CS3071 – Exercise 3 Report

Edward Bergman

#15335633

Task:

Our understanding of the task was to establish the process Coco/R uses for compilation through the context of compiling a “Tastier” program and identifying how one implementation of a symbol table may work.

Using the symbol table provided, with symbols generated from the identities, the specific goal was to output information about symbols that were stored in the symbol table generated by SymTab.cs.

Method:

Using http://www.ssw.uni-linz.ac.at/Coco/Tutorial/, a tutorial set of slides for Coco/R, the process by which Coco/R works was better understood.

As we had access to the Symbol Table directly, outputting meta information of symbols was relativly straightforward. Deduce what type of object the symbol is and if the symbol is of the object type variable, output the type of the variable. This template of a function was used to convert SymbolObj -> string :

```
string ObjTypeToString(Obj o){
	if(o is typeof(var)){
		return vartype(o);
	else{
		return objecttype(o);
	}
}
```

Remarks:

It would have been prefferable to have the meta information printed above the assembly function. As we have only have access to the CloseScope() method, all information is printed after the function. Another note is that once within a scope, we could not figure how to identify the name of the enclosing scope object, only the level of the scope.

Learnt:

We have learnt how a symbol table could be implemented in a high level language, all be it a simpler language to compile for (Tastier). This exercise combined with reading material on how Coco/R tokens work for the Parser they generated has lead to a better understanding of how compilers work. Further understanding of how things such as static type checking could be implemented but also how debuggers could make use of this information during run-time once they have information from the symbol table.