Finite Nondeterministic Automata: Simulation and Minimality*

Cristian S. Calude,† **Elena Calude,**‡ **Bakhadyr Khoussainov**§

Abstract

Motivated by recent applications of finite automata to theoretical physics, we study the minimization problem for nondeterministic automata (with outputs, but no initial states). We use Ehrenfeucht–Fraïsse-like games to model automata responses and simulations. The minimal automaton is constructed and, in contrast with the classical case, proved to be unique up to an isomorphism. Finally, we investigate the partial ordering induced by automata simulations. For example, we prove that, with respect to this ordering, the class of deterministic automata forms an ideal in the class of all automata.

1 Introduction

Automata have been used as toy models for physical particles for many years (see [12, 6, 7, 9]). Recent papers (see [10, 14, 17, 15, 8, 16, 3, 2, 5, 4]) have imposed the notion of "finite automaton with outputs and no initial states" as a basic model. In this context the interest is not directed to the languages accepted by various automata but to automata "behaviour" and "simulations". The "behaviour" of an automaton is described by its "responses" to various experiments (expressed as sequences of input symbols). An automaton A simulates the behaviour of an automaton B in case A can perform any computation B can perform and the outputs produced will be the same.

The case of deterministic automata (both complete and incomplete) being disposed in [2, 5], we concentrate our attention on nondeterministic automata. Various models of simulations will be considered and investigated. In constructing the minimal nondeterministic automaton we will rely on the notion of "indistinguishable states"[1] which will

*The first and third authors have been partially supported by AURC A18/XXXXX/62090/F3414056, 1996; the second author has been partially supported by AURC A18/XXXXX/62090/F3414070, 1997.

†Computer Science Department, The University of Auckland, Private Bag 92109, Auckland, New Zealand, e-mail: cristian@cs.auckland.ac.nz.

‡Computer Science Department, The University of Auckland, Private Bag 92109, Auckland, New Zealand, e-mail: elena@cs.auckland.ac.nz.

§Computer Science Department, The University of Auckland, Private Bag 92109, Auckland, New Zealand and Mathematics Department, Cornell University, Ithaca, USA, e-mail: bmk@cs.auckland.ac.nz.

[1] Informally, two states p, q of A are "distinguishable" if there is an experiment which makes A react (respond) differently on p and q.

be described by an equivalence induced by a suitable class of Ehrenfeucht–Fraïsse-like games. Minimal automata will be proven to be unique up to an isomorphism; this situation differs from the classical theory of nondeterministic automata (see for instance, [1, 11, 13, 18]) but it parallels and extends the theory of deterministic automata developed in [2, 5]. While for the deterministic case all constructions made use of "automata responses" only, i.e., no information about the internal machinery was necessary, *for nondeterministic automata we need the full internal machinery.*

Here is a brief review of the paper. Section 2 is devoted to basic notions and notations. Section 3 introduces automata trajectories and responses. Section 4 introduces and briefly discusses a class of Ehrenfeucht–Fraïsse-like games useful in modeling the nondeterministic automaton behaviour. In Section 5 we review five unsuccessful attempts in modeling the notion of "state indistinguishability"; this discussion motivates the introduction, in Section 6, of a well-behaved equivalence relation which will be essential for defining the notion of simulation and for constructing the minimal nondeterministic automaton in Section 7. Finally, we investigate the partial ordering induced by automata simulations.

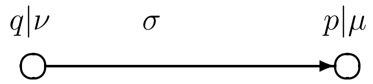
2 Notations

We begin by introducing some notations and basic definitions. If S is a finite set, then $|S|$ denotes the cardinality of S. Let Σ be a finite set (sometimes called alphabet); the set $\Sigma^\star$ stands for the set of all finite words over Σ with the empty word denoted by λ. The length of a string x is denoted by $|x|$. We fix two finite alphabets Σ and O: Σ contains **input** symbols, and O contains **output** symbols. A **nondeterministic finite automaton** over the alphabet Σ and O is a triple $A = (S_A, \nabla_A, F_A)$, where

- S_A is a finite nonempty set of **states**,
- ∇_A is a function from $S_A \times \Sigma$ to the set 2^{S_A} of all subsets of S_A, called the **transition table**,
- F_A is a mapping from the set of states S_A into the output alphabet O, called **output function**.

The above definition does not include the so called *initial states* which makes our definition different from the classical one.

In drawing graph representations of automata, we denote states by $\circ$ and label them with symbols from the output alphabet.[2] The picture



means that there is a transition σ from q to p, that is $p \in \nabla_A(q, \sigma)$, and $F_A(q) = \nu,\ F_A(p) = \mu$.

[2]Sometimes, we omit the name of the state.

In contrast with the fact that minimal *deterministic* automata (with initial states) accepting the same language are isomorphic, for nondeterministic automata (with initial states) there exist minimal *non*-isomorphic nondeterministic automata A and B which accept the same language (for the classical theory of automata see [1, 11, 13, 18]). We give an example. The graph representation of A is in Figure 1; the output function is given by $F_A(s_0) = F_A(s_1) = 1$ and the initial state is s_0. The automaton accepts the language $\{a^n b^m \mid n, m \geq 0\}$.

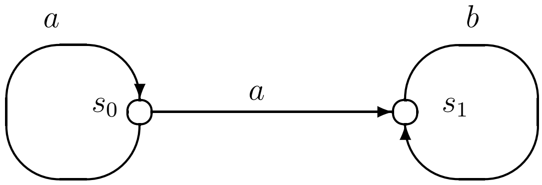


Figure 1.

In Figure 2 we have an automaton B whose initial state is p_0 and $F_B(p_0) = F_B(p_1) = 1$; B accepts the same language as A.

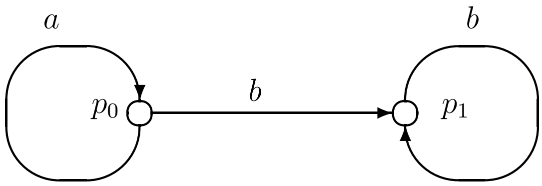


Figure 2.

Both nondeterministic automata are minimal but they are not isomorphic. Informally, one can say even more: neither A nor B *simulate* each other; they accept the same language just by chance. This type of negative phenomenon does not occur under an appropriate definition of simulation for nondeterministic automata with no initial states.

3 Trajectories and Responses

Let $A = (S_A, \nabla_A, F_A)$ be a nondeterministic automaton. 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 $s_{i+1} \in \nabla_A(s_i, \sigma_{+1})$ 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 $R_A(s, w)$.

These functions permit the *identification* of those states of A which give the same response to the same inputs. Indeed, we can consider two equivalence relations $\equiv_0$ and $\equiv_1$ defined as follows. We say that two states p and q of S_A are $\equiv_1$–**equivalent** if for all $w \in \Sigma^\star$, $R_A(p,w) = R_A(q,w)$. Similarly, we say that two states p and q of S_A are $\equiv_0$–**equivalent** if for all $w \in \Sigma^\star$, $f_A(p,w) = f_A(q,w)$.

It is clear that if $p \equiv_1 q$, then $p \equiv_0 q$. The example below shows that in contrast to the deterministic case (see [2]), $\equiv_1$ is not the same as $\equiv_0$.

Example 3.1 *Consider the automaton A whose state diagram is given in Figure 3. We have $p \equiv_0 q$ and $p \not\equiv_1 q$.*

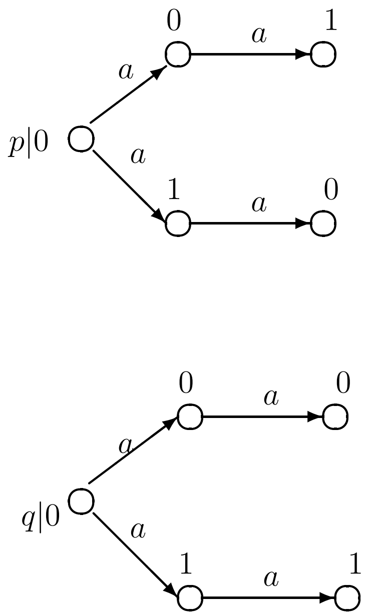


Figure 3.

Example 3.2 *The automaton A in Figure 4 has the following property: there exist two states $p, q \in S_A$ such that $p \equiv_1 q$, but for all $p' \in \nabla_A(p,\sigma)$ and $q' \in \nabla_A(q,\sigma)$, we have $p' \not\equiv_1 q'$.*

Indeed, it is not hard to see that for all $w \in \Sigma^*$ $R_A(p,w) = R_A(q,w)$. It follows that $p \equiv_1 q$. However, no $p' \in \nabla_A(p,a)$ is $\equiv_1$–equivalent to any $q' \in \nabla_A(q,a)$.

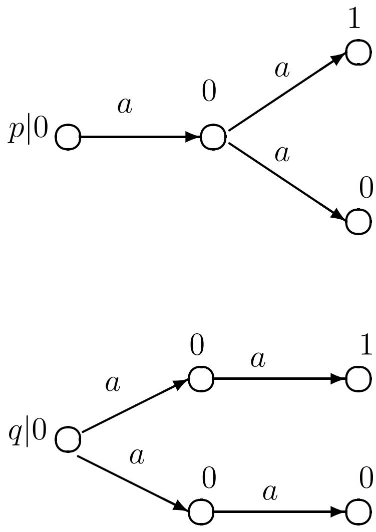


Figure 4.

Motivated by the phenomenon described in Example 3.2[3] we will be interested in those equivalence relations on S_A which are "well–behaved" with respect to the transition table ∇_A. Here is the appropriate definition. An equivalence relation $\equiv$ on S_A is **well–behaved** if for all $p \equiv q$ $(p, q \in S_A)$ and for every $\sigma \in \Sigma$ the following properties hold:

1. For every $p' \in \nabla_A(p, \sigma)$ there is a state $q' \in \nabla_A(q, \sigma)$ such that $p' \equiv q'$.
2. For every $q' \in \nabla_A(q, \sigma)$ there is a state $p' \in \nabla_A(p, \sigma)$ such that $q' \equiv p'$.

A well-behaved equivalence relation $\equiv$ should guarantee that any two $\equiv$–equivalent states *simulate* each other. Having a well–behaved equivalence relation $\equiv$, one can consider the factor automaton $A/_\equiv$ and prove that it is minimal.[4]

4 Game Responses

The above analysis of total and final responses suggests a game–theoretic approach in formalizing the notion of "response". Informally, the behaviour of a nondeterministic automaton A receiving an input w can be thought as a game with two players: Player 0 and Player 1. A move of any player consists of picking up a state of A. Player 0 picks a state p. Player 1 tries to pick up a state q such that the observer cannot distinguish p and q using responses coming from p and q; Player 0 tries to prove the opposite. For the sake of completeness we include some simple facts about finite games.

Let T be a finite tree, and W be a set of some paths from T. Nodes on even positions are **positions of Player** 0; the remaining nodes are **positions of Player** 1. A **play** is

[3] See also Lemma 2.2 in [2].

[4] Example 3.2 shows that the equivalence relation $\equiv_1$ is not well–behaved, so $\equiv_1$ cannot be used for constructing the minimal automaton.

a finite sequence of nodes

$$x_0 y_0 \dots x_k y_k$$

such that x_0 is the root of T and the sequence $x_0 y_0 \dots x_k y_k$ is a path in T. A **game** is the pair (T, W).

A strategy for Player 0 (Player 1) is a function which maps every position x of Player 0 (Player 1) to a child (i.e., an immediate successor) of x. For instance, Player 0 can follow a strategy g and an initial play according to this strategy can be:

$$g(x_0)y_0 g(y_0)y_1 g(y_1)y_2 g(y_2),$$

where x_0 is the root of T.

We say that Player 1 **wins** the game (T, W) if there is a strategy g for Player 1 such that every play played following g belongs to W; otherwise Player 1 looses.

Fact 4.1 *In the game (T, W) one of the players wins. If Player 1 does not win this game, then there is a strategy g for Player 0 such that every play played following g does not belong to W.*

Proof. Let C_W be the set of all nodes in T which are the last elements of the paths in W. We mark elements of T as follows:

Stage 0. Every element in C_W is marked.

Stage $i + 1$. Consider a node x. If x is a position of Player 0, then x is marked at this stage if all children of x are marked. Otherwise we do not mark x at this stage (x may be marked at later stages). If x is a position of Player 1, then x is marked if some child of x is marked. Otherwise x is not marked at this stage.

Clearly there is a stage after which no node will be marked. Thus, there are two cases:

Case 1. If the root is marked, then Player 1 wins. The winning strategy for Player 1 is the following: if x is marked and is a position for Player 1, then take a marked child of x.

Case 2. If the root is not marked, then Player 0 wins. The winning strategy for Player 0 is the following: if x is an unmarked position for Player 0, then take an unmarked child of x. $\square$

From the proof of this fact we get the following:

Corollary 4.2 *Consider the game (T, W). A strategy g is a winning strategy for Player 1 if and only if every play according to g goes through marked nodes.* $\square$

s

5 Unsuccessful Models

Fix a nondeterministic automaton A, two states $p, q \in S_A$ and a string $w = \sigma_1 \ldots \sigma_n \in \Sigma^\star$.

We define a finite game $G_w(p,q)$, called w–**response game**, with two players: Player 0 and Player 1. Player 0 always moves first, and Player 1 responds to each move. A **play** is a sequence

$$p_1 q_1 p_2 q_2 \ldots p_k q_k$$

such that the following conditions hold:

1. $p = p_1$, $q = q_1$,
2. $q_{i+1} \in \nabla_A(q_i, \sigma_i)$, for each $1 \leq i \leq k-1$, $p_{i+1} \in \nabla_A(p_i, \sigma_i)$.

Thus every play is a sequence of states. The letters on even positions are called **positions of Player** 0; the others are **positions of Player** 1. Since w is finite, every play in a w–response game is finite. A **strategy** for Player 0 (Player 1) is a function which maps the set of all finite words of even (odd) length from $S_A^\star$ to S_A. Note that since $G_w(p,q)$ is finite, every strategy of this game is a function with finite domain, hence the number of strategies is finite. If g is a strategy for Player 1, and

$$p_1 q_1 p_2 q_2 \ldots p_k$$

is a play played by Player 1 following g, then the next move of Player 1 is $g(p_1 q_1 p_2 q_2 \ldots p_k)$. For example, the following is an initial segment of a play according to g:

$$p\ q\ p_1\ g(pqp_1)\ p_2\ g(pqp_1 g(pqp_1)p_2)\ p_3\ g(pqp_1 g(pqp_1)p_2 g(pqp_1 g(pqp_1)p_2)p_3).$$

Similarly, Player 0 can follow a strategy in the game $G_w(p,q)$.

We say that Player 1 **wins** the play

$$p_1 q_1 p_2 q_2 \ldots p_k q_k$$

if $R_A(p_i, \sigma_i \ldots \sigma_n) = R_A(q_i, \sigma_i \ldots \sigma_n)$, for all $1 \leq i \leq k$. Otherwise, Player 0 wins. A player **wins the game** if it has a strategy g such that the player wins every play following g. Since $G_w(p,q)$ is a finite game, one of the players wins the game, by Fact 4.1.

We say that the states p and q are $\equiv_2$–**equivalent** if for every $w \in \Sigma^\star$, Player 1 wins the games $G_w(p,q)$ and $G_w(q,p)$. The next result follows from the definition.

Lemma 5.1 *For all states p, q, if $p \equiv_2 q$, then $p \equiv_1 q$.*

Lemma 5.2 *The relation $\equiv_2$ is an equivalence relation.*

Proof. It is clear that the relation is symmetric and reflexive. Suppose that $p \equiv_2 q$ and $q \equiv_2 s$. We need to show that $p \equiv_2 s$, that is Player 1 wins both games $G_w(p,s)$ and $G_w(s,p)$. We explain how Player 1 wins the game $G_w(p,s)$; by symmetry, one can then see how Player 1 wins the other game $G_w(s,p)$. Let g_1 and g_2 be winning strategies of Player 1 in games $G_w(p,q)$ and $G_w(q,s)$, respectively. Then the winning strategy g for Player 1 in the game $G_w(p,s)$ can be described by the following instructions:

First, think of any move of Player 0 as a move in the game $G_w(p,q)$. Secondly, using the strategy g_1, respond to the move as you were in the game $G_w(p,q)$. Thirdly, consider the response of Player 1 as a move of Player 0 in the game $G_w(q,s)$. Finally respond, using the strategy g_2, to the move as you were in the game $G_w(s,q)$.

It is not hard to see that this strategy g is a winning strategy for Player 1. $\square$

Unfortunately the equivalence relation $\equiv_2$ is not well-behaved.

Example 5.3 *The automaton A in Figure 5 has the following property: there exist two states $p \equiv_2 q$, but for all $p' \in \nabla_A(p,\sigma)$ and $q' \in \nabla_A(q,\sigma)$, we have $p' \not\equiv_2 q'$.*

Indeed, $p \equiv_2 q$, but for all $p' \in \nabla_A(p,\sigma)$ and $q' \in \nabla_A(q,\sigma)$, $p' \not\equiv_2 q'$.

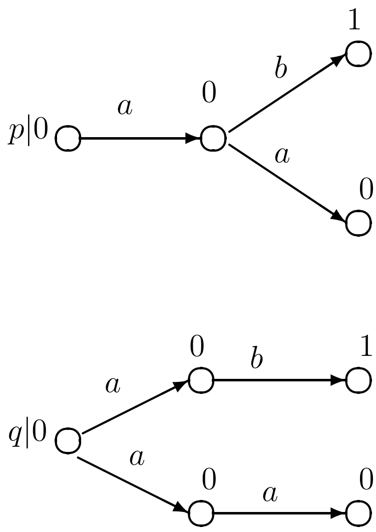


Figure 5.

The above example suggests a modification of the game $G_w(p,q)$. In the new game, called $G(p,q,w)$, every play is the same as in $G_w(p,q)$, but we say that Player 1 **strongly wins** the play

$$p_1 q_1 p_2 q_2 \dots p_k q_k$$

if $p_i \equiv_1 q_i$, for all $1 \le i \le k$. Again, since $G(p,q,w)$ is a finite game, one of the players wins the game. We say that the states p and q are **strongly $\equiv_3$–equivalent**, and we denote this by $\equiv_3$, if for every $w \in \Sigma^\star$, Player 1 strongly wins the games $G(p,q,w)$ and $G(q,p,w)$.

Lemma 5.4 *For all states p, q, if $p \equiv_3 q$, then $p \equiv_2 q$, and hence $p \equiv_1 q$.*

Again, however, the negative phenomenon occurs:

Example 5.5 *There is an automaton A such that $p \equiv_3 q$ for some $p, q \in S_A$, but for all $p' \in \nabla_A(p,\sigma)$ and all $q' \in \nabla_A(q,\sigma)$, $p' \not\equiv_3 q'$.*

The states of A accessible from p, respectively, q are given in Figure 6.

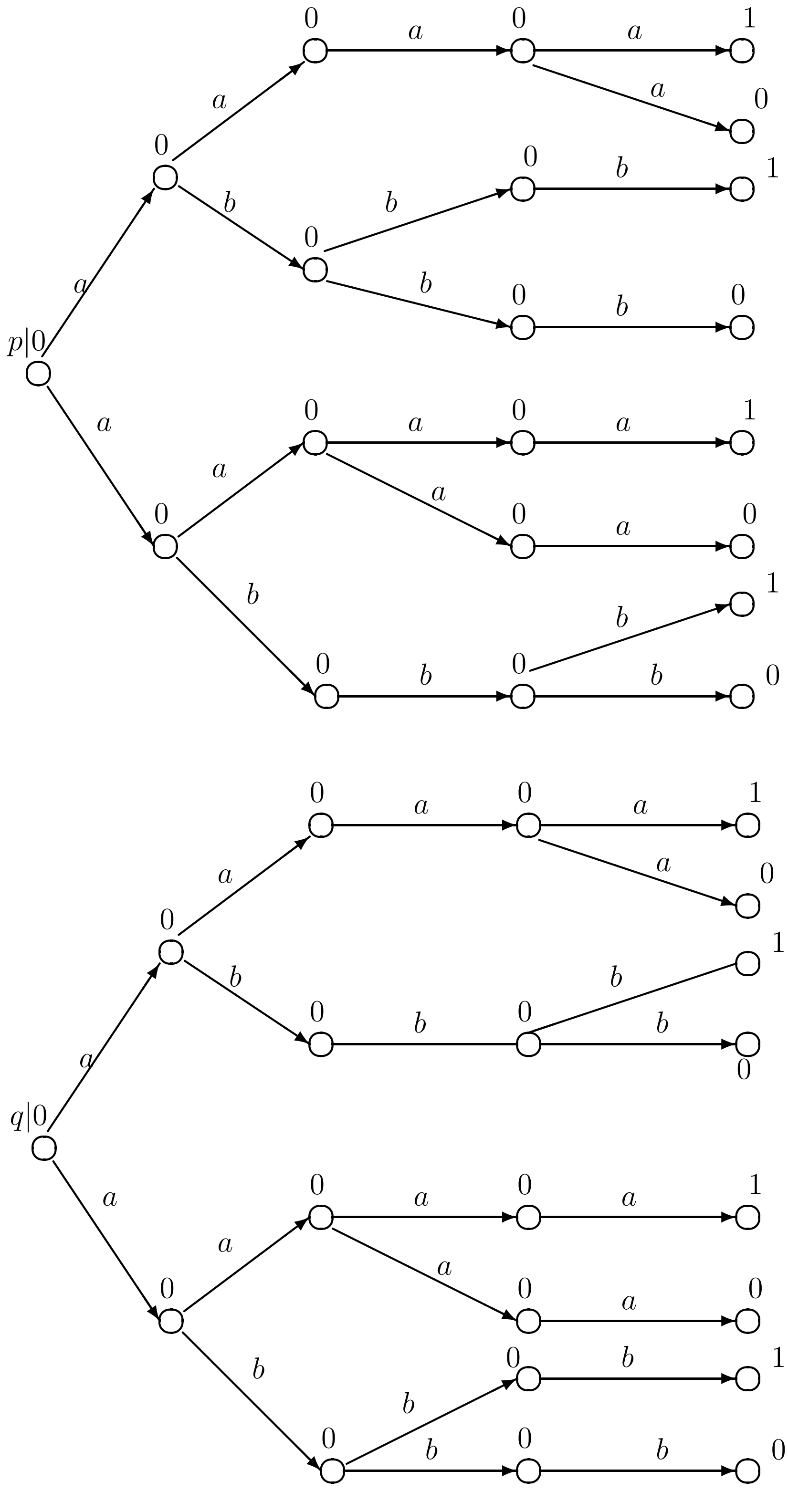


Figure 6.

The above analysis shows that we need to further refine the equivalence relation $\equiv_3$. To this aim we define two new equivalence relations. For $p, q \in S_A$ and $w \in \Sigma^\star$, consider again the game $G(p, q, w)$. A **continuation** of this game is any game $G(p, q, wu)$, where $u \in \Sigma^\star$. Clearly, if Player 1 wins $G(p, q, wu)$, then he wins $G(p, q, w)$ too. One of the main reasons that the equivalence relations $\equiv_3$ and $\equiv_2$ are not well–behaved is hidden in the following fact: In the game $G(p, q, w)$ Player 1 can *not* predict future actions of Player 0 when a new input u is inserted into A after w. In other words, a winning strategy for Player 1 in the game $G(p, q, w)$ can not always be extended to a winning strategy in any continuation of the game. Thus we are led to say that Player 1 **strategically wins** the game $G(p, q, w)$ if there is a strategy h for Player 1 in the game $G(p, q, w)$ such that for all $u \in \Sigma^\star$ the strategy h can be extended to a winning strategy of the game $G(p, q, wu)$.

Clearly, if Player 1 strategically wins the game $G(p, q, w)$, then he wins the game $G(p, q, w)$ itself. Now this definition allows us to consider an equivalence relation $\equiv_4$ finer than $\equiv_3$. We say that p and q are **$\equiv_4$–equivalent** if for every w, Player 1 strategically wins both games $G(p, q, w)$ and $G(q, p, w)$. Thus, if $p \equiv_4 q$, then for every w there is a winning strategy g (g') for Player 1 in the game $G(p, q, w)$ $((G(q, p, w))$ such that for all $u \in \Sigma^\star$ the strategy g (g') can be extended to a winning strategy in the game $G(p, q, wu)$ $((G(q, p, wu))$.

There is another possibility to refine $\equiv_3$ by defining a new game, denoted by $G(p, q, n)$, as follows: A **play** is a sequence

$$p_0 q_0 p_1 q_1 \dots p_k q_k$$

of states such that $p = p_0$, $q = q_0$, and for every $1 \leq i \leq k-1$ there are $\sigma_1, \sigma_2 \in \Sigma$ such that $p_{i+1} \in \nabla_A(p_i, \sigma_1)$ and $q_{i+1} \in \nabla_A(q_i, \sigma_2)$, $1 \leq k \leq n$. Thus, in this play Player 0 chooses $p_0 = p$, Player 1 chooses $q_0 = q$, Player 0 responds by taking any state p_1, etc. We say that Player 1 **wins** the play if for all $1 \leq i \leq n$ and $\sigma \in \Sigma$, $p_{i+1} \in \nabla_A(p_i, \sigma)$ if and only if $q_{i+1} \in \nabla_A(q_i, \sigma)$, and $F_A(p_i) = F_A(q_i)$.

We say that Player 1 **wins** the game $G(p, q, n)$ if there is a winning strategy h for Player 1 in the game $G(p, q, n)$. If Player 1 wins the game $G(p, q, n)$, then clearly he wins the game $G(p, q, w)$, for all $w \in \Sigma^*$, $|w| \leq n$. Note that if Player 1 wins the game $G(p, q, n+1)$, then he wins the game $G(p, q, n)$ as well. Two states p and q of A are **$\equiv_5$–equivalent** if for every n, Player 1 wins the games $G(p, q, n)$ and $G(q, p, n)$.

Now the following lemma is a consequence of definitions and Lemma 5.2.

Lemma 5.6

1. *The relations $\equiv_4$ and $\equiv_5$ are equivalence relations.*

2. *For all states p, q and $i = 4, 5$, if $p \equiv_i q$, then $p \equiv_3 q$.*

Again, it turns out that neither $\equiv_4$ nor $\equiv_5$ are well–behaved. We state this fact without giving any examples and turn our interest to the construction of a well–behaved equivalence relation.

Fact 5.7 *The equivalence relations $\equiv_4$ and $\equiv_5$ are not well–behaved.*

6 A Well–Behaved Equivalence Relation

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. Again, 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. The difference from the previous games is 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

$$p_0 p_1 \dots p_k$$

and

$$q_0 q_1 \dots q_k$$

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

$$p_0 p_1 \dots p_t$$

and

$$q_0 q_1 \dots q_t$$

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 $F_A(p_i) = F_B(q_i)$.

From the definition of the game $\Gamma(p, q, n)$ we have the following lemma.

Lemma 6.1 *If a player wins the game $\Gamma(p, q, n)$ then he wins the game $\Gamma(q, p, n)$.*

To formulate the next theorem we suppose that in the game $\Gamma(p, q, n)$ the automata A and B coincide. 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.

Theorem 6.2 *The relation $\equiv$ is a well-behaved equivalence relation on S_A.*

Proof. The first part of the theorem follows, with a slight modification, from the proof of Lemma 5.2. Suppose that $p \equiv q$ and $q \equiv s$. We need to show that $p \equiv s$, that is Player 1 wins the game $\Gamma(p, s, n)$ for every n. Let g_1 and g_2 be winning strategies for Player 1 in games $\Gamma(p, q, n)$ and $\Gamma(q, s, n)$, respectively. Then a winning strategy g for Player 1 in the game $\Gamma(p, s, n)$ can be described as follows. Suppose that at the end of stage k $(k < n)$ of a play the players have produced two sequences

$$p_0 p_1 \dots p_k$$

and

$$s_0 s_1 \dots s_k$$

where $p_0 = p$ and $s_0 = s$. If at stage $k+1$ Player 0 chooses a state p_{k+1} from $\bigcup_{\sigma\in\Sigma} \nabla_A(p_k, \sigma)$, then Player 1 follows the instructions bellow:

> *First, think of this move of Player* 0 *as a move in the game* $\Gamma(p, q, n)$*. Secondly, using the strategy* g_1*, respond to the move as you were in the game* $\Gamma(p, q, n)$*. Thirdly, consider this response of Player* 1 *as a move of Player* 0 *in the game* $\Gamma(q, s, n)$*. Finally respond, using the strategy* g_2*, to the move as you were in the game* $\Gamma(q, s, n)$*.*

On the other hand, if Player 0 chooses a state s_{k+1} from $\bigcup_{\sigma\in\Sigma} \nabla_A(s_k, \sigma)$, then Player 1 follows the instructions:

> *First, think of this move of Player* 0 *as a move in the game* $\Gamma(q, s, n)$*. Secondly, using the strategy* g_2*, respond to the move as you were in the game* $\Gamma(q, s, n)$*. Thirdly, consider this response of Player* 1 *as a move of Player* 0 *in the game* $\Gamma(p, q, n)$*. Finally respond, using the strategy* g_1*, to the move as you were in the game* $\Gamma(p, q, n)$*.*

In both cases the strategy is clearly a winning strategy for Player 1.

We prove the second part. Suppose that p is $\equiv$–equivalent to q. We need to show that for every $\sigma \in \Sigma$ and every $p' \in \nabla_A(p, \sigma)$ there is a $q' \in \nabla_A(q, \sigma)$ such that p' is $\equiv$–equivalent to q'. Let $q_1, \dots, q_s$ be all states belonging to $\nabla_A(q, \sigma)$. Suppose that none of q_i is $\equiv$–equivalent to p'. Then for every q_i there is an n_i such that Player 0 wins the game $\Gamma(p', q_i, n_i)$. Let h_i be a strategy for Player 0 to win the game $\Gamma(p', q_i, n_i)$. Then Player 0 wins also any continuation of the game, $\Gamma(p', q', n_i + t)$, for every natural number t. Let n be the maximal number among all $n_1, \dots, n_s$ and consider the game $\Gamma(p, q, n)$. Suppose that in this game the first move of Player 0 is p'. If Player 1 responses by *not* taking a state from $\{q_1, \dots, q_s\}$, then clearly Player 1 looses the game. On the other hand, if Player 1 chooses a state q_i, then Player 0 simply follows the strategy h_i. It is clear that in this case Player 0 wins the game $\Gamma(p, q, n)$ which contradicts the fact that $p \equiv q$. □

7 Simulations and Minimality

Let A an B be nondeterministic automata. We say that A **is simulated by** B, or equivalently, B **simulates** A, 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 $A \leq B$.[5] Thus, the function h in this definition means that Player 1 wins the game $\Gamma(p, h(p), n)$, for every n.

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 $\sigma \in \Sigma$, $[q] \in \nabla_{M(A)}([s], \sigma)$ if and only if $q \in \nabla_A(s, \sigma)$.

[5]Note that the simulation relation defined above coincides with the simulations of deterministic automata, in case A and B are deterministic; see [2].

3. $F_{M(A)}([s]) = F_A(s)$.

The next lemma, concerning the relationship between A and $M(A)$, is an exact analogue of the case for deterministic automata (see [2]).

Lemma 7.1 *The automata A and $M(A)$ simulate each other.*

Proof. We prove that automaton A is simulated by $M(A)$ via the mapping $s \mapsto [s]$, for all $s \in S_A$. We need to show that Player 1 has a strategy to win the game $\Gamma(s, [s], n)$, for each n. Suppose that at the end of stage k ($k < n$) of a play the players have produced two sequences

$$s_0 s_1 \dots s_k$$

and

$$[p_0][p_1] \dots [p_k]$$

where $s_0 = s$ and $[p_0] = [s]$. By induction, we can assume that $p_k \equiv s_k$. Suppose that at stage $k+1$ Player 0 chooses a s_{k+1} from $\bigcup_{\sigma \in \Sigma} \nabla_A(s_k, \sigma)$. Since $s_k \equiv p_k$, by Theorem 6.2, there exists a $p_{k+1} \in \bigcup_{\sigma \in \Sigma} \nabla_A(p_k, \sigma)$ such that $s_{k+1} \equiv p_{k+1}$. Hence Player 1 picks up this p_{k+1}. Suppose that at stage $k+1$ Player 0 chooses a $[p_{k+1}]$ from $\bigcup_{\sigma \in \Sigma} \nabla_{M(A)}([p_k], \sigma)$. Again by the same theorem Player 1 can choose a s_{k+1} such that $s_{k+1} \equiv [p_{k+1}]$.

Similarly, one can prove that the automaton $M(A)$ is simulated by A via the mapping $[s] \mapsto \min[s]$, where $\min[s]$ is the minimal element in $[s]$ under some fixed linear ordering in S_A. $\square$

Say that two automata A and B are equivalent (and denote this by $A \sim B$) if $A \leq B$ and $B \leq A$. Clearly, the relation $\sim$ is an equivalence relation. A nondeterministic automaton A is **minimal** if for every nondeterministic automaton B such that $A \sim B$ one has $|S_A| \leq |S_B|$.

Our goal is to prove that each class $[A] = \{B | \ A \sim B\}$ contains a minimal automaton which is unique up to an isomorphism. We recall that two automata A and B are **isomorphic** if there is a bijective mapping $h : S_A \to S_B$ such that for all $s, p \in S_A, \sigma \in \Sigma$, $p \in \nabla_A(s, \sigma)$ if and only if $h(p) \in \nabla_B(h(s), \sigma)$ and $F_A(s) = F_B(h(s))$.

Lemma 7.2 *The automaton $M(A)$ is minimal.*

Proof. The proof is similar to the deterministic case. Suppose that B is minimal. Let $h : S_{M(A)} \to S_B$ be a mapping such that $M(A)$ is simulated by B via h. Then h is one–to–one. Otherwise, there exist two states $[p] \neq [q]$ in $S_{M(A)}$ such that $h([p]) = h([q])$. Hence $p \equiv h(p)$, $h(p) = h(q)$, and $h(q) \equiv q$. It follows that $[p] \equiv [q]$, and consequently, $p \equiv q$, i.e., $[p] = [q]$. This is a contradiction. Thus, $|S_{M(A)}| \leq |S_B|$. $\square$

In the last step we show the unicity up to an isomorphism of the minimal automaton.

Lemma 7.3 *If B is minimal and $A \sim B$, then B is isomorphic to $M(A)$.*

Proof. Suppose that B is minimal. There exists a mapping $h : S_{M(A)} \to S_B$ such that $M(A)$ is simulated by B via h. From the proof of Lemma 7.2 we see that h must be a one–to–one mapping. Since the automaton B is minimal, h must be onto. Indeed, assume by contradiction that there is a mapping $g : S_B \to S_{M(A)}$ such that B is simulated by $M(A)$ via g and $g(p) = g(q)$, for some $p, q \in S_B$. Hence $p \equiv q$. Since $M(B) \sim B$, B cannot minimal, a contradiction. Consequently, h is a bijection from $S_{M(A)}$ to S_B.

We need to prove that h is an isomorphism. It is clear that $F_{M(A)}([s]) = F_B(h([s]))$, for all $s \in S_A$. Suppose that $[s] \in \nabla_{M(A)}([p], \sigma)$. We need to show that $h[s] \in \nabla_B(h([p]), \sigma)$. Since $[p] \equiv h([p])$, there exists a $q \in \nabla_B(h([p]), \sigma)$ such that $q \equiv [s]$. Hence $q \equiv h([s])$ since h establishes a simulation. If $q \neq h([s])$, then since $q \equiv h([s])$, we have $|S_{M(B)}| < |S_B|$. This is again a contradiction with the assumption that B is minimal. Hence $q = h([s])$ and $h([s]) \in \nabla_B(h([p]), \sigma)$. □

The above lemmas prove the main theorem of this section.

Theorem 7.4 *For every nondeterministic automaton A, the automaton $M(A)$ satisfies the following properties:*

1) The automata A and $M(A)$ simulate each other.

2) The automaton $M(A)$ is minimal.

3) The automaton $M(A)$ is unique up to isomorphism.

8 Simulation as a Partial Ordering

The goal of this section is to investigate the partial ordering induced by $\leq$, the simulation of nondeterministic automata. Recall that $[A] = \{B \mid B \sim A\}$. We say that $[A]$ is **simulated by** B, and denote this by $[A] \leq [B]$, if $A \leq B$. In other words, the relation $\leq$ naturally induces a partial ordering in the class K of all equivalences classes $[A]$. We add to K the empty automaton E with meaning that $E \leq [A]$, for every automaton A. Thus, we have a partially ordered set $\mathcal{K} = (K, \leq)$ with the least element E. In this section we investigate this partially ordered set and give a characterization of $\leq$ in terms of embeddings of minimal automata.

A **morphism** from an automaton A to an automaton B is a mapping $h : S_A \to S_B$ having the following properties:

1. $F_A(s) = F_B(h(s))$, for all $s \in S_A$,
2. $p \in \nabla_A(s, \sigma)$ if and only if $h(p) \in \nabla_B(h(s), \sigma)$, for all $p, s \in S_A$ and $\sigma \in \Sigma$,
3. for all $q \in \nabla_B(h(s), \sigma)$, there is a $p \in \nabla_A(s, \sigma)$ such that $q = h(p)$.

If h is one to one, then A is **embedded** into B.

The following lemma follows from the above definition.

Lemma 8.1 *If there is a morphism from A to B, then $A \leq B$.*

Proof. Indeed, suppose that h establishes a morphism from A to B. We need to show that Player 1 wins the game $\Gamma(p, h(p), n)$ for each $p \in S_A$ and positive integer n. Suppose that at the end of stage k ($k < n$) of a play the players have produced two sequences

$$p_0 p_1 \dots p_k$$

and

$$s_0 s_1 \dots s_k$$

where $p_0 = p$ and $s_0 = h(p)$. Suppose that at stage $k+1$ Player 0 chooses a state p_{k+1} from $\bigcup_{\sigma \in \Sigma} \nabla_A(p_k, \sigma)$. Then Player 1 chooses $h(p_{k+1})$. Suppose that at stage $k+1$ Player 0 chooses a s_{k+1} from $\bigcup_{\sigma \in \Sigma} \nabla_A(s_k, \sigma)$. Then since h is a morphism there is p_{k+1} such that $h(p_{k+1}) = s_{k+1}$. Hence the response of Player 1 is simply p_{k+1}. One can see that this is indeed a winning strategy for Player 1. $\square$

The following result connects the ordering $\leq$ with the notion of embedding.

Theorem 8.2 *For all $[A], [B] \in K$, $[A] \leq [B]$ if and only if $M(A)$ is embedded into $M(B)$.*

Proof. In view of the previous Lemma 8.1, it is not hard to check that if $M(A)$ is embedded into $M(B)$, then $[A] \leq [B]$.

Suppose that $[A] \leq [B]$. Consider the minimal automata $M(A)$ and $M(B)$. There is a mapping $h : S_{M(A)} \to S_{M(B)}$ such that $M(A)$ is simulated by $M(B)$ via h. The function h must be injection. Otherwise, using a standard reasoning from the previous section we can prove that $M(A)$ is not minimal. Similarly, one can see that h is an embedding. $\square$

In fact the above proof gives a stronger result.

Corollary 8.3 *For all A and B if $[A] \leq [B]$, there is a unique embedding of $M(A)$ into $M(B)$.*

Now we show some other algebraic properties of the partially ordered set $\mathcal{K}$. A lower (upper) lattice is a partial ordered set in which every two elements have a supremum (infimum). A lattice is a partial ordered set which is both an upper and lower lattice.

Lemma 8.4 *$(K, \leq)$ is an upper lattice.*

Proof. Take two classes $[A]$ and $[B]$ and assume that S_A and S_B are disjoint. Therefore we can consider a new automaton, denoted by $A \vee B$, which is obtained by taking the union of the set of states, transition diagrams, and output functions of the automata A and B. It is clear that $[A] \leq [A \vee B]$ as well as $[B] \leq [A \vee B]$. We want to show that for any $[C]$ if $[A] \leq [C]$ and $[B] \leq [C]$, then $[A \vee B] \leq [C]$. Indeed, suppose that $[A] \leq [C]$ via $h_1 : S_A \to S_C$ and $[B] \leq [C]$ via $h_2 : S_B \to S_C$. The function $h = h_1 \cup h_2$ is clearly well-defined and one can easily see that $[A \vee B] \leq [C]$ via h. $\square$

Lemma 8.5 *$(K, \leq)$ is a lower lattice.*

Proof. Take two classes $[A]$ and $[B]$ and assume that S_A and S_B are disjoint. Consider the automata $M(A)$ and $M(B)$ as well as all automata C such that C can be embedded into $M(A)$ and $M(B)$. The number of all nonisomorphic automata which can be embedded into A and B is finite. If this number is 0, then clearly $E = [A] \wedge [B]$. Let

$$A_1, \ldots A_n$$

be all automata embedded into A as well as into B with pairwise disjoint domains. Then it is not hard to see that the automaton

$$A_1 \vee \ldots \vee A_n,$$

denoted by $A \wedge B$, has the property that for all A_i, $A_i \leq A \wedge B$. Moreover $[A \wedge B] \leq [A]$ and $[A \wedge B] \leq [B]$. It follows that $\mathcal{K}$ is a lower lattice. □

A **covering** of a class $[A]$ is a class $[B]$ such that $[A] \leq [B]$, $[A] \neq [B]$, and for all $[C]$ with $[A] \leq [C] \leq [B]$, either $[A] = [C]$ or $[B] = [C]$.

Lemma 8.6 *Suppose that $|O| > 1$. Then every element of $\mathcal{K}$ has infinitely many coverings.*

Proof. Let $\sigma \in \Sigma$ and suppose that $0, 1 \in O$. For each prime number p consider the automaton A_p with the following properties:

1. S_A has exactly p number of states $s_1, \ldots, s_p$,
2. $\nabla_{A_p}(s_i, \sigma) = \{s_{i+1}\}$, for all $i \leq p - 1$, and $\nabla_{A_p}(s_p, \sigma) = \{s_1\}$,
3. $F_{A_p}(s_2) = \ldots = F_{A_p}(s_p) = 0$ and $F_{A_p}(s_1) = 1$.

It is not hard to see that if $p \neq p'$, then neither A_p nor $A_{p'}$ simulate each other. It can also be checked that $A_p = M(A_p)$, for all p. Finally, take any automaton A and suppose that $|S_{M(A)}| = n$. Then for all $p > n$, $[A] \vee [A_p]$ is a covering of $[A]$. □

An **ideal** of $\mathcal{K}$ is a subset $\mathcal{I} \subset K$ such that for all $[A], [B] \in \mathcal{I}$ the following properties hold:

1. If $[A] \in \mathcal{I}$ and $[B] \leq [A]$, then $[B] \in \mathcal{I}$.
2. If $[A], [B] \in \mathcal{I}$, then $[A] \vee [B] \in \mathcal{I}$.

Lemma 8.7 *The set $K^d = \{[A] \mid A$ is deterministic$\}$ is an ideal of $\mathcal{K}$.*

Proof. If $[B] \leq [A]$ and $[A] \in K^d$, then $M(B)$ is embedded into $M(A)$. Hence $M(B)$ is a deterministic automaton, so $[B] \in K^d$. If $[A], [B] \in \mathcal{I}$, then clearly the disjoint union of A and B is a deterministic automaton. Hence $[A] \vee [B]$ belongs to K^d. □

From the above lemmas we get the following:

Theorem 8.8 *The partially ordered set $\mathcal{K}$ is a lattice each element of which has infinitely many coverings. Moreover, the set K^d is an ideal of $\mathcal{K}$.*

Acknowledgment

We thank Dr. Marjo Lipponen for her critical comments and suggestions.

References

[1] Brauer, W. *Automatentheorie.* Teubner, Stuttgart, 1984.

[2] Calude, C., Calude, E., and Khoussainov, B. Deterministic automata: simulation, universality and minimality. *Annals of Pure and Applied Logic* 90, 1-3 (1997), 263-276.

[3] Calude, C., Calude, E., Svozil, K., and Yu, S. Physical versus computational complementarity I. *International Journal of Theoretical Physics* 36 (1997), 1495-1523.

[4] Calude, C. S., and Lipponen, M. Computational complementarity and sofic shifts. in X. Lin (ed.). *Theory of Computing 98, Proceedings of the 4th Australasian Theory Symposium, CATS'98,* Springer-Verlag, Singapore, 1998, 1998, 277-290.

[5] Calude, E., and Lipponen, M. Deterministic incomplete automata: simulation, universality and complementarity. *Journal of Universal Computer Science* 11 (1997), 1180-1193.

[6] Chaitin, G. J. An improvement on a theorem by E. F. Moore. *IEEE Transactions on Electronic Computers EC-14* (1965), 466–467.

[7] Conway, J. H. *Regular Algebra and Finite Machines.* Chapman and Hall Ltd., London, 1971.

[8] Dvurečenskij, A., Pulmannová, S., and Svozil, K. Partition logics, orthoalgebras and automata. *Helvetica Physica Acta 68* (1995), 407–428.

[9] Finkelstein, D., and Finkelstein, S. R. Computational complementarity. *International Journal of Theoretical Physics 22*, 8 (1983), 753–779.

[10] Grib, A. A., and Zapatrin, R. R. Automata simulating quantum logics. *International Journal of Theoretical Physics 29*, 2 (1990), 113–123.

[11] Hopcroft, J. E., and Ullman, J. D. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, MA, 1979.

[12] Moore, E. F. Gedanken-experiments on sequential machines. In *Automata Studies*, C. E. Shannon and J. McCarthy, Eds. Princeton University Press, Princeton, 1956.

[13] Salomaa, A. *Computation and Automata.* Cambridge University Press, Cambridge, 1985.

[14] Schaller, M., and Svozil, K. Partition logics of automata. *Il Nuovo Cimento 109B* (1994), 167–176.

[15] Schaller, M., and Svozil, K. Automaton partition logic versus quantum logic. *International Journal of Theoretical Physics 34*, 8 (August 1995), 1741–1750.

[16] Schaller, M., and Svozil, K. Automaton logic. *International Journal of Theoretical Physics 35*, 5 (May 1996), 911–940.

[17] Svozil, K. *Randomness & Undecidability in Physics.* World Scientific, Singapore, 1993.

[18] Yu, S. Regular languages. In *Handbook of Formal Languages, Vol. 1-3*, G. Rozenberg and A. Salomaa, Eds. Springer-Verlag, Heidelberg, 1997, 41-110.