow),$ $D(q_4) = (\leftarrow, \leftarrow), D(q_5) = (\rightarrow, \rightarrow), D(q_6) = (\rightarrow, \uparrow), D(q_7) = (\leftarrow, \leftarrow)$ $D(q_8) = (\rightarrow, \uparrow), D(q_{rej}) = (\uparrow, \uparrow), D(q_{acc}) = (\uparrow, \uparrow)$			

It can be easily verify that each $V_{(\sigma_1,\sigma_2)}$ is unitary and obeying unitary time evolution defined by local interactions between its symbols. Figure 5 shows the computation process of $M_{2-2MQFA}$ for L_4. The state transition functions and head functions are given in Table 5. □

4. CONCLUSION

In this paper, we focused on variant of 2QFA named two-way multihead quantum finite automata. There are various quantum variants which have been proved more powerful than classical ones. Firstly, we have proved that a language $L = a^*b(a^2)^*a$ can be recognized by 2QFA, which cannot be recognized by any 1QFA and multi-letter QFA. We investigated the language recognition capability of 2MQFA. There are many context-free languages and context-sensitive languages recognized by 2MQFA other than presented in this paper. We have designed a 2MQFA for a prime language with two-heads as compared to three-heads DMFA. Although, DMFA can be converted to RMFA with the same number of heads for any language. Therefore, it has been proved that 2MQFA takes less number of heads to compute a prime language over DMFA and RMFA. Furthermore, we have shown the recognizability results of 2-2MQFA as compared to 2QFA, 2QCFA and proved that by adding an extra head it becomes more powerful. Hence, the class of languages recognized by 2QFA is a subset of class of languages recognized by 2-2MQFA. In future, we will investigate multidimensional 2QFA and its comparison with classical counterparts.

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

References

[1] M. Amano and K. Iwama, Undecidability on quantum finite automata, in *Proc. of the Thirty-first Annual ACM Symposium on Theory of Computing*. ACM (1999) 368–375.

[2] A. Ambainis and J. Watrous, Two-way finite automata with quantum and classical states. *Theor. Comput. Sci.* **287** (2002) 299–311.

[3] A. Belovs, A. Rosmanis and J. Smotrovs, Multi-letter reversible and quantum finite automata, in *International Conference on Developments in Language Theory*. Springer (2007) 60–71.

[4] A.S. Bhatia and A. Kumar, Modeling of RNA secondary structures using two-way quantum finite automata. *Chaos, Solitons Fract.* **116** (2018) 332–339.

[5] A.S. Bhatia and A. Kumar, Quantifying matrix product state. *Quant. Inf. Process.* **17** (2018) 41.

[6] R.P. Feynman, Simulating physics with computers. *Int. J. Theor. Phys.* **21** (1982) 467–488.

[7] D. Ganguly and K.S. Ray, 2-tape 1-way quantum finite state automata. Preprint arXiv:1607.00811 (2016).

[8] D. Ganguly and K. Chatterjee, K.S. Ray, 1-way multihead quantum finite state automata. *Appl. Math.* **7** (2016) 1005.

[9] L.K. Grover, A fast quantum mechanical algorithm for database search, in *Proc. of the Twenty-eighth Annual ACM Symposium on Theory of Computing*. ACM (19960 212–219.

[10] S. Gudder, Quantum computers. *Int. J. Theor. Phys.* **39** (2000) 2151–2177.

[11] I. Hill III, Introduction to automata theory, languages, and computation, Addison Wesley, Boston, Ma (1979).

[12] M. Holzer and M. Kutrib, A. Malcher, Multi-head finite automata: Characterizations, concepts and open problems. Preprint arXiv:0906.3051 (2009).

[13] M. Holzer, M. Kutrib and A. Malcher, Complexity of multi-head finite automata: origins and directions. *Theor. Comput. Sci.* **412** (2011) 83–96.

[14] O.H. Ibarra, On two-way multihead automata. *J. Comput. Syst. Sci.* **7** (1973) 28–36.

[15] A. Kondacs and J. Watrous, On the power of quantum finite state automata, in *38th Annual Symposium on Foundations of Computer Science, 1997*. IEEE (1997) 66–75.

[16] M. Kunc and A. Okhotin, Reversible two-way finite automata over a unary alphabet. Technical Report 1024, Turku Centre for Computer Science (2011).

[17] M. Kutrib and A. Malcher, One-way reversible multi-head finite automata, in *International Workshop on Reversible Computation*. Springer (2012) 14–28.

[18] K.-J. Lange, P. McKenzie and A. Tapp, Reversible space equals deterministic space. *J. Comput. Syst. Sci.* **60** (2000) 354–367.

[19] L. Li and D. Qiu, Determination of equivalence between quantum sequential machines. *Theor. Comput. Sci.* **358** (2006) 65–74.

[20] L. Li, D. Qiu, Determining the equivalence for one-way quantum finite automata. *Theor. Comput. Sci.* **403** (2008) 42–51.

[21] L. Li, D. Qiu, X. Zou, L. Li, L. Wu and P. Mateus, Characterizations of one-way general quantum finite automata. *Theor. Comput. Sci.* **419** (2012) 73–91.

[22] P. Mateus, D. Qiu and L. Li, On the complexity of minimizing probabilistic and quantum automata. *Inf. Comput.* **218** (2012) 36–53.

[23] B. Monien, Two-way multihead automata over a one-letter alphabet. *RAIRO: ITA* **14** (1980) 67–82.

[24] C. Moore and J.P. Crutchfield, Quantum automata and quantum grammars. *Theor. Comput. Sci.* **237** (2000) 275–306.

[25] K. Morita, Two-way reversible multi-head finite automata. *Fund. Inf.* **110** (2011) 241–254.

[26] K. Morita, A deterministic two-way multi-head finite automaton can be converted into a reversible one with the same number of heads, in *International Workshop on Reversible Computation*. Springer (2012) 29–43.

[27] M.A. Nielsen and I.L. Chuang, Quantum computation and quantum information. Cambridge University Press (2010).

[28] D. Qiu, Characterization of sequential quantum machines. *Int. J. Theor. Phys.* **41** (2002) 811–822.

[29] D. Qiu and S. Yu, Hierarchy and equivalence of multi-letter quantum finite automata. *Theor. Comput. Sci.* **410** (2009) 3006–3017.

[30] D. Qiu and M. Ying, Characterizations of quantum automata. *Theor. Comput. Sci.* **312** (2004) 479–489.

[31] D. Qiu, L. Li, X. Zou, P. Mateus and J. Gruska, Multi-letter quantum finite automata: decidability of the equivalence and minimization of states. *Acta Tnf.* **48** (2011) 271.

[32] D. Qiu, L. Li, P. Mateus and A. Sernadas, Exponentially more concise quantum recognition of non-RMM regular languages. *J. Comput. Syst. Sci.* **81** (2015) 359–375.

[33] M.O. Rabin and D. Scott, Finite automata and their decision problems. *IBM J. Res. Dev.* **3** (1959) 114–125.

[34] A.L. Rosenberg, On multi-head finite automata. *IBM J. Res. Dev.* **10** (1966) 388–394.

[35] P.W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, in *35th Annual Symposium on Foundations of Computer Science*. IEEE (1994) 124–134.

[36] I.H. Sudborough, Bounded-reversal multihead finite automata languages. *Inf. Cont.* **25** (1974) 317–328.

[37] Two-way deterministic finite automata. Available at: https://en.wikipedia.org/wiki/two-way_deterministic_finite_automaton (2017).

[38] J. Wang, Handbook of Finite State Based Models and Applications. CRC Press (2012).

[39] S. Zheng, L. Li, D. Qiu and J. Gruska, Promise problems solved by quantum and classical finite automata. *Theor. Comput. Sci.* **666** (2017) 48–64.

[40] S. Zheng, D. Qiu and J. Gruska, Power of the interactive proof systems with verifiers modeled by semi-quantum two-way finite automata. *Inf. Comput.* **241** (2015) 197–214.

[41] S. Zheng, L. Li and D. Qiu, Two-tape finite automata with quantum and classical states. *Int. J. Theor. Phys.* **50** (2011) 1262–1281.

[42] S. Zheng, D. Qiu and L. Li, Some languages recognized by two-way finite automata with quantum and classical states. *Int. J. Found. Comput. Sci.* **23** (2012) 1117–1129.

[43] S. Zheng, J. Gruska and D. Qiu, On the state complexity of semi-quantum finite automata. *RAIRO: ITA* **48** (2014) 187–207.

[44] S. Zheng, D. Qiu, J. Gruska, L. Li and P. Mateus, State succinctness of two-way finite automata with quantum and classical states. *Theor. Comput. Sci.* **499** (2013) 98–112.