International Journal of Theoretical Physics, Vol. 22, No. 8, 1983

Computational Complementarity

David Finkelstein and Shlomit Ritz Finkelstein

School of Physics, Georgia Institute of Technology, Atlanta, Georgia 30332

Received May 29, 1982

Interactivity generates paradox in that the interactive control by one system C of predicates about another system-under-study S may falsify these predicates. We formulate an "interactive logic" to resolve this paradox of interactivity. Our construction generalizes one, the Galois connection, used by Von Neumann for the similar quantum paradox. We apply the construction to a *transition system*, a concept that includes general systems, automata, and quantum systems. In some (classical) automata S, the interactive predicates about S show quantumlike complementarity arising from interactivity: The interactive paradox generates the quantum paradox. Some classical S's have noncommutative algebras of interactively observable coordinates similar to the Heisenberg algebra of a quantum system. Such S's are "hidden variable" models of quantum theory not covered by the hidden variable studies of Von Neumann, Bohm, Bell, or Kochen and Specker. It is conceivable that some quantum effects in Nature arise from interactivity.

1. INTRODUCTION

1. The famous incompletenesses associated with the names of Einstein and Gödel are both limitations on our ability to predict activity of a quantum and an automaton, respectively. Both involve self-referential activity importantly: Gödel's theorem derives from the paradox of "This statement is false"; according to Bohr (and see Wheeler, 1982) a quantum phenomenon includes its own observation. It is natural to ask, therefore, the following:

Question I: Are these incompletenesses aspects of one underlying principle?

This includes the question: Can quantum paradoxes be paradoxes of self-reference? This possibility has also been considered by Brown (1969) and Kaufmann and Varela (private communications), and Zwick (1978).

The Gödel incompleteness deals with theorem-proving automata, while Einstein's has to do with quantum systems. To unite these we must find a common language for these two kinds of system. Question I thus leads to the following:

Question II: What system concept appropriately encompasses both quanta and automata? (See Section 3.)

2. Let S be the system under study and C the control system (possibly including ourselves) throughout this paper. "Control" includes preparation and measurement; creation, modification, and destruction. In order to recognize when quantum effects occur in S, we seek the effective *interactive predicate algebra* of S relative to C. When this algebra is nondistributive (or, in another widespread formulation, partial; see Kochen and Specker 1967), the whole system CS exhibits quantum complementarity. Thus Question I also leads to the following:

Question III: What is the interactive predicate algebra of S? (See Section 4.)

3. Question III confronts us with the paradox of interactivity: *Control falsifies*. In general, the interactive control by C of a predicate about S will falsify that predicate. Question III thus includes the following question: Can the interactive paradox generate quantum paradoxes?

When a classical S has an interactive predicate algebra of the quantum kind, CS provides a classical model of quantum theory; in the terminology of Von Neumann (1932), a "hidden variables" theory of the quantum paradoxes. Thus the answer to Question III will also tell us if hidden variables can account for quantum indeterminacy and complementarity.

4. Question III is a sharpening and a generalization of questions that have long been asked about computers and other automata. The control of computer properties (i.e., predicates) through data channels alone is a matter of great practical importance, and is studied by Moore (1956) and many others. It is a natural extension of that work to ask for the predicate algebra of the properties that are thus controllable.

5. Interactive logic, unlike classical, deals with time. When we control (the truth of) predicates about a system, the time *order* of the control operations effects the results of our experiments. Let us call such a logic *noncommutative*. The commutative Boolean logic is a degenerate idealized limit of noncommutative logic. Noncommutative logic first appears in the quantum theory of Von Neumann (1932). That noncommutative logic applies to macroscopic situations (as well as to quantum situations, which are usually microscopic) is often asserted. Bohr's suggestions of complementarity in psychology may be considered instances. Resemblances between automata and quantum systems have also been pointed out long ago. Moore (1956, p. 138) speaks of "an analogue of the uncertainty

principle," illustrating it with a specific four-state Moore machine (paragraph 20) termed by Conway (1971) "Moore's Uncertainty Principle." Here we compare the interactive predicate algebras of quanta and automata. Our construction generalizes one used by Von Neumann for quantum systems, the Galois connection (paragraph 28). It suggests a formulation and solution of the "hidden variables" problem of Von Neumann (Section 5). This raises our final question:

Question IV: Do automata, like quanta, have noncommutative Heisenberg algebras of coordinates? (See Section 6.)

2. BASIC ASSUMPTIONS

6. Question I brings in the limited complexity of *CS* (Chaitin 1966, 1982). In this first approach we put aside Question I and deal naively with Questions II, III, and IV, assuming that there are infinite time, space, and materiel for any experiments and calculations we wish to do and record.

7. We make no assumptions about the determinism of the systems we study. The quantum effects we study here arise in both determinate and indeterminate systems. They arise not from an underlying indeterminacy but solely from interactivity, specifically the interference of interactive controls. Although we limit ourselves in the main to finite automata, infinity too is irrelevant to the quantum effects we study.

8. Von Neumann emphasized the importance of probabilistic logics. Indeed, the probability of a transition is more informative than the possibility of a transition, which we study. Yet a probability estimate for some event to occur in one system, if it is to be useful, must be approximately equivalent to a certain highly special yes-or-no predicate about a suitable set or ensemble of similar systems; a predicate about frequency of occurrence in the set. Thus probability theory is the poor man's set theory. Since there are other important reasons for developing set theory, we put aside probability estimates until after that development. In any case, a knowledge of the yes-or-no possibilities and necessities together determine the probabilities of quantum theory, in accord with the quantum law of large numbers (Finkelstein, 1963).

9. A theory of one class of phenomena may belong to two domains (among others) that we must distinguish for clarity. For theories in *domain S* (as we shall call it), the controller *C* does not enter explicitly into the equations but only indirectly if at all, in principles of relativity and coordinate transformations. In *domain S*, measurements on *S* give us coordinates of *S*. In *domain S* we are unselfconscious, extroverted, and pragmatic.

In *domain CS*, *C* enters explicitly into the equations of the theory together with *S*. Indeed, it may be a problem to separate *S* and *C* in *domain CS*. In *domain CS*, measurements (the same ones mentioned in connection with *domain S*) are expressed in the theory as interactions between *C* and *S*, during which the coordinates of *S* may change. At the limit of domain *CS* is the universal theory, where *CS* is the whole universe. In domain *CS* we are self-conscious, introspective, and grandiose.

It is important to state the domain explicitly even when the same equations can be interpreted in either domain. The enthusiastic physicist may decide that his beloved equations, Einstein's, Schroedinger's, or whatever, can be used in either *domain S*, as they were originally designed, or *CS*. He then has two distinct theories with one set of symbols and equations. The difference is entirely in the semantics of his two theories, not in the syntax: The same experience leads him to utter syntactically different expressions in the same syntactic system, depending on the domain.

In the domain of microphysics our experiences are strongly shaped by the natures of both *C* and *S*. To move a theory from domain *CS* to *S* we randomize and average over *C*, likely with great change in syntax. We expect then that the *CS* theory is virtually unrecognizable from the *S* theory. (This possibility is denied by the kind of operationalism which requires even a universal theory to speak in terms of operational observables.) Then it is even more important to state the domain of the theory.

In this work we enter domain *CS*. To be sure, the states that we assign to some computer *S* are in principle accessible to our direct control and unchanged by that control, since we may remove the cabinet and bypass the data channels with electronic voltmeters; but for pragmatic reasons we deliberately refrain from this direct control and limit ourselves to control through data channels which pass through a console that serves as a surrogate *C*.

10. Let us call the logic we use for predicates about the whole system *CS the CS logic*. In the present work we restrict ourselves to classical automata and the *CS* logic is Boolean. If we wish to deal with interactive paradoxes in quantum automata (and any actual machine is a quantum system when we look at it hard enough) we would use a non-Boolean or quantum *CS* logic, describing the automaton by wave functions and operators in the familiar quantum language. There may still be a difference in structure between the *CS* logic and the interactive logic, the logical analog of the dynamical difference between bare and renormalized mass. The methods used in this study also work for the more general case of non-Boolean *CS* logic.

11. The interactive logic we study is that of *S* relative to *C*, and we hold both *S* and *C* fixed. A full relativity requires us to treat *S* and *C*

symmetrically, to consider a plurality of them, and to formulate what makes a system a control system. We leave these fascinating questions for later.

3. TRANSITION SYSTEMS

12. *Notation and terminology about spaces and systems.* A *space* is a class of possibilities. If S is a space then S', S'', and $'S$ are variables with S as domain, and S_1, S_2, $_1S$ and $_2S$ are constant members of S.

We mix Russell–Whitehead dot punctuation with parentheses as convenient. The signs "$:=$" and "$=:$" indicate a definition and divide the expression defined from the defining. The colon abuts the defined.

We use the following arrow notation for exponentials:

$S \to T := $ (mappings: $S \to T$) or the exponential with radix T and exponent S. Thus $2 \to 3 = 9$.

Definition. CS is a *transition system with state space* $S := C$ is a subclass of the power $SS \to 2$. That is, each C' is a class of ordered dyads $S', 'S$.

Definition. $S' : C' : 'S :=$ the ordered dyad $S', 'S$ belongs to the class C'. This is the *control relation*.

13. *Interpretation.* The interpretation of the control relation is defined in examples to follow. Briefly, the control relation $S' : C' : 'S$ means that the sequence $S', C', 'S$ of the three events S' ($:= S$ being in the state S'), $C'(:=$ the input–output operation C'), $'S(:= S$ being in the state $'S$) is *possible*.

Definition. $S' \cdot C' = 'S := 'S$ is the one S'' such that $S' : C' : S''$ holds. If for all S' there is some $'S$ such that $S' \cdot C' = 'S$, we call C' a function, and if every control is a function we call the system *functional*.

Even if the control relation is functional, the system need not be deterministic: it may stop instead of making the one possible transition. In the work to come functional controls may be deleted. More important are *subfunctional* controls ($:=$ partial functions$:=$ functions from subclasses of S to S). We call S' the initial state, $'S$ the final state, of the transition $S' \cdot 'S$.

14. *Boolean vector logic.* A *Boolean vector* (or matrix) is one whose elements are 0 (i.e., "false") and 1 (i.e., "true"). Each subclass s of a space S may be regarded as a Boolean vector by taking the points of S in a fixed order $S_1, S_2, \ldots$, taking S' as vector index, and taking as S' component of the vector s the Boolean ($:= 0$ or 1) quantity $s \cdot S' := 1$ for S' in s, $= 0$ otherwise. In particular S itself is a row of 1's only, and the null class 0 is a row of 0's only.

Likewise we regard a control C_1 as a Boolean matrix with matrix element $S' : C_1 : 'S := 1$ for dyad S', $'S$ in C_1, $:= 0$ otherwise. We write SS for the matrix of 1's only, and **0** for the matrix of 0's. We call such a Boolean matrix a *control matrix* of the system. The system is specified by the class consisting of its control matrices.

Definition. Let C_1 and C_2 be relations and the matrices representing them. The *Boolean matrix product* C_1C_2 is the matrix representing the relational product $C_1 \cdot C_2$. It is computed by multiplying the same elements one multiplies to compute the usual matrix product, but then combining these products by a Boolean join ($\vee$) rather than addition: $0 \vee 0 = 0$, $0 \vee 1 = 1 \vee 0 = 1 \vee 1 = 1$.

15. *Graph of CS. Definition.* The *(labeled) graph* of a system CS is C regarded as a set of sets of ordered dyads $S' \cdot 'S$. It is also the graphical representation of C by labeled arcs. A labeled arc

$$S' \overset{C'}{\rightarrowtail} 'S$$

is in the graph if and only if the control relation $S' \colon C' \colon 'S$ holds.

12 bis. *Definition.* $S' \cdot C' = 0 \colon =$ there is no $'S$ such that $S' \colon C' \colon 'S$ holds, and we say "$S' \cdot C'$ stops the system." If $S \cdot C_1 = 0$, we say "C_1 stops the system." In Boolean vector logic, the row representing the state S' is annihilated by the control matrix C'.

Definition. The final state class $S \cdot C'$ of control $C' \colon =$ the class of all $'S$ such that for some S', $S' \colon C' \colon 'S$ holds. In Boolean vector logic, the Boolean matrix product of S with C'.

16. *Example: The system S(n).* The state space S has n points and will be identified with the cardinal $n = (0, \ldots, n-1)$. The controls in C are the subfunctions in $S \rightarrow S$. So the control matrices are those with exactly one 1 on each row.

This system may be pictured as a die with n faces, one of which is always "up." This system is functional.

17. *Example: The identity controller.* This system has N states. For each state there is a control, an identity subfunction, testing for that state. We may take S, the space of states, to be N, the cardinal number with N members $0, \ldots, N-1$. The control space is

$$C \colon = (S_1 S_1) \cdot \vee \cdot (S_2 S_2) \cdot \vee \cdot \cdots$$

the class of all singlet subclasses of the diagonal of the Cartesian product SS. That is,

$$S' \colon C' \colon 'S \colon = . \; S' = 'S \text{ and } C' = (S'S')$$

Each control matrix is a projector on an axis of the S' frame and has one 1 on its diagonal.

18. *Example: The counter.* The space is again N. There is exactly one control C_1 in the space C. $S' \cdot C_1 \cdot {}'S$ if and only if ${}'S = S' + 1$. Thus C_1 "sequences" the state through the values 0, 1,…,$N-1$, and stops on $N-1$. The graph of this machine (in the sense of paragraph 15) is

$$S_0 \overset{C_1}{\rightarrowtail} S_1, \overset{C_1}{\rightarrowtail} \cdots \overset{C_1}{\rightarrowtail} S_N$$

The control matrix has 1's only in the diagonal above the principle diagonal.

19. *Example: The Mealy–Moore machine.* The graph of a Mealy machine, defined by Hopcroft and Ullmann 1979, is already the graph of a transition system. In the graph of a Mealy automaton, each arc bears two labels, one input and one output. From our point of view this distinction is not important. Both input and output are interactive control operations. We regard the pair of labels as a single label, and the space of such pairs is the control space C of our system.

The Moore machine is the special case of a Mealy machine where the output of a transition depends only on the final state of the transition, and its representation by a transition system therefore follows.

These machines have specified initial state in their original description. We prefer to leave the initial state indefinite. The special considerations required to impose a special initial state are simple and omitted.

20. *Example. The Moore Uncertainty Automaton* (expressed as a transition system instead of a Moore machine). The graph of this machine is Figure 1.

The relation $S{:}C{:}S$ holds in exactly the following cases:

$$\begin{array}{ll}
S_1{:}a'{:}S_4, & S_1{:}b{:}S_3 \\
S_2{:}a'{:}S_4, & S_2{:}b'{:}S_4 \\
S_3{:}a{:}S_2, & S_3{:}b'{:}S_4 \\
S_4{:}a'{:}S_4, & S_4{:}b{:}S_1
\end{array}$$

With $C_1 = a, C_2 = b, C_3 = a', C_4 = b'$, the control matrices are

$$\begin{array}{llll}
C_1 = 0000 & C_2 = 0010 & C_3 = 0001 & C_4 = 0000 \\
0000 & 0000 & 0001 & 0001 \\
0100 & 0000 & 0000 & 0001 \\
0000 & 1000 & 0001 & 0000
\end{array}$$

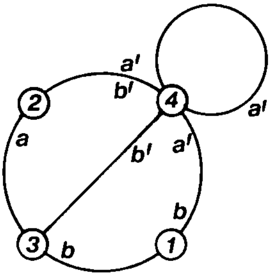

Fig. 1. Graph of the Moore uncertainty automaton. The vertices are the states S_1, S_2, S_3, S_4. The arcs are labeled a, b, a', b'. The label neighbors the final state of the arc, thus directing the arc. The transitions labeled a', b' all lead to S_4. Every transition to S_4 results in output 0, while all others result in output 1.

From the graph we see that the controls a, a' distinguish S_2 from S_3 but confuse S_2 with S_1, and the controls b, b' distinguish S_2 from S_1 but confuse S_2 with S_3. Thus one distinction is complementary to the other.

21. *Control sequences.* Given a sequence of control operations, a system may act sequentially unless it reaches a state and a control for which no transition is possible, when it stops. In sequential operation, the final state of one event becomes the initial state of the next event. This leads us to formulate a more general kind of control operation called a control sequence, including simple controls as special cases. A *control sequence* is the relational product of a sequence of control operations (considered as relations between states). The space Q of a control sequence ("queue") is:

$$Q := 1 \vee C \vee CC \vee CCC \vee \cdots =: \mathrm{seq}(C)$$

The terms in this join represent sequences of length 0, 1, 2, 3,..., and $\vee$ represents the set-theoretic join.

We extend concepts defined for control operations to control sequences in a natural way. For example, $S'{:}Q'{:}'S$ is the relational product of the relations in the control sequence Q'.

In Boolean vector logic, a control sequence of length L is represented by the Boolean product of L control matrices. $Q' = 0$ means that the control sequence Q' stops the system. SQ' is the *final state class* of the control sequence Q', and $Q'S$ is the *initial state class* of Q'.

We multiply two control sequences relationally, making Q a semigroup whose identity is the null sequence, which we designate by Q_1. If necessary we adjoin the sequence-zero Q_0, consisting of the control-zero Q_0 only, with the property

$$Q'Q_0 = 0 = Q_0Q'$$

for all Q'. Thus Q_0 stops the system. In Boolean vector logic, Q_0 is represented by the matrix **0** and Q_1 is represented by the unit matrix.

22. *Definition.* The *possibility relation* $Q'{:}\,p{:}\,Q''$ between control sequences holds if and only if $Q'Q'' \neq 0$. The orthogonality relation $Q'{:}\,o{:}\,Q''$ is the negation of the possibility relation. $Q'{:}\,o{:}\,Q''$ means that the control sequence $Q'Q''$ stops the system whatever the initial state, and in Boolean vector logic that the Boolean product of the two sequential control matrices Q', Q'' is the matrix **0**. The value (true or false) of the possibility relation $Q'{:}\,p{:}\,Q''$ is that of the Boolean inner product $SQ' \cdot Q''S$.

Since the o relation is generally not symmetric, we say that Q' is initially orthogonal to Q'' or *excludes* Q'', and that Q'' is finally orthogonal to Q', or *is excluded* by Q', when this relation holds.

4. INTERACTIVE PREDICATE ALGEBRA OF A SYSTEM

23. *Lattice background.* A *lattice* is a partially ordered set L with dyadic *l.u.b.* (join) $L' \vee L''$ and *g.l.b.* (conjoin) $L'\&L''$ for all L' and L''. If they exist, the overall lower bound and upper bound of the partially ordered set are called $L0$, the lattice-zero, and $L1$, the lattice-one. We write

$$L' \leqslant L''$$

when L' is in the partial order relation to L''; and $L' < L''$ if further $L' \neq L''$; and $L' <' L''$ if still further $L' < L''' < L''$ holds for no L''' ($:= L''$ "covers" L'). The *multiplicity* *mult*(L') of lattice element L' is the greatest number of $<'$ signs that can be interpolated between $L0$ and L' in a *covering sequence*, one of the form

$$L0 <' \cdots <' L'$$

Lattice elements of multiplicity $1, 2, \ldots, m$ are called singlets, doublets, $\ldots, m$-tuplets. For other concepts of lattice theory see Birkhoff and Von Neumann (1936), Birkhoff (1948), and Holland (1970).

Lattice 2: = the (Boolean) lattice with two members $L0$, $L1$.

Lattice ($N \rightarrow 2$) (for any non-negative integer N): = the Boolean lattice of predicates of an object with N possibilities in its space: = the power set of the cardinal number N.

Lattice $1+N+1$: = the lattice with 1 zero, N singlets, and 1 one, for any non-negative integer N.

The smallest non-Boolean lattice is lattice $1+1+1$. The smallest non-Boolean ortholattice is lattice $1+4+1$. This is the simplest nonclassical quantum logic.

24. *Definition: The Kleene algebra.* The Kleene algebra (actually a lattice) of the system CS: =

$$K := (Q \to 2)$$

the lattice of subclasses of the control sequence space Q of paragraph 21. Members of K are predicates about control sequences, and some will be interactive predicates about S.

K inherits from Q a natural semigroup multiplication: If K' and K'' are subclasses of Q, the product $K'K''$: = the class of all sequential products $Q'Q''$ for Q' in K', Q'' in K''. The identity $K1$ of *this* semigroup is $(q1)$, the class whose only member is the null sequence. We designate the null class, the zero of this product, by $K0$.

25. *The central principle: To predicate is to exclude possibilities.* One standard problem for us now is: If the past Q' belongs to given K', what is the class $''K$ of all possible future $''Q$? Equivalently, and more conveniently, we ask for the complement $'K$, the class of all $'Q$ excluded by K'. We write $K' \perp$ for this class $'K$. Our central principle is that each such initial class K' determines an initial predicate of the system, and that two classes K' and K'' determine the same predicate if and only if they exclude the same future possibilities; that is, if and only if

$$K' \perp = K'' \perp$$

It is shown in the next paragraphs that this equality holds if and only if K' and K'' have the same "closure." These closures may therefore be identified $1-1$ with initial predicates of the system.

26. *Relation concepts.* (Birkhoff 1948.) Let $A{:}o{:}Z$ be any dyadic relation with "initial" domain A and "final" domain Z. In the quantum application A is the space of an initial determination, Z is that of a final determination, represented by a vector and a covector and o is orthogonality. If a is any subclass of A, then $a \perp$ will mean the class of all members of Z standing in the relation o to *every* member of a. Dually we define $\perp z$:

$$a \perp := (Z' \text{ in } Z\text{: for all } A' \text{ in } a\text{: } A'{:}o{:}Z')$$

$$\perp z := (A' \text{ in } A\text{: for all } Z' \text{ in } z\text{: } A'{:}o{:}Z')$$

We call $a \perp$ the *final o class* of a, and $\perp z$ the *initial o class* of z.

In Boolean vector logic, A and Z are (diagonal vectors of) two possibly distinct spaces, a and z are subclasses and vectors in these spaces, and o is a Boolean form in the product of the two spaces dual to A and Z, a Boolean matrix with two covariant indices if A and Z have contravariant indices. Let n represent complementation: $0n = 1$, $1n = 0$. If v is a Boolean vector then vn is its complement; if m is a Boolean matrix then nmn is its complement. Let $p = (non)$ be the complement of the matrix o. Then $a \perp$ is represented by $(ap)n$, and $\perp z$ by $(pz)n$.

There are the following tautologies, valid for any relation and classes:

(i) $\perp$ reverses inclusion.
If $a \leqslant b$ then $b \perp \leqslant a \perp$.
If $y \leqslant z$ then $\perp z \leqslant \perp y$.

(ii) $a \leqslant \perp (a \perp), z \leqslant (\perp z) \perp$.

Sometimes we omit parentheses when the association is clear, writing $\perp a \perp$ and $\perp z \perp$.

27. *The triple-$\perp$ identities.* We deduce the inclusion $a \perp \geqslant \perp a \perp \perp$ from (ii) by applying $\perp$ to both sides. We deduce the transposed inclusion $a \perp \leqslant \perp a \perp \perp$ by substituting $a \perp$ for z in (ii). Therefore we have the important identities valid for any relation o

$$\perp a \perp \perp = a \perp$$

$$\perp \perp z \perp = \perp z$$

(Birkhoff, 1948.) They imply that $m \rightarrow \perp m \perp = \perp \perp m \perp \perp$ is a closure operation, and we define the initial and final *closure* of subclasses a or z by

$$\mathrm{cl}(a) := \perp (a \perp)$$

$$(z)\mathrm{cl} := (\perp z) \perp$$

An initially (or finally) *closed* class is one equaling its initial (or final) closure.

28. *Definition. The initial (or final) lattice, L(o) (or (o)L),* := the class of initially (or finally) closed subclasses of the initial (or final) domain of o, partially ordered by inclusion. For each of these lattices we define two (dual) lattice operations "join" and "conjoin," and designate them by $\cup$ and $\cap$, for both initial and final lattices. They generalize the Boolean operations *or* and *and*, respectively. The join $a = a_1 \cup a_2$ of two initial (or final) lattice members is the closure of the set-theory join:

$$a_1 \cup \ a_2 := \mathrm{cl}(a_1 \vee a_2)$$

where $\vee$ is the set-theory join (union). The conjoin of two lattice members is their set-theory conjoin (intersection).

Definition: The (Galois) *connection* of the relation o: = the pair of lattice dual morphisms

$$z \to \perp z$$

$$a \to a \perp$$

The lattice property of these classes of classes does not depend on any postulated properties of the relation o, but holds identically for all o.

29. *Example: Boolean algebra.* If $A = Z$ is any class and $A'\!:o\!:Z' := (A' \neq Z')$, then $L(o)$ is the Boolean algebra $A \to 2$ of subclasses of A, the power set. Classical logic is founded on this example.

30. *Example: Projective geometry.* If A is a linear space and Z is the dual space, and $A'\!:o\!:Z' := A'Z' = 0$ (the vector A' is annulled by the covector Z'), then $L(o)$ is the *projective geometry* of A, whose elements are the "flats" of A, the subspaces of A of all dimensions. Quantum logic is founded on this example.

31. *Lemma.* If a and b are subclasses of A, then

$$a \perp = b \perp \cdot = \cdot \operatorname{cl}(a) = \operatorname{cl}(b)$$

The left-to-right implication is clear from the definition of cl. The converse implication follows from the triple-$\perp$ equation of 27. ■

This lemma is important in connection with the central principle (25). It permits us to identify the Galois lattice of o with the predicate lattice of the transition system with orthogonality relation o. There is a further justification of this principle similar to the correspondence principle of quantum theory: In cases where an effective predicate algebra is already familiar, this principle agrees with established usage. These cases are the classical ones, like $S(n)$, and the quantum ones, both discussed below.

32. *Definition.* The *interactive predicate lattices* of a system are the initial and final lattices of the orthogonality relation of the system. (See paragraph 28.) We call members of these lattices initial and final predicates.

Therefore if our knowledge about the past is expressed by a class K' in the Kleene lattice of the system, the predicate expressing this knowledge is the initial closure $\operatorname{cl}(K')$.

33. *Product.* Besides the lattice join and conjoin, there is a natural time-ordered *product* of predicates in interactive logic. If K' and K'' represent interactive predicates, then we represent the product of these predicates by $\operatorname{cl}(K'K'')$, the closure of the semigroup product for K (paragraph 24). This product coincides with the conjoin in Boolean logic, but in general is not commutative.

34. *Example.* The systems $S(n)$, the identity controller, and the counter of paragraphs 16 and following have for their initial lattice the Boolean lattice $(n \to 2)$.

These cases illustrate how the lattice analysis sees through the console to the interactive state. The identity controller has n keys on its console, one for each state, while the counting system has only one key, and the system n has exponentially many $(n \to (n+1))$ keys. But all have the same predicate lattice, which is the Boolean one belonging to the state space $S = n$. This is because all three systems permit sufficient control of the internal state from the console. A predicate is uniquely defined by what excludes it, and for the counting system, actuating the one control $C1$ m times is excluded by any number of subsequent actuations greater than $n - m$, and not by any lesser number. These exclusions define the initial predicate uniquely.

The following lemmas are useful in computing the interactive logics of automata.

35. *Lemma.* If $Q'Q'' = Q'''$ then $(Q''')\mathrm{cl} \leqslant (Q')\mathrm{cl}$ and $\mathrm{cl}(Q''') \leqslant \mathrm{cl}(Q'')$. This tells us that longer histories define stronger predicates. In particular, $\mathrm{cl}(K_1)$ (where K_1 is the Kleene "lattice-one"; see paragraphs 23 and 24) is the predicate lattice-one:

$$\mathrm{cl}(K_1) = L1$$

36. *Lemma.* The (initial or final) lattice of an automaton (or of any finite dyadic relation, for that matter) consists of:

The lattice-zero $L0$ (see paragraph 23).

The closures $\mathrm{cl}(Q')$ of the individual control sequences Q'.

The joins of the above.

Proof seems otiose.

37. *Computing the lattice of a system.* Given the controls of a system, how do we compute its interactive predicate algebras? The following algorithm flows from the definition of paragraph 32.

Preliminaries. Index the controls with indices $C', 'C = C_1, \ldots, C_m$ and order the control sequences Q' according to numerical value, reading a sequence of controls as an integer in the base $m+1$. Index the states with indices $S', 'S = S_1, \ldots, S_n$. We write S_{123}, for example, for the class (S_1, S_2, S_3) and for the Boolean vector $S_1 + S_2 + S_3$ of this class. We induce on the length $L = 0, 1, 2, \ldots$ of the control sequences considered. For each L we form the list of all the Boolean vectors SQ' and $'QS$ for all Q' and $'Q$ of length $\leqslant L$. (These vectors represent the classes defined in paragraph 21.)

This list of classes defines the possibility relation $Q' : p : 'Q$ by

$$Q' : p : 'Q = SQ' \cdot 'QS$$

the Boolean inner product. This relation defines the lattice we seek.

ALGORITHM

Step A. ($L = 0$.) List the Boolean vector $S = (1, \ldots, 1)$.

Step B. (Increment L.) Suppose SQ' has been listed for all Q' of length $L - 1$. List $SQ'C'$ for all such SQ' and all C'. (Do this until $L = n$.) This lists all the final state classes.

Note: It is permissible to delete from the list any final state class SQ' expressible as the join of other final state classes in the list, and all its suffixed forms $SQ'C' \ldots C''$. In particular null classes and redundancies may be deleted.

Index the list with $s' = s_1, s_2, \ldots$.

Step C. List all the initial state classes $'QS$ dually to steps $A - B$ for sequences $'Q$ of length $\leqslant n$. Index the list with $'s = {}_1s, {}_2s, \ldots$.

Step D. Compute the Boolean matrix $s' \cdot 's$, the Boolean inner product of all the listed initial and final vectors. This matrix represents the possibility relation p.

Step E. Compute the final o class $s' \perp$ of each class s'.

Step F. Compute the initial closure $\mathrm{cl}(s') = n(p(s' \perp))$ of each s'.

Step G. Compute the $\vee$ joins (closures of the $\bigvee$ joins) of the classes of step F, paralleling steps $E - F$. These constitute the initial lattice.

38. *The interactive predicate algebra of the Moore Uncertainty Automaton* (see paragraph 20). We compute this lattice as an example of the above algorithm. The states are S_1, S_2, S_3, S_4. The controls are $C_1 = a$, $C_2 = b$, $C_3 = a'$, $C_4 = b'$. We designate products $C_1C_2\ldots$ by $C_{12\ldots}$. Steps A and B generate the following sequence of final state classes, where members later deleted by the Note of step B are starred.

Step A:

$$S = (1111)^*$$

Step B:

$$SC_1 = (0100) = S_2, \qquad SC_2 = (1010)^*, \qquad SC_3 = (0001) = S_4$$

$$SC_{22} = (0010) = S_3, \qquad SC_{32} = (1000) = S_1$$

Sequences with $L > 2$ generate no new final state classes. Thus $s_1 = S_1$, $s_2 = S_2$, $s_3 = S_3$, $s_4 = S_4$.

Step C:

$$S = (1111)^*$$

$$C_1S = (0010)^*, \qquad C_2S = (1001)^*, \qquad C_3S = (1101) = S_{124},$$

$$C_4S = (0110) = S_{23} \qquad C_{21}S = (1000) = S_1, \qquad C_{22}S = (0001) = S_4$$

Thus ${}_1s = S_1$, ${}_2s = S_3$, ${}_3s = S_4$, ${}_4s = S_{23}$, ${}_5s = S_{124}$

Step D. The 4×5 Boolean matrix representing p is

$$(s':p:'s) = \begin{matrix} 10001 \\ 00011 \\ 01010 \\ 00101 \end{matrix}$$

Step E. The o classes of $s_1, \ldots, s_4$ are represented by the Boolean vectors in $'s$ space

$$s_1 \perp = (01110), \qquad s_2 \perp = (11100), \qquad s_3 \perp = (10101), \qquad s_4 \perp = (11010)$$

Step F. The initial closures of $s_1, \ldots, s_4$ are themselves:

$$\mathrm{cl}(S_1) = S_1, \ldots, \mathrm{cl}(S_4) = S_4$$

Step G. The distinct $\vee$ joins of the s' are the lattice-one $L_1 = S_{1234}$ and

$$S_1 \vee S_2 = (1100) = S_{12}, \qquad S_1 \vee S_3 = (1110) = S_{123}, \qquad S_1 \vee S_4 = (1001) = S_{14}$$

$$S_2 \vee S_3 = (0110) = S_{23}, \qquad S_2 \vee S_4 = (0101) = S_{24}, \qquad S_3 \vee S_4 = (0111) = S_{234}$$

$$S_1 \vee S_2 \vee S_4 = (1101) = S_{124}$$

The 13 closures (including L_0) thus obtained constitute the initial lattice of the Moore uncertainty automaton (Figure 2). The dual calculation of the final lattice (omitted here) controls the accuracy, since it must result in the dual lattice.

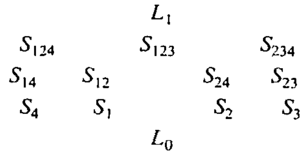

Fig. 2. Lattice of the Moore Uncertainty Automaton. Every element is a union of a set of the singlets S_1, S_2, S_3, S_4 indicated by its suffix. For example S_{234} is the union $S_2 \vee S_3 \vee S_4$. The suffixes thus define the partial ordering of the lattice elements. This mode of representation is quite generally applicable to interactive predicate algebras.

Since $S_2 = S_2\&(S_1 \vee S_3) \neq (S_2\&S_1)\vee(S_2\&S_3) = L0$, the lattice is nondistributive. It is modular since all paths to any lattice point from $L0$ have the same length. It is not orthocomplementable since there are four singlets but only three cosinglets.

This is not a quantumlike automaton, but something stranger.

39. For any system, the initial lattice is atomistic, and there is a monotone (implication preserving) 1–1 map on the initial lattice into the lattice $(S \rightarrow 2)$ of the space of its states; and another such map on the final lattice. ("Atomistic" means every member is a union of singlets, "atoms.")

Therefore interactive predicates, like S predicates, can be identified with certain (closed) classes of states, as far as there implication relations are concerned. Moreover, their conjoins are set-theoretic conjoins (intersections). Their joins and negations, however, are not set theoretic.

5. ABSTRACT TRANSITION SYSTEMS

40. The transition system used to represent automata in the preceding two sections is concrete in that the controls act on given states. All that is used to construct the interactive predicate algebra, however, is semigroup Q of control sequences. In some physical situations there are no accessible classical states, but only a collection of control operations C generating a semigroup Q of control sequences. The outcome of an individual experiment with control sequence Q need not be determined by Q, except in the case $Q = 0$, which stops the experiment. We call a semigroup Q with this interpretation an *abstract transition system*. The interactive predicate algebra of Q is again taken to be the Galois connection of the orthogonality relation

$$Q' : o : 'Q := Q' \cdot 'Q = 0$$

41. *Example: The system S(n, R).* The following example is constructed to include the infinite predicate algebras used in quantum theory and also finite predicate algebras arising from finite automata.

Let R be any ring. Elements of R are called *R numbers*. We will use R numbers as components of vectors and matrices and as values of inner products. The most important R's are

$$R = \mathbb{C} := \text{the complex number field}$$

$$R = \mathbb{Z} := \text{the integers}$$

$$R = \mathbb{Z}_p := \text{the integers modulo the integer } p$$

In orthodox quantum theory we use $R = \mathbb{C}$, but just as the distinction between real numbers and rational is unphysical, the distinctions between complex, integer, and mod-p quantum theories are somewhat unphysical, in the following sense: Any physical theory that can be expressed as a complex quantum theory can be expressed as closely as desired within an integer quantum theory, and for sufficiently large p within mod-p quantum theory.

Definition. $V(n,R):=$ the right module $n \rightarrow R:=$ the class of n-component vectors with components in R, regarded as a right module. $V(n, R)$ is a (right) linear space if R is a division algebra (i.e., field). We write a vector in $V(n, R)$ as a row of n R numbers.

Definition: $M(n,R):=$ the ring of $n \times n$ matrices with elements in R. If v is a vector and m is a matrix, their natural product is written vm.

Definition: $S(n,R):=$ the abstract system with control semigroup $M(n, R)$.

When $R = \mathbb{C}$, this becomes $S(n,\mathbb{C})$, the (complex) quantum system with multiplicity (i.e., "degeneracy") n. We further abbreviate $S(n,\mathbb{Z}_p) =: S(n, p)$.

Definition: $L(n,R) = L(o)$, where o is the orthogonality relation of the semigroup $M(n, R)$.

When $R = \mathbb{C}$, this lattice is isomorphic to the lattice of subspaces of $V(n,\mathbb{C})$, the predicate algebra of orthodox quantum theory.

42. *Example.* The lattice $1+4+1$ is isomorphic to $L(2,3)$.

Proof. The subspaces of $V(2,3)$ are 0, $V(2,3)$, and those defined by single binary vectors in $V(2,3)$. Every binary vector (a, b) with components in $\mathbb{Z}_3$ can be brought to the form $(a, 1)$ by scalar multiplication, since $\mathbb{Z}_3$ is a field, except those of the form $(a,0)$, which can be brought to the form $(1,0)$. Therefore there are only four rays (one-dimensional subspaces) in this linear space, each determined by a vector according to the following list:

$$S_0:(0,1)$$

$$S_1:(1,1)$$

$$S_2:(2,1)$$

$$S_3:(1,0)$$

With the lattice constants L_0 and L_1, these constitute the lattice $1+4+1$. ■

The lattice diagram for the lattice $1+4+1$ is Figure 3. If R is infinite then each predicate of $S(n, R)$ is a class containing an infinity of controls,

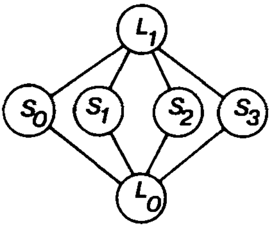

Fig. 3. Lattice 1 + 4 + 1. The lattice elements are L_0, the four singlets S_1, S_2, S_3, S_4, and the doublet L_1. Ascending paths represent the lattice order relation.

matrices in $M(n, R)$. A more economical system with the same logic can be made by singling out one of these matrices. This is done in the most common formulation of quantum theory:

43. *The quantum system $S(n, R, *)$.*

Definition. $(n, R)V$, the dual module to $V(n, R)$:= The set of linear functions: $V(n, R) \to R$. Elements of $(n, R)V$ are called covectors when elements of $V(n, R)$ are called vectors.

*Definition: *, the R conjugation*:= Complex conjugation if $R = \mathbb{C}$, := the identity transformation if $R = \mathbb{Z}$ or $\mathbb{Z}_p$.

*Definition: *, the adjoint operation on $V(n, R)$ $V(n, R)V$*:= The Hermitian adjoint (complex conjugate transpose) if $R = \mathbb{C}$,:= the transpose if $R = \mathbb{Z}$ or $\mathbb{Z}_p$. If for some nonzero v in $V(n, R)$, $v^*v = 0$, * is called *singular*. For $n > 2$ and $R = \mathbb{Z}_p$, * is singular. For $n = 2$, * is is singular for most p but not for some primes $p = 3, 7, 11, \ldots$. In the usual way, * is extended to the matrices $M(n, R)$:

$$v(mu) = (vm^*)u$$

for any vector v and covector u. We call m in $M(n, R)$ *self-adjoint* (with respect to *) if $m = m^*$.

Definition: Projector:= Idempotent self-adjoint m in $M(n, R)$:

$$m = mm = m^*$$

Definition: $V(n, R, *)$:= $V(n, R)$ with * as further element of structure.

Definition: $S(n, R, *)$:= the abstract transition system with semigroup generated by the projectors in $M(n, R)$.

44. *Examples of $S(2, R, *)$.* For some R this system represents an optical bench on which are mounted rotatable polarizing filters. The characteristic direction of such a polarizer, a ray in the plane of the polarizer, defines the control operation completely and is the prototype of all quantum ("psi") controls. Such a control may be represented nonuniquely by a unit vector C' in the ray. Malus's law, giving the conditional transition probability

$$P = (C'^*C'')^2$$

between two such polarizers with directions C' and C'', in terms of the scalar product (C'^*C'') of their polarizer orientation vectors, is the prototype of all quantum transition probabilities. In this example we do not see the state but only the controls and relations between them. In particular Malus tells us that two polarization controls are orthogonal in the systems sense (defined in paragraph 22) when their polarization vectors are orthogonal in the Euclidean sense, whence the name *orthogonal* for this logical relation. The simplest example of $S(n, R, *)$ that exhibits quantum logic is $S(2, 3, *)$, the *four-state polarizer.*

If we represent each polarizer by a projector C' instead of a vector, Malus's law for transition probability reads

$$P = \operatorname{tr}(C'C'')/\operatorname{tr}(1)$$

[Here tr(1) is the multiplicity, 2 for the photon polarizer.] We shall forget the probabilities provided by this law and remember just that the transition is possible if and only if the product of the operators is not 0. Control operations more general than polarization are represented by operators more general than projectors and may involve phase shift, rotation and attenuation.

According to the optimism now conventional in quantum physics, we can measure any normal operator. (We consider only finite-dimensional Hilbert spaces.) In the same spirit we suppose any projector C' represents a possible control. We represent the sequential product of controls by the product of the corresponding projectors. We suppose that such an operation C is selected from a keyboard attached to the optical bench, at which, say, the binary code for a polarization, attenuation, or other control parameter may be entered as a control character. A photon is then sent into the control element. If the photon stops in the control element, the recoil is observed and lights a "stop" sign. Otherwise there is no output.

More complex quanta than the photon, such as nuclei, atoms, and molecules, can undergo mechanized experiments of the same kind. Each control operation C corresponds to a generalized polarizer, an operation of the kind represented by a projector in a Hilbert space associated with the quantum. The sequential combination of such control operations is associated with the product of their projectors. It is peculiar to classical, commutative theories that such a product is again a projector. If this product is zero, the quantum stops in the experimental apparatus, and a new quantum must be produced for the next experiment.

45. *Hidden variables.* In the present formulation there is a natural "hidden variables" problem:

Given: the abstract transition system of a quantum theory, typically $S(n,\mathbb{C})$.

Find: a concrete transition system with the same semigroup (and hence the same interactive predicate algebra).

The given semigroup is one of projective algebraic (say, complex) matrices, multiplied algebraically, and represents a quantum system. The desired semigroup is one of Boolean matrices, multiplied relationally, and may be taken to represent a classical automaton. There is a familiar trick for turning an abstract group into a concrete group of transformations: Let the group act on itself by group multiplication. This trick does not work for all semigroups, but it works for the ones that concern us. Here are some examples of "concretizations" of abstract transition systems.

46. *Example: The concrete S(2,3,*).* There are four states S_1, S_2, S_3, S_4 and four controls C_1, C_2, C_3, C_4. The control C_n is the class of all pairs (S_m, S_n) with $m \neq n+2$ modulo 4.

This system represents operations performed on a beam of light falling from above by a polarizing filter lying in a horizontal plane when the polarization direction is restricted to four essentially different possibilities, such as the compass points E, SE, S, and SW, which define the four control operations C_4, C_1, C_2, C_3. The graph of this finite automaton is Figure 4. The nth control C_n maps mth state S_m into S_n unless S_m is diagonally opposite S_n, which is the case $m = n+2 \bmod 4$.

47. *Example: The Concrete S(n, R, *).* This solves the above hidden variables problem for orthodox quantum theory as the special case $R = \mathbb{C}$. We take the state space S of the automaton with transition system $S(n, R, *)$ to be the class of singlet projectors in $M(n, R)$. This entails an exponential, possibly infinite, increase in the dimension of the control matrices when we pass from the $n \times n$ algebraic matrices of the quantum theory to the Boolean matrices of the concrete transition system, which is actually a deterministic automaton. We take the control space $C = S$. The control relation is defined by

$$S' : C' : 'S : = S'C' \cdot 'S \neq 0$$

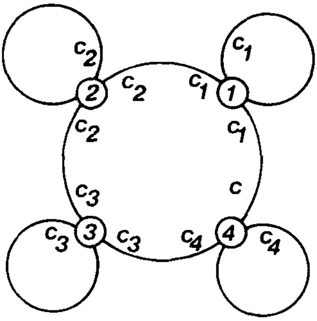

Fig. 4. Graph of the four-state polarizer. There are four vertices S_1, S_2, S_3, S_4 and 12 arcs. The control labels C_1, C_2, C_3, C_4 also direct the arc by standing next to the final state. This finite deterministic automaton is a hidden variables theory for the "quantum system" $S(2,3)$.

6. HEISENBERG RING OF A SYSTEM

48. Once the interactive predicate algebra of an operational system has been determined, we may seek the algebra of coordinates of the system. The algebraic elements we seek include what are called the observables in quantum theory. Since the product of two observables is sometimes not truly observable, we call these elements *operators* in general. In the quantum case, the algebraic representation of the predicate lattice provides one for the control operations and the observables also. The operators of Boolean systems commute, and those of quantum systems do not. It is well known that in both classical and quantum mechanics we identify each predicate with a projector in a linear algebra suited to the system. For systems in general, each physical quantity H (such as the Hamiltonian or energy), is defined in principle as a complete set of orthogonal predicates $P(n)$ paired with values $E(n)$ of the quantity, and, when an algebraic representation exists, is then identified with a spectral sum of its numerical values multiplying its projectors:

$$H := E(1)P(1) + \cdots E(n)P(n)$$

The association of a predicate with a projector in a ring is often tantamount to associating a predicate with an ideal, and thus to mapping the predicate lattice into the ideal lattice of the ring.

We call a ring used to represent the predicate lattice of a system in this way a *ring of the system*, and a Heisenberg ring of the system when more explicitness seems wanted. We write

$$\mathrm{Ring}(o)$$

for the class of Heisenberg rings of a system with orthogonality relation o.

49. If * is the transpose operation, reversing relations and products, then

$$\mathrm{Ring}(o^*) = \mathrm{Ring}(o)^*$$

50. The Kleene algebra of a system represents the system interactive predicate algebra much as a Heisenberg ring does. The predicates of the system may be identified with "ideals" of the Kleene algebra. K is neither linear algebra nor ring, but is an algebra in the sense of universal algebra, with the two semigroup operations $K'K''$ and $K' \vee K''$ obeying a distributive law. The orthogonality relation defines a lattice of members of K, which we wish to map into a lattice of ideals of a ring, preferably a ring of matrices.

An automaton now has two lattices associated with it: the Boolean one determined by its states, and the possibly non-Boolean interactive one. Similarly, the automaton may have two algebras of operators, the commutative one of its space of states, and the possibly noncommutative interactive algebra. The relation between the two predicate lattices, and the two algebras, is not a homomorphism. Similarly, two partial algebras (in the sense of Kochen and Specker, 1967) may be defined and are not necessarily homomorphic. The hidden variable theory of Kochen and Specker postulates such a homomorphism, and therefore does not describe the present models.

Many of the benefits of an algebraic representation first appear when we compose simple systems to make composite systems. This kind of composition is also part of interactive logic, but of its set theory, not its predicate algebra, and will not be considered in this paper.

51. *Example. The identity controller system.* A Heisenberg ring of the diagonal n-state system is the ring

$$(S \rightarrow \mathbb{Z})$$

of integer-valued functions on the space S, here an n-tuplet, with the addition and multiplication usual for integer-valued functions. The commutativity of this ring reflects the Boolean structure of the interactive predicate algebra of the system, and means that the operators of this system commute, as do all operators according to classical physics.

52. *Example: The four-state polarizer.* A ring for this system is Ring(2, 3), consisting of 2×2 matrices with $\mathbb{Z}(3)$ coefficients $0, 1, 2$. Vectors for $S(q=1)$ an $S(q=-1)$ may be chosen to be $(1, 0)$ and $(0, 1)$; $S(p=1)$ and $S(p=-1)$ may be chosen to be $(1, 1)$ and $(-1, 1)$, which cannot be normalized to 1, however. With these assignments, the spectral sums define Hermitian operators p and q, each a complete commuting set by itself, which do not commute:

$$q := S(q=1)S(q=1)^* - S(q=-1)S(q=-1)^* = \begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}$$

$$p := S(p=1)S(p=1)^* - S(p=-1)S(p=-1)^* = \begin{matrix} 0 & -1 \\ -1 & 0 \end{matrix}$$

$$pq - qp = \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$$

The matrices for p and q look like multiples of the familiar Pauli matrices, and have the noncommutativity typical of quantum systems, despite their origin in a deterministic automaton. Their elements, however, are integers modulo 3. The operators p and q are not observables of S in the classical sense, being undefined in some states. For example, no value of p is assigned to $S(q=1)$. They are interactive observables of CS. In domain CS, we model the control processes and their relations, not the quantities controlled.

53. When the ring of operators of a system is the ring of a module, the model elements ("vectors") are here called predicate vectors of the system. Each defines a singlet predicate represented by the projector onto the vector. We see here how superposable predicate vectors or wave functions arise in a classical system with an interactive logic.

7. CONCLUSIONS

54. We have not answered Question I. Both the quantum and logical incompleteness limit the predictions a system can make about a subsystem, to be sure. Here, however, we have studied only such limitations as arise from interactivity, and interactivity seems irrelevant to Gödel incompleteness. Nevertheless interactivity gives rise to non-Boolean logics which

sometimes resemble quantum logic. To predict a system takes a more complex system (Chaitin, 1966, 1982). We have not touched upon the limits to our knowledge of an automaton imposed by the limited complexity of the observing system, ourselves.

55. Instead we have answered Questions II, III, and IV:

Both automata and quantum systems define transition systems and orthogonality relations. The predicate algebras of the interactive logics of these systems are the Galois lattices of their orthogonality relations.

These predicate lattices differ in general from the Boolean predicate lattice of their state spaces because of the classical phenomenon of interactive interference, exemplified by the telephone busy signal (when we call our home, for example, to find out if the line is open, and thereby close the line to another caller.)

56. Specifically, the classical paradox of interactivity can generate the quantum paradoxes. If quantum kinematics did not exist, we would have to invent it in order to cope with some interactive automata. The quantumlike (orthomodular nondistributive) logics of multiplicity ≤ 2 are finite and can be simulated by finite automata, but not those of higher multiplicity. But predicate algebras like $L(n, p)$ of paragraph 23 for large n and $p \gg n$, when provided with a central ("superselection") square root of -1, seem good physical approximants to the standard quantum logics $L(n,\mathbb{C})$ over the complex numbers $\mathbb{C}$ and are realized by the finite automata $S(n, p)$ (see paragraph 41). These correspond to "Hilbert" spaces with a few self-orthogonal vectors.

57. Some finite automata exhibit interactive predicate algebras more deviant than those of quantum logic. For example, they may not admit any orthocomplement operation, singular or not. And some exhibit Boolean predicate algebras.

58. Infinite automata can exhibit quantum predicate lattices $L(n,\mathbb{C})$ of multiplicity n greater than 2, with infinitely many nonorthogonal singlet predicates.

59. *Implications for quantum physics.* These models seem to suggest new approaches to quantum physics.

There is still the main highway, where all obey the quantum principle of superposition, and basic physical theories use the usual quantum logic of Hilbert space or approximately equivalent discrete modules (Finkelstein 1982). This route has a unity that is unprecedented in science and can never be achieved in a classical theory: Its predicates are part of and generate its transformations.

But now a thin trail seems to double back toward the classical mode of thought about the possibility structure of a microsystem, cutting across some of our most cherished principles. On this path we think of internal states of a completely specifiable nature, hidden variables, without complementarity, and reconstruct quantum logic and quantum superposition at a higher level through the classical interference of controls, bypassing several well-known roadblocks on the main highway, such as the hidden variable theorems of Von Neumann (1932), and Kochen and Specker (1967).

This construction differs from most attempts to derive quantum logic from hidden variables in that predicate vectors are not postulated but constructed, and are appropriately related to experiments by that construction. Specifically each predicate vector describes neither the subject nor the object, but a history of interactions between them. As a result, predicate vectors are sometimes called "nonlocal," to distinguish how they relate to experiments from how physical fields relate to experiment, or are said to "collapse," because the beginning and end of an experiment may involve different controls. The predicate vectors constructed here have the relation to experiment sometimes expressed in these misleading ways. In the present theory there is no difficulty with the relation that exists in quantum theory between superposition of predicate vectors (ket vectors) and incompatibility or interference of experiments; nor with the multiplication of predicate vectors that represents the composition of systems. The rules for superposition and multiplication of predicate vectors follow from their operational definition, and agree with quantum theory.

It is not clear how far this trail goes. To push it further we must study how interactive predicate lattices compose when we build complex systems out of simple, and formulate an interactive set theory, as here we have formulated an interactive predicate algebra. For example, in order to study the Einstein–Podolsky–Rosen effect for automata, we must compose an automaton S of a line $S=T\cdots U\cdots V$ of at least three simpler ones T, U, V with "coherent interaction" between C, T, U, and V, and study communication from C to the central automaton U, to T and V, the two ends of the line, and back to C.

The principles of special relativity also require us to extend our language to composite systems, those having spatial structure.

60. *Implications for systems theory.* We have posed a basic question that should be asked about any system: What is its interactive predicate algebra (Question III)? An older question is: What are the states and transition function of the system? The simplest interactive system may exhibit quantum effects. For example, an interactive experimenter working with a finite deterministic automaton with transition system $S(n, p)$ (for any positive integers n, p) cannot describe the system better than with a "wave function" or transition amplitude, which will have n components belonging to the integers modulo p. Question III has the advantage that the predicate algebra

can be found from experiment and is often a more economical data structure than the states and the transition function.

61. *Implications for logic.* In this first paper, we have taken up only the interactive predicate lattice. We have hardly touched on the negation or orthocomplementation operation of the predicate algebra. Moreover, for application the statistical or probabilistic logic is more important than the possibilistic structure developed here. Finally, we have not formulated the levels of interactive logic corresponding to set theory. Interactive set theory is an aspect of interactive logic necessary for application to problems of large automata and artificial intelligence, which involve increasingly parallel control structure. Thus question I, questions of quantum physics, and questions of systems science and artificial intelligence all point to interactive set theory.

There is a parallel between the evolutions of physical geometry and physical logic from absolute and static structures to contingent and dynamic ones (Finkelstein, 1966). Riemann's theory of curved surfaces, an inner model of non-Euclidean geometry within Euclidean, stimulated the evolution of geometry from Euclid to Einstein. The present study likewise provides an inner model, now of non-Aristotelian logic within Aristotelian, and may stimulate the logical evolution. Previously, as described in Finkelstein (1966), working without such a guide, a dynamical logical structure was described by a quantum theory with hypercomplex (say, quaternion) amplitudes instead of complex. In the present models we find that "hypocomplex" rather than hypercomplex systems arise naturally. By this we mean number systems smaller than the complex, not larger, with complex quantum logic as an ideal unattainable limit. The missing structure is not located outside our ken, in extra dimensions, but inside, in the fine structure of the dimensions we already know.

ACKNOWLEDGMENTS

We thank G. Chaitin and F. J. Vareia for discussions of Question I; the Lindisfarne Institute for hospitality during some months of this work; and J. Bub for discussions of the importance of possibility and for unpublished work on Von Neumann. This paper is based on research supported by National Science Foundation Grant No. PHY800-7921.

REFERENCES

Artin, E. (1957). *Geometric Algebra*, Interscience, New York.

Bell, J. S. (1964). On the Einstein–Podolsky–Rosen paradox, *Physics* (*U.S.A.*), 3, 195.

Bell, J. S. (1966). On the problem of hidden variables in quantum mechanics, *Rev. Mod. Phys.*, **38**, 447.

Birkhoff, G. (1948). *Lattice Theory*, Amer. Math. Soc. Colloquium Publications, Vol. 25, rev. ed., New York.

Birkhoff, G., and Von Neumann, J. (1936). The logic of quantum mechanics, *Ann. Math.*, **37**, 823.

Bohm D. (1952). A suggested interpretation of the quantum theory in terms of "hidden" variables I, Ii, *Phys. Rev.*, **85**, 166, 180.

Brown, G. S. (1969). *Laws of Form*, Allen and Unwin, London.

Chaitin, G. (1966). On the length of programs for computing finite binary sequences, *J. Assoc. Computing Machinery*, **13**, 547, 16, 145.

Chaitin, G. (1982). Algorithmic Information Theory 1982, *Encycl. of Statistical Sciences* **1**, Wiley, New York, p. 38.

Conway, J. H. (1971). *Regular algebra and finite machines*, Chapman and Hall, London.

Finkelstein, D. (1963). The logic of quantum physics, *Trans. N.Y. Acad. Sci.*, **25**, 621–663.

Finkelstein, D. (1982). Quantum sets and Clifford algebras, *Int. J. Theor. Phys.*, **21**, 489.

Finkelstein, D. (1966). Matter, space, and logic, *Boston Colloquium on the Philosophy of Science*, Vol. 5.

Holland, Jr., S. S., (1970). The current interest in orthomodular lattices, in *Trends in Lattice Theory*, ed. Abbott, J. C., Van Nostrand Reinhold, New York. Reprinted in *The Logico-Algebraic Approach to Quantum Mechanics*, Vol. I, ed. Hooker, C. A., Reidel, Dordrecht (1975).

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

Kochen, S., and Specker, E. P. (1967). The problem of hidden variables in quantum mechanics, *J. Math. Mech.*, **17**, 59–87.

Kaufman, L., private communication.

Moore, E. F. (1956). Gedanken experiments on sequential machines, in *Automata Studies*, ed. Shannon, C. E., and McCarthy, J., Princeton University Press, Princeton, New Jersey.

Vareia, F. J., private communication.

Von Neumann, J. (1960). *Continuous geometry*, Princeton University Press, Princeton, New Jersey.

Von Neumann, J. (1954). Unsolved problems in mathematics, Address to International Mathematical Congress, Amsterdam, September 2, 1954, unpublished. Manuscript, Von Neumann archives, Library of Congress, Washington, D.C.

Von Neumann, J. (1932). *Mathematische grundlagen der quantenmechanik*, Springer Verlag, Berlin. Reprinted, Dover, New York (1943).

Wheeler, J. A. (1982). The computer and the universe, *Int. J. Theor. Phys.*, **21**, 557.

M. Zwick, (1978). Quantum measurement and Gödel's proof, *Speculations Sci. Technol.*, **1**, 135.