

Using Graphvis and the dot file format made creating the parser very easy to debug, whenever I had an issue parsing, I was able to refer to the tree in either this form or the simplified form. While I kept the visual representation of the tree simple, there was lots of room to add more. This will be a technique I will keep around when I make my next compiler.

Oddities of Ada:

- Text, whether printed to the console or read from a file, were difficult to work with. New line characters are not present in files read from Ada, requiring reading the file line-by-line and refer to the "End_Of_Line" Ada function.
- The built-in methods of converting integers to strings contain a blank space in front of positive numbers to save room for a negative sign
- When using an If statement with multiple conditions it will check each one despite having enough information to conclude the output of an `and` or `or` statement. The terms of `and then` and `and or` had to be used to prevent this.
- Like VHDL and the language that we described, all variables must be declared at the beginning of a function before the `begin`. This led to using better variable names and encouraged me to re-use variables when their values are no longer needed.

Downsides of my Compiler:

String comparisons always result in false: I was torn between having strings be a fixed sized, as mentioned by a student in class, and reallocating space for every string assignment. I ended up reallocating memory for every assignment, while this led to any length strings, I was unable to figure out string comparisons in time.

No Implicit or Explicit type Casting: Working in Ada which is extremely strongly typed, I did not wish to add implicit type casting and I hoped to add type casting as done in C. Unfortunately, I ran out of time to implement this in my compiler.

GetBool() accepts the number 1 and 0: A side effect of not having string comparisons is not being able to read T/F or True/False in LLVM assembly to assign to Boolean operators. Substituting 1 and 0 works though.

Error Checking / Resynchronization: This aspect of my compiler is nowhere near where I had hoped, although it does work to an extent. This was something that I should have taken into consideration earlier.