



Master-Arbeit Nr. 1058

**Entwurf und Implementierung eines Parsers, Verifiers und Interpreters  
für den Zwischencode in einem Compiler für eine experimentelle Variante  
der Programmiersprache Go**



**Methoden**

Programmierung  
Compilerbau

**Themengebiete**

Rechnerarchitektur  
Compiler

**Hintergrund**

Am Institut wird derzeit eine experimentelle Variante der Programmiersprache Go mit dem Namen GoSUB entwickelt, um damit ein Betriebssystem für einen objektbasierten Prozessor zu realisieren. Es existiert bereits ein GoSUB-Compiler, dessen Frontend Quelltexte zunächst in eine stapelbasierte Zwischendarstellung (Intermediate Representation) umsetzt. Ein Backend liest diesen vom Frontend generierten Zwischencode ein, führt ihn in die SSA (Static Single Assignment Form) über und generiert daraus C-Code, der von einem C-Compiler (GCC) schließlich in nativen Code für einen PC übersetzt wird. Sowohl Frontend und Backend des Compilers sind in Java geschrieben.

**Aufgabenstellung**

In einer vorausgegangenen Arbeit wurde das Frontend des Compilers neu in GoSUB implementiert. In der Arbeit sollen nun weitere Komponenten neu in GoSUB implementiert werden. Dabei sind die Ziele im Einzelnen:

- mit Hilfe von bestehenden Scanner- und Parsergeneratoren einen Parser für den vom Frontend erzeugten Zwischencode implementieren
- den eingelesenen Zwischencode nach dem Vorbild des Bytecode-Verifiers der JVM verifizieren
- einen Interpreter für den Zwischencode implementieren, um GoSUB-Code tatsächlich ausführen zu können

Bei der Implementierung der Komponenten soll nicht einfach der bestehende Java-Code in GoSUB-Code übersetzt werden. Stattdessen sollen die von GoSUB bereitgestellten Mechanismen und Abstraktionen bestmöglich zur Modellierung des Compiler-Frontends eingesetzt werden.

**Kontakt**

Dipl.-Ing. Matthias Meyer  
Raum 1.334 (ETI II), Telefon 685-67975, E-Mail [matthias.meyer@ikr.uni-stuttgart.de](mailto:matthias.meyer@ikr.uni-stuttgart.de)