We show that techniques for monad composition can be used nicely for modelling object-oriented programming concepts. In this functional setting, we develop a new model for composing objects from individual features in a modular way. Features are similar to abstract subclasses, but separate the core functionality of a subclass from overwriting methods. We view method overwriting more generally as resolving interactions between two features. The interaction handling is specified separately and added when features are composed. This generalizes inheritance as found in object-oriented languages and leads to a new view of objects in a functional setting. Our concepts are implemented in Gofer and generalize some monadic programming techniques, where objects correspond to monads, features to monad transformers, and feature interactions are resolved by lifting functions through monad transformers.
entries: