

# Improving a C++ Static Code Analyzer

## 1. Motivation

In C++, operator overloading can be used to replace built-in types (e.g., *double*) with user-defined ones.

This is done to introduce new semantics to an existing code base, for example, multi-precision data types (*Boost.Multiprecision*) but also algorithmic differentiation types:

<pre>1 double f(double x) { 2   double y = x * x + x; 3   return y; 4 }</pre>		<pre>1 #include "adouble.h" 2 adouble f(adouble x) { 3   adouble y = x * x + x; 4   return y; 5 }</pre>
---	---	---

Fig. 1: *double* is replaced by the user-defined type *adouble*

However, this type change leads to compiler errors. This is because the C++ standard treats user-defined types differently compared to built-ins. As such, the developer is forced to fix several problematic code statements.

We have developed a static code analyzer, based on the Clang/LLVM compiler infrastructure, to find such problematic code constructs before the type change. This enables the developer to fix the problems without interpreting thousands of lines of compiler error output.

## 2. Areas of Work

### 1. Matching Capabilities

The static code analyzer produces false positives in some cases.

You would write unit tests and apply the analyzer to real code, identifying false positives and fixing the code matcher.

### 2. Source Transformation

In addition to identifying problematic code constructs, the matcher can also automatically transform the relevant source code for automatic fixes.

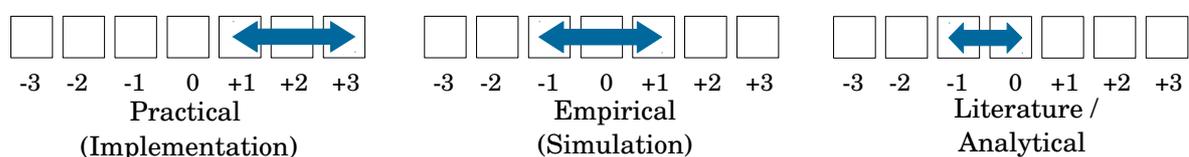
You would improve the source transformation system to correctly apply the necessary fixes.

## 3. Technologies

The matcher is publicly available under: <https://github.com/ahueck/opovlint/tree/devel>

Key technologies are **C++11** and the **Clang/LLVM compiler framework**.

You will gain knowledge about a state-of-the-art compiler infrastructure and how to leverage it's libraries to write your own C++ code analysis tools.



## Contact

Alexander Hüeck, M. Sc.  
S1|22 Room 412  
Tel. +49 6151 16-7 55 77  
[alexander.hueck@sc.tu-darmstadt.de](mailto:alexander.hueck@sc.tu-darmstadt.de)



Date  
06/09/2017