Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 319 (2015) 217–237

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Reversible Monadic Computing

Chris Heunen[1,2]

Department of Computer Science
University of Oxford
United Kingdom

Martti Karvonen[3]

Department of Mathematics and Systems Analysis
Aalto University
Finland

Abstract

We extend categorical semantics of monadic programming to reversible computing, by considering monoidal closed dagger categories: the dagger gives reversibility, whereas closure gives higher-order expressivity. We demonstrate that Frobenius monads model the appropriate notion of coherence between the dagger and closure by reinforcing Cayley's theorem; by proving that effectful computations (Kleisli morphisms) are reversible precisely when the monad is Frobenius; by characterizing the largest reversible subcategory of Eilenberg–Moore algebras; and by identifying the latter algebras as measurements in our leading example of quantum computing. Strong Frobenius monads are characterized internally by Frobenius monoids.

Keywords: Frobenius monad, dagger category, reversible computing, quantum measurement

1 Introduction

The categorical concept of a *monad* has been tremendously useful in programming, as it extends purely functional programs with nonfunctional effects. For example, using monads one can extend a functional programming language with nondeterminism, probabilism, stateful computing, error handling, read-only environments, and input and output [51]. Haskell incorporates monads in its core language. On the theoretical side, there are satisfyingly clean categorical semantics. Simply typed λ-calculus, that may be regarded as an idealized functional programming language,

[1] Supported by the Engineering and Physical Sciences Research Council Fellowship EP/L002388/1. We thank an anonymous referee for Example 6.4, and Jorik Mandemaker, Sean Tull, and Maciej Pirog for helpful discussions.

[2] Email:heunen@cs.ox.ac.uk

[3] Email:martti.karvonen@aalto.fi

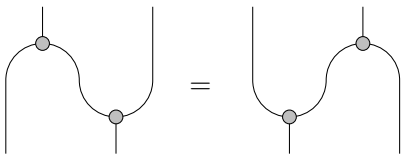
http://dx.doi.org/10.1016/j.entcs.2015.12.014

takes semantics in Cartesian closed categories [31]. The functional programming concept of a monad is modeled by the categorical concept of a monad [36].

In classical computation it is not always possible to reconstruct the input to an algorithm from its output. However, by using auxiliary bits, any classical computation can be turned into a *reversible* one [48]. Such a computation uses invertible primitive gates, and composition preserves invertibility. As discarding information requires work, reversible computations could in principle be implemented at higher speeds. The only operation costing power is the final discarding of auxiliary bits.

This is brought to a head in *quantum computing*, where any deterministic evolution of quantum bits is invertible, unlike the eventual measurement that converts quantum information to classical information. Another novelty in quantum computing is that it is impossible to copy or delete quantum information. This leads to a linear type theory of resources rather than a classical one [47]: quantum computing takes semantics in monoidal categories, rather than Cartesian ones [2].

Led by quantum computing, this article extends the categorical semantics of monadic programming to reversible computing. To allow for a linear type theory we consider monoidal closed categories. To allow for reversible computations, we consider *dagger categories*; in general these correspond to bidirectional computations rather than invertible ones, which in the quantum case comes down to the same thing. To allow for monadic effects, we introduce *Frobenius monads*. In the presence of a dagger, any monad gives rise to a comonad; a Frobenius monad is one that interacts with its comonad counterpart via the following *Frobenius law*:



(1)

Here we used the graphical calculus for monoidal categories [44,34], that will be explained further in Section 2, along with several examples. [4]

Our main contribution is to take reversal as a primitive and so justify the claim that Frobenius monads are precisely the right notion as follows:

- Section 3 justifies the Frobenius law as a necessary (and sufficient) consequence of coherence between the dagger and closure. In a reversible setting, it is natural to consider *involutive* monoids. In a monoidal closed category, any monoid embeds into a canonical one by Cayley's theorem. We prove that this embedding preserves the involution induced by the dagger if and only if the monoid satisfies the Frobenius law. This derivation from first principles is a noncommutative generalization of [41, Theorem 4.3] with a new proof.
- Section 4 characterizes Frobenius monads *internally*. Monads are an *external* notion. A good example is the writer monad, that allows programs to keep auxiliary

[4] We often need to reason simultaneously about morphisms *in* a monoidal category and endofunctors *on* it. Unfortunately there is no sound and complete graphical proof calculus that would handle this yet. Therefore we cannot use the graphical calculus exclusively and also have to use traditional commutative diagrams.