Journal of Universal Computer Science, vol. 3, no. 11 (1997), 1180-1193
submitted: 8/8/97, accepted: 23/10/97, appeared: 28/11/97 © Springer Pub. Co.

Minimal Deterministic Incomplete Automata[1]

Elena Calude
(Department of Computer Science, The University of Auckland, Private Bag 92109, Auckland, New Zealand
Email: elena@cs.auckland.ac.nz)

Marjo Lipponen
(Department of Computer Science, The University of Auckland, Private Bag 92109, Auckland, New Zealand[2]
Email: marjo@cs.auckland.ac.nz.)

Abstract: We construct a minimal automaton for an output-incomplete Moore automaton. The approach is motivated by physical interpretation of seeing deterministic finite automata as models for elementary particles. When compared to some classical methods our minimal automaton is unique up to an isomorphism and preserves also the undefined or unspecified behaviour of the original automaton.

Key Words: Minimality, incomplete automata

Category: F.1.1

1 Introduction

In recent years theoretical physics has found finite deterministic automata as alternative, discrete models for particle behaviour, see [7, 9, 13, 16, 14, 5, 15, 2]. Incomplete automata seem to capture the natural behaviour of particles even better. Incomplete automata have also turned out to be interesting from the point of view of dynamical systems, giving the possibility to extend the notion of computational complementarity of discrete models to sofic shifts, see Calude and Lipponen [3].

Several authors, such as Ginsburg [8] and Mikolajczak [11] (for an extensive list of references, see Reusch and Merzenich [12]), have tackled the problem of finding minimal realizations for incomplete Moore or Mealy automata. Solutions to this problem depend on the precise definition of minimal realization. If the purpose is to find an automaton which produces the same outputs for the given input sequences, then instead of obtaining one minimal automaton, the method usually induces a family of automata each of which represents the original one.

Our approach is to find a minimal automaton which has the same behaviour compared to the original one. By the same behaviour we mean that also the undefined or unspecified behaviour – which plays a crucial part in quantum physics – is preserved, not only the responses. This guarantees that minimization loses as little information of the original automaton as possible. Another important factor is to consider the total responses, the outputs produced by all

[1] Proceedings of the *First Japan–New Zealand Workshop on Logic in Computer Science*, special issue editors D.S. Bridges, C.S. Calude, M.J. Dinneen and B. Khoussainov.

[2] On leave from the Department of Mathematics, University of Turku, Finland.

the states visited in the complete computation of the input instead of only the final responses, the outputs of the final states.

In Calude and Lipponen [4] we considered Moore automata that are input-incomplete but output-complete, meaning that all the states in the automaton emit an output but for some pairs of states and inputs the transition to the next state is undefined. We proved that the factor automaton $M(A)$ (with respect to the equivalence relation between the states) simulates the original automaton A, i.e., produces the same outputs for the same input sequences and nothing more. This automaton was showed to be minimal, that is, to have the least number of states among all the automata that simulate A, and to be unique up to an isomorphism. Furthermore, $M(A)$ is natural from the point of view of dynamical systems [3] since it generates exactly the same sofic shifts as A.

In Section 2 we will extend our results to output-incomplete Moore automata. It turns out that, in contrast to the classical models, our model is a natural extension from that of the complete automata (where all the transitions and outputs are defined). So the minimal complete automaton (as well as the minimal output-complete automaton) is obtained by exactly the same method.

Section 3 compares our method to one of the recent models by Mikolajczak [11], by means of an example. We explain why the classical minimal model does not satisfy the properties we set to the minimal automata though it usually has less states and is clearly optimal in another context.

2 Minimality

If S is a finite set, then $|S|$ denotes the cardinality of S. A **partial** function $f : A \overset{\circ}{\to} B$ is a function defined for some elements from A. In case f is not defined on $a \in A$ we write $f(a) = \infty$. Let $D(f) = \{a \in A \mid f(a) \neq \infty\}$ denote the **domain** of f. If $D(f) = A$, we say that f is **total**. Two partial functions f and g are equal, $f = g$, if $D(f) = D(g)$ and $f(a) = g(a)$, for every $a \in D(f)$. If Σ is a finite set, called alphabet, then Σ^* stands for the set of all finite words over Σ and the empty word, denoted by λ. By w^+ we mean all nonempty powers w^i of the word $w \in \Sigma^*$ whereas w^* includes also w^0, the empty word. The length of a word w is denoted by $|w|$. For further details, see Hopcroft and Ullman [10].

A **deterministic (finite) Moore automaton** over the alphabets Σ (**input** symbols) and O (**output** symbols) is a system $A = (S_A, \Delta_A, F_A)$, where S_A is the (finite and nonempty) **set of states**, Δ_A is the **transition table**, from $S_A \times \Sigma$ to the set of states S_A, and F_A is the **output mapping** from the set of states S_A into the output alphabet O.

If both the transition table and the output mapping are total then the corresponding automaton is said to be **complete**. If they are partial we can distinguish two types of incompleteness. We say that A is **input-incomplete** if the transition mapping Δ_A is partial and **output-incomplete** if the output mapping F_A is partial. If in the first case there is no transition from a state p labelled by a, we write $\Delta_A(p, a) = \infty$, and if in the second case the output of the state p is not specified, we write $F_A(p) = \Omega$ where Ω is a new symbol, $\Omega \notin O$. Thus we extend the partial function $F_A : S_A \overset{\circ}{\to} O$ to a total function $F_A : S_A \to O \cup \{\Omega\}$ which may have unspecified outputs. With this in mind, the following definitions are extensions of the ones presented in Calude and Lipponen [4].

The transition diagram can be naturally extended to a partial function $\Delta_A : S_A \times \Sigma^* \overset{\circ}{\to} S_A$, as follows: for every $s \in S_A$, $w \in \Sigma^*$ and $\sigma \in \Sigma$,

$$\Delta_A(s, \lambda) = s,$$
$$\Delta_A(s, \sigma w) = \begin{cases} \Delta_A(\Delta_A(s,\sigma), w), & \text{if } \Delta_A(s,\sigma) \neq \infty, \\ \infty, & \text{otherwise.} \end{cases}$$

We say that a word $w \in \Sigma^*$ is **applicable** to the state $p \in S_A$ if the transition $\Delta_A(p, w)$ is defined. The set of all applicable words to p is denoted by $W_A(p)$, and hence

$$W_A(p) = \{w \in \Sigma^* \mid \Delta_A(p, w) \neq \infty\}$$

consists of all words leading to complete computations on the state p.

If some of the outputs are unspecified we can set a further restriction to applicable words. Following Mikolajczak [11], a word $w \in \Sigma^*$ is said to be **admissable** for a state $p \in S_A$ if w is applicable to p and the output of the last state of the complete computation of w is specified. A set of all admissable words for a state p is denoted by $\hat{W}_A(p)$,

$$\hat{W}_A(p) = \{w \in \Sigma^* \mid w \in W_A(p) \text{ and } F_A(\Delta_A(p, w)) \neq \Omega\}.$$

For any state $p \in S_A$ the applicable words of p have always the property that if uv is applicable for p then also u is applicable for p. This is not the case for admissable words. Indeed, it is possible that $uv \in \hat{W}_A(p)$ but $u \notin \hat{W}_A(p)$, see Example 1. However, the following properties are straight consequences of the definition.

Lemma 1. *Let $A = (S_A, \Delta_A, F_A)$ be an incomplete automaton.*

1) *For all $p \in S_A$ and $u, v \in \Sigma^*$, if $uv \in \hat{W}_A(p)$ then $v \in \hat{W}_A(\Delta_A(p, u))$.*
2) *For all $p \in S_A$, $\lambda \in \hat{W}_A(p)$ iff $F_A(p) \neq \Omega$.*

The responses of an automaton $A = (S_A, \Delta_A, F_A)$ to an input signal $w \in \Sigma^*$ are defined as follows:

- The **total response** of A is the partial function $R_A : S_A \times \Sigma^* \overset{\circ}{\to} (O \cup \{\Omega\})^*$ such that

  $$R_A(s, \lambda) = F_A(s), \text{ and}$$
  $$R_A(s, \sigma_1 \dots \sigma_n) = F_A(s) F_A(\Delta_A(s, \sigma_1)) F_A(\Delta_A(s, \sigma_1\sigma_2)) \dots \dots F_A(\Delta_A(s, \sigma_1 \dots \sigma_n)),$$

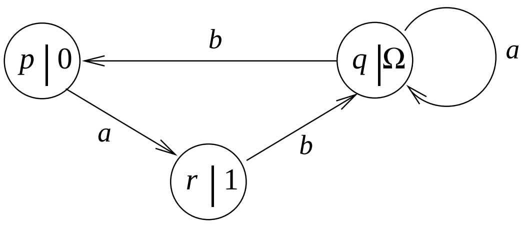
  for $s \in S_A$, $\sigma_1 \dots \sigma_n \in W_A(s)$, $\sigma_i \in \Sigma$, $n \geq 1$ and $1 \leq i \leq n$.
- The **final response** of A is the partial function $f_A : S_A \times \Sigma^* \overset{\circ}{\to} O \cup \{\Omega\}$ such that $f_A(s, w) = F_A(\Delta_A(s, w))$, for all $s \in S_A$ and $w \in W_A(s)$.

Thus, the total response is a sequence of outputs emitted by all the states that are visited in the complete computation of the input, whereas the final response is the output emitted only by the last state. Notice that $D(R_A) = D(f_A)$ and always $|R_A(s,w)| = |w| + 1$ with the convention that $|\Omega| = 1$.

If A is output-incomplete then the computation of the applicable word may visit states whose outputs are not specified. The physical interpretation of such unspecified behaviour lies in quantum physics. Say, we know that an electron goes through one of the two holes but we do not know which one. By Heisenberg's uncertainty principle any apparatus to determine the exact behaviour will disturb the electron enough to destroy the interference pattern (see, for instance, Feynman [6] for further details). To preserve this unspecified behaviour we will use the symbol Ω like any other output symbol and require that if two states respond in the same way for a given input then the unspecified outputs appear in the corresponding places of the output sequences.

Example 1. Let $\Sigma = \{a, b\}$, $O = \{0, 1\}$ and consider the three-state output-incomplete automaton A presented below.

A:



The state p emits an output 0, $F_A(p) = 0$, the state r emits an output 1, $F_A(r) = 1$, and the output of the state q is not specified, $F_A(q) = \Omega$.

The words a, aba^ib, $i \geq 0$, are admissable for the state p, the words ba^ib, ba^iba for the state r and the words a^ib, a^iba for the state q, respectively. On the other hand, the word ab is applicable to the state p but not admissable, $ab \in W_A(p) \setminus \hat{W}_A(p)$, whereas the word b is not applicable to the state p since $\Delta_A(p, b) = \infty$.

By definition, $R_A(p, aba) = 01\Omega\Omega$, $R_A(q, aba) = \Omega\Omega01$, and $R_A(r, aba) = \infty$. We also have $f_A(p, aba) = \Omega$, $f_A(q, aba) = 1$, and $f_A(r, aba) = \infty$.

In Calude and Lipponen [4] we defined in what conditions an automaton simulates the behaviour of another automaton, that is, responds in exactly the same way to the same input sequences. We used the abbreviation β–simulation instead of behavioral simulation. Formally, we say that the automaton $A = (S_A, \Delta_A, F_A)$ is β–**simulated** by the automaton $B = (S_B, \Delta_B, F_B)$ if there is a mapping $h : S_A \to S_B$ such that for all $s \in S_A$ and $w \in \Sigma^*$, $R_A(s, w) = R_B(h(s), w)$. If A and B both β–simulate each other, then they are said to be β–**equivalent**. Furthermore, if the mapping $h : S_A \to S_B$ is one-to-one and onto, and for all $s \in S_A$ and $\sigma \in W_A(s) \cap \Sigma$, $h(\Delta_A(s, \sigma)) = \Delta_B(h(s), \sigma)$, then A and B are **isomorphic**. An automaton A is **minimal** if every automaton B which is β–equivalent to A has at least as many states as A, $|S_A| \leq |S_B|$.

In order to obtain the minimal automaton we first define how two states can be distinguished by means of a "measurable experiment", i.e., by the responses of the automaton to an input $w \in \Sigma^*$. Following Calude and Lipponen [4], we say that the experiment is *not relevant* if it is applicable to neither p nor q; hence, another experiment is required. On the other hand, if the experiment is *relevant* then we have three further possibilities: w is applicable to either p or q but not to both, or w is applicable for both p and q and $R_A(p,w) \neq R_A(q,w)$ or $R_A(p,w) = R_A(q,w)$. In the first two cases w *distinguishes* between p and q, and in the third case w *cannot distinguish* between p and q.[3]

Consequently, two states $p, q \in S_A$ are **indistinguishable**, $p \equiv_f q$, iff

$$f_A(p,w) = f_A(q,w) \quad \text{for all } w \in \Sigma^*.$$

If p and q are not indistinguishable we say that they are **distinguishable**. In the same way we can define the relation $\equiv_R$ by using the total response R_A. Notice that this relation, in a sense, reflects β-simulation; in other words, A is β-simulated by B iff for every state $p \in S_A$ there is a state $h(p) \in S_B$ such that p and $h(p)$ are indistinguishable.

The following properties show that the relations $\equiv_f$ and $\equiv_R$ hold simultaneously (hence we will simply use $\equiv$ in the sequel) and are well-behaved, that is, the transition function preserves indistinguishability.

Lemma 2. *Let $A = (S_A, \Delta_A, F_A)$ be an incomplete automaton. Then for all $p, q \in S_A$,*

1) *$p \equiv_f q$ iff $p \equiv_R q$,*
2) *if $p \equiv_f q$ then $\Delta_A(p,w) \equiv_f \Delta_A(q,w)$ for all $w \in W_A(p)$,*
3) *if $p \equiv_f q$ then $F_A(p) = F_A(q)$.*

Proof. 1) If $p \equiv_R q$ then $p \equiv_f q$ by definition. So assume that $p \equiv_f q$. 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 all words w whose length is at most n. Then for any $\sigma \in \Sigma$,

$$\begin{aligned} R_A(p,w\sigma) &= R_A(p,w)F_A(\Delta_A(p,w\sigma)) \\ &= R_A(p,w)f_A(p,w\sigma) \\ &= R_A(q,w)f_A(q,w\sigma) \\ &= R_A(q,w)F_A(\Delta_A(q,w\sigma)) \\ &= R_A(q,w\sigma), \end{aligned}$$

if $w\sigma \in W_A(p)$; otherwise, $R_A(p,w\sigma) = \infty = R_A(q,w\sigma)$.

2) Let $w \in W_A(p) = W_A(q)$ and denote $r = \Delta_A(p,w)$ and $s = \Delta_A(q,w)$. The applicable words of r and s can be obtained from the applicable words of p and q,

$$W_A(r) = \{u \mid wu \in W_A(p)\} \quad \text{and} \quad W_A(s) = \{u \mid wu \in W_A(q)\}.$$

Since $W_A(p)$ and $W_A(q)$ are equal, also the sets $W_A(r)$ and $W_A(s)$ are equal. On the other hand, for any $w \in W_A(r)$,

$$f_A(r,u) = f_A(p,wu) = f_A(q,wu) = f_A(s,u).$$

[3] From the mathematical point of view this describes exactly the behaviour of two partial functions.

Thus r and s are indistinguishable, too.

3) The statement follows from the fact that $F_A(p) = f_A(p, \lambda) = f_A(q, \lambda) = F_A(q)$. $\square$

Notice that the similar proof as in Lemma 2 shows that we can define β–simulation equivalently in terms of final responses instead of total responses.

Indistinguishability is an equivalence relation since it is reflexive, symmetric and transitive; hence, we can construct a factor automaton $M(A)$ for any given automaton $A = (S_A, \Delta_A, F_A)$. Let $[s]$ denote the equivalence class of the state $s \in S_A$, $[s] = \{p \in S_A \mid s \equiv p\}$. The following algorithm shows how these classes can be found.

Algorithm

1. Let $S_A = \{q_1, q_2, \ldots, q_k\}$. Construct an empty table

q_2					
q_3					
$\vdots$					
q_{k-1}					
q_k					
	q_1	q_2	$\cdots$	q_{k-2}	q_{k-1}

2. If the states q_i and q_j, $i < j$, have different outputs or exactly one of the transitions $\Delta_A(q_i, a)$ and $\Delta_A(q_j, a)$ is defined and the other one is not defined, for some $a \in \Sigma$, then the states are distinguishable and we place $\times$ in the entry (q_i, q_j).
3. If the states q_i and q_j have exactly the same transitions and outputs, then they are indistinguishable and we place $\vee$ in the entry (q_i, q_j).
4. For any entry (q_i, q_j) which is still unmarked, we place all the pairs (s_i, s_j), $i \neq j$, where $s_i = \Delta_A(q_i, a)$ and $s_j = \Delta_A(q_j, a)$ for $a \in W_A(q_i) \cap \Sigma$.
5. If any of the entries (s_i, s_j) contains $\times$, we place $\times$ in the entry (q_i, q_j).
6. If all the entries (s_i, s_j) contain $\vee$, we place $\vee$ in the entry (q_i, q_j).
7. After repeating 5 and 6 (at most $k - 1$ times), we place $\vee$ in the remaining entries.

A closer look at the previous procedure shows that it follows naturally from the definition of indistinguishability. In Step 2 the states q_i and q_j are distinguishable either by the empty word, $f_A(q_i, \lambda) = F_A(q_i) \neq F_A(q_j) = f_A(q_j, \lambda)$, or by the word $w = a$ for which $\Delta_A(q_i, a) = \infty$ and $\Delta_A(q_j, a) \neq \infty$ (or vice versa). In Step 3 the states are indistinguishable if they have the same outputs and the transitions to the next states are exactly the same. In Step 5 the states q_i and q_j are distinguishable if at least one of the pairs s_i and s_j are distinguishable because of Lemma 2. On the other hand, in Step 6 where all the pairs (s_i, s_j) are indistinguishable, also q_i and q_j are indistinguishable since the possibility of having different outputs and transitions has already been excluded in Step 2. Finally, after repeating the two previous steps and no other entries can be filled, all the remaining unmarked pairs have to be indistinguishable. This is because we have not found any word that distinguishes them and by the result in Calude and Lipponen [4], to test the condition $R_A(q_i, w) \neq R_A(q_j, w)$ it is sufficient to consider all the words of length $|S_A| - 1$.

Define a new automaton $M(A) = (S_{M(A)}, \Delta_{M(A)}, F_{M(A)})$ such that $S_{M(A)} = \{[s] \mid s \in S_A\}$, and for all $[s] \in S_{M(A)}$,

$$F_{M(A)}([s]) = F_A(s),$$
$$\Delta_{M(A)}([s], w) = \begin{cases} [\Delta_A(s, w)], & \text{if } w \in W_A(s), \\ \infty, & \text{otherwise.} \end{cases}$$

Because of Lemma 2, we have indeed a well-defined automaton which is minimal and unique up to an isomorphism. Before proving the main theorem, we want to point out that if $S_A = \{s_1, s_2, \ldots, s_n\}$ then for each class $[s]$, $s \in S_A$, we can fix a unique representative of the class to be the state s_{rep} having the minimum index between all the states in the same class. This means that

a) there is $rep \in \{1, 2, \ldots, n\}$ such that $s_{rep} \equiv s$, and
b) for all $s_j \in S_A$ for which $s_j \equiv s$ we have $rep \leq j$.

Lemma 3. *Let $A = (S_A, \Delta_A, F_A)$ and $B = (S_B, \Delta_B, F_B)$ be two incomplete automata. If $W_A(p) = W_B(q)$ for some $p \in S_A$ and $q \in S_B$, then*

$$W_A(\Delta_A(p, w)) = W_B(\Delta_B(q, w))$$

for all $w \in W_A(p)$.

Proof. Assume that there is a word u such that $u \in W_A(\Delta_A(p, w))$ but $u \notin W_B(\Delta_B(q, w))$. By definition, this implies that $wu \in W_A(p) \setminus W_B(q)$, a contradiction. □

Theorem 4. *For every incomplete automaton A,*

1) *$M(A)$ and A are β–equivalent.*
2) *$M(A)$ is minimal.*
3) *If B and A are β–equivalent and B is minimal, then $M(A)$ and B are isomorphic.*

Proof. 1) We split the proof in two parts and first prove that $M(A)$ is β–simulated by A via a mapping $h : S_{M(A)} \to S_A$, where $h([s]) = s_{rep}$. The mapping h is well defined since for any $p, q \in S_A$ and $[p] = [q]$, we have $p_{rep} = q_{rep}$, so $h([p]) = h([q])$. Furthermore, h verifies the definition for β–simulation. Indeed, for all $[s] \in S_{M(A)}$ and $w \in \Sigma^*$, $f_{M(A)}([s], w) = F_{M(A)}(\Delta_{M(A)}([s], w)) = F_{M(A)}([\Delta_A(s, w)]) = F_A(\Delta_A(s, w)) = f_A(s, w) = f_A(h([s]), w)$. For the other part, let $g : S_A \to S_{M(A)}$ be a mapping such that $g(s) = [s]$. Then for all $p \in S_A$ and $w \in \Sigma^*$, $f_A(p, w) = F_A(\Delta_A(s, w)) = f_{M(A)}(g(p), w)$ which shows that A is β–simulated by $M(A)$.

2) To prove that $M(A)$ is minimal, let B be any automaton which is β–equivalent to A via the mappings $i : S_A \to S_B$ and $j : S_B \to S_A$. We will show that $M(A)$ has fewer states than B by showing that the mapping $l : S_B \to S_{M(A)}$ defined by $l(s) = [j(s)]$ is an onto function. Indeed, for any $[q] \in S_{M(A)}$, there is a state $i(q) \in S_B$ such that $l(i(q)) = [j(i(q))] = [q]$. This is because for all

$s \in S_A$, we have $s \equiv j(i(s))$, and for all $t \in S_B$, we have $t \equiv i(j(t))$. So l is an onto function from the finite set S_B to another finite set $S_{M(A)}$, and hence, $|S_{M(A)}| \leq |S_B|$.

3) Let A and B be β–equivalent via the mappings $i : S_A \to S_B$ and $j : S_B \to S_A$. We will show that the mapping $l : S_B \to S_{M(A)}$, used in 2) is an isomorphism between B and $M(A)$. By minimality of $M(A)$, $|S_{M(A)}| \leq |S_B|$, and by minimality of B, $|S_B| \leq |S_{M(A)}|$, so $|S_{M(A)}|$ and $|S_B|$ are equal. Since we have an onto mapping between two finite sets with the same cardinality, it follows that l is also one-to-one.

To conclude the proof we have to show that for all $s \in S_B$ and $\sigma \in \Sigma$, $l(\Delta_B(s,\sigma)) = \Delta_{M(A)}(l(s),\sigma)$ Since $M(A)$ is minimal, it is sufficient to prove that the states are indistinguishable in $M(A)$. First we notice that

$$f_B(s,w) = f_{M(A)}(l(s),w) \tag{1}$$

since according to the definition of l, $f_{M(A)}(l(s),w) = f_{M(A)}([j(s)],w)$ and by the construction of $M(A)$, $f_B(s,w) = f_A(j(s),w) = f_{M(A)}([j(s)],w)$.

Next we notice that

$$W_{M(A)}(\Delta_{M(A)}(l(s),\sigma)) = W_{M(A)}(l(\Delta_B(s,\sigma))).$$

Indeed, by equation (1), we have $W_{M(A)}(l(s)) = W_B(s)$ which, by Lemma 3 implies that $W_{M(A)}(\Delta_{M(A)}(l(s),\sigma)) = W_B(\Delta_B(s,\sigma))$, for all $\sigma \in W_{M(A)}(l(s))$, and again by equation (1), $W_B(\Delta_B(s,\sigma)) = W_{M(A)}(l(\Delta_B(s,\sigma)))$.

Finally, $l(\Delta_B(s,\sigma))$ and $\Delta_{M(A)}(l(s),\sigma)$ are indistinguishable in $M(A)$ since for all $w \in W_{M(A)}(l(\Delta_B(s,\sigma)))$,

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

□

Corollary 5. *For any two minimal incomplete automata A and B the following are equivalent:*

1) *A and B are β–equivalent.*
2) *A and B are isomorphic.*

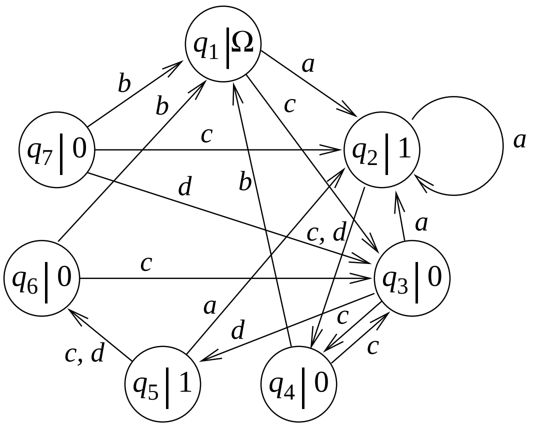
Since Ω has been treated just as an output symbol, the whole concept of output-incomplete Moore automaton can actually be embedded into an output-complete automaton studied in Calude and Lipponen [4]. For each output-incomplete automaton A there is a corresponding output-complete automaton A^c (where the symbol Ω is replaced by some specified symbol $x \notin O$) such that

A is minimal iff A^c is minimal over the alphabet $O \cup \{x\}$. On the other hand, every output-complete or complete automaton can be regarded as a subcase of the output-incomplete automata since a total function is a special case of a partial function. The minimal complete automaton obtained in this way is exactly the same as the one obtained in Calude, Calude and Khoussainov [1]. This guarantees that our model is consistent.

In Calude and Lipponen [4] we also proved that every finite class of pairs of input-incomplete automata and initial states (A_i, q_i) can be embedded into a finite class which has a unique minimal universal input-incomplete automaton (without initial states). By **universal automaton** we mean an automaton U_C whose behaviour, in a way, represents the behaviour of the class $C = \{(A_i, q_i)\}$. By this we mean that for each state $p \in S_U$ there is an initial state $q_i \in S_{A_i}$ such that p and q_i are indistinguishable, and vice versa, for each initial state q_i there is a corresponding state $p \in S_U$. We can prove that this kind of minimal universality can be obtained also for output-incomplete Moore automata.

3 Two models of minimal automata

In this section we will discuss some major differences between the model of minimal automaton presented in the previous section and the model in Mikolajczak [11]. To this aim we construct both the minimal automaton $M(A)$ and the minimal cover A_{min} for an automaton $A = (S_A, \Delta_A, F_A)$ presented in Figure 1.



	a	b	c	d	output
q_1	q_2	-	q_3	-	Ω
q_2	q_2	-	q_4	q_4	1
q_3	q_2	-	q_4	q_5	0
q_4	-	q_1	q_3	-	0
q_5	q_2	-	q_6	q_6	1
q_6	-	q_1	q_3	-	0
q_7	-	q_1	q_2	q_3	0

Figure 1: Automaton A and its transition table Δ_A

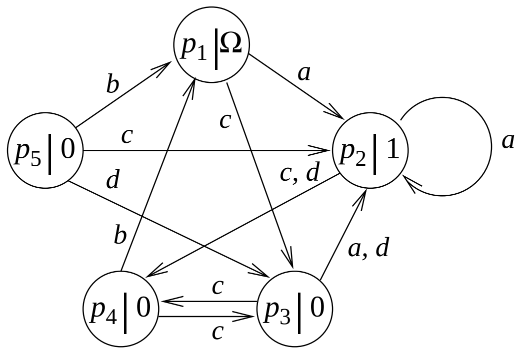
Minimal automaton $M(A)$

We will first construct the equivalence classes using the results and the algorithm of the previous section. If we go through the states $S_A = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$

we notice that Step 2 places $\times$ to the entries (q_1, q_2), (q_1, q_3), (q_1, q_4), (q_1, q_5), (q_1, q_6), (q_1, q_7), (q_2, q_3), (q_2, q_4), (q_2, q_6), (q_2, q_7), (q_3, q_5), (q_4, q_5), (q_5, q_6), (q_5, q_7) because of the difference between the outputs, for instance, $F_A(q_1) = \Omega$ and $F_A(q_5) = 1$, and to the entries (q_3, q_4), (q_3, q_6), (q_3, q_7), (q_4, q_7), (q_6, q_7) because of the difference between the transitions that are defined or undefined, for instance, $\Delta_A(q_3, b) = \infty$ and $\Delta_A(q_7, b) = q_1$. Step 3 places $\vee$ in the entry (q_4, q_6). Step 4 assigns the pair (q_4, q_6) to the remaining entry (q_2, q_5), and by Step 6, we place $\vee$ in the entry (q_2, q_5) since the entry (q_4, q_6) contains $\vee$. We have obtained the following table where each entry containing $\times$ contains, for clarification, also the word which distinguishes between the states unless the word is empty (the outputs of the states are different) in which case we do not write anything.

q_2	$\times$					
q_3	$\times$	$\times$				
q_4	$\times$	$\times$	$w = a$ $\times$			
q_5	$\times$	(q_4, q_6) $\vee$	$\times$	$\times$		
q_6	$\times$	$\times$	$w = a$ $\times$	$\vee$	$\times$	
q_7	$\times$	$\times$	$w = a$ $\times$	$w = d$ $\times$	$\times$	$w = d$ $\times$
	q_1	q_2	q_3	q_4	q_5	q_6

According to the table only the pairs (q_2, q_5) and (q_4, q_6) are indistinguishable. Hence we have five equivalence classes, denoted by $p_1 = [q_1]$, $p_2 = [q_2] = [q_5]$, $p_3 = [q_3]$, $p_4 = [q_4] = [q_6]$ and $p_5 = [q_7]$. The automaton $M(A)$ which by Theorem 4 is unique up to an isomorphism is presented in Figure 2.



	a	b	c	d	output
p_1	p_2	-	p_3	-	Ω
p_2	p_2	-	p_4	p_4	1
p_3	p_2	-	p_4	p_5	0
p_4	-	p_1	p_3	-	0
p_5	-	p_1	p_2	p_3	0

Figure 2: Minimal automaton $M(A)$

Minimal cover A_{min}

We will next construct the minimal cover A_{min} following the method in Mikolajczak [11] but without going very deeply into the details.

Two states $p, q \in S_A$ are said to be **incompatible** iff there is a (possibly empty) word $w \in \hat{W}_A(p) \cap \hat{W}_A(q)$ such that $f_A(p, w) \neq f_A(q, w)$. If p and q are not incompatible then they are said to be **compatible**. Notice that this relation is both reflexive and symmetric but not transitive so it cannot be an equivalence relation, and hence, does not generate a factor automaton. However, the relation induces a maximal cover on the set of states. The following table shows which states in A are compatible ($\vee$) and which are incompatible ($\times$). If an entry (p, q) contains a pair of states (r, s) then this means that p and q are incompatible only if r and s are incompatible.

	q_1	q_2	q_3	q_4	q_5	q_6
q_2	(q_3, q_4) $\vee$					
q_3	(q_3, q_4) $\vee$	$\times$				
q_4	$\vee$	$\times$	$\vee$			
q_5	(q_3, q_6) $\vee$	(q_4, q_6) $\vee$	$\times$	$\times$		
q_6	$\vee$	$\times$	(q_3, q_4) $\vee$	$\vee$	$\times$	
q_7	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$

The **maximal set of compatible states** T_{max} is a subset of the state set S_A such that every pair of its states is compatible, and adding an extra state s to the set T_{max} results that the set $T_{max} \cup \{s\}$ is not compatible. The maximal set of compatible states of A is

$$T_{max} = \{(q_1, q_2, q_5), (q_1, q_3, q_4, q_6), q_7\}.$$

We can now construct the family $A_{min} = \{A_k\}$ with minimal number of states. Let s_1 represent the states (q_1, q_2, q_5), s_2 the states (q_1, q_3, q_4, q_6), and s_3 the state q_7, respectively. The transition table Δ_{min} is defined from Δ_A as follows:

	a	b	c	d	output
s_1	s_1	$-$	s_2	s_2	1
s_2	s_1	s_1/s_2	s_2	s_1	0
s_3	$-$	s_1/s_2	s_1	s_2	0

Since there are two transitions where the next state has two possible outcomes, such as $\Delta(s_2, b) = s_1$ or $\Delta(s_2, b) = s_2$ and the automaton is deterministic, we have altogether $2 \times 2 = 4$ three-state automata in the family A_{min} and each A_k, $k = 1, \ldots, 4$, represents the automaton A. By our interpretation this means that for any state $q_i \in S_A$ and admissable word $w \in \hat{W}_A(q_i)$ there is a state s_j such that $f_A(q_i, w) = f_{A_k}(s_j, w)$, and vice versa, for any state s_j and an input $w \in W_A(s_j)$, there is a state $q_i \in S_A$ such that $w \in \hat{W}_A(q_i)$ and $f_{A_k}(s_j, w) = f_A(q_i, w)$.

Contrasting the models

The main differences between $M(A)$ and A_{min} can be found in the way they preserve the total responses and the undefined or unspecified behaviour of A, in other words, how much information we lose in the minimization process.

By Lemma 2, we may use equivalently total responses and final responses in constructing the minimal automaton $M(A)$. The minimal cover A_{min} has no such property. Each automaton A_k, $k = 1, \ldots, 4$, represents the automaton A with respect to final responses but not with respect to total responses. Assume that the transitions of automata A_k, $k = 1, \ldots, 4$, are the same as in Δ_{min} except that

$$\begin{aligned}
&\Delta_{A_1}(s_2, b) = s_1 = \Delta_{A_1}(s_3, b),\\
&\Delta_{A_2}(s_2, b) = s_1,\ \ \Delta_{A_2}(s_3, b) = s_2,\\
&\Delta_{A_3}(s_2, b) = s_2,\ \ \Delta_{A_3}(s_3, b) = s_1,\\
&\Delta_{A_4}(s_2, b) = s_2 = \Delta_{A_4}(s_3, b).
\end{aligned}$$

The response of the state q_7 to an input $w = bc$ is $R_A(q_7, bc) = 0\Omega 0$ while the responses of s_3 in automata A_k are

$$\begin{aligned}
R_{A_1}(s_3, b) &= 010 = R_{A_3}(s_3, b),\\
R_{A_2}(s_3, b) &= 000 = R_{A_4}(s_3, b).
\end{aligned}$$

So the unspecified output is interpreted into both 0 and 1. This is because the state q_1 with unspecified output is represented by both s_1 which emits an output 1 and s_2 which emits an output 0.

We also notice that the automaton A_k can respond to a given input sequence in the way that is not possible in the automaton A if we consider total responses. For instance, $R_{A_1}(s_2, cdc) = 0000$ but

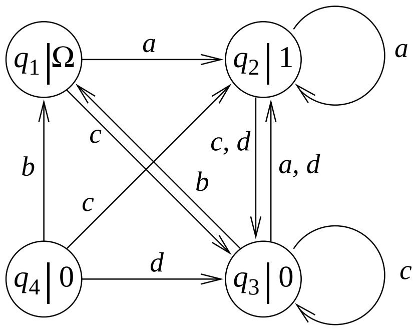
$$\begin{aligned}
&R_A(q_1, cdc) = \Omega 010,\\
&R_A(q_2, cdc) = \infty = R_A(q_5, cdc),\\
&R_A(q_3, cdc) = \infty,\\
&R_A(q_4, cdc) = 0010 = R_A(q_6, cdc),\\
&R_A(q_7, cdc) = 0100.
\end{aligned}$$

If we consider the uniqueness of the two models we first notice the following result which is a straight consequence of Theorem 4.

Corollary 6. *Two incomplete automata A_1 and A_2 are β–equivalent iff their minimal automata $M(A_1)$ and $M(A_2)$ are isomorphic.*

The result for the minimal cover A_{min} is much weaker. Indeed, the automaton B presented in Figure 3 has exactly the same minimal cover as the automaton A if we denote that s_1 represents the states (q_1, q_2), s_2 the states (q_1, q_3) and s_3 the state q_4, respectively. Because of this A and B produce the same final responses for the words that are admissable for both of them but we cannot say anything about other words. For instance, $w = dddd$ is admissable for the state q_2 in B since $\Delta_B(q_2, w) = q_2$ but w is not admissable for any state in A, $\Delta_A(q_i, w) = \infty$ for all $i = 1, \ldots, 7$. We also notice that the minimal automata $M(A)$ and $M(B) = B$ are not isomorphic. Hence if we consider any of the automata A_k, we cannot recover many features of the original automaton A.

The following result shows the relation between indistinguishability and compatibility.



	a	b	c	d	output
q_1	q_2	-	q_3	-	Ω
q_2	q_2	-	q_3	q_3	1
q_3	q_2	q_1	q_3	q_2	0
q_4	-	q_1	q_2	q_3	0

Figure 3: Automaton B whose cover B_{min} is isomorphic to A_{min}

Lemma 7. *Let A be an incomplete automaton. For any states $p, q \in S_A$, if p and q are indistinguishable then they are compatible, and if there is a third state $r \in S_A$ such that q and r are compatible then also p and r are compatible.*

Proof. For the first part, assume that p and q are incompatible. Then there is a word $w \in \hat{W}_A(p) \cap \hat{W}_A(q)$ such that $f_A(p, w) \neq f_A(q, w)$. But this means that w distinguishes between p and q.

For the second part, assume that p and q are indistinguishable and q and r are compatible. This means that for any word $w \in \hat{W}_A(q) \cap \hat{W}_A(r)$, the responses $f_A(q, w)$ and $f_A(r, w)$ are equal. But since $p \equiv q$, the sets $\hat{W}_A(p)$ and $\hat{W}_A(q)$ are the same and the responses are the same. Hence $w \in \hat{W}_A(p) \cap \hat{W}_A(r)$ and $f_A(p, w) = f_A(r, w)$. □

Lemma 7 implies that our method shortens the algorithm to construct the minimal cover A_{min}.

Theorem 8. *The minimal cover A_{min} is isomorphic to the minimal cover $M(A)_{min}$.*

Proof. By Lemma 7, the number of states in A_{min} is exactly the same as the number of states in $M(A)_{min}$, since any time p is represented by some state s of A_k also q is represented by s. So the pair (p, q) can be replaced everywhere by the class $[p]$ without affecting the number of the elements in the maximal set of compatible states T_{max}. On the other hand, since p and q are indistinguishable, they respond in exactly the same way for all input sequences. Hence the transitions of the states in A_{min} are isomorphic to the transitions in $M(A)_{min}$. □

To illustrate the previous proof, consider the automaton $M(A)$ in Figure 2. Then

$$T_{max} = \{(p_1, p_2), (p_1, p_3, p_4), p_5\},$$

and the transition table of $M(A)_{min}$ is exactly the same as Δ_{min} if we let s_1 represent the states (p_1, p_2), s_2 the states (p_1, p_3, p_4), and s_3 the state p_5, respectively.

Acknowledgement

We would like to thank Cris Calude for his various comments on earlier versions of the paper.

The first author was partially supported by AURC Grant A18/62090/F3414070. The second author is supported by Alfred Kordelin Foundation.

References

1. CALUDE, C., CALUDE, E., AND KHOUSSAINOV, B. Deterministic automata: simulation, universality and minimality. *Annals of Pure and Applied Logic 1105* (1997). in press.
2. CALUDE, C., CALUDE, E., SVOZIL, K., AND YU, S. Physical versus computational complementarity I. *International Journal of Theoretical Physics 36* (1997), 1495–1523.
3. CALUDE, C. S., AND LIPPONEN, M. Computational complementarity and sofic shifts. *CDMTCS Research Report 63* (1997).
4. CALUDE, E., AND LIPPONEN, M. Deterministic incomplete automata: simulation, universality and complementarity. *CDMTCS Research Report 39* (1997).
5. DVUREČENSKIJ, A., PULMANNOVÁ, S., AND SVOZIL, K. Partition logics, orthoalgebras and automata. *Helvetica Physica Acta 68* (1995), 407–428.
6. FEYNMAN, R. P. *The Character of Physical Law.* MIT Press, Cambridge, MA, 1965.
7. FINKELSTEIN, D., AND FINKELSTEIN, S. R. Computational complementarity. *International Journal of Theoretical Physics 22*, 8 (1983), 753–779.
8. GINSBURG, S. *An Introduction to Mathematical Machine Theory.* Addison Wesley, Reading Massachusetts, 1962.
9. GRIB, A. A., AND ZAPATRIN, R. R. Automata simulating quantum logics. *International Journal of Theoretical Physics 29*, 2 (1990), 113–123.
10. HOPCROFT, J. E., AND ULLMAN, J. D. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, MA, 1979.
11. MIKOLAJCZAK, B. Minimization of automata. In *Algebraic and Structural Automata Theory*, B. Mikolajczak, Ed. Elsevier Science Publishers, Amsterdam, 1991, pp. 83–118.
12. REUSCH, B., AND MERZENICH, W. Minimal coverings for incompletely specified sequential machines. *Acta Informatica 22* (1986), 663–678.
13. SCHALLER, M., AND SVOZIL, K. Partition logics of automata. *Il Nuovo Cimento 109B* (1994), 167–176.
14. SCHALLER, M., AND SVOZIL, K. Automaton partition logic versus quantum logic. *International Journal of Theoretical Physics 34*, 8 (August 1995), 1741–1750.
15. SCHALLER, M., AND SVOZIL, K. Automaton logic. *International Journal of Theoretical Physics 35*, 5 (May 1996), 911–940.
16. SVOZIL, K. *Randomness & Undecidability in Physics.* World Scientific, Singapore, 1993.