

Dependent type systems provide a novel way of reasoning about program correctness, by embedding behavior of the program into the more expressive type system. Correctness is achieved by not allowing incorrect states to be representable. Languages like Idris show that dependent type systems are practically useful, not only for formal proofs, but also for creating fewer bugs in production. But the purity of computation poses a problem for composability of stateful computations and of side effects. Effect handlers provide one possible solution for this problem. In this thesis we propose an effect extension of dependent type systems. The resulting system not only makes it possible to provide guarantees about correctness of a program, but also make it easy to compose such guarantees using effects. We formalize the type system and present a prototype implementation.