ELSEVIER Annals of Pure and Applied Logic 90 (1997) 263–276

ANNALS OF
PURE AND
APPLIED LOGIC

Deterministic automata
Simulation, universality and minimality

Cristian Calude*,[1], Elena Calude, Bakhadyr Khoussainov[1]

Computer Science Department, The University of Auckland, Private Bag 92109, Auckland, New Zealand

Received 10 December 1996; accepted 9 July 1997
Communicated by S.N. Artemov

Abstract

Finite automata have been recently used as alternative, discrete models in theoretical physics, especially in problems related to the dichotomy between endophysical/intrinsic and exophysical/extrinsic perception (see, for instance [3, 6, 18–21]). These studies deal with Moore experiments; the main result states that it is impossible to determine the initial state of an automaton, and, consequently, a discrete model of Heisenberg uncertainty has been suggested. For this aim the classical theory of finite automata – which considers automata with initial states – is not adequate, and a new approach is necessary. A study of finite deterministic automata *without initial states* is exactly the aim of this paper. We will define and investigate the complexity of various types of simulations between automata. Minimal automata will be constructed and proven to be unique up to an isomorphism. We will build our results on an extension of Myhill–Nerode technique; all constructions will make use of "automata responses" to simple experiments only, i.e., no information about the internal machinery will be considered available.

AMS Classification: 68Q68, 81-08

Keywords: Automata simulations; Universal automaton; Minimal automaton

1. Introduction

Recent applications of automata to theoretical physics (see [6, 7, 12, 18–21]) have shown that the classical theory of finite automata – which considers automata with initial states – is not adequate, and a new approach is necessary. Briefly, here is the story.

The theory of relativity altered the classical concept of physical objectivity but left open the possibility of a supreme mathematician who, in Einstein's view, neither cheats

* Corresponding author. E-mail: cristian@cs.auckland.ac.nz.

[1] Partially supported by AURC A18/XXXXX/62090/F3414056, 1996.

PII S0168-0072(97)00039-0

nor plays dice. Quantum mechanics went one step further: it not only did situate the experimenter *in* the universe, but it has stated that the experimenter can be modeled as a "sturdy, classical entity" composed of a macroscopic number of microscopic objects. The observer – who can neither predict nor control certain "spontaneous" microphysical events – is bound by complementarity, – i.e., informally speaking, either experiences one certain type of observation (exclusive) or a different, complementary one. Complementarity is tied up with measurement, a highly controversial matter, as contemplations by Wigner [24], Wheeler [22], and Bell [1], among many others, show.[2]

Moore [14] was the first to study some experiments on finite deterministic automata[3] in an attempt to understand what kind of conclusions about the internal conditions of a finite machine it is possible to draw from input–output experiments. Machines 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. A (simple) Moore experiment can be described as follows: a copy of the 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 its initial state. The experimenter will study the 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".[4] Moore's experiments have been studied from a mathematical point of view by various researchers, notably by Ginsburg [9], Gill [8], Chaitin [4], Conway [5], Brauer [2], Calude [3]. The main conclusion of these studies is that *it is impossible to determine the initial state of an automaton*, and, consequently, a discrete model of Heisenberg uncertainty has been suggested. For this aim the classical theory of finite automata – which considers automata with initial states – is not adequate, so in this paper we are going to study finite deterministic automata *without initial states*. We will define and study the complexity of various types of simulations between automata. Minimal automata will be constructed and proven to be unique up to an isomorphism; this situation parallels and extends the classical theory of deterministic automata (see, for instance, [2, 13]). We will build our results on an extension of Myhill–Nerode technique; all constructions will only make use of "automata responses" to simple experiments, i.e., no information about the internal machinery will be considered available.

[2] The easiest way to prove this fact is to observe that in certain instances it is possible to "reconstruct" the quantum wave function after its alleged "collapse" [11]. Thereby, not a single (quantum) bit of information should remain available from the previous "measurement". In such a scenario, it is possible to "measure" complementary observables: the price to be paid amounts to the total ignorance of the first "measurement outcome".

[3] To emphasize the conceptual nature of his experiments, Moore has borrowed from physics the word "Gedanken".

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

2. Notation

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 $|x|$. In what follows all automata will be *finite*, i.e, they operate with a finite number of states, on finite input and output alphabets.

We fix two finite alphabets Σ and $O: \Sigma$ contains *input* symbols, and O contains *output* symbols. A finite deterministic automaton consists of a finite set of states and a set of transitions from state to state that occur on input symbols chosen from Σ. For each symbol there is exactly one transition out of each state, possibly back to the state itself. Any state "emits" an output from the set O. Formally, a *deterministic (finite) automaton* over the alphabets Σ and O is a triple $A=(S_A, \Delta_A, F_A)$, where

- S_A is a finite nonempty set called the *set of states*,
- Δ_A is a function from $S_A \times \Sigma$ to the set of states S_A, called *the transition table*, and
- F_A is a mapping from the set of states S_A into output alphabet O, called the *output function*.

The above definition does not include the so-called *initial states*; this fact makes our definition different from the classical ones.[5]

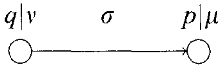
In this paper we will deal only with deterministic automata, hence, we will omit the word deterministic.

Let A be an automaton. We can naturally extend the transition diagram Δ_A to a function, also denoted by Δ_A, from $S_A \times \Sigma^{\star}$ to S_A as follows:

$$\Delta_A(s,\lambda)=s, \qquad \Delta_A(s,\sigma w)=\Delta_A(\Delta_A(s,\sigma),w)$$

for all $s \in S_A$, $\sigma \in \Sigma$ and $w \in \Sigma^{\star}$.

In drawing graph representations of automata, we denote states by o and label them with symbols from the output alphabet. The picture



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

2.1. Responses

Our goal is to define the notion of response of the automaton $A=(S_A, \Delta_A, F_A)$ to an *input signal*, that is, to a word from $\Sigma^{\star}$. We give several definitions to formalize this notion.

[5] Ginsburg [10, pp. 5–42], has studied Mealy automata without initial states; however, as we shall prove in the final section, his results on simulation and minimality cannot be used for Moore automata as in this context and in opposition with the classical situation, Moore automata *are not* particular cases of Mealy automata.

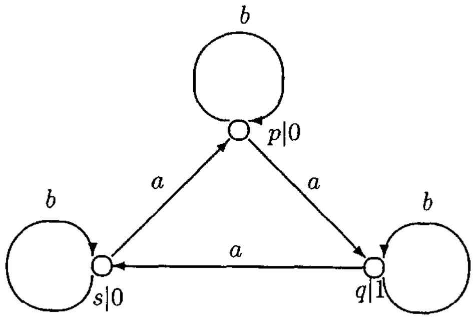


Fig. 1.

- The *total response* of the automaton A is the function $R_A : S_A \times \Sigma^\star \to O^\star$ defined as follows:

$$R_A(s,\lambda) = F_A(s),$$

$$R_A(s,\sigma_1\cdots\sigma_n) = F_A(s)F_A(\Delta_A(s,\sigma_1))F_A(\Delta_A(s,\sigma_1\sigma_2))\cdots F_A(\Delta_A(s,\sigma_1\cdots\sigma_n)),$$

  where $\sigma_i \in \Sigma$, $s \in S_A$, $n \geqslant 1$, $1 \leqslant i \leqslant n$.
- The *final response* of A is the function $f_A : S_A \times \Sigma^\star \to O$ 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 $i_A : S_A \times \Sigma^\star \to O$ defined, for all $s \in S_A$ and $w \in \Sigma^\star$, by $i_A(s,w) = F_A(s)$.

Here is an example. Consider $\Sigma = \{a,b\}$, $O = \{0,1\}$ and the three-state automaton A presented in Fig. 1.

The output function is defined by $F_A(s) = F_A(p) = 0$ and $F_A(q) = 1$. Clearly, $R_A(s,aab) = 0011$, $R_A(p,aab) = 0100$, $R_A(q,aab) = 1000$. We also have $f_A(s,aab) = 1$, $f_A(p,aab) = 0$, $f_A(q,aab) = 0$, $i_A(s,aab) = 0$, $i_A(p,aab) = 0$, $i_A(q,aab) = 1$.

2.2. Strong simulation

We continue by giving a formal definition of the strong simulation. 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 on the outputs.

Let $A = (S_A, \Delta_A, F_A)$ and $B = (S_B, \Delta_B, F_B)$ be automata. We say that

- A is *strongly simulated by B*, or, equivalently, *B strongly simulates A* if there is a mapping $h : S_A \to S_B$ such that
  1. For all $s \in S_A$ and $\sigma \in \Sigma$, $h(\Delta_A(s,\sigma)) = \Delta_B(h(s),\sigma)$.
  2. For all $s \in S_A$ and $w \in \Sigma^\star$, $R_A(s,w) = R_B(h(s),w)$.

  We denote this by $A \ll B$.

- 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
  1. For all $s \in S_A$ and $\sigma \in \Sigma$, $h(\Delta_A(s, \sigma)) = \Delta_B(h(s), \sigma)$.
  2. For all $s \in S_A$ and $w \in \Sigma^\star$, $f_A(s, w) = f_B(h(s), w)$ $(i_A(s, w) = i_B(h(s), w))$.

  We denote this fact by $A \ll_f B$ $(A \ll_i B)$.

Lemma 2.1. *If $h : S_A \to S_B$ and B strongly simulates A via h (or $A \ll_f B$ or $A \ll_i B$ via h), then for all $s \in S_A$ and $w \in \Sigma^\star$ we have $h(\Delta_A(s, w)) = \Delta_B(h(s), w)$.*

Proof. The proof follows directly using the induction on the length of w. If $|w| = 1$, then $w = \sigma$ for some $\sigma \in \Sigma$ and hence $h(\Delta_A(s, \sigma)) = \Delta_B(h(s), \sigma)$. Suppose that $h(\Delta_A(s, w)) = \Delta_B(h(s), w)$ for all w such that $|w| \leqslant n - 1$. Let now, w be $v\sigma$ with $|v| = n - 1$. Then

$$\begin{aligned} h(\Delta_A(s, w)) &= h(\Delta_A(\Delta_A(s, v), \sigma)) \\ &= \Delta_B(h(\Delta_A(s, v)), \sigma) = \Delta_B(\Delta_B(h(s), v), \sigma) = \Delta_B(h(s), w). \qquad \square \end{aligned}$$

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

Theorem 2.1. *Let A and B be automata. The following conditions are equivalent:*
(1) *The automaton A is strongly simulated by B.*
(2) *The automaton A is strongly i-simulated by B.*
(3) *The automaton A is strongly f-simulated by B.*

Proof. The implication (1)⇒(2) is obvious. We prove the implication (2)⇒(3). Suppose that B strongly i-simulates A via $h : S_A \to S_B$. We need to show that the equality

$$f_A(s, w) = f_B(h(s), w) \tag{1}$$

holds for all $s \in S_A$ and $w \in \Sigma^*$. Indeed, we substitute $\Delta(s, w)$ for s in the relation $h(\Delta_A(s, \sigma)) = \Delta_B(h(s), \sigma)$ to get

$$f_A(s, w) = F_A(\Delta_A(s, w)) = F_B(h(\Delta_A(s, w)) = F_B(\Delta_B(h(s), w)) = f_B(h(s), w).$$

We prove the implication (3)⇒(1) by induction on the length of w. For $w = \lambda$ the equality $R_A(s, w) = R_B(h(s), w)$ follows from the definition of strong f-simulation. Suppose that $R_A(s, w) = R_B(h(s), w)$ holds for all $s \in S_A$ and $w \in \Sigma^\star$ with $|w| \leqslant n - 1$. Let $w \in \Sigma^*, |w| = n - 1$ and $\sigma \in \Sigma$. Then

$$\begin{aligned} R_A(s, w\sigma) &= R_A(s, w) F_A(\Delta_A(\Delta_A(s, w), \sigma)) \\ &= R_B(h(s), w) F_A(\Delta_A(s, w\sigma)) \\ &= R_B(h(s), w) F_B(\Delta_B(h(s), w\sigma)) \\ &= R_B(h(s), w\sigma). \qquad \square \end{aligned}$$

2.3. Behavioral simulation

From an algebraic point of view, strong simulations are morphisms between automata; they make essential use of the internal machinery of automata, i.e. of the transition Δ_A, which is sometimes difficult to access. In this section we discuss another notion of simulation, the behavioral simulation, which is weaker than strong simulation. The behavioral simulation makes use only of outputs produced by automata.

To motivate the formalization, we begin by presenting two automata A and B such that neither A strongly simulates B nor B strongly simulates A. However, in a behavioral sense these two automata "simulate" each other, i.e. they have the same behavior. The input alphabet is $\{a\}$ and the output alphabet is $\{0,1\}$. Fig. 2 gives a graph representation of A and Fig. 3 represents B.

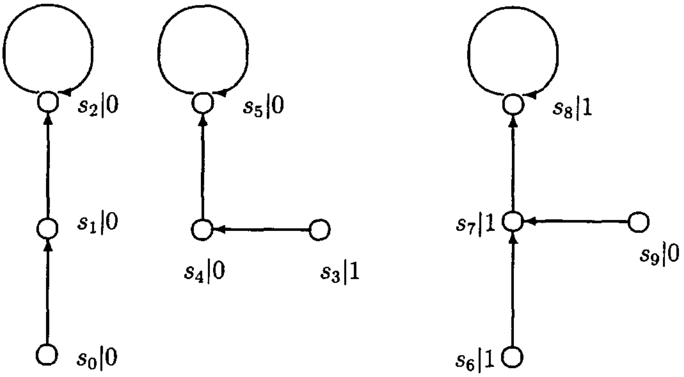


Fig. 2.

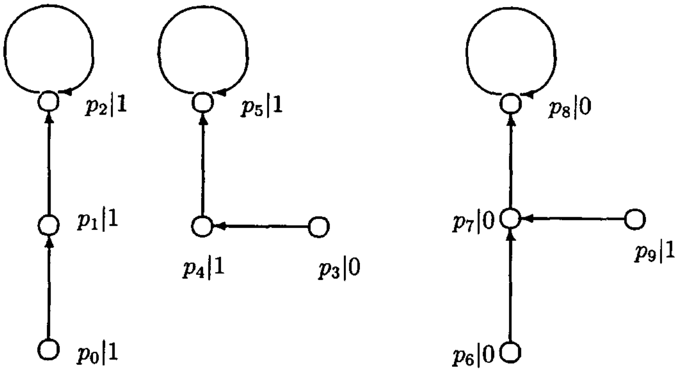


Fig. 3.

It is not hard to see that A cannot strongly simulate B and B cannot strongly simulate A. However, the mapping $h : S_A \to S_B$ defined by

$$h(s_0) = p_6, h(s_1) = p_7, h(s_2) = p_8, h(s_6) = p_0, h(s_7) = p_1,$$

$$h(s_8) = p_2, h(s_4) = p_7, h(s_5) = p_8, h(s_3) = p_9, h(s_9) = p_3,$$

satisfies the following property: for all $s \in S_A$ and all $w \in \Sigma^\star$,

$$R_A(s,w) = R_B(h(s),w),$$

and the reader can immediately find a function $h' : S_B \to S_A$ such that for all $s \in S_B$ and all $w \in \Sigma^\star$, $R_A(h'(s),w) = R_B(s,w)$.

Now, we are ready to give the definition of behavioral simulation (called in what follows, simply, simulation). Let $A = (S_A, \Delta_A, F_A)$ and $B = (S_B, \Delta_B, F_B)$ be 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 $s \in S_A$ and $w \in \Sigma^\star$, $R_A(s,w) = R_B(h(s),w)$. We denote this by $A \leqslant B$.
- A is *f-simulated* (*i-simulated*) by B, or, equivalently, B *f-simulates* (*i-simulates*) A if there is a mapping $h : S_A \to S_B$ such that for all $s \in S_A$ and $w \in \Sigma^\star$, $f_A(s,w) = f_B(h(s),w)$ $(i_A(s,w) = i_B(h(s),w))$. We denote this fact by $A \leqslant_f B$ $(A \leqslant_i B)$.

A partial analogue of Theorem 2.1 holds true.

Theorem 2.2. *An automaton A is simulated by B if and only if A can be f-simulated by B.*

Proof. It is clear that if $A \leqslant B$, then $A \leqslant_f B$. Assume that $f_A(s,w) = f_B(h(s),w)$ holds true for all $s \in S_A$ and $w \in \Sigma^*$. We prove, by induction on the length of w, that for all $s \in S_A$ and $w \in \Sigma^*$,

$$R_A(s,w) = R_B(h(s),w). \tag{2}$$

Clearly, $R_A(s,\lambda) = F_A(s) = f_A(s,\lambda) = f_B(h(s),\lambda) = R_B(h(s),\lambda)$.

Finally, if (2) holds true, then for every $\sigma \in \Sigma$ we have

$$\begin{aligned} R_A(s,w\sigma) &= R_A(s,w)F_A(\Delta_A(s,w),\sigma) \\ &= R_A(s,w)f_A(s,w\sigma) \\ &= R_B(h(s),w)f_B(h(s),w\sigma) \\ &= R_B(h(s),w)F_B(\Delta_B(h(s),w\sigma)) \\ &= R_B(h(s),w\sigma). \qquad \square \end{aligned}$$

A counterexample 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 $w \in \Sigma^*$.

2.4. Generalized Myhill–Nerode equivalences

Let $A = (S_A, \Delta_A, F_A)$ be an automaton. Let R be one of the response functions on A, i.e., let $R \in \{R_A, f_A, i_A\}$. Two states p and q from S_A are R-equivalent if for all $w \in \Sigma^\star$, $R(p,w) = R(q,w)$. If p and q are R-equivalent we denote this fact by $p \equiv_R q$.

Intuitively, if p and q are R-equivalent, then all computations of A which begin from p cannot be R-distinguished by computations of A which begin from q and vice-versa. It is immediate that $\equiv_R$ is an equivalence relation on S_A.

Lemma 2.2. *Let A be an automaton. Then for all $p, q \in S_A$, $p \equiv_{R_A} q$ if and only if $p \equiv_{f_A} q$.*

Proof. Clearly, if $p \equiv_{R_A} q$, then $p \equiv_{f_A} q$. Conversely, suppose that $p \equiv_{f_A} q$. First, $R_A(p,\lambda) = F_A(p) = f_A(p,\lambda) = f_A(q,\lambda) = F_A(q) = R_A(q,\lambda)$. Assume now that $R_A(p,w) = R_A(q,w)$, for some $w \in \Sigma^\star$. Then, for every $\sigma \in \Sigma$, we have

$$\begin{aligned} R_A(p,w\sigma) &= R_A(p,w) f_A(p,w\sigma) \\ &= R_A(q,w) f_A(q,w\sigma) \\ &= R_A(q,w\sigma). \quad \Box \end{aligned}$$

Lemma 2.3. *For all $p \equiv_{f_A} q$ and every $w \in \Sigma^\star$, we have $\Delta_A(p,w) \equiv_{f_A} \Delta_A(q,w)$.*

Proof. For all $u \in \Sigma^\star$ we have

$$\begin{aligned} f_A(\Delta_A(p,w),u) &= F_A(\Delta_A(\Delta_A(p,w),u)) \\ &= F_A(\Delta_A(p,wu)) \\ &= f_A(p,wu) \\ &= f_A(q,wu) \\ &= f_A(\Delta_A(q,w),u). \end{aligned}$$

Remarks. (a) For all $p, q \in S_A$, $p \equiv_{R_A} q$ (or, equivalently, $p \equiv_{f_A} q$) implies $F_A(p) = F_A(q)$.

(b) Note that $p \equiv_{R_A} q$ implies $p \equiv_{i_A} q$, but the converse implication fails to hold true.

2.5. Universal minimal automata

Suppose that we have a finite class $\mathscr{C}$ 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 $\mathscr{C}$ if the following conditions hold:
(1) For any $1 \leqslant i \leqslant n$ there is a state $s \in S_A$ such that $R_A(s,w) = R_{A_i}(q_i,w)$, for all $w \in \Sigma^\star$.
(2) 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*. It is not hard to see that every automaton A (with no initial states) naturally defines a class $\mathscr{C}(A)$ for which A itself is universal. Indeed, let $q_1,\ldots,q_n \in S_A$ be all states of A and for each i, define $A_i = A$. Clearly A is universal for the class $\mathscr{C}(A) = \{(A_1,q_1),\ldots (A_n,q_n)\}$.

Not every finite class of finite automata with initial states has a universal automaton. However, every class can be embedded into a complete one.

Proposition 2.1. *Let $\mathscr{C}'$ be a finite class of pairs of automata and initial states. There is a complete class $\mathscr{C}$ containing $\mathscr{C}'$.*

Proof. Let $\mathscr{C}' = \{(A_i = (S_i, \Delta_i, F_i), q_i) \mid 1 \leqslant i \leqslant n\}$. Assume that all the states of these automata are pairwise disjoint. Consider the automaton A obtained by taking the union of all these automata, that is

$$A = \left(\bigcup_i^n S_i, \bigcup_i^n \Delta_i, \bigcup_i^n F_i\right).$$

Consider now the class $\mathscr{C}(A)$ as defined above. Take $\mathscr{C} = \mathscr{C}' \cup \mathscr{C}(A)$ and note that $\mathscr{C}'$ is contained in $\mathscr{C}$ and A is universal for $\mathscr{C}$. □

It turns out that the notion of universality is closely related to the notion of simulation.

Theorem 2.3. *The automata A and B simulate each other if and only if A and B are universal for the same class.*

Proof. Suppose that A and B simulate each other via $h_1 : S_A \rightarrow S_B$ and $h_2 : S_B \rightarrow S_A$. Consider the class $\mathscr{C}(A)$, for which the automaton A is universal. We show that B is universal for $\mathscr{C}(A)$. Suppose that $(A_1 = (S_A, \Delta_A, F_A), q_1,)$ belongs to $\mathscr{C}(A)$. Then for all $w \in \Sigma^\star$, we have $R_{A_1}(q_1,w) = R_B(h(q_1),w)$. For every $q \in S_B$ there exists a state $q' \in S_A$ such that for the pair $(A' = (S_A, \Delta_A, F_A), q')$ we have $R_{A'}(q',w) = R_B(q,w)$, for all $w \in \Sigma^\star$. Hence, B is universal for $\mathscr{C}(A)$.

Now, assume that A and B are universal for the class $\mathscr{C} = \{(A_1,q_1),(A_2,q_2)\ldots, (A_n,q_n)\}$. For every $q \in S_A$ there exists an i such that $R_A(q,w) = R_{A_i}(q_i,w)$, for all $w \in \Sigma^\star$. Since $(A_i,q_i) \in \mathscr{C}$ and B is universal for $\mathscr{C}$ there is a state $p(q_i) \in S_B$ – say the minimal one according to a fixed linear order defined on the set of all states – such that $R_{A_i}(q_i,w) = R_B(p(q_i),w)$, for all $w \in \Sigma^\star$. Hence, A is simulated by B via mapping $q \rightarrow p(q_i)$. Similarly, B can be simulated by A. □

Our next goal is to show that every complete class has a minimal universal automaton. An automaton A universal for the class $\mathscr{C}$ is *minimal* if for every automaton B universal for the class $\mathscr{C}$ we have $|S_A| \leqslant |S_B|$.

From this definition and Theorem 2.3 above we obtain the following:

Corollary 2.1. *The following are equivalent*:
(1) *The automaton A is minimal for every class $\mathscr{C}$ for which it is universal.*
(2) *For every automaton B, if $A \leqslant B$ and $B \leqslant A$, then $|S_A| \leqslant |S_B|$.*

Informally, a minimal automaton A is one which has the minimal possible number of states among all automata which have the same "computational power" as A. We will soon show that the notion of minimality implies uniqueness up to an "isomorphism".

An automaton $A = (S_A, \Delta_A, F_A)$ is *isomorphic* to $B = (S_B, \Delta_B, F_B)$ if there is a one to one onto mapping $h : S_A \to S_B$ such that for all $\sigma \in \Sigma$ we have $h(\Delta_A(s,\sigma)) = \Delta_B(h(s),\sigma)$ and $F_A(s) = F_B(h(s))$.

Clearly, if A is isomorphic to B, then A and B strongly simulate (hence simulate) each other. However, the converse implication is not always true. A simple example consists of two automata A and B over the alphabets $\Sigma = \{a\}$ and $O = \{0,1\}$. The automaton A has two states p and q such that $F_A(p) = 0$, $F_A(q) = 0$, $\Delta_A(p,a) = p$, $\Delta_A(q,a) = q$. The automaton B has one state s such that $F_A(s) = 0$ and $\Delta_A(s,a) = s$. Thus, A and B strongly simulate each other but they are not isomorphic.

Let $A = (S_A, \Delta_A, F_A)$ be an automaton. We provide a construction of a minimal automaton for $\mathscr{C}(A)$ using the generalized Myhill–Nerode equivalence relation $\equiv_{f_A}$ on S_A. We shall omit the index f_A and write simply $\equiv$ instead of $\equiv_{f_A}$.[6] For any $s \in S_A$, $[s]$ denotes the equivalence class of s under $\equiv$, that is $[s] = \{p \in S_A \mid s \equiv p\}$.

Define a new automaton $M(A)$ as follows:

- The set of states of $M(A)$ is $S_{M(A)} = \{[s] \mid s \in S_A\}$.
- For all $[s]$ and $\sigma \in \Sigma$, put $\Delta_{M(A)}([s],\sigma) = [\Delta_A(s,\sigma)]$.
- For all $[s]$, put $F_{M(A)}([s]) = F_A(s)$.

Lemma 2.3 and its following remarks show that the automaton $M(A)$ is well defined. The next result proves that the process of constructing the automata $M(A), M(M(A)), \ldots$ converges from the first stage.

Fact 2.1. *For every automaton A, the automata $M(A)$ and $M(M(A))$ are isomorphic.*

Proof. Indeed, if $[p] \equiv_{f_{M(A)}} [q]$, then clearly $p \equiv_{M(A)} q$, that is $[p] = [q]$. Hence, the mapping $[p] \to [\![p]\!]$ is an isomorphism from $M(A)$ to $M(M(A))$. □

Lemma 2.4. *The automaton $M(A)$ simulates A.*

[6] Note that by Lemma 2.2 we could equally use $\equiv_{R_A}$.

Proof. The automaton A is simulated by $M(A)$ via the mapping $h : S_A \to S_{M(A)}$ defined by $h(s) = [s]$. Indeed, by the definition of $[s]$ we see that for all $w \in \Sigma^\star$, we have $R_A(s,w) = R_{M(A)}([s],w)$. □

The above just proved lemma shows that $A \leqslant M(A)$. The next lemma takes care of the case $M(A) \leqslant A$.

Lemma 2.5. *The automaton A simulates $M(A)$.*

Proof. We can assume that the set of states of the automaton A is linearly ordered. Therefore, each class $[s]$ contains a minimal element $\min[s]$ with respect to the order. We define the mapping $h : S_{M(A)} \to S_A$ by setting $h([s]) = \min[s]$, for all $[s] \in S_{M(A)}$. Thus, for all $[s] \in S_{M(A)}$ and $w \in \Sigma^\star$ we have

$$R_{M(A)}([s],w) = R_A(s,w) = R_A(\min[s],w) = R_A(h([s]),w).$$

It follows that A simulates $M(A)$. □

Lemma 2.6. *Let A be an automaton.*
(1) *The automaton $M(A)$ is minimal.*
(2) *If B and A simulate each other and B is minimal, then $M(A)$ and B are isomorphic.*

Proof. Suppose that C and A simulate each other and C is minimal. To prove the first part of the lemma suppose that $|S_C| < |S_{M(A)}|$. There is a mapping $h : S_{M(A)} \to S_C$ such that $M(A)$ is simulated by C via h. Since $|S_C| < |S_{M(A)}|$ there exist two distinct states $[p],[q] \in S_{M(A)}$ such that $h([p]) = h([q])$. It follows that

$$R_A(p,w) = R_{M(A)}([p],w) = R_C(h([p]),w) = R_{M(A)}([q],w) = R_A(q,w)$$

for all $w \in \Sigma^\star$. It follows that $p \equiv q$, hence, $[p] = [q]$. This is a contradiction, hence, $M(A)$ is minimal.

We now prove (2). Since $M(A)$ and B simulate each other there is a mapping $h : S_{M(A)} \to S_B$ such that $R_{M(A)}([s],w) = R_B(h([s]),w)$, for all $w \in \Sigma^\star$. By (1) we see that h is one–one. The function h is also onto since B is minimal. We know that

$$R_{M(A)}([s],w) = R_B(h([s]),w) \tag{3}$$

for all $s \in S_A$ and $w \in \Sigma^\star$ and we need to show that for all $[s] \in S_{M(A)}$ and $\sigma \in \Sigma$,

$$F_{M(A)}([s]) = F_B(h([s])),$$

and

$$h(\Delta_{M(A)}([s],\sigma)) = \Delta_B(h([s]),\sigma). \tag{4}$$

The first equality follows directly from (3) by $w = \lambda$. To prove the second relation we show that $h(\Delta_{M(A)}([s],\sigma))$ is equivalent (in B) to $\Delta_B(h([s]),\sigma)$. For every $w \in \Sigma^\star$

we have

$$\begin{aligned}
f_B(h(\Delta_{M(A)}([s],\sigma)),w) &= f_{M(A)}(\Delta_{M(A)}([s],\sigma),w)\\
&= F_{M(A)}(\Delta_{M(A)}(\Delta_{M(A)}([s],\sigma),w))\\
&= F_{M(A)}(\Delta_{M(A)}([s],\sigma w)\\
&= f_{M(A)}([s],\sigma w)\\
&= f_B(h([s]),\sigma w)\\
&= F_B(\Delta_B(h([s]),\sigma w)\\
&= F_B(\Delta_B(\Delta_B(h([s]),\sigma),w))\\
&= f_B(\Delta_B(h([s]),\sigma),w).
\end{aligned}$$

Hence, the state $h(\Delta_{M(A)}([s],\sigma))$ is equivalent to the state $\Delta_B(h([s]),\sigma)$ in B. Since B is minimal, the equality (4) follows, so $M(A)$ is isomorphic to B. □

We have proved:

Theorem 2.4.

(1) *Any complete class has a minimal universal automaton which is unique up to an isomorphism.*

(2) *Any two minimal automata which simulate each other are isomorphic.*

Corollary 2.2. *Let A and B be minimal automata. The following are equivalent:*

(1) *The automata A and B strongly simulate each other.*

(2) *The automata A and B simulate each other.*

(3) *The automata A and B are isomorphic.*

Remarks. (a) A minimal automaton can be characterized by Moore's condition A (see [14, 3]): every pair of distinct states (p,q) is distinguishable by an experiment, i.e., there exists $w\in\Sigma^*$ such that $R_A(p,w)\neq R_A(q,w)$.

(b) *Every automaton A is strongly simulated by $M(A)$.* However, the converse does not hold, as the example of the automaton A in Fig. 2 shows. Therefore, $M(A)$ cannot be a minimal automaton in the class of all automata B such that $A\ll B$ and $B\ll A$.

(c) If A is an automaton, $A=(S_A,\Delta_A,F_A)$, and $q\in S_A$, then the (classical) minimal automaton corresponding to the pair (A,q), i.e. to A with the initial state q, can be obtained from the universal minimal automaton $M(A)$ by restricting $\Delta_{M(A)}$ and $F_{M(A)}$ to the set $\{[p]\mid p\in S_A,\ \Delta_A(q,w)=p',$ for some $w\in\Sigma^*, p'\equiv_{R_A} p\}$. In other words, from $M(A)$ one can immediately deduce the classical minimal automaton (but the converse is not true). This is another reason for calling $M(A)$ universal minimal.

(d) Moore [14] considers strongly connected automata and defines the notion of indistinguishability for these automata. For strongly connected automata indistinguishability coincides with simulation; the resulting minimal automata are isomorphic.

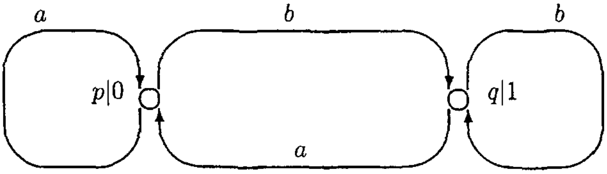


Fig. 4.

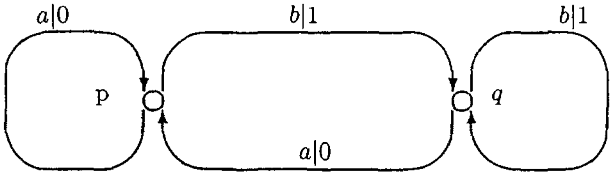


Fig. 5.

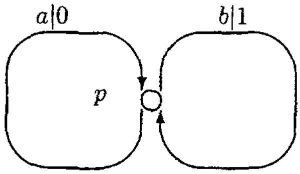


Fig. 6.

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

(e) Ginsburg [10] has proved the existence of minimal Mealy automata without initial states (for more general behavioristic automata see [15, 16]). Classically, Moore automata are particular cases of Mealy automata, so we may ask ourselves the question: Does Theorem 2.4 follow from Ginsburg's construction? The answer is *negative* and here is a counter example. The Moore automaton in Fig. 4 is minimal in the class of Moore automata. When considered in the class of Mealy automata (see Fig. 5) it *is not* a minimal Mealy automaton; the equivalent minimal Mealy automaton, see Fig. 6, in the larger class of Mealy automata, *is not* anymore a Moore automaton.

Acknowledgements

Professor S. Rudeanu [17] has suggested some improvements, specifically he proposed simpler proofs for Theorem 2.1 and Lemma 2.7. Dr. Marjo Lipponen has also suggested some modifications. We offer them our gratitude.

References

[1] J.S. Bell, Against Measurement, Phys. World 3 (1990) 33–40.

[2] W. Brauer, Automatentheorie, Teubner, Stuttgart, 1984.

[3] C. Calude, E. Calude, K. Svozil, S. Yu, Physical versus Computational Complementarity I, Internat. J. Theoret. Phys. 36 (1997) 1495–1523.

[4] G.J. Chaitin, An improvement on a theorem by E.F. Moore, IEEE Trans. Electron. Comput. EC-14 (1965) 466–467.

[5] J.H. Conway, Regular Algebra and Finite Machines, Chapman & Hall, London, 1971

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

[7] D. Finkelstein, S.R. Finkelstein, Computational complementarity, Internat. J. Theoret. Phys. 22 (8) (1983) 753–779.

[8] A. Gill, State-identification experiments in finite automata, Inform. Control 4 (1961) 132–154.

[9] S. Ginsburg, On the length of the smallest uniform experiment which distinguishes the terminal states of the machine, J. Assoc. Comput. Mach. 5 (1958) 266–280.

[10] S. Ginsburg, An Introduction to Mathematical Machine Theory, Addison-Wesley, Reading, MA, 1962.

[11] D.B. Greenberger, A. Yasin, Haunted measurements in quantum theory, Found. Phys. 19(6) (1989) 679–704.

[12] A.A. Grib, R.R. Zapatrin, Automata simulating quantum logics, Internat. J. Theoret. Phys. 29(2) (1990) 113–123.

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

[14] E.F. Moore, Gedanken-experiments on sequential machines, in: C.E.S. and J. McCarthy (Ed.), Automata Studies, Princeton University Press, Princeton, 1956

[15] S. Rudeanu, Behaviouristic automata I, minimal automata, An. Univ. Bucureşti Mat. 38 (1989) 64–71.

[16] S. Rudeanu, Behaviouristic automata II, The synthesis problem, An. Ştiinţ. Univ. 'Al. I. Cuza' Iaşi Secţ. I a Mat. (N.S.) 35 (1989) 303–312.

[17] S. Rudeanu, Email to C. Calude, 19 March, 1997.

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

[19] M. Schaller, K. Svozil, Automaton partition logic versus quantum logic, Internat. J. Theoret. Phys. 34 8 (August 1995) 1741–1750.

[20] M. Schaller, K. Svozil, Automaton logic, Internat. J. Theoret. Phys. 35 (5) (May 1996) 911–940.

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

[22] J.A. Wheeler, Law without law, in: J.A. Wheeler, W.H. Zurek (Eds.), Quantum Theory and Measurement, Princeton University Press, Princeton, 1983, pp. 182–213.

[23] J.A. Wheeler, W.H. Zurek, Quantum Theory and Measurement, Princeton University Press, Princeton, 1983.

[24] E.P. Wigner, Remarks on the mind-body question, in: I.J. Good (Ed.), The Scientist Speculates, Heinemann and Basic Books, London, 1961, pp. 284–302; reprinted in [Ref. 23, pp. 168–181].