Loops in Flow Graphs

Jakub Barteček (xbarte09@stud.fit.vutbr.cz)
Ondřej Kanich (xkanic00@stud.fit.vutbr.cz)

This presentation deals with the problem of finding loops in flow graphs. This information is very useful in machine-independent optimizations, because programs spent a lot of time by executing loops. Specifically, we will talk about the algorithmic detection of natural loops (the special type of loops that covers most loops in real programs). To do that, we will need information about the domination relation in flow graphs and about back edges. The domination relation is used to determine the entry point of loops. In the presentation we will show the way of construction this relation for every flow graph.

To find back edges we will need information how to construct Depth-First Spanning Tree (DFST). When we transform Flow Graphs into DFST then we can classify edges in this tree into three classes. One of these classes is called retreating edges. If certain condition is met we call these retreating edges back edges. With use of the back edges we can determine whether DFST is reducible or not.

All methods that will be described can be written also algorithmically. With all these methods we can determine natural loops.

References

AHO, A. V., LAM, M. S., SETHI, R., ULLMAN, J. D.: *Compilers, Principles, Techniques and Tools*. Canada: Pearson Education, 2006, s. 74-90. ISBN 0-32148-681-1.