Compiler Construction

Fabio Stalder

2024

Abstract

Abstract Here

1. what is a compiler
    1.1. general idea
    1.2. how to implement
2. my different approach (QHS)
    2.1. define lots of macros, use lots of macros. W
    2.2. write direct assembly code
    2.3. only few responsibilities for Compiler
3. approach for comparison
    3.1. 3 compilers THS, QHS and gcc
    3.2. required features
        i. output as assembly
        ii. C like syntax
        iii. basic OOP features
        iv. every type defined as a class
    3.3. comparison
        i. ease of use
        ii. speed of compilation
        iii. speed of assembly output
        iv. customizability, freedom and possibility for extension
        v. error handling (required skill as a programmer)
4. building the compilers
    4.1. written in C++
    4.2. THS compiler
        i. compiler instructions in code

- ii. switch to assembly code in lexer
- iii. the Assign function problem (args for Assign function are set by calling Assign function)

4.3. QHS compiler

- i. super simple lexer
- ii. no parser
- iii. compiler instructions
  - A. orderStack
  - B. typeStack
  - C. defining identifiers
- iv. preambel defining all the stuff
  - A. simple shortcuts (braces, semicolons)
  - B. more complicated stuff (classes, functions)