7th International Conference on Applied Informatics
Eger, Hungary, 2007.

Properties of C++ Template Metaprograms*

Norbert Pataki[a], Tamás Kozsik[a], Zoltán Porkoláb[a]

Dept. of Programming Languages and Compilers,
Fac. of Informatics, Eötvös Loránd University, Budapest
e-mail:{patakino, kto, gsd}@elte.hu

Abstract

Verifying properties of programs is a common way to ensure the proper behaviour of those programs. *Invariants, pre-* and *postconditions* are program properties often used when proving correctness of programs.

C++ template metaprograms (TMPs) are special progams interpreted by the compiler. Metaprograms are widely used for the following purposes: executing algorithms in compile-time, optimizing runtime programs and emitting compilation errors and warnings to enforce certain semantic checks.

In this paper we step to "meta-meta-level": we present a technique to make safer C++ TMPs with static asserts. We describe how to check invariants, pre- and postconditions of TMPs and enforce the compiler to refuse metaprograms if any of the specified program properties is dissatisfied. We present some examples where semantic errors in TMPs are revealed by our method.

Keywords: template metaprogramming, program correctness

MSC: 68N19 Other programming techniques

Norbert Pataki
Pázmány Péter sétány 1/c., H-1117 Budapest, Hungary

Tamás Kozsik
Pázmány Péter sétány 1/c., H-1117 Budapest, Hungary

Zoltán Porkoláb
Pázmány Péter sétány 1/c., H-1117 Budapest, Hungary

*Supported by GVOP-3.2.2.-2004-07-0005/3.0