How to invoke and Interpret Results

First to invoke the program, extract the zip file in a directory. Then, run “make” on the command line in a linux terminal in the directory that you extracted the files in. After that, do “make test”. Make test will first run the nameErrors.b file which tests errors in the code such as Multiply declared identifiers and the other errors described in the P4 spec. It will output ERROR and the specific error message, line number, and character number that the error came from. The way the test is currently written, there should be 10 errors output by the nameErrors.b file. Because there were errors while analyzing, the unparser should not run and therefore nameErrors.out should be empty. After that file is done, it will run test.b. Test.out file should have proper syntax and name analysis should run to completion with no errors. This means that the test.out file will hold the results of the name analysis. To interpret the results, open the test.out file in a text editor. When each variable identifier is used (not declared) it should show the type in parenthesis right after where it was used. For example, this would be in the test.out file:

```
int x;
x(int) = 10;
```

If all variables when used have their types after it, then the name analysis ran properly. If it does not, then there is an issue with the name analyzer.

Assumptions:

We made a few assumptions in our error messages. For example, in VarDeclNode we had a catch for if there was an EmptySymTableException and WrongArugmenetException. For these cases we just reported back the unexpected exception and which node caused that error then exited the program. Since this wasn’t covered in the specifications for P4 we assumed this would be a case where we want to stop processing. All of these assumptions of errors in which we stop processing can all be found by doing a “control-f” then searching for “System.err.println” as all of the other exceptions from the spec are done using ErrMsg.fatal. Other than that, we made no other assumptions about the functionality of P4.

Anything Else:

Nope, we feel pretty good about the name analyzer.