From Monads to Programming with Monads

Presenter: Yihe Huang

The Essence of Functional Programming

```
@inproceedings{wadler1992essence,
  title={The essence of functional programming},
  author={Wadler, Philip},
  booktitle={Proceedings of the 19th ACM SIGPLAN-SIGACT symposium on
Principles of programming languages},
  pages={1--14},
  year={1992},
  organization={ACM}
}
```

Summary

In this paper, Philip Walder introduced to the audience how to design modular programs using the concept of Monads. The paper shows how to use monads to construct modular programs, such that adding new features to the existing program feels natural and non-disruptive. Without going deep into the category theoretic origin of the concept, Philip demonstrated the powerfulness of Monads by going through an example of implementing an interpreter for a small language. By starting from a monadic construct, the paper shows that features such as error checking, state tracking, input/output, etc. can be added to the interpreter without modifying the structure of the interpreter program; only the monads involved need to be modified and their corresponding bind operators be redefined. Monads can be used to represent non-allowed entities in pure languages such as side effects and state, and elevating them to be visible in the type system, making it convenient to reason about such effects.

Evaluation

Philip Walder's running example of the interpreter manages to convey key intuitions about the concept to an unfamiliar audience. The interpreter is a special program, but it's one of those programs that invoke side-effects quite frequently and therefore a pretty good fit for the topic. By showing that side-effects, which usually need to be handled in an ad-hoc way in many programming languages, can be encoded as part of the structure of the program using Monad, the paper laid the foundation for the incorporation of Monads in Haskell, and eventually many other programming languages. The paper itself, however, still leaves many questions open, such as how can one combine two monads. Nevertheless, it is a good read and one of the best introductions to monads one can find.