CSE 442 Compiler Documentation

Section 1 Summary

Our group started with the goal to create a compiler in C for the classes language R. Over the course of the semester we developed and tested what is our final compiler. Now, we did not reach all of the goals from the beginning of the semester. Our compiler cannot produce full programs directly from R to assembly language.

We have been able to accomplish inserting records, arrays, function types, and expressions to the symbol table. Some functions can be inserted to the symbol table as well but can cause segmentation faults in certain cases. We can also detect syntax errors within the R code at this step including: same name within a scope, assigning variables to the wrong types, and checks for multiple type declarations of the same type with the same name. Additionally, we can generate and print out some IR code in certain circumstances. Our IR generation can handle assigning integer values to a variable and addition of said variables. This can be verified in the .ir file generated by the compiler. Unfortunately, we have the steps in place to begin assembly generation but we cannot compile any real x86 assembly. The design is to loop through our IR code array and generate the assembly code based on said IR lines.

Section 1.1 Invoking The Compiler

In order to invoke the compiler, use the following commands to invoke the compiler. Before doing any of the commands, make sure to use the command make compiler.

Tokens
/compiler -tok [filename]
Output the tokens from flex to the .tok file

Type Checking and Symbol Table
/compiler -st [filename]
Print out the Symbol Table and type check the R code

IR code Generation
/compiler -ir [filename]
Create and print out IR code to the .ir file

Help
/compiler -help
Print out the help message listing all the commands

Section 2 Program Examples and Expected Behavior

Basic Main Block with Integers

R Code	IR Generation

{ [integer : intx, intx2, intx3, intx4]
intx := 2;
intx2 := 1;
intx3 := intx + intx2;
}
	IR Code:
	main
	intx=2
	intx2=1
	intx3=intx2+intx

Alternate Main with Integers

R Code	IR Generation

{ [integer : intx, intx2, intx3, intx4]
intx4 := 2;
intx2 := 1;
intx := intx 3+ intx2;
}
	IR Code:
	main
	int4=2
	intx2=1
	intx=intx3+intx2