Sage Strieker

Dr. Johnson

CMSI-401

9 September, 2020

Proposal for Error Handling Improvements in Swift Compiler Analysis Project

I had the opportunity this summer to connect with a woman named Holly Borla, who works at Apple on the Swift compiler. She offered me the opportunity to be her mentee so that I could learn how to make meaningful contributions to the open-source code that makes up the Swift compiler. The compiler is written in C++, and it uses a recursive-descent parser, a variety of tools and algorithms to semantically analyze the abstract syntax tree that results from parsing, and LLVM for code generation. That being said, sometimes after a program is passed to the compiler, the error messages resulting from semantic analysis are not very readable because of how the constraint system is generated as it determines the type of each expression for type inference. If an error is detected, something called a constraint fix occurs, meaning that one of the constraints in the constraint system is modified to log that failure. This log is eventually made into something called a diagnostic, which is made visible to the user. However, error messages that get surfaced to the user can be vague or misleading, which affects the swift programmer’s work. This is true for constraint fixes on features that reduce boilerplate code, such as property wrappers and function synthesizers. My goal for this project is to enhance swift programmers’ experience by improving the error messages that get reflected in constraint fixes for function synthesizers and property wrappers.

This project relates to a variety of different skills that I have learned throughout my time at LMU. It will give me a better understanding of the intricacies of Swift, which we learned about in CMSI-386. It will give me the chance to apply what I learned in CMSI-488 about creating the front-end of a compiler. I get the opportunity to go more in-depth about type-inference, which was a concept we covered briefly in CMSI-488. I will apply my knowledge of different algorithms that I learned about in CMSI-485 and CMSI-282 to similar algorithms used in the compiler. This project has a significant impact! It will improve the experience that real software engineers have while writing in Swift on various platforms through diagnostics. This project is important in relation to compilers study because there is extensive research done on what errors are appropriate to reveal to users during parsing and semantic analysis. The Swift compiler code is open-sourced, and I already have all the internal tools needed to work on this project. Holly knows the technology for the compiler very well, so I won’t be at risk of getting lost while I’m working. She has confidence that I can complete my project to enhance the Swift compiler within the course of the semester. Swift has over 2 million users and it has a wide variety of meaningful applications in computer science education. It’s very rare to get the chance to work on an existing compiler, like Swift, that is so widely used by so many people. I don’t want to waste this opportunity!