To Professor G. Rozenberg

for His 60th Birthday

Automata: From Uncertainty to Quantum

Cristian S. Calude and Elena Calude

Department of Computer Science, University of Auckland, New Zealand
Institute of Information Sciences, Massey University at Albany, New Zealand
cristian@cs.auckland.ac.nz, e.calude@massey.ac.nz

Abstract. Automata are simple mathematical objects with unexpected computational, mathematical, modelling and explanatory capabilities. This paper examines some relations between automata and physics. Automata will be used to model quantum uncertainty and quantum computation. Finally, mathematical proofs will be discussed from the perspective of quantum automata.

1 Modelling Quantum Uncertainty with Automata

1.1 Moore Automata

All automata we are going to consider are *finite* in the sense that they have a finite number of states, a finite number of input symbols, and a finite number of output symbols. The deterministic or non-deterministic behaviour of such a machine will be contextually clear.

First we will look at *deterministic automata* each of which consists of a finite set S_A of states, an input alphabet Σ, and a transition function $\delta_A : S_A \times \Sigma \to S_A$. Sometimes a fixed state, say 1, is considered to be the *initial state*, and a subset of S_A denotes the *final states*. A *Moore automaton* is a deterministic automaton having an *output function* $F_A : S_A \to O$, where O is a finite set of output symbols. At each time the automaton is in a given state q and is continuously emitting the output $F_A(q)$. The automaton remains in state q until it receives an input signal σ, when it assumes the state $\delta(q, \sigma)$ and starts emitting $F_A(\delta_A(q, \sigma))$. In what follows $\Sigma = \{0, 1\}$ having $O = \Sigma$, so, from now on, a Moore automaton will be just a triple $A = (S_A, \delta_A, F_A)$.

Let Σ^* be the set of all finite sequences (words) over the alphabet Σ, including the empty word e. The transition function δ can be extended to a function $\overline{\delta}_A : S_A \times \Sigma^* \to S_A$, as follows: $\overline{\delta}_A(q, e) = q$, for all $q \in S_A$, $\overline{\delta}_A(q, \sigma w) = \overline{\delta}_A(\delta_A(q, \sigma), w)$, for all $q \in S_A$, $\sigma \in \Sigma, w \in \Sigma^*$.

The output produced by an experiment started in state q with input $w \in \Sigma^*$ is described by the *total response* of the automaton A, given by the function $R_A : S_A \times \Sigma^* \to \Sigma^*$ defined by $R_A(q, e) = f(q)$, $R_A(q, \sigma w) = f(q)R_A(\delta(q, \sigma), w), q \in S_A, \sigma \in \Sigma, w \in \Sigma^*$, and the output function f.

W. Kuich, G. Rozenberg, and A. Salomaa (Eds.): DLT 2001, LNCS 2295, pp. 1–14, 2002.

1.2 Moore's Uncertainty Revisited

Moore [38] has studied some experiments on deterministic automata trying to understand what kind of conclusions about the internal conditions of a machine it is possible to draw from input-output experiments. To emphasize the conceptual nature of his experiments, Moore has borrowed from physics the word "Gedanken".

A (simple) Moore experiment can be described as follows: a copy of a deterministic machine will be experimentally observed, i.e. the experimenter will input a finite sequence of input symbols to the machine and will observe the sequence of output symbols. The correspondence between input and output symbols depends on the particular chosen machine and on its initial state. The experimenter will study sequences of input and output symbols and will try to conclude that "the machine being experimented on was in state q at the beginning of the experiment".[1] Moore's experiments have been studied from a mathematical point of view by various researchers, notably by Ginsburg [27], Gill [26], Chaitin [17], Conway [20], Brauer [6], Salomaa [42].

Following Moore [38] we shall say that a state q is "indistinguishable" from a state q' (with respect to Moore's automaton $A = (S_A, \delta_A, F_A)$) if every experiment performed on A starting in state q produces the same outcome as it would starting in state q'. Formally, $R_A(q, x) = R_A(q', x)$, for all words $x \in \Sigma^+$. An equivalent way to express the indistinguishability of the states q and q' is to require, following Conway [20], that for all $w \in \Sigma^*$, $F_A(\overline{\delta}_A(q, w)) = F_A(\overline{\delta}_A(q', w))$.

A pair of states will be said to be "distinguishable" if they are not "indistinguishable", i.e. if there exists a string $x \in \Sigma^+$, such that $R_A(q, x) \neq R_A(q', x)$.

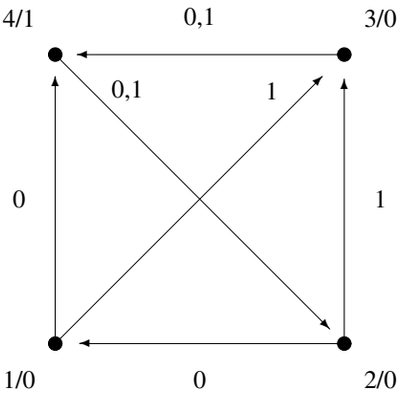

Fig. 1.

Moore [38] has proven the following important theorem: *There exists a Moore automaton A such that any pair of its distinct states are distinguishable, but there is no*

[1] This is often referred to as a *state identification experiment.*

experiment which can determine what state the machine was in at the beginning of the experiment. He used the automaton displayed in Figure 1 and the argument is simple. Indeed, each pair of distinct states can be distinguished by an experiment; however, there is no (unique) experiment capable to distinguish between every pair of arbitrary distinct states. If the *experiment starts with* 1, then x cannot distinguish between the states $1, 2$ and if the *experiment starts with* 0, then x cannot distinguish between the states $1, 3$.

Moore's theorem can be thought of as being a *discrete analogue* of the Heisenberg uncertainty principle. The state of an electron E is considered specified if both its velocity and its position are known. Experiments can be performed with the aim of answering either of the following:

1. What was the position of E at the beginning of the experiment?
2. What was the velocity of E at the beginning of the experiment?

For a Moore automaton, experiments can be performed with the aim of answering either of the following:

1. Was the automaton in state 1 at the beginning of the experiment?
2. Was the automaton in state 2 at the beginning of the experiment?

In either case, performing the experiment to answer question 1 changes the state of the system, so that the answer to question 2 cannot be obtained. This means that it is only possible to gain partial information about the previous history of the system, since performing experiments causes the system to "forget" about its past.

An exact quantum mechanical analogue has been given by Foulis and Randall [24, Example III]: Consider a device which, from time to time, emits a particle and projects it along a linear scale. We perform two experiments. In experiment α, the observer determines if there is a particle present. If there is not, the observer records the outcome of α as the outcome $\{4\}$. If there is, the observer measures its position coordinate x. If $x \geq 1$, the observer records the outcome $\{2\}$, otherwise $\{3\}$. A similar procedure applies for experiment β: If there is no particle, the observer records the outcome of β as $\{4\}$. If there is, the observer measures the x-component p_x of the particle's momentum. If $p_x \geq 1$, the observer records the outcome $\{1, 2\}$, otherwise the outcome $\{1, 3\}$. Still another quantum mechanical analogue has been proposed by Giuntini [28]. A pseudo-classical analogue has been proposed by Cohen [19] and by Wright [44].

Moore's automaton is a simple model featuring an "uncertainty principle" (cf. Conway [20]), later termed "computational complementarity" by Finkelstein and Finkelstein [23].

It would be misleading to assume that any automaton state corresponds to a *bona fide* element of physical reality (though, perhaps, hidden). Because, whereas in models of automaton complementarity it might still be possible to pretend that initially the automaton *actually is in a single automaton state*, which we just do not know (such a state can be seen if the automaton is "screwed open"), quantum mechanically this assumption leads to a Kochen-Specker contradiction [32,43].

Two non-equivalent concepts of computational complementarity based on automata have been proposed and studied in Calude, Calude, Svozil and Yu [15]. Informally, they can be expressed as follows. Consider the class of all elements of reality (or "properties", and "observables") and consider the following properties.

A Any two distinct elements of reality can be mutually distinguished by a suitably chosen measurement procedure, Bridgman [7].

B For any element of reality, there exists a measurement which distinguishes between this element and all the others. That is, a distinction between any one of them and all the others is operational.

C There exists a measurement which distinguishes between any two elements of reality. That is, a single pre-defined experiment operationally exists to distinguish between an arbitrary pair of elements of reality. (Classical case.)

It is easy to see that there exist automata with property **C**. More interestingly, there exist automata which have *CI* that is **A** but not **B** (and therefore not **C**) as well as automata with *CII*, i.e. **B** but not **C**. Properties *CI, CII* are called *complementarity principles*. Moore's automaton in Figure 1 has indeed *CI* . To get *CII* we can use again Moore's automaton but with different output functions, for example those in Figure 2:

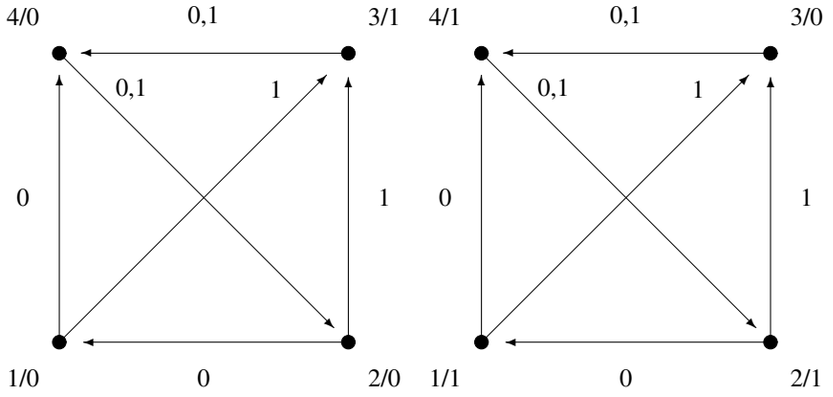

Fig. 2.

According to the philosophical view called realism, *reality* exists and has definite properties irrespective whether they are observed by some agent. Motivated by this view point, Einstein, Podolsky and Rosen [22] suggested a classical argument showing that quantum mechanics is incomplete. EPR assumed a) the non-existence of action-at-a-distance, b) that some of the statistical predictions of quantum mechanics are correct, and c) a reasonable criterion defining the existence of an element of physical reality. They considered a system of two spatially separated but quantum mechanically correlated particles. A "mysterious" feature appears: By counterfactual reasoning, quantum mechanical experiments yield outcomes which cannot be predicted by quantum theory; hence the quantum mechanical description of the system is incomplete!

One possibility to complete the quantum mechanical description is to postulate additional "hidden-variables" in the hope that completeness, determinism and causality will

be thus restored. But then, another conundrum occurs: Using basically the same postulates as those of EPR, Bell [4,5] showed that no deterministic local hidden-variables theory can reproduce all statistical predictions of quantum mechanics. Essentially, the particles on either side appear to be "more correlated" than can be expected by a classical analysis assuming locality (i.e., the impossibility of any kind of information or correlation transfer faster than light).

The complementarity *CII* mimics, in a sense, the state of *quantum entanglement* and may be conceived as a *toy model* for the *EPR effect*, cf. Greenberger, Horne and Zeilinger [29]. Being experimentally testable, *CII* falls into the class of *puzzle mysteries* (see Penrose [40]). For a probabilistic approach see [14,11]; for complementarity for Mealy automata see [16].

2 Simulation, Bisimulation, Minimization

The complementarity principles discussed above suggest that the classical theory of finite automata–which considers automata with initial states–is not adequate for modeling physical phenomena, hence the need to look at automata *without initial states*. We will first study deterministic automata, then nondeterministic automata, and finally a comparison between these two types of automata will be presented. Various types of *simulations* will play a central role. This section is based on Calude, Calude and Khoussainov [12, 13] and [10].

We have already discussed the total response of an automaton A. The *final response* of A is the function $f_A : S_A \times \Sigma^\star \to \Sigma$ defined, for all $s \in S_A$ and $w \in \Sigma^\star$, by $f_A(s, w) = F_A(\delta_A(s, w))$. The *initial response* of A is the function $i_A : S_A \times \Sigma^\star \to \Sigma$ defined, for all $s \in S_A$ and $w \in \Sigma^\star$, by $i_A(s, w) = F_A(s)$.

Informally, an automaton A is *strongly simulated by* B if B can perform all computations of B *exactly in the same way*. We say that A and B are *strongly equivalent* if they strongly simulate each other. Intuitively, a strong simulation has to take into account the "internal machinery" of the automaton, not only the outputs. Let $A = (S_A, \delta_A, F_A)$ and $B = (S_B, \delta_B, F_B)$ be automata. We say that

1. A is *strongly simulated by* B, if there is a mapping $h : S_A \to S_B$ such that (a) for all $s \in S_A$ and $\sigma \in \Sigma$, $h(\delta_A(s, \sigma)) = \delta_B(h(s), \sigma)$, and (b) for all $s \in S_A$ and $w \in \Sigma^\star$, $R_A(s, w) = R_B(h(s), w)$.
2. A is *strongly f-simulated* (*i–simulated*) by B, or, equivalently, B *strongly f-simulates* (*i–simulates*) A if there is a mapping $h : S_A \to S_B$ such that (a) for all $s \in S_A$ and $\sigma \in \Sigma$, $h(\delta_A(s, \sigma)) = \delta_B(h(s), \sigma)$, and (b) for all $s \in S_A$ and $w \in \Sigma^\star$, $f_A(s, w) = f_B(h(s), w)$ $(i_A(s, w) = i_B(h(s), w))$.

Clearly, the strong simulation implies both strong f as well as strong i–simulations. In fact, *all these three notions are equivalent.*

From an algebraic point of view, strong simulations are morphisms between automata; they make essential use of the internal machinery of automata. The behavioral simulation, which is weaker than strong simulation (it makes use only of outputs produced by automata) turns to be more important.

Let $A = (S_A, \delta_A, F_A)$ and $B = (S_B, \delta_B, F_B)$ be two automata. We say that A is *simulated by* B if there is a mapping $h : S_A \to S_B$ such that for all $s \in S_A$ and $w \in \Sigma^\star$, $R_A(s, w) = R_B(h(s), w)$. We say that A is *f-simulated* (*i–simulated*) by B if there is a mapping $h : S_A \to S_B$ such that for all $s \in S_A$ and $w \in \Sigma^\star$, $f_A(s, w) = f_B(h(s), w)$ ($i_A(s, w) = i_B(h(s), w)$). *An automaton A is simulated by B iff A can be f–simulated by B.* A counter-example showing that i–simulation is not equivalent to simulation can be found easily as $i_A(s, w) = F_A(s)$, for all $s \in S_A$ and $w \in \Sigma^*$.

Suppose that we have a finite class $\mathcal{C}$ containing pairs (A_i, q_i) of automata $A_i = (S_i, \delta_i, F_i)$ and initial states $q_i \in S_i$, $i = 1, \ldots, n$. An automaton $A = (S_A, \delta_A, F_A)$ is *universal* for the class $\mathcal{C}$ if (a) for any $1 \leq i \leq n$ there is a state $s \in S_A$ such that $R_A(s, w) = R_{A_i}(q_i, w)$, for all $w \in \Sigma^\star$, and (b) for any $s \in S_A$ there is an i such that $R_A(s, w) = R_{A_i}(q_i, w)$, for all $w \in \Sigma^\star$. Every finite class which possesses a universal automaton is said to be *complete*. Not every finite class of automata with initial states has a universal automaton. However, *for every finite class of pairs of automata and initial states $\mathcal{C}'$, there is a complete class $\mathcal{C}$ containing $\mathcal{C}'$.* More, *the automata A and B simulate each other iff A and B are universal for the same class.*

Two states p and q are R_A–equivalent if for all $w \in \Sigma^\star$, $R_A(p, w) = R_A(q, w)$; by $[s]$ we denote the equivalence class of s. To an automaton A we associate the automaton $M(A)$ as follows:

(a) The set of states of $M(A)$ is $S_{M(A)} = \{[s] | s \in S_A\}$.
(b) For all $[s]$ and $\sigma \in \Sigma$, put $\delta_{M(A)}([s], \sigma) = [\delta_A(s, \sigma)]$.
(c) For all $[s]$, put $F_{M(A)}([s]) = F_A(s)$.

The automata $M(A)$ and $M(M(A))$ are isomorphic and they simulate each other. Furthermore, $M(A)$ is minimal and if B and A simulate each other and B is minimal, then $M(A)$ and B are isomorphic. Hence, *any complete class has a minimal universal automaton which is unique up to an isomorphism, and any two minimal automata which simulate each other are isomorphic.*

We note that (i) a minimal automaton can be characterized by Moore's condition **A**, (ii) from $M(A)$ one can immediately deduce the classical minimal automaton (but the converse is not true), (iii) for strongly connected automata indistinguishability coincides with simulation and the resulting minimal automata are isomorphic.[2]

A *nondeterministic automaton* over Σ is a triple $A = (S_A, \nabla_A, F_A)$, where S_A and F_A are as in the definition of a deterministic automaton, but ∇_A is a function from $S_A \times \Sigma$ to the set 2^{S_A} of all subsets of S_A. Again, there are several ways to introduce the notion of "response" of A to an input sequence of signals. Take $w = \sigma_1 \ldots \sigma_n \in \Sigma^\star$ and $s_0 \in S_A$. A *trajectory* of A on s_0 and w is a sequence $s_0, s_1, \ldots, s_n$ of states such that $s_{i+1} \in \nabla_A(s_i, \sigma_{i+1})$ for all $0 \leq i \leq n-1$. A trajectory $s_0, s_1, \ldots, s_n$ *emits* the output $F_A(s_0)F_A(s_1) \cdots F_A(s_n)$.

The *total response*, denoted by R_A, is a function which to any $(s, w) \in S_A \times \Sigma^\star$ assigns the set $R_A(s, w)$ of all outputs emitted by all trajectories of A on s and w. The *final response* of A is a function f_A which to any pair $(s, w) \in S_A \times \Sigma^\star$ assigns the subset of all last symbols occurring in words in $R_A(s, w)$.

[2] However, Moore's procedure cannot be used to construct a minimal automaton indistinguishable from a given not strongly connected automaton.

Let A and B be two, not necessarily distinct, nondeterministic automata. Take states $p \in S_A$ and $q \in S_B$, and fix a positive integer $n \geq 1$. We define a game $\Gamma(p, q, n)$ between two players: Player 0 and Player 1. Player 0 tries to prove that outputs emitted by trajectories which begin in p are different from outputs emitted by trajectories originated in q. Player 1 tries to show the opposite. Note that Player 0 (Player 1) is *not* restricted to consider computations which begin from p (q) only. Player 0 (Player 1) is allowed to pick up any instance of a computation which begins from q (p) as well.

Here is a description of a **play**. Every play has at most n stages. Each stage begins with a move of Player 0 and ends with a response of Player 1.

Stage 0. Player 0 picks up either p or q. Player 1 responds by picking up the other state.

Stage $k+1 \leq n$. At the end of stage k we have two sequences

$$p_0 p_1 \ldots p_k \quad \text{and} \quad q_0 q_1 \ldots q_k$$

where $p_0 = p$ and $q_0 = q$. Now Player 0 chooses a state either from $\bigcup_{\sigma \in \Sigma} \nabla_A(p_k, \sigma)$ or from $\bigcup_{\sigma \in \Sigma} \nabla_B(q_k, \sigma)$. If Player 0 chooses a p_{k+1} from $\bigcup_{\sigma \in \Sigma} \nabla_A(p_k, \sigma)$, then Player 1 responds by choosing a state q_{k+1} from $\bigcup_{\sigma \in \Sigma} \nabla_B(q_k, \sigma)$. If Player 0 chooses a q_{k+1} from $\bigcup_{\sigma \in \Sigma} \nabla_A(q_k, \sigma)$, then Player 1 responds by choosing a state p_{k+1} from $\bigcup_{\sigma \in \Sigma} \nabla_B(p_k, \sigma)$. This ends a description of stage $k+1$ of a play.

Let

$$p_0 p_1 \ldots p_t, \quad \text{and} \quad q_0 q_1 \ldots q_t$$

be sequences produced during a play. We say that Player 1 **wins** the play if for all $0 < i \leq t$, $\sigma \in \Sigma$, we have $p_i \in \nabla_A(p_{i-1}, \sigma)$ iff $q_i \in \nabla_B(q_{i-1}, \sigma)$ and $F_A(p_i) = F_B(q_i)$.

Finally, we say that that p is **$\equiv$–equivalent** to q if Player 1 wins the game $\Gamma(p, q, n)$, for all positive integers n.

The automaton A *is simulated* by the automaton B if there is a mapping $h : S_A \to S_B$ such that for all $s \in S_A$, the states s and $h(s)$ are $\equiv$–equivalent. We denote this fact by $A \leq B$. The simulation relation defined above coincides with the simulations of deterministic automata, in case A and B are deterministic.

Let A be a nondeterministic automaton. We define the automaton $M(A)$ as follows:

1. The set of states $S_{M(A)}$ of $M(A)$ is $\{[s] \mid s \in S_A\}$, where $[s] = \{q \in S_A \mid s \equiv q\}$.
2. For all $[q], [s] \in S_{M(A)}$ and $\sigma \in \Sigma$, $[q] \in \nabla_{M(A)}([s], \sigma)$ iff $q \in \nabla_A(s, \sigma)$.
3. $F_{M(A)}([s]) = F_A(s)$.

An analogue result holds true for nondeterministic automata: *The automata A and $M(A)$ simulate each other, the automaton $M(A)$ is minimal and unique up to an isomorphism.*

The equivalence used for $M(A)$ is constructed in terms of a special *game*. The minimal automaton can be equally constructed using a specific bisimulation (see [10]), i.e. a non-empty relation $\asymp \subset S_A \times S_B$ satisfying the following two conditions for all $p \asymp q$ ($p \in S_A, q \in S_B$):

1. $\nabla_A(p, \sigma) \asymp \nabla_B(q, \sigma)$, for all $\sigma \in \Sigma$,
2. $F_A(p) = F_B(q)$.

More precisely, for A we consider the greatest bisimulation $\asymp_A$ of $\Xi(A, A)$, the set of all bisimulations from $S_A \times S_A$.

The subset construction shows that from the point of view of recognized languages, deterministic automata are as powerful as the nondeterministic ones (see [31,42,34]). It is not difficult to see that if A and B are bisimilar automata, then the deterministic automata obtained from A and B by the subset construction are also bisimilar. However, *there exist infinitely many nondeterministic (strongly connected) automata each of which is not bisimilar with any deterministic automaton.*

The compatibility between the bisimulation approach and the simulation approach for deterministic automata follows from the following result: *Let A and B be deterministic automata and $h : S_A \to S_B$ a function. Then, the following statements are equivalent: the function h is a morphism iff the graph of h is a bisimulation iff the automaton A is strongly simulated by the automaton B via h.*

3 Quantum Automata

There are three basic theoretical models designed to study the power and limitations of quantum computing: quantum finite automata (QFA), quantum Turing machines and quantum cellular automata. All these quantum models are obtained from their classical probabilistic counterparts by applying the following changes:

- ◇ probabilities of transitions are substituted by probabilities amplitudes,
- ◇ each computation takes place in the inner-product space over the set of finite configurations;
- ◇ each computation is unitary.

Like classical automata, QFA have a finite set of states, a finite input alphabet and a transition function that specifies how the automaton's state changes. QFA are different from their classical counterparts in that they can be in a superposition of states that are required to have unit norm. On reading an input, a quantum finite automaton changes its superposition of states preserving the unit norm. Measurements, given by an observable, can be applied in order to determine the automaton's current state. When an observable is applied to a state, that state changes probabilistically to its projection onto one of the subspaces. The probability depends on the amplitudes.

In what follows we will discuss three models of QFA: measure-once QFA ($MO-QFA$), measure-many QFA ($MM-QFA$) and ancilla QFA.

3.1 Measure-Once Quantum Automata

The $MO-QFA$, introduced by Moore and Crutchfield [37], were inspired by stochastic automata of Rabin [41] and real-time dynamical recognizers, see Moore [35]. We will use the equivalent definition given in Brodsky and Pippenger [8].

An $MO-QFA$ is a 5-tuple $M = (S, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is the finite input alphabet with an end-marker symbol \$, $\delta : S \times \Sigma \times Q \to \mathcal{C}$ ($\mathcal{C}$ is the set of complex numbers and $\overline{\alpha}$ is the conjugate of α) is the transition function ($\delta(q, \sigma, q')$ represents the probability density amplitude that flows from state q to state

q' upon reading symbol σ), the state q_0 is the initial configuration of the system, and F is the set of accepting states. For all states $q_1, q_2 \in Q$ and symbol $\sigma \in \Sigma$ the function δ must be unitary, thus satisfying the condition:

$$\sum_{q' \in S} \overline{\delta(q_1, \sigma, q')} \delta(q_2, \sigma, q') = \begin{cases} 1, & \text{if } q_1 = q_2, \\ 0, & \text{otherwise.} \end{cases}$$

The end-marker \$ is assumed to be the last symbol of each input and is the last symbol read before the computation terminates. At the end of a computation M measures its configuration; if it is in an accepting state then it accepts the input, otherwise it rejects. The configuration of M is a superposition of states and it is represented by an n-dimensional complex unit vector, where n is the number of states. This vector is denoted by $|\psi\rangle = \Sigma_{i=1}^{n} \alpha_i |q_i\rangle$, where $\{|q_i\rangle\}$ is an orthonormal basis corresponding to the states of M. The coefficient α_i is the probability density amplitude of M being in state q_i. Since $|\psi\rangle$ is a unit vector, it follows that $\Sigma_{i=1}^{n} |\alpha_i|^2 = 1$. The transition function δ is represented by a set of unitary matrices $U_\sigma, \sigma \in \Sigma$, where U_σ represents the unitary transitions of M upon reading σ. If M is in configuration $|\psi\rangle$ and reads symbol σ, then the new configuration of M is

$$|\psi'\rangle = U_\sigma |\psi\rangle = \textstyle\sum_{q_i, q_j \in S} \alpha_i \delta(q_i, \sigma, q_j) |q_j\rangle.$$

A measurement is represented by a diagonal zero-one projection matrix P where p_{ii} is 1 or 0 depending whether $q_i \in F$. The probability of M accepting string x is

$$p_M(x) = \langle \psi_x | P | \psi_x \rangle = ||P|\psi_x\rangle||^2,$$

where $|\psi_x\rangle = U(x)|q_0\rangle = U_{x_n} U_{x_{n-1}} \dots U_{x_1} |q_0\rangle$.

Physically, this can be interpreted as follows. We have a quantum system prepared in a superposition of initial states. We expose it over time to a sequence of input symbols, one time-step per symbol. At the end of this process, we perform a measurement on the system and $p_M(x)$ is the probability of this measurement having an accepting outcome. Note that p_M is the probability of a particular event, not a general measure (on a space coded by strings).

The power of $MO - QFA$ depends on the type of acceptance, i.e. accept with bounded/unbounded-error probability. A language L is accepted with bounded-error probability by an $MO - QFA$ if there exists an $\varepsilon > 0$ such that every string in L is accepted with probability at least $\frac{1}{2} + \varepsilon$ and every string not in L is rejected with probability at least $\frac{1}{2} + \varepsilon$. The language L is accepted with unbounded-error probability by an $MO - QFA$ if every string in L is accepted with probability at least $\frac{1}{2}$ and every string not in L is rejected with probability at least $\frac{1}{2}$.

The main results are due to Brodsky and Pippenger [8]:

1. *The class of languages accepted by $MO - QFA$ with bounded-error probability coincides with the class of group languages, a proper subset of regular languages.*[3]

[3] A group automaton (GFA) is a DFA such that for every state q and input symbol σ, there exists exactly one state q' such that $\delta(q', \sigma) = q$. Equivalently, a DFA is reversible if for every $\sigma \in \Sigma$ there exists a string $x \in \Sigma^*$ such that for every state q, $\delta(q, \sigma x) = q$; see Bavel and Muller [3].

2. *Any language accepted by an $MO-QFA$ with bounded-error probability can also be accepted by a deterministic probabilistic automaton with bounded-error probability.*
3. *Some $MO-QFA$ with unbounded-error probability accept non-regular languages, for example, the language $\{x \in \{0,1\}^* \mid x$ has an equal number of 0's and 1's$\}$.*

3.2 Measure-Many Quantum Automata

For bounded-error acceptance, the power of $MO-QFA$ is too limited. One way of adding power to QFA is by introducing intermediate measurements. However, doing a measurement that causes the superposition to collapse to a single state would turn the QFA into a probabilistic automaton. A possible solution is to partition the set of states in three subsets–the accepting, rejecting and non-halting states–and use the spans of these sets as observables. A measurement is performed after every transition.

Inspired by the classical one-tape deterministic automata two types of $MM-QFA$, namely 1-way QFA ($1QFA$) and 2-way QFA ($2QFA$), have been introduced by Kondacs and Watrous [33].

An one-way measure-many quantum automaton ($1QFA$) is a 6 tuple $M = (S, \Sigma, q_0, S_a, S_r, \delta)$, where Σ is the finite alphabet with two end-marker symbols #, \$, S is the finite set of states, q_0 is the initial state, $S_a \subseteq S$ is the set of accepting states, $S_r \subseteq S$ is the set of rejecting states, $S_a \cap S_r = \emptyset$. The transition function δ is given by: $\delta : S \times \Sigma \times S \to \mathcal{C}$.

The computation of M is performed in the inner-product space $l_2(S)$, i.e. with the basis $\{|q\rangle \mid q \in S\}$, using the unary linear operators $V_\sigma, \sigma \in \Sigma$, defined by $V_\sigma(|q\rangle) = \sum_{q' \in S} \delta(q, \sigma, q')|q'\rangle$.

1. *Any language recognized by an $1QFA$ with bounded-error probability is regular,* cf. Kondacs and Watrous [33].
2. *All group languages (i.e., languages recognized by group automata) are recognized by $1QFA$* (cf. Brodsky and Pippenger [8]), *but not all regular languages are recognized by $1QFA$; for example, the language $\{ab\}^*a$ cannot be recognized by $1QFA$ with bounded-error probability,* cf. Kondacs and Watrous [33].
3. *An $1QFA$ can accept a language with probability higher than $\frac{7}{9}$ iff the language is accepted by a deterministic reversible automata,*[4]

The definition of two-way measure-many quantum automata ($2QFA$) is more complex, because of the effort to make their evolution unitary. The price paid is in the "size of quantum memory" which can grow proportional to the size of the input. *$2QFA$ accept with bounded-error probability all regular languages in linear time. Their capability goes beyond regular languages; for example, the non-context-free language $\{a^n b^n c^n | n \geq 0\}$ is recognized by a $2QFA$,* cf. Kondacs and Watrous [33].

[4] According to Brodsky and Pippenger [8], a DFA is reversible if for every state q and input symbol σ, there exists at most one state q' such that $\delta(q', \sigma) = q$, and if there exist distinct states q_1, q_2 such that $\delta(q_1, \sigma) = q = \delta(q_2, \sigma)$, then $\delta(q, \Sigma) = \{q\}$. This notion of reversibility is equivalent to the one used by Ambainis and Freivalds [1]. Group automata are reversible in the above sense, but the converse is false.

Ambainis and Watrous [2] have introduced a model of two-way measure-many quantum automata in which both quantum and classical states are used, but the tape head position is classical. These automata have better computational capabilities than $2QFA$. For example, *the language* $\{a^n b^n | n \geq 0\}$ *can be recognized by a two-way automata with quantum and classical states in polynomial time* (a classical probabilistic automaton recognizes it in exponential time).

3.3 Ancilla QFA

To avoid the restriction to unitary transitions (which is quite strong) ancilla qubits have been added: with them, each transition can be unitary. Formally, this is done by adding an output tape to the QFA, cf. Paschen [39].

An ancilla QFA is a 6-tuple $M = (S, \Sigma, \Omega, \delta, q_0, F)$, where S, Σ, q_0 and F are as for $MO-QFA$, Ω is the output alphabet and the transition function $\delta : S \times \Sigma \times S \times \Omega \to \mathcal{C}$ verifies the following condition: for all states $q_1, q_2 \in S$ and $\sigma \in \Sigma$

$$\sum_{q \in S, \omega \in \Omega} \overline{\delta(q_1, \sigma, q, \omega)}, \delta(q_2, \sigma, q, \omega) = \begin{cases} 1, & \text{if } q_1 = q_2, \\ 0, & \text{otherwise.} \end{cases}$$

The main result in Paschen [39] is: *For every regular language L, there is a non-negative integer k such that an ancilla QFA using k ancilla qubits can recognize L exactly. These quantum automata can recognize with one-sided unbounded error some non-regular languages.*[5]

3.4 More Comments

Several types of quantum automata (QFA) have been proposed in the literature (see more in Gruska [30]). Some of them are more powerful than their classical counterpart. Others, as we have seen, *are less powerful*. This is the first problem: in principle, any quantum computational system is a generalization of a classical counterpart, so its computational power should not be *less than that of the classical system*. What is the explanation of this anomalie?

According to Moore [36], "The only case in which quantum automata are weaker than classical ones is when they are required to be unitary throughout their evolution, i.e. when measurements are only allowed at the end. This imposes a strict kind of reversibility, and (for instance) prevents languages like $\{w \in (a+b)^* \mid w \text{ contains no } aa\}$ from being recognized by a finite-state quantum machine. If you allow measurements during the computation as well as at the end (which seems reasonable) they include all classical automata."

Ciamarra [18] suggests a different reason, namely the lack of reversibility. A quantum computation is performed through a unitary operator, which is *reversible*, so the computation performed by a quantum automaton should be reversible till measurement. However, no model of quantum automata is reversible as from the final state $U_w|q_0\rangle$ one cannot retrace the computation because w is unknown; one can compute backward

[5] An automaton M accepts a language L with one-sided unbounded error if M accepts all strings of L with certainty and rejects strings not in L with some positive probability (or vice-versa).

from the operator U_w, but this information *is not* encoded in the final state. In spite of this, the class of languages recognized accepted by $MO - QFA$ with bounded-error probability coincides with the class of group languages, that is languages recognized by reversible classical automata! Classically, reversibility can be guaranteed by introducing the so-called *garbage* which can be recycled so that it grows linearly with the input size. Quantum mechanically, recycling is forbidden as the *garbage* might be entangled with the computational system. Ciamarra [18] suggests a model of quantum automaton which is strictly reversible (modeling classical reversible automata) and has at least the power of classical automata.

Reversibility is a very important notion in both classical and quantum computing (see for example, Frank, Knight, Margolus [25]). It seems that to date we don't have a satisfactory formal definition, which may be the cause of various anomalies in quantum computing, as the one discussed above.

4 Proofs and "Quantum" Proofs

Classically, there are two equivalent ways to look at the mathematical notion of proof: a) as a finite sequence of sentences strictly obeying some axioms and inference rules, b) as a specific type of computation. Indeed, from a proof given as a sequence of sentences one can easily construct a machine producing that sequence as the result of some finite computation and, conversely, giving a machine computing a proof we can just print all sentences produced during the computation and arrange them in a sequence. This gives mathematics an immense advantage over any science: any proof is an explicit sequence of reasoning steps that can be inspected at *leisure*; *in theory*, if followed with care, such a sequence either reveals a gap or mistake, or can convince a skeptic of its conclusion, in which case the theorem *is considered proven*. We said, *in theory*, because the game of mathematical proofs is ultimately a social experience, so it is contaminated to some degree by all "social maladies".

This equivalence has stimulated the construction of programs which perform like *artificial mathematicians*.[6] From proving simple theorems of Euclidean geometry to the proof of the four-color theorem, these "theorem provers" have been very successful. Of course, this was a good reason for sparking lots of controversies (see [9]).

Artificial mathematicians are far less ingenious and subtle than human mathematicians, but they surpass their human counterparts by being infinitely more patient and diligent. What about making errors? Are human mathematicians less prone to errors? This is a difficult question which requires more attention.

If a conventional proof is replaced by a "quantum computational proof" (or a proof produced as a result of a molecular experiment), then the conversion from a computation to a sequence of sentences may be impossible, e.g., due to the size of the computation. For example, a quantum automaton could be used to create some proof that relied on quantum interference among all the computations going on in superposition. The quantum automaton would say "your conjecture is true", but there will be no way to exhibit all trajectories followed by the quantum automaton in reaching that conclusion.

[6] Other types of "reasoning" such as medical diagnosis or legal inference have been successfully modeled and implemented; see, for example, the British National Act which has been encoded in first-order logic and a machine has been used to uncover its potential logical inconsistencies.

In other words, the quantum automaton has the ability to check a proof, but it may fail to reveal a “trace" of the proof for the human being operating the quantum automaton. Even worse, any attempt to *watch* the inner working of the quantum automaton (e.g. by “looking" at any information concerning the state of the on going proof) may compromise for ever the proof itself!

These facts may not affect the essence of mathematical objects and constructions (which have an autonomous reality quite independent of the physical reality), but they seem to have an impact of how we learn/understand mathematics (which is thorough the physical world). Indeed, our glimpses of mathematics are revealed only through physical objects, human brains, silicon computers, quantum automata, etc., hence, according to Deutsch [21], they have to obey not only the axioms and the inference rules of the theory, but the *laws of physics* as well.

References

1. Ambainis, A., Freivalds, R. 1-way quantum finite automata: strengths, weaknesses and generalizations, *Proceedings of 39th IEEE FOCS* (1998), 332-341.
2. Ambainis, A., Watrous, J. Two-way finite automata with quantum and classical states, *Technical Report*, CC/9911009, 1999.
3. Bavel, Z., and Muller, D. E. Connectivity and reversibility in automata, *J. Assoc. Comput. Mach. 17* (1970), 231–240.
4. Bell, J. S. On the Einstein Podolsky Rosen paradox, *Physics*, 1 (1964), 195–200. Reprinted in [5] pp. 14–21.
5. Bell, J. S. *Speakable and Unspeakable in Quantum Mechanics*, Cambridge University Press, Cambridge, 1987.
6. Brauer, W. *Automatentheorie*, Teubner, Stuttgart, 1984.
7. Bridgman, P. W. A physicists second reaction to Mengenlehre, *Scripta Mathematica 2* (1934), 101–117, 224–234.
8. Brodsky, A., Pippenger, N. Characterisation of 1-way quantum finite automata, quant-ph/9903014, 1999.
9. Calude, A. S. The journey of the four colour theorem through time, *The New Zealand Mathematics Magazine*
   38, 3 (2001), 1-10.
10. Calude, C. S., Calude, E. Bisimulations and behaviour of nondeterministic automata, in G. Rozenberg, W. Thomas (eds.) *Developments in Language Theory. Foundations, Applications, and Perspectives*, World Scientific, Singapore, 2000, 60-70.
11. Calude, C. S., Calude, E., Chiu, T., Dumitrescu, M., and Nicolescu, R. Testing computational complementarity for Mermin automata, *J. Multi Valued Logic*, 6 (2001), 47-65.
12. Calude, C. S., Calude, E., Khoussainov, B. Deterministic automata: Simulation, universality and minimality, *Annals of Applied and Pure Logic* 90, 1-3 (1997), 263-276.
13. Calude, C. S., Calude, E., Khoussainov, B. Finite nondeterministic automata: Simulation and minimality, *Theoret. Comput. Sci.* 242, 1-2 (2000), 219–235.
14. Calude, C. S., Calude, E., Svozil, K. Computational complementarity for probabilistic automata, in C. Martin-Vide, V. Mitrana (eds.). *Where Mathematics, Computer Science, Linguistics and Biology Meet*, Kluwer, Amsterdam 2000, 99-113.
15. Calude, C. S., Calude, E., Svozil, K., and Yu, S. Physical versus computational complementarity I, *International Journal of Theoretical Physics 36* (1997), 1495–1523.
16. Calude, C. S., Calude, E., Ştefănescu, C. Computational complementarity for Mealy automata, *EATCS Bull.* 66 (1998), 139–149.

17. CHAITIN, G. J. ewblock An improvement on a theorem by E. F. Moore. *IEEE Transactions on Electronic Computers EC-14* (1965), 466–467.
18. CIAMARRA, M. P. Quantum reversibility and a new model of quantum automaton, in R. Freivalds (ed.). *The 13th International Symposium on Foundations of Computation Theory (FCT'2001)*, Riga, Latvia, Springer-Verlag, Lect. Notes Comput. Sci. 2138, 2001, 376-379.
19. COHEN, D. W. *An Introduction to Hilbert Space and Quantum Logic*, Springer, New York, 1989.
20. CONWAY, J. H. *Regular Algebra and Finite Machines*, Chapman and Hall Ltd., London, 1971.
21. DEUTSCH, D. Quantum theory, the Church-Turing principle and the universal quantum computer, *Proceedings of the Royal Society London*, A 400 (1985), 97–119.
22. EINSTEIN, A., PODOLSKY, B., AND ROSEN, N. Can quantum-mechanical description of physical reality be considered complete? *Physical Review 47* (1935), 777–780.
23. FINKELSTEIN, D., AND FINKELSTEIN, S. R. Computational complementarity, *International Journal of Theoretical Physics 22*, 8 (1983), 753–779.
24. FOULIS, D. J., AND RANDALL, C. Operational statistics. i. Basic concepts, *Journal of Mathematical Physics 13* (1972), 1667–1675.
25. FRANK, M., KNIGHT, T., MARGOLUS, N. Reversibility in optimally scalable computer architectures, In *Unconventional Models of Computation*, C. S. Calude, J. Casti, M. Dinneen, Eds., Springer-Verlag, 1998, 165–182.
26. GILL, A. State-identification experiments in finite automata, *Information and Control 4* (1961), 132–154.
27. GINSBURG, S. On the length of the smallest uniform experiment which distinguishes the terminal states of the machine, *J. Assoc. Comput. Mach.* 5 (1958), 266–280.
28. GIUNTINI, R. *Quantum Logic and Hidden Variables*. BI Wissenschaftsverlag, Mannheim, 1991.
29. GREENBERGER, D. B., HORNE, M., AND ZEILINGER, A. Multiparticle interferometry and the superposition principle, *Physics Today 46* (August 1993), 22–29.
30. GRUSKA, J. *Quantum Computing*, McGraw-Hill, London, 1999.
31. HOPCROFT, J. E., AND ULLMAN, J. D. *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, MA, 1979.
32. KOCHEN, S., AND SPECKER, E. P. The problem of hidden variables in quantum mechanics, *Journal of Mathematics and Mechanics 17*, 1 (1967), 59–87.
33. KONDACS, A., WATROUS, J. On the power of quantum finite state automata, *Proceedings of 38th IEEE FOCS*, 1997, 66-75.
34. KOZEN, D. *Automata and Computability,* Springer-Verlag, New York, 1997.
35. MOORE, C. Dynamical recognizers: real-time language recognition by analogue computers, *Theoret. Comput. Sci.* 201 (1998), 99–136.
36. MOORE, C. Email to C. S. Calude, 10 May 2001.
37. MOORE, C. CRUTCHFIELD, J. P. Quantum automata and quantum grammars, *Theoret. Comput. Sci.* 237 (2000), 275–306.
38. MOORE, E. F. Gedanken-experiments on sequential machines, In *Automata Studies*, C. E. Shannon and J. McCarthy, Eds., Princeton University Press, Princeton, 1956, 129–153.
39. PASCHEN, K. Quantum finite automata using ancilla qubits, manuscript, May 2001.
40. PENROSE, R. *Shadows of the Minds, A Search for the Missing Science of Consciousness*, Oxford University Press, Oxford, 1994.
41. RABIN, M.O. Probabilistic automata, *Information and Control* 6 (1963), 230–244.
42. SALOMAA, A. *Computation and Automata*, Cambridge University Press, Cambridge, 1985.
43. SVOZIL, K. *Randomness & Undecidability in Physics*, World Scientific, Singapore, 1993.
44. WRIGHT, R. Generalized urn models, *Foundations of Physics 20* (1990), 881–903.