

## **Zusammenfassung**

Obwohl die Assemblersprache größtenteils durch Hochsprachen ersetzt wurde, ist es dennoch sinnvoll die Assemblerprogrammierung zu lehren, um ein Verständnis für die Interaktion mit Computerhardware zu bekommen. Dabei muss sich jedoch mit Konzepten befasst werden, die bei Hochsprachen vom Compiler übernommen werden, wie die Verwaltung des Stacks und die Nutzung von Registern. Das kann zu einer Vielzahl an unbekanntem Fehlern führen, die in Hochsprachen nicht auftreten. Daher befasst sich diese Arbeit mit der Klassifizierung von Fehlern, die in studentischen RISC-V Assemblerübungsabgaben gemacht wurden, sowie der Entwicklung eines Programms, das mithilfe statischer und dynamischer Codeanalyse automatisch Fehler in den Bereichen Datenzugriffe und Calling Convention erkennt. Die Klassifizierung erfolgt in drei Hauptkategorien: Syntaxfehler, Logikfehler und Semantikfehler. Die statische Analyse verwendet eine abgewandelte Form der Konstantenweitergabe, um Arrayzugriffe zu überprüfen. Für die dynamische Analyse wird der RARS Simulator erweitert, um Datenzugriffe und Calling-Convention-Verstöße während der Laufzeit zu überwachen. Die Analyse von 706 studentischen Abgaben ergibt eine Übersicht über die häufigsten Fehler, sowie Korrelationen zwischen verschiedenen Fehlern. Es können sogar einige Abgaben als fehlerhaft erkannt werden, die von Tutoren fälschlicherweise mit der vollen Punktzahl bepunktet wurden.