

Monads (and their categorical dual – comonads) are important concepts in category theory and while monads enjoy their popularity in functional languages (mainly due to the programming language Haskell), comonads are often forgotten. In this work we present a definition of comonads suitable for programming and give examples of their use. One of the more important examples is zipper – a structure used to represent position. We show that zipper can be automatically derived for any regular type and show that this operation is very reminiscent of derivative from mathematical analysis. We also show worked examples of various problems that comonads can help solve in the language Haskell. All relevant proofs for the theoretical part of this work are machine checked in the language Agda.