

The template system of C++ is expressive enough to allow the programmer to write programs which are evaluated during compile time. This can be exploited for example in generic programming. However, these programs are very often hard to write, read and maintain. We introduce a simple translation from lambda calculus into C++ templates and show how it can be used to simplify C++ metaprograms. This variation of lambda calculus is then extended with Hindley-Milner type system and various other features (Haskell-like syntax, user-defined data types, tools for interaction with existing C++ template code and so on). We then build a compiler capable of transforming programs written in this language into C++ template metaprograms.