Compilers - Checkpoint Two

Braydon Johnson, Neivin Mathew

March 22, 2017

1 Summary

For the second checkpoint, we implemented type checking for programs written in the C- language.

At this point, the compiler can recognize different scopes in C- programs and can build a symbol table for the entire program, along with recognizing symbols in more specific scopes. The symbol table enables the compiler to recognize when variables are used without declaration and if they are redefined within the same scope. Additionally, the compiler can display the different variable scopes in a program.

The compiler can also check types of expressions in C- programs. It has the ability to ensure that array ranges are integers, assignments and other operations are valid, and checking function return types.

Additionally, we also improved upon our initial design from Checkpoint One by fixing various bugs, and enhancing the error recovery of the parser. We also simplified some of our grammar rules by using the precedence and associativity directives defined in CUP.