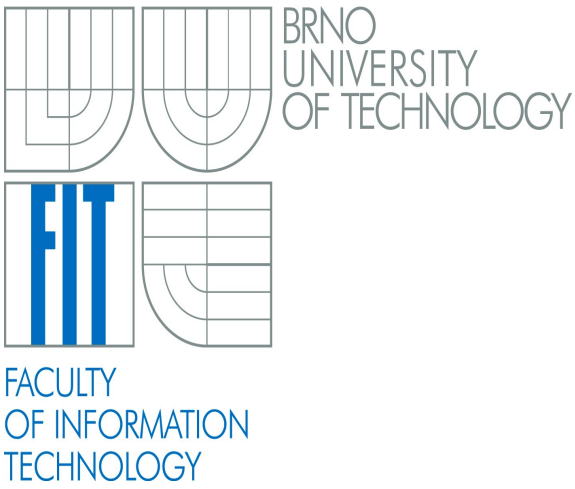

Compiler construction

Abstract of the presentation

Name	E-mail
Ivan Šišák	xsisak02@stud.fit.vutbr.cz
Jakub Randa	xranda00@stud.fit.vutbr.cz

1 Abstract

The compilation of the source code is not only a transformation code of one language to another language for example assembly language. In the past and in the present the main goal is transform code and make it faster. For this reason the developers of the compilers focus on the optimization of the source code. They try to remove parts of the source code which could be removed and the result of the program will be same.

First method is watching the data flow. It means developers watch how data are created, how variables change their values, where variables are destroyed and so on. They want to analyze the code and remove the unnecessary parts.

How to do it is interprocedural analysis. In our presentation we will describe another way of the analysis. We will introduce the dataflow in a logical representation. And we create rules which help (predicates) us to define the algorithm for Data-flow analysis.

For logical representation we use Datalog which is language similar to Prolog. All rules will be introduced in a example.

Another topic we will discuss in our presentation is the simple pointer analysis with no procedure calls using the interprocedural analysis. We will describe the difficulties in pointer analysis, especially those related to the object-oriented programming, such as handling the polymorphism.

Then we will define a simple model for pointers (in our case the C language pointers) and references (Java), which will be used in further examples. Considering the data flow, there are some differences between these two, like that C pointer variables can point to another pointer variables. Because of that, we will use Java references in following examples.

Using the Java example, we describe the difference between flow sensitive and flow insensitive analysis, we will discuss pros and cons of both of them. The most important theme of this part of our presentation will be the expressing whole analysis using Datalog rules. Again, we will demonstate this on an example.

The last topic of the pointer analysis part of our presentation will be the type information and its use in pointer analysis. We will show how to use this type information on the previous example, where we will modify Datalog rules, so we ensure proper type-dependent variable assigning.