Some No-Go Results in Quantum Domain Theory

Robert Furber

Edinburgh University

1st July, 2022

Outline

1. Domain Theory in 5 minutes
2. Finite-Dimensional Quantum Programs
3. Infinite-Dimensional Quantum Programs (C^*-algebras)

What is Domain Theory

- Algebraic expressions *e.g.*

$$x^2y^4 \qquad\qquad \frac{x}{\sqrt{x^2+y^2}}$$

  can be interpreted as functions on *e.g.* on $\mathbb{R}^2 \setminus \{(0,0)\}$, which is then called the *domain*.
- How do we interpret expressions in programming languages as functions, *e.g.*

$$x \qquad\qquad \lambda x.x$$

$$\lambda x.f(f(x)) \qquad\qquad \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

- What is their domain? What do the variables vary over?
- A *dcpo* (directed-complete partial order), with functions being *Scott continuous*.
- Why?

Application: Defining Recursive Functions

- Consider a recursively defined function, *e.g.*

```
len([]) := 0
len(x : xs) := len(xs) + 1
```

- Make the function take an argument to play the role of the recursive call:

```
len'(f, []) := 0
len'(f, x : xs) := f(xs) + 1
```

- Solve: `len(xs) = len'(len, xs)`
- How? By iteration.

Application: Defining Recursive Functions II

- Take $\perp$ to be the totally undefined partial function.
- The zeroth-order approximation: take $f = \perp$, so `len`$_0$`(xs) := len'(`$\perp$`,xs)`, so:

  `len`$_0$`([]) = 0`
  `len`$_0$`(x: xs) = `$\perp$`(xs) + 1 = `$\perp$

- So `len`$_0$ works for an empty list only.

Application: Defining Recursive Functions III

- Define the $(n+1)$th-order approximation by feeding the nth-order approximation back in: $\texttt{len}_{n+1}\texttt{(xs) = len'(len}_n\texttt{,xs)}$.
- For example:

$$\texttt{len}_1\texttt{([]) = 0}$$
$$\texttt{len}_1\texttt{(x:[]) = len}_0\texttt{([]) + 1 = 1}$$
$$\texttt{len}_1\texttt{(x}_1\texttt{:x}_2\texttt{:xs) = }\bot$$

- Partial functions are ordered by "definedness", so $\texttt{len}_0 \leq \texttt{len}_1 \leq \cdots$ forms a monotone increasing sequence.
- Define $\texttt{len} := \bigvee_{n=0}^{\infty} \texttt{len}_\texttt{n}$.
- This is a total function doing what we want and satisfying the required equation.

What are Domains then?

- A *dcpo* is a partially ordered set where all monotone *nets* have least upper bounds (*directed-complete partial order*).
- A morphism of dcpos is required to preserve this structure, *i.e.* be order preserving and preserve least upper bounds of monotone nets. This is called being *Scott continuous*.
- We represent a (part of a) program as a Scott-continuous function between dcpos with a bottom element.
- The most basic set-up is to just use dcpos with a bottom element $\perp$. These form a cartesian-closed category and have the right structure to complete the definition of a recursive function as described above.
- Often further requirements are added, such as being a continuous or algebraic or bifinite dcpo.

Relationship to Quantum Programming Languages

- Matrices form a partial order under the *Löwner order*, defined by the cone of positive matrices.
- This cone is a "bounded dcpo".
- Density matrices of trace ≤ 1 form a (continuous) dcpo.
- The set of CPTN maps[1] $M_n \to M_m$ are always Scott continuous and form a (continuous) dcpo.
- First proved in [Sel04b, Example 2.7] (QPL 2004).
- This domain structure is used to define recursive functions and loops in [Sel04a].

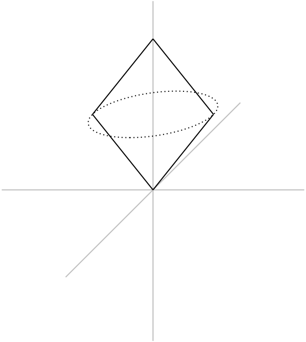
[1] Completely Positive Trace-Nonincreasing, *a.k.a. superoperators*

Example? Approximating Unitaries

- An attempted example application.
- In reality we cannot just use all unitaries as gates because there are uncountably many (and languages are countable).
- A universal set of gates is chosen and used to approximate any other to a desired degree of accuracy (Solovay-Kitaev).
- Can this process of approximation be done domain-theoretically in the Löwner order?
- Is there a countable set $\mathcal{B}$ of CPTN maps such that for every CPTN map f there exists a monotone net $(f_i)_{i\in I}$ in $\mathcal{B}$ with $f = \bigvee_{i\in I} f_i$?
- No.

Why not?

- First show analogous fact for approximating 1-dimensional projections in M_2 considered as a C*-algebra:



- Key fact: If $p = |\psi\rangle\langle\psi|$ is a 1-dimensional projection, $a \in M_n$ is positive, and $a \leq p$, then $a = \alpha p$ for some $\alpha \in [0, 1]$.

Reasoning

- If $\mathcal{B} \subseteq M_2$ and for all 1-dimensional projections p, there exists $(a_i)_{i\in I}$ in $\mathcal{B}$ with $\bigvee_{i\in I} a_i = p$ then we can pick some non-zero $a_i \leq p$.
- So we can define a function f from 1-dimensional projections to $\mathcal{B}$, such that $0 \neq f(p) \leq p$. Since $f(p) = \alpha p$, the function f is injective.
- Therefore $|\mathcal{B}| \geq |\mathbb{R}|$, so is not countable.

Unitaries?

- The same argument works for the unit interval of M_n (for $n \geq 2$). What about CPTN maps?
- Use Choi-Jamiołkowski: $\mathrm{CP}(M_n, M_m) \cong (M_{nm})_+$. But this is only an isomorphism CP maps with positive elements.
- Re-do the argument where the dcpo in question is the positive part of the unit ball of an arbitrary norm on M_n.
- Specialize to the case of the operator norm $M_n \to M_m$ considered as a norm on M_{nm} under Choi-Jamiołkowski.
- Conclusion - there is no countable set $\mathcal{B}$ of CPTN maps such that for all CPTN maps $f : M_n \to M_m$ there exists $(f_i)_{i \in I}$ in $\mathcal{B}$ and $\bigvee_{i \in I} f_i = f$.

Unitaries? II

- Alternative statement: The set of CPTN maps is a continuous dcpo, but does not have a countable basis.
- In particular, if we have a programming language that represents a universal set of unitaries and measurement in the computational basis, we cannot write a program to approximate an arbitrary unitary operator or an arbitrary completely positive map.
- It is necessary to use the norm topology of M_n for approximation, we cannot use domain theoretic topologies (the Scott topology and the Lawson topology).
- We need to use non-Löwner-monotone sequences, because for monotone sequences the limit and the least upper bound are the same.

Infinite-Dimensional Continuous Dcpos

- Since M_n is a continuous dcpo, we can ask if this holds for any infinite-dimensional C*-algebras.
- Infinite-dimensional W*-algebras have been used for program semantics by several authors.
[Cho14, Ren14, CW16, KLM20, JKL$^+$22]
- Remark: not every C*-algebra is a dcpo (*e.g.* $C([0,1])$)
- If it is, it is called *monotone complete*.
- Monotone-complete C*-algebras have a good theory of projections (they are AW*-algebras). In particular, projections form a lattice.
- A W*-algebra A is a monotone-complete C*-algebra that is separated by its Scott-continuous[2] linear maps $A \to \mathbb{C}$.

[2] Called *normal* in the operator algebra community.

Continuity of C*-algebras

- Key lemma: For a monotone-complete C*-algebra A, $\mathrm{Proj}(A)$ is a continuous lattice if $[0,1]_A$ is a continuous dcpo.
- Warning! It is not the case that a subdcpo of a continuous dcpo is continuous.
- But it is known that a sublattice of a continuous lattice is continuous – continuous lattices are characterized by a(n infinitary) distributive law.
- $\mathrm{Proj}(A)$ is a sublattice of $[0,1]_A$ even though $[0,1]_A$ is not a lattice. ($\mathrm{Proj}(A) \hookrightarrow [0,1]_A$ preserves all joins and meets).
- It turns out that a sublattice of a continuous dcpo is a continuous lattice.

Continuity of the Projection Lattice

- Answering a question of Mathys Rennela, Nik Weaver [Wea13] worked out that a W*-algebra A can only have a continuous projection lattice if it is a product of finite-dimensional matrix algebras: $A \cong \prod_{i \in I} M_{n_i}$.
- In fact, this holds for AW*-algebras, and since monotone-complete C*-algebras are AW*-algebras, we can conclude that $[0,1]_A$ is not continuous unless it is a product of finite-dimensional matrix algebras.
- We know that $[0,1]_A$ is continuous for $A \cong \prod_{i \in I} M_{n_i}$ essentially by Selinger's earlier proof plus standard domain-theoretic reasoning.
- Kornell [Kor18] calls such algebras *hereditarily atomic*, so we have that $[0,1]_A$ is a continuous dcpo iff A is hereditarily atomic.

References I

- Kenta Cho, *Semantics for a quantum programming language by operator algebras*, Proceedings of the 11th workshop on Quantum Physics and Logic, Kyoto, Japan, 4-6th June 2014 (Bob Coecke, Ichiro Hasuo, and Prakash Panangaden, eds.), Electronic Proceedings in Theoretical Computer Science, vol. 172, Open Publishing Association, 2014, pp. 165–190.
- Kenta Cho and Bram Westerbaan, *Von Neumann Algebras form a Model for the Quantum Lambda Calculus*, `https://arxiv.org/abs/1603.02133`, 2016.
- Xiaodong Jia, Andre Kornell, Bert Lindenhovius, Michael Mislove, and Vladimir Zamdzhiev, *Semantics for Variational Quantum Programming*, Proc. ACM Program. Lang. **6** (2022), no. POPL, 1–31.

References II

- Andre Kornell, Bert Lindenhovius, and Michael W. Mislove, *Quantum CPOs*, Proceedings 17th International Conference on Quantum Physics and Logic, QPL 2020, Paris, France, June 2 - 6, 2020 (Benoît Valiron, Shane Mansfield, Pablo Arrighi, and Prakash Panangaden, eds.), EPTCS, vol. 340, 2020, pp. 174–187.
- André Kornell, *Quantum sets*, `https://arxiv.org/abs/1804.00581v1`, 2018.
- Mathys Rennela, *Towards a Quantum Domain Theory: Order-enrichment and Fixpoints in W*-algebras*, Electronic Notes in Theoretical Computer Science **308** (2014), 289–307, Proceedings of the 30th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXX).

References III

Peter Selinger, *Towards a Quantum Programming Language*, Mathematical Structures in Computer Science **14** (2004), no. 4, 527–586.

______, *Towards a Semantics for Higher-Order Quantum Computation*, Proceedings of the 2nd International Workshop on Quantum Programming Languages, TUCS General Publications, vol. 33, Turku Centre for Computer Science, 2004, pp. 127–143.

Nik Weaver, *Projections in a W^*-algebra as a continuous lattice?*, `https://mathoverflow.net/q/133037`, 2013.