Anthony Colaiacovo – 1091370 – Compilers (CIS*4650)

Documentation for Checkpoint 3

What has been done for this checkpoint?

The primary goal for this checkpoint is to generate assembly code and run it on the TinyMachine. In terms of what I have accomplished, I have made code that will generate assembly, but it does not work properly, so I really have not accomplished very much this time around, since the program does not work.

Related techniques and the design process:

The design process was similar to the previous assignments, where it involved re-implementing the AbsynVisitor interface. In this case you also had to change the AbsynVisitor interface so I started with that. Then I tried to get fac.cm working but was unable to. I started with simple things like SimpleVar and AssignExp and was able to get those working. But unfortunately, any time I wanted to call output or something like that, I was unable to get it to work properly with function arguments.

Lessons learned in the implementation process:

To spend more time thinking before writing code. Especially in situations like this where the work is very meticulous. In this case I was trying to get it to work for a long time by playing around with offsets and I have learned that simply is not going to work with a project like this. If I had been more thorough and taken my time, I may have been able to accomplish more.

Assumptions, limitations, and possible improvements:

Unfortunately, because I was not able to complete the project there are a large number of possible improvements. I don't think assumptions or limitations are worth touching on because I didn't really make any. I guess I will say that you can assign values to variables, and those variables are assumed to be of the SimpleVar type, and not an array. In terms of possible improvements, the project still needs to have function calls working, arrays working, global variables working, while loops working, if/else statements working, OpExp's working, and returns working.

Acknowledgements

Both the course slides and Dr. Song's sample code for the tiny language had great influence over how this project was implemented. The course slides described all of the classes needed to make the abstract syntax tree, and I used those to create java files. Those java files also take the form of Dr. Song's sample AST files. Beyond this, I also used the emit functions given to us in slide deck number 11 for the creation of this