

Evolutionary Logic Synthesis of Quantum Finite State Machines for Sequence Detection

Martin Lukac¹ and Marek Perkowski²

¹*Intelligent Integrated Systems Laboratory GSIS, Tohoku University,*

²*Department of Electrical and Computer Engineering, Portland State University*

¹*Japan*

²*USA*

1. Introduction

Quantum Finite State Machines (QFSM) are a well known model of computation that was originally formalized by Watrous [Wat95a, Wat95b, Wat97], Kondacs [KW97] and more generally Quantum Turing Machines (QTM) have been described by Bernstein [BV97]. In particular the 2-way QFSM have been shown to be more powerful than classical FSM [KW97]. Thus the interest in quantum computational models of automata and machines is not only theoretical but has also possible applications realization of future quantum computer and robotics controllers.

In this chapter we present the evolutionary approach to the synthesis of QFSM's specified by a quantum circuits. This approach was originally proposed by [LP09] and is possible on yet only theoretical basis. In particular this approach requires a selective qubit-initialization in a quantum register. In contrast the current methodology and approaches to practical Quantum Computation, the current practical realization of quantum computation always starts with the initialization of the whole quantum register and terminates by the measurement of either all of the qubits or by the measurement of a given subset of qubits. Moreover in general there is no reuse of any element of the quantum register.

In this text we analyze in details what type of QFSM can be successfully synthesized.

The evolutionary approach will evaluate the results based on both the correctness and the cost of the evolved machines. Multiple parameters such as type of error evaluation, synthesis constraints and evolutionary operators will be discussed when evaluating to the obtained results.

In particular we show how to synthesize QFSMs as sequence detectors and illustrate their functionality both in the quantum world and in the classical (observable) world. The application of the synthesized quantum devices is illustrated by the analysis of recognized sequences.

Finally, we provide analytic method for the used evolutionary approach and we describe the experimental protocol, and its heuristic improvements. We also discuss the results. In addition, we investigate the following aspects of the Evolutionary Quantum Logic Synthesis:

- Quantum probabilistic FSM and Reversible FSM.
- Hardware acceleration for the Fitness evaluation using CBLAS [cbl] and using CUBLAS [cud] (CUDA[cud] implemented Basic Linear Algebra Subprograms (BLAS)[cbl] subroutines).

Source: New Achievements in Evolutionary Computation, Book edited by: Peter Korosec,
ISBN 978-953-307-053-7, pp. 318, February 2010, INTECH, Croatia, downloaded from SCIYO.COM

2. Background in quantum computing

In Quantum Computing the information is represented by a Quantum Bit also called qubit. The wave equation is used to represent a qubit or a set of them. Equation 1 shows a general form in the Dirac notation.

$$\begin{aligned} |\phi\rangle &= e^{i\rho} \cos\theta + e^{i(\rho+\psi)} \sin\theta \\ &= e^{i\rho} (\cos\theta + e^{i\psi} \sin\theta) \end{aligned} \quad (1)$$

In Dirac notation $|\cdot\rangle$ represents a column vector, also called a *ket*. The *bra* element denoted $\langle\cdot|$ stands for hermitian conjugate. In this manner a bra-ket $\langle\cdot|\cdot\rangle$ represents the inner, dot-vector product while $|\cdot\rangle\langle\cdot|$ represents the outer vector product. The general equation (1), $e^{i\rho} \cos\theta|0\rangle + e^{i(\rho+\psi)} \sin\theta|1\rangle$ can be written as $\alpha|0\rangle + \beta|1\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$ and $|\alpha|^2$ is the probability of observing the state $|0\rangle$ while $|\beta|^2$ is the probability of observing $|1\rangle$.

In general, to describe basis states of a Quantum System, the Dirac notation is preferred to the vector-based Heisenberg notation. However, Heisenberg notation can be more practical to represent the exponential growth of the quantum register. Let two orthonormal quantum states be represented in the vector (Heisenberg) notation eq. 2.

$$\begin{aligned} |\uparrow\rangle = |0\rangle &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ |\downarrow\rangle = |1\rangle &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \quad (2)$$

Different states in this vector notation are then multiplications of all possible states of the system, and for a two-qubit system we obtain (using the Kronecker product [Gru99, Gra81, NC00]) the states represented in eq. 3:

$$\begin{aligned} |00\rangle &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} & |10\rangle &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\ |01\rangle &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & |11\rangle &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned} \quad (3)$$

The Kronecker product exponentially increases the dimension of the space for matrices as well:

$$I \otimes X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4)$$

This tensor product operation for a parallel connection of two wires is shown in Figure 1.

Assume that qubit a (with possible states $|0\rangle$ and $|1\rangle$) is represented by $|\Psi_a\rangle = \alpha_a|0\rangle + \beta_a|1\rangle$ and qubit b is represented by $|\Psi_b\rangle = \alpha_b|0\rangle + \beta_b|1\rangle$. Each of them is represented by the

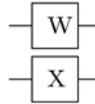


Fig. 1. Circuit representing the $W \otimes X$ operation

superposition of their basis states, but put together the characteristic wave function of their combined states will be:

$$\begin{aligned}
 |\Psi_a \Psi_b\rangle &= \alpha_a \alpha_b |00\rangle + \alpha_a \beta_b |01\rangle \\
 &+ \beta_a \alpha_b |10\rangle + \beta_a \beta_b |11\rangle
 \end{aligned}
 \tag{5}$$

with α_a and β_b being the complex amplitudes of states of each EP respectively. As shown before, the calculations of the composed state used the Kronecker multiplication operator. Hence comes the possibility to create quantum memories with extremely large capacities and the requirement for efficient methods to calculate such large matrices.

Quantum Computation uses a set of Quantum properties. These are the measurement, the superposition and the entanglement. First, however, the principles of multi-qubit system must be introduced.

2.1 Multi-Qubit System

To illustrate the superposition let’s have a look at a more complicated system with two quantum particles a and b represented by $|\psi_a\rangle = \alpha_0|0\rangle + \beta_a|1\rangle$ and $|\psi_b\rangle = \alpha_b|0\rangle + \beta_b|1\rangle$ respectively. For such a system the problem space increases exponentially and is represented using the Kronecker product [Gru99].

$$|\psi_a\rangle \otimes |\psi_b\rangle = \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix} \otimes \begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \alpha_0 \alpha_1 \\ \alpha_0 \beta_1 \\ \beta_0 \alpha_1 \\ \beta_0 \beta_1 \end{bmatrix}
 \tag{6}$$

Thus the resulting system is represented by $|\psi_a \psi_b\rangle = \alpha_a \alpha_b |00\rangle + \alpha_a \beta_b |01\rangle + \beta_a \alpha_b |10\rangle + \beta_a \beta_b |11\rangle$ (5) where the double coefficients obey the unity (completeness) rule and each of their powers represents the probability to measure the corresponding state. The superposition means that the quantum system is or can be in any or all the states at the same time. This superposition gives the massive parallel computational power to quantum computing.

2.2 Entanglement and projective measurements

Assume the above two-particle vector (two-qubit quantum system) is transformed using the quantum circuit from Figure 2.

This circuit executes first a Hadamard transform on the top qubit and then a Controlled-Not operation with the bottom qubit as the target. Depending on the initial state of the quantum register the output will be either $|\psi_a \psi_b\rangle = \alpha_a \alpha_b |00\rangle \pm |\beta_a \beta_b |11\rangle$ or $|\psi_a \psi_b\rangle = \alpha_a \beta_b |01\rangle \pm |\beta_a \alpha_b |10\rangle$. Thus it is not possible to estimate with 100% probability the initial state of the quantum register.

Let $|ab\rangle = |00\rangle$ at level a (Figure 2). The first step is to apply the $[H]$ gate on the qubit-a and the resulting state at level b of the circuit is

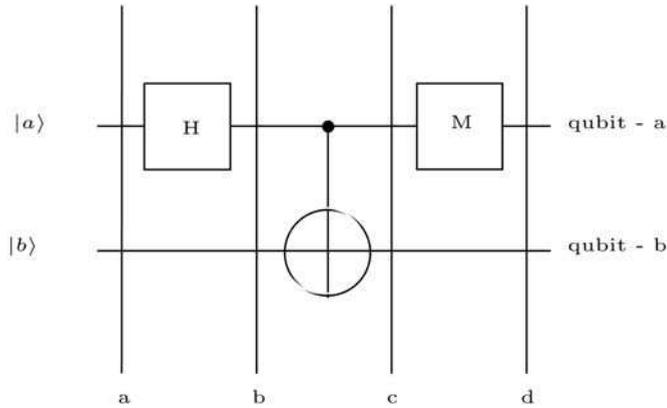


Fig. 2. EPR producing circuit

$$\begin{aligned}
 |ab\rangle &\rightarrow (H \otimes W)|ab\rangle \\
 &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle \\
 &= \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} \tag{7}
 \end{aligned}$$

Next the application of the CNOT gate results in:

$$|\psi_a \psi_b\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{8}$$

For an output 0 (on the qubit-a), the projective measurement of the first (topmost) qubit (qubit-a on Figure 2) on this stage would collapse the global state (with a single measurement) to the state $|00\rangle$:

$$|ab\rangle \rightarrow \frac{M_0|ab\rangle}{\sqrt{\langle ab|M_0^\dagger M_0|ab\rangle}} = [1 \ 0 \ 0 \ 0]^T = |00\rangle \tag{9}$$

with

$$M_0|ab\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

and

$$\begin{aligned}
 \sqrt{\langle ab|M_0^\dagger M_0|ab\rangle} &= \sqrt{\frac{1}{2}} [1 \ 0 \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\
 &= \frac{1}{\sqrt{2}} [1 \ 0 \ 0 \ 1] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
 &= \frac{1}{\sqrt{2}}
 \end{aligned} \tag{11}$$

Similarly, the probability of measuring output on the qubit-a in state $|0\rangle$ is:

$$\begin{aligned}
 p(0) &= \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \\
 &= \frac{1}{2}
 \end{aligned} \tag{12}$$

If one would look to the output of the measurement on the second qubit (qubit-b), the probability for obtaining $|0\rangle$ or $|1\rangle$ is in this case the following:

$$\begin{aligned}
 p(0) &= \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \\
 &= p(1) = \frac{1}{2}
 \end{aligned} \tag{13}$$

Thus the expectation values for measuring both values 0 or 1 on each qubit independently are $\frac{1}{2}$.

If however one looks on the second and non-measured qubit (if the qubit-a is measured, it is the qubit-b, and vice versa) and calculates the output probabilities, the output is

contradictory to the expectations given by standard probabilistic distribution such as a coin toss $q = 1 - p$. To see this let's start in the state

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \quad (14)$$

and measure the qubit-a and obtain a result. In this case assume the result of the measurement is given by:

$$|\Psi\rangle \rightarrow \frac{M_0|\Psi\rangle}{\sqrt{\langle\Psi|M_0^\dagger M_0|\Psi\rangle}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (15)$$

Then measuring the second qubit (qubit-b) will not affect the system because the measurement of the qubit-a has collapsed the whole system into a single basis state:

$$|\Psi\rangle \xrightarrow{M} |00\rangle \quad (16)$$

The probability for obtaining a $|1\rangle$ on the qubit-b is thus 0 and the measurement on qubit-b (after having measured qubit-a) has no effect on the system at all. The states of qubits are thus correlated. This non-locality paradox was first described by Einstein-Podolsky-Rosen work[EPR35] and is known as the EPR paradox. This particular phenomenon is one of the most powerful in quantum mechanics and quantum computing, as it allows together with superposition the speedup of finding solutions to certain types of problems. Finally, it can be noted that mathematically, the entangled state is such that it cannot be factored into simpler terms. For example, the state $\frac{(|00\rangle+|01\rangle)}{\sqrt{2}} \rightarrow \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|0\rangle$ and thus it can be factored. However, the states as those introduced in eq. 15 cannot be transformed in such a manner and are thus entangled; physically implying that they are related through measurement or observation. \square

2.3 Single-Qubit quantum gates

We are now concerned with matrix representation of operators. The first class of important quantum operators are the one-qubit operators realized in the quantum circuit as the one-qubit (quantum) gates. Some of their matrix representations can be seen in equation 17.

$$\begin{aligned} a) \quad X &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & b) \quad Y &= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} & c) \quad Z &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \\ d) \quad H &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & e) \quad V &= \frac{(1+i)}{2} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix} & f) \quad \text{Phase} &= \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \end{aligned} \quad (17)$$

Each matrix of an Operator has its inputs from the top (from left to right) and the outputs on the side (from top to bottom). Thus taking a state $|\psi\rangle$ (eq.18) and an unitary operator H (eq. 19)

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (18)$$

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{19}$$

the result of computation is represented in equation 20.

$$H|\Psi\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{\alpha+\beta}{\sqrt{2}} \\ \frac{\alpha-\beta}{\sqrt{2}} \end{bmatrix} \tag{20}$$

$$U = \begin{matrix} & & 00 & 01 & 10 & 11 \\ & & \downarrow & \downarrow & \downarrow & \downarrow \\ \begin{matrix} 00 \leftarrow \\ 01 \leftarrow \\ 10 \leftarrow \\ 11 \leftarrow \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} & & & & \end{matrix} \tag{21}$$

Equation 21 shows the inputs (input minterms) on the top of the matrix and the output minterms on the left side. Thus for an input $|10\rangle$ (from the top) the output is $|11\rangle$ (from the side).

2.4 Multi-Qubit quantum gates

The second class of quantum gates includes the Controlled-U gates. Schematic representation of such gates can be seen in Figure 3. Gates in Figure 3a – Figure 3c represent the general structures for single-control-qubit single-qubit gate, two-control-qubit single-qubit gate, single-control-qubit two-qubit gate and two-control-qubit two-qubit gate respectively. The reason for calling these gates *Controlled* is the fact that they are based on two operations: first there is one or more control bits and second there is a unitary transformation similar to matrices from equation 17 that is controlled. For instance the Feynman gate is a Controlled-NOT gate and has two input qubits a and b as can be seen in

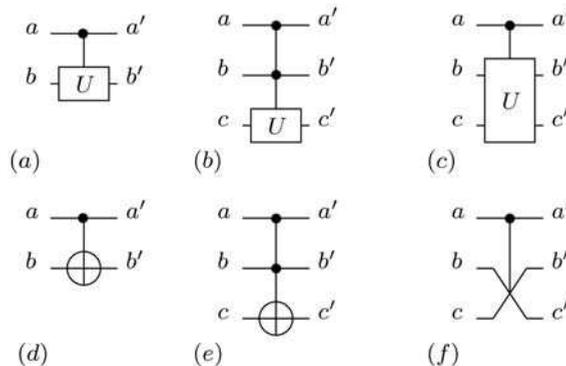


Fig. 3. Schematic representation of Controlled-U gates: a) general structure of single-qubit controlled-U gate (control qubit a, target qubit, b) two-qubit controlled, single-qubit operation, c) single-qubit controlled, two-qubit target quantum gate, d) Feynman (CNOT), e) Toffoli (CCNOT), f) Fredkin. a, b, c are input qubits and a', b' and c' are respective outputs.

Figure 3. Its unitary matrix with input and output minterms is shown in eq. (21). Thus qubits controlling the gate are called the control qubits and the qubits on which the unitary transform is applied to are called the target qubits.

Figures 3d - Figure 3f represent special cases where the controlled unitary operator is Not, Not and Swap, respectively. The respective unitary matrices are in equations 21, 22a and 22b.

Equation 21 shows that if the input state is for instance $|00\rangle$ (from the top) the output is given by $U|00\rangle = p_{00}|00\rangle = 1 \times |00\rangle$. Similarly for all other possible input /output combinations.

$$(a) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (b) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (22)$$

The *Controlled-U* gate means that while the controlled qubit a is equal to 0 the qubits on output of both wires are the same as they were before entering the gate ($a' = a$, $b' = b$). Now if qubit a equals to 1, the result is $a' = a$ and $b' = -b$ according to matrix in equation (17.a). It can be easily verified that the CCNOT (Toffoli) gate is just a Feynman gate with one more control qubit and the Fredkin gate is a controlled swap as shown on Figure 3.

A closer look at equations (21 and 22) gives more explanation about what is described in eq. 21: CNOT, eq. 22a : Toffoli and eq. 22b : Fredkin gates. For instance, equation 21 shows that while the system is in states $|00\rangle$ and $|01\rangle$ the output of the circuit is a copy of the input. For the inputs $|10\rangle$ and $|11\rangle$ the second output is inverted and it can be seen that the right-lower corner of the matrix is the NOT gate. Similarly in the other two Controlled gates the NOT gate matrix can be found.

2.5 NMR-based quantum logic gates

The NMR (Nuclear Magnetic Resonance) technology approach to quantum computing [Moo65, PW02, DKK03] is the most advanced quantum realization technology used so far, mainly because it was used to implement the Shor algorithm [Sho94] with 7 qubits [NC00]. Yet other technologies such as Ion trap [DiV95], Josephson Junction [DiV95] or cavity QED [BZ00] are being used. The NMR quantum computing has been reviewed in details in [PW02, DKK03] and for this paper it is important that it was so far the NMR computer that allowed the most advanced algorithm (7 qubit logic operation) to be practically realized and analyzed in details. Thus it is based on this technology that the constraints of the synthesis are going to be established for the cost and function evaluation. Some prior work on synthesis has been also already published [LLK+06] and few simple cost functions have been established.

For the NMR-constrained logic synthesis the conditions are:

- Single qubit operations: rotations R_x, R_y, R_z for various degrees of rotation θ . With each unitary rotation (R_x, R_y, R_z) represented in equation 23

$$\begin{aligned}
 R_x(\theta) &= e^{-i\theta X/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}X = \begin{pmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \\
 R_y(\theta) &= e^{-i\theta Y/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Y = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \\
 R_z(\theta) &= e^{-i\theta Z/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}
 \end{aligned}
 \tag{23}$$

- Two-qubit operation; depending on approach the Interaction operator is used as J_{zz} or J_{xy} for various rotations θ

Thus a quantum circuit realized in NMR will be exclusively built from single qubit rotations about three axes x, y, z and from the two-neighbor-qubit operation of interaction allowing to realize such primitives as CNOT or SWAP gates. Examples of gates realized using NMR quantum primitives are shown in Figure 5 to Figure 8.

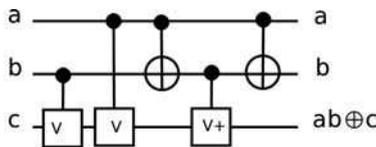


Fig. 4. Structure of the Toffoli gate

$$\text{---} \boxed{X} \text{---} = \text{---} \boxed{iR_x\left(\frac{\pi}{2}\right)} \text{---}$$

Fig. 5. Single pulse Logic gate - NOT

$$\text{---} \boxed{H} \text{---} = \text{---} \boxed{iR_x\left(\frac{\pi}{2}\right)} \boxed{R_z(\pi)} \text{---}$$

Fig. 6. Two-pulses logic gate - Hadamard

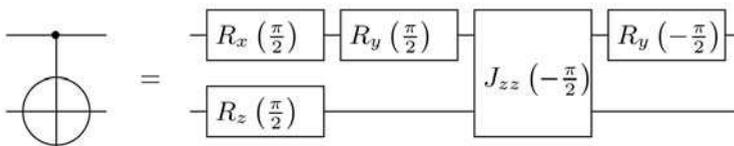


Fig. 7. Detailed Realization of Feynman Gate with five EM pulses.

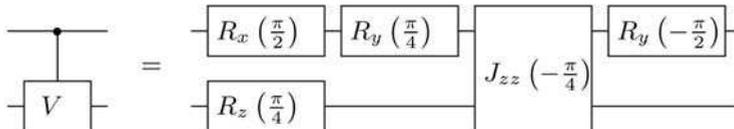


Fig. 8. Five-pulses logic gate - Controlled-V

Also, the synthesis using the NMR computing model using EM pulses, is common to other technologies such as Ion Trap [CZ95, PW02] or Josephson Junction [BZ00]. Thus the cost model used here can be applied to synthesize circuits in various technologies, all of these

technologies having the possibility to express the implemented logic as a sequence of EM pulses.

3. Quantum finite state machines

The paradigms of quantum circuits from Section 2 are applied in this paper to the synthesis of computational models such as QFSM as defined in [LPK09]. This section briefly introduces the knowledge about Quantum computational models and their properties as well as specifies the types of devices that are going to be synthesized. We describe the 1-way Quantum Finite State Machines (FSM) from both the theoretical (computational) point of view as well as from the engineering (circuit) point of view. Most of the work in this area is still on the theoretical level but the proofs of concept quantum devices [Dun98, SKT04, MC06, RCHCX+08, YCS09] allow to speculate that such models will be useful for quantum logical devices that will appear in close future.

3.1 1-way quantum finite automata

Quantum Finite State Machines (QFSM) are a natural extension of classical (probabilistic) FSM's. Two main types of QFSM are well known: One-way QFSM (1QFSM) [AF98, MC00] and two-way QFSM (2QFSM)[AW02, KW97]. As will be illustrated and explained the 1QFSM, can accept sequentially classical input, quantize it, process it and measures its quantum memory after each operation (Figure 9). In this work the focus is on the synthesis of the 1QFSM from Figure 9(b). From now on the general designation of QFSM will refer to 1QFSM in this work. Other type of described QFSMs will be specifically named.

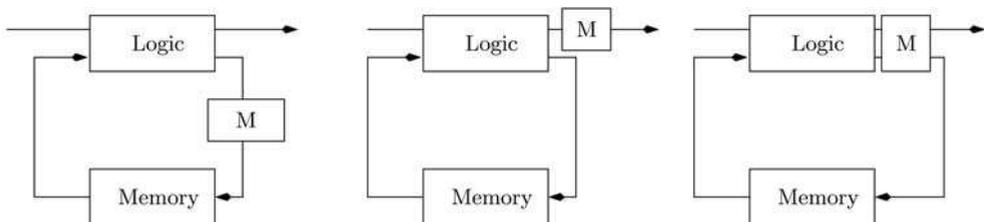


Fig. 9. Schematic representation of a 1QFSM; (a) after each computation step the machine state is measured, (b) after each computation step the output is measured, (c) after each computational step the machine state and the output state are measured.

In contrast to that, the 2QFSM is designed to operate on quantum input data (allowing to put the reading head in superposition with the input tape, and requiring all the input data to be present at once for the maximum efficiency) and the measurement is done only at the end of a whole process.

Definition 3.1

Quantum State Machine - a QFSM is a tuple $\Gamma = \{Q, \Lambda, q_0, Q_{ac}, Q_{rj}, \delta\}$, where Q is a finite set of states, σ is the input alphabet, δ is the transition function. The states $q_0 \in Q$, $Q_{ac} \subset Q$ and $Q_{rj} \subset Q$ are the initial states, the set of accepting states and the set of rejected states, respectively. \square

The QFSM machine action maps the set of machine states and the set of input symbols into the set of complex machine next states. The computation of such machine is required to be

done using unitary operators and is performed on the basis set B^q using unitary operators U_θ , $\theta \in \Theta$. In particular the QFSM uses a set of Unitary Operators corresponding to the input of input characters on the input tape. Thus for a given string to be processed and prior to the whole process termination (string either accepted or rejected), the overall processing can be represented as:

$$MU_{\theta_n}MU_{\theta_{n-1}}MU_{\theta_{n-2}} \dots MU_{\theta_3}MU_{\theta_2}MU_{\theta_1}|q_0\rangle \tag{24}$$

with MU_{θ_n} being the application of the U_{θ_n} operator to the current state and creating the configuration $U_{\theta_n}|q\rangle$ followed by the measurement of the current state M (projecting the state into G).

The 1QFSM was proven to be less powerful or equally powerful to its classical counterpart 1FSM [Gru99, KW97] in that it can recognize the same classes of regular languages as the classical FSM can recognize.

The above described 1QFSM is also called the measure-many quantum finite automaton [KW97]. A model called measure-once quantum finite automata was also introduced and studied by Moore [MC00]. The measure-many 1QFSM is similar to the concepts of the 2QFSM. For comparison we illustrate the main differences between the 1QFSM and 2QFSM below.

Example 3.1.1 1QFSM

Let $Q = \{|q_0\rangle, |q_1\rangle\}$ be two possible states (including the accepting and rejecting states) of a single-qubit machine M and with transition functions specified by the transitions defined in eq. 25 corresponding to the state diagram in Figure 10a.

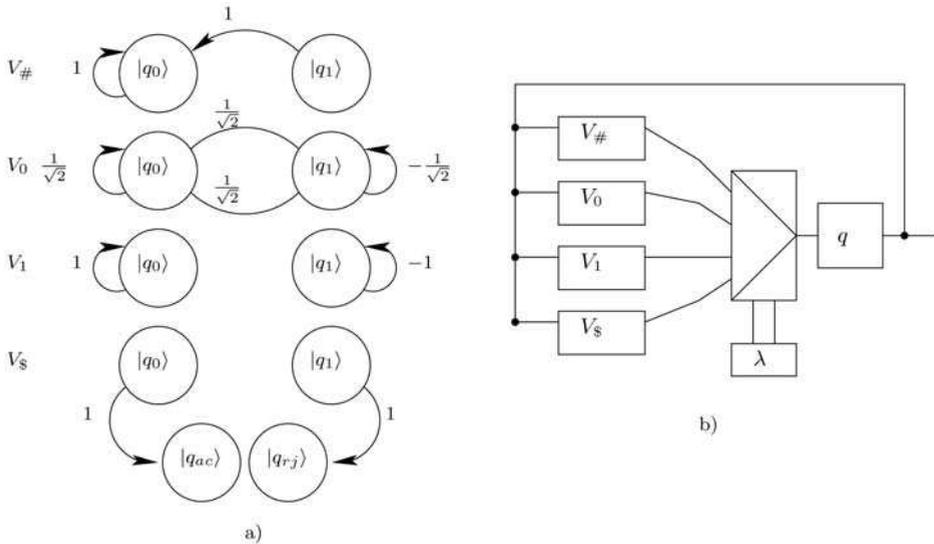


Fig. 10. (a) State transition diagram for the 1QFSM defined by the transition function 25, (b) the representation of the QFSM using quantum multiplexers. Observe two control outputs $|q\rangle$ specifying the machine action/states and the input symbols selecting the appropriate unitary transform V_λ for $\lambda \in \{\#, \$, 0, 1\}$.

$$\begin{aligned}
 V_{\#}|q_i\rangle &= |q_0\rangle & V_0|q_0\rangle &= \frac{1}{\sqrt{2}}|q_0\rangle + \frac{1}{\sqrt{2}}|q_1\rangle \\
 V_{\$}|q_0\rangle &= |q_{ac}\rangle & V_0|q_1\rangle &= \frac{1}{\sqrt{2}}|q_0\rangle - \frac{1}{\sqrt{2}}|q_1\rangle \\
 V_{\$}|q_1\rangle &= |q_{rj}\rangle & V_1|q_0\rangle &= |q_0\rangle \\
 & & V_1|q_1\rangle &= -|q_1\rangle
 \end{aligned}
 \tag{25}$$

The machine M , specified in eq. 25 represents a state machine that uses the H gate when the input is 0 ($V_0 = H$) and the Pauli-Z rotation gate when the input is 1 ($V_1 = Z$). Observe that machine M would have different behavior for measure-once and measure-many implementation. In the measure-many case, the machine generates a quantum coin-flip while receiving input 0 and while receiving input 1 the Pauli-Z rotation is applied. Observe in the measure-once case, that for example for the string input $\theta = "010"$ the many-measure machine will implement a NOT using $[H][Z][H]$. \square

Note that in this approach to QFSM each input symbol $\lambda \in \{\#, \$, 0, 1\}$ is represented by a unitary transform that can be seen as shown in Figure 10. No measurement is done here on $|q\rangle$ while the sequence of quantum operators is applied to this state. The 2QFSM operates on a similar principle as the 1QFSM model but with the main difference being the application of the measurement. This is schematically shown in Figure 11 for the completeness of explanation.

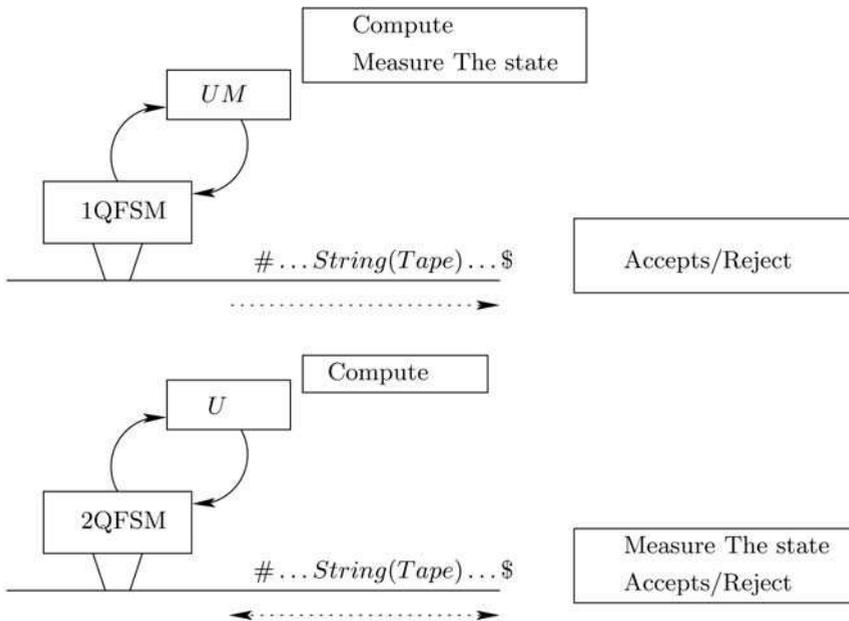


Fig. 11. Schematics representing the difference between the 1QFSM and 2QFSM. On the top, the 1QFSM - for each input character read from left to right from the tape, a unitary transform U is applied on the state and the state is measured. On the bottom, the 2QFSM moves on the input tape left and right, the unitary transform U is applied on the state and only once the computation is terminated the final state is observed/measured.

3.2 Quantum logic synthesis of sequence detectors

The problem to synthesize the QFSM is to find the simplest quantum circuit for a given set of input-output sequences thus letting the state assignment problem for this machine be directly solved by our synthesis algorithm. This direct synthesis approach can be applied to binary, multiple-valued and fuzzy quantum machines with no principle differences - only fitness functions are modified in an evolutionary algorithm [LPG+03, LP05].

Let us assume that there exists a sequential oracle that represents for instance Nature, robot control or robot's environment. In our example this oracle is specified by a state diagram in Figure 12a. This oracle can represent partial knowledge and a deterministic or probabilistic machine of any kind. Assume that there is a clearing signal (denoted by an arrow in Figure 12a) to set the oracle into its initial state. By giving initial signals and input sequences and observing output sequences the observer can create a behavior tree from Figure 12b.

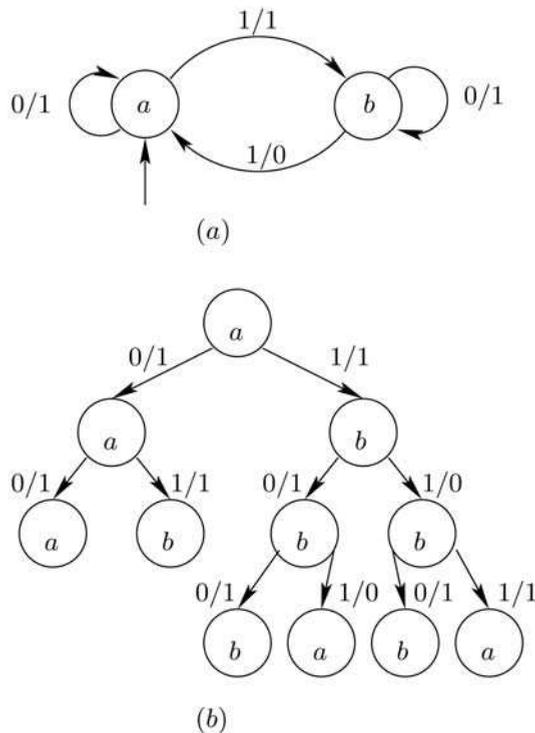


Fig. 12. Example of a deterministic oracle and its diagnostic tree.

As in general this oracle is never fully known, we perform experiments with it to determine some of its input-output behaviors. Assume that the oracle from Figure 12a is represented by the sequences from the experiments. These input-output sequences are shown in eq. 26 with $|iqo\rangle$ represents the input qubit, the state qubit and the output qubit respectively. Observe that the diagnostic tree from Figure 12(b) shows the state with $\{a, b\}$ and the inputs and the outputs as 0 and 1.

$$\begin{aligned}
 |iq0\rangle &\rightarrow |iqo\rangle \\
 000 &\rightarrow 011 \\
 001 &\rightarrow 011 \\
 100 &\rightarrow 111 \\
 101 &\rightarrow 110 \\
 110 &\rightarrow 101 \\
 111 &\rightarrow 101
 \end{aligned}
 \tag{26}$$

As the full knowledge of the oracle is in general impossible - the oracle is approximated by sets of input-output sequences and the more such sequences that we create - the more accurate characterization of the oracle as a QFSM can be created.

The overall procedure for the detection of a sequence of length j can be summarized as follows:

1. Initialize all qubits of the quantum register to the initial desired state,
2. repeat j times:
 - a. Initialize the input qubit to a desired state and set the output qubit to $|0\rangle$
 - b. Apply the quantum operator on the quantum register of the QFSM
 - c. Measure the output qubit and observe the result

Using the procedure describe above one can synthesize quantum circuits for oracles being well known universal quantum gates such as Fredkin. The input-output sequences found from this oracle are next used to synthesize the QFSM from Figure 13a. Figure 13b shows the state-diagram of the machine.

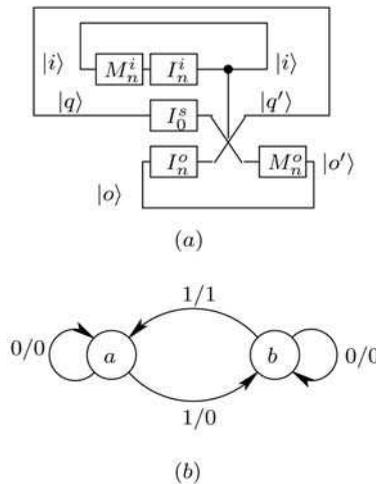


Fig. 13. Example of implementation of Fredkin gate as a quantum FSM of first class. Observe the notation where $|i\rangle$ is the input, $|q\rangle$ is the machine state and $|o\rangle$ is the machine output.

We will call the machine in Figure 13(a) the QFSM of the first class. This is because both the output and the input qubits are initialized after each computation. Observe that it is represented with feedback lines as in Figure 9 with input and output being initialized for each input and the state initialized only once - at the beginning of the computation. The interested reader can read more on this representation in [LP09], however it is important to

understand that the feedback lines are shown here only as the equivalent notation to the classical FSM as in Figure 9. The circuit-based approach to QFSM does not require this notation as this "loop" is represented by the fact that the quantum qubit preserves its state [LP09].

A set of input-output sequences defining partially the "Fredkin QFSM" is represented in eq. 27.

$$\begin{aligned}
 |iq0\rangle &\rightarrow |iqo\rangle \\
 000 &\rightarrow 000 \\
 001 &\rightarrow 000 \\
 100 &\rightarrow 100 \\
 101 &\rightarrow 100 \\
 110 &\rightarrow 101 \\
 111 &\rightarrow 101
 \end{aligned}
 \tag{27}$$

A class two QFSM has in turn the initialization I_n applied only to the input qubit. This way the generated sequence is now expressed not only as a function $f(i, q) \oplus |0\rangle$ but rather as $f(i, q) \oplus |o\rangle$. This means that now the output is directly dependent also on the previous output state. This QFSM of the second class is shown in Figure 14. The difference between the QFSM of the first and of the second class can be seen on the output qubit $|o\rangle$ where in the case of the QFSM of the first class the initialization I_n^o means the initialization of the output at each computation step while the class two QFSM uses I_0^o initializes the output only once, at the beginning of the computation.

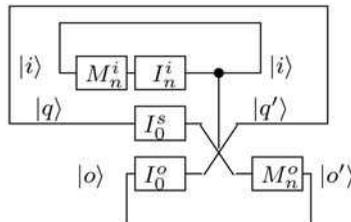


Fig. 14. Example of implementation of Fredkin gate as a quantum FSM of second class where the output is initialized only once and the measurement is done either after each input or only completely at the end.

For instance, a class two QFSM constructed from a "Fredkin oracle" differs from the class by different possible state transition. This is shown in Table 1. The first column represent the current state of the quantum register build from the input, state and output qubits $|iqo\rangle$. The second column shows the state transitions of the class one QFSM. Observe that as the output qubit is always being initialized to $|0\rangle$ only four possible initial states exists (see eq. 27). The third column representing the state transitions of the class two QFSM and as can be seen in this case the state transition function is the full "Fredkin oracle" function.

Moreover, the difference between the first and the second class of these QFSM's has also deeper implications. Observe that the QFSM presented in this paper, if implemented without the measurement on the output and the input qubit (the measurement is executed only after l computational steps) the QFSM becomes the well-known two-way QFSM

PS	NS_{one}	NS_{two}
$ iqo\rangle$	$ iqo\rangle$	$ iqo\rangle$
000	000	000
001	—	001
010	010	010
011	—	011
100	100	100
101	—	110
110	101	101
111	—	111

Table 1. Comparison of the state transition between the class one and class two QFSMs

[KW97] because the machine can be in superposition with the input and the output. This is equivalent to stating that the reading head of a QFSM is in superposition with the input tape as required for the time-quadratic recognition of the $\{a^n b^n\}$ language [KW97].

Observe that to represent the 1-way and the 2-way QFSM in the circuit notation the main difference is in the missing measurement operations between the application of the different CU (Controlled-U) operations. This is represented in Figures 15 and 16 for 1-way and the 2-way QFSMs, respectively.

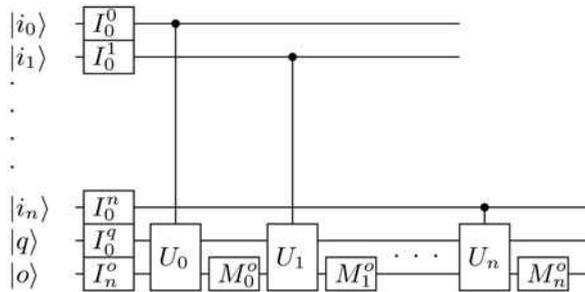


Fig. 15. Example of circuit implementing 1-way QFSM.

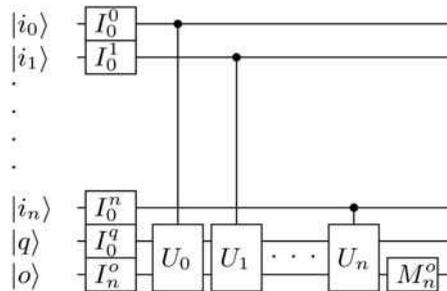


Fig. 16. Example of circuit implementing 2-way QFSM.

An interesting example of QFSM is a machine with quantum controls signals. For instance a circuit with the input qubit in the superposition generating the EPR quantum state [NC00] is shown in Figure 17.

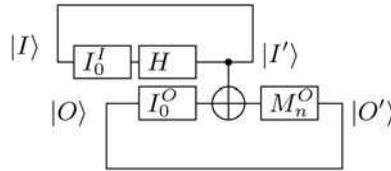


Fig. 17. Example of the EPR circuit used as a QFSM.

Observe the behavior of this QFSM as both class one and class two machine given in Table 2. In this case the distinction between the class one and class two machines is negligible because any measurement of the system collapses the whole system as the result of the entanglement present in it.

PS	NS_{one}	NS_{two}
$ iqo\rangle$	$ iqo\rangle$	$ iqo\rangle$
00	00+11	00+11
01	-	01+10
11	-	00+11
10	01+10	01+10

Table 2. Comparison of the state transition between the class one and class two EPR circuit QFSM

Figure 17 shows that because of the entanglement this machine has two distinct possible recognizable sequences. When the machine uses exclusively the output qubit initialized to $|0\rangle$ the possible initial states are only $|00\rangle$ and $|10\rangle$ because the measurement of the output state resulting in $|11\rangle \xrightarrow{I''_n} |10\rangle$ and $|01\rangle \xrightarrow{I''_n} |00\rangle$.

4. Evolutionary algorithms and quantum logic synthesis

In general the evolutionary problem solving can be split into two main categories; not separated by the methods that each of the trends are using but rather by the problem representation and by the type of problem solved. On one hand, there is the Genetic Algorithm (GA) [Gol89, GKD89] and Evolutionary strategies (ES) [Bey01, Sch95] that in general represents the information by strings of characters/integers/floats and in general attempts to solve combinatorial problems. On the other hand the design of algorithms as well as state machines was traditionally done by the Genetic Programming (GP) [Koz94, KBA99] and the Evolutionary Programming (EP) [FOW66, ES03].

Each of this approaches has its particular advantages and each of them has been already more or less successfully applied to the Quantum Logic synthesis. In the EQLS field the main body of research was done using the Genetic Programming (GP) for the synthesis of either quantum algorithms and programs [WG98, Spe04, Lei04, MCS04] or some specific types of quantum circuits [WG98, Rub01, SBS05, SBS08, LB04, MCS05]. While the GP approach has been quite active area of research the Genetic Algorithm approach is less popular and recently only [LP08, YI00] were using a Genetic Algorithm for the synthesis of quantum circuits. However, it was shown in [LP09] that it is also possible to synthesize quantum finite state machines specified as quantum circuit using a GA. The difference between the popularity of the usage between the GP and the GA for EQLS is mainly due to

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

