



Compiler Construction

Object-oriented Type Systems

Abstract

This presentation will focus on type systems in object-oriented programming languages. Type systems compare data types and operations between them and generate type errors. They are important, because they can make programs more readable, reliable and efficient than untyped languages. We will be talking about static and dynamic type checking and its realization using constraints and dynamic checks.

Then we will focus on three features that object-oriented programming provides and can be problematic for type systems. These are type inference, inheritance and genericity. Type inference allows deduction of a type based on context. Inheritance and genericity mechanisms enable construction of new classes that are just slightly different than already existing classes without otherwise necessary code duplication. Inheritance allows addition of variables and methods, and if a parent method is accessible, its body can be overridden. Genericity allows substitution of type annotations. All of these features bring problems, that type systems need to solve. We will present methods, which can handle these problems.

Authors: Beránek Michal,
Polach Radim,

xberan38@stud.fit.vutbr.cz
xpolac33@stud.fit.vutbr.cz

Faculty of Information Technology
Brno University of Technology