

Partition logics of automata

M. Schaller and K. Svozil

Institut für Theoretische Physik

Technische Universität Wien

Wiedner Hauptstraße 8-10/136

A-1040 Vienna, Austria

e1360dab@AWIUNI11.EDVZ.UniVie.AC.AT

May 15, 2012

Abstract

We introduce a new type of orthomodular poset which is obtained by considering the pasting of partitions of a set. These partition logics appear in the experimental investigation of finite automata and can be related to certain quantum mechanical systems.

1 Introduction

The input-output analysis of finite automata yields a fresh insight into the quantum mechanical feature of complementarity on a very elementary level. To substantiate this claim it is necessary to interrelate the lattice theoretic [1] approach for a representation of quantum physics, pioneered by G. Birkhoff and J. von Neumann [2] with the theory of finite automata, in particular of Moore and Mealy automata [8, 3, 4, 5].

Informally stated, the motivation behind this investigation is the construction of primitive *experimental statements* or *propositions* about automata [11]. Such experimental statements can be grouped into partitions of the set of internal automaton states; they form the basis of the formal investigation into the corresponding logics. In particular, there exist automata for which

validation of one experimental statement makes impossible the validation of another experimental statement and *vice versa*.

2 Orthomodular Posets

For a detailed introduction to orthomodular posets and lattices see [10] and [7].

A bounded poset P has a smallest element 0 and a largest element 1 , such that $0 \leq x \leq 1$ holds for all $x \in P$.

Let (P, \leq) be a bounded poset. An orthocomplementation on P is a unary operation $'$ on P satisfying

- (i) if $x \leq y$ then $y' \leq x'$,
- (ii) $x'' = x$,
- (iii) the supremum $x \vee x'$ exists and $x \vee x' = 1$.

An orthoposet is a bounded poset with an orthocomplementation.

The relation orthogonal \perp for elements x, y of an orthoposet P is defined by $x \perp y$ (x is orthogonal to y) if $x \leq y'$ holds.

An orthomodular poset P (abbr. OMP) is an orthoposet P satisfying

- (i) if $x \perp y$ then the supremum $x \vee y$ exists in P
- (ii) $x \leq y$ implies $y = x \vee (x' \wedge y)$.

An orthomodular lattice (abbr. OML) is an OMP which is also a lattice.

A subalgebra is a subset M which is closed under the operations $', \vee, \wedge$ and which contains 0 and 1 . The subalgebra ΓA generated by an arbitrary subset A of P is the smallest subalgebra of P containing A if it exists.

If $x \leq y$ then it is obvious that $\Gamma\{x, y\} = \{0, 1, x, x' \wedge y, y', y, x \wedge y', x'\}$ (see [7]).

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

Lemma 1 (i) if $a, b \in L$ and $a \perp b$ then $a \leftrightarrow b$;
(ii) if $a, b \in L$ and $a \leq b$, then $a \leftrightarrow b$;
(iii) if $a, b \in L, a \leftrightarrow b, a = a_1 \vee c, b = b_1 \vee c$ for three mutually orthogonal elements $a_1, b_1, c \in L$ then $a \vee b$ and $a \wedge b$ exist and $a \vee b = a_1 \vee b_1 \vee c$ and $a \wedge b = c$.

For a proof see [10].

Let \mathfrak{L} be a non-empty set of OMPs such that all $P, Q \in \mathfrak{L}$ satisfy the following condition:

$P \cap Q$ is a sub-OMP of both P and Q , and the partial orderings and the orthocomplementations of P and Q coincide on $P \cap Q$.

On $L = \bigcup \mathfrak{L}$ define the relation \leq and $'$ as follows:

$x \leq y$ iff $x \leq_P y$ for some $P \in \mathfrak{L}$

$x' = x'^P$ iff $x \in P$

The algebra $(L, \leq, ')$ is called the pasting of the set \mathfrak{L} .

A block of an OMP P is a maximal Boolean subalgebra of P . Every element x of P 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.

Let \mathfrak{B} be a non-empty set of Boolean algebras such that for all $B, C \in \mathfrak{B}$ the following conditions hold:

(i) $0 = 0_B = 0_C, 1 = 1_B = 1_C$,

(ii) for all $x \in B \cap C$ holds: $x' = x'^B = x'^C$,

(iii) $B \cap C$ is a subalgebra of both B and C .

(the indices indicate that the operations belong to the respective Boolean algebras)

Define on the set $L = \bigcup \mathfrak{B}$ a unary operation $'$ and a relation \leq by:

$x' = x'^B$ if $x \in B$ for a $B \in \mathfrak{B}$,

$x \leq y$ if $x \leq_B y$ for a $B \in \mathfrak{B}$.

The set L together with $'$ and \leq is called the pasting of the set \mathfrak{B} .

Consider a set M and a set \mathfrak{R} of partitions of M . Every partition $P \in \mathfrak{R}$ generates a Boolean algebra $(B_P, \subseteq, ')$ where B_P is the union set of P , \subseteq is

the inclusion and $'$ is the set-complement. We will call the pasting of this Boolean algebras a partition logic and denote it by (M, \mathfrak{A}) .

Lemma 2 *A partition logic is an OMP if the following two conditions hold:*

- (i) *the induced relation \leq is transitive.*
- (ii) *if $a \perp b$ then the supremum $a \vee b$ exists.*

Proof: Let (M, \mathfrak{A}) be a partition logic which satisfies the conditions. Obviously (M, \mathfrak{A}) is an orthocomplemented poset. Let $a, b \in (M, \mathfrak{A})$ and $a \perp b$. Then there exists a $P \in \mathfrak{A}$ such that $a, b \in B_P$. In B_P the supremum exists and $a \vee_{B_P} b = a \cup b$. If $c \in (M, \mathfrak{A})$, $a \leq c$ and $b \leq c$ then $a \cup b \subseteq c$ holds. Since the supremum $a \vee b$ exists in (M, \mathfrak{A}) we conclude that $a \vee b = a \cup b \leq c$. The orthomodular law holds, because for $a \leq b$ and $a, b \in B_P$ we have $a \vee (a' \wedge b) = a \vee_{B_P} (a' \wedge_{B_P} b) = b$.

For a more general proposition, see [9].

Every OMP can be viewed as the pasting of its blocks (see [10],[7] and [9]). Another way to disassemble an OMP L in Boolean algebras is the following one: Define $\mathfrak{C}(L) := \{\Gamma\{x, y\} | x, y \in L; x \leq y\}$. Then L is the pasting of the set $\mathfrak{C}(L)$.

A non-void subset I of an OMP L is called an ideal if it satisfies the following conditions:

- (i) if $a \in I$ and $b \leq a$ then $b \in I$,
- (ii) if $a, b \in I$ and $a \perp b$ then $a \vee b \in I$.

A prime ideal of an OMP L is an ideal P , $P \neq L$ which satisfies the following condition:

$a \perp b$ implies $a \in P$ or $b \in P$

The set of all prime ideals of an OMP L will be denoted by $P(L)$.

Lemma 3 *Let P , $P \neq L$ be an ideal. The following conditions are equivalent.*

- (i) *P is a prime ideal.*
- (ii) *$x \in P$ iff $x' \notin P$ for all $x \in P$.*

Proof: (i) implies (ii). $x \perp x'$ holds, hence $x \in P$ or $x' \in P$. If $x \in P$ and $x' \in P$ then also $1 = x \vee x' \in P$, which is a contradiction to $P \neq L$.
(ii) implies (i). Let $x, y \in P$ be orthogonal elements. One of the elements x, x' is in P . If $x \in P$ then condition (i) is satisfied. On the other hand, if $x' \in P$ then also $y \in P$ because $y \leq x'$ holds.

Lemma 4 *Let P be a prime ideal.
 $a \leftrightarrow b$ and $a \wedge b \in P$ implies $a \in P$ or $b \in P$.*

Proof: Let $a \leftrightarrow b$ and $a \wedge b \in P$. Then there exist three mutually orthogonal elements a_1, b_1, c , such that $a = a_1 \vee c, b = b_1 \vee c$ and $c = a \wedge b$. $a_1 \perp b_1$ implies $a_1 \in P$ or $b_1 \in P$. If $a_1 \in P$ then also $a = a_1 \vee c \in P$. On the other hand, if $b_1 \in P$ then also $b = b_1 \vee c \in P$.

An OMP L is called prime if for all $a, b \in L, a \neq b$ there exists a prime ideal P of L such that $a \in P, b \notin P$ or $a \notin P, b \in P$.

Let L_1, L_2 be OMPs. A mapping $f : L_1 \rightarrow L_2$ is called a morphism if the following conditions are satisfied:

- (i) $f(0) = 0$;
- (ii) $f(a') = f(a)'$ for all $a \in L_1$;
- (iii) $a, b \in L_1$ and $a \perp b$ implies $f(a \vee b) = f(a) \vee f(b)$.

A morphism f is called an isomorphism if f is injective and f^{-1} is also a morphism.

The following theorem can be seen as a generalization of Birkhoff's representation theorem for OMPs. In our case we have to require that the OMP is prime. (For Boolean algebras Stone's theorem hold, which state that every distributive lattice is prime.)

Theorem 1 *An OMP L is isomorphic to a partition-logic iff L is prime.*

Proof: (i) Suppose first that L is isomorphic to an partition-logic (M, \mathfrak{P}) . We may assume that $L = (M, \mathfrak{P})$. Let $A, B \in L$ and $A \neq B$. Then the set $C := (A - B) \cup (B - A)$ is not empty and we can choose a point $p \in C$.

Define a set $P \subseteq L$ by $P := \{X \in L \mid p \notin X\}$. Then only one of the elements A, B is in P and we can easily check that P is a prime ideal.

(ii) Let L be a prime OMP. We denote by $\mathfrak{P}(P(L))$ the power set of $P(L)$. Define a mapping $p : L \rightarrow \mathfrak{P}(P(L))$ by $p(x) := \{P \in P(L) \mid x \notin P\}$.

We first proof some properties of p :

$p(0) = \emptyset$ is obvious.

From Lemma 3 we know that $x \in P$ iff $x' \notin P$. Hence, $p(x') = P(L) - p(x)$. Let $x, y \in L$ and $x \perp y$. We verify that $p(x \vee y) = p(x) \cup p(y)$ holds. Let $P \in p(x \vee y)$. This implies $x \vee y \notin P$. If $x \in P$ and $y \in P$ then also $x \vee y \in P$ which is contradiction. Hence, $x \notin P$ or $y \notin P$ and therefore $P \in p(x) \vee p(y)$. Let now $P \in p(x)$, i.e. $x \notin P$. If $x \vee y \in P$ then also $x \in P$ which is again a contradiction. Hence, $x \vee y \notin P$ and $P \in p(x \vee y)$. This proofs $p(x) \subseteq p(x \vee y)$; in the same way $p(y) \subseteq p(x \vee y)$.

We define now the isomorphic partition logic: Let $x, y \in L$ and $x \perp y$. We set $R(\{x, y\}) := \{p(x), p(y), p(x' \wedge y')\}$. Let us verify that $R(\{x, y\})$ is a partition of $P(L)$. From $x \leq y'$ we obtain $p(x) \subseteq p(y')$ and further $p(x) \cap p(y) = \emptyset$. $p(x' \wedge y') = p((x \vee y)') = P(L) - (p(x) \cup p(y))$ proofs that $R(\{x, y\})$ is indeed a partition. Set $\mathfrak{R} = \{R(\{x, y\}) \mid x \perp y; x, y \in L\}$. Then $K = (P(L), \mathfrak{R})$ is a partition logic. Further, K is the range of p . If x, y are different elements of L then there exists a prime ideal P such that P includes one of the elements without including the other. Therefore, p is injective. Hence, $p : L \rightarrow K$ together with the properties of p above is an isomorphism.

Let L be a Greechie logic (an OMP which can be described by a Greechie diagram, see [10]). Let A be the set of all atoms of L and let \mathfrak{B} the set of all blocks of L .

A set $W \subseteq A$ is called a weight if for all $B \in \mathfrak{B}$ we have $W \cap B = \{a\}$ for an atom $a \in A$, i.e. the intersection of W and an arbitrary block consists of exactly one atom.

The set of all weights on L will be denoted by $W(X)$.

Lemma 5 *Let $\varphi : P(L) \rightarrow W(X)$ be the mapping defined by $\varphi(P) = \{a \in X \mid a \notin P\}$. Then φ is bijective.*

For a proof see [10].

We only remark that the inverse mapping is given by $\varphi^{-1} = \{x \in L \mid x \leq a'\}$

for an atom $a \in W$ }.

There exists a finite OMP, which is not prime, and hence, which is not isomorphic to a partition-logic. The OMP is given by the following Greechie-diagram.

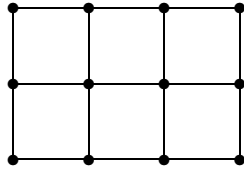


Fig. 1

It is easy to check that the OMP has no prime ideals.

A concrete logic is a set Ω and a collection Δ of subsets of Ω such that the following conditions are satisfied:

- (i) $\emptyset \in \Delta$
- (ii) if $A \in \Delta$ then $\Omega - A \in \Delta$
- (iii) if $A, B \in \Delta$ and $A \cap B = \emptyset$ then $A \cup B \in \Delta$

The relation \leq is given by the inclusion and the ortocomplement $'$ is given by the complement in Ω . By this definition every concrete logic is an OMP.

A two-valued state on an OMP L is a mapping $s : L \rightarrow \{0, 1\}$ such that

- (i) $s(1) = 1$;
- (ii) if $a \perp b$ and $a, b \in L$ then $s(a \vee b) = s(a) + s(b)$.

The set of all two-valued states on L is denoted by $\mathfrak{S}(L)$.

Lemma 6 *Let L be an OMP. Then the mapping $\chi : \mathfrak{S}(L) \rightarrow P(L), \chi(s) = \{x \in L | s(x) = 0\}$ is bijective.*

Proof: Let L be an OMP, s a two-valued state on L . We show that the set $P = \{x \in L | s(x) = 0\}$ is a prime ideal of L . $s(0) = 0$ and $s(1) = 1$ implies $P \neq \emptyset$ and $P \neq L$.

Let $b \in P$ and $a \leq b$. Then $s(b) = 0$ implies $s(a) = 0$. Hence, $a \in P$.

Let $a \perp b$ and $a, b \in P$. Then $a \vee b \in P$ because $s(a \vee b) = s(a) + s(b) = 0$. Let $x \perp y$. $s(x \vee y) = s(x) + s(y) \leq 1$ implies $s(x) = 0$ or $s(y) = 0$. Hence, $x \in P$ or $y \in P$.

The inverse mapping is given by $\chi^{-1}(P)(x) = 0$ if $x \in P$ and $\chi^{-1}(P)(x) = 1$ if $x \notin P$.

An OMP L is called rich if if the following implication holds: $(\{s \in \mathfrak{S}(L) | s(a) = 1\} \subseteq \{s \in \mathfrak{S}(L) | s(b) = 1\})$ implies $a \leq b$.

Theorem 2 (Gudder [6]): *An OMP L is isomorphic to a concrete logic iff L is rich.*

For a proof see [6] or [10].

Lemma 7 *Every rich OMP is prime.*

Proof: Let L be a rich OMP and let $a, b, a \neq b$ be elements of L . Let us assume that $(\{s \in \mathfrak{S}(L) | s(a) = 1\} = \{s \in \mathfrak{S}(L) | s(b) = 1\})$. The richness of L implies $a \leq b$ and $b \leq a$, hence $a = b$, which is a contradiction to $a \neq b$. It follows there is a state $s \in \mathfrak{S}(L)$ such that $s(a) = 1, s(b) = 0$ or $s(a) = 0, s(b) = 1$. Lemma 6 implies that this is equivalent to the existence of a prime ideal which includes one of the elements a, b , without including the other. This proves also that every concrete logic is a partiton logic.

There exists an OMP which is prime, but not rich. The OMP is given by the following Greechie-diagram.

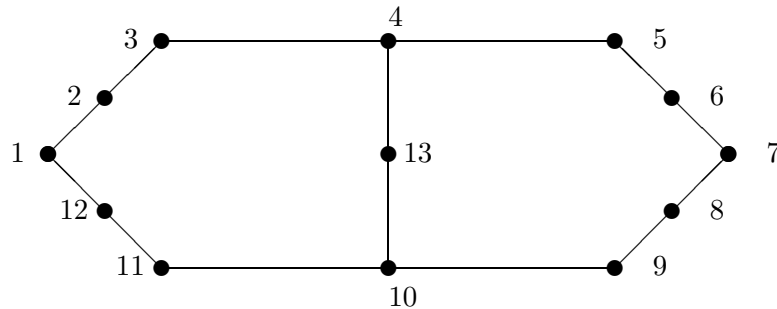


Fig. 2

We construct an isomorphic partition logic: The weights of L are:

$$\begin{aligned} W_1 &= \{1, 5, 9, 13\}, W_2 = \{1, 4, 6, 9\}, W_3 = \{1, 5, 8, 10\}, \\ W_4 &= \{2, 5, 9, 12, 13\}, W_5 = \{2, 5, 8, 11, 13\}, W_6 = \{2, 4, 6, 9, 12\}, \\ W_7 &= \{2, 4, 7, 11\}, W_8 = \{2, 4, 6, 8, 11\}, W_9 = \{2, 5, 8, 10, 12\}, \\ W_{10} &= \{3, 7, 11, 13\}, W_{11} = \{3, 6, 8, 11, 13\}, W_{12} = \{3, 7, 10, 12\} \\ W_{13} &= \{3, 6, 8, 10, 12\}. \end{aligned}$$

For Greechie logics the composition $q := \varphi \circ p : L \rightarrow \mathfrak{P}(W(X))$ is also a morphism and the restriction $q : L \rightarrow q(L)$ is again an isomorphism (lemma 5 and theorem 1). We can also restrict q to the set of atoms: $q : X \rightarrow \mathfrak{P}(W(X))$. The function q is given by $q(x) = \{W \in W(X) | x \in W\}$.

The following Greechie shows the picture of q for the atoms. (The letter W is not written).

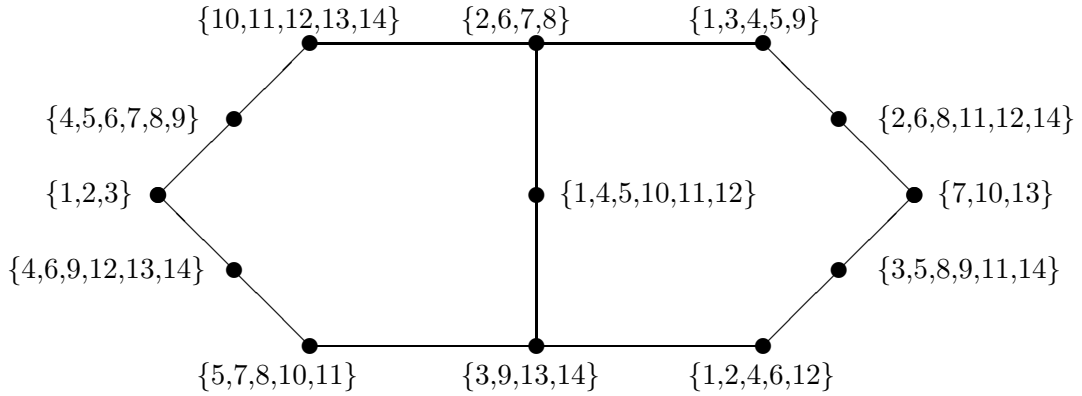


Fig. 3

For a proof that the OMP from figure 2 is not rich, see [10].

3 Automata Logic

In the sequel we want to investigate the logic of discrete, deterministic systems. For this purpose we use the automaton concept.

An alphabet is a finite set, the elements of an alphabet are called symbols. A word (string) is a finite sequence of symbols juxtaposed. Let Σ be an alphabet. The set of all words over Σ
 $\Sigma^* = \{x_1 \dots x_n | x_i \in \Sigma, 1 \leq i \leq n, 0 \leq n\}$
 together with the concatenation is the free monoid with neutral element ϵ (for $n = 0$). $|w|$ denotes the length of a word $w \in \Sigma^*$: $|x_1 \dots x_n| = n$.

A Mealy automaton is a 5-tuple $M = (Q, \Sigma, \Delta, \delta, \lambda)$, where

- (i) Q is the set of states,
- (ii) Σ is the input alphabet,
- (iii) Δ is the output alphabet,
- (iv) $\delta : Q \times \Sigma \rightarrow Q$ is the transition function,
- (v) $\lambda : Q \times \Sigma \rightarrow \Delta$ is the output function.

The functions δ and λ can be generalized to functions $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$ and $\hat{\lambda} : Q \times \Sigma^* \rightarrow \Delta^*$.

Let $q \in Q, w \in \Sigma^*, a \in \Sigma$.

- (i) $\hat{\delta}(q, \epsilon) = q$
- (ii) $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$
- (i) $\hat{\lambda}(q, \epsilon) = \epsilon$
- (ii) $\hat{\lambda}(q, wa) = \hat{\lambda}(q, w)\lambda(\hat{\delta}(q, w), a)$

In the sequel we write also δ and λ for $\hat{\delta}$ and $\hat{\lambda}$

We need also the generalization of λ to a function $\lambda : \mathfrak{P}(Q) \rightarrow \mathfrak{P}(\Sigma^*)$.

Let P be a subset of Q . We set

$$\lambda(P, w) = \bigcup_{q \in P} \lambda(q, w).$$

The relation $\stackrel{w}{\equiv}$ for states $p, q \in Q$ and $w \in \Sigma^*$ is defined by

$$p \stackrel{w}{\equiv} q \text{ if } \lambda(p, w) = \lambda(q, w).$$

$\stackrel{w}{\equiv}$ is an equivalence relation. For the partition generated by $\stackrel{w}{\equiv}$ we write $Q / \stackrel{w}{\equiv}$ and the equivalence classes are denoted by $[q]_w$.

Two states $p, q \in Q$ are called distinguishable if there is a word $w \in \Sigma^*$ such that $\lambda(p, w) \neq \lambda(q, w)$.

We shall investigate the following distinguishing problem: If an automaton M and its description is given (i.e. the 5-tuple $M = (Q, \Sigma, \Delta, \delta, \lambda)$), but

not the state of the automaton, which propositions can we make about the initial state? To learn something about the initial state, we have to enter a word and observe the output. If the input word is determined before we observe the first output, the experiment is called a simple one. Alternatively, the input word can depend on its previous output, permitting the experiment to branch. The later experiments are called branch experiments.

Let us first consider only simple experiments. Every simple experiment is described by an input word $w \in \Sigma^*$. We assign to every state $q \in Q$ the proposition “The automaton is in the state q ” and write for this proposition again q . In the same way, we identify a subset P of Q with the disjunction of the propositions in P , in words “The automaton is in a state $q \in Q$ ” (more exactly q stands for “The initial state of the automaton at the begin of the experiment is q ”). A proposition is called decidable if there is a simple experiment which determines the truth value.

In generally, it is not possible to determine what state the automaton was in at the beginning of the experiment even when any pair of the automaton states is distinguishable (Moore’s Uncertainty Principle [8], [3]). So the question we asked is not trivial.

Every partition $Q / \stackrel{w}{\equiv}$ generates a Boolean algebra, denoted by $B(Q / \stackrel{w}{\equiv})$. The mapping $\varphi : Q / \stackrel{w}{\equiv} \rightarrow \lambda(Q, w)$ defined by $\varphi([q]_w) = \lambda(q, w)$ is bijective. We also generalize φ to a function $\varphi : B(Q / \stackrel{w}{\equiv}) \rightarrow \mathfrak{P}(\Sigma^*)$:

$$\varphi(A) = \bigcup_{[q]_w \subseteq A} \lambda(q, w).$$

Lemma 8 *A proposition $P \subseteq Q$ is decidable by a simple experiment $w \in \Sigma^*$ iff $P \in B(Q / \stackrel{w}{\equiv})$.*

Proof: $\lambda(w)$ denotes the observed output of the experiment. Of course, $\lambda(w) \in \lambda(Q, w)$ holds.

(i) Let $P \in B(Q / \stackrel{w}{\equiv})$. It follows

P is true iff $\lambda(w) \in \varphi(P)$.

Hence, P is decidable.

(ii) Let $P \in \mathfrak{P}(Q)$, but $P \notin B(Q / \stackrel{w}{\equiv})$. Then there exists a $W \in \lambda(Q, w)$ such that $\emptyset \subset \varphi^{-1}(W) \cap P \subset \varphi^{-1}(W)$, where the inclusion is proper. There

are points $p, q \in Q$ such that $q \in \varphi^{-1}(W) \cap P$ and $p \in \varphi^{-1}(W) - P$. If the observed output is W then P is not decidable, because the initial state of the automaton could be p as well as q . But the true value of the proposition P depends on this fact.

The pasting of all partitions $Q/\stackrel{w}{\equiv}, w \in \Sigma^*$ generates a partition logic, denoted by $L(M) = (Q, \{Q/\stackrel{w}{\equiv} \mid w \in \Sigma^*\})$.

Lemma 9 (*Svozil [11]*) *To every partition logic L there exists an automaton M such that $L = L(M)$.*

Proof: Let $L = (Q, \mathfrak{P})$ be a partition logic. We will construct a minimal automaton M in the sense, that M has at most three outputs. Therefore we use the disassembling $\mathfrak{C}(L)$.

We define $M = (Q, \mathfrak{C}(L), \{0, 1, 2\}, \delta, \lambda)$.

Let p be an arbitrary element of Q .

Let $q \in Q, C \in \mathfrak{C}(L)$ and denote the atoms of C by A_0, A_1, A_2 .

We define:

$\delta(q, C) := p$ and

$\lambda(q, C) = i, 0 \leq i \leq 2$ if $q \in A_i$.

Obviously, $L = L(M)$.

Notice, that nonisomorphic automata can possess the same partition logic.

We now want to clarify in which way the logical connection of propositions is expressed in the partition logic. The interpretation of the orthocomplement is quite straightforward. The orthocomplement A' of a proposition A is the logical negation of A , i.e. A' has the meaning "non A ". Notice, that every experiment which decides A also decides A' .

The relation \leq can be interpreted by:

$A \leq B$ iff ("if A is true then B is true" and there is a experiment which decides both propositions A and B).

That means \leq is not an idealistic relation, but a measurable one.

Two propositions of a partition logic are simultaneously measurable, if both propositions can be decided by the same experiment. In quantum logic

two propositions are simultaneously measurable if they generate a distributive subalgebra. This does not hold for partition logics.

Consider two automata with the partitions:

$\{\{\{1\}, \{2\}, \{3\}, \{4\}\}\}$ and
 $\{\{\{1\}, \{2\}, \{3, 4\}\}, \{\{1\}, \{3\}, \{2, 4\}\}, \{\{1\}, \{4\}, \{2, 3\}\},$
 $\{\{2\}, \{3\}, \{1, 4\}\}, \{\{2\}, \{4\}, \{1, 3\}\}, \{\{3\}, \{4\}, \{1, 2\}\}\}.$

The pasting of the both is the same, the Boolean algebra 2^4 , but by the first the two propositions $\{1, 2\}$ and $\{1, 3\}$ are simultaneously measurable, but not by the second partition. For a further discussion of the relationship between partition logics and quantum logics, see K. Svozil [11].

At last we will consider branch experiments. Formally, we can describe branch experiments by a mapping $E : \Delta^* \rightarrow \Sigma \cup \{\epsilon\}$ (see also [3]).

The branch experiment E is carried out in the following way:

(i) $E(\epsilon)$ denotes the initial input symbol.

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

$\lambda_E(q)$ denotes the observed output, if the automaton was in the initial state q . Similar to simple experiments we define an equivalence relation

$q \stackrel{E}{\equiv} p$ iff $\lambda_E(q) = \lambda_E(p)$.

Therefore we can use the same methods as for simple experiments. They yield a partition logic $L_{branch}(M)$ by the same way. Note that $L_{simple}(M) \subseteq L_{branch}(M)$ which is the simple consequence of the fact that every simple experiment is also a branch experiment.

References

- [1] Birkhoff, G.: Lattice Theory, Second Edition. New York: American Mathematical Society 1948
- [2] Birkhoff, G. and von Neumann, J.: The logic of quantum mechanics, Ann. Math. **37**, 823–843 (1936)
- [3] Conway, J. H.: Regular Algebra and Finite Machines. London: Chapman and Hall Ltd. 1971

- [4] Finkelstein, D. and Finkelstein, S. R.: Computational complementarity. *International Journal of Theoretical Physics* **22**, 753–779 (1983)
- [5] Grib, A. A. and Zapatin, R. R.: Automata simulating quantum logics. *International Journal of Theoretical Physics* **29**, 113–123 (1990); Macroscopic realizations of quantum logics. *Ibid.* **31**, 1669–1687 (1992).
- [6] Gudder, S.: *Stochastic Methods in Quantum Mechanics*. Amsterdam: North-Holland 1979
- [7] Kalmbach, G.: *Orthomodular Lattices*. New York: Academic Press 1983
- [8] Moore, E.F.: Gedanken-Experiments on Sequential Machines. In: Shannon, C. E. and McCarthy, J. (eds.) *Automata Studies*. Princeton: Princeton University Press 1956
- [9] Navara, M. and Rogalewicz, V.: The pasting constructions for orthomodular posets. *Math. Nachr.* **154**, 157–168 (1991).
- [10] Pták, P. and Pulmannová, S.: *Orthomodular Structures as Quantum Logics*. Amsterdam: Kluwer Academic Publishers 1991
- [11] Svozil, K.: *Randomness and Undecidability in Physics*. Singapore: World Scientific 1993