Automaton logic

M. Schaller and K. Svozil

Institut für Theoretische Physik, Technische Universität Wien

Wiedner Hauptstraße 8-10/136, A-1040 Vienna, Austria

e-mail: svozil@tph.tuwien.ac.at

Abstract

The experimental logic of Moore and Mealy type automata is investigated.

key words: automaton logic; partition logic; comparison to quantum logic; intrinsic measurements

1 Introduction

1.1 Motivation

Already in 1956, Moore [20] presented an explicit example of a four-state automaton featuring an "automaton uncertainty principle" at a very elementary level. The formalism introduced by Moore has been extended by Conway [6] and Chaitin [5], among others. See [14, 4] for a recent review on Moore and Mealy automata.

In an article entitled "computational complementarity", D. Finkelstein and S. R. Finkelstein [8] were the first to study the *experimental logic* of very general automata; i.e., the ordered structure of propositions arising from experiments on automata, and the relationship to quantum physics. Based on this research, Grib and Zapatrin [11, 12] investigated an automaton type, whose corresponding "macrostatements" (propositions about automaton ensembles), model arbitrary orthomodular lattices [13]. In another interesting development, Crutchfield [7] described the measurement process by introducing a hierarchy of automata.

This article goes back to Moore's original approach and deals with an algebraic characterization of the experimental logic of Moore and Mealy type automata.

1.2 Classical logic versus quantum logic versus automaton logic

In the followinf, we shall describe, in a somewhat simplified style, the construction of the logic calculus of classical physical systems, quantum systems and automata.

Let $\mathfrak{S}$ be a classical system. We denote the set of all observables of the system by $(A_i)_{i\in I}$. It is characteristic for classical systems that all $(A_i)_{i\in I}$ are simultaneously measurable. We denote the outcome of such a measurement by $(x_i)_{i\in I}$. The set of all possible outcomes forms the observation space O. The most general form of a prediction concerning $\mathfrak{S}$ is that the point $(x_i)_{i\in I}$ determined by actually measuring $(A_i)_{i\in I}$, will lie in a subset S of O. We may call the subsets of O the "experimental propositions" concerning $\mathfrak{S}$. These subsets form a Boolean algebra (which is equal to the power set of O). Associated with the system $\mathfrak{S}$ is the phase space Γ. According to the concept of a phase space, the state of $\mathfrak{S}$ is represented by a point $p \in \Gamma$, which determines the outcome of the measurements $(A_i)_{i\in I}$ in a deterministic way. We may assume a mapping $f : \Gamma \to O$, which describes this correspondence. Each experimental proposition $\mathfrak{S}$ corresponds to a subset Γ_S of Γ by $\Gamma_S = f^{-1}(S)$. These subsets Γ_S form the propositional calculus of the system $\mathfrak{S}$, which is also a Boolean algebra [using $f^{-1}(S\cup T) = f^{-1}(S)\cup f^{-1}(T)$, $f^{-1}(S\cap T) = f^{-1}(S)\cap f^{-1}(T)$ and $f^{-1}(S') = (f^{-1}(S))'$].

The situation in quantum mechanics is as follows. Let $\mathfrak{S}$ be a quantum system and let $(B_j)_{j\in J}$ be a set of compatible measurements. The experimental propositions concerning the measurement of $(B_j)_{j\in J}$ are again subsets of the observation space O_J of all possible outcomes $(x_j)_{j\in J}$ (now, O_J depends on the set $(B_j)_{j\in J}$). According to the quantum mechanical formalism, the subsets Γ_S of the phase space Γ have to be replaced by closed subspaces of an appropriate Hilbert space H (or equivalently, by projections operators p_S of H). The set $L(H)$ of all closed subspaces is called the propositional calculus (quantum logic) of the system $\mathfrak{S}$. $L(H)$ forms a complete atomistic orthomodular lattice (cf. [17, 23, 24]). The story of quantum logic goes back to the seminal paper of Birkhoff and von Neumann [1]. The interest in quantum logic was revived through the investigations of Jauch [16] and Piron [22]. The historical development and the different approaches to quantum logic can be found in [15].

At last we turn to automata logic. An automaton (Mealy or Moore automaton) is a finite deterministic system with input and output capabilities. At any time the automaton is in a state q of a finite set of states Q. The state determines the future input–output behavior of the automaton. If an input is applied, the automaton assumes a new state, depending both on the old state and on the input. An output is emitted which depends on the old state and the input (Mealy automaton) or only on the new state (Moore automaton). Automaton experiments are conducted by applying an input sequence and observing the output sequence. The automaton is thereby treated as a black box with known description but unknown initial state. Let E be an automaton experiment and let O_E be the observation space, i.e., O_E is the set of all possible outcomes of E. Because of the deterministic nature of the automaton, for every experiment E there exists a mapping $\lambda_E : Q \to O_E$, determining the outcome of E, and depending on the initial state of the automaton. As in the classical and quantum case, experimental propositions concerning the experiment E are subsets S_E of O_E. For every experiment E, the inverse images of the sets S_E under λ_E forms a Boolean algebra (more exactly, a field of sets). The elements of this Boolean algebra are subsets of the state set Q. We obtain a propositional calculus, termed the automaton logic, if we "paste" all Boolean algebras corresponding to all experiments together. This calculus forms a partition logic [27, 25]. Intuitively, as has already been observed by Moore [20], it may occur that the automaton undergoes an irreversible state change, i.e., information about the automaton's initial state is lost. A second, later experiment may therefore be affected by the first experiment, and vice versa. Hence, both experiments are incompatible. In this setup, the observer has a qualifying influence on the measurement result insofar as a particular observable has to be chosen among a class of non-co-measurable observables. But the observer has no quantifying influence on the measurement result insofar as the outcome of a particular measurement is concerned.

2 Orthomodular Posets

The appropriate algebraic structures to describe the logic of automata are found in the theory of orthomodular posets. Orthomodular structures arose from lattice theory [2, 10, 28] and quantum logic [1, 9]. The basic notion of orthomodular posets will be defined first. Then, a new type of logic, termed partition logic, will be introduced. We shall prove a representation theorem, which identifies certain

orthomodular posets with partition logics. Some examples of the new concepts will be given. More detailed introductions into the theory of orthomodular structures can be found in the book of G. Kalmbach [17] and in the book of P. Pták and S. Pulmannová [24]. The books by J. Jauch [16] and C .Piron [22], among others, deal with physical applications, mainly in the context of quantum mechanics.

2.1 Basic Definitions

Definition 2.1.1 *An orthomodular poset (OMP) is a set L endowed with a partial order $\leq$ and a unary operation $'$, called the orthocomplement, such that the following conditions for all $a, b \in L$ are satisfied:*

(i) L possesses a least and a greatest element 0 and 1, and $0 \neq 1$;

(ii) $a \leq b$ implies $b' \leq a'$;

(iii) $(a')' = a$;

(iv) if $a \leq b'$, then the supremum $a \vee b$ exists;

(v) if $a \leq b$, then $b = a \vee (a' \wedge b)$ (orthomodular law).

The symbols $\vee, \wedge$ denote the lattice-theoretic operations induced by $\leq$. If an OMP is an lattice, we call it an *orthomodular lattice* (OML). An OMP L does neither have to be distributive nor a lattice. On the other hand, de Morgan's law is valid in L: If $a \vee b$ exists in L, then $a' \wedge b'$ exists also and $a' \wedge b' = (a \vee b)'$ [use condition (ii)]. In particular, $1' = 0$ and $0' = 1$. Moreover, condition (v) yields $a \vee a' = 1$ for any $a \in L$ (and, dually, we also have $a \wedge a' = 0$ for any $a \in L$). The *orthogonality relation* $\perp$ for elements a, b of an OMP L is defined by

$a \perp b$ (a is orthogonal to b) if $a \leq b'$

holds. A pair $a, b \in L$ is called *compatible*, denoted by $a \leftrightarrow b$, if there exist three mutually orthogonal elements a_1, b_1, c such that $a = a_1 \vee c$ and $b = b_1 \vee c$.

We now exhibit some basic examples of OMPs. Every Boolean algebra is an OMP. The lattice $L(H)$ of all projection operators on a (real or complex) Hilbert space H (or, equivalently, the lattice of all closed subspaces of H) is an OMP, with the relation $\leq$ given by the inclusion and with the operation $'$ given by the formation of the orthocomplement in H.

Definition 2.1.2 *A subset M of L is called a sub-OMP of L if the following conditions are satisfied:*

(i) $0 \in M$;

(ii) if $a \in M$, then $a' \in M$;

(iii) if $a, b \in M$ and $a \perp b$, then $a \vee b \in M$ (the supremum is taken in L).

The sub-OMP ΓA generated by an arbitrary subset A of L is the smallest sub-algebra of L containing A; it always exists.

Definition 2.1.3 *Let L_1, L_2 be OMPs. A mapping $f : L_1 \to L_2$ is called a morphism (of OMPs) if the following conditions are satisfied:*
(i) $f(0) = 0$;
(ii) $f(a') = f(a)'$;
(iii) if $a \perp b$, then $f(a \vee b) = f(a) \vee f(b)$
A morphism $f : L_1 \to L_2$ is called an isomorphism (of OMPs) if f is injective, maps L_1 onto L_2 and the mapping f^{-1} is also a morphism.

Lemma 2.1.4 *A bijective mapping $f : L_1 \to L_2$ is an isomorphism iff the following conditions are satisfied:*
(i) $f(a') = f(a)'$;
(ii) $a \leq b$ iff $f(a) \leq f(b)$.

Proof. (i) If f is a morphism, f preserves the order. If $a \leq b$ for $a, b \in L_1$, then the orthomodular law yields $b = a \wedge c$ for a $c \in L_1$ such that $c \perp a$. Then, $f(b) = f(a) \wedge f(c)$, and, therefore, $f(a) \leq f(b)$. For an isomorphism f, f and f^{-1} are morphism, hence we get condition (ii).
(ii) The converse direction is trivial.

We shall prove three lemmas about the compatibility relation. The propositions and their proofs are taken from [24].

Lemma 2.1.5 *Let L be an OMP and $a, b \in L$:*
(i) If $a \perp b$, then $a \leftrightarrow b$;
(ii) If $a \leq b$, then $a \leftrightarrow b$;
(iii) If $a \perp b$, then $b = (a \vee b) \wedge a'$.

Proof. (i) Since $a \perp b, b \perp 0$ and $0 \perp a$, we can write $a = a \vee 0$ and $b = b \vee 0$.
(ii) According to the orthomodular law we can write $b = a \vee c$ for $c = b \wedge a'$. Therefore, $c \perp a$ and we have $a = a \vee 0$ and $b = b \vee c$.
(iii) $a \perp b$ implies $b' = a \vee (a' \wedge b)$ according to the orthomodular law. Forming the orthocomplement and using De Morgan's law, we obtain $b = (a \vee b) \wedge a'$.

Lemma 2.1.6 *Suppose that $a \leftrightarrow b$. Then, every pair in the set $\{a, a', b, b'\}$ is compatible.*

Proof. It suffices to prove that the assumption $a \leftrightarrow b$ implies $a' \leftrightarrow b$. The other assertions are not difficult to prove. Suppose that $a \leftrightarrow b$. Then, $a = a_1 \vee c$ and $b = b_1 \vee c$, where a_1, b_1, c are mutually orthogonal in L. Since $a \perp b_1$, we have $(a \vee b_1) \wedge b_1' = a$ (Lemma 2.1.5.iii). Thus, $a' = (a \vee b_1)' \vee b_1$. We need to check that the elements c, b_1 and $(a \vee b_1)'$ are mutually orthogonal. This is the case, since $b_1 \leq a \vee b_1 = ((a \vee b_1)')'$ and $c \leq a \vee b_1 = ((a \vee b_1)')'$.

Lemma 2.1.7 *If $a, b \in L$ and $a \leftrightarrow b$, then $a \vee b$ and $a \wedge b$ exist in L. Moreover, if $a = a_1 \vee c$ and $b = b_1 \vee c$ for mutually orthogonal elements a_1, b_1, c, then $a \vee b = a_1 \vee b_1 \vee c$ and $a \wedge b = c$. Further, we have $a_1 = a \wedge b', b_1 = b \wedge a'$. Hence, the elements a_1, b_1, c are uniquely determined by a and b.*

Proof. Since a_1, b_1, c are mutually orthogonal, we find that the supremum $a_1 \vee b_1 \vee c$ exists in L. Furthermore, $a \leq a_1 \vee b_1 \vee c$ and $b \leq a_1 \vee b_1 \vee c$. Let e be an element of L such that $a \leq e$ and $b \leq e$. The inequalities $a_1 \vee c \leq e, b_1 \vee c \leq e$ imply $(a_1 \vee c) \vee (b_1 \vee c) \leq e$. Hence, $a_1 \vee b_1 \vee c$ is the supremum of a, b. The existence of $a \wedge b$ follows from Lemma 2.1.6 and from the equality $a \wedge b = (a' \vee b')'$. We now show that $a \wedge b = c$. On the one hand, $c \leq a \wedge b$. On the other hand, $a \wedge b = (a_1 \vee c) \wedge b \leq (b' \vee c) \wedge b = c$ (Lemma 2.1.5.iii). Finally, we have $a_1 \leq a$ and $a_1 \leq b'$. Moreover, by Lemma 2.1.6, $a \wedge b'$ exists in L. Thus, $a_1 \leq a \wedge b'$. Furthermore, $a \wedge b' = (a_1 \vee c) \wedge b' \leq (a_1 \vee c) \wedge c' = a_1$. The proof of the equality $b_1 = b \wedge a'$ is similar.

2.2 Ideals and States

The definition of an ideal is similar to the definition of a lattice ideal. Additionally we require in condition (ii) that the elements a and b have to be orthogonal.

Definition 2.2.1 *Let L be an OMP. A nonvoid subset I of L is called an ideal if it satisfies the following conditions:*
(i) $a \in I, b \leq a$ imply $b \in I$;
(ii) $a, b \in I, a \perp b$ imply $a \vee b \in I$.

Definition 2.2.2 *An ideal P of L, $P \neq L$ is called prime if $a \perp b$ implies $a \in P$ or $b \in P$.*

We denote by $P(L)$ the set of all prime ideals of L. Let $\mathfrak{P}(P(L))$ the power set of $P(L)$). We define a mapping $p : L \to \mathfrak{P}(P(L))$ by $p(a) = \{P \in P(L) \mid a \notin P\}$. p is called the *p–function*.

Lemma 2.2.3 *Let P, $P \neq L$, be an ideal. The following conditions are equivalent:*
(i) P is a prime ideal;
(ii) $a \wedge b \in P$ and $a \leftrightarrow b$ imply $a \in P$ or $b \in P$;
(iii) $a \in P$ iff $a' \notin P$.

Proof. (i) implies (ii). Since $a \leftrightarrow b$, there exist three mutually orthogonal elements a_1, b_1, c such that $a = a_1 \vee c$ and $b = b_1 \vee c$. From Lemma 2.1.7 we know that $c = a \wedge b$. $a_1 \perp b_1$ implies $a_1 \in P$ or $b_1 \in P$. Therefore, according to the definition of an ideal, $a \in P$ or $b \in P$.

(ii) implies (iii). We remark that $0 = a \wedge a' \in P$. We know from Lemma 2.1.6 that $a \leftrightarrow a'$. Hence, $a \in P$ or $a' \in P$. If both a and a' are in P, then also $1 = a \vee a' \in P$ and $P = L$, which contradicts our assumption $P \neq L$.

(iii) implies (i). Let $a, b \in L$ and $a \perp b$. We have to prove that $a \in P$ or $b \in P$. If $a \in P$ the condition is satisfied. Let us assume $a \notin P$. It follows that $a' \in P$. Since $b \leq a'$, we obtain $b \in P$ according to the definition of an ideal.

Remark: Compared to lattice prime ideals, the compatibility of the elements a and b is required additionally.

Lemma 2.2.4 *The p–function possesses the following properties:*
(i) $p(0) = \emptyset$;
(ii) $p(a') = p(a)'$;
(iii) if $a \perp b$, then $p(a \vee b) = p(a) \cup p(b)$;
(iv) if $a \leq b$, then $p(a) \subseteq p(b)$.

Proof. (i) $p(0) = \{P \in P(L) \mid 0 \notin P\} = \emptyset$;
(ii) $p(a') = \{P \in P(L) \mid a' \notin P\} =$
$= \{P \in P(L) \mid a \in P\}$ (using Lemma 2.2.3) $= P(L) \setminus p(a)$;
(iii) $P \in p(a \vee b) \Leftrightarrow a \vee b \notin P \Leftrightarrow a \notin P$ or $b \notin P \Leftrightarrow$
$\Leftrightarrow P \in p(a)$ or $P \in p(b) \Leftrightarrow P \in p(a) \cup p(b)$;
(iv) Let $a \leq b$ and $P \in p(a)$. Then $a \notin P$, and this implies $b \notin P$ according to the definition of an ideal.

Definition 2.2.5 *A state (i.e.,. a two-valued state) on an OMP L is mapping $s : L \to [0,1]$ (i.e.,. a mapping $s : L \to \{0,1\}$) such that*
(i) $s(1) = 1$;
(ii) if $a \perp b$, then $s(a \vee b) = s(a) + s(b)$.

States are probability measures on an OMP. The definition is not exact when applied to infinite OMPs (cf. [24]), but in this work we only deal with finite OMPs. Furthermore, in what follows we need the concept of two-valued states, which is strongly connected to the definition of a prime ideal. We denote by $S(L)$ the set of all two-valued states on L.

Lemma 2.2.6 *Suppose that $a,b \in L$ and $a \leq b$. Then, $s(a) \leq s(b)$ for any $s \in S(L)$.*

Proof. Using the orthomodular law, we can write $b = a \vee (b \wedge a')$, and therefore $s(b) = s(a) + s(b \wedge a') \geq s(a)$.

Lemma 2.2.7 *(i) Let P be a prime ideal. Define a mapping $s : L \to \{0,1\}$ by*

$$s(x) = \begin{cases} 0, \text{ if } x \in P \\ 1, \text{ if } x \notin P \end{cases}$$

Then, s is a two-valued state.

(ii) Let s be a two-valued state. Set $P = \{x \in L \mid s(x) = 0\}$. Then, P is a prime ideal.

Proof. (i) Since $1 \notin P$, the condition $s(1) = 1$ is satisfied. Suppose that $a \perp b$. We have to show that $s(a \vee b) = s(a) + s(b)$. We first assume that $a \vee b \in P$. Then also $a,b \in P$, and we obtain $0 = s(a \vee b) = s(a) + s(b) = 0 + 0 = 0$. Let us now assume that $a \vee b \notin P$. According to the definition of a prime ideal, one of both elements a,b has to be in P. If both a and b are in P, then also $a \vee b \in P$, which contradicts our assumption. Thus, we obtain $1 = s(a \vee b) = s(a) + s(b) = 1 + 0$.

(ii) At first we prove that P is an ideal. Let $a \in P$ and $b \leq a$. Since $b \leq a$, we know by Lemma 2.2.6 that $s(b) \leq s(a)$. Together, we obtain $s(b) = 0$; hence $b \in P$. Let us now assume that $a,b \in P$ and $a \perp b$. We have $s(a \vee b) = s(a) + s(b) = 0 + 0 = 0$. Therefore, we obtain $a \vee b \in P$. Finally we have to show that P is prime. Let a,b two elements of L such that the relation $a \perp b$ holds. We have $s(a \vee b) = s(a) + s(b) \leq 1$. Hence, $s(a) = 0$ or $s(b) = 0$ and therefore $a \in P$ or $b \in P$.

As we shall see later, the set of all prime ideals $P(L)$ (i.e.,. the state space $S(L)$) can be very poor (in the extreme case it can be empty). It seems therefore useful to distinguish the cases when $P(L)$ is relatively big.

Definition 2.2.8 *(i) An OMP L is called rich if the following implication holds:*
$\{P \in P(L) \mid a \notin P\} \subseteq \{P \in P(L) \mid b \notin P\}$ *implies* $a \leq b$.
(ii) An OMP L is called prime if for all $a, b \in L, a \neq b$ there exist a prime ideal $P \in P(L)$ containing exactly one of both a and b.

Using the p–function, we can write:
(i) L is rich if $p(a) \subseteq p(b)$ implies $a \leq b$; and
(ii) L is prime if $a \neq b$ implies $p(a) \neq p(b)$.

Lemma 2.2.9 *Every rich OMP is prime.*

Proof. Let L be a rich OMP. For all $x \in L$ we set $p(x) = \{P \in P(L) \mid x \notin P\}$. Let a, b be two arbitrary elements of L. If $p(a) = p(b)$ then $a = b$ by the richness of L. Therefore, for $a \neq b$ also $p(a) \neq p(b)$, and a prime ideal containing exactly one of both a and b exists.

2.3 Concrete Logics and Partition Logics

Definition 2.3.1 *A concrete logic is a pair (Ω, Δ), where Ω stands for a set and Δ stands for a collection of subsets of Ω satisfying:*
(i) $\emptyset \in \Delta$;
(ii) if $A \in \Delta$ then $\Omega \backslash A \in \Delta$;
(iii) if $A, B \in \Delta$ and $A \cap B = \emptyset$, then $A \cup B \in \Delta$.

A routine check of the axioms (i)-(v) in definition 2.3.1 shows that a concrete logic becomes an OMP if we take the set inclusion for the relation $\leq$ and the set complement for the orthocomplement $'$.

A simple example of a concrete logic is a pair (Ω, Δ), where Ω is a finite set of even cardinality and Δ is the collection of all subsets of Ω with an even number of elements.

Theorem 2.3.2 *(Gudder) An OMP L is isomorphic to a concrete logic iff L is rich.*

Proof. (i) Suppose first that L is isomorphic to a concrete logic (Ω, Δ). We may assume that $L = \Delta$. Take $A, B \in \Delta$ such that $A \not\leq B$. We have to prove that $p(A) \not\subseteq p(B)$. $A \not\leq B$ implies $A \backslash B \neq \emptyset$ and therefore we can choose a point $q \in A \backslash B$.

Put $P = \{C \in \Delta \mid q \notin C\}$. A routine check verifies that P is a prime ideal. From the definition of P it follows that $P \in p(A)$, but $P \notin p(B)$.

(ii) Conversely, suppose that L is rich. Put $\Omega = P(L)$ and put $\Delta = \{p(a) \mid a \in L\}$, where p denotes the the p–function. Let us show that (Ω, Δ) is a concrete logic. From Lemma 2.2.4.i,ii, we know that $p(0) = \emptyset \in \Delta$ and that $p(a)' = P(L) \backslash p(a) \in \Delta$ for any $p(a) \in \Delta$. Now, let $p(a)$ and $p(b)$ be orthogonal elements of Δ. Since $p(a) \cap p(b) = 0$, we obtain $p(a) \subseteq P(L) \backslash p(b) = p(b')$. By the richness of L we conclude that $a \perp b$. Hence, $a \vee b$ exists and $p(a) \cup p(b) = p(a \vee b) \in \Delta$ (using 2.2.4.iii), proving that (Ω, Δ) is indeed a concrete logic. From lemma 2.2.4.iv we know that $a \leq b$ implies $p(a) \subseteq p(b)$. Conversely, from the richness of L we know that $p(a) \subseteq p(b)$ implies $a \leq b$. Hence, by Lemma 2.1.4, the mapping $p : L \to \Delta$ is an isomorphism.

Remark: Theorem 2.3.2 and Theorem 2.3.9 are related to the Birkhoff-Stone representation theorem for distributive lattices and Boolean algebras, respectively.

Definition 2.3.3 *A relation $\approx$ on a set M is called an equivalence relation if it satisfies the following conditions for all $a, b, c \in M$:*

(i) $a \approx a$ (reflexivity);

(ii) $a \approx b$ implies $b \approx a$ (symmetry);

(iii) $a \approx b$ and $b \approx c$ implies $a \approx c$ (transitivity).

Definition 2.3.4 *Let M be a set. A collection $\mathfrak{A}$ of subsets of M is called a partition of M if it has the following properties:*

(i) $A \cap B = \emptyset$ or $A = B$ for all $A, B \in \mathfrak{A}$;

(ii) $\bigcup \mathfrak{A} = \bigcup_{A \in \mathfrak{A}} A = M$.

Let $\approx$ be an equivalence relation on M and let $a \in M$. The *equivalence class of a modulo $\approx$* is the set $[a] = \{b \in M \mid a \approx b\}$. The set of all equivalence classes modulo $\approx$, written $M/\approx$, is called the *quotient set of M by $\approx$*. It is easy to check that $M/\approx$ forms a partition of M, called the *partition induced by $\approx$*, or the *partition corresponding to $\approx$*. Let $\mathfrak{A}$ and $\mathfrak{B}$ be partitions of a set M. We call $\mathfrak{A}$ *finer* than $\mathfrak{B}$ or say that $\mathfrak{A}$ is a *refinement* of $\mathfrak{B}$ if for every $A \in \mathfrak{A}$ there exists a $B \in \mathfrak{B}$ such that $A \subseteq B$.

The following definition of the pasting technique is due to Navara and Rogalewicz [21].

Definition 2.3.5 *Let $\mathfrak{L}$ be a family of OMPs satisfying the following condition:*

For all $P,Q \in \mathfrak{L}$, $P \cap Q$ is a sub-OMP of both P and Q, and the partial orderings and the orthocomplementations coincide on $P \cap Q$.

Define on the set $L = \bigcup \mathfrak{L} = \bigcup_{P \in \mathfrak{L}} P$ a relation $\leq$ and a unary operation $'$ as follows:

(i) $a \leq b$ iff there exists a $P \in \mathfrak{L}$ such that $a,b \in P$ and $a \leq_P b$;

(ii) $a' = b$ iff there exists a $P \in \mathfrak{L}$ such that $a,b \in P$ and $a^{\prime P} = b$

(the indices indicate that the operations belong to the respective OMP). The set L together with $\leq$ and $'$ is called the pasting *of the family $\mathfrak{L}$.*

Let P be a partition of a set M. The *Boolean algebra generated by P* is the set $B_P = \{\bigcup S = \bigcup_{A \in S} A \mid S \subseteq P\}$, together with the inclusion and the complement.

Definition 2.3.6 (Partition logic) *Let $\mathfrak{R}$ be a family of partitions of a set M. The pasting of the Boolean algebras $B_R, R \in \mathfrak{R}$, is called a partition logic, denoted by $(M,\mathfrak{R})$.*

It follows from the definition that the orthocomplement A' of a partition logic $(M,\mathfrak{R})$ is identical with the set complement $M \backslash A$. Further $A \leq B$ implies $A \subseteq B$. The converse is in the general not true.

Lemma 2.3.7 *A partition logic $P = (M,\mathfrak{R})$ is an OMP iff the following conditions are satisfied:*

(i) the relation $\leq$ is transitive;

(ii) if $A \perp B$ ($A \leq B'$), then the supremum $A \vee B$ exists.

Proof. (i) If P is an OMP, the two conditions are satisfied.

(ii) Let $P = (M,\mathfrak{R})$ be a partition logic satisfying the two conditions. Let $A \in P$. Then there exists a $R \in \mathfrak{R}$ such that $A \in B_R$. In B_R we have $A \leq_{B_R} A$ and therefore also $A \leq A$, hence, the relation $\leq$ is reflexive. Let $A,B \in P$ and assume $A \leq B, B \leq A$. We obtain $A \subseteq B$ and $B \subseteq A$, obtaining $A = B$. Hence, $\leq$ is also symmetric. Taking also condition (i) into account, we showed that $\leq$ is an order relation. The axioms (i)-(iii) of definition 2.1.1 follow trivially. Axiom (iv) is identical with condition (ii). Let $A,B \in P$ and $A \leq B'$. Then, there exist a $R \in \mathfrak{R}$, such that $A,B \in B_R$, and therefore also $A \cup B \in B_R$ and $A,B \leq A \cup B$. By condition (ii) the supremum $A \vee B$ exists. We obtain $A,B \subseteq A \vee B \subseteq A \cup B$, which implies $A \vee B = A \cup B$. In the same way, if $A' \leq B$ holds, we obtain $A \wedge B = A \cap B$.

Let $A \leq B$. Then, $A \vee (A' \wedge B) = A \cup (A' \cap B) = B$, satisfying the orthomodular law.

A *block* of an OMP L is a maximal Boolean subalgebra of L. Every element x of L is contained in at least one block, since the Boolean subalgebra generated by x (and consisting of $x, x', 0, 1$) can be embedded into a maximal one. We denote the set of all blocks of an OMP L by $\mathfrak{B}(L)$.

Theorem 2.3.8 *Let L be an OMP. Then, L is the pasting of its blocks $\mathfrak{B}(L)$.*

For a proof see Navara and Rogalewicz [21].

Let L be an OMP and let $x, y \in L$ with $x \leq y$. The sub-OMP $\Gamma\{x, y\}$ generated by the set $\{x, y\}$ is equivalent to the set $\{0, 1, x, x', y, y', x' \wedge y, x \vee y'\}$ (some of the elements may coincide, for instance, if $x = 0$ or $x = y$). Moreover, $\Gamma\{x, y\}$ is a Boolean algebra. We put $\mathfrak{C}(L) = \{\Gamma\{x, y\} \mid x, y \in L \text{ and } x \leq y\}$. L is the pasting of the family $\mathfrak{C}(L)$.

Theorem 2.3.9 *An OMP L is isomorphic to a partition logic iff L is prime.*

Proof. (i) The proof is analogous to the proof of theorem 2.3.2.i. First, suppose that L is isomorphic to a partition logic $R = (M, \mathfrak{R})$. We may assume that $L = R$. Take $A, B \in R$ such that $A \neq B$. $A \neq B$ implies $(A \backslash B) \cup (B \backslash A) \neq \emptyset$ and therefore we can choose a point $q \in (A \backslash B) \cup (B \backslash A)$. Put $P = \{C \in R \mid q \notin C\}$. A routine check verifies that P is a prime ideal. From the definition of P it follows that exactly one of both A, B is element of P. Therefore L is prime.

(ii) Conversely, suppose that L is prime. Put $M = P(L)$. Let $\Gamma \in \mathfrak{C}(L)$. Define a partition R_Γ of $P(L)$ by $R_\Gamma = \{p(a) \mid a \text{ is atom of } \Gamma\}$ (p is the p–function). It follows from Lemma 2.2.4 that R_Γ is indeed a partition of $M = P(L)$. Put $\mathfrak{R} = \{R_\Gamma \mid \Gamma \in \mathfrak{C}(L)\}$ and let R be the partition logic $(M, \mathfrak{R})$. We propose that $p : L \to R$ is an isomorphism. p is injective by the primeness of L, p is surjective by the construction of R. For every $\Gamma \in \mathfrak{C}(L)$, the restriction of p to Γ, $p \mid \Gamma : \Gamma \to R_\Gamma$, is an isomorphism. Since by Lemma 2.1.4, L is the pasting of $\mathfrak{C}(L)$ and R is the pasting of $\mathfrak{R}$, L and R are also isomorphic.

Remark: If every element of L can be written as a supremum of a finite set of atoms, we may also use the family $\mathfrak{B}(L)$ instead of the family $\mathfrak{C}(L)$.

Corollary 2.3.10 *Every concrete logic is a partition logic.*

Proof. The proposition follows from Lemma 2.2.9, Theorem 2.3.2 and Theorem 2.3.9.

We shall see later that there exist OMPs which are prime but not rich. The class of concrete logics is therefore a proper subclass of the class of partition logics.

2.4 Greechie logics

In this part we introduce a technique to design OMPs with special properties.

Definition 2.4.1 *Let $\mathfrak{B}$ be a system of Boolean algebras. We say that $\mathfrak{B}$ is almost disjoint if for any pair $A, B \in \mathfrak{B}$ at least one of the following conditions is satisfied:*
(i) $A = B$;
(ii) $A \cap B = \{0,1\}$;
(iii) $A \cap B = \{0,1,x,x'\}$,
where x is an atom in both A and B. Moreover, $x' = x'_A = x'_B$.

Definition 2.4.2 *Let $\mathfrak{B}$ be an almost disjoint system of Boolean algebras. A finite sequence $(B_0, B_1, \ldots, B_{n-1})$ of elements of $\mathfrak{B}$ is called a loop of order n if the following conditions are satisfied (the computation of the i, j, k is modulo n):*
(i) $B_i \cap B_{i+1} = \{0,1,x_i,x'_i\}$ for $0 \leq i \leq n-1$;
(ii) $B_i \cap B_j = \{0,1\}$ for $j \neq i-1, i, i+1$;
(iii) $B_i \cap B_j \cap B_k = \{0,1\}$ for distinct indices i, j, k.

Observe that every loop $(B_0, B_1, \ldots, B_{n-1})$ uniquely determines a sequence of atoms $(e_0, \ldots, e_{n-1})$ such that e_i is the common atom of B_i and B_{i+1}.

Lemma 2.4.3 *Let $\mathfrak{B}$ be an almost disjoint system of Boolean algebras and let L be the pasting of $\mathfrak{B}$. Then, $\leq$ is a partial order and the operation $'$ is an orthocomplementation.*

For a proof see [17, 24].

Theorem 2.4.4 *(Greechie) Let $\mathfrak{B}$ be an almost disjoint system of Boolean algebras and let L be the pasting of $\mathfrak{B}$. Then,*
(i) L is an OMP iff $\mathfrak{B}$ does not contain a loop of order 3.
(ii) L is an OML iff $\mathfrak{B}$ does not contain a loop of order 3 or 4.

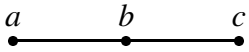


Figure 1:

Figure 2:

For a proof see [17, 24].

Definition 2.4.5 *An OMP is called a Greechie logic if the following conditions are satisfied:*

(i) every element of L can be written as supremum of at most countably many mutually orthogonal atoms in L;

(ii) the collection of all blocks in L forms an almost disjoint system.

A useful way of exhibiting the Greechie logics is the drawing of Greechie diagrams. Let L be a logic and $\mathfrak{B}$ be a system of blocks of it. Then, $L = \bigcup \mathfrak{B}$. The Greechie diagram associated with L consist of a set of points and a set of lines. The points are in one-to-one correspondence with the atoms of L; the lines are in one-to-one correspondence with the blocks of L.

For instance, the drawing in Fig. 1 represents the Boolean algebra $\mathbf{2}^3$ with the atoms a, b and c. If L is not a Boolean algebra, then it contains several blocks which may or may not have atoms in common. If two distinct blocks drawn in Fig. 2 of L have exactly one atom c in common, then the corresponding edges have a corner at c. For instance, the Greechie diagram drawn in Fig. 3 corresponds to the Hasse diagram drawn in Fig. 4.

Note that the Greechie diagrams allow us to detect the presence of the loops of order 3 or 4 and, therefore, indicate whether the structure in question is or is not an OMP (OML). A loop of order 3 shows up as a "triangle" and a loop of order

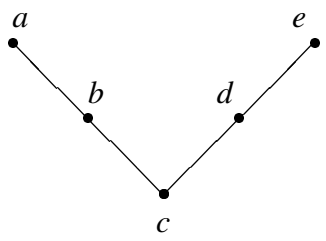


Figure 3:

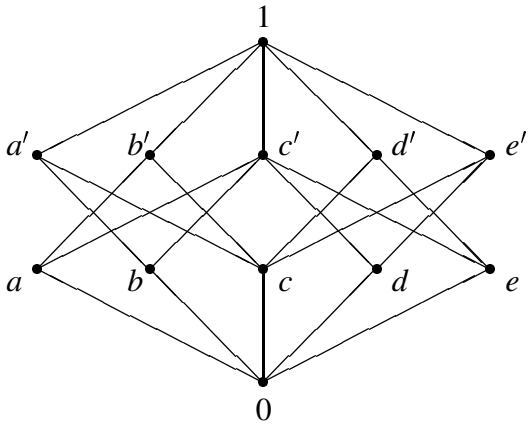


Figure 4:

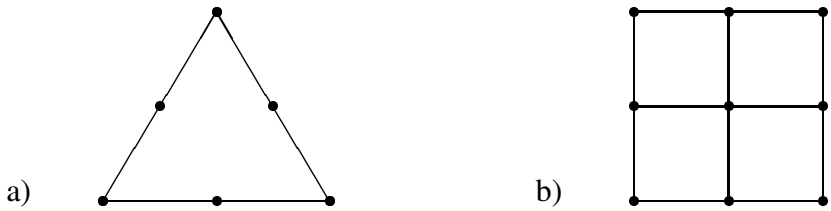


Figure 5:

4 as a "square". For instance, the Greechie diagram drawn in Fig. 5a) does not define an OMP, the Greechie diagram drawn in Fig. 5b) defines an OMP, which is not a lattice.

Definition 2.4.6 *Let L be a Greechie logic. Let X be the set of all atoms of L and let $\mathfrak{B}$ the system of all blocks of L. A subset $W \subseteq X$ is called a weight (on the Greechie diagram) if $|W \cap B| = 1$ for any block $B \in \mathfrak{B}$ ($|A|$ denotes the cardinal number of the set A).*

$W(X)$ will denote the set of all weights on L.

Lemma 2.4.7 *Let L be a Greechie logic and let X be the set of its atoms. Let $\varphi : P(L) \to W(X)$ be the mapping defined by the formula $\varphi(P) = \{x \in X \mid x \notin P\}$. Then, φ is an isomorphism of sets.*

Proof. $\varphi(P)$ is a weight on X for any $P \in P(L)$. The mapping $\varphi : P(L) \to W(X)$ is injective. To show that φ is also surjective, take a weight $W \in W(X)$. Put $P = \{a \in L \mid \text{there exists a } x \in W \text{ such that } a \leq x'\}$. A routine check yields that P is a prime ideal of L. Since $\varphi(P) = W$, the proof is completed.

We may use the one-to-one correspondence between prime ideals and weights to construct OMPs with special properties.

Consider the Greechie diagram of Fig. 6. According to Theorem 2.4.4 the associated Greechie logic is an OMP, termed $W_{3,4}$ by Greechie. We propose that it

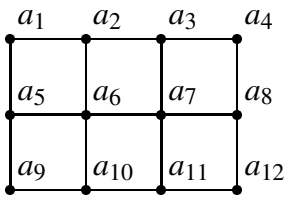


Figure 6: $W_{3,4}$

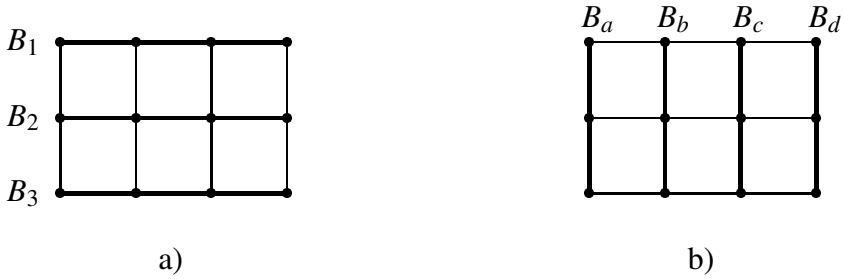


Figure 7: Two disjoint coverings of $W_{3,4}$

possesses no prime ideals. Consider Fig. 7. In these figures the bold lines indicate a disjoint covering of $W_{3,4}$ by its blocks. In Fig. 7.a, the covering consists of 3 blocks B_1, B_2 and B_3. Therefore,
$|W| = |W \cap X| = |W \cap (B_1 \cup B_2 \cup B_3)| = |W \cap B_1| + |W \cap B_2| + |W \cap B_3| = 3$ for any $W \in W(X)$. In Fig. 7.b, there is a disjoint covering consisting of 4 blocks B_a, B_b, B_c and B_d, and, therefore $|W| = 4$ for any $W \in W(X)$. This is a contradiction. Hence, there is no weight and no prime ideal on $W_{3,4}$.

The latter fact is also seen by the following simple reasoning. Assume that the OMP $W_{3,4}$ is isomorphic to a partition logic $(M, \mathfrak{R})$. Let $x \in M$. x has to be element in one of the atoms of B_a. Without loss of generality we may assume that $x \in a_1$. x has to be element in one of the atoms of B_b. Since $x \in a_1$, $x \in a_2$ is not possible, because a_1, a_2 are atoms of the same block B_1. Without loss of generality we may assume that $x \in a_6$. x has to be element in one of the atoms of B_c. The only choice left is $x \in a_{11}$. x has to be element in one of the atoms of B_d. But every choice $x \in a_4, x \in a_8$ or $x \in a_{12}$ is in contradiction to $x \in a_1, x \in a_6$ and $x \in a_{11}$, respectively. Therefore, the OMP $W_{3,4}$ is not isomorphic to a partition logic. Furthermore, there exist OMLs such that $P(L) = \emptyset$. An example is the

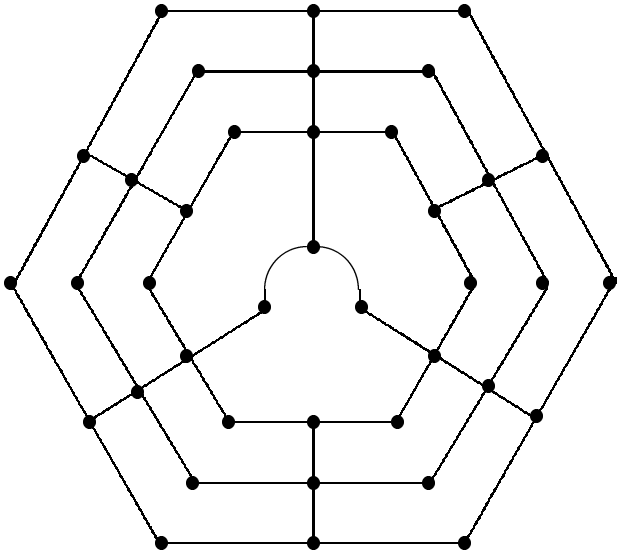


Figure 8: The spider

"spider" lattice of Fig. 8 (cf. [24], p.37).

If a Greechie logic is prime (rich), we may also use the one-to-one correspondence between prime ideals and weights to construct the isomorphic partition logic (the isomorphic concrete logic). We give two examples. Consider the Greechie diagram of the OMP L, drawn in Fig. 9. Let X be the set of its atoms $\{a_1,\ldots,a_9\}$. L possesses 6 weights, $W(X)=\{W_1,\ldots,W_6\}$; see Fig. 10. According to Lemma 2.4.7, instead the mapping $p:L\to\mathfrak{P}(P(L))$ the mapping $q:X\to\mathfrak{P}(W(X))$ is used. q is defined by $q(a)=\{W\in W(X)\mid a\in W\}$ for all $a\in X$. For instance, $q(a_1)=\{W_1,W_2\}$. We obtain the partition logic of Fig. 11. (The numbers denote the corresponding weights.) A check of the axioms in Definition 2.3.1 shows that L is a concrete logic.

The second example describes a partition logic which is not a concrete logic. The example is taken from [24], p. 39. Consider the Greechie diagram of the OMP L, drawn in drawn in Fig. 12,. Let X be the set if its atoms $\{a_1,\ldots,a_{13}\}$. L possesses 14 weights, $W(X)=\{W_1,\ldots,W_{14}\}$, drawn in Fig. 13 (the numbers denote the atoms in a weight). We obtain the partition logic of Fig. 14 (the

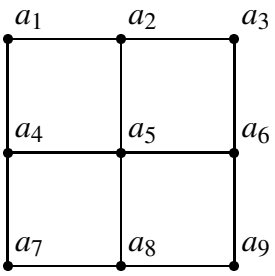


Figure 9: Example 1

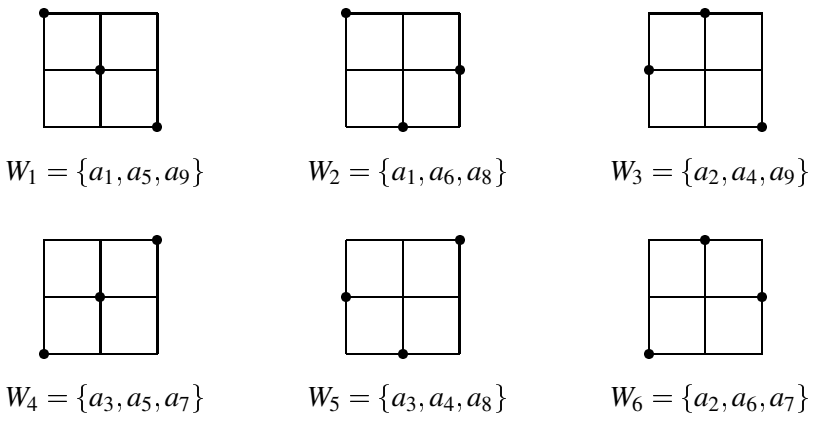


Figure 10: The weights of Example 1

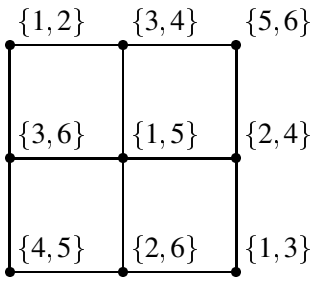


Figure 11: The isomorphic partition logic to example 1

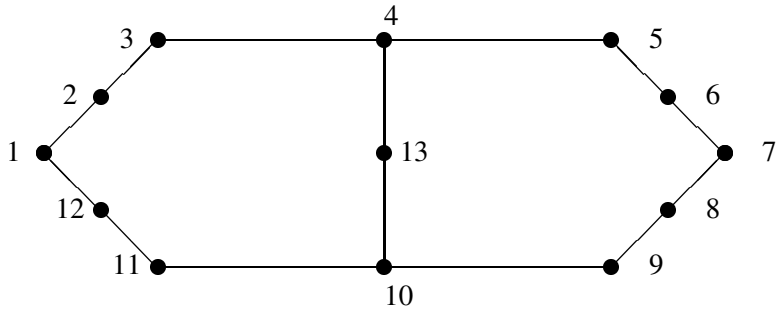
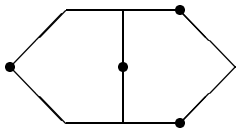
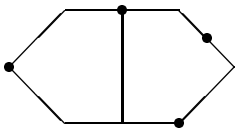


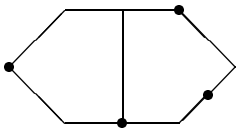
Figure 12: Example 2



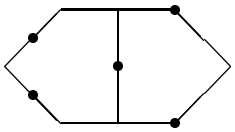
$W_1 = \{1,5,9,13\}$



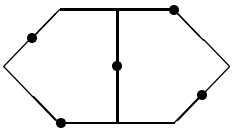
$W_2 = \{1,4,6,9\}$



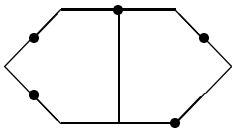
$W_3 = \{1,5,8,10\}$



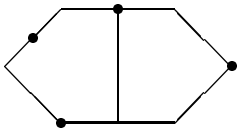
$W_4 = \{2,5,9,12,13\}$



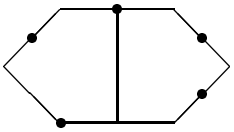
$W_5 = \{2,5,8,11,13\}$



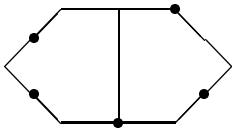
$W_6 = \{2,4,6,9,12\}$



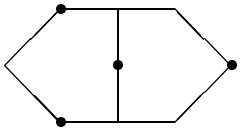
$W_7 = \{2,4,7,11\}$



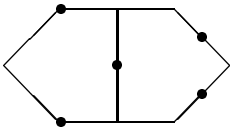
$W_8 = \{2,4,6,8,11\}$



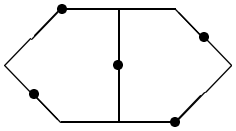
$W_9 = \{2,5,8,10,12\}$



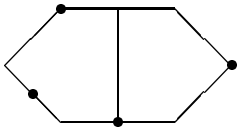
$W_{10} = \{3,7,11,13\}$



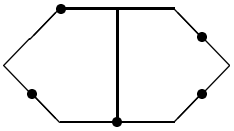
$W_{11} = \{3,6,8,11,13\}$



$W_{12} = \{3,6,9,12,13\}$



$W_{13} = \{3,7,10,12\}$



$W_{14} = \{3,6,8,10,12\}$

Figure 13: The weights of Example 2

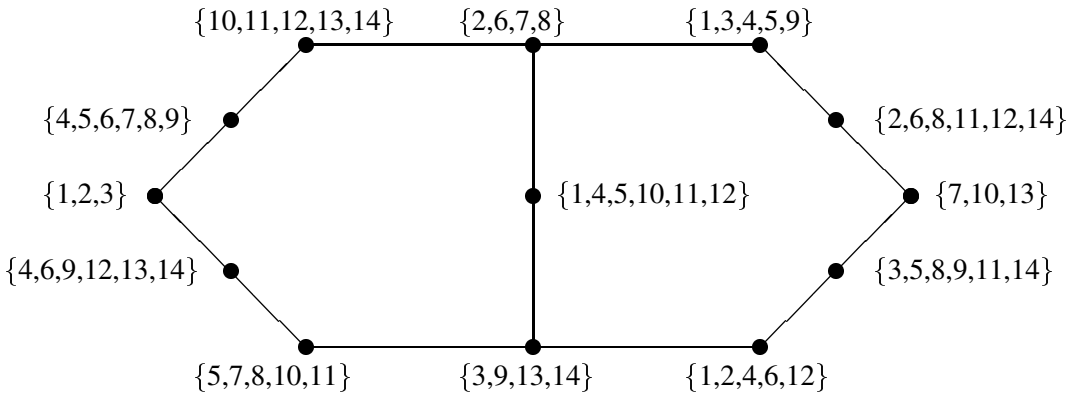


Figure 14: The isomorphic partition logic to example 2

numbers denote the corresponding weights). We propose that L is not a concrete logic. The disjoint sets $\{1,2,3\}$ and $\{7,10,13\}$ are both in L, but not their union $\{1,2,3,7,10,13\}$ (we identify L with its isomorphic partition logic). Therefore, condition (iii) of Definition 2.3.1 is not satisfied, and, L is not a concrete logic (cf. [24], p. 39).

3 Automata Theory

More detailed introductions to automata theory can be found in [3, 4, 6, 14].

3.1 Basic Definitions

An *alphabet* is a finite nonvoid set. The elements of an alphabet are called *symbols*. A *word* (or *string*) is a finite (possibly empty) sequence of symbols. The *length* of a word w, denoted by $|w|$, is the number of symbols composing the string. The *empty word* is denoted by ε. Σ^* denotes the set of all words over an alphabet Σ. The *concatenation* of two words is the word formed by writing the first, followed by the second, with no intervening space. Let Σ be an alphabet. Σ^*

with the concatenation as operation forms a monoid, where the empty word ε is the identity. A (*formal*) *language* over an alphabet Σ is a subset of Σ^*.

Definition 3.1.1 *A Moore automaton M is a five-tuple $M = (Q, \Sigma, \Delta, \delta, \lambda)$, where*
(i) Q is a finite set, called the set of states;
(ii) Σ is an alphabet, called the input alphabet;
(iii) Δ is an alphabet, called the output alphabet;
(iv) δ is a mapping $Q \times \Sigma$ to Q, called the transition function;
(v) λ is a mapping Q to Δ, called the output function.

Let us sketch the appropriate picture informally. At any time the automaton is in a state $q \in Q$, emitting the output $\lambda(q) \in \Delta$. If an input $a \in \Sigma$ is applied to the automaton, in the next discrete time step the automaton instantly assumes the state $p = \delta(q,a)$ and emits the output $\lambda(p)$.

Definition 3.1.2 *A Mealy automaton is a five-tuple $M = (Q, \Sigma, \Delta, \delta, \lambda)$ where $Q, \Sigma, \Delta, \delta$ are as in the Moore automaton and λ is a mapping from $Q \times \Sigma$ to Δ.*

A Mealy automaton emits the output at the instant of the transition from one state to another. The output depends both on the previous state and on the input.

We use directed graphs, called *transition diagrams*, to describe Moore and Mealy automata. The vertices of the graph correspond to the states of the automaton. For a Moore automaton, every vertex is labeled by a pair $(q/x), q \in Q, x \in \Delta$, where q is the corresponding state of the automaton and $x = \lambda(q)$ is the associated output with this state. If there is a transition from state q to state p on input a, then there is an arc labeled a from state q to state p in the transition diagram. For Mealy automata, the vertices are labeled with the corresponding state. If there is a transition from state q to state p on input a, then there is an arc from state q to state p labeled $(a, \lambda(p,a))$. For example, Fig. 15 represents a Moore automaton and Fig. 17 represents a Mealy automaton.

To formally describe the behavior of a automaton, it is desirable to extend the transition function δ to apply to a state and an input *word*, rather than to a state and to a single symbol. We define a mapping $\hat{\delta}$ from $Q \times \Sigma^*$ to Q. We shall denote by $\hat{\delta}(q,w)$ the state in which the automaton is after reading w, starting from state q. Formally, we define

(i) $\hat{\delta}(q, \varepsilon) = q$, and

(ii) $\hat{\delta}(q,wa) = \delta(\hat{\delta}(q,w),a)$ for $w \in \Sigma^*$ and $a \in \Sigma$.

We also extend the output function λ to a mapping $\hat{\lambda} : Q \times \Sigma^* \to \Delta^*$. Let $a_1, \ldots, a_n \in \Sigma$. We define

$\hat{\lambda}(q, a_1 \cdots a_n) = \lambda(q)\lambda(\delta(q,a_1))\lambda(\delta(q,a_1a_2)) \cdots \lambda(\delta(q,a_1 \cdots a_n))$

for Moore automata and

$\hat{\lambda}(q, a_1 \cdots a_n) = \lambda(q,a_1)\lambda(\delta(q,a_1),a_2) \cdots \lambda(\delta(q,a_1 \cdots a_{n-1}),a_n)$

for Mealy automata. $\hat{\lambda}(q,w)$ is the output sequence obtained by applying an input sequence $a_1 \cdots a_n$. Since $\hat{\delta}(q,a) = \delta(q,a)$ and $\hat{\lambda}(q,a) = \lambda(q,a)$ for any input symbol a (i.e.,. $\hat{\lambda}(q) = \lambda(q)$), we may again use δ (i.e.,. λ) in place of $\hat{\delta}$ (i.e.,. $\hat{\lambda}$). Note that for a word w with $|w| = n$, the length of the output sequence is $n+1$ for a Moore automaton and n for a Mealy automaton.

Let p, q be any two states belonging to the state set Q. Then, p is *equivalent* to (*indistinguishable* from) q, written as $p \equiv q$ iff $\lambda(p,w) = \lambda(q,w)$ for all possible words $w \in \Sigma^*$. Otherwise the states are said to be *distinguishable*. We call an automaton *minimal* if any two states of the automaton are distinguishable. We say that a word $w \in \Sigma^*$ distinguishes the two states p and q if $\lambda(q,w) \neq \lambda(p,w)$. A somewhat weaker equivalence property is that of *k-equivalence*. For each positive integer k we say that p is k-equivalent to state q, written as $p \overset{k}{\equiv} q$, iff $\lambda(p,w) = \lambda(q,w)$ for all input sequences $w \in \Sigma^*$ of length k. Both, equivalence $\equiv$ and k-equivalence $\overset{k}{\equiv}$ are equivalence relations that obey the reflexive, symmetric and transitive laws. We denote the partition corresponding to $\equiv$ by $Q/\equiv$ and the partition corresponding to $\overset{k}{\equiv}$ by $Q/\overset{k}{\equiv}$.

Theorem 3.1.3 *(i) (Moore) Let $M = (Q, \Delta, \Sigma, \delta, \lambda)$ be a Moore automaton with n states and m outputs. Further, let λ be onto. Then, two distinguishable states can be distinguished by some word of length at most $n - m$.*

(ii) (Huffman/Mealy) Two distinguishable states of a Mealy automaton with n states can be distinguished by some word of length at most $n - 1$.

Proof. (i) We denote by $f(k)$ the number of equivalence classes of $\overset{k}{\equiv}$ and by $f(\infty)$ the number of equivalence classes of $\equiv$. Then, plainly

$m = f(0) \leq f(1) \leq f(2) \leq \ldots \leq f(\infty) \leq n$

and so we can define N as the least k with $f(k) = f(k+1)$. We propose $f(N) = f(N+1) = f(N+2) = \ldots = f(\infty)$. $p \overset{N+1}{\equiv} q$ implies $\delta(p,a) \overset{N}{\equiv} \delta(q,a)$ for all $a \in \Sigma$ and therefore also $\delta(p,a) \overset{N+1}{\equiv} \delta(q,a)$ (using $f(N) = f(N+1)$). Together, with

$\lambda(p) = \lambda(q)$ we obtain $p \overset{N+2}{\equiv} q$, proving the equality chain above.
$m = f(0) < f(1) < \ldots < f(N) = f(\infty) \leq n$ implies $m+N \leq n$ and any two distinguishable states are distinguishable by a word of length at most $N \leq n-m$.

(ii) The proof is analogous to (i).

Let $M_1 = (Q_1, \Sigma, \Delta, \delta_1, \lambda_1)$, $M_2 = (Q_2, \Sigma, \Delta, \delta_2, \lambda_2)$ be two automata of the same type (both are either Moore or Mealy automata). A state $q_1 \in Q_1$ is said to be *equivalent* to a state $q_2 \in Q_2$ iff $\lambda_1(q_1, w) = \lambda_2(q_2, w)$ for all $w \in \Sigma^*$. The two automata M_1 and M_2 are said to be *equivalent*, if for each state $q_1 \in Q_1$ there exists an equivalent state $q_2 \in Q_2$, and, conversely, for each state $q_2 \in Q_2$ there exists an equivalent state $q_1 \in Q_1$.

Theorem 3.1.4 *Let $M = (Q, \Sigma, \Delta, \delta, \lambda)$ be a Moore or Mealy automaton. Then, there exists a minimal automaton equivalent to M.*

Proof. Put $M^m = (Q/\equiv, \Sigma, \Delta, \delta^m, \lambda^m)$. Define $\delta^m([q], a) = [\delta(q, a)]$ for all $[q] \in Q/\equiv$ and all $a \in \Sigma$. If M is a Moore automaton, define $\lambda^m([q]) = \lambda(q)$. If M is a Mealy automaton define $\lambda^m([q], a) = \lambda(q, a)$. According to the construction, M^m is minimal. Every state $q \in Q$ is equivalent to the state $[q] \in Q/\equiv$. Therefore, also M and M^m are equivalent.

Now, let M_1 be a Moore automaton and M_2 be a Mealy automaton. There can never be equivalence in the above sense between these automata because the output of a Moore automaton to the input $w \in \Sigma^*$ contains one more symbol than the output of the Mealy automaton. However, we may neglect the first output symbol of a Moore automaton $M = (Q, \Sigma, \Delta, \delta, \lambda)$ by using a reduced output function $\lambda' : Q \times \Sigma^* \to \Delta^*$ defined by
$\lambda'(q, a_1 \cdots a_n) = \lambda(\delta(q, a_1)) \cdots \lambda(\delta(q, a_1 \cdots a_n))$.
Note that $\lambda(q, w) = \lambda(q)\lambda'(q, w)$. We may prove the following equivalence theorems, equating the Mealy and Moore models.

Theorem 3.1.5 *If $M_1 = (Q, \Sigma, \Delta, \delta, \lambda_1)$ is a Moore automaton, then there exists a Mealy automaton M_2 equivalent to M_1.*

Proof. Put $M_2 = (Q, \Sigma, \Delta, \delta, \lambda_2)$, where $\lambda_2(q, a) = \lambda(\delta(q, a))$ for any $q \in Q$ and any $a \in \Sigma$. The two automata are equivalent.

Theorem 3.1.6 *Let $M_1 = (Q, \Sigma, \Delta, \delta_1, \lambda_1)$ be a Mealy automaton. Then, there exists a Moore automaton M_2 equivalent to M_1.*

Proof. Put $M_2 = (Q \times \Delta, \Sigma, \Delta, \delta_2, \lambda_2)$. Define $\delta_2((q,x),a) = (\delta_1(q,a), \lambda(q,a))$ and $\lambda_2((q,x)) = x$ for any $(q,x) \in Q \times \Delta$ and $a \in \Sigma$. Then, the states $q \in Q$ of M_1 and $(q,x) \in Q \times \Delta$, x arbitrary, of M_2 are equivalent. Therefore, also M_1 and M_2 are equivalent.

3.2 Automata experiments

In what follows we assume that we are dealing with a Moore or Mealy automaton, which is contained in a black box with input-output interface. Thus, we are only allowed to observe the input and output sequences associated with the box. To conduct an experiment, the experimenter applies an input sequence and notes the resulting output sequence. Using this output sequence, the experimenter tries to interpret the information contained in the sequence to determine the values of the unknown parameters. If there is enough information in the output sequence, the experimenter will state conclusions about the unknown parameters. If, however, the results are inconclusive, the experimenter can decide to extend the experiment by applying another input sequence to obtain more information. Alternatively the experimenter may terminate the experiment with the conclusion that the desired parameter cannot be measured.

Two general types of problems have to be distinguished. The first one deals with a situation in which very little about the device is known except that it is a Moore or Mealy automaton with a given input set and that it is one particular automaton from a general class of automaton. In this case, we are dealing with an *automaton identification problem*. To solve this problem we must determine the model that can be used to describe the automaton's input-output behavior.

The second general class includes measurement and control problems. In this case, we conduct experiments on an automaton with a known transition table (i.e the five-tuple $(Q, \Sigma, \Delta, \delta, \lambda)$). Here, we are interested in measuring and/or controlling various parameters of the automaton.

The types of experiments that we can perform are limited by the number of identical copies of the automaton we have available for investigation, the amount of flexibility that we allow the experimenter, and the amount of a priori information available about the automaton's internal behavior. Usually, when we are carrying out an experiment, we assume that only a *single* copy of the automaton is

available. Such an experiment is called a *simple experiment*. On occasion, however, we have several identical copies of the automaton or a single automaton with a "reset" button. Experiments that take advantage of the availability of effectively more than one copy of an automaton are called *multiple experiments*.

The amount of flexibility that we allow the experimenter in selecting the input sequences is an important consideration. If the input sequence is fixed in advance, we say that the experimenter is required to perform a *preset experiment*. If the experimenter can modify the input sequence in response to information gained from the output sequences, we call this an *adaptive (branch) experiment* in which the input consists of a succession of subsequences, each corresponding to a decision on the experimenter's part.

We shall describe two important measurement problems. In the first, the *terminal state identification (homing) problem*, we are dealing with an automaton with an unknown initial state q. The goal is to identify the final state of the automaton. We apply an appropriate input sequence $w \in \Sigma^*$ and observe the resulting output $\lambda(q,w)$. On the basis of this observation we are able to specify the terminal state $p = \delta(q,w)$. The terminal-state identification problem is always solvable.

The *initial-state identification (diagnosing) problem* deals with the problem of trying to determine the unknown initial state of the automaton. To solve this problem we apply an appropriate input word to the automaton or we carry out an adaptive experiment. From the observation of the corresponding output, we are able to make propositions of the initial state. Not all initial-state identification problems have unique solutions. More exactly, there exist automata such that the initial state of the automaton is not determinable. The first automaton of this kind was invented to demonstrate that particular feature by Moore [20]. It is quite remarkable that Moore's original motivation for the introduction of Moore automata was the modeling of the Heisenberg uncertainty principle.

Consider the Moore automaton of Fig. 15. All four states are mutually distinguishable: The first free output symbol distinguishes q_4, which has output 1, from all other states, which have output 0.
To distinguish between q_1 and q_2 we apply the input 0 ($\lambda(q_1,0) = 01$, $\lambda(q_2,0) = 00$).
To distinguish between q_1 and q_3 we apply the input 1 ($\lambda(q_1,1) = 00$, $\lambda(q_3,1) = 01$).
To distinguish between q_2 and q_3 we apply the input 0 ($\lambda(q_2,0) = 00$, $\lambda(q_2,0) = 01$).

Nevertheless, the initial state is not determinable. Any experiment which dis-

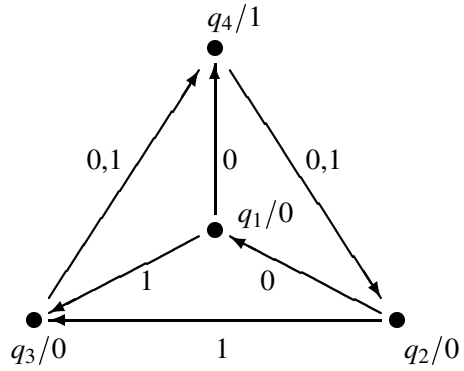


Figure 15: Moore's uncertainty automaton

tinguishes between q_1 and q_2 cannot distinguish between q_1 and q_3. Conversely, any experiment which distinguishes between q_1 and q_3 cannot distinguish between q_1 and q_3. Note that any experiment which begins with the input 1 does not permit q_1 to be distinguished from q_2 (since in either case the first input is 0 and the second state is q_3, so that no future inputs can produce different outputs). Similarly, any experiment which begins with the input 0 does not permit q_1 to be distinguished from q_3. Moore [20] speaks of an "analogue of the Heisenberg uncertainty principle," which was termed "Moore's uncertainty Principle" by Conway [6]. D. Finkelstein and S. R. Finkelstein have called this feature "computational complementarity."

Note that, as has already been pointed out by Moore, if an arbitrary number of identical automaton copies in the same initial state were available, the initial-state problem would be solvable by multiple experiments for any minimal automaton. In this setup, for every pair $\{p,q\}$ of states, one could take a "fresh" automaton copy and apply an input word which distinguishes the two states p and q. From the observed outputs one could then determine the initial state.

A preset experiment is completely specified by an input word $w \in \Sigma^*$. Formally, an adaptive experiment can be defined by a mapping $E : \Delta^* \to \Sigma \cup \{\varepsilon\}$. The experiment E is carried out in the following way:

(i) If the automaton is a Mealy automaton, $E(\varepsilon)$ denotes the first input symbol. For a Moore automaton, $E(x)$ denotes the first input symbol, where x is the first

observed output symbol, which comes free.

(ii) Let us assume the input $w \in \Sigma^*$ was applied and the output $W \in \Delta^*$ was observed. Then, we apply the input $E(W)$ to the automaton. The experiment terminates if $E(W) = \varepsilon$.

The class of preset experiments is a subclass of the class of adaptive experiments. For every experiment E we denote the obtained output of an initial state q by $\lambda_E(q)$. λ_E defines a mapping Q to Δ^*.

3.3 Propositional Calculus of Automata

In the following, we shall investigate the logic of the initial-state identification problem. We call a proposition regarding the initial state of the automaton *experimentally decidable* if there is an experiment which determines the truth value of the proposition. The most general form of a prediction concerning the initial state q of the automaton is that the initial state q is contained in a subset P of the state set Q. Therefore, we may identify propositions concerning the initial state with subsets of Q. A subset P of Q is then identified with the proposition that the initial state is contained in P. More explicitly, we are dealing with propositions of the form, *"the initial state of the automaton is in P"*, where P is a subset of the set of automaton states Q.

We are now dealing with the problem of which subsets of the state set are experimentally decidable. Note, for instance, that the proposition $\{q_1\}$ (i.e. the proposition "the initial state of the automaton is q_1") regarding Moore's uncertainty automaton (cf. Fig. 15) is not decidable.

Definition 3.3.1 (Automaton Propositional Calculus) *Let E be an experiment (a preset or adaptive one). We define an equivalence relation on the state set Q by $q \overset{E}{\equiv} p$ iff $\lambda_E(q) = \lambda_E(p)$*

for any $q, p \in Q$. We denote the partition of Q corresponding to $\overset{E}{\equiv}$ by $Q/\overset{E}{\equiv}$. The propositions decidable by the experiment E are the elements of the Boolean algebra generated by $Q/\overset{E}{\equiv}$, denoted by B_E. There is also another way to construct the experimentally decidable propositions of an experiment E. Let $\lambda_E(P) = \bigcup_{q \in P} \lambda_E(q)$ be the direct image of P under λ_E for any $P \subseteq Q$. We denote the direct image of Q by O_E, $O_E = \lambda_E(Q)$. It follows that the most general form of a prediction concerning the outcome W of the experiment E is that W lays in a subset of O_E. Therefore,

the experimentally decidable propositions consist of all inverse images $\lambda_E^{-1}(S)$ of subsets S of O_E, a procedure which can be constructively formulated (e.g.; as an effectively computable algorithm), and which also leads to the Boolean algebra B_E. Let $\mathfrak{B}$ be the set of all Boolean algebras B_E. We call the partition logic $R = (Q, \mathfrak{B})$ an automaton propositional calculus.

This calculus possesses the following properties:

(i) R contains two special propositions: the proposition $\emptyset$, that the automaton is in no initial state, which is always false, and, the proposition Q , that the automaton is in an arbitrary state, which is always true. $\mathbf{0} \equiv \emptyset$ is the least element and $\mathbf{1} \equiv Q$ is the greatest element in R.

(ii) Let $A \in R$. Any experiment which decides A decides also $A' = Q \backslash A$. Moreover, A is true iff A' is false.

(iii) Let $A, B \in R$. $A \leq B$ holds iff

(a) there is an experiment which decides both propositions A and B.

(b) A implies B (whenever A is true, then also B is true), which is also expressed by $A \subseteq B$.

The use of a nontransitive implication relation is not new (cf. [26, 18, 19]).

We shall give some examples. First, we shall construct the propositional calculus of Moore's original uncertainty automaton (cf. Fig. 15). There are 3 different partitions accessible by experiments. The preset experiment ε corresponds to observing only the first free output of the Moore automaton without any input. Therefore it yields the partition $Q/(\varepsilon) = \{\{q_1, q_2, q_3\}, \{q_4\}\}$.
The preset experiment 0, i.e., input of 0, yields the partition
$Q/(0) = \{\{q_1, q_3\}, \{q_2\}, \{q_4\}\}$.
The preset experiment 1, i.e., the input of 1, yields the partition
$Q/(1) = \{\{q_1, q_2\}, \{q_3\}, \{q_4\}\}$.
$Q/(0)$ and $Q/(1)$ are finer partitions than $Q/(\varepsilon)$ and we may neglect $Q/(\varepsilon)$ by forming the propositional calculus. We obtain the partition logic drawn in Fig. 16 (the numbers denote the corresponding states). A Hilbert space representation of the partition logic is drawn in Fig. 21.

The automaton defined by Fig. 17 yields a propositional calculus drawn in Fig. 18, which is also found in the quantum logic of two-dimensional Hilbert space.

Every automaton proposition calculus is by definition a partition logic. Conversely, to every partition logic, a Mealy automaton can be effectively constructed which possesses that partition logic as propositional calculus (cf. [27]). Let $R =$

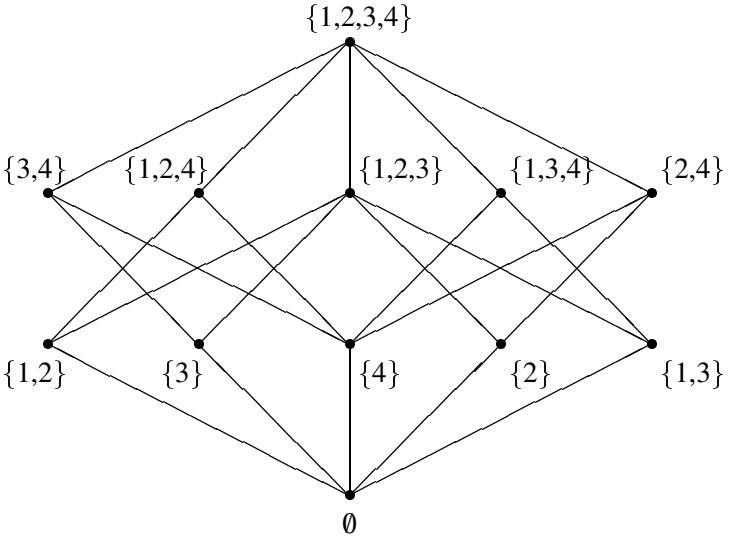


Figure 16: Hasse diagram of Moore's uncertainty automaton

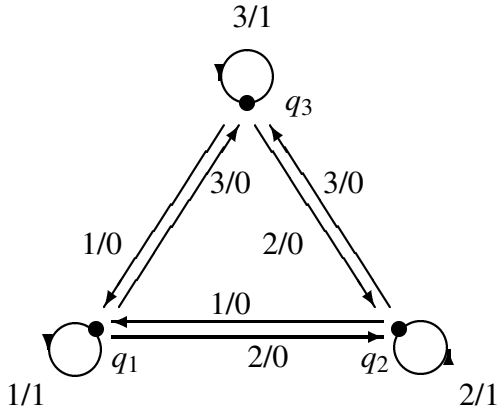


Figure 17: Quantumlike Mealy automaton

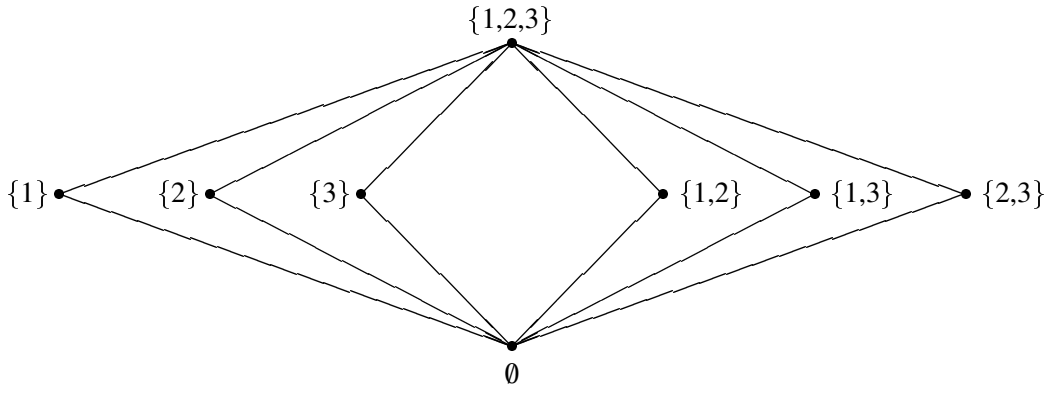


Figure 18: Hasse diagram of the automaton logic of the quantumlike Mealy automaton

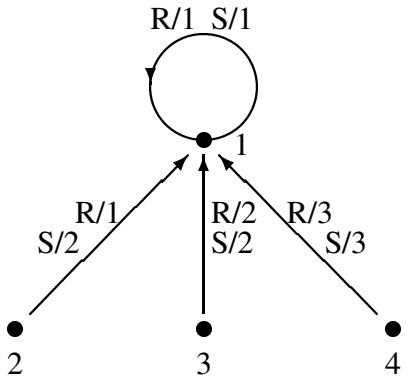


Figure 19: Mealy automaton yielding the partition logic of Moore's uncertainty automaton

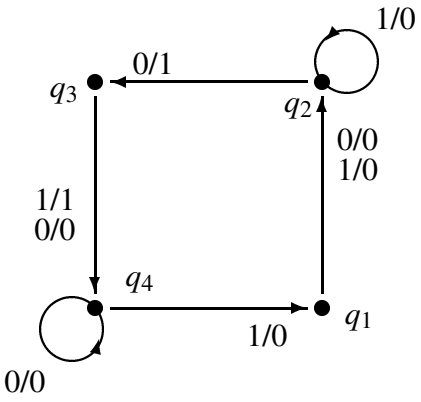


Figure 20: Mealy automaton yielding a nontransitive propositional calculus

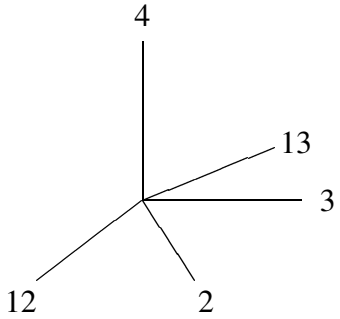


Figure 21: Identification of atoms with rays in three-dimensional real Hilbert space. If $v(a)$ is the subspace spanned by a, $v(12) \perp v(3)$, $v(2) \perp v(13)$, $v(12) \perp v(4)$, $v(2) \perp v(4)$, $v(3) \perp v(4)$, $v(13) \perp v(4)$, $v(12) \neq v(2)$

$(Q,\mathfrak{R})$ be a partition logic. We rewrite every $P \in \mathfrak{R}$ as an indexed family $P = (P_i)_{i\in I_n}$, where the index set I_n denotes the set $\{1,\ldots,n\}$ of natural numbers. We assume that $P_i \neq P_j$ for $i \neq j$. N denotes the greatest number of elements in a partition $P \in \mathfrak{R}$. We put $M = (Q,\mathfrak{R},I_N,\delta,\lambda)$. Next, the transition and output functions δ and λ have to be properly defined. Let p be an arbitrary element of Q. For all $q \in Q$ and all $P \in \mathfrak{R}$ we define
(i) $\delta(q,P) = p$ and
(ii) $\lambda(q,P) = i$ iff $q \in P_i$.
In doing so, we obtain as the automaton propositional calculus the partition logic $(M,\mathfrak{R})$. Instead of $\mathfrak{R}$, we could also use the decomposition $\mathfrak{C}(R)$, yielding an automaton with at most three outputs.

We illustrate this construction by an example: Consider the partition logic of Moore's original automaton. It is given by $(Q,\mathfrak{P}) = (\{1,2,3,4\},\{R = \{R_1 = \{1\},R_2 = \{2,3\},R_3 = \{4\}\},S = \{S_1 = \{1,2\},S_2 = \{3\},S_3 = \{4\}\}\}$. We obtain the Mealy automaton $M = \{Q,\{R,S\},\{1,2,3\},\delta,\lambda\}$ where δ and λ are represented by the transition diagram Fig. 19.

We have already remarked that not every partition logic is an orthomodular poset. An automaton example for this case is given in Fig. 20. The finest partition accessible by experiments are $Q/(00) = \{\{1\},\{2\},\{3,4\}\}$ and $Q/(10) = \{\{1,2\},\{3\},\{4\}\}$ (the numbers denote the corresponding states). Here, $\{1\} \leq \{1,2\}$ and $\{1,2\} \leq \{1,2,3\}$ holds, but $\{1\} \leq \{1,2,3\}$ does not hold.

References

[1] Birkhoff, G. and von Neumann, J. (1936). The logic of quantum mechanics, *Annals of Mathematics* **37**, 823-843.

[2] Birkhoff, G. (1948). *Lattice Theory, Second Edition*, American Mathematical Society, New York.

[3] Booth, T. L. (1967). *Sequential automata and Automata Theory*, Wiley and Sons, New York, London, Sydney.

[4] Brauer, W. (1984). *Automatentheorie*, Teubner, Stuttgart.

[5] Chaitin, G. J. (1965). *IEEE Transactions on Electronic Computers*, **EC-14**, 466.

[6] Conway, J. H. (1971). *Regular Algebras and Finite automata*, Clowes and Sons, London.

[7] Crutchfield, J. P. (1993). Observing Complexity and the Complexity of Observation, *Inside versus Outside*, ed. by H. Atmanspacher and G. J. Dalenoort, Springer, Berlin, pp. 235-272.

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

[9] Giuntini, R. (1991). *Quantum Logic and Hidden Variables*, BI Wissenschaftsverlag, Mannheim.

[10] Grätzer, G. (1971). *Lattice Theory*, Freeman, San Francisco.

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

[12] Grib, A. A. and Zapatrin, R. R. (1992). Macroscopic realizations of quantum logics, *International Journal of Theoretical Physics*, **31**, 1669-1687.

[13] Grib, A. A., Svozil, K. and Zapatrin, R. R. (1995). Empirical logic of finite automata: microstatements versus macrostatements, *preprint*.

[14] Hopcroft, J. E. and Ullman, J. D. (1979). *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading-Mass.

[15] Jammer, M. (1974). *The Philosophy of Quantum Mechanics*, Wiley and Sons, New York, Sydney, Toronto.

[16] Jauch, J. (1968). *Foundations of Quantum Mechanics*, Addison-Wesley, Reading, Mass.

[17] Kalmbach, G. (1983). *Orthomodular Lattices*, Academic Press, London, New York.

[18] Kochen S., Specker E. (1965). Logical structures arising in quantum theory, *Symposium on the Theory of Models, Proceedings of the 1963 International Symposium at Berkely*, North-Holland, Amsterdam, pp. 177-189.

[19] Kochen S., Specker E. (1965). The calculus of partial propositional functions, *Proceedings of the 1964 International Congress for Logic, Methodology and Philosophy of Science, Jerusalem*, North-Holland, Amsterdam,pp. 45-57.

[20] Moore, E.F. (1956). *Gedankenexperiments on sequential automata*, in *Automata Studies*, ed. by C.E.Shannon and J.McCarthy, Princeton Univ. Press, Princeton, pp. 129-153.

[21] Navara, M. and Rogalewicz, V. (1991). The pasting construction for orthomodular posets, *Mathematische Nachrichten*, **154**, 157-168.

[22] Piron, C. (1976). *Foundations of Quantum Physics*, Benjamin, Reading, Mass.

[23] Piziak, R. (1991). Orthomodular Lattices and Quadratic Spaces: A Survey, *Rocky Mountain Journal of Mathematics*, **21**, 951-992.

[24] Pták, P. and Pulmannová, S. (1991). *Orthomodualar Structures as Quantum Logics*, Kluwer, Dordrecht.

[25] Schaller, M. and Svozil, K. (1994). Partition Logics of Automata, *Il Nuovo Cimento*, **109 B**, 167-176.

[26] Specker E. (1960). Die Logik nicht gleichzeitig entscheidbarer Aussagen, *Dialectica*, **14**, 239-246.

[27] Svozil, K. (1993). *Randomness and Undecidability in Physics*, World Scientific, Singapore.

[28] Szász, G. (1963). *Introduction to Lattice Theory*, Academic Press, New York.