Cartesian Closed Categories and Functional Programming

R. A. Kycia[1]

[1]Faculty of Science Masaryk University

May 15, 2018

Abstract

The lecture presents the route from Cartesian Closed Category to λ-Calculus and Monads. The presentation will be focused on understanding of ideas and solid motivation - the detailed exposition can be found in the References.

Tutorial

Topics presented:

1. **Cartesian product and category with all finite products - reminder.**
   Bibliography: Chapters 7 and 8 of [6]; Chapter 5 of [1];

2. **Exponential object and Cartesian Closed Category.**
   Bibliography: Chapter 13 of [6]; Chapter 1 of [5]; Chapter 6 of [1];

3. **Simply typed λ-calculus - a sketch.**
   Biblography: For basic presentation see Chapter 6 of [1]; For detailed presentation see Chapter 2 of [5];

4. **λ-calculus in action.**
   Bibliography: Detailed exposition with applications can be found in [4]; For a brief summary see [3];

5. **Connecting functional and non-functional world - Monads and Kleisli category** - this part was not presented due to limited time;
   Informal introduction: In functional programming the program consists of applications of functions on some data. The monads are used to isolate mathematical functions (which for a given data give always the same value) from 'functions' that can return on the same input data different values depending on circumstances (e.g., function that provides user data from a standard input which input data is the standard input device representation, and which asks user to input some number can give: a number, a non-number - possibly error or an input-output error depending on the state of the device and user input). The monad in Functional programming is in fact Kleisli category build on monad definition. The route (one of many) to understanding them is to follow the path in the Bibliography.
   Bibliography: Start from graphical illustration of monad action in Haskell-like programming language from [2], then see formal definition of monads in functional programming in [7] (especially the Chapter 'Formal definition') and correlate it with the definition of composition law of the Kleisli category (the Kelisli composition) in [8]; See also Chapter VI of [9];

References

[1] Michael Barr, Charles Wells, *Category Theory for Computing Science*, Prentice-hall International Series in Computer Science, 1990

[2] Aditya Bhargava, Functors, Applicatives, And Monads In Pictures (15.05.2018)

[3] Yan Huang, Slides for the CS302 course at the University of Virginia (15.05.2018)

[4] Greg Michaelson, *An Introduction to Functional Programming Through Lambda Calculus*, Dover Books on Mathematics, Dover Publications 2011

[5] Alfio Martini, *Category Theory and the Simply-Typed λ-Calculus*, Report No. 96-7, Technische Universität Berlin (1996), Report available at `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.5806` (15.05.2018)

[6] Peter Smith, *Category Theory: A Gentle Introduction*, University of Cambridge, Manuscript available at: `http://www.logicmatters.net/categories/` (15.05.2018)

[7] Monads (functional programming); Wikipedia (15.05.2018)

[8] Kleisli category; nLab (15.05.2018)
Kleisli category; Wikipedia (15.05.2018)

[9] Saunders Mac Lane, *Categories for the Working Mathematician*, Springer; 2nd ed., 2010