# Scalable and concurrent flow analysis for Go

December 2016

| | |
|---|---|
| candidate | Andreas Fladstad |
| supervisors | Volker Stolz and Martin Steffen |
| group | PMA |
| type | 60 ECTS |
| study program | computer science |
| planned date of completion | May 2018 |

## Short description

The task is to design, develop and implement an effenct program analysis for Go, with the purpose increased efficiency and scalability, using Go's concurrency features.

## Background and motivation

Go [3] is a relatively new general-purpose language, which gained traction recently, being backed by Google and being used to implement applications such as Docker and dropbox. Recently a taint analysis as a special form of flow analysis has been implemented [1]. Data flow analysis, including taint analysis is a standard way of inferring relevant properties about programs. As usual, such static analyses have to make compromises between efficiency and scalability on the one hand and precision on the other. This thesis will explore approaches to improve the efficiency of data flow analysis resp. taint analysis for go, in particular

## Problem setting

To achieve the above goals, the thesis plans to address the following point.

- review the current approaches for paralellizing flow analyses and approaches for scalability.
- design a corresponding analysis and implement it.
- exploit the possibility of method summaries for "caching" intermediate analysis results.
- run experiments on hand-made examples, resp. existing open-source software projects, gaining empirical results
- contribute to publications/conference submissions.

The work is done in connection with the *GoRETech*-project.

**Keywords:** program analysis, concurrency, Go language, security, taint analysis, parallelization

## References

[1] E. Bodden, V. K. I. Pun, V. Stolz, M. Steffen, and A. K. Wickert. Information flow analysis for Go. 2016. Accepted for publication.

[2] A. A. A. Donovan and B. W. Kernighan. *The Go Programming Language*. Addison-Wesley, 2015.

[3] The Go programming language specification. `https://golang.org/ref/spec`, Nov. 2016.

[4] M. Summerfield. *Programming in Go*. Addison-Wesley, 2012.