Journal of Logical and Algebraic Methods in Programming ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Equational reasoning with lollipops, forks, cups, caps, snakes, and speedometers

Ralf Hinze *, Dan Marsden

Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford, OX1 3QD, England, United Kingdom

ARTICLE INFO

Article history:
Received 2 June 2015
Received in revised form 9 September 2015
Accepted 18 December 2015
Available online xxxx

Keywords:
String diagram
Calculational proof
Category theory
Adjunction
Resumption monad

ABSTRACT

Proofs in elementary category theory typically involve either the pasting together of commuting diagrams or calculational reasoning using chains of equalities. Much of the effort in these styles can be consumed with trivial administrative steps involving functoriality, naturality, and the handling of identities. As an alternative, we advocate the use of string diagrams, a two-dimensional form of notation which silently deals with these distracting bookkeeping steps. While originally developed to reason about functors and natural transformations, we show that string diagrams can also easily accommodate objects and arrows. Throughout the paper we emphasize how the topological freedom inherent in the notation can be exploited to aid the use of geometric intuition in the development of proofs. Indeed, drawing string diagrams is a bit of an art: good diagrammatic choices can make all the difference.

1. Introduction

Proofs in elementary category theory typically involve either the pasting together of commuting diagrams [7] or calculational reasoning using chains of equalities [4]. The first style is sometimes referred to as “diagram chasing” since the focus of a proof is chased around a diagram; the second is also known as “squiggolling” since it often involves the use of “squiggly” symbols.

Both styles have their merits. Commutative diagrams capture an abundance of type information, and invoke a certain amount of visual intuition. Equational proofs are familiar from many other branches of science and mathematics. They are also compact and carry a clear orientation from assumptions to goals.

Unfortunately, both styles also have serious limitations. The usual equational style of reasoning forces us to abandon the vital type information that guides proof attempts and protects against errors. Further, much of the effort in these proofs can be consumed with trivial administrative steps involving functoriality, naturality, and the introduction and elimination of identities. Commuting-diagram-style proofs retain the type information, but the proof style is unfamiliar to many in other fields of mathematics and computer science, and the resulting proofs often lack motivation and a clear direction of argument. We are often led towards proofs that reason using natural transformations component-wise, again leading to distracting administration of naturality conditions.

In order to recover the best of both these approaches, we advocate the use of *string diagrams* [8], a two-dimensional form of notation, that retains the vital type information whilst permitting an equational style of reasoning. This notation

* Corresponding author.
E-mail addresses: ralf.hinze@cs.ox.ac.uk (R. Hinze), daniel.marsden@cs.ox.ac.uk (D. Marsden).

http://dx.doi.org/10.1016/j.jlamp.2015.12.004
2352-2208/

silently deals with distracting bookkeeping steps, such as naturality and functoriality issues, leaving us to concentrate on the essentials. This is an important aspect in any choice of notation, as advocated by Backhouse [1]. The resulting diagrams and proofs are highly visual and we can often exploit topological intuition to identify suitable steps in our reasoning. Indeed, developing and drawing string diagrams is a bit of an art: good diagrammatic choices can make the difference between a tangled knot and a regular weave.

Our approach will be by example; after introducing string-diagrammatic concepts in Section 2 we will apply them to calculations in elementary category theory, involving classical concepts such as monads and distributive laws in Section 3, and adjunctions and adjoint squares in Section 5. String diagrams were initially developed to reason about functors and natural transformations. In Section 4 we show how to incorporate objects and arrows into the string-diagram notation, using algebras and homomorphisms as running examples. We present a worked-out application in Section 6, revolving around the so-called resumption monad, and conclude in Section 7.

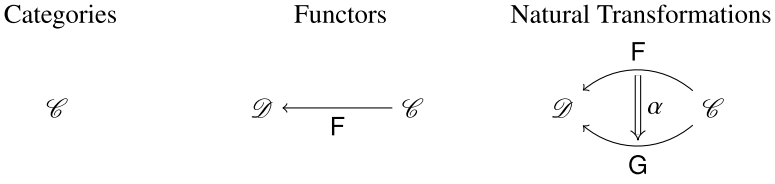
Finally, we should emphasize that we concentrate on the categorical concepts. In particular, we do not provide any applications of the constructions introduced.

2. Basics of string diagrams

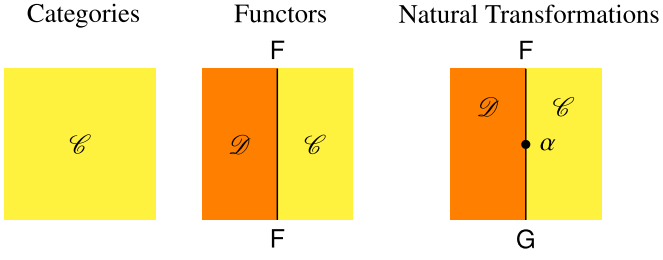
Elementary category theory is concerned with categories, functors, and natural transformations. As described in Mac Lane [7]:

> … "category" has been defined in order to be able to define "functor" and "functor" has been defined in order to be able to define "natural transformation".

The traditional notation for these entities represents categories by vertices, functors by arrows between vertices, and natural transformations as double arrows in the region between functor arrows, as illustrated in the following table:

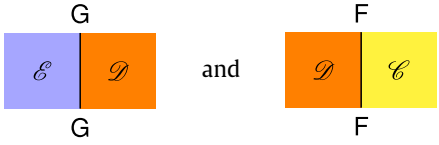


String diagrams are the *topological* or *Poincaré dual* of the traditional notation. Natural transformations are now vertices, functors are edges connecting natural transformations, and categories are regions between the functor edges:

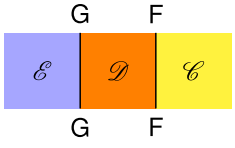


In a sense, string diagrams return the visual focus to the conceptually more significant natural transformations. This seemingly trivial change in perspective provides category theory with a new and very distinctive visual flavor. We aim to demonstrate that string diagrams provide an effective tool for equational reasoning about elementary category theory.

In order to approach practical problems, it is important to be able to use *composition* to construct more complex structures and equations between them. Given functors $\mathsf{F} : \mathscr{C} \to \mathscr{D}$ and $\mathsf{G} : \mathscr{D} \to \mathscr{E}$, in pictures:



their composite $\mathsf{G} \circ \mathsf{F} : \mathscr{C} \to \mathscr{E}$ is drawn horizontally as follows:



Please cite this article in press as: R. Hinze, D. Marsden, Equational reasoning with lollipops, forks, cups, caps, snakes, and speedometers, J. Log. Algebraic Methods Program. (2016), http://dx.doi.org/10.1016/j.jlamp.2015.12.004