Annals of Pure and Applied Logic 168 (2017) 404–469

Contents lists available at ScienceDirect

Annals of Pure and Applied Logic

www.elsevier.com/locate/apal

Semantics of higher-order quantum computation via geometry of interaction ☆

Ichiro Hasuo [a,*,1], Naohiko Hoshino [b,2]

[a] *Department of Computer Science, The University of Tokyo, Hongo 7-3-1, Tokyo 113-8656, Japan*
[b] *Research Institute for Mathematical Sciences, Kyoto University, Kitashirakawa-Oiwakecho, Kyoto 606-8502, Japan*

ARTICLE INFO

Article history:
Available online 20 October 2016

MSC:
03B70
18C20
18C50
81P68

Keywords:
Higher-order computation
Quantum computation
Programming language
Geometry of interaction
Denotational semantics
Categorical semantics

ABSTRACT

While much of the current study on quantum computation employs low-level formalisms such as quantum circuits, several high-level languages/calculi have been recently proposed aiming at structured quantum programming. The current work contributes to the semantical study of such languages by providing interaction-based semantics of a functional quantum programming language; the latter is, much like Selinger and Valiron's, based on linear lambda calculus and equipped with features like the ! modality and recursion. The proposed denotational model is the first one that supports the full features of a quantum functional programming language; we prove adequacy of our semantics. The construction of our model is by a series of existing techniques taken from the semantics of classical computation as well as from process theory. The most notable among them is Girard's *Geometry of Interaction (GoI)*, categorically formulated by Abramsky, Haghverdi and Scott. The mathematical genericity of these techniques—largely due to their categorical formulation—is exploited for our move from classical to quantum.

☆ An earlier version of this paper [37] has been presented at the Twenty-Sixth Annual IEEE Symposium on Logic in Computer Science (LICS 2011), 21–24 June 2011, Toronto, Ontario, Canada.

* Corresponding author.
E-mail addresses: ichiro@is.s.u-tokyo.ac.jp (I. Hasuo), naophiko@kurims.kyoto-u.ac.jp (N. Hoshino).
URLs: http://www-mmm.is.s.u-tokyo.ac.jp/~ichiro (I. Hasuo), http://www.kurims.kyoto-u.ac.jp/~naophiko (N. Hoshino).

[1] Supported by Grants-in-Aid No. 24680001, 15K11984 & 15KT0012, JSPS; by the JSPS-INRIA Bilateral Joint Research Project "CRECOGI"; and by Aihara Innovative Mathematical Modelling Project, FIRST Program, JSPS/CSTP.
[2] Supported by Grants-in-Aid No. 26730004 & 15K11984, JSPS; and by the JSPS-INRIA Bilateral Joint Research Project "CRECOGI".

http://dx.doi.org/10.1016/j.apal.2016.10.010
0168-0072/

1. Introduction

1.1. Quantum programming languages

Computation and communication using quantum data has attracted growing attention. On the one hand, quantum computation provides a real breakthrough in computing power—at least for certain applications—as demonstrated by Shor's algorithm. On the other hand, quantum communication realizes "unconditional security" e.g. via quantum key distribution. Quantum communication is being physically realized and put into use.

The extensive research efforts on this new paradigm have identified some challenges, too. On quantum computation, aside from a few striking ones such as Shor's and quantum search algorithms, researchers are having a hard time finding new "useful" algorithms. On quantum communication, the counter-intuitive nature of quantum data becomes an additional burden in the task of getting communication protocols right—which has proved extremely hard already with classical data.

Structured programming and *mathematically formulated semantics* are potentially useful tools against these difficulties. Structured programming often leads to discovery of ingenious algorithms; well-formulated semantics would provide a ground for proving a communication protocol correct.

In this direction, there have been proposed several high-level languages tailored for quantum computation (see [79] for an excellent survey). Among the first ones is QCL [63] that is imperative; the quantum IO monad [32] and its successor Quipper [33] are quantum extensions of Haskell that facilitate generation of quantum circuits. Closely related to the latter two is the one in [22], that is an (intuitionistic) λ-calculus with quantum stores.

Another important family—that is most strongly oriented towards mathematical semantics—is those of *quantum λ-calculi* that are very often based on *linear λ-calculus*. While λ-calculus is a prototype of functional programming languages and inherently supports higher-order computation, linearity in a type system provides a useful means of prohibiting duplication of quantum data ("no-cloning"). Examples of such languages are found in [14,16,31,73–75,81].

1.2. Denotational semantics of quantum programming languages

Models of linear logic (and hence of linear λ-calculus) have been studied fairly well since 1990s; therefore denotational models for the last family of quantum programming languages may well be based on those well-studied models. Presence of quantum primitives—or more precisely *coexistence of "quantum data, classical control"*—poses unique challenges, however. It thus seems that denotational semantics for quantum λ-calculi has attracted research efforts, not only from those interested in quantum computation, but also from the semantics community in general, since it offers unique and interesting "exercises" to the semantical techniques developed over many years, many of which are formulated in categorical terms and hence are aimed at genericity.

Consider a quantum λ-calculus that is essentially a linear λ-calculus with quantum primitives. It is standard that *compact closed categories* provide models for the latter; so we are aiming a compact closed category 1) with a quantum flavor, and 2) that allows interpretation of the ! modality that is essential in duplicating classical data. This turns out to be not easy at all. For example, the requirement 1) makes one hope that the category **fdHilb** of finite-dimensional Hilbert spaces and linear maps would work. This category however has no convenient "infinity" structure that can be exploited for the requirement 2). Moving to the category **Hilb** of possibly infinite-dimensional Hilbert spaces does not work either, since it is not compact closed.

A few attempts have been made to address this difficulty. In [74] a categorical model is presented that is fully abstract for the !-free fragment of a quantum λ-calculus is presented. It relies on Selinger's category