A Compiler for Panda

(En oversætter for Panda)

Kian Banke Larsen

Advisor: Kim Skak Larsen

1.2.2023

A modern compiler is organized into phases, where the basic ones cover lexical and syntactic analysis, resulting in an abstract syntax tree. Subsequent phases analyze and adorn the abstract syntax tree, building a symbol table, performing type checking, and finally generating target code (assembler, for instance). This material is known from DM565 for a simple imperative language. The goal of this project is to go beyond this basic language and basic compiler.

In this project, we are interested in implementing such a language and extending it with additional types, including Booleans and some form of floats. Additionally, we may be interested in providing constructs for composite types in the form of records (structs) and some sequencing constructor such as arrays.

The main focus with regards to advanced techniques beyond a basic compiler will be register allocation, using techniques described in [1] or [2]. Handling this efficiently requires dataflow analysis via a flow-graph, construction of an interference graph, graph coloring, and translation back to instructions using a combination of the registers and the stack, when the available registers do not suffice.

The compiler will be written in Python, using a `flex/bison` equivalent package such as `ply` for scanning and parsing. The target language will be X86 assembler.

The report should document and discuss the developed compiler and the

interesting choices made in the process. It should be organized by phases and extensions beyond a basic compiler. The project will be evaluated on language design and usability, extent of advanced additions and phases, and maintainability. For the connection between the implementation and the report, the evaluation will focus on motivation, descriptions of advanced features relative to a basic language and compiler, and documented correctness and effect.

References

[1] Andrew W. Appel. *Modern Compiler Implementation in C*. Cambridge University Press, 1997.

[2] Keith Cooper and Linda Torczon. *Engineering a Compiler*. Elsevier, 3rd edition, 2022.