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

Quantum Logic as a Basis for Computations†

Jarosław Pykacz[1]

Received December 8, 1999

It is shown that computations can be founded on the laws of the genuine (Birkhoff–von Neumann) quantum logic treated as a particular version of Łukasiewicz infinite-valued logic. A new way of encoding nonexact data which encodes both the value of a number and its "fuzziness" is introduced. A simple example of a full adder that works in the proposed way is given and it is compared with other designs of quantum adders existing in the literature. A controversy between the meaning of the very term "quantum logic" as used recently within the theory of quantum computations and the traditional meaning of this term is briefly discussed.

1. INTRODUCTION

In the rapidly developing theory of quantum computations (QC) it is sometimes claimed that QC should be (Peres, 1985) or that they are (Vedral and Plenio, 1998) based on a new kind of nonclassical (non-Boolean) logic. This hypothetical logic is sometimes called *quantum logic* (Turchette *et al.*, 1995; Vedral and Plenio, 1998) in spite of the fact that since Birkhoff and von Neumann's (1936) historic paper this term has had a defined (although not unique) meaning: It usually denotes an *orthomodular lattice* (sometimes more general *orthomodular poset*) admitting an *order determining set of probability measures*, i.e., it denotes an algebraic structure that mimics the order-theoretic properties of the set of all closed subspaces of a Hilbert space. Contrary to this "orthodox" meaning of the term *quantum logic* adopted throughout the traditional physical, mathematical, logical, and philosophical literature (see the nearly 2000 entries in Pavičić's (1992) "Bibliography on quantum logics and related structures"), the meaning ascribed to this term

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

[1] Instytut Matematyki, Uniwersytet Gdański, 80-952 Gdańsk, Poland; e-mail: pykacz@ksinet.univ.gda.pl.

within the QC theory is rather vague, which, for the time being, does not allow one to compare these two objects. Such a comparison is of course a very important problem, but it will be possible only when *quantum logic in the QC sense* is precisely defined. Therefore, throughout the rest of this paper the term *quantum logic* (usually abbreviated as QL) will be used exclusively in its original sense.

In spite of the above-mentioned objections, QC theorists are right when they claim that some operations on quantum bits (qubits), especially those that yield superpositions of qubits, and also the superpositions of qubits themselves "have no logical meaning from the point of view of conventional computer science" (Peres, 1985). Nevertheless, some logic gates frequently used within the QC theory, like NOT, controlled-NOT, and Toffoli gates, work in a purely classical way and of course the same refers to any computational network constructed exclusively out of these gates. In particular, a full adder, which, as a touchstone of any theory of computations, is desribed in the QC papers by Peres (1985) and Vedral *et al.* (1996, 1998), according to Peres (1985), is "effectively classical" since superpositions of qubits "appear in the *dynamics* but not in the *logic* of the quantum computer." Let us note that in the case of these adders the nonclassical operation of forming superpositions may appear only at the stage of preparing the input data, but, contrary to some other, more sophisticated quantum algorithms (see, e.g., Aharonov, 1998; Vedral *et al.*, 1996, 1998; Rieffel and Polak, 1998), it does not appear within the algorithm itself.

The problem of establishing connections between a QL and a logic that underlies more sophisticated quantum algorithms is a very interesting one and will surely attract much attention in the future, as soon as the latter object is fully recognized and precisely defined. In the present paper, which is a step toward fulfilling Peres (1985) program to "try to generalize computer science, so that it would admit a continuous logic where $\alpha|\uparrow\rangle + \beta|\downarrow\rangle$ (with complex coefficients α and β) would have a meaning," I will give an example of a full adder that works according to the laws of a QL instead of a two-element Boolean algebra. The paper grew out of the author's part of a joint contribution (Pykacz and Zapatrin, 1997) in which it was, however, erroneously claimed that all previously studied quantum computations were based on the classical logic. I am greatly indebted to an anonymous referee of the present paper for drawing my atttention to the fact that such a judgement is, in general, unfounded.

The paper is devoted to the very general problem of the possibility of performing computations with the use of a QL instead of a two-element Boolean algebra. The studies are confined to a purely mathematical description of such computations, and the possibility of their physical realization is not studied (therefore, the standard QC formalism is not adopted here).

However, since a QL is believed to be the "intrinsic" logic of quantum systems, I do hope that such a possibility could be found in the future. The given example shows that a computing machine that would work according to the laws of a QL would accommodate in a natural way "probabilistic" (or "fuzzy") features of quantum theory. Therefore, it would be automatically a "fuzzy" computer well suited to deal with vague or imprecise data.

2. QL AS A FAMILY OF FUZZY SETS AND AS MANY-VALUED LOGIC

Birkhoff and von Neumann (1936), the founding fathers of the QL theory, in their historic paper noticed that a lattice of closed subspaces of a Hilbert space (or, equivalently, a lattice of orthogonal projections onto these subspaces), which is a standard example of a QL, forms an order-theoretic structure which is a proper generalization of a Boolean algebra. Since Lindenbaum–Tarski algebra of any theory governed by laws of the classical logic is a Boolean algebra, Birkhoff and von Neumann concluded that the logical structure of quantum mechanics is nonclassical.

Since 1936 the QL theory has greatly evolved and branched into a multitude of approaches such that the term "quantum logic" does not have a unique meaning throughout the literature. In the present paper by a *quantum logic* I mean a *partially ordered, orthocomplemented, orthomodular set* (abbreviated *orthomodular poset*) admitting an *order determining set of probability measures*. Therefore, a QL is an order-theoretic structure which is a nondistributive generalization of a Boolean algebra. The reader interested in exact definitions of these notions is referred to any of numerous textbooks on the QL theory, for example, Beltrametti and Cassinelli (1981) or Pták and Pulmannová (1991).

Formal similarity of some operations on fuzzy sets to order-theoretic operations defined on orthomodular posets led the author to study the possibility of representing an abstract QL in a form of a suitable family of fuzzy sets. These attempts were completed in Pykacz (1994), where it was shown that any orthomodular poset L admitting an ordering set of probability measures S can be isomorphically represented as a family $\mathcal{L}(S)$ of fuzzy subsets of S such that:

(i) $\mathcal{L}(S)$ contains the empty set $\emptyset$.
(ii) $\mathcal{L}(S)$ is closed under the standard fuzzy set complementation, i.e.,

$$A \in \mathcal{L}(S) \quad \text{implies} \quad A' = 1 - A \in \mathcal{L}(S) \tag{1}$$

(iii) $\mathcal{L}(S)$ is closed under Giles unions of pairwisely *weakly disjoint* sets, i.e., if $A_i \sqcap A_j = \emptyset$ for $i \neq j$, then $\sqcup_i A_i \in \mathcal{L}(S)$.

(iv) The empty set is the only set in $\mathcal{L}(S)$ that is weakly disjoint with itself, i.e., if $A \sqcap A = \emptyset$, then $A = \emptyset$.

Where *Giles union* $A \sqcup B$ and *Giles intersection* $A \sqcap B$ are pointwisely defined as follows:

$$(A \sqcup B)(x) = \min\,[A(x) + B(x), 1] \tag{2}$$

$$(A \sqcap B)(x) = \max\,[A(x) + B(x) - 1, 0] \tag{3}$$

Since fuzzy sets remain in the same relation to an infinite-valued logic as traditional sets to the classical two-valued logic, all operations on fuzzy sets encountered in the above-quoted theorem can be further expressed in the language of the infinite-valued Łukasiewicz logic, which was also done in Pykacz (1994). Therefore, any orthomodular poset with an ordering set of probability measures (only such orthomodular posets are physically interesting), i.e., any "algebraic" QL, gives rise to a "genuine logical" QL, i.e., to a system of infinite-valued propositional functions defined on a set of states of a quantum sytem with negation, disjunction, and conjunction defined as logical connectives the truth-values of which are calculated by the formulas (1), (2), and (3), respectively.

One of the most interesting consequences of the above-described construction is the hypothesis that disjunction and conjunction of quantum mechanical propositions should not be represented by order-theoretic operations of join (the least upper bound) and meet (the greatest lower bound), as it was originally suggested by Birkhoff and von Neumann (1936), but rather by Giles union (2) and intersection (3) in fuzzy set models of quantum logics or by infinite-valued logical counterparts of these operations if we pass to the infinite-valued Łukasiewicz logic. Actually, the most recent results (Pykacz, n.d.) show that in fuzzy set models of quantum logics Giles operations coincide with order-theoretic ones whenever they are both defined, which explains why the Birkhoff and von Neumann hypothesis was quite reasonable in 1936, many years before the birth of the fuzzy set theory, and why it persisted so many years in spite of various interpretational difficulties, like the existence of joins and meets of incompatible elements in lattice models of quantum logics. Giles operations, contrary to order-theoretic ones, are pointwisely defined: the knowledge of $A(x)$ and $B(x)$ suffices to calculate, via formulas (2) and (3), the values $(A \sqcup B)(x)$ and $(A \sqcap B)(x)$ without knowing $A(y)$ and $B(y)$ for all other $y \in S$, which is necessary in the case of "globally defined" joins and meets. This feature of Giles operations also enabled Pykacz and Zapatrin (1997) to use them in computations based on the laws of a QL.

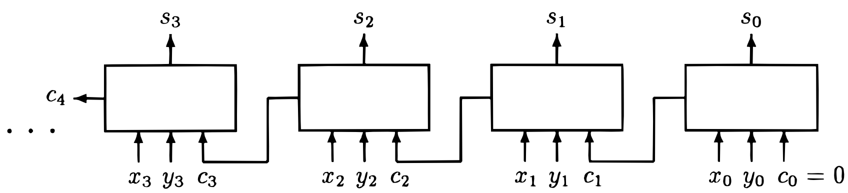

Fig. 1. Classical adder.

3. AN EXAMPLE OF QL-BASED CALCULATIONS

I shall quote now from Pykacz and Zapatrin (1997) an example of a logical network able to perform calculations according to the laws of a QL instead of a two-element Boolean algebra. The procedure consists in taking a simple network consisting of Boolean logic gates ("classical adder") and then replacing Boolean operations by their QL counterparts.

3.1. Classical Adder

One of the simplest mathematical models of a classical adder, often described in standard textbooks (see, e.g., Harrison, 1995), consists of the sequence of Boolean logic gates represented in Fig. 1. Each of these gates is endowed with three inputs x_i, y_i, and c_i, and two outputs s_i and c_{i+1}, and works according to the truth table in Table I, where x_i and y_i are ith binary digits of two numbers that we add, s_i is the ith binary digit of the sum (sum bit), $c_0 = 0$, and the carry bit c_{i+1} emerging as the output from the ith gate enters the next gate together with x_{i+1} and y_{i+1} as one of its inputs.

Instead of describing Boolean logic gates by their truth tables, one can do it with the aid of suitable Boolean functions. In the case of a classical adder these functions can be of the following form (see, e.g., Harrison, 1995):

$$s_i = \bar{x}_i\,\bar{y}_i c_i + \bar{x}_i\, y_i\bar{c}_i + x_i\,\bar{y}_i\bar{c}_i + x_i y_i c_i \tag{4}$$

$$c_{i+1} = \bar{x}_i\, y_i c_i + x_i\bar{y}_i\, c_i + x_i y_i\bar{c}_i + x_i y_i c_i \tag{5}$$

Table I

x_i	y_i	c_i	s_i	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

where $\overline{0} = 1$, $\overline{1} = 0$, $0 + 0 = 0$, and $0 + 1 = 1 + 0 = 1 + 1 = 1$, i.e., operations $^{-}$, $\cdot$, and $+$ represent, respectively, complementation, meet, and join in the two-element Boolean algebra or, equivalently, negation, conjunction, and disjunction in the classical two-valued logic.

It should be noticed that the formulas (4) and (5) are not the only possible forms of Boolean functions compatible with the given truth table. The other, equivalent, forms can be obtained from (4) and (5) with the aid of De Morgan and distributivity laws valid in any Boolean algebra and in the classical two-valued logic. For example, the other form of a Boolean function which yields proper values for s_i is

$$s_i = (\bar{x}_i + \bar{y}_i + c_i)(\bar{x}_i + y_i + \bar{c}_i)(x_i + \bar{y}_i + \bar{c}_i)(x_i + y_i + c_i) \qquad (6)$$

3.2. QL-Based Adder

In Pykacz and Zapatrin (1997) we passed from the classical adder to a QL-based adder by replacing classical operations of the two-valued logic or the two-element Boolean algebra by their QL counterparts while leaving both the types of logic gates and the topology of the network unchanged. We based our construction on the results of Pykacz (1994) briefly mentioned in Section 2 and on the hypothesis that QL conjunction and disjunction should not be modeled by meets and joins, but by Giles intersections (3) and unions (2), or by respective logical connectives of the infinite-valued Łukasiewicz logic.

However, it is a typical feature of a many-valued logic, noticed already by its founding father Jan Łukasiewicz, that the number of tautologies of a logic decreases with the increasing number of truth-values. Therefore, many logical formulas which are equivalent in the two-valued logic cease to be equivalent in a QL, which was reinterpreted in Pykacz (1994) as a kind of a partial infinite-valued logic. Consequently, much care is required in choosing the proper form of a classical Boolean function as a starting point for our procedure of changing a classical adder into a QL-based adder.

The proper QL expressions for s_i and c_{i+1} are

$$s_i = (x_i' \sqcup y_i' \sqcup c_i) \sqcap (x_i' \sqcup y_i \sqcup c_i') \sqcap (x_i \sqcup y_i' \sqcup c_i') \sqcap (x_i \sqcup y_i \sqcup c_i) \quad (7)$$

$$c_{i+1} = (x_i' \sqcap y_i \sqcap c_i) \sqcup (x_i \sqcap y_i' \sqcap c_i) \sqcup (x_i \sqcap y_i \sqcap c_i') \sqcup (x_i \sqcap y_i \sqcap c_i) \quad (8)$$

where, according to the results of Pykacz (1994) quoted in Section 2, $a' = 1 - a$, $a \sqcup b = \min(a + b, 1)$, $a \sqcap b = \max(a + b - 1, 0)$, and values of the arguments range over the whole interval [0, 1].

It can be easily seen that although formula (7) yields the same numerical results as both (4) and (6) when $x_i, y_i, c_i \in \{0, 1\}$, it is the formal QL counterpart of formula (6), not (4). Similarly, (8) is a QL version of (5).

It should be stressed that the described QL-based adder differs in its working principle from the quantum adders studied by Peres (1985) and Vedral *et al.* (1996, 1998). The difference follows from the fact that the logic gates of which the latter adders are composed perform classical Boolean operations [cf. the truth-tables in Fig. 3 of Peres (1985) and Fig. 1 of Vedral *et al.* (1996)]. These logic gates are reversible, by which these quantum adders differ from the classical adder described in the previous subsection, since reversibility is a necessary requirement for logic gates used in QC. Nevertheless, their underlying logic is still the classical two-valued logic since, as noticed by Peres (1985), in these quantum adders, "In each elementary logical step, no generic quantum property (interference, nonseparability, indeterminism) can be detected." This means that these adders are "quantum" only with respect to their physical working principles: if one prepares the input data in the form of a superposition of qubits representing various numbers, the adders will work in a "parallel" way acting simultaneously but independently on all elements of the superposition. However, the logical operations which form the mathematical basis of their computations are still classical operations on the two-element Boolean algebra $\{0, 1\}$.

Contrary to these quantum adders, logic gates of our QL-based adder perform genuine QL operations, so the adder is based on the "genuine" QL. Let us note that allowed values of all numbers that appear in the formulas (7) and (8) belong to the whole interval [0, 1]. According to the orthodox interpretation of a QL, these numbers represent probabilities that propositions about quantum systems (elements of a QL) turn out to be true when suitable dichotomic ("yes–no") experiments are done. Therefore, according to the orthodox interpretation, a QL is still a two-valued, although nondistributive logic. In Pykacz (1994) a QL was reinterpreted as a particular kind of infinite-valued (and still nondistributive) Łukasiewicz logic and within this interpretation the above-mentioned numbers represent nonclassical truth-values of propositions. However, the orthodox "probabilistic" interpretation of the numbers that appear in the formulas (7) and (8) is still very useful and in fact provides a basis for a new way of encoding numbers described in the next section.

4. PROBABILISTIC BINARY AND GRAY CODES

Utilizing a many-valued logic in the process of computation makes the usual binary way of encoding numbers in the form of sequences of 0's and 1's problematic, if not useless. On the other hand, although any experimental proposition about a quantum system (i.e., an element of a QL) is, according to the adopted many-valued interpretation of a QL, represented by a many-valued propositional function *before* the suitable experiment is done, it "collapses" into the classical, two-valued proposition *after* completing of the experiment. For example, the sentence, “A photon will pass through a polarizer,” is a many-valued propositional function defined on a set of all possible states of the photon, whose truth-value in any fixed state equals the probability that the photon in this state will actually pass through the polarizer. However, when the experiment consisting in casting photons at the polarizer is completed, the statement, “A photon passed through a polarizer,” is either true or false, i.e., it belongs to the realm of the classical two-valued logic and we can denote its truth-value either by 0 or by 1.

This observation led to the new way of encoding numbers proposed in Pykacz and Zapatrin (1997) and called there the *probabilistic binary code*. According to this, a number is encoded in the form of a sequence $\{p_i\}$ of real numbers $p_i \in [0, 1]$ and the number p_i placed at the ith position in this probabilistic binary expansion represents the probability of getting 1 at this place when a suitable measurement is done. This way of encoding numbers enables us to encode not only the value of a number, but also a “degree of certainty,” “vagueness,” or “fuzziness” of this value. Therefore, it should be particularly well suited to deal with unexact data and could be possibly useful in calculations performed with the aid of quantum physical systems. For example, the number 3, whose binary expansion is $11 = \ldots 011$, when represented by the triple (0.1, 0.9, 0.8), would actually appear with the probability $(1 - 0.1) \times 0.9 \times 0.8 = 0.648$, while represented by the triple (0.2, 0.8, 0.7), it would actually appear with the probability $(1 - 0.2) \times 0.8 \times 0.7 = 0.448$, i.e., the “degree of certainty” that the second triple actually represents the number 3 is, as expected, lower than that of the first triple. Of course a sequence

$$\overbrace{(1/2, 1/2, \ldots, 1/2)}^{n \text{ terms}}$$

represents total lack of knowledge, i.e., such a sequence represents any integer between 0 and 2^n with equal probability $1/2^n$.

With the proposed convention in mind, we can make a “classical simulation” of the work of the QL-based adder described in the previous section, calculating step by step the probabilities of getting 1 as values of consecutive sum bits, and carry bits with the aid of formulas (7) and (8).

Such calculations performed, for example, on a number “nearly 2” represented in a probabilistic binary code by a triple $X = (0.1, 0.9, 0.1)$ (instead of 010) and a number “almost 3” represented by a triple $Y = (0.1, 0.8, 0.8)$ (instead of 011) give as a final result a triple $X + Y = (1.0, 0.3, 0.9)$, which could be expressed, e.g., as “close to 5” since the binary expansion of 5 is 101.

As already stated, this result should be interpreted probabilistically, so we expect that if the adder were made to run 1000 times, one would expect to obtain 1000 triples $Z = (z_2, z_1, z_0)$ in such a way that $z_2 = 1$ in all of them, $z_1 = 1$ in 300 triples (i.e., $z_1 = 0$ in 700 triples) and $z_0 = 1$ in 900 triples, i.e., the number $5 = 101$ should be obtained with probability $(1 - 0.3) \times 0.9 = 0.63$, while obtaining the other combinations of 0's and 1's which represent other integers would be much less probable.

Encoding numbers with the aid of the probabilistic binary code is plagued by a serious disadvantage which I shall explain by an example. Let us assume that a number $3 = \ldots 011$ is represented, as before, by a triple (0.1, 0.9, 0.8), so the probabilty of getting the desired result is relatively high: 0.648. Probabilities of getting wrong results are the following: $p_{0=000} = 0.018$, $p_{1=001} = 0.072$, $p_{2=010} = 0.162$, $p_{4=100} = 0.002$, $p_{5=101} = 0.008$, $p_{6=110} = 0.018$, and $p_{7=111} = 0.072$. The values of these probabilities are rather counterintuitive since one would rather prefer the probabilities of getting numbers close to 3 to be higher than probabilities of getting numbers that differ from 3 in a more significant way. Especially counterintuitive is the extremely low probability of getting the number $4 = 100$, which stands in the closest vicinity of 3. Unfortunately, the described phenomenon cannot be avoided: note that when we pass from $3 = 011$ to $4 = 100$, all three bits in the binary expansion are changed. Therefore, the maximal probability of getting the sequence 011, obtained as a product of probabilities of getting single bits, unavoidably leads to the minimal probability of getting the sequence 100.

Fortunately, there does exist a system of encoding numbers by sequences of 0's and 1's in which the described disadvantage is greatly reduced. This system of encoding numbers is called the Gray code and it has the property that only one bit changes during a passage from one integer to the next. A simple rule for encoding natural numbers in this way is as follows: Represent 0 by a sequence consisting only of zeros. To get to the next number, always change the least significant bit that yields a new number, i.e., a number that did not appear before. Table II shows some first natural numbers represented by straight binary and Gray codes, the number of bits that change in the straight binary representation while passing from an integer to the next one, and, as an example, already calculated probabilities of getting both desired and wrong results when the number 3 is represented by the triple (0.1, 0.9, 0.8) in the probabilistic binary code, and respective probabilities when the number 3 is represented by the triple (0.1, 0.9, 0.2) instead of 010 in the probabilistic version of the Gray code.

We see that although the probability distribution of getting wrong numbers from the probabilistic Gray expansion still has some anomalies (slight increase at the very beginning and at the very end of the distribution), it is

Table II

Number	Straight binary code	Number of bits that change	Gray code	Probability of getting 3 from the triple (0.1, 0.9, 0.8) in straight binary	Probability of getting 3 from the triple (0.1, 0.9, 0.2) in Gray code
0	000	1	000	0.018	0.072
1	001	2	001	0.072	0.018
2	010	1	011	0.162	0.162
3	011	3	010	0.648	0.648
4	100	1	110	0.002	0.072
5	101	2	111	0.008	0.018
6	110	1	101	0.018	0.002
7	111	4	100	0.072	0.008

nevertheless much better than in the probabilistic straight binary expansion originally proposed in Pykacz and Zapatrin (1997). Since there do exist simple networks of exclusive-OR gates that convert straight binary to Gray and Gray to straight binary codes, encoding numbers with the use of the probabilistic Gray code should not lead to any computational difficulties.

5. FINAL REMARKS ON THE CONTROVERSIAL TERMINOLOGY

A QL theorist would probably be very disappointed upon looking through QC papers that bear the words "quantum logic" in their title, like that of Turchette *et al.* (1995), and not seeing any familiar orthomodular structures there. Even if the term "quantum logic" does not explicitly appear in a QC paper, nearly all of them contain the expression "quantum logic gate," which is intuitively understood by a QL theorist as a logic gate that works according to the laws of a QL. On the other hand, a QC theorist may be equally disappointed looking through any of the "genuine" QL papers that appear each year in a number comparable to QC papers.

Such misunderstandings are unavoidable as long as different groups of people attach the same name to different objects. In the discussed case the situation is even worse since the object "quantum logic" is not uniquely defined by QL nor by QC theorists.

What could be done to judge whether quantum logic in the QL sense or in the QC sense is more legitimately called "quantum logic"? In my opinion first of all the logical background of QC should be carefully studied. The emerging logic will surely occur to be nonclassical, first, because of the extensive use of superpositions with complex coefficients, which also shows that this logic cannot be a version of an "orthodox" QL since in the latter, complex numbers do not appear at all.[2] Only when this object is extracted from technicalities of QC papers will it be possible to check to what extent it conforms with anything that bears the name "logic" in mathematical sciences.

On the other hand, the term "quantum logic" most frequently understood in the QL theory as a synonym of the term "orthomodular lattice" or "orthomodular poset" may also yield serious misunderstandings since, in a sense, it is neither "quantum" nor "logic": It is not "quantum" in a sense that it is not *exclusively* quantum, since Boolean algebras, which are algebraic structures characteristic to classical physics, form a proper subfamily of the family of orthomodular posets, so this term denotes also structures characteristic to classical theories. It is not "logic" since it is an algebraic structure which could be at most thought of as a Lindenbaum–Tarski algebra of a hypothetical *quantum logic proper*, whatever it might be.

Such considerations suggest that the term "orthomodular algebra" coined by Burmeister and Mączyński (1994) might be the most proper synonym for the unfortunate term "quantum logic" used in the algebraic sense, especially in that it naturally fits into the sequence of other "algebras" (Boolean algebras, orthoalgebras, effect algebras, . . .) that are studied within the theory of quantum structures. Maybe members of the International Quantum Structures Association should discuss such a change in the QL terminology during the next Biannual IQSA Meeting in the year 2000?

ACKNOWLEDGMENTS

This work was supported by the University of Gdańsk Research Grant BW/5100-5-0190-8 and by the Polish-Flemish Research Project 007/R97/R98.

REFERENCES

Aharonov, D. (1998). Quantum computations, in *Annual Reviews of Computational Physics*, Vol. VI, D. Stauffer, ed., World Scientific, Singapore; also quant-ph/9812037.

Beltrametti, E.G., and Cassinelli, G. (1981). *The Logic of Quantum Mechanics*, Addison-Wesley, Reading, Massachusetts.

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

[2] There is an exception: Von Weizsäcker's almost forgotten "complementarity logic" [see pp. 376–379 of Jammer (1974) for brief review and references], which should be surely studied anew in the QC context.

Burmeister, P., and Mączyński, M. (1994). Orthomodular (partial) algebras and their representations, *Demonstratio Mathematica*, **27**, 701–722.

Harrison, M. A. (1995). *Introduction to Switching and Automata Theory*, McGraw-Hill, New York.

Jammer, M. (1974). *The Philosophy of Quantum Mechanics*, Wiley, New York.

Pavičić, M. (1992). Bibliography on quantum logics and related structures, *International Journal of Theoretical Physics*, **31**, 373–461.

Peres, A. (1985). Reversible logic and quantum computers, *Physical Review A*, **32**, 3266–3276.

Pták, P., and Pulmannová, S. (1991). *Orthomodular Structures as Quantum Logics*, Kluwer, Dordrecht.

Pykacz, J. (1994). Fuzzy quantum logics and infinite-valued Łukasiewicz logic, *International Journal of Theoretical Physics*, **33**, 1403–1432.

Pykacz, J. (n.d.). In preparation.

Pykacz, J., and Zapatrin, R. R. (1997). Non-classical logics for quantum computations, in *Photonic Quantum Computing, Proceedings of SPIE*, S. P. Hotaling and A. R.Pirich, eds., International Society for Optical Engineering, pp. 150–161.

Rieffel, E., and Polak, W. (1998). An introduction to quantum computing for non-physicists; quant-ph/9809016.

Turchette, Q. A., Hood, C. J., Lange, W., Mabuchi, H., and Kimble, H. J. (1995). Measurement of conditional phase shifts for quantum logic, *Physical Review Letters*, **75**, 4710–4713.

Vedral, V., Barenco, A., and Ekert, A. (1996). Quantum networks for elementary arithmetic operations, *Physical Review A*, **54**, 147–153.

Vedral, V., and Plenio, M. B. (1998). Basics of quantum computation, *Prog. Quant. Electron.* **22**, 1–40; also quant-ph/9802065.