States on Hilbert Lattices

(Vienna, April 22, 2008)

Mladen Pavičić

`pavicic@grad.hr ; Web: http://m3k.grad.hr/pavicic`

University of Zagreb

Early Ideas and Results

Ancient Result: There is an isomorphism between a *Hilbert lattice* (a complete atomic orthomodular lattice which satisfies the *superpostion princile* and has > 2 atoms) and the set of all closed subspaces of a Hilbert space.

Early Ideas and Results

Ancient Result: There is an isomorphism between a *Hilbert lattice* (a complete atomic orthomodular lattice which satisfies the *superpostion princile* and has > 2 atoms) and the set of all closed subspaces of a Hilbert space.

If we wanted to connect a Hilbert lattice with measured *physical states* of a quantum system described by a Hilbert space equation, we should impose *quantum states* on the lattice.

Early Ideas and Results

Ancient Result: There is an isomorphism between a *Hilbert lattice* (a complete atomic orthomodular lattice which satisfies the *superpostion princile* and has > 2 atoms) and the set of all closed subspaces of a Hilbert space.

If we wanted to connect a Hilbert lattice with measured *physical states* of a quantum system described by a Hilbert space equation, we should impose *quantum states* on the lattice.

What is a *quantum state* on a lattice? Are there *classical states*?

States

Definition. A state on a lattice L is a function $m : \mathrm{L} \longrightarrow [0,1]$ (for real interval $[0,1]$) such that $m(1) = 1$ and $a \perp b \;\Rightarrow\; m(a \cup b) = m(a) + m(b)$, where $a \perp b$ means $a \leq b'$.

States

Definition. A state on a lattice L is a function $m : \mathrm{L} \longrightarrow [0,1]$ (for real interval $[0,1]$) such that $m(1) = 1$ and $a \perp b \;\Rightarrow\; m(a \cup b) = m(a) + m(b)$, where $a \perp b$ means $a \leq b'$.

Definiton. A nonempty set S of states on L is called a strong set of *classical* states if

$$(\exists m \in S)(\forall a, b \in \mathrm{L})((m(a) = 1 \;\Rightarrow\; m(b) = 1) \;\Rightarrow\; a \leq b)$$

States

Definition. A state on a lattice L is a function $m : \mathrm{L} \longrightarrow [0,1]$ (for real interval $[0,1]$) such that $m(1) = 1$ and $a \perp b \ \Rightarrow \ m(a \cup b) = m(a) + m(b)$, where $a \perp b$ means $a \leq b'$.

Definiton. A nonempty set S of states on L is called a strong set of *classical* states if

$$(\exists m \in S)(\forall a, b \in \mathrm{L})((m(a) = 1 \ \Rightarrow \ m(b) = 1) \ \Rightarrow \ a \leq b)$$

and a strong set of *quantum* states if

$$(\forall a, b \in \mathrm{L})(\exists m \in S)((m(a) = 1 \ \Rightarrow \ m(b) = 1) \ \Rightarrow \ a \leq b)$$

Non-Quantum States

There are many *non-quantum states*:

Non-Quantum States

There are many *non-quantum states*:

- *classical* ({0,1}, [0,1], etc.)

Non-Quantum States

There are many *non-quantum states*:

- *classical* ({0,1}, [0,1], etc.)
- *full* but not strong

Non-Quantum States

There are many *non-quantum states*:

- *classical* ({0,1}, [0,1], etc.)
- *full* but not strong
- *states on non-Hilbert lattices* (obvious but nevertheless often persued)

Non-Quantum States

There are many *non-quantum states*:

- *classical* ({0,1}, [0,1], etc.)
- *full* but not strong
- *states on non-Hilbert lattices* (obvious but nevertheless often persued)

Some of these states might be useful.

Usefulness

States that are not “quantum-useful”:

Usefulness

States that are not "quantum-useful":

- *classical* — they turn a Hilbert lattice into a Boolean algebra

Usefulness

States that are not “quantum-useful”:

- *classical* — they turn a Hilbert lattice into a Boolean algebra
- *full but not strong* — they just show that there are othomodular lattices that are not Hilbert ones — with our algorithms and programs we can always generate a pile of them if some application pops up

Usefulness ctnd.

States that might be “quantum-useful”:

Usefulness ctnd.

States that might be “quantum-useful”:

- *states on non-Hilbert lattices*

Usefulness ctnd.

States that might be “quantum-useful”:

- *states on non-Hilbert lattices*
- Frederic F. Shultz, *J. Comb. Theory A* **17**, 317 (1974) $\longrightarrow$ Mirko Navara, *Int. J. Theor. Phys.* **47**, 36 (2008)

Efficiency

A good scenario: states generate Hilbert lattice equations - introduced by Radosław Godowski in 1981

Efficiency

A good scenario: states generate Hilbert lattice equations - introduced by Radosław Godowski in 1981

Originally Godowski used lattices.

Efficiency

A good scenario: states generate Hilbert lattice equations - introduced by Radosław Godowski in 1981

Originally Godowski used lattices.

Lattices are complicated Hasse and Dicht diagrams.

Efficiency

A good scenario: states generate Hilbert lattice equations - introduced by Radosław Godowski in 1981

Originally Godowski used lattices.

Lattices are complicated Hasse and Dicht diagrams.

However, they might be simplifed when considered as Greechie diagrams and hypergraphs

MMP diagrams

They can be further simplified:

MMP diagrams

They can be further simplified:

- only look at atoms within, say, a Hasse diagram and dismiss the whole remaining structure

MMP diagrams

They can be further simplified:

- only look at atoms within, say, a Hasse diagram and dismiss the whole remaining structure

- the atoms uniquely determine the diagrams so we can redefine them

MMP diagrams

They can be further simplified:

- only look at atoms within, say, a Hasse diagram and dismiss the whole remaining structure

- the atoms uniquely determine the diagrams so we can redefine them

We call the diagrams MMP diagrams when they do not refer to any structure—when they are just dots and lines; vertices and edges; atoms and blocks.

MMP definitions

They are graphs defined as follows:

(1) Every atom (vertex, point) belongs to at least one block (edge, line).

(2) If there are at least two atoms then every block is at least 2-element.

(3) Every block which intersects with another block is at least 3-element.

(4) Every pair of different blocks intersects in at most one (two, three) atom(s).

(5) Smallest loops are of order 1 (2,3,4,5)

Finding States

We express the set of constraints imposed by states as a linear programming problem.

Finding States

We express the set of constraints imposed by states as a linear programming problem.

If m is a state, then each 3-atom block with atoms a, b, c imposes the following constraints:

$$\begin{aligned} m(a)+m(b)+m(c) &= 1 \\ m(a')+m(a) &= 1 \\ m(b')+m(b) &= 1 \\ m(c')+m(c) &= 1 \\ m(x) &\geq 0, \qquad x=a,b,c,a',b',c' \end{aligned}$$

Lattice Equation

A *condensed state equation* is an abbreviated version of a lattice equation constructed as follows: all (orthogonality) hypotheses are discarded, all meet symbols, $\cap$, are changed to $+$, and all join symbols, $\cup$, are changed to juxtaposition.

Lattice Equation

A *condensed state equation* is an abbreviated version of a lattice equation constructed as follows: all (orthogonality) hypotheses are discarded, all meet symbols, $\cap$, are changed to $+$, and all join symbols, $\cup$, are changed to juxtaposition.

E.g. $a \perp d \perp b \perp e \perp c \perp f \perp a \Rightarrow$
$(a \cup d) \cap (b \cup e) \cap (c \cup f) = (d \cup b) \cap (e \cup c) \cap (f \cup a)$

Lattice Equation

A *condensed state equation* is an abbreviated version of a lattice equation constructed as follows: all (orthogonality) hypotheses are discarded, all meet symbols, $\cap$, are changed to $+$, and all join symbols, $\cup$, are changed to juxtaposition.

E.g. $a \perp d \perp b \perp e \perp c \perp f \perp a \Rightarrow$
$(a \cup d) \cap (b \cup e) \cap (c \cup f) \ = \ (d \cup b) \cap (e \cup c) \cap (f \cup a)$

which, in turn, can be expressed by the condensed state equation
$ad + be + cf = db + ec + fa.$

States $\longrightarrow$ Equations

When our program finds the states then we obtain various constraints, such as: $m(B)+m(C)+m(1) \le 1; m(2)+m(E)+m(8) = 1$

States $\longrightarrow$ Equations

When our program finds the states then we obtain various constraints, such as:
$m(B)+m(C)+m(1) \leq 1; m(2)+m(E)+m(8) = 1$

From these we obtain a form of condensed state equations, e.g.,
$45 + 9A + E8 + 6D = 56 + 89 + 4A + DE$

States $\rightarrow$ Equations

When our program finds the states then we obtain various constraints, such as:
$m(B)+m(C)+m(1) \leq 1; m(2)+m(E)+m(8) = 1$

From these we obtain a form of condensed state equations, e.g.,
$45 + 9A + E8 + 6D = 56 + 89 + 4A + DE$

Replacing the atoms with variables, the final condensed state equation becomes:
$ab + cd + ef + gh = bg + fc + ad + he$