International Journal of Theoretical Physics, Vol. 39, No. 3, 2000

Quantum Logic for Quantum Computers[†]

Mladen Pavičić[1]

Received December 8, 1999

The following results obtained within a project of finding the algebra of states in a general-purpose quantum computer are reported: (1) All operations of an orthomodular lattice, including the identity, are fivefold-defined; (2) there are nonorthomodular models for both quantum and classical logics; (3) there is a four-variable orthoarguesian lattice condition which contains all known orthoarguesian lattice conditions including six- and five-variable ones. Repercussions to quantum computers operating as quantum simulators are discussed.

1. INTRODUCTION

A computer is a computational device in which 2×2 unitary matrices called *logical gates* act on elementary bits $|0\rangle = (1, 0)$ and $|1\rangle = (0, 1)$ and on bits obtained by such operations. A classical gate is, for example, a NOT gate, which flips bits in the following way: $\mathsf{NOT}|0\rangle = \mathsf{NOT}(1, 0) = |1\rangle$ and $\mathsf{NOT}|1\rangle = \mathsf{NOT}(0, 1) = |0\rangle$ and which can be represented as

$$\mathsf{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{1}$$

A quantum gate which is characteristic of existing experimental hardware is the *controlled* NOT gate, which acts on two qubits in a conditional way [as simple NOT gate on the second (target) qubit provided the first (control) qubit is 1] as follows:

[†] This paper is dedicated to the memory of Gottfried T. Rüttimann.

[1] Department of Mathematics, University of Zagreb, GF, Kačićeva 26, Zagreb, Croatia; E-mail: mpavicic@faust.irb.hr; web page: http://m3k.grad.hr/pavicic and Department of Physics, University of Maryland Baltimore County, Baltimore, MD 21250, USA; E-mail: pavicic@umbc.edu.

0020-7748/00/0300-0813$18.00/0 © 2000 Plenum Publishing Corporation

$$\mathsf{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2}$$

The transformation CNOT—and all other classical operations transformed to quantum gates by making them *controlled* ones—are obviously unitary, they preserve superpositions, and they cannot be decomposed into a tensor product of two single-bit transformations, but without qubit rotations and without phase shifts,

$$\begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}, \quad \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix}, \quad \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \tag{3}$$

genuine quantum tasks cannot be processed. For example, even the simplest problem of a photon passing two successive polarizers (quantum Malus law) could not be solved. On the other hand, these nonclassical rotations and phase shifts essential for quantum computers depend on classical continuous variables and this causes problems which focus on later.

An even more essential difference between classical and quantum computers is contained in elementary information units themselves. A classical unit is always either 0 or 1 (one bit). A quantum unit—called a *qubit*—is a two-state quantum system. We describe the system by a unit vector in the Hilbert space $\mathscr{H}^2$ over the field of complex numbers. We denote the two orthogonal states by $|0\rangle = (1, 0)$ and $|1\rangle = (0, 1)$. The states make an orthogonal basis for $\mathscr{H}^2$. In a quantum computer we deal with a large number n of qubits which build up a composite Hilbert space $\mathscr{H} = \mathscr{H}^2 \otimes \ldots \otimes \mathscr{H}^2$. The computational basis, i.e., the basis of this space, consists of the following 2^n vectors: $|00 \cdots 00\rangle$, $|00 \cdots 01\rangle, \ldots, |11 \cdots 11\rangle$, where, e.g., $|00\rangle$ means $|0\rangle \otimes |0\rangle$. Classical bits correspond to quantum states: $i_1 i_2 \cdots i_n \leftrightarrow |i_n\rangle \equiv |i_1 \cdots i_n\rangle$.

To compute the function $f\colon i_1 i_2 \ldots i_n \mapsto f(i_1, \ldots, i_n)$ means to let the corresponding states evolve according to the time evolution unitary operator U:

$$|i_1 i_2 \cdots i_n\rangle \mapsto U|i_1 i_2 \cdots i_n\rangle = |\, f(i_1, \ldots, i_n)\rangle \tag{4}$$

More explicitly,

$$|\Psi_f\rangle = \exp\left(-\frac{i}{\hbar}\int \mathscr{H}\, dt\right)|\Psi_o\rangle = U|\Psi_o\rangle \tag{5}$$

which follows directly from the Schrödinger equation. The unitarity of U assures reversibility and therefore prevents energy dissipation. This can be achieved with classical devices as well, but only at the cost of exponentially growing hardware or exponentially increasing time. The reason is simple: n classical states describing a system in a classical computer can only be specified by ascribing values to all 2^n basis states. Quantum computers, on the other hand, achieve the aim as well as a parallel way of computing—which is their most attractive feature—by using superposition, which puts n quantum states in a superposition of all 2^n basis states in one step. Again, for a parallel computation a classical computer would need to either exponentially growing hardware or exponentially increasing time.

Consider, for example, the following state of two particles, known as the *entangled* state (Pavičić and Summhammer, 1994; Pavičić, 1995) of the particles, which can then also be used for a teleportation of states or Bell experiments or quantum cryptography (we omit the normalization factors throughout):

$$|00\rangle + |11\rangle \tag{6}$$

Here neither of the two qubits has a definite state: the state of the system is not a tensor product of the states, and we cannot find a_1, a_2, b_1, b_2 such that

$$(a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) = |00\rangle + |11\rangle$$

since

$$\begin{aligned}&(a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle)\\ &\qquad = a_1a_2|00\rangle + a_1b_2|01\rangle + b_1a_2|10\rangle + b_1b_2|11\rangle\end{aligned}$$

and $a_1b_2 = 0$ implies that either $a_1a_2 = 0$ or $b_1b_2 = 0$. These states represent situations that have no classical counterpart. These are also states that provide exponential growth of quantum state spaces with the number of particles.

To see this let us consider the following superposition of n qubits:

$$\sum_{i_1i_2\cdots i_n=0}^{1} |i_1i_2 \cdots i_n\rangle \tag{7}$$

Applying the linear unitary operation which computes f, from Eq. (4), to this state yields

$$\sum_{i_1i_2,\ldots,i_n=0}^{1} |f(i_1i_2 \cdots i_n)\rangle \tag{8}$$

Hence, U computes f *parallelly* on all the 2^n possible inputs i.

To achieve such parallel computing in a realistic computer, we start with an initial state $|i\rangle$ which corresponds to an "input" to the computation. We then perform elementary operations on the system using the quantum gates defined above. The operations correspond to the computational steps in the computation, just like logical gates are the elementary steps in classical computers, and are performed on an isolated system, so the evolution can always be described by a unitary matrix operating on the state of the system.

Therefore we can always implement a unitary operator which is given by the Hamiltonian of a given process or state of a system as a set of instructions on how to transform input states in time. But the crucial problem is initial states themselves. Can we write down a general input state

$$|\Psi_0\rangle = |i_1 i_2 \cdots i_n\rangle \tag{9}$$

by means of quantum gate operations over elementary input propositions so as to correspond to a general wave function of the Hilbert space which describes the input state? The answer is currently in the negative. There is no known finite and definite recipe for such a correspondence. But in recent years much has been achieved to narrow the gap between an algebra of elementary propositions (corresponding to pure states) and the Hilbert space description. Let us consider the most important details in a possible construction of a quantum machine language which could mimic a quantum system and therefore directly correspond to its Hilbert space representation.

In Section 3 we show (using only lattice theory) that both a proper quantum logic of propositions and a proper classical logic of propositions have models that are not even orthomodular. Therefore in both classical and quantum computers one must use algebras instead. In Section 2 we analyze such an algebra—orthomodular lattice—which is usually considered to be an algebra underlying quantum measurement and a Hilbert space representation, and show that all its binary operations are ambiguous and that bare orthomodular lattices cannot be employed in quantum computers. In Section 4 we show how one can construct Hilbert lattices which enable a direct representation in a Hilbert space of quantum computational simulation and provide a lattice for the purpose, which at the same time eliminates the so-called 4-dim postulate. We also show that quantum theory is at least as incompatible with the strong form of the Church–Turing principle as any classical theory, contrary to Deutsch's claim (Deutsch, 1985).

2. ALL OPERATIONS IN ORTHOMODULAR LATTICES ARE AMBIGUOUS

The Birkhoff–von Neumann requirement (Kalmbach, 1983)

$$a \to_i b \quad \Rightarrow a \le b, \qquad i = 1, \ldots, 5 \tag{10}$$

where $a \to_1 b \overset{\text{def}}{=} a' \cup (a \cap b)$, $a \to_2 b \overset{\text{def}}{=} b' \to_1 a'$, $a \to_3 b \overset{\text{def}}{=} (a' \cap b) \cup$

$(a' \cap b') \cup (a \rightarrow_1 b)$, $a \rightarrow_4 b \overset{\text{def}}{=} b' \rightarrow_3 a'$, and $a \rightarrow_5 b \overset{\text{def}}{=} (a \cap b) \cup (a' \cap b) \cup (a' \cap b')$, not only holds in every orthomodular lattice, but also amounts to the orthomodularity itself in the sense that condition (10) added to an ortholattice makes it orthomodular (Pavičić, 1987)

Since in any orthomodular lattice (Pavičić and Megill, 1998a)

$$a \cup b = (a \rightarrow_i b) \rightarrow_i (((a \rightarrow_i b) \rightarrow_i (b \rightarrow_i a)) \rightarrow_i a), \qquad i = 1, \ldots, 5 \tag{11}$$

this means that all operations (fivefold negation follows trivially) in an orthomodular lattice are fivefold-definable.

At first sight it still seems that one can prove a conjecture that the relation of equation in the lattice is uniquely definable. The reasons for such a conjecture are the following. All five quantum implications $a \rightarrow_i b$ collapse to the classical one $a \rightarrow_0 b \overset{\text{def}}{=} a' \cup b$ in a distributive lattice. Even more,

$$a \rightarrow_i b = a \rightarrow_j b, \qquad i \neq j, \quad i, j = 0, \ldots, 5 \tag{12}$$

makes an ortholattice distributive (Pavičić, 1987). On the other hand,

$$a \rightarrow_0 b \Rightarrow a \leq b \tag{13}$$

also makes an ortholattice distributive (Pavičić, 1998). In any orthomodular lattice we have

$$a \leftrightarrow_i b = a \equiv b, \qquad i = 1, \ldots, 5 \tag{14}$$

where $a \leftrightarrow_i b \overset{\text{def}}{=} (a \rightarrow_i b) \cap (b \rightarrow_i a)$ and $a \equiv b \overset{\text{def}}{=} (a \cap b) \cup (a' \cap b')$. The identity operation $a \equiv b$ reduces to $a \equiv_0 b \overset{\text{def}}{=} (a' \cup b) \cap (b' \cup a)$ in a distributive theory and since $a \equiv b$ is an equivalence relation in an othomodular lattice and $a \equiv_0$ is an equivalence relation in a distributive lattice, we could hope that the conjecture does hold, i.e., that the relation of equation '$=$' in orthomodular lattices can be uniquely defined and connected to the operation of identity by the rule

$$a \equiv b = 1 \Leftrightarrow a = b \tag{15}$$

which is known to make an ortholattice orthomodular (Pavičić, 1993) and which can be compared to the rule

$$a \equiv_0 b = 1 \Leftrightarrow a = b \tag{16}$$

which makes an ortholattice distributive (Pavičić, 1998).

Unfortunately, the conjecture does not hold. The reason is simple. In a distributive lattice $a \rightarrow_i b$, $i = 1, \ldots, 5$, all merge to $a \rightarrow_0 b$ and therefore $(a \rightarrow_i b) \cap (b \rightarrow_j a)$, $i \neq j$, $i, j = 1, \ldots, 5$, must merge to $a \equiv_0 b \overset{\text{def}}{=} (a \rightarrow_0$

$b) \cap (b \to_0 a)$. But in an orthomodular lattice the former biimplications $(a \to_i b) \cap (b \to_j a)$ are equal—depending on the values of i and j—to the following *five* identities: $a \equiv b$, $a \equiv_1 b \overset{\text{def}}{=} (a \cup b') \cap (a' \cup (a \cap b))$, $a \equiv_4 b \overset{\text{def}}{=} (a \cup b') \cap (b \cup (a' \cap b'))$, $a \equiv_3 b \overset{\text{def}}{=} (a' \cup b) \cap (a \cup (a' \cap b'))$, and $a \equiv_3 b \overset{\text{def}}{=} (a' \cup b) \cap (b' \cup (a \cap b))$ as given in the Table I (Pavičić and Megill, 1999).

The expressions $a \equiv_i b$, $i = 1, \ldots, 4$, are all asymmetrical and at first we would think it would be inappropriate to name them identities. Also, $a \equiv_i b = a \equiv_j b$ when added to an ortholattice does not make it even orthomodular, but apparently weakly distributive (see next section), as opposed to Eq. (12). Nevertheless, we are able to prove the following theorem (Pavičić and Megill, 1999).

Theorem 2.1. An ortholattice in which

$$a \equiv_i b = 1 \Rightarrow a = b, \qquad i = 1, \ldots, 4 \tag{17}$$

holds is an orthomodular lattice and vice versa.

Hence, putting together Eq. (15) and Eq. (17), we have an indication that the relation of equivalence which establishes a connection between quantum logic and its models might turn out to be based on several different operations of identity at the same time, thus making a direct evaluation of elementary logical propositions impossible. However, as we will see in the next section, there is an even more important reason why we cannot use proper quantum logic to evaluate quantum propositions, and this is that a proper quantum logic is not orthomodular. An even bigger surprise is the result that even standard classical logic need not be orthomodular.

3 NONORTHOMODULAR MODELS FOR BOTH QUANTUM AND CLASSICAL LOGICS

A crucial difference between logics and lattices as their models is that properties that play a decisive role in lattices do not play such a role in logics

Table I. Products $(a \to_i b) \cap (b \to_j a)$, $i = 0, \ldots, 5$ (Rows), $j = 0, \ldots, 5$ (Columns)[a]

$i\ j$	$b \to_0 a$	$b \to_1 a$	$b \to_2 a$	$b \to_3 a$	$b \to_4 a$	$b \to_5 a$
$a \to_0 b$	$a \equiv_0 b$	$a \equiv_4 b$	$a \equiv_3 b$	$a \equiv_2 b$	$a \equiv_1 b$	$a \equiv b$
$a \to_1 b$	$a \equiv_1 b$	$a \equiv b$	$a \equiv b$	$a \equiv b$	$a \equiv_1 b$	$a \equiv b$
$a \to_2 b$	$a \equiv_2 b$	$a \equiv b$	$a \equiv b$	$a \equiv_2 b$	$a \equiv b$	$a \equiv b$
$a \to_3 b$	$a \equiv_3 b$	$a \equiv b$	$a \equiv_3 b$	$a \equiv b$	$a \equiv b$	$a \equiv b$
$a \to_4 b$	$a \equiv_4 b$	$a \equiv_4 b$	$a \equiv b$	$a \equiv b$	$a \equiv b$	$a \equiv b$
$a \to_5 b$	$a \equiv b$	$a \equiv b$	$a \equiv b$	$a \equiv b$	$a \equiv b$	$a \equiv b$

[a] "Identities" $a \equiv_i b$, $i = 1, \ldots, 4$, are asymmetrical.

at all. To explain this difference, let us consider the orthomodularity and distributivity properties. When we add the orthomodularity (distributivity) property to an ortholattice it becomes an orthomodular (distributive) lattice. We can compare what happens in a logic by looking at a lattice we obtain by mapping logical axioms $\vdash A$ to an ortholattice, where they take the form $a = 1$; here $a = v(A)$ and v is a morphism from the logic to the lattice. As we have shown (Pavičić and Megill, 1998b), the property $(a \cup (a' \cap (a \cup b))) \equiv (a \cup b) = 1$ that we obtain by mapping the logical formula for "orthomodularity" $\vdash(A \vee (\neg A \wedge (A \vee B)) \equiv (A \vee B)$ into an ortholattice is true in all ortholattices. On the other hand, as we have shown (Pavičić and Megill, 1999), $(a \cap (b \cup c)) \equiv_0 ((a \cap b) \cup (a \cap c)) = 1$, which we obtain by an analogous mapping of the distributivity, is true in all weakly distributive lattices which are not even orthomodular.

The reason for such different structures of logics as opposed to their models lies in completely different syntax of $a = 1$ lattice equations (which correspond to logical wwf's) and $a = b$ lattice equations. For example, another way of expressing orthomodularity is $\vdash A \vee (B \wedge (\vdash A \vee \vdash B)) \equiv A \vee B$, whose lattice mapping is $((a \rightarrow_1 b) \rightarrow_0 b) \equiv (a \cup b) = 1$, which, when added to an ortholattice, makes it weakly orthomodular. This means that the "orthomodularity" from quantum logic sometimes maps to an ortho-property and sometimes to a weakly orthomodular property, but never to an orthomodular property. The reader can find details on weakly orthomodular logics and lattices in Pavičić and Megill (1998b, 1999).

Classical logic also does not necessarily map its syntactical structure to its model. More precisely, it does if valuated on {0,1} or if valuated by classical Kolmogorovian probability functions. Therefore, our results show that for classical logics there are nonorthomodular models which do not use {0,1} valuation of propositions. Since all standard applications of classical logic invoke exactly such valuation, the above discovery most probably will not have serious repercussions for classical reasoning and computing. Quantum logic as well as orthomodular lattices, on the other hand, in principle cannot ascribe definite values to their propositions and cannot have {0, 1} valuation at all. This might have serious repercussions for quantum computers, as we will see in the next section. The reader can find detailed soundness and completeness proofs for both quantum and classical logics in Pavičić and Megill (1999).

4 QUANTUM ALGEBRA FOR QUANTUM COMPUTERS

Computational instructions to a quantum computer for handling inputs to give desired outputs have been simply called quantum logic (Christianson *et al.*, 1998). The latter logic, however, cannot be a proper logic, especially if we want it to be a general machine language capable of solving and simulating any given Hamiltonian. Recently devised algorithms such as factorization of big numbers in cryptography (Shor, 1997) or searching big databases in networks (Grover, 1997) are certainly ingenious, but do not use any general quantum algebra. They make direct use of hardware-prepared and hardware-processed input states. In order to build up a general quantum algebra, input states must satisfy additional conditions which do not result from qubit superposition, entanglement, and rotation and phase shift control. Algebraically these conditions amount to an extension of orthomodular lattices which we call the Hilbert lattice (HL) and will consider in this section as a structure isomorphic to a Hilbert space description of an arbitrary quantum system.

Classical computer states obey all the conditions required by the Boolean algebra (distributivity, etc.). As opposed to this, quantum computer states which appear in the known algorithms (e.g., Shor's and Grover's) do not obey all the conditions required by HL. On the other hand, it is still unclear how one can implement HL conditions in a quantum computer. So, even the Schrödinger equation, describing the evolution of states in a quantum computer, must be simulated by a specially designed approximative algorithm (Boghosian and Taylor, 1998), Such a quantum computer is therefore still not what it could eventually be: a quantum simulator which mimics quantum systems by giving precise instructions on how to produce input states, how to evolve them, and how to read off the final states (Feynman, 1982, 1986). Let us analyze conditions which quantum states should obey in order to enable full quantum computing, i.e., proper quantum mathematics.

In order to enable an isomorphism between an orthocomplemented orthomodular lattice and the corresponding Hilbert space we have to add further conditions to the lattice. The conditions correspond to the essential properties of any quantum system such as superposition. Combining Holland (1995) and Ivert and Sjödin (1978), we can state the conditions as follows:

Definitional (the first 3) and Additional Conditions for a Hilbert Lattice

- *Completeness*: The meet and join of any subset of a lattice always exist.
- *Atomicity*: Every nonzero element in HL majorizes an atom which is a nonzero element $a \in \mathrm{HL}$ with $0 < b \leq a$ only if $b = a$.
- *Superposition principle:* (Atom c is a superposition of atoms a and b if $c \neq a$, $c \neq b$, and $c \leq a \cup b$.)

  1. Given two different atoms a and b, there is at least one other atom c, $c \neq a$ and $c \neq b$, that is a superposition of a and b.

2. If the atom *c* is a superposition of the distinct atoms *a* and *b*, then *a* is a superposition of *b* and *c*.

- *Unitary operators*: Given any two orthogonal atoms *a* and *b* in HL, there is a unitary operator *U* such that $U(a) = b$.
- *Infinite orthogonality:* HL contains a countably infinite sequence of orthogonal elements.

It is well known that if the HL is of dimension ≥4, then there exists a field *F* and a vector space *E* over *F* such that HL is orthoisomorphic to the lattice L_E of *E*-closed subspaces of *E*.

In an "orthomodular approach" the condition ≥4 must be postulated (Maczyński, 1972). But if we found a condition which must be satisfied in HL and which requires at least four nonequivalent variables, then the condition would be automatically satisfied. A natural idea is to look for conditions equivalent to those in Eqs. (15) and (16) with new "full quantum identities" which would solve the ambiguity problem of the relation of equality '=' and of lattice operations. The following definitions do the job.

Definition 4.1:

$$a \overset{c}{\equiv}_i b \overset{\text{def}}{=} ((a \to_i c) \cap (b \to_i c)) \cup ((a' \to_i c) \cap (b' \to_i c)); \quad i = 1, 3, \quad (18)$$

$$a \overset{c}{\equiv}_i b \overset{\text{def}}{=} ((c \to_i a) \cap (c \to_i b)) \cup ((c \to_i a') \cap (c \to_i b')); \quad i = 2, 4, \quad (19)$$

$$a \overset{c,d}{\equiv}_i b \overset{\text{def}}{=} a \overset{d}{\equiv}_i b \cup (a \overset{d}{\equiv}_i c \cap b \overset{d}{\equiv}_i c) \quad i = 1, \ldots, 4. \quad (20)$$

Theorem 4.2. An ortholattice to which

$$a \overset{c}{\equiv}_i b = 1 \Leftrightarrow a \to_i c = b \to_i c, \qquad i = 1, 3, \quad (21)$$

$$a \overset{c}{\equiv}_i b = 1 \Leftrightarrow c \to_i a = c \to_i b, \qquad i = 2, 4 \quad (22)$$

are added is a variety of OML which fails in lattice $\hat{L}$ (Fig. 1a) for $i = 1$, 2, 3, 4.

Theorem 4.3. An ortholattice to which

$$a \overset{c,d}{\equiv}_i b = 1 \Leftrightarrow a \to_i d = b \to_i d, \qquad i = 1, 3 \quad (23)$$

is added is a variety of OML which fails in lattice *L38* (Fig. 1b).

The reader can check that the equations really fail in the quoted lattices (and million of others with up to 50 blocks) after compiling *lattice.c*[2] written in C (McKay, Megill, and Pavičić, 2000).

[2] ftp://ftp.shore.net/members/ndm/quantum-logic.

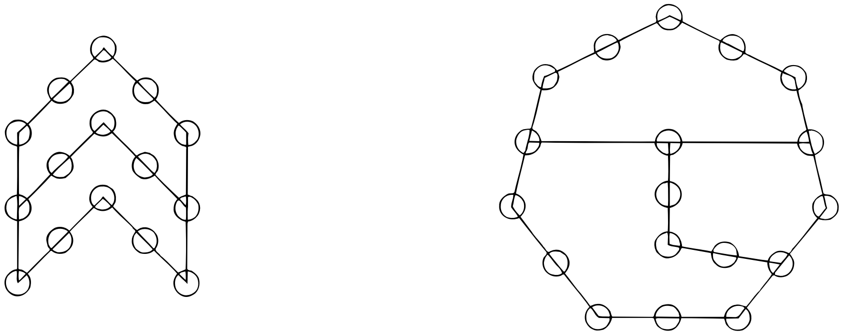

Fig. 1. (a) $\hat{L}$ from Godowski and Greechie (1984); (b) *L38* from Pavičić and Megill (1999).

The new identities $\overset{c}{\equiv}_1$ and $\overset{c,d}{\equiv}_1$ when equal to one are relations of equivalence. The previous theorems narrow down the ambiguity of operations $\to_i$ (and therefore of relations $\le$ and $=$ as well) to two. The role of the Sasaki projection $\phi_a b = (a \to_1 b')'$ of b on a in the the covering property which is a consequence of the superposition principle then apparently resolves the ambiguity completely.

In the end we are able to prove that the previous theorems follow from the following one (Megill and Pavičić, 2000).

Definition 4.4. A 3OA is an OML in which the following additional condition is satisfied:

$$(a \to_1 c) \cap (a \overset{c}{\equiv}_1 b) \le (b \to_1 c) \tag{24}$$

A 4OA is an OML in which the following additional condition is satisfied:

$$(a \to_1 d) \cap (a \overset{c,d}{\equiv}_1 b) \le (b \to_1 d) \tag{25}$$

Theorem 4.5. Every 4OA is a 3OA, but there exist 3OAs that are not 4OAs. 4OA fails in *L38* and $\hat{L}$ and 3OA only in $\hat{L}$ (Fig. 1a). Equation (22), $i = 1$, follows from Eq. (24), and Eq. (23), $i = 1$, follows from Eq. (25).

The 4OA law (25) is equivalent to the orthoarguesian law discovered by A. Day (cf. Godowski and Greechie, 1984). Thus the orthoarguesian law may be expressed by an equation with only four variables instead of six. In addition, we are able to prove that apparently all known orthoarguesian derivates follow from, or are identical to either 4OA or 3OA laws given above (Megill and Pavičić, 2000).

We therefore obtained the result that HL must be of dimension ≥ 4 and that therefore, with the aforecited additional conditions, is orthoisomorphic to the lattice of subspaces of a Hilbert space. On the other hand, as a consequence of the aforestated additional conditions, we obtain that the number of atoms of a lattice (pure states) of any Hilbert space of dimension ≥ 3 must be infinite (Ivert and Sjödin, 1978). This is in a direct relation to a coordinatization of Hilbert spaces. For example, if we want to have a complete description of a spin-1 system, we cannot achieve this in the spin space alone. We have to include the orientations of preparations and measurements in space (otherwise we would not have even the Malus law) in our description and these are continuous variables. In a qubit preparation within a quantum computer the continuous variable is the angle α in Eq. (3). Thus the lattice is infinite although the number of input qubits and the unitary transformation needed to calculate the result of a measurement remains finite-dimensional (Deutsch, 1985). This invalidates Deutsch's claim: "[‘Quantum’ Church–Turing principle] is so strong that it is *not* satisfied by Turing's machine in classical physics. Owing to continuity of classical dynamics, the possible states of a classical system necessarily form a continuum. Yet there are only countably many ways of preparing a finite input for [a quantum Turing machine]. Consequently [it] cannot perfectly simulate any classical system."

This distinction does not hold because, as we have seen, a continuum appears with quantum spin systems as well and on the other hand preparing a finite input does not contradict the existence of a continuum of possible states within a lattice. Infinite number of states does not mean an infinite number of calculated spin projections for a quantum system or positions and momentoms for a classical system. The infinity contained in the continuous variables is actually not a problem, but an essential feature which enables the Hilbert space representation with the help of M. P. Solèr's recent discovery: we need not *postulate* (as was considered necessary until several years ago) a complex (or real or quaternionic) field for our Hilbert space—it *follows* from the infinite orthogonality of the lattice (Holland, 1995).

5 CONCLUSION

States of general-purpose quantum computer must, apart from conditions imposed by the standard quantum logical gates, satisfy additional conditions given in Section 4 and required to yield a general algebra of the states. One of the conditions is also the four-variable orthoarguesian law given by Eq. (25) which gives all known orthoarguesian equations (including six-variable ones) and which eliminates the so-called 4-dim lattice postulate. The obtained algebra, which is a Hilbert lattice, is then isomorphic to the subspaces of the Hilbert spaces which characterize general computation algorithms. Propositions a, b, c of the lattice are therefore connected to probabilistic outcomes of a calculation of an observable A by means of $\mu(a) = \langle\Psi|P_A|\Psi\rangle$, where $P_{A,E}$ (E is a Borel set) is a projector of A, μ is the pure full [$\mu(a) \leq \mu(b)$

$\Rightarrow a \leq b$] strongly convex ($\mu_j \in \mathcal{S}$ & $\Sigma\, c_j = 1 \Rightarrow \Sigma\, c_j\mu_j \in \mathcal{S}$) probability measure on HL: μ: HL $\mapsto [0, 1]$, and we obtain Ψ from the Gleason theorem: if μ is a pure probability measure, then there exists a vector $\Psi \in \mathcal{H}$ which satisfies the above $\mu(a)$. The mean value of the operator A is then given by the spectral theorem. However, there are several obstacles to a direct calculation of this mean value on a quantum computer.

In Section 3 we showed that no calculation can be carried out within propositional quantum logic since the latter can be modeled with a nonorthomodular model. (In addition, we show that the standard classical logic has a nonorthomodular model, too, and explain why this is of no consequence for classical computers.)

In Section 2 we showed that a bare orthomodular lattice cannot be used as a satisfactory algebra of states because all operations in the lattice are fivefold-defined, including the identity and the relation of equivalence.

Taken together, a quantum computer could simulate quantum systems as described by infinite dimensional Hilbert space, if one found a way how to substitute Hilbert lattice equations for the conditions from Section 4. Alternative route would be to formulate a description of quantum systems by means of finite dimensional Hilbert space.

REFERENCES

Boghosian, B. M., and Taylor, W. (1998). Simulating quantum mechanics on a quantum computer, *Physica D* **120**, 30–42.

Christianson, B., Knight, P. L., and Beth, T. (1998). Implementations of quantum logic, *Phil. Trans. R. Soc. Lond. A* **356**, 1823–1838.

Deutsch, D. (1985). Quantum theory, the Church–Turing principle and the universal quantum computer, *Proc. R. Soc. Lond. A* **400**, 97–117.

Feynman, R. P. (1982). Simulating physics with computers, *Int. J. Theor. Phys.* **21**, 467–488.

Feynman, R. P. (1986). Quantum mechanical computers, *Found. Phys.* **16**, 507–531.

Godowski, R., and Greechie, R. (1984). Some equations related to the states on orthomodular lattices, *Demonstratio Math.* **17**, 241–250.

Grover, L. K. (1997). Quantum computers can search arbitrarily large databases by a single query, *Phys. Rev. Lett.* **79**, 4709–4712.

Holland, Jr., S. S. (1995). Orthomodularity in infinite dimensions; a theorem of M. Solèr, *Bull. Am. Math. Soc.* **32**, 205–234.

Ivert, P.-A., and Sjödin, T. (1978). On the impossibility of a finite propositional lattice for quantum mechanics, *Helv. Phys. Acta* **51**, 635–636.

Kalmbach, G. (1983). *Orthomodular Lattices*, Academic Press, London.

McKay, B. D., Megill, N. D., and Pavičić, M. (2000). Isomorph-free exhaustive generation of Greechie diagrams and automated checking of their passage of orthomodular lattice equations, *Int. J. Theor. Phys.*, in press.

Maczyński, M. J. (1972). Hilbert space formalism of quantum mechanics without the Hilbert space axiom, *Rep. Math. Phys.* **3**, 209–219.

Megill, N. D. and Pavičic, M. (2000). Equations and state properties that hold in all closed subspaces of an infinite dimensional Hilbert space, *Int. J. Theor. Phys.*, in press.

Pavičić, M. (1987). Minimal quantum logic with merged implications, *Int. J. Theor. Phys.* **26**, 845–852.

Pavičić, M. (1993). Nonordered quantum logic and its YES–NO representation, *Int. J. Theor. Phys.* **32**, 1481–1505.

Pavičić, M. (1995). Spin-correlated interferometry with beam splitters: Preselection of spin-correlated photons, *J. Opt. Soc. Am. B* **12**, 821–828.

Pavičić, M. (1998). Identity rule for classical and quantum theories, *Int. J. Theor. Phys.* **37**, 2099–2103.

Pavičić, M., and Megill, N. D. (1998a). Quantum and classical implication algebras with primitive implications, *Int. J. Theor. Phys.* **37**, 2091–2098.

Pavičić, M., and Megill, N. D. (1998b). Binary orthologic with modus ponens is either orthomodular or distributive, *Helv. Phys. Acta* **71**, 610–628.

Pavičić, M., and Megill, N. D. (1999). Non-orthomodular models for both standard quantum logic and standard classical logic: Repercussions for quantum computers, *Helv. Phys. Acta* **72**, 189–210.

Pavičić, M., and Summhammer, J. (1994). Interferometry with two pairs of spin correlated photons, *Phys. Rev. Lett.* **73**, 3191–3194.

Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comp.* **26**, 1484–1509.