Proposed Course:
Category Theory in Programming
by Sergey Cherkis

Programming, in fact much of human problem solving or mathematical construciton of proofs, consists of decomposing a given problem into smaller subproblems, then composing the solutions of these subproblems. The key step here is the composition. That is the reason for the recent rise of category theory constructs in very practical programming tasks, since category theory is exactly the mathematical framework for composing.

This course will consists of an introduction to category theory with examples in Haskell programing language.

Prerequisites:

Math323 or Math432 or Math534

Texts:

- Richard Bird, "Thinking Functionally with Haskell," Cambridge
- Bartosz Milewski, "Category Theory for Programmers"
  https://github.com/hmemcpy/milewski-ctfp-pdf
- Sanders Mac Lane, "Categories for the Working Mathematician," Springer
  https://link.springer.com
- Benjamin C. Pierce, "Basic Category Theory for Computer Scientists (Foundations of Computing)," The MIT Press
  ISBN-10 : 0262660717 and ISBN-13 : 978-0262660716

Approximate Schedule

(One week per item.)

- Categories (definition and examples). Types and Functions.
- Pure functions and effects.
- Monoids
- Kleisli Categories and Monads.
- Universal Constructions: products and coproducts, algebraic data types.
- Functors and Natural transformations: Functor, Applicative, Monad.
- Functoriality, covariant and contravariant functors, Reader and Writer Monads.
- Limits and colimits
- Free monoids
- Yoneda lemma.
- Yoneda embedding.
- Adjunction.
- (Time permitting) Kan extensions.

Learning Outcomes

By the end of this course the students are expected to be able to

- use algebraic data types, type classes, monads and monad transformers for write programs.
- use equational reasoning to write proofs.
- use higher order functions to express common patters of computation.