Optimizing compilers – dependence testing

The main motivation for parallelization of compilers is, as always, speed. We would like to use standard programming languages (C, C++, Java etc.) on common parallel computers. This leads to series of problems we need to deal with. For instance, how do we detect that the program is fit to be run in parallel? If it is, how do we use this information on a specific architecture? How do we use available components of this architecture so that we exploit the advantages of parallelism to the maximum?

This presentation focuses on detecting data dependencies in source programs. Basically the testing focuses on loops and data dependencies between loop cycles. If a loop does not have data dependency between any two iterations, then it is possible to execute it in parallel. If there are dependencies, then we might want to try to remove the dependencies or to run the compilation in parallel, even if the dependencies cannot be removed.

The presentation presents theoretical foundations which are necessary for understanding dependencies and how to work with them. It also includes examples to demonstrate the practical aspects of this issue.

Michal Ruprich, xrupri00@stud.fit.vutbr.cz
Martin Vymlátil, xvymla01@stud.fit.vutbr.cz