OPTIMIZING COMPILERS: PRELIMINARY TRANSFORMATIONS

Paulík Miroslav, Bc.
xpauli00@stud.fit.vutbr.cz

Chernikava Alena, Ing.
xcerni07@stud.fit.vutbr.cz

Abstract: Dependence testing is used for analyzing dependences between statements in a program. Moreover, for all tests there are some requirements such that an input program must fit some standard form. Unfortunately, programs are not typically written with dependence testing in mind, therefore compiler must handle these issues. To address problems of this sort, a number of transformations can be applied prior to dependence testing with the goal of making testing more accurate.

These transformations can make many more subscripts easier to implement and amenable to accurate testing. In addition, preliminary transformations are used not only for dependence testing but also in other phases (for example in optimization). Here we will describe algorithms for some preliminary transformations and apply in practice some of them.

In our presentation, we will give an introduction to these preliminary transformations. We will focus on a Loop normalization and a Dead code elimination algorithms. If there will be still some time left, we will end up with a basic introduction to some other transformations such as a Constraint Propagation algorithm.