Linear Logic for Generalized Quantum Mechanics

Vaughan Pratt*
Dept. of Computer Science
Stanford University, CA 94305
pratt@cs.stanford.edu

Abstract

Quantum logic is static, describing automata having uncertain states but no state transitions and no Heisenberg uncertainty tradeoff. We cast Girard's linear logic in the role of a dynamic quantum logic, regarded as an extension of quantum logic with time nonstandardly interpreted over a domain of linear automata and their dual linear schedules. In this extension the uncertainty tradeoff emerges via the "structure veil." When VLSI shrinks to where quantum effects are felt, their computer-aided design systems may benefit from such logics of computational behavior having a strong connection to quantum mechanics.

1 Motivation

VLSI designers will eventually need to reckon with quantum mechanics (QM), certainly within 50 years and possibly by the 2010's. Computer-aided design (CAD) tools will need to adapt accordingly.

Does this entail changing only the theory or also the logic of CAD verification systems? That is, will it be enough just to change those nonlogical axioms defining how electronics works, or will it prove necessary to dig deeper and tamper with logic itself?

Birkhoff and von Neumann [BvN36] incorporate quantum uncertainty into Boolean logic by interpreting conjunction $A \wedge B$ standardly but modifying negation $A^\perp$ to mean certainly not A, in the sense that $A^\perp$ holds in just those states having probability 0 of being mistaken for a state in which A holds. This relationship of unmistakability is an irreflexive symmetric binary relation on states called *orthogonality* and denoted $x \perp y$. Disjunction as the De Morgan dual of conjunction does not distribute over conjunction in this *quantum logic* (QL), suggesting that the emergence of quantum phenomena in VLSI will force changes to CAD verifiers at as fundamental a level as their Boolean logic.

A set of states with such an orthogonality relation amounts to an automaton with fuzzy states but without transitions. Hence this quantum logic is a static logic expressing uncertainty but not progress. In addition the uncertainty so represented is not full Heisenberg uncertainty, in that it obeys no complementarity tradeoff. And the logic is not a real logic in that it lacks an implication with a deduction theorem or currying principle.

Quantum logic as a faithful abstraction of quantum mechanics must necessarily be extendable to include these dynamic elements of time and Heisenberg uncertainty. Time is standardly added to quantum logic in terms of the group of automorphisms of the underlying ortholattice, and from this one recovers Heisenberg uncertainty [Var68].

In this paper we raise the problem of formulating this extension as a *language* extension. That is, what propositional logic of quantum mechanics incorporates not only QM's uncertainty but also its central dynamic qualities? While such a question would have had no precedent in 1936, today we have a great variety of logics combining the elements of truth and behavior [Flo67, Hoa69, Sal70, dBdR72, Dij76, Pra76, Pnu77], making our problem a very reasonable one.

*This work was supported by ONR under grant number N00014-92-J-1974, and a gift from Mitsubishi.

To indicate the form such an extension might take, we cast Girard's linear logic [Gir87] as a "generalized dynamic quantum logic." It is made dynamic by incorporating a temporal element. It is generalized in that nonstandard notions of time are admitted. We give several interpretations of linear logic that convey to varying degrees the dynamic quality of the extension, relative to the base construed as a static logic.

The model we find most evocative of physical behavior is that of a restricted class of linear automata and their dual linear schedules, state spaces and their dual event spaces. (Elsewhere we describe in detail the yet more restricted class of chains as rigid local behaviors [Pra92b].) In the section on measurement we anticipate a broader class of linear automata, partial distributive lattices, but defer its detailed treatment to another paper.

However the notion of time contemplated in this model is that of computer science rather than physics, in that it is constituted as a partial order rather than a group. Furthermore the complementary parameters of time and information behave more symmetrically for linear automata than do the corresponding complementary physical parameters of time and energy for the Schrödinger wave equation. In the latter, attributes are represented by (the eigenvectors of) operators on the Hamiltonian, which denotes energy, whereas time parametrizes a group of unitary operators $U(s+t) = U(s)U(t)$ transforming Hilbert space and acting as automorphisms of the underlying ortholattice.

But despite this departure from orthodox quantum mechanics we may observe the quantum-like notions of complementarity and Heisenberg uncertainty in linear automata. Our claim is that these are the analogues for ordered time of the phenomena of those names encountered in orthodox quantum mechanics. The corresponding mathematical basis for the the respective complementarities is duality: Stone duality for computer science, Pontrjagin duality for physics. We hope in the future to formulate and apply dualities obtained from other metrics such as real- or complex-valued distance [CCMP91] to dynamic quantum logic.

It is often asked "how could anything real be like quantum mechanics?" A hybrid computational/quantum mechanics that substitutes certain elements of perceived reality for quantum reality while retaining the uncertainty tradeoff of standard quantum mechanics creates a bridge between perceived reality and quantum reality that may prove helpful in making this transition. Having such a hybrid might make it easier for systems designers and physicists collaborating on quantum CAD to talk to each other. And it may conceivably find some application in the pedagogy of standard QM and elsewhere.

We leave as an open problem the interpretation of linear logic for quantum mechanics with its standard notion of time. It seems clear that we must replace the ordered monoid $\{0 < 1\}$ with the group of complex numbers, with or without origin (the latter when viewed as a multiplicative group), as the automaton and the schedule interpreting linear logic's constant $\perp$, and also as the ring of scalars over which the spaces are defined. What is presently lacking is a quantum-mechanically orthodox organization of the duality between the temporal metric on events and the information (energy?) metric on states.

To summarize, we extend static quantum logic to a richer dynamic quantum logic, but modeled for computational rather than physical processes. This gives computer science a quantum mechanics of its own, while leaving open the question of whether orthodox quantum mechanics can model linear logic along similar lines.

The rest of the paper is organized as follows. We start with quantum logic as a conventional propositional logic with one of each operation: conjunction, negation, disjunction, and give the semantics making it the logic of an orthomodular lattice. We then present linear logic as the extension of quantum logic, construed as a *static* logic, with a second *dynamic* set of connectives, further endowed with one implication for each set, along with a connective $!A$ coordinating the two sets. We then exhibit both of Girard's models of linear logic: the nonconstructive quantum-logic-like phase space model and the constructive coherence spaces model, preparing the reader for the latter with a minitutorial in constructive logic. Next we describe state spaces and their complementary event spaces, first as a model of concurrent behavior, then as a constructive model of linear logic, giving a sense in which linear logic is a logic of behavior. Finally we describe the "structure veil," interpreted as Heisenberg uncertainty and illustrated with partial distributive lattices, the details of which will appear in another paper.

2 Quantum Logic

2.1 Static Logic

Common to quantum and linear logic are the constants 0 and 1, conjunction $A \wedge B$, disjunction $A \vee B$, and negation $A^\perp$. These are the constants and operations of a bounded lattice, satisfying (i) $0 \vdash A$ and $A \vdash 1$; (ii) ($C \vdash A$ and $C \vdash B$) iff $C \vdash A \wedge B$; (iii) $A \vee B \vdash C$ iff ($A \vdash C$ and $B \vdash C$). For both these logics we require that negation, $A^\perp$, be a *duality*: it must be (iv) *contravariant*: $A \vdash B$ iff $B^\perp \vdash A^\perp$, and (v) an *involution*: $A^{\perp\perp} = A$.

The nonconstructive interpretation of these conditions is that a model is a partially ordered set $(X, \leq)$, made an algebra with these constants and operations, and with $A \vdash B$ interpreted as $A \leq B$. We will treat the constructive interpretation of the conditions later.

We shall think of this logic as *static* in the sense that it talks about states of affairs rather than about events of change. While the apparently temporal proposition "It is Tuesday" is a legitimate atom P of this logic, the validity of $P \wedge P = P$ reveals its static nature. In this logic asserting P twice does not make it any more true. As we will see later, linear logic extends this static logic with a dynamic conjunction $\otimes$ for which $A \otimes A$ does not in general equal A.

2.2 Orthoframes and Ortholattices

The following notions pervade both quantum and linear logic. A *Kripke structure* (X, R) consists of a set X of possible worlds or *states* together with a binary relation $R \subseteq X^2$ of *accessibility* between states. We identify each proposition about such states with the set of states in which it holds. An *orthoframe* [Gol74] is a Kripke structure $(X, \perp)$ where $\perp$ is a symmetric irreflexive binary relation of *orthogonality* relating pairs of states not mistakable for each other.

For any subset $Y \subseteq X$ we define $Y^\perp$ to be $\{z \mid \forall y \in Y[z \perp y]\}$, those states not mistakable for any state of Y. A $\perp$-*closed* subset A is one for which $A = Y^\perp$ for some Y. It may be verified that the $\perp$-closed subsets A are just those satisfying $A^{\perp\perp} = A$; we take these for the propositions of quantum logic. These are closed under arbitrary intersection and hence form a complete lattice, with the join of a set of propositions being the intersection of all propositions including them all. Moreover $A^\perp$ is both a duality (making $A = A^{\perp\perp}$ an alternative condition to be $\perp$-closed) and a complement, and satisfies the usual De Morgan's laws [Bir67, p.123].

A lattice $(L, \vee, 0, \wedge, 1, \perp)$ for which $A^\perp$ is a duality and a complement is called an *ortholattice.* A *Boolean algebra* is a distributive ortholattice, i.e. all non-Boolean ortholattices necessarily fail distributivity.

2.3 Quantum Logic

Quantum mechanics is based on a Hilbert space of states, a metrically complete inner product space, made an orthoframe by interpreting orthogonality standardly for Euclidean space. Although completeness does not have a first-order definition for either inner product spaces or orthoframes [Gol84], the condition of orthomodularity [BvN36] on the associated orthoframe does express completeness exactly [AA66]. (Orthomodularity can be defined as either $A \vdash B$ and $A^\perp \wedge B = 0$ implies $A = B$, or $A \wedge (A^\perp \vee (A \wedge B)) \vdash B$. The second "ortho" in "orthomodular ortholattice" can be dropped.)

Quantum logic is defined as the logic of orthomodular lattices. It is a faithful abstraction of "concrete" QM, whose attributes are projections of Hilbert space onto itself, hence subspaces of Hilbert space.

Distributivity fails in the standard model. For simplicity consider R^3 instead of $\mathcal{H}$. Take A to be the X-axis and B to be a diagonal on the XY plane. Then $A^\perp$ is the YZ plane, and $(A \vee A^\perp) = 1$ while $A \wedge B$ and $A^\perp \wedge B$ are both 0. Hence $(A \vee A^\perp) \wedge B = B$ while $(A \wedge B) \vee (A^\perp \wedge B) = 0$, a violation of distributivity. Birkhoff and von Neumann cite as a physical example of the same breakdown of distributivity the case of A and $A^\perp$ as the observations of a wave-function (a state in $\mathcal{H}$) on one side or the other of some plane, and B the observation of a wave-function in a symmetric state about that plane [BvN36].

2.4 Limitations of Quantum Logic

Quantum logic has been intensively studied in the intervening 56 years. A very recent bibliography of the subject lists 1851 papers [Pav92]. Evidently this appealingly simple system has touched a responsive chord with

many philosophers of physics.

But quantum logic may be *too* abstract to be useful. First, it is a very minimal logic of QM, not even addressing time, in particular saying nothing about how wave functions evolve and how propositions may change their values with time. Nor does Heisenberg uncertainty appear, even implicitly; all that is represented is whether or not any two given states are mistakable for each other.

Second, QL has no reasonable implication. Dalla Chiara [Dal86] surveys a number of candidates for a quantum logic implication, all of which are lacking in various ways. There does exist the well-known *Sasaki hook* $A{\to}B = A^\perp{\vee}(A{\wedge}B)$, and Finch [Fin70] has given a conjunction $A.B = A{\wedge}(A^\perp{\vee}B)$ to go with it satisfying $A.1 = A = 1.A$, $A.0 = 0 = 0.A$ (i.e. $A.B$ has the same unit and annihilator as $A \wedge B$), $A.A = A$ (idempotence), and $A.B \vdash C$ iff $B \vdash A{\to}C$. However if A.B is associative, commutative, or monotone in A [RR91, Prop. 1.2], or if there exists an implication $\to'$ such that $A.B \vdash C$ iff $A \vdash B{\to'}C$, the lattice collapses to a Boolean algebra, showing that all of those conditions are unsound for the standard model. Other implications are similarly flawed.

These objections are overcome in the extension of quantum logic to linear logic as a dynamic quantum logic.

One might also question whether the notions of duality and complement should be conflated in the one operation. This rules out such models of truth as chains with more than two truth values including fuzzy logic, as well as our linear automata. This is not however a valid objection to standard quantum logic, only to generalized quantum logics, for which the natural duality may not be complementation.

A conceivable objection might be raised on the basis of Goldblatt's observation that completeness of inner product spaces cannot be expressed in the first-order language of orthoframes, witnessed by the inner product subspace of countable-dimension Hilbert space $\mathcal{H}$ consisting of all ultimately zero sequences, which is incomplete but elementarily equivalent to $\mathcal{H}$ [Gol84]. We do not find this objectionable either. Transitive closure of binary relations is likewise not first-order expressible, yet this does not negate the importance of being able to reason about transitive closure, a central concern for both database managers and program verifiers. Instead both first-order approximations (induction in Peano arithmetic) and exact second-order solutions (dynamic logic [Pra80], relation algebras with transitive closure [NT77, Ng84]) have been proposed and used. The same considerations should apply to orthomodularity, which has a straightforward second-order definition which becomes not only first order but an equation when expressed in the language of ortholattices.

3 Linear Logic

3.1 Dynamic Connectives

To the connectives of static or additive logic,[1] linear logic adds a fifth, a static implication $A{\Rightarrow}B$. Then to the resulting five items it adds a parallel set of five more items, the *dynamic* or multiplicative connectives and constants, yielding the following language.

Static :	$A \wedge B$	1	$A \vee B$	0	$A{\Rightarrow}B$
Dynamic :	$A{\otimes}B$	$\top$	$A{\parr}B$	$\bot$	$A{\multimap}B$

In addition there is a unary connective $!A$ expressing static logic in dynamic terms. We write $A^\perp$ for $A{\multimap}\bot$, called linear negation,[2] and $?A$ for $(!(A^\perp))^\perp$, the De Morgan dual of $!A$. This completes the language of linear logic.

The idempotence of $A \wedge A$ and dually $A \vee A$ does not hold for the dynamic connectives. Except for these, the dynamic logic on its own behaves like the static logic. The conjunctions and disjunctions are associative and commutative, and the four constants are the four units of their associated connectives. However no distributivity laws hold within each logic.

A full axiomatization of constructive linear logic is presented in algebraic language in Section 3.5, to which readers preferring crisp definitions are referred. Here we shall continue to convey the flavor of linear logic in algebraically less demanding terms.

[1] Girard [Gir87] notates $A \wedge B$ as $A\&B$, $A \vee B$ as $A \oplus B$, and 1 as $\top$. To reduce the burden on the reader we have stuck with more standard notation. The only dynamic symbol not Girard's is the constant $\top$.

[2] This makes $A^\perp$ a dynamic connective. One is tempted to speculate that quantum logic's $A^\perp$ would benefit from a dynamic interpretation.

Thus far the only basis for differentiating the two logics is that negation is defined with dynamic implication. A more fundamental difference is that the dynamic logic draws distinctions that the static does not. As the names "static" and "dynamic" suggest, these distinctions can be thought of as motion-related. We shall shortly give several formal interpretations of linear logic. To ease the transition let us start with a relatively informal interpretation.

We view propositions as moving at various speeds. The dynamic logic takes the motion into account to arrive at reliable conclusions. The static logic ignores the motion, drawing its conclusions under the assumption that all propositions are stationary. Some propositions are indeed stationary, including all propositions $!A$, which we think of as A halted or frozen, and for these static logic comes to the correct conclusion. But for moving propositions dynamic logic reasons that certain combinations, such as cars moving towards each other at high speed, will combine messily, while static logic naively assumes no damage.

Static logic's optimistic reasoning is formalized by the identity $!(A \wedge B) = !A \otimes !B$. That is, if frenetically moving A and B are first optimistically combined *statically* ($A \wedge B$) and then brought to a halt, the statically predicted effect is that of halting A and B individually to yield $!A$ and $!B$, and then using dynamic reasoning to infer that these stationary objects will combine harmlessly.

The constructive formalization of $!A$ will later be seen to make $!A = F(U(A))$, the Free dynamic object generated by the Underlying static (intuitionistic or cartesian closed) structure of A. (In the jargon, $!$ is a comonad on the category of dynamic behaviors having a cartesian closed Eilenberg-Moore category of underlying or static structures.)

The informal velocity metaphor does not convey all of the intuition, so we supplement it with a second informal metaphor. Instead of calling the static reasoner an optimist we will think of her as an angel, and the dynamic reasoner as a mere mortal. To an angel everything including mortals seems frozen in time, while to a mortal angels exist only as an article of faith. Angels reason with static connectives, mortals with dynamic.

The identity $!(A \wedge B) = !A \otimes !B$ presents a mortal view of an angelic conjunction as a mortal conjunction. More secularly, we may regard $!A$ as an X-ray view of A, delivering a visible conjunction of parts normally seen in conjunction only in the mind's (optimist's, angel's) eye.

Although static implication can be considered defined by its Deduction Theorem, we regard its real definition as $A \Rightarrow B = !A \multimap B$. We view dynamic $A \multimap B$ as the *employment* of mortal B in the *observation* of experiment A. Thus $A \Rightarrow B$ denotes B employed as before but now equipped with an X-ray machine. This expansion of B's powers confers on B the vision of an angel.

We turn now to the more formal models of linear logic. We begin with Girard's nonconstructive model, phase space, then treat constructive logic in preparation for the constructive coherence spaces model.

3.2 Phase Space

The linear logic of phase space is that of lattices with a duality (ortholattices less the requirement that $A^\perp$ be a complement), based on a not necessarily irreflexive orthoframe, and extended to talk about a monoid structure, the dynamical component. Formally, a *phase space* $(X, +, \mathbf{0}, \perp)$ is a commutative monoid $(X, +, \mathbf{0})$ (say the reals under addition) together with a subset $\perp \subseteq X$, the unit of $\parr$, defining a binary relation $x \perp y$ holding just when $x + y \in \perp$. We no longer require irreflexivity for orthogonality, whence we cannot guarantee that $A^\perp$ will be a complement.

The monoidal structure serves to define the dynamic connectives. $A \otimes B$ is defined as $\{a + b \mid a \in A, b \in B\}^{\perp\perp}$, with De Morgan dual $A \parr B$. We take $\top = \perp^\perp$. Dynamic implication $A \multimap B$ is given by $A^\perp \parr B$.

In this model think of a proposition as denoting a set of abstract tasks, each identified with the quantity of say energy required for it. Call two tasks *compatible* when the energy they consume acting jointly sums to one of the quantities in the set $\perp$, which consists of all compatible such binary sums. Hence $A^\perp$ consists of all quantities compatible with every quantity in A. $A^\perp$ is contravariant: the bigger A gets the harder it becomes to belong to $A^\perp$. Now suppose A is a set of trains and B a set of stations to be visited by those trains. Then $A \otimes B$ is the set of all arrivals of trains at stations, each arrival identified with the quantity of energy jointly expended by the train and the station on that arrival. (Arrivals of the same joint energy will be coalesced.) This set is then $\perp$-closed to make it a proposition. The static connectives serve merely to combine these sets logically, with conjunction being interpreted standardly and disjunction being nonstandard on account of the fuzziness created by nonorthogonality.

The operation $!A$ is an *interior* operator, namely monotone ($A \vdash B$ implies $!A \vdash !B$), idempotent ($!!A = A$), and decreasing ($!A \vdash A$). Its image (and also its fixpoints) are the *open* or *free* sets. (These suffice to determine $!A$, namely as the largest open set contained in A.)

Furthermore $!A$ is a *dynamic-topological* interior, meaning that $!(A \wedge B) = !A \otimes !B$ and $!1 = \top$. (Static-topological would have meant ordinary topological interior, satisfying $!(A \wedge B) = !A \wedge !B$ and $!1 = 1$. Traditional intuitionistic logic is necessarily of this static kind since it lacks a separate dynamic intersection distinct from static intersection, cf. Stone's topological treatment of intuitionistic logic at the end of his paper [Sto37].)

We define the dual notion of *closure*, $?A$, as $(!(A^\perp))^\perp$, and *static implication* $A \Rightarrow B$ as $!A \multimap B$.

We have seen that the language of linear logic is that of quantum logic expanded with dynamic connectives and constants, and $!A$ and $?A$. The laws of linear logic, at least those of its phase space model, conservatively extend those of quantum logic less the axioms of complementation ($A \vee A^\perp = 1$) and orthomodularity. $A \otimes B$ is associative with $\top$ as its unit. $A \parr B$ is the De Morgan dual of $A \otimes B$, and $\perp$ the negation of $\top$.

We have $A^\perp = A \multimap \perp$. Also $(A \otimes B) \multimap C = A \multimap (B \multimap C)$. Hence setting $C = \perp$ yields $(A \otimes B)^\perp = A \multimap B^\perp$. And we have two inference rules: from $(!A)^n \vdash B$ infer $!A \vdash B$, and from $A \vdash B$ infer $!A \vdash B$.

Characteristic properties of nonconstructive linear logic are $A \vee A = A = A \wedge A$.

These are a representative but not complete list of properties of the phase space model.

3.3 Constructive Logic

So far both quantum logic and linear logic have been nonconstructive in the sense that the models have been posets. Both Girard's coherence spaces and our linear automata make the transition to *constructive logic*, which we define here. Since few readers with an interest in either hardware verification or quantum mechanics are likely to be familiar with constructive logic, and since there are no really accessible crash courses in the subject, we take the liberty of going into some detail.

Whereas nonconstructive logic takes truth to be the purpose of mathematics, constructive logic takes proof to be its practice. It is the same with travel and communication. Existence of routes tells whether you can get there, but when you need to get there you need a particular route. Existence of a derivation or parse tree establishes grammaticality, but when you need to understand you need a particular phrase structure. Since physics treats the practice of the universe rather than its purpose, physicists are likely to find constructive logic more useful than nonconstructive.

Consider the nonconstructive definition of $A \vee B$ as the unique binary operation on a poset satisfying the rule $A \vee B \vdash C$ iff ($A \vdash C$ and $B \vdash C$). This definition treats $A \vdash B$ as the fact $A \leq B$, "and" as Boolean conjunction, and "iff" as equality of truth values of such facts.

We adopt the constructive interpretation of $A \vdash B$ as a set of proofs of B from A (called a *homset* $\mathrm{Hom}(A, B)$). Proofs $A \xrightarrow{f} B$ and $B \xrightarrow{g} C$ *compose* (the *cut rule*) to prove $A \xrightarrow{fg} C$, an associative operation. And $A \vdash A$ always contains the trivial or identity proof of A from itself. These conditions will be recognized as those defining a *category* whose objects are propositions and whose morphisms are proofs.

Continuing with the same example, we interpret "and" as cartesian product of such homsets, and "iff" as a bijection between homsets. The primary purpose of the bijection is not to serve as a constraint on the interpretation of $A \vee B$ but to give a method for dealing with conjunctions when they are encountered. However it does succeed in constraining the interpretation indirectly, namely via the assumption that the method will always be implementable.

The category of sets and binary relations supplies an example interpretation of propositions and proofs, making $A \vdash B$ the homset of all $2^{|B| \times |A|}$ binary relations from A to B. The right hand side of the rule, $(A \vdash C) \times (B \vdash C)$, is then the homset of all $2^{(|A|+|B|) \times |C|}$ pairs f, g of relations $A \xrightarrow{f} C$ and $B \xrightarrow{g} C$. Then the method will be implementable provided $A \vee B$ has been interpreted as any set D of cardinality $|A| + |B|$, permitting the implementation of a bijection between $D \vdash C$ and $(A \vdash C) \times (B \vdash C)$. Such a bijection is called an *adjunction*.

This is just a local bijection for a single pair A, B of propositions. We further require that the collection of all such bijections of the logic be *natural* [Mac71, p.16,p.77], a global consistency condition whose effect is to force D to look like the same juxtaposition or *disjoint sum* $A + B$ to all such bijections.

Now the rules and axioms of a constructive logic are not merely conditions on the interpretation of connectives, but methods for dealing with connectives. A corollary of this orientation is that the above bijection must be

regarded as a part of the logic: each different choice of a bijection meeting these conditions determines a different constructive logic. A constructive logic is thus like an algebra, with proofs for elements and bijections between proofs for operations. (Propositions A can participate here by being identified with the trivial proof $A \xrightarrow{1_A} A$.) The process of proving is called *derivation*, and yields a particular proof.

This is not to say that the rules do not constrain. The connectives must be interpreted so as to guarantee the implementability of the methods. This constraint can be thought of as the nonconstructive component of a constructive logic.

Returning to nonconstructively defined $A \vee B$, it is possible to derive associativity, $(A \vee B) \vee C = A \vee (B \vee C)$, from the definition, as the truth value 1. The corresponding constructive situation is that we can derive $(A \vee B) \vee C \cong A \vee (B \vee C)$, not as a truth value but as a certain isomorphism in some category.

Why the switch from equality to isomorphism? Our above sets-and-relations example seemed on the surface to define $A \vee B$ associatively, namely as juxtaposition or disjoint union $A + B$. The trouble is, we cannot rely on those elements in $A \cap B$ to keep track in $A \vee B$ of their respective origins, forcing us into some system of marking to keep track. But such a system will typically mark elements of A twice in $(A \vee B) \vee C$ and only once in $A \vee (B \vee C)$, with the result that $(A \vee B) \vee C = A \vee (B \vee C)$ may fail. However the two sets *are* the same size, so at least we can always put them in bijection. Furthermore there is a unique "sensible" bijection, namely what would be the identity if only we could erase the marks without confusion. *This* bijection is the isomorphism that we are able to derive constructively in the logic. The bijection $A \vee B \vdash C$ iff ($A \vdash C$ and $B \vdash C$) specified by our constructive logic uniquely determines the associativity isomorphism. Furthermore the naturality of the bijection guarantees the naturality of the isomorphism.

So what's the point of all this? Nonconstructive logic was easy, why are we making it so complicated with naturality and isomorphisms and so on?

In fact a nonconstructive reformulation of our constructive logic of automata that captured its essence would be highly desirable. At present we do not see how to do it. Meanwhile to take some of the edge off the complexity of this constructivity we look on the bright side as follows.

Constructive logic is object-oriented logic. A proposition is not an abstraction outside the domain of discourse but an object inside the domain. Much of modern mathematics consists of methods for manipulating relations and functions between objects. By organizing these methods into a calculus we obtain an attractive formal system. And by drawing analogies with logic we make the rules of this system seem more familiar—they become the familiar rules of traditional logic or the not-so-familiar rules of constructive quantum logic.

3.4 Coherence Spaces

So far our models have been single orthoframes, together with a monoid in the case of phase space. Now we break up that one orthoframe into many smaller orthoframes that we then take as our propositions. We pass from nonconstructive to constructive linear logic by replacing the inclusion between propositions A, B with a set $A \vdash B$ of "proofs" of B from A.

A *coherence space* is an orthoframe $(X, \perp)$. (Girard calls this a *web* and defines a coherence space to be the poset of the coherent subsets of the web under inclusion.) We say that distinct x, y are *coherent* when $x \perp y$ and *incoherent* otherwise. The *coherent subsets* are those $Y \subseteq X$ satisfying $Y^2 - I \subseteq \perp$: all distinct pairs in Y are coherent. (Incoherence or *conflict* should be thought of here not so much as partial interference as in quantum logic but as complete incompatibility.)

A binary relation R from $(X, \perp)$ to $(Y, \perp')$ is *coherent* when any two distinct y's that R relates x to are coherent, and *stable* when any two distinct x's that the converse of R relates y to are incoherent. The category **Coh** of coherence spaces has all orthoframes for its objects, and for its maps from $(X, \perp)$ to $(Y, \perp')$ all coherent stable binary relations R from X to Y.

Define the *dual* of the orthoframe $(X, \perp)$ to be the orthoframe $(X, \perp')$ obtained as $\perp' = X^2 - I - \perp$. In the dual, coherence and incoherence are interchanged. If we dualize the objects of **Coh** and also replace every binary relation by its transpose, we obtain a category of orthoframes and their stable coherent binary relations once again. Since no orthoframes or relations have been gained or lost, we must have obtained **Coh** back again, but with its objects permuted. We call the correspondence induced by this dualizing of objects and reversing of arrows a *contravariant isomorphism* or *duality* between **Coh** and itself. If we define the *opposite* of **Coh**, written

$\mathbf{Coh}^{\mathrm{op}}$, to be **Coh** with its edges reversed then we can dispense with the adjective "contravariant" and say that this duality is an *isomorphism* between **Coh** and $\mathbf{Coh}^{\mathrm{op}}$. (This duality is harder to see with the linear stable function approach.)

We now give the coherence space interpretations of the operations of linear logic. $A^\perp$ is the dual defined in the preceding paragraph. $A \vee B$ is their disjoint union or juxtaposition. (Hence if A has 3 elements and B 4 then $A \vee B$ has 7, and $\perp$ is defined as for each of A and B separately, with $a \perp b$ false for any $a \in A$ and $b \in B$.) $A \wedge B$ is their cartesian product, with $(a,b) \perp (a',b')$ just when $a \perp a'$ and $b \perp b'$. This is very much like vector spaces over $\{0,1\}$ but with vectors being allowed to interfere (nonorthogonality).

The additives juxtapose spaces side by side, coherently for $A \wedge B$ and incoherently for $A \vee B$, representing "both" or "either" respectively. The multiplicatives can be thought of as a more intimate mixing consisting of the interactions of A and B: one interaction for every pair (a,b) from A, B respectively, with pairs cohering pairwise for $A \otimes B$ and dually for $A \parr B$.

We take $!A$ to consist of all *finite* coherent subsets of A, with $Y \perp Z$ just when $Y \times Z \subseteq \perp$, that is, every element of Y coheres with every element of Z. The remaining operations are defined standardly.

(In Girard's version of coherence spaces, the binary relations become functions between coherence spaces. A binary relation from X to Y can described as a function $f : X \to 2^Y$. The functions corresponding to coherent relations have only coherent subsets of Y in their image, the set of which we can therefore substitute for 2^Y. We can linearly extend the domain of f to the coherent subsets of X by defining $f(Z) = \bigcup_{z \in Z} f(z)$ at each coherent $Z \subseteq X$, so that the extended f is a linear function between coherence spaces. The functions corresponding to stable relations 2^Y map distinct coherent pairs to disjoint subsets of Y, so we call such functions stable. Then **Coh** can equivalently be defined as the category of coherence spaces defined in this way and the linear stable functions between them.)

3.5 Algebraic Definition of Constructive Linear Logic

Before going on to the next interpretation of linear logic, we shall stop here to consider the essence of these constructive models. Readers uncomfortable with categories may prefer to skip this section.

We understand classical propositional logic in terms of the class of its models, namely Boolean algebras. These can be succinctly defined as complemented distributive lattices.

Following Seely [See89] we may define linear logic analogously, with constructivity calling for the substitution of categories for algebras. A *constructive model of linear logic* consists of a *-autonomous category $\mathcal{D}$ with all finite products and a comonad ! with natural isomorphisms $!(A \times B) \cong !A \otimes !B$ and $!1 \cong \top$.

A *-autonomous category [Bar79] is a representably self-dual closed symmetric monoidal category $\mathcal{D}$. Monoidal means that there exists a binary operation (functor) $\otimes : \mathcal{D}^2 \to \mathcal{D}$ and an object $\top$ of $\mathcal{D}$ having $A \otimes (B \otimes C) \cong (A \otimes B) \otimes C$ and $\top \otimes A \cong A \cong A \otimes \top$, satisfying certain *coherence* conditions [Mac71, p.161]. Symmetric monoidal adds $A \otimes B \cong B \otimes A$. Closed means that there exist a functor $\multimap : \mathcal{D}^{\mathrm{op}} \times \mathcal{D} \to \mathcal{D}$ having $A \otimes B \vdash C$ iff $A \vdash B \multimap C$ (the "Deduction Theorem," aka Currying). (We say "having" rather than "satisfying" as a reminder that this is not just a condition but a particular natural bijection.)

Self-dual means that there exists an isomorphism $-^\perp : \mathcal{D} \to \mathcal{D}^{\mathrm{op}}$. Representably self-dual adds the requirement that $\mathcal{D}$ contains an object $\perp$ with which we may define $A^\perp = A \multimap \perp$.

This defines the multiplicative or dynamic part of the generic model of linear logic. The other multiplicative connectives are defined as $\top = \perp^\perp$ and $A \parr B = (A^\perp \otimes B^\perp)^\perp = A^\perp \multimap B$.

For the additive or static part, $\mathcal{D}$ has all finite products, specifically $A \wedge B$ and the empty product 1. We define $A \vee B = (A^\perp \wedge B^\perp)^\perp$ and $0 = 1^\perp$.

Finally, relating the dynamic and static parts, there exists a comonad structure $(!, \epsilon, \delta)$ on $\mathcal{D}$ with natural isomorphisms $!(A \times B) \cong !A \otimes !B$ and $!1 \cong \top$. We may understand this as a cartesian closed category $\mathcal{E}$ and a functor $F : \mathcal{E} \to \mathcal{D}$ with right adjoint U and with $F(A \wedge B) \cong F(A) \otimes F(B)$ and $F(1) \cong \top$. The comonad is then $(FU, \epsilon, F\eta U)$ where η and ϵ are the unit and counit of the adjunction, with the further requirement that $\mathcal{E}$ be recoverable as the Eilenberg-Moore category of this comonad.

We define $?A = (!(A^\perp))^\perp$ and $A \Rightarrow B = !A \multimap B$.

This ends the category theory.

4 Linear Automata and Schedules

We now describe a framework that is a cross between linear algebra and automata theory, namely linear automata and their complementary linear schedules. While this is not exactly quantum mechanics, its associated logic is closer to quantum logic than either classical (Boolean) or intuitionistic (Heyting) logic. And it offers internally consistent accounts of time and measurement. We focus here on the special case of linear automata as state spaces. In the section on measurement we will mention briefly a more general class of linear automata based on partial distributive lattices, to be dealt with in more detail in a forthcoming paper.

4.1 State Spaces and Event Spaces

If one thinks of a finite-dimensional vector space as populated with column or ket vectors it is natural to think of its dual as consisting of row or bra vectors of the same dimension. We define the notions of state space and event space [Pra92c, Pra92a] in which events are to states as bras are to kets. We thereby obtain a discrete yet linear algebra of behaviors.

We read into the duality of events and states a complementarity[3] of time and information [Pra92b] analogous to that of time and energy in mechanics. That our mechanics is more quantum than classical follows from a Heisenberg-like uncertainty principle found naturally, i.e. without special provision, in our model. Such uncertainty is a litmus test for quantum rather than classical behavior in any phase-space-like situation such as the complementarity of position and momentum.

The behavior of an automaton is to alternately wait in a *state* and perform a transition or *event*. We may think of the state as bearing information representing the "knowledge" of the automaton when in that state, and the event as modifying that information. At the same time we may think of the event as taking place at a moment in time, and the state as modifying or whiling away time.

Thus states *bear* information and *change* time, while events *bear* time and *change* information.

We shall express this duality by giving two complementary representations of such information. A state space will be a set of states ordered by information content, while an event space will be a set of events ordered by time.

State spaces generalize bc-domains (Scott domains or bounded-complete algebraic cpo's) [Gun92], while their complementary event spaces generalize Winskel's event structures [Win86]. This particular duality is one small fragment of Birkhoff-Stone duality [Bir33, Sto36, Sto37, Pri70], with the partial distributive lattices alluded to in the section on measurement constituting a much larger fragment.

A state space can be thought of as a representation of behavior somewhere in between a formal language as a set of traces and a conventional state automaton. Loops of the latter are unrolled and confluences are split apart. Forks (branch points) however are left untouched as representing decisions, and a new kind of confluence representing concurrency is introduced. The alphabet of symbols labelling transitions to indicate what they do is omitted, permitting us to focus on the underlying structure of computation.

Formally, a *state space* is a nonempty partially ordered set X every nonempty subset $Y \subseteq X$ of which has a meet or infimum $\bigwedge Y$. We refer to the state $\bigwedge Y$ as the *branch* for Y, the state from which one must make decisions about which portion of Y to narrow down to. Noting that the branch $\bigwedge X$ for the whole space is the least element of X, we call this the *initial state* and abbreviate it to q_0

A state space is interpreted as an automaton by interpreting $x \leq y$ as a transition from x to y, with the least element as the initial state. Its maximal directed sets (those Y for which every pair of elements of Y has an upper bound in Y) correspond to the alternative paths. A state space that is a tree (no confluences) has no concurrency and exhibits purely branching behavior, with a maximal directed set being a chain. Conversely a directed state space has only one alternative, the whole space, and represents purely concurrent behavior (possibly temporally constrained). A finite Boolean algebra with n atoms represents the concurrent and temporally unconstrained execution of n events. A finite distributive lattice generalizes this to events scheduled by a poset, where the events correspond to the prime elements, those not the join of two incomparable elements, and the order on the prime elements gives the temporal constraints. Nondistributive lattices are to be regarded as quotients of distributive lattices, i.e. certain states are identified or synchronized.

[3] We shall see later that the terms dual, complement, and converse form a natural trio with distinct technical meanings.

We think of states as containers of knowledge. The intended interpretation of $x \leq y$ is that y is a refinement of, or addition to, the state of knowledge of x. For example in state x we may know that we are in California and going very fast, while in state $y \geq x$ we may know that we are crossing the Golden Gate at 65 mph.

The knowledge in the branch $\bigwedge Y$ is the disjunction of the knowledge at the states in Y. Equivalently, $\bigwedge Y$ is that state that knows as much as possible without knowing something contradicting some state of Y.

An *event space* is a nonempty partially ordered set X every nonempty subset $Y \subseteq X$ of which has a join or supremum $\bigvee Y$. We refer to $\bigvee Y$ as the *completion* of Y. Noting that the completion $\bigvee X$ for the whole space is the greatest element of X, we call this the *final event* and abbreviate it to ∞

We think of events as markers of time. The intended interpretation of $x \leq y$ is that x precedes or is simultaneous with y.

Just as vector spaces may be transformed with linear transformations that hold the origin fixed and preserve linear combinations, so may state spaces be transformed with state space transformations. A *state map* $f : X \to Y$ is a function between two state spaces that preserves the initial state and all branches. That is, $f(q_0) = q_0$, and $f(\bigwedge Y) = \bigwedge f(Y)$ where $f(Y)$ denotes the image $\{f(y) \mid y \in Y\}$ of Y under f.

The initial state represents absolute rather than relative ignorance. No transformation of a state space can change the status of q_0; in particular no states can be added below or to one side of q_0, only above. We may identify q_0 with the state of the universe at the Big Bang, namely complete ignorance of all facts. (Facts which are always true are deemed to contain zero knowledge for this purpose: if $1 + 1 = 2$ is a universal truth then it was known at the Big Bang.) In q_0 no events have happened yet.

We call the situation $\bigwedge Y = q_0$ a *dilemma* or blind choice. The basic dilemma is  (horn-shaped) where Y consists of the top two elements. We have no information in q_0 yet we are obliged to make a choice there.

The dual notion of *event map* $f : X \to Y$ is a function between two event spaces that preserves the final event and all completions.

The final event ∞ is as absolute as the initial state, for similar reasons. For every state q, ∞ has not happened in q. (Compare this with: for every event u, u has not happened in q_0.) That is, ∞ never happens.

Informally, the time of the completion $\bigvee Y$ is the supremum of the times of completion of the events in Y. Equivalently, $\bigvee Y$ is that event that is as early as possible while still following every event of Y.

We call the situation $\bigvee Y = \infty$ a *conflict*, dual to dilemma. The basic conflict is  with Y the bottom two elements. The completion of Y is the event that never happens, hence not all of Y can complete.

4.2 State Space Interpretation of LL

We return now to linear logic to interpret its propositions and connectives formally for state and event spaces. In the following section we then give the informal behavioral intuition behind this interpretation. This language of sums and products of spaces is a close cousin of Birkhoff's generalize arithmetic of posets [Bir42], elaborated on elsewhere [Pra92a].

The *dual* $A^\perp$ of a state space A is obtained as the order dual of $A - \{q_0\}$ with q_0 then added back in at the bottom. That is, holding q_0 fixed, turn the rest of A upside down. This operation can be seen to interchange  and  , for example. (These are Hasse diagrams of state spaces, with q_0 at the bottom.)

To show $A^\perp$ is a state space, let Y be $A^\perp - \{q_0\}$. It suffices to show that any nonempty subset $Z \subseteq Y$ has a meet in $A^\perp$. Now if Z has no lower bound in Y then q_0 is the meet of Z in $A^\perp$. Otherwise let W be the set of lower bounds on Z in Y. In A, Z is a set of lower bounds on W, hence $\bigwedge W$ in A is an upper bound on Z in A, therefore not equal to q_0, therefore the join of W in Z. This join is at once a lower bound on Z and dominates W, hence it is the meet of Z.

We define the *converse* $A^\smile$ of a state space to be its order dual, an event space. We define the *complement* A^- to be $A^{\perp\smile}$, the converse of the dual of A, an event space. The complement can be obtained simply by removing the initial state q_0 and reinstalling it as the final event ∞. The complement of an event space is the state space obtained via the inverse procedure: move top to bottom.

The complement of a state space turns states into their corresponding events, turning the dilemma  into the conflict  , the *concurrence*  into  , and the *guarded choice*  into  .

Static conjunction $A \wedge B$ is product of state spaces viewed as posets: form the cartesian product of their underlying sets and then order $(x, y) \leq (x', y')$ just when $x \leq x'$ and $y \leq y'$. It is straightforward to verify that $A \wedge B$ so defined is a state space. Define $A \vee B$ by De Morgan's law to be $(A^\perp \wedge B^\perp)^\perp$. The one-state space serves as unit for both static connectives.

We write $A \multimap B$ for the set of state maps from A to B. We partially order $A \multimap B$ pointwise: $f \leq g$ in $A \multimap B$ just when $f(x) \leq g(x)$ for all x in A. To see that $A \multimap B$ is a state space it suffices to show that the pointwise meet $\bigwedge Z$ of any nonempty set Z of maps in $A \multimap B$ is a state map, accomplished as follows.

$$\begin{aligned}
(\textstyle\bigwedge Z)(q_0) &= \bigwedge_{f \in Z} f(q_0) \\
&= \bigwedge_{f \in Z} q_0 \\
&= q_0
\end{aligned}$$

$$\begin{aligned}
(\textstyle\bigwedge Z)(\bigwedge Y) &= \bigwedge_{f \in Z} f(\bigwedge Y) \\
&= \bigwedge_{f \in Z} \bigwedge_{y \in Y} f(y) \\
&= \bigwedge_{y \in Y} \bigwedge_{f \in Z} f(y) \\
&= \bigwedge_{y \in Y} (\bigwedge Z)(y) \\
&= \textstyle\bigwedge((\bigwedge Z)(Y)) \quad \blacksquare
\end{aligned}$$

We define $A \otimes B = (A \multimap B^\perp)^\perp$, and define $A \parr B$ by De Morgan's law to be $(A^\perp \otimes B^\perp)^\perp$. The 2-element state space  serves as the common unit for both dynamic connectives.

We now define $!A$. An order filter Y of A is an upward-closed subset of A, one such that if $x \leq y$ then $x \in Y$ implies $y \in Y$. The order filters on A ordered by inclusion form a complete lattice and hence a state space, the dual of which we define to be $!A$. Those elements of $!A$ (construed as order filters of A) maximal in $!A$ with respect to containing a given element x of A form a poset isomorphic to the underlying poset $U(A)$ of A.

We call $!A$ the *free state space on* $U(A)$. $!A$ has the property that for any state space B, the state maps from $!A$ to B, when restricted to A, are exactly the poset (i.e. monotone) maps from $U(A)$ to $U(B)$. Thus whereas $A \multimap B$ is the state space of all state maps from A to B, $!A \multimap B$ is the larger state space of all poset maps from A to B. The effect of ! has been to strip off the structure of A to permit a clearer view, as discussed earlier and again below in the section on measurement.

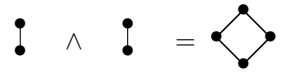
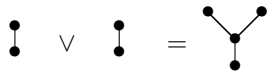
Then $?A$ is $(!(A^\perp))^\perp$, and $A \Rightarrow B$ is $!A \multimap B$, as usual.

4.3 Behavioral Interpretation

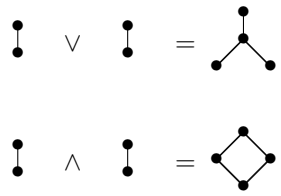
The state space interpretations of the connectives of linear logic suggest natural behavioral interpretations.

We give the state space perspective for the static operations, then pass to event spaces for the rest. The meaning of any connective in state spaces is just the De Morgan dual of the corresponding connective for event spaces and vice versa.

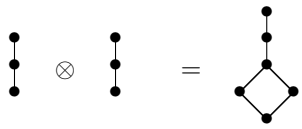
Recall that we defined $A \wedge B$ as cartesian product: this corresponds to the usual definition from automata theory of the concurrent automaton $A \| B$ (the automaton interleaving the traces of A and B) as the product of

the two automata, with each transition from (a, b) to (a', b) corresponding to a transition of A and from (a, b) to (a, b') a transition of B. We illustrate this with the example of two one-transition automata running concurrently to yield the product automaton  . The dual operation $A \vee B$ is *guarded choice* of A and B. The choice of two one-transition automata is illustrated by  , showing the introduced guard state. $A \vee B$ can be thought of as A and B juxtaposed, but with their initial states identified and additional branches (decision states, just one in the preceding example) filled in as needed for all subsets of the juxtaposition containing states of both A and B (a sort of free completion to a state space). The filling-in creates the "guard" in the form of branches for the choice.

The event space operations for concurrence and choice are respectively $A \vee B$ and $A \wedge B$, dual to the state space connectives. Here the complementary examples are as follows.

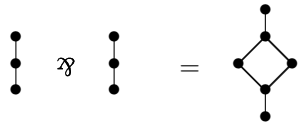


We now shift to the event space perspective for the remaining connectives. $A \otimes B$ is the *flow* of A and B, the idea of one flowing through the other. For example two consecutive trains and two consecutive stations flow through each other as follows:



creating a square schedule of 4 arrivals of trains at stations, with the arrival of the first train at the second station concurrent (but not necessarily simultaneous) with the arrival of the second train at the first station. (Only the bottom three events and the second from the top are physical events, the other two are a join and ∞.)

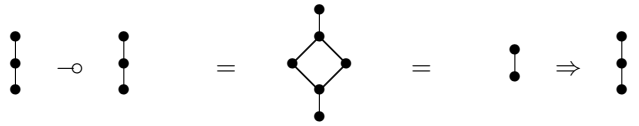
The dual connective, illustrated as follows, is $A \parr B$:



We call this *mingle*, but we do not have a simple familiar example of it. The common unit of both connectives is  .

The behavioral interpretation of the remaining connectives, still for event spaces, all involve the notion of measurement. This can be read either before or after the following section on measurement: each helps to reinforce the other.

We read $A \multimap B$ as A *observed-by* B, illustrated by the left equation in the diagram below. This is peer observation, thought of as B unable to see A clearly due to A's "structure veil."



In contrast $!A \multimap B$ or $A \Rightarrow B$ is *A diagnosed-by B*: B observing A rendered transparent by !, as illustrated by the equation on the right in the diagram above. Here $!A$ is the *skeleton* or underlying structure of A, put back in the physical world in the form of the free event space on that underlying structure: $!A = F(U(A))$. The idea of underlying structure comes across much more clearly in this constructive model than in the nonconstructive phase space model.

The identity $A \multimap (B \multimap C) \cong (A \otimes B) \multimap C$ identifies "while observing B, C observes A" with "C observes A flowing through B." So the effect of A drifting into C's field of view while observing B is for A to flow through B to put them both in the field of view together, interacting strongly.

The identity $A \multimap B \cong B^{\perp} \multimap A^{\perp}$ might be the equivalence of B's acquiring position information from A with A's acquiring momentum from B.

The identity $A \Rightarrow (B \Rightarrow C) \cong (A \wedge B) \Rightarrow C$ illustrates the difference superiority makes in observing. "While diagnosing B, C diagnoses A" is equivalent to "C diagnoses one of A or B." With $\multimap$, the observer had to settle for observing an intimate mixing of the two objects, making it hard to resolve them. With $\Rightarrow$ the observer has no difficulty distinguishing A from B and gets to choose which of A or B to diagnose. Noting that $(A \wedge B) \Rightarrow C \cong !(A \wedge B) \multimap C \cong (!A \otimes !B) \multimap C$, we may think of ! as freezing A and B before running them together; in their frozen state they do not mix, making it easier for C not only to tell them apart but to choose which to observe.

5 Measurement

We now draw attention to a feature of a generalization of the linear automaton model, namely a naturally arising Heisenberg uncertainty principle. Such a principle is usually associated with the Fourier transform, which transforms a sharply defined value in one domain, represented by a delta function, to a diffuse spread of values in the complementary domain. Our analysis finds this uncertainty in a wider range of phenomena, the essential common feature of which is duality. The Fourier transform is associated with Pontrjagin duality, but one can also find signs of it in Stone duality and other dualities, as we shall do here.

By "certainty" we have in mind precision, which we correlate with variety of possible measurements. A certainty or precision of n bits corresponds to having 2^n possible measurements; conversely the existence of m distinguishable measurements is associated with a precision of $\log_2 m$ bits.

We define a *measurement* to be a function $f : A \to B$ where A is the measured object and B the measurement alphabet. For example A may be the set of pixels on your computer screen, B the set of possible colors of each pixel, and $f : A \to B$ one of the $|B|^{|A|}$ possible screen images, $2^{2^{20}}$ in the case of a monochrome 1K×1K screen.

Such a screen is specified to a precision of 2^{20} bits, one bit per pixel. If $|B| = 256$, corresponding to 8 bits per pixel, the precision would become 2^{23} bits.

But now suppose that for some reason not every screen can occur, for example if every white pixel is required to have at least four white pixels touching it along an edge or at a corner. The precision then drops, that is, uncertainty increases.

A natural kind of restricted function is the *homomorphism.* This is a function between two similar algebras, e.g. two groups, that preserves structure, e.g. $f(x + y) = f(x) + f(y)$, $f(x^{-1}) = f(x)^{-1}$, etc. For example there are four functions from the two-element group C_2 to itself, only two of which are group homomorphisms.

In this last example, the passage from functions to group homomorphisms is associated with an increase of one bit in uncertainty. It is as though the effect of group structure were a veil hiding part of the group.

State spaces and their dual event spaces are too small a class to exhibit more than the special case of Heisenberg uncertainty, where both the automaton and its dual appear equally uncertain. In a separate paper we will describe a uniform generalization of state and event spaces to a single category PDL of partial distributive lattices. Informally a PDL is a distributive lattice where any given meet or join may or may not be defined. Maps of such preserve those meets and joins that exist. This provides a uniform framework embedding both state and event spaces in the one category, along with other structures such as sets, posets, semilattices, complete semilattices, distributive lattices, and Boolean algebras. A state space has all nonempty meets and the empty join, and no nonempty joins or the empty meet. An event space has the exact opposite. A set has no meets or joins save the trivial one-element ones: $\bigvee\{x\} = \bigwedge\{x\} = x$. At the other extreme a complete atomic Boolean algebra (CABA) has all meets and joins.

The original motivation for this extension was to accommodate behaviors that are recognized by the concurrency community as important but that state spaces cannot represent. PDL's turn out to have a natural characterization as a programming language of all monotone Boolean operations.

Let us first consider sets and CABA's. We interpret a set as a schedule having no structure, i.e. no veil, with every event clearly distinguishable and constituting an independent primitive event. Call this the *discrete* or *purely conjunctive* schedule, all of whose events must happen. When we observe such a schedule to see which events have happened so far it is possible to obtain any of the 2^n possible outcomes. Thus we associate n bits of observable information with the n-event discrete schedule.

The dual of an n-element set is an n-atom CABA. As with any PDL we can view it as either a schedule or an automaton (not with the same behavior however). Viewed as an automaton this is just the automaton complementary to our purely conjunctive schedule but with time reversed, corrected by taking its converse (recall that complement is composition of converse with dual, $A^- = A^{\perp\smile}$).

There are only n Boolean algebra homomorphisms from it to the 2-element CABA preserving all meets and joins, namely the n projections onto one of the n dimension. This corresponds to choosing exactly one state, as an event. Furthermore, ordered pointwise, the set of these events is discretely ordered, showing that we have faithfully recovered the dual event space (so the uncertainty of the complement of an object can be determined directly from the object's cardinality).

We thus see that there are n bits of information visible in the discrete schedule but only $\log_2 log_2 N$ bits visible in the N-state automaton dual to it. This is as indistinct as any N-state automaton can get.

The same calculations show that the N-element CABA, as a schedule, is a maximally veiled schedule, and its complementary automaton is perfectly focused. This is the *purely disjunctive* behavior: choose a state and stay there, the automaton having no transitions.

As sets start to acquire meets or joins, their dual Boolean algebras concomitantly lose meets and joins. This corresponds to the "veil" gradually shifting from the CABA side to the set side. Event and state spaces are in the exact middle of this tradeoff: every schedule has the cardinality of its complementary automaton, hence two such complementary N-element views of behavior share their $log_2 N$ bits equally between them. The Fourier analysis equivalent of this symmetry is obtained for white noise, where both the signal and its dual spectrum are (approximately?) normally distributed.

Chains (linear orders) represent rigid (no branching) local (no concurrency) behavior and form a natural subcategory of event spaces where the triumvirate of duality, complementarity, and converse can be studied closely yet painlessly [Pra92b]. An n-event linear schedule is quite indistinct, having only $log_2 n$ bits, as does its complementary $(n \pm 1)$-state linear automaton (the ± 1 depending on which bounds are coded in the schedule). In effect all we can observe of a sequential schedule is which transition between two events we are presently in, the rest of the schedule before and after is invisible to mortals.

The role of $!A$ in these models of behavior is, as we said, to pull the structure veil aside to reveal the inner workings (underlying poset) of A, permitting a greater variety of measurements (more precision of observation) than with the veil in place. If $!A$ is chosen to be the underlying poset then for chains with say a top we gain only one additional observation, the one that does not preserve that top. Hence this $!A$ for chains does not clarify the view appreciably.

However by removing yet more structure, corresponding in this case to having a different $!A$ operation yielding the underlying *set*, we can then see the whole linear schedule in sharp relief. This illustrates that structure need not be monolithic: the doctor can undress you for a first examination, then use the X-ray machine to see through your skin to obtain even sharper detail.

Since $\perp$ is discrete (two elements), Planck's constant $\hbar$ has not entered. If it had, we would construe $!A$ as shrinking $\hbar$ (locally to A) to zero to make A behave classically.

6 Conclusion

We have extended Birkhoff and von Neumann's propositional quantum logic of uncertain states to a logic of the duality of time and information. Its joint programming language and verification logic is linear logic. More behavioral phenomena are captured by this linear logic of behavior than by quantum logic alone. On the other

hand the extension is not faithful to the orthodox view of quantum mechanical dynamics, and we raise as an open problem how to formulate the dynamic quantum logic of standard quantum mechanics.

One might ask why the well-behaved static implication $A \Rightarrow B$ of linear logic had not previously been found for quantum logic. The answer seems to hinge on the need for static implication to refer to the dynamic part, $A \Rightarrow B = !A \multimap B$, even when static conjunction does not. That negation is obtained via dynamic implication, $A^\perp = A \multimap \perp$, and static disjunction is in turn obtained as the De Morgan dual of static conjunction via that negation, indicates that these too are tied to the dynamic part. This suggests that the root of quantum logic's implication problem might lie with not bringing time into the logic.

Acknowledgement. I am very grateful to John Baez for his generously given time and physics advice.

References

[AA66] I. Amemiya and H. Araki. A remark on Piron's paper. *Pubs. of the Res. Inst. of Math. Sciences, Kyoto University Series A2*, 2:423–427, 1966.

[Bar79] M. Barr. $*$-*Autonomous categories, LNM 752*. Springer-Verlag, 1979.

[Bir33] G. Birkhoff. On the combination of subalgebras. *Proc. Cambridge Phil. Soc*, 29:441–464, 1933.

[Bir42] G. Birkhoff. Generalized arithmetic. *Duke Mathematical Journal*, 9(2), June 1942.

[Bir67] G. Birkhoff. *Lattice Theory*, volume 25. A.M.S. Colloq. Publications, 1967.

[BvN36] G. Birkhoff and J. von Neumann. The logic of quantum mechanics. *Annals of Mathematics*, 37:823–843, 1936.

[CCMP91] R.T Casley, R.F. Crew, J. Meseguer, and V.R. Pratt. Temporal structures. *Math. Structures in Comp. Sci.*, 1(2):179–213, July 1991.

[Dal86] M.L. Dalla Chiara. Quantum logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume III, pages 427–469. Reidel, Dordrecht, 1986.

[dBdR72] J.W. de Bakker and W.P. de Roever. A calculus for recursive program schemes. In M. Nivat, editor, *Automata, Languages and Programming*, pages 167–196. North Holland, 1972.

[Dij76] E.W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, Englewood Cliffs, N.J., 1976.

[Fin70] P.D. Finch. Quantum logic as an implication algebra. *Bull. Aust. Math. Soc.*, 1:101–106, 1970.

[Flo67] R.W. Floyd. Assigning meanings to programs. In J.T Schwartz, editor, *Mathematical Aspects of Computer Science*, pages 19–32, 1967.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Gol74] Robert Goldblatt. Semantic analysis of orthologic. *J. Philosophical Logic*, 3:19–35, 1974.

[Gol84] Robert Goldblatt. Orthomodularity is not elementary. *J. Symbolic Logic*, 49:401–404, 1984.

[Gun92] C.A. Gunter. *Semantics of Programming Languages*. MIT Press, 1992.

[Hoa69] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12:576–580, 1969.

[Mac71] S. Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, 1971.

[Ng84] K.C. Ng. *Relation Algebras with Transitive Closure*. PhD thesis, University of California, Berkeley, 1984. 157+iv pp.

[NT77] K.C. Ng and A. Tarski. Relation algebras with transitive closure, Abstract 742-02-09. *Notices Amer. Math. Soc.*, 24:A29–A30, 1977.

[Pav92] M. Pavičić. Bibliography on quantum logics and related structures. *Int. J. of Theoretical Physics*, 31(3):373–460, 1992.

[Pnu77] A. Pnueli. The temporal logic of programs. In *18th IEEE Symposium on Foundations of Computer Science*, pages 46–57, October 1977.

[Pra76] V.R. Pratt. Semantical considerations on Floyd-Hoare logic. In *Proc. 17th Ann. IEEE Symp. on Foundations of Comp. Sci.*, pages 109–121, October 1976.

[Pra80] V.R. Pratt. Dynamic algebras and the nature of induction. In *12th ACM Symposium on Theory of Computation*, Los Angeles, April 1980.

[Pra92a] V.R. Pratt. Arithmetic + logic + geometry = concurrency. In *Proc. First Latin American Symposium on Theoretical Informatics, LNCS 583*, pages 430–447, São Paulo, Brazil, April 1992. Springer-Verlag.

[Pra92b] V.R. Pratt. The duality of time and information. In *Proc. of CONCUR'92, LNCS 630*, pages 237–253, Stonybrook, New York, August 1992. Springer-Verlag.

[Pra92c] V.R. Pratt. Event spaces and their linear logic. In *AMAST'91: Algegraic Methodology and Software Technology*, Workshops in Computing, pages 1–23, Iowa City, 1992. Springer-Verlag.

[Pri70] H.A. Priestley. Representation of distributive lattices. *Bull. London Math. Soc.*, 2:186–190, 1970.

[RR91] L. Román and B. Rumbos. Quantum logic revisited. *Foundations of Physics*, 21(6):727–734, 1991.

[Sal70] A. Salwicki. Formalized algorithmic languages. *Bull. Acad. Pol. Sci., Ser. Sci. Math. Astr. Phys.*, 18(5), 1970.

[See89] R.A.G Seely. Linear logic, $*$-autonomous categories and cofree algebras. In *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 371–382, held June 1987, Boulder, Colorado, 1989.

[Sto36] M. Stone. The theory of representations for Boolean algebras. *Trans. Amer. Math. Soc.*, 40:37–111, 1936.

[Sto37] M. Stone. Topological representations of distributive lattices and brouwerian logics. *Časopis Pěst. Math.*, 67:1–25, 1937.

[Var68] V.S. Varadarajan. *Geometry of Quantum Theory.* Springer-Verlag, second edition, 1968.

[Win86] G. Winskel. Event structures. In *Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets 1986, LNCS 255*, Bad-Honnef, September 1986. Springer-Verlag.