Wilsey Compiler Theory: Final Project Writeup

Nathan Henry 8/3/2022

Compiler in Ada

Compiler Capabilities

My compiler successfully compiles and runs every test program given. The one program that I notice it runs with an error is the recursiveFibonacci program. I have tested other recursive programs successfully though. My compiler also gives meaningful output on all the incorrect programs except the one that uses the character '#' in error. Ada tries its best to be platform-generic with it's file reading and I struggled with newline and whitespace. I interpret any character that is not alphanumeric or underscore as whitespace.

What my Compiler does do:

- Completed Lexer generating tokens
- Completed Parser that creates a PNG file of the tree (run ./makegraph.sh)
- Generates LLVM Assembly Language to be run by the LLVM interpreter or run through Clang to produce a native binary
- Compiler Warnings:
  - Warning: Line: 19 Make sure that floats are specified with a .0
  - Missing Dot at the end of file, continuing anyways

- Compiler Errors:
  - Line: 10 Error: Variable: ZACH already declared in this scope.
  - Error: Type Mismatch on Line: 13. double /= i32
- Allows for strings of any size, a string assignment allocates a new slot of memory each time.

What My Compiler Does Not Do:

- String Comparisons: String comparisons of any length will always result to false.
- String Operations: Addition, Subtraction, Multiplication, and Division on strings are all undefined operations
- Implicit Type Casting. Using a literal without a decimal point will cause a warning and sometimes an error when referencing a variable defined as a Float. Assigning a float to an integer will result in a compiler error as shown above.
- Resynchronization is very limited, for instance an incorrectly formed IF statement will send the entire parser to failure. If I more wisely used my time, this would have been a rewarding part to work on.
- GetBool() only accepts 1's and 0's as the input, as anything else would require the use of string comparisons. A quick fix could be to add a message such as "Choice? 0: False, 1:True"