Quantum Computing: A new Paradigm and it's Type Theory

Martin Wehr

August 12, 1996

Abstract

To use quantum mechanical behavior for computing has been proposed by Feynman. Shor gave an algorithm for the quantum computer which raised a big stream of research. This was because Shor's algorithm did reduce the yet assumed exponential complexity of the security relevant factorization problem, to a quadratic complexity if quantum computed.

In the paper a short introduction to quantum mechanics can be found in the appendix. With this material the operation of the quantum computer, and the ideas of quantum logic will be explained. The focus will be the argument that a connection of quantum logic and linear logic is the right type theory for quantum computing. These ideas are inspired by Vaughan Pratt's view that the intuitionistic formulas argue about states (i.e physical quantum states) and linear formulas argue about state transformations (i.e computation steps).

1 Introduction

A calculus for programs on quantum computers is strongly missed. Here we present the material to build such a calculus. The approach goes along the lines of typed programming languages.

In the first section the propositions-as-types paradigm is explained. This idea will be applied to model the quantum parallel computation in linear logic. The notion of quantum concurrency is explained in section 2. Parallel computations are constructed by two kind of operations measurements and reversible computations. The measurement process in a quantum system is an interaction. The measurement clearly extracts some information but it also affects the system. So measurements do input/output of quantum computing. The inherently probabilistic nature of measurements is explained by quantum logic in section 4.

The quantum computations are done reversible so they can be implemented deterministic. In section 5 the notions for the mathematical semantic of the reversible computations is explained. The treatment is grounded on Vaughan Pratt's work [Pra93, Pra94a] as a strong interconnection of Chu spaces, linear logic and concurrency. This leads to a proposal for a Chu space which models the behavior of quantum computation.

2 Type Theory

To go a step further in reliability of software one approach is to treat programming languages formal. This should lead to logical and mathematical understood language constructs. One very fruitful source in formal treatment of programming languages is the Curry-Howard correspondence. The idea of the correspondence is expressed in its synonymous term propositions-as-types-paradigm. The observation of Curry was a striking analogy between the theory of functionality and the theory of implication.

Let's present the theory of implication (also known as minimal logic) in natural deduction style as introduced by Martin Löf [NPS90]. The judgments of the theory have form $\Gamma \vdash A$, expressing that from hypotheses Γ formula A can be derived, where hypotheses Γ is a set of formulas. In natural deduction style each logical connective (here implication $\rightarrow$ is the only one) comes with two groups of rules; the introduction rules and the elimination rules.

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \; (\to I) \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B} \; (\to E)$$

$$\Gamma, A \vdash A \; (TAUT)$$

The rules above are well known logical principles. For instance ($\to E$) is the modus ponens and ($\to I$) is the deduction theorem. The correspondence interprets every logical formula as a type. Here $A \to B$ is the type of functions mapping values of type A to results of type B. Further to every rule of the system corresponds a construct of a program calculus. In our example the minimal logic corresponds to the simply typed lambda calculus. The judgment $\Gamma \vdash e : A$ expresses now that program e is well typed under the typotheses Γ. Here Γ is a set of typing assumptions $x : B$ expressing that variable x has type B and can appear in program e.

$$\begin{array}{lcl} (abstraction) & \dfrac{\Gamma, x : A \vdash e : B}{\Gamma \vdash \lambda x.e : A \to B} & (\to I) \\ (application) & \dfrac{\Gamma \vdash e : A \to B \qquad \Gamma \vdash e' : A}{\Gamma \vdash ee' : B} & (\to E) \\ (var) & \Gamma, x : A \vdash x : A & (TAUT) \end{array}$$

Finally logical derivations become well typed programs. This setting suggests to take as mathematical semantics the set theory. So implication $A \to B$ will be interpreted as the set of all maps from set A to set B, the typing relation $e : A$ becomes the element relation $\in$ in set theory. It was Milners idea in [Mil78] to observe, if we insert in every set the element $Wrong$ and interpret impossible operations (for instance application to a constant $1e$) as $Wrong$, then the principle of type theory *well typed programs can't go wrong* can be formal proved. So type theory takes all grammatical possible programs (here $e ::= x \mid \lambda x.e \mid ee'$) and carves out the well typed programs which are the ones with a correctness property.

3 Quantum Computing

The difference between classical and quantum computing can be explained by the representation of the information and the possible operations, see for instance [Eke94, Sho94, Bra96].

A classical bit can be only in two states 0 or 1, whereas a quantum bit can be in the superposition of the two states $|0\rangle$ and $|1\rangle$. Quantum mechanics relates the outcomes of an experiment to a base of a Hilbert space. In case of discrete outcomes (here outcomes should be 0 or 1) every outcome is related to a vector of an orthonormal base of the Hilbert space. A (pure) state of the system is then represented as a vector of norm 1 in the Hilbert space. So an information which can be classical represented in a n bit register, needs $2^n - 1$ complex numbers to be fully described as a quantum state.

There are two possible operations in quantum computing; measurement (which corresponds to projections in Hilbert space) and unitary transformations. Projections are used for input/output in quantum computing algorithms. Unitary transformations are inner product preserving linear maps in Hilbert spaces, so they transform physical states. The unitary transformation do the computations which are reversible.

The general model for representation of concurrency are transition systems [WN95]. So a quantum computer can be represented as a probabilistic transition system.

states	$\psi \in \mathbb{H}$ with $\|\|\psi\|\| = 1$	normed vectors of the space
actions	$P \in \mathbb{P}(\mathbb{H})$	projections
	$U \in \mathbb{U}(\mathbb{H})$	unitary transformations
transitions	$\psi \xrightarrow{P} \widehat{P\psi}$	with probability $\langle \psi, P\psi \rangle$
	$\psi \xrightarrow{P} \widehat{\psi - P\psi}$	with probability $1 - \langle \psi, P\psi \rangle$
	$\psi \xrightarrow{U} U\psi$	with probability 1

where operation $\hat{}: \mathbb{H} \to \mathbb{H}$ sends a non-zero vector to a unit vector by the map $\psi \mapsto \frac{\psi}{||\psi||}$.

4 The Static Part: Quantum Logic

Here we explain in logical terms the input/output operations in quantum computing. The theory was founded by Birkhoff from the logical side and von Neumann from the operator side, good accounts are [Var85, BC81, Coh89].

A partial order is given by the tuple $(L, \sqsubseteq)$ where L is a set and $\sqsubseteq$ is a binary relation on the set. The relation $\sqsubseteq$ must be reflexive and transitive. The dual order $(L, \sqsupseteq)$ is given by transposing the relation. A logic with L the set of formulas and $\vdash$ the inference relation is an example for a partial order. A partial order satisfying antisymmetry $a \sqsubseteq b \wedge b \sqsubseteq a \implies a = b$ is called a preorder. The unit 1 is given by $a \sqsubseteq 1$ for all elements a. The dual is the zero 0. Taking the logical equivalent formulas of a logic we get an example for a preorder; this construction is called the Lindenbaum algebra of a logic. In case of a logic the unit is $true$ and the zero is $false$.

The greatest lower bound of a, b is called the meet written $a \sqcap b$. The dual notion is the join $a \sqcup b$. A lattice $(L, \leq, \sqcap, \sqcup)$ is a preorder which is closed under the operation of meet and join. Given a set V the power set $(\mathcal{P}(V), \subseteq, \cap, \cup)$ is an example for a lattice. A distributive lattice is satisfying the equation $a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c)$ and also if $\sqcup$ and $\sqcap$ are exchanged. A triple (a, b, c) satisfying the equations above is called a distributive triple.

A orthocomplementation on a lattice is given by the operation $\_^{\perp} : L \to L$ with:

$$\begin{array}{ll}
\textbf{(involution)} & a^{\perp\perp} = a \\
\textbf{(contravariance)} & a \sqsubseteq b \implies b^{\perp} \sqsubseteq a^{\perp} \\
\textbf{(complement)} & a \sqcup a^{\perp} = 1
\end{array}$$

A distributive orthocomplemented lattice is called a boolean algebra. The Lindenbaum algebra of classical proposition logic is an example for a boolean lattice with logical negation $\neg$ interpreted as orthocomplementation.

A quantum logic is defined by weakening a boolean algebra. A quantum logic is a orthocomplemented lattice satisfying the orthomodular identity:

$$\text{if } a \sqsubseteq b \text{ then } b = a \sqcup (b \sqcap a^{\perp})$$

The subspaces of a Hilbert space are an example for a quantum logic. In this case $\sqsubseteq$ will be modeled by the subspace relation, $\sqcap$ is the set theoretic intersection, $a \sqcup b$ is the span generated by the subspaces a and b, and $a^{\perp}$ is the orthogonal subspace to a.

Now we are given the axiomatic setting of quantum logic. In the following we want to reconstruct from this setting the main notions of Hilbert spaces and quantum theory. This construction is similar to the stone duality [Var85] which reconstructs classical logic from its models.

Two elements of a quantum logic are orthogonal $a \perp b$ iff $a \sqsubseteq b^{\perp}$. We define sum and subtraction in quantum logic as $a + b = a \sqcup b$ iff $a \perp b$ and $a - b = a \sqcap b^{\perp}$. Now we rephrase orthomodular identity in a more familiar form $if\ a \sqsubseteq b\ then\ b = a + (b - a)$.

One of the main points of quantum theory is that the measures affect the state of the system. If a set of measures lead to the same final state if all measures are applied in any order, these measures are called compatible. This physically motivated notion is reflected in quantum logic. Two proposition are called compatible $(a, b)C$ if one of the following equivalences holds :

$$\begin{array}{ll}
\text{iff} & \exists a', b', c \text{ with } a = a' + c \text{ and } b = b' + c \\
\text{iff} & \text{representation of } a, b \text{ as above with} \\
 & a' = a \sqcap b^{\perp} \qquad b' = b \sqcap a^{\perp} \qquad c = a \sqcap b \\
\text{iff} & \begin{array}{l} a = (a \sqcap b^{\perp}) \sqcup (a \sqcap b) \\ b = (b \sqcap a^{\perp}) \sqcup (b \sqcap a) \end{array}
\end{array}$$

Further a triple (a, b, c) is distributive if there are two commutative tuples in it. Given a set of commutative propositions with uses of the connectives $\sqcup$ and $\sqcap$ we can construct a sublattice of the given quantum logic. This sublattice is distributive so it is a boolean algebra. So the commutative propositions show classical behavior.

4.1 States on a logic

A *state* on a quantum logic L is defined as a map s form L to the real interval $[0, 1]$. Further if $a \perp b$ then $s(a \sqcup b) = s(a) + s(b)$ and $s(1) = 1$. So a state assigns to each proposition the probability to be true. A state s is a dispersion free or classical state if $s(a) = 0$ or $s(a) = 1$ for every proposition a. So a classical state is determined with no probabilities. A set of reals $\{\alpha_1, \ldots, \alpha_k\}$ with $0 < \alpha_i < 1$ and $\sum_{i=1}^{k} \alpha_i = 1$ is called a set of convex coefficients. For convex coefficients α_i and $\{s_1, \ldots, s_k\}$ a set of states, the convex combination $s(a) = \sum_{i=1}^{k} \alpha_i s_i(a)$ is a state. The set of states on a quantum logic is closed under convex combinations. A pure state is a state which can't be represented as as a non-trivial convex combination. This leads to the core of quantum theory *in quantum logic exist pure states with dispersion.* This is opposed to the western traditional sight on nature. If we observe uncertainty in outcome prediction of an experiment we assume a lack of knowledge about the system. This kind of uncertainty is called epistemic. Now quantum theory states a kind of uncertainty which is grounded in nature; ontological.

Let Ω be the set of states. We define the *state space* M by

$$M = \{\sum \lambda_k s_k \mid \lambda_k \in \mathbb{R},\ s_k \in \Omega \text{ for } k = 1, \ldots, n\}$$

Let further M^* be the dual space to M, that is the set of bounded linear functionals on M. Given a Hilbert space $\mathbb{H}$ lets take in the following the projections $\mathbb{P}(\mathbb{H})$ as our quantum logic. A vector state on this projection logic is defined by

$$s_x : \mathbb{P}(\mathbb{H}) \to \mathbb{R}$$
$$s_x(P) = \langle Px, x\rangle \text{ for } P \in \mathbb{P}(\mathbb{H}) \text{ and } x \in \mathbb{H} \text{ a unit vector}$$

A mixture $\langle y\rangle$ is given by a set $\{t_1, \ldots, t_n\}$ of convex coefficients and a orthonormal sequence $\langle y_k\rangle$ of vectors. A mixture state $s_{\langle y\rangle}$ is defined by $s_{\langle y\rangle} = \sum t_k s_{y_k}$.

Gleason Theorem
For every state s on the projection logic L there exists a mixture $\langle y\rangle$ with $s = s_{\langle y\rangle}$. If further $f \in M^*$ there exists a unique hermitian operator A with $f(s) = \sum t_k \langle Ay_k, y_k\rangle$

So Hilbert spaces are the right model for quantum logic where states can be represented by mixtures and observables can be represented by hermitian operators.

5 The Dynamic Part: Linear Logic

Here we try to find a mathematical frame to model the computational side of quantum computing. In this section three notions will get connected by the mathematical frame of Chu spaces. The three notions are unitary transformations, concurrency calculli and linear logic. Unitary transformations are the innerproduct preserving operators in Hilbert spaces. So the physical interpretation of quantum theory let's see them as transition probability preserving maps on states. Linear logic models resource consciousness [Wad93, Gir95, Ale94]. The central connective of linear logic is linear implications $A \multimap B$ which can be constructively interpreted as functions which use their argument exactly once. No copying and no loss of input information is possible, this is the property of the mapping of orthonormal bases corresponding to a unitary transformation. Further the constructive intend of linear logic in the sense of the Curry-Howard isomorphism is concurrency as found by Samson Abramsky. We start now in making these ideas formal by developing this in the frame of Chu spaces.

5.1 Chu spaces

Chu spaces are the mathematical modeling of the idea of duality [Pra94b, Pra95a, Pra95b, Gup94, Gor95]. It can be seen as the abstraction of the stone duality for classical logic. The stone duality observes that boolean algebras and its valuations can be modeled by a set system. Either boolean formulas are the elements and valuations the sets or dually the other way. A Chu space over K is a triple $\mathcal{A} = (X, \models, A)$ where K is a set and $\models$ is a valued relation $\models: X \times A \to K$. The Chu_2 spaces where $K = \{0,1\}$ can be seen as logics A with its valuations X. In Chu_2 we write $x \models a$ for $\models (x,a) = 1$. The statement $x \models a$ can be interpreted as "state x fulfills proposition a".

In the following we will see some constructions on Chu spaces, which form a model for linear logic, and can be seen as language constructs on a concurrency calculus. Finally we give two proposals how to embed quantum computing in Chu spaces. There is obvious a lot of insights fitting well together, but still left the full picture to be worked out.

The dual of a Chu space $\mathcal{A} = (X, \models, A)$ is simply its transpose $\mathcal{A}^\perp = (A, \models^\smile, X)$ where the transpose on binary relations $R \subseteq X \times Y$ is given by $R^\smile = \{(y,x) \mid (x,y) \in R\}$.

Let's assume in the following given Chu spaces $\mathcal{A} = (X, \models_A, A)$ and $\mathcal{B} = (Y, \models_B, B)$. Chu transforms are the structure preserving maps. A Chu transform $(f,g) : \mathcal{A} \to \mathcal{B}$ is defined by two maps $f : a \to B$, $g : Y \to X$ satisfying the adjointness condition for all $a \in A$ and all $y \in Y \quad g(y) \models_A a$ iff $y \models_B f(a)$. Composition is defined as $(f,g)(f',g') = (f'f, gg')$ and identity is given by (id_A, id_X). So Chu spaces as objects and Chu transforms as morphisms form a category.

The tensor product $\mathcal{A} \otimes \mathcal{B}$ is defined as $(A \times B, \models, Z)$ where $z \in Z$ iff $z \subseteq A \times B$ and for every $b \in B\ \exists x \in X\ \forall a \in A\ (a,b) \in z \implies x \models a$ and also for every $a \in A\ \exists y \in Y\ \forall b \in B\ (a,b) \in Z \implies y \models b$. That is z is a relation such that every row of z can be aproximated by a $x \in X$ and every column of z can be aproximated by a $y \in Y$.

The linear implication $\mathcal{A} \multimap \mathcal{B}$ is the internalization of Chu transforms. We define $\mathcal{A} \multimap \mathcal{B} = (A \times Y, \models, A \to B)$ where $A \to B$ represents the set of Chu transforms (f,g) and we define $(f,g) \models (a,y) = g(y) \models_A a = y \models_B f(a)$. With use of the dual $\_^\perp$ we construct "par" as the de Morgan dual of the tensor $A \invamp B = (A^\perp \otimes B^\perp)^\perp$. The logical equations which hold in linear logic can be proved to be isomorphisms, for instance the "Currying" $(A \otimes B) \multimap C \cong A \multimap (B \multimap C)$.

Altogether the category of Chu spaces forms a $*$-autonomous category [IM93, Abr93b, AGN, Bie95, Pav94]. These structure has been identified as model for linear logic. Linear logic has been identified as type theory for concurrency by Abramsky [Abr93a, BS94, GN95, Gay95]. Here we model concurrency in the frame of Chu spaces and use the logical connections as language constructs

5.2 Behavioral interpretation

We want to interpret a Chu space $\mathcal{A} = (X, \models, A)$ as a process where X should denote states and A should denote events. The judgment $x \models a$ will be interpreted that in state x the event a has happened. The set X and A have to have some structure to be interpreted in that way. A *state space* is a partial order $(X, \leq)$ of states. For every nonempty subset $Y \subseteq X$ should exist the meet $\bigwedge Y$ and this state is called the *branch* of Y. The state $\bigwedge X$ is called the initial state q_0. The relation $x \leq y$ will be seen as transition from state x to y.

A *event space* A is defined dual to the state space. A is a partial ordered set so that for every non empty subset $Y \subseteq A$ there is a join $\bigvee Y$ called the *completion* of the events Y. $\bigvee A$ is the *final* event ∞.

A state map $f : X \to X'$ is a function between two state spaces X and X' which preserve meets $f(\bigwedge Y) = \bigwedge f(Y)$ and map the initial state to the initial state $f(q_0) = q_0$.

The dual $X^\perp$ of a state space is obtained by the order dual of $X - \{q_0\}$ with q_0 added back to the bottom. The set $X \multimap Y$ of state maps from state space X to state space Y forms a state space. The partial order on $X \multimap Y$ is given by the pointwise order $f \leq g$ iff $\forall x \in X\ f(x) \leq g(x)$. We have to show the pointwise meet $\bigwedge Z$ for non empty $Z \subseteq X \multimap Y$ is meet preserving and clearly $(\bigwedge Z)(q_0) = \bigwedge_{f \in Z} f(q_0) = q_0$.

The partial order can be carved out of a Chu space and the above process constructs can be seen as the behavioral interpretation of Chu spaces. Given a relation $R \subseteq X \times A$ we define R^- to be the complement relation $R^- = X \times A \setminus R$ and $R^\# = R^{\smile -}$ is the complement of the reverse. For $R' \subseteq A \times Y$ let $R; R' = \{(x,y) \mid xRa \wedge aR'y\}$ be the composition of relations. For the two binary relations $R \subseteq U \times V$, $T \subseteq U \times W$ we define the right residual to be $R \setminus T \subseteq V \times W$ by equivalent characterizations :

(i) The operation satisfying $R; S \subseteq T$ iff $S \subseteq R \setminus T$ (observe similarity to Currying)

(ii) As the pairs $(v, w) \in V \times W$ such that $uRv \implies uTw$ for all $u \in U$

(iii) As the relation $(R^{\smile}; T^{-})^{-}$

(vi) As the relation $(T^{\#}, R)^{\#}$

So for a Chu space $\mathcal{A} = (X, \models, A)$ we define the transition relation $\rightarrow \stackrel{def}{=} \models \setminus \models$ on states X. So $x \rightarrow y$ can be seen such that every event a which has occured in state x has also occured in state y. Note the similarity of $\_^{\#}$ the complement of the reverse with the complex conjugate of the transpose of Hilbert space operators. If we see Hilbert spaces as Chu spaces with $K = 2$ substituted with $\mathbb{C}$, then right residuation in Chu_K will be $\psi \rightarrow \psi'$ the complex transition probability for states ψ and ψ'.

Here are two proposals for modeling quantum computing as the behavioral interpretation of Chu spaces.

Axiomatic approach

Take the $Chu_{[0,1]}$ space where the real interval $[0,1]$ is taken as valuation of the modeling relation$\models$. A Chu space $\mathcal{A} = (X, \models, A)$ will consist of a quantum logic A and the set X of states for that logic. So naturally we define $x \models a = x(a)$. The linear constructs on Chu spaces will give tensor $\otimes$ and linear implication $\multimap$ which should be interpreted as the type of unitary transforms. But unitary transforms are defined as a Hilbert space notion so we need the complex number numbers to state this formally.

Semantic approach

Every Chu space will be in correspondence to a Hilbert space $\mathbb{H}$. So $\mathcal{A} = (X, \models, A)$ consists of the projection logic $A = \mathbb{P}(\mathbb{H})$ to the Hilbert space and the set X of mixtures $\langle y \rangle$ of $\mathbb{H}$. The relation $\models$ will be interpreted as $\langle y \rangle \models P = s_{\langle y \rangle}(P)$. So Gleason's theorem gives the connection between the axiomatic and the semantic approach. Gleason states if the quantum logic of an axiomatic Chu space is the projection logic of a Hilbert space then there is a one-to-one correspondence to a semantic Chu space.

So tensor should be identified as the Hilbert space construct, as done in Girard's Banach space model of linear logic [Gir96], and $\multimap$ should denote unitary transforms. Further $x \rightarrow y$ should be identified as transition probability between states. Left residuation $a \rightarrow b$ of propositions $a, b \in A$ should work out as subspace relations.

Another proposal for right residuation might be $x \rightarrow y$ as indication for the subspace relation of the spaces spanned by the mixture representations $\langle x \rangle$ and $\langle y \rangle$. For left residuation $a \rightarrow b$ the commutation relation $(a, b)C$ might be also interesting. The right residuation notions which lead to the right state and event relations should be achieved by defining the operations ; $\_^{\smile}$ $\_^{-}$ on valued relations $X \times A \rightarrow [0,1]$ well.

6 Conclusion

The approach should lead to a typed concurrency calculus similar to Milners π-calculus [Mil92, Pie94]. The core of the π-calculus deals with names and communication of names. Unitary transforms deal with exchange of base vectors (or generaly the exchange of orthogonal unit vectors). On this core the calculus should be build.

The mathematical intend of names is Milners actual research [MMP95, Mil93] on action calculi which subsumes π-calculus, λ-calculus and petri-nets. In Dusko Pavlovic's study [Pav93] the distinction between names and variables in the abstract setting for action calculi is identified in terms of polynomes. The variables are seen as polynomial extensions like $\mathbb{Z}[x]$ whereas the name are seen as algebraic extensions like $\mathbb{Z}[\sqrt{2}]$ or $\mathbb{Z}[i]$. In Girard'ss Banach space model for linear logic the embedding of classical propositions by $!A$ is done in terms of analytical functions i.e. converging sums of monomes. The key to this idea is the isomorphism $!A = \bigoplus_{i=0}^{\infty} \bigotimes_{j=0}^{i} A$, which tells that $!A$ is isomorphic to a undetermined number of uses of the linear proposition A.

Finally through the eigen space problem to every operator A in the finite dimensional Hilbert spaces belongs the characteristic polynom $\chi = det(A - \lambda I) \in \mathbb{C}[\lambda]$. So the different kind of operators can be characterized by their eigen values which are roots of the characteristic polynom χ. For projections χ has only 0 and 1 as roots, for unitary transforms all roots of χ have absolute value 1, and the roots of a hermitian operator are in $\mathbb{R}$.

This intriguing similarity should also lead to insights of quantum concurrency.

References

[Abr93a] Samson Abramsky. Computational interpretation of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.

[Abr93b] Samson Abramsky. Interaction categories. In M. Ryan G. Burn, S. Gay, editor, *Theory and Formal Methods 1993*, pages 57–69. Springer Verlag, 1993.

[AGN] Samson Abramsky, Simon Gay, and Rajagopal Nagarajan. Interaction categories and the foundations of typed concurent programming.

[Ale94] Vladimir Alexiev. Application of linear logic to computation: An overview. *Bulletin of the IGPL*, 2(1):31 pp, March 1994. URL ftp.cs.ualberta.ca:pub/TechReports/TR93-18.

[BC81] Enrico G. Beltrametti and Gianni Cassinelli. *The Logic of Quantum Mechanics*. Adison-Weseley, 1981.

[Bie95] G.M. Biermann. What is a categorical model of intuitionistic linear logic. In *Proc. of Typed Lambda Calculus and Applications*, April 1995.

[Bra96] Gilles Brassard. New trends in quantum computing. In *13th Symposium on Theoretical Computer Science*, Grenoble, France, February 1996.

[BS94] G. Bellin and P.J. Scott. On the π-calculus and the linear logic. *Theoretical Computer Science*, 135:11–65, 1994.

[Coh89] David W. Cohen. *An Introduction to hilbert Space and Quantum Logic*. Springer-Verlag, 1989.

[Eke94] Artur Ekert. Quantum computation. In *Proceedings of the ICAP meeting, Boulder, CO.* Preprint at URL http://feynman.stanford.edu/qcomp/ekert/index.html, 1994.

[Gay95] Simon Gay. *Linear Types for Communicating Processes*. PhD thesis, University of London, January 1995.

[Gir95] J.Y. Girard. Linear logic: its syntax and semantics. In Jean-Yves Girard, Yves lafont, and Laurent Regnier, editors, *Advances in Linear Logic*, pages 1–42. Cambridge University Press, 1995. London Mathematical Society Lecture Notes Series 222.

[Gir96] Jean-Yves Girard. Coherent banach spaces: A continuous denotational semantics. Technical report, Institut de Mathematiques de Luminy, URL ftp://iml.univ-mrs.fr/pub/girard/banach.ps.Z, May 1996.

[GN95] Simon Gay and Rajagopal Nagarajan. A typed calculus of synchronous processes. In *Logic and Computer Science*, pages 210–220, 1995.

[Gor95] Rajeev Gore. Cut-free display calculi for relation algebras. URL http://arp.anu.edu.au/pub/techreports/1995/TR-ARP-19-95.ps.gz, 1995.

[Gup94] Vineet Gupta. *Chu Spaces: A Model of Concurrency.* PhD thesis, Stanford University, August 1994.

[IM93] Samson Abramsky Ian Mackie, Leopoldo Roman. An internal language for autonomous categories. In M. Ryan G. Burn, S. Gay, editor, *Theory and Formal Methods 1993*, pages 235–246. Springer Verlag, 1993.

[Mil78] Robin Milner. A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, 17:348–375, December 1978.

[Mil92] Robin Milner. Invited lecture: The polyadic π-calculus. LNCS 620, 1992.

[Mil93] Robin Milner. Action calculi, or syntactic action structures. In S. Sokolowski A.M. Borzyszkowaski, editor, *MFCS '93 Mathematical Foundations of C.S.*, pages 105–121. LNCS 711, August 1993.

[MMP95] Alex Mifsud, Robin Milner, and John Power. Control structures. In *Logic and Computer Science*, pages 188–198, 1995.

[NPS90] Bengt Nordström, Kent Petersson, and Jan M. Smith. *Programming in Martin-Löf's Type Theory.* Claredon Press, Oxford, 1990.

[Pav93] Dusko Pavlovic. Categorical logic of names and abstraction in action calculi. *Mathematical Structures in Computer Science*, 11:1–20, 1993.

[Pav94] Dusko Pavlovic. Categorical logic of concurrency and interaction i: synchronous processes. In C.L. Hankin et al., editor, *Theory and Formal Methods of Computing 1994*, pages 105–141. World Scientific, 1994.

[Pie94] Benjamin C. Pierce. Programming in the pi-calculus. Technical report, Department of Computer Science University of Edinburgh, March 1994.

[Pra93] Vaughan Pratt. Linear logic for generalized quantum mechnics. In *Proc. Workshop on Physics and Computation*, pages 166–180, Dallas, 1993. IEEE.

[Pra94a] Vaughan Pratt. Chu spaces: Automata with quantum aspects. In *Proc. Workshop on Physics and Computation*, pages 186–195, Dallas, 1994. IEEE.

[Pra94b] Vaughan Pratt. Chu spaces: Complementarity and uncertainty in rational mechanics. Technical report, Budapest, 1994. Course notes, TEMPUS summer school, 35 pp.

[Pra95a] Vaughan Pratt. Chu spaces and their interpretation as concurrent objects. In J. van Leeuwen, editor, *Computer Science Today: Recent Trends and Developments*, pages 392–405. Springer-Verlag, 1995. LNCS 1000.

[Pra95b] Vaughan Pratt. Rational mechanics and natural mathematics. In *TAPSOFT' 95*, pages 108–122, Aarhus, Denmark, 1995. Springer-Verlag. LNCS 915.

[Sho94] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In *Proccedings of the* 35^{th} *Annual Symposium on Foundations of Computer Science*, pages 124–134, Santa Fe, November 1994. IEEE, IEEE press.

[Var85] V. S. Varadarajan. *Geometry of Quantum Theory.* Springer Verlag, 1985. 1985.5299 in Uni Stuttgard 3H5754.

[Wad93] Philip Wadler. A taste of linear logic. In *Mathematical Foundations of Computer Science.* Springer Verlag LNCS 711, 1993.

[WN95] Glynn Winskel and Mogens Nielsen. Models for concurrency. In *Summer School on Logics of Computation*, pages 1–78. ESPRIT CLICS-II, University of Cambridge, September 1995.

Appendix

Here we recall the basic functional analytic and quantum theory facts. We are always working with vector spaces over the field $\mathbb{C}$ of complex numbers.

Definition. A *inner product space* is a vector space V with a inner product $\langle , \rangle : V \times V \to \mathbb{C}$ with

$$\begin{array}{lcll} \langle \lambda x, y\rangle & = & \lambda\langle x, y\rangle & \text{linear ...} \\ \langle x+y, z\rangle & = & \langle x, z\rangle + \langle y, z\rangle & \text{... in the first argument} \\ \langle x, y\rangle & = & \langle y, x\rangle^* & \text{commutation implies complex conjugation} \\ \langle x, x\rangle & = & 0 & \text{iff } x = 0 \end{array}$$

Two vectors $x, y \in V$ are orthogonal to each other $x \perp y$ iff $\langle x, y\rangle = 0$. Given a subspace S of $\mathbb{H}$ the orthogonal subspace is defined by $S^\perp = \{x \mid x \perp y \text{ for all } y \in S\}$

Definition. A norm $||.|| : V \to \mathbb{R}$ on a vector space is defined by the properties:

$$\begin{array}{lcll} ||x|| & = & 0 & \text{iff } x = 0 \\ ||\lambda x|| & = & |\lambda|\,||x|| & \\ ||x+y|| & \leq & ||x|| + ||y|| & \text{triangle inequality} \end{array}$$

The norm of a inner product space is defined as $||x|| \overset{def}{=} \sqrt{\langle x, x\rangle}$.
A Cauchy sequence in a vector space with norm is a sequence $\langle x_k\rangle$ satisfying:

$$\forall \epsilon \in \mathbb{R} \;\; \exists n \in \mathbb{N} \;\; \forall l, m \geq n \;\; ||x_l - x_m|| < \epsilon$$

A vector space with norm is *complete* iff every Cauchy sequence converges. A *Hilbert space* $\mathbb{H}$ is a complete inner product space.

Operators on a Hilbert space

A operator on a Hilbert space is a liner map $A : \mathbb{H} \to \mathbb{H}$. The identity map is named I. A operator A is bounded iff:

$$\exists b \in \mathbb{R} \;\; \forall x \in \mathbb{H} \;\; ||Ax|| \leq b||x||$$

$\mathbb{B}(\mathbb{H})$ is the set of all bounded operators. For every bounded operator A there exists a unique operator $A^\#$ with $\langle x, Ay\rangle = \langle A^\# x, y\rangle$ called the *adjoint* operator to A. We define some properties for operators :

self adjoint or hermitian	A	$=$	$A^\#$
projection	P	$=$	$PP^\#$
unitary	$UU^\#$	$=$	I

The set of projection $\mathbb{P}(\mathbb{H})$ are in one to one correspondence with the subspaces of $\mathbb{H}$. The map $img(P) = \{Px \mid x \in \mathbb{H}\}$ is a bijection. We define P^S to be the projection related to the subspace S of $\mathbb{H}$.

The eigen space problem to a operator A is defined to find the complex numbers λ and vectors x which solve the equation $Ax = \lambda x$. The solution values λ are called eigen values. The set of solution vectors to a eigen value form a subspace called the eigen space to the eigen value. The eigen values of a hermitian operator A are all real and the eigen spaces are all orthogonal to each other.

Quantum mechanics the discrete case

Quantum mechanics models the outcomes of experiment in size of particles. Here we want to deal with the case that the outcomes V of the experiment A are a subset of $\mathbb{N}$. A test $M_A(W)$ is a experiment with yes-no outcome stating that performing experiment A leads to an outcome which will be an element of $W \subseteq V$. A state of a system is given by a weight function $\omega : V \to [0,1]$ with $\sum_{v \in V} \omega(v) = 1$. So $\omega(v)$ gives the probability of getting outcome v when performing the experiment.

The idea of quantum theory is to associate to every outcome v a base vector x_v of a Hilbert space. So we get a indexed orthonormal base $\langle x_v \rangle_{v \in V}$. A (pure) state of the system can be represented by a unit vector $\psi \in \mathbb{H}$. The weight function associated to a unit vector is given by

$$\omega_\psi : V \to [0,1] \qquad v \mapsto |\langle \psi, x_v \rangle|^2$$

Further to $W \subset V$ corresponds the subspace spanned by $S = \langle x_v \rangle_{v \in W}$. The probability for the yes outcome of a test $M_A(W)$ in system state ψ can be computed by $\langle \psi, P^S \psi \rangle$. The tests on a system are the propositions of quantum logic. So the propositions are related to subspaces of the space or to projections on the Hilbert space.

There were some simplification in the explanations above. In a real system to every outcome of an experiment v there is a corresponding subspace S_v and the Hilbert space is spanned by the orthogonal subspaces $\langle S_v \rangle_{v \in V}$. This leads to the notion of an observable of an experiment A:

$$\mathcal{O}_A : V \to \mathbb{P}(\mathbb{H}) \qquad v \mapsto P^{S_v} \qquad \text{with } v \neq v' \implies S_v \perp S_{v'}$$

The notion of an observable can be extended to the continuous case with some measure theory.

Given a hermitian operator A then we get a map $\mathcal{O}_A : \mathbb{R} \to \mathbb{P}(\mathbb{H})$ that relates to each eigen value the projector to its eigen space. This map is a observable because the eigen spaces are orthogonal to each other. So hermitian operators are in one-to-one correspondence with the observables with real outcome.