(MR3300308) REVIEW OF "F-ING MODULES" BY ANDREAS ROSSBERG, CLAUDIO ROSS AND DEREK DREYER (2015)

DANKO ILIK

Functional programming languages can be seen as versions of lambda caluclus suitable for writing programs that will actually run on a computer. However, besides only a type system for the implemented lambda calculus, functional languages need to add a system for encapsulation of sub-routines, called a module system, that is indispensable when combining sub-routines into large programs.

A module system presents another kind of type system on top of the type system of the underlying lambda calculus, but this super-type system is usually not studied in terms of orthodox proof theoretic calculi. The paper under review is a contribution in this direction. It shows how to give a direct encoding of most features of module systems of existing functional programming languages from the ML family, in terms of System F_ω, a relatively well studied extension of Girard's and Reynold's System F which is in turn in Curry-Howard correspondence to minimal intuitionistic second-order logic. The encoding however leaves out the possibility of defining recursive modules.

The paper is also valuable as a survey of previous works in the semantics of module systems.