Assignment – Stage 2.

The for command was added to the FunChecker as follows:

Variable declaration:

```
for_var_decl                    returns [Type type]
        :       ^(FORVAR INT ID t=expr)
                                        { define($ID.text, Type.INT, $FORVAR);
                                          $type = t;
                                        }
        ;
```

It first declares the control variable of the loop and then return the type of the variable.

The proper for command (inside “com”):

```
|       ^(FOR t1=for_var_decl t2=expr com)
                                        { checkType(Type.INT, t1, $FOR);
                                          checkType(Type.INT, t2, $FOR);
                                        }
```

It firstly gets the type of the variable returned by “for_var_decl”, then gets the type of the second expression of the for command and finally it checks both types to see if they are correct. In this case both must be of type int.

The changes in the FunChecker.fun file were highlighted by adding //EXTENSION FOR comments in order to identify where the changes were made.

In order to achieve this second assignment objectives, tests were created as can be seen in the attached files. Below, each test is described:

rightTypeAndScopeFor.fun:
In this file a simple for was created, going from 2 to 5. No syntactic errors given. No contextual errors given.

wrongTypeFor.fun
In this file both expressions used were of the type bool, giving a type error:

```
Contextual analysis ...
2 scope/type errors
line 8:1 type is bool, should be int
line 8:1 type is bool, should be int
Compilation failed
```

wrongScopeFor.fun
In this file two different for loops were created but with the same control variable name. As the control variable scope extend from its declaration to the end of the function block, the second for attempts to redeclare “i”, the control variable, giving the scope error below:

```
Contextual analysis …
1 scope/type errors
line 0:0 i is redeclared
Compilation failed
```