Abstract
It is a well known fact that there is a correspondence between propositions and type, and similarly a correspondence between a proof of a proposition and a program of a type; this is the Curry-Howard correspondence. In this talk I will describe a program, Djinn, which takes a Haskell type and produces a program of that type, using the Curry-Howard correspondence. For the subset of Haskell types that Djinn can handle (no recursive types) it does quite well. For instance, it can derive the code for all the standard monads in Haskell, including continuations and call/cc.