Title: The logic of quantum actions: reasoning about change in quantum systems

Date: Jul 19, 2005 04:45 PM

URL: http://pirsa.org/05070103

Abstract:

Overview

0. Intro
1. The Algebra of Quantum Actions on a Hilbert space
2. Quantum Dynamic Algebras

Intro

We will look at quantum systems as *dynamic* systems.

A system is its potential (and actual) behaviour. We adopt the standard way of looking at systems in Computer Science and in Modal Logic: as *dynamic* systems, or "processes".

To reason about **dynamic systems**, one can use a **logic of actions (or programs)**, which can be described in various ways: propositional dynamic logic (PDL), dynamic frames, labeled transition systems, dynamic algebra.

I will focus on a **Quantum Dynamic Algebra** as a complete axiomatization of the algebra of quantum actions.

TeXShop File Edit Format Macros Typeset Preview Window Help din 22:51 PM

C4.pdf

The Algebra of Quantum Actions on a Hilbert space

Let H be a Hilbert space.

We introduce five notions: *states, (testable) properties, quantum tests, unitary actions, quantum actions.*

TeXShop File Edit Format Macros Typeset Preview Window Help din 22:51 PM

C4.pdf

States and Properties

States : rays (one dimensional subspaces) of H, denoted by Σ

Logical Properties : sets of states $S \subseteq \Sigma$

In Quantum Logic, one usually concentrates only on properties that can in principle be tested by measurements. These are called **experimental** (or *testable*) properties.

In a Hilbert space, these correspond to *closed linear subspaces* of H.

The family of all closed linear subspaces of H is denote by $\mathcal{L}(H)$ and $\mathcal{L}$ is the family of all corresponding subsets $P = \bigcup W$ of Σ (corresponding to closed linear subspaces $W \subseteq H$).

TeXShop File Edit Format Macros Typeset Preview Window Help din 22:52 PM

C4.pdf

$\mathcal{L}$ forms a complete atomic orthomodular lattice, with inclusion as partial order, intersection as meet, and $\sim S = \{t : \forall s \in S t \perp s\}$ as orthocomplement.

The join in this lattice is denoted by $\sqcup$ and satisfies

$$S \sqcup T = \sim (\sim S \cap \sim T)$$

The atoms of $\mathcal{L}$ correspond to elements of Σ.

Quantum Actions (or Programs)

A *quantum action (or program)*, in our sense, is an abstraction of real-life quantum programs, capturing only their input-output (relational) behaviour.

As **quantum actions** over the Hilbert space H, we take all the **binary relations** on states in Σ induced by *projectors* and *unitary evolutions* on H, then we close this family under relational composition $R\dot{R}'$ and arbitrary unions $\bigcup_i R_i$ of relations.

The family of quantum actions over H is denoted by $\mathcal{Q}(\Sigma)$.

TeXShop File Edit Format Macros Typeset Preview Window Help din 22:55 PM

C4.pdf

Composition, Union, Measurements

Composition is needed to represent sequential composition of programs, while unions are needed to represent *non-deterministic choice*. A measurement can be represented as a union of mutually orthogonal projectors.

$\mathcal{Q}(\Sigma)$ forms a complete lattice with inclusion, in which the relational union $\bigcup$ is the join. Moreover, $(Q(\Sigma), ;\, \bigcup)$ forms a quantale.

An action $\pi \in \mathcal{Q}(\Sigma)$ is *deterministic* if its underlying relation $\pi \subseteq \Sigma \times \Sigma$ is a partial function.

We denote by $\mathcal{D} = \mathcal{D}(\Sigma)$ the family of all deterministic quantum actions.

TeXShop File Edit Format Macros Typeset Preview Window Help din 22:57 PM

C4.pdf

Properties as Actions (or Programs)

For any property $S \in \mathcal{L}$, the projector onto the closed linear subspace W_S corresponding to S will be denoted as $S?$. We read this as the *quantum test* of S

There exists a bijective correspondence between testable properties $S \in \mathcal{L}$ and the projectors $S? \in \mathcal{Q}(\Sigma)$ on the corresponding subspaces. So one can embedd $\mathcal{L}$ into $\mathcal{Q}(\Sigma)$.

From now on, we *identify* a testable property S with the corresponding test $S?$ and hence we have:

$$\mathcal{L} \subseteq \mathcal{Q}(\Sigma)$$

Note that the order relations $\leq$ on $\mathcal{L}$ and $\subseteq$ on $\mathcal{Q}(\Sigma)$ differ!

TeXShop File Edit Format Macros Typeset Preview Window Help din 22:57 PM

C4.pdf

Properties as Actions (or Programs)

For any property $S \in \mathcal{L}$, the projector onto the closed linear subspace W_S corresponding to S will be denoted as $S?$. We read this as the *quantum test* of S

There exists a bijective correspondence between testable properties $S \in \mathcal{L}$ and the projectors $S? \in \mathcal{Q}(\Sigma)$ on the corresponding subspaces. So one can embedd $\mathcal{L}$ into $\mathcal{Q}(\Sigma)$.

From now on, we *identify* a testable property S with the corresponding test $S?$ and hence we have:

$$\mathcal{L} \subseteq \mathcal{Q}(\Sigma)$$

Note that the order relations $\leq$ on $\mathcal{L}$ and $\subseteq$ on $\mathcal{Q}(\Sigma)$ differ!

$p \leq q$ in $\mathcal{L}$ while for the corresponding projectors in $\mathcal{Q}(\Sigma)$: $p \not\subseteq q$

Image, Strongest Postcondition

The *image* of set $S \subseteq \Sigma$ of states via an action $\pi \in \mathcal{Q}(\Sigma)$ is given by :

$$\pi(S) = \{t : (s,t) \in \pi\}$$

For a state $s \in \Sigma$ we put $\pi(s) = \pi(\{s\})$.

The image of a testable property (via a non-deterministic action) is not necessarily testable! For this reason we introduce:

The *strongest (testable) postcondition* ensured by executing action π on property S is given by

$$\pi[S] = \sim\sim (\pi(S)) = \bigsqcup_{s \in S} \{s\}$$

Weakest Precondition

The *weakest precondition* of π with respect to (postcondition) $T \subseteq \Sigma$ is given by:

$$[\pi]T = \{s : \pi(s) \subseteq T\}$$

If T is testable then $[\pi]T$ is also testable.

Note that we have the following Galois Duality, for all $S, T \subseteq \Sigma$:

$$S \subseteq [\pi]T \text{ iff } \pi(S) \subseteq T$$

For *testable* properties $S, T \in \mathcal{L}$ we can strengthen this to:

$$S \subseteq [\pi]T \text{ iff } \pi[S] \subseteq T$$

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:02 PM

C4.pdf

Kernel of an action

The *Kernel* of an action $\pi \in \mathcal{Q}(\Sigma)$ is given by:

$$Ker(\pi) = \{s \in \Sigma : \pi(s) = \emptyset\}$$

This yields a *testable property* $Ker(\pi) \in \mathcal{L}$, expressing the *impossibility* (necessary failure) of action π.

For test actions $S?$, the kernel corresponds to the orthocomplement of S:

$$Ker(S?) = \sim S$$

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:03 PM

C4.pdf

Kernel as generalized orthocomplement

This suggests generalizing the orthocomplement notation to all quantum programs, by putting:

$$\sim: \mathcal{Q}(\Sigma) \to \mathcal{Q}(\Sigma), \ \sim \pi := Ker(\pi)$$

As an action (test), this represents a *test for failure of the action* π

Observe that in general for $Q \in \mathcal{Q}(\Sigma)$: $\sim\sim \pi \neq \pi$.

But $\sim\sim \pi = \pi$ iff $\pi \in \mathcal{L}$.

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:05 PM

C4.pdf

Conjunction and Choice

Note that we have the following, for any family of $\{P_i\}_i$ of testable properties :

$$\bigwedge_i P_i = \sim (\bigcup_i \sim P_i)$$

This relates propositional conjunction with choice of actions, and helps us understand Piron's definition of the meet of testable properties in terms of the choice between yes-no questions (actions).

As a consequence we also have:

$$\bigsqcup_i P_i = \sim\sim \bigcup_i P_i$$

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:06 PM

C4.pdf

Composition and Weakest Precondition

Note that we have the following, for any testable properties P:

$$[\pi]P = \sim (\pi . \sim P)$$

So the weakest precondition of properties is related to the composition of actions.

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:07 PM

C4.pdf

Unitary evolutions:

Note that the family $\mathcal{U} \subseteq \mathcal{Q}(\Sigma)$ of all *"unitary evolutions"* can be characterized as the set of all reversible orthogonality-preserving actions:

$u \in \mathcal{U}$ iff

$u \in \mathcal{Q}$, $\exists u'\ u \cdot u' = u' \cdot u = 1$, and
$u \cdot \sim P = \sim (u \cdot P)$ for all $P \in \mathcal{L}$.

Deterministic actions:

The following are equivalent:

- $\pi \in \mathcal{D}$
- $\forall P \in At(\mathcal{L}) \exists Q \in At(\mathcal{L})\ \pi(P) \leq Q$
- $\forall P \in At(\mathcal{L}) \exists Q \in At(\mathcal{L})\ P \leq [\pi] Q$

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:09 PM

C4.pdf

3. Quantum Dynamic Algebras

We abstract away now from the concrete algebra of quantum actions over a Hilbert space, and define an abstract algebraic structure, that captures all the main properties of $\mathcal{Q}(\Sigma)$.

We obtain *Quantum Dynamic Algebras*, as a *complete axiomatization of the algebra of quantum actions.*

This is related to the algebraic semantics for PDL (cfr. Dynamic Frame) in terms of **dynamic algebra's with tests and actions** (Pratt & Kozen '79). We extend this idea, linking it both to the work on quantales to capture the dynamics of quantum systems, and to the work of Solèr and Mayet and others on the complete axiomatization of the lattice of complete subspaces of a Hilbert space.

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:11 PM

C4.pdf

A **quantum dynamic algebra** is a structure:

$(\mathcal{Q}, \bigcup, \cdot, \sim)$

consisting of: a set $\mathcal{Q}$ of *quantum programs*, and operations:

$\bigcup : \mathcal{P}(\mathcal{Q}) \to \mathcal{Q}$
$\cdot : \mathcal{Q} \times \mathcal{Q} \to \mathcal{Q}$
$\sim : \mathcal{Q} \to \mathcal{Q}$ (*"test for failure" of an action*),

subject to a number of conditions (to be given later).

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:12 PM

C4.pdf

Notations

We put

$$\mathcal{L} := \{\sim x : x \in \mathcal{Q}\}$$

for the family of all "tests" (or "testable properties"). We use variables $p, q, \ldots$ for elements of $\mathcal{L}$.

$$0 := \bigcup \emptyset \qquad 1 := \sim 0$$
$$[x]p := \sim (x \cdot \sim p) \qquad \bigwedge_i p_i := \sim \bigcup \sim p_i$$
$$p \leq q \text{ iff } p \wedge q = p \qquad p \perp q \text{ iff } p \leq \sim q$$
$$\bigsqcup_i p_i := \sim\sim \bigcup_i p_i$$

We also enote by $At(\mathcal{L})$ the set of all *atoms* of $\mathcal{L}$ (corresponding to "states"): :

$$\{p \in \mathcal{L} \mid \forall q \in \mathcal{L}(0 \neq q \leq p \Rightarrow q = p)\}$$

A **quantum dynamic algebra** is a structure:

$(\mathcal{Q}, \bigcup, \cdot, \sim)$

consisting of: a set $\mathcal{Q}$ of *quantum programs*, and operations:

$\bigcup : \mathcal{P}(\mathcal{Q}) \rightarrow \mathcal{Q}$
$\cdot : \mathcal{Q} \times \mathcal{Q} \rightarrow \mathcal{Q}$
$\sim : \mathcal{Q} \rightarrow \mathcal{Q}$ (*"test for failure" of an action*),

subject to a number of conditions (to be given later).

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:12 PM

C4.pdf

Notations

We put

$$\mathcal{L} := \{\sim x : x \in \mathcal{Q}\}$$

for the family of all "tests" (or "testable properties"). We use variables $p, q, \ldots$ for elements of $\mathcal{L}$.

$$0 := \bigcup \emptyset \qquad 1 :=\sim 0$$
$$[x]p :=\sim (x\cdot \sim p) \qquad \bigwedge_i p_i :=\sim \bigcup \sim p_i$$
$$p \leq q \text{ iff } p \wedge q = p \qquad p \perp q \text{ iff } p \leq\sim q$$
$$\bigsqcup_i p_i :=\sim\sim \bigcup_i p_i$$

We also enote by $At(\mathcal{L})$ the set of all *atoms* of $\mathcal{L}$ (corresponding to "states"): :

$$\{p \in \mathcal{L} \mid \forall q \in \mathcal{L}(0 \neq q \leq p \Rightarrow q = p)\}$$

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:13 PM

C4.pdf

More Notations

Let $\mathcal{U}$ be the set of all *"unitary evolutions"*, defined as the reversible orthogonality-preserving programs:

$$x \in \mathcal{U}$$

iff:

$x \in \mathcal{Q}$, $\exists y \; x \cdot y = y \cdot x = 1$, and
$x \cdot \sim p = \sim (x \cdot p)$ for all $p \in \mathcal{L}$.
We use variables u to denote the elements of $\mathcal{U}$.

The set $\mathcal{D}$ of *deterministic programs* can be defined as:

$$\mathcal{D} := \{x \in \mathcal{Q} : \forall a \in At(\mathcal{L}) \exists b \in At(\mathcal{L}) a \leq [x]b\}$$

Observe for any deterministic program $x \in \mathcal{D}$ and any atom $a \in \mathcal{L}$, *there exists at most one* atom $b \in \mathcal{L}$ such that:

$$a \leq [y]b, a \not\leq \sim y$$

If such a b exists, we write

$$x(a) = b$$

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:14 PM

C4.pdf

Strongest Post-Condition

The *strongest post-condition* internalizes the image-set inside $\mathcal{L}$, by taking its (quantum) join:

$$x[p] := \bigsqcup x(p)$$

This is an element of $\mathcal{L}$ which represents the (test corresponding to the) biorthogonal closure of the image. But, for *deterministic* programs, this closure coincides with the image, as defined above:

$$x(a) = x[a]$$

Moreover, we have:

$$x \text{ deterministic }, a \in At(\mathcal{L}) \implies x[a] \in At(\mathcal{L}) \cup \{0\}$$

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:15 PM

C4.pdf

Axioms for QDA's

1. $0 \in \mathcal{L}$; or, equivalently: $\sim 1 = 0$

2. $(\mathcal{Q}, \bigcup, 1)$ is a quantale generated by the set $\mathcal{L} \cup \mathcal{U}$.

3. *Choice*: $[\bigcup_i x_i]p = \bigwedge_i [x_i]p$; or, equivalently: $\sim \bigcup_i x_i = \bigwedge \sim x_i$

4. *Composition*: $[\pi \cdot \sigma]p = [\pi][\sigma]p$; or, equivalently: $\sim (x \cdot y) = [x] \sim y$

5. *Adequacy* : $p \wedge q \leq [q]p$, and also $p \wedge [p]q \leq q$

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:15 PM

C4.pdf

Axioms for QDA's

1. $0 \in \mathcal{L}$; or, equivalently: $\sim 1 = 0$

2. $(\mathcal{Q}, \bigcup, 1)$ is a quantale generated by the set $\mathcal{L} \cup \mathcal{U}$.

3. *Choice*: $[\bigcup_i x_i]p = \bigwedge_i [x_i]p$; or, equivalently: $\sim \bigcup_i x_i = \bigwedge \sim x_i$

4. *Composition*: $[\pi \cdot \sigma]p = [\pi][\sigma]p$; or, equivalently:
   $\sim (x \cdot y) = [x] \sim y$

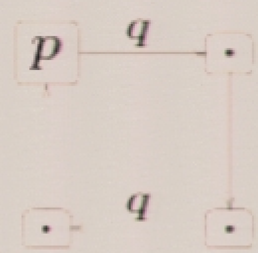
5. *Adequacy* : $p \wedge q \leq [q]p$, and also $p \wedge [p]q \leq q$

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:17 PM

C4.pdf

6. *Proper Superpositions*:
if $p, q \neq 0$ then there exists $r \in \mathcal{L}$ such that $r \not\perp p$, $r \not\perp q$.

7. *Self-Adjointness Axiom*: For all $p, q \in \mathcal{L}$: $p \leq [q] \sim [q] \sim p$



8. "Covering Law":

9. *"Atomicity"*: $\mathcal{L}$ is *atomistic*, i.e.:
   $p \leq \bigsqcup\{q \in At(\mathcal{L}) : q \leq p\}$

10. *Mayet's Condition*. There exist $p, q, r \in \mathcal{L}$, $u \in \mathcal{U}$, such that:
    $p \leq [u]p\,,\ p \neq [u]p\,,\ q \neq 0, r \neq 0\,,\ q \perp r$.

11. *Actions are determined by their behavior on atoms*:
    If $x(a) = y(a)$ for all $a \in At(\mathcal{L})$, then $x = y$.

12. *Image commutes with unions*: $(\bigcup_{i\in I} x_i)(p) = \bigcup_{i\in I} x_i(p)$.

The statement of the last two axioms 11. and 12. may look set-theoretical, but these two axioms can be replaced by only one non-set-theoretical axiom:

Representation Theorem

Our axioms completely characterise quantum actions (or programs):

Theorem 1. Every Quantum Dynamic Algebra is isomorphic to an algebra of concrete quantum actions (i.e. a subalgebra of the algebra $\mathcal{Q}(\Sigma)$ of quantum programs over some infinite-dimensional Hilbert space).

We can use this algebraic setting to *reason* about quantum actions. Hence this setting gives us an algebraic semantics for the proof system of quantum dynamic logic QDL.

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:23 PM

C4.pdf

Representation Theorem

Our axioms completely characterise quantum actions (or programs):

Theorem 1. Every Quantum Dynamic Algebra is isomorphic to an algebra of concrete quantum actions (i.e. a subalgebra of the algebra $\mathcal{Q}(\Sigma)$ of quantum programs over some infinite-dimensional Hilbert space).

We can use this algebraic setting to *reason* about quantum actions. Hence this setting gives us an algebraic semantics for the proof system of quantum dynamic logic QDL.

Representation Theorem

Our axioms completely characterise quantum actions (or programs):

Theorem 1. Every Quantum Dynamic Algebra is isomorphic to an algebra of concrete quantum actions (i.e. a subalgebra of the algebra $\mathcal{Q}(\Sigma)$ of quantum programs over some infinite-dimensional Hilbert space).

We can use this algebraic setting to *reason* about quantum actions. Hence this setting gives us an algebraic semantics for the proof system of quantum dynamic logic QDL.

Language of QDL

Take the **Syntax** of Propositional Dynamic Logic with Converse

(consisting of *propositions* φ and *programs* π):

$$\begin{array}{lllllll} \varphi & ::= & p & \mid c & \mid \neg\varphi & \mid \varphi \wedge \varphi & \mid [\pi]\varphi \\ \pi & ::= & \varphi? & \mid U & \mid U^{\dagger} & \mid \pi \cup \pi & \mid \pi;\pi \end{array}$$

Language of QDL

Take the **Syntax** of Propositional Dynamic Logic with Converse

(consisting of *propositions* φ and *programs* π):

$$\begin{array}{lllllll} \varphi & ::= & p & \mid c & \mid \neg\varphi & \mid \varphi \wedge \varphi & \mid [\pi]\varphi \\ \pi & ::= & \varphi? & \mid U & \mid U^{\dagger} & \mid \pi \cup \pi & \mid \pi ; \pi \end{array}$$

TeXShop File Edit Format Macros Typeset Preview Window Help din 23:25 PM

C4.pdf

Language of QDL

Take the **Syntax** of Propositional Dynamic Logic with Converse

(consisting of *propositions* φ and *programs* π):

$$\begin{array}{lcllllll} \varphi & ::= & p & \mid c & \mid \neg\varphi & \mid \varphi \wedge \varphi & \mid [\pi]\varphi \\ \pi & ::= & \varphi? & \mid U & \mid U^{\dagger} & \mid \pi \cup \pi & \mid \pi ; \pi \end{array}$$