NAME: Ben Puryear
FILE: hw5-writeup.pdf
DATE: Spring 2023
DESC: This pdf will provide a short description of any challenges and/or issues I faced and how I addressed them in this assignment.

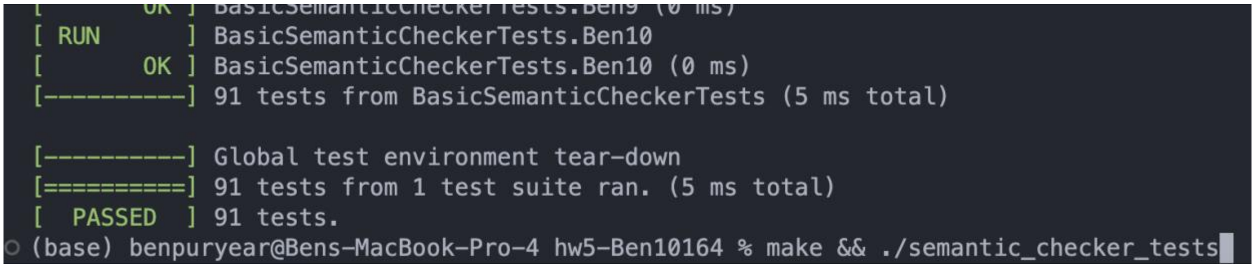
Challenges faced:

I faced many challenges in this assignment, but unlike the previous ones, none of which originated from the Lexer. Paths took me a while to conceptually plan out, but took little time to implement whereas type checking arrays seemed like an easy task, yet took a while to implement. I struggled with the specific rules for when arrays can be null, and what can be set to null. I used the debugger to discover that I was forgetting to set the value of .is_array for a few special cases as well as not testing for .is_array if the RHS is void.

Testing:

All Tests Passing

./semantic_checker_tests.cpp



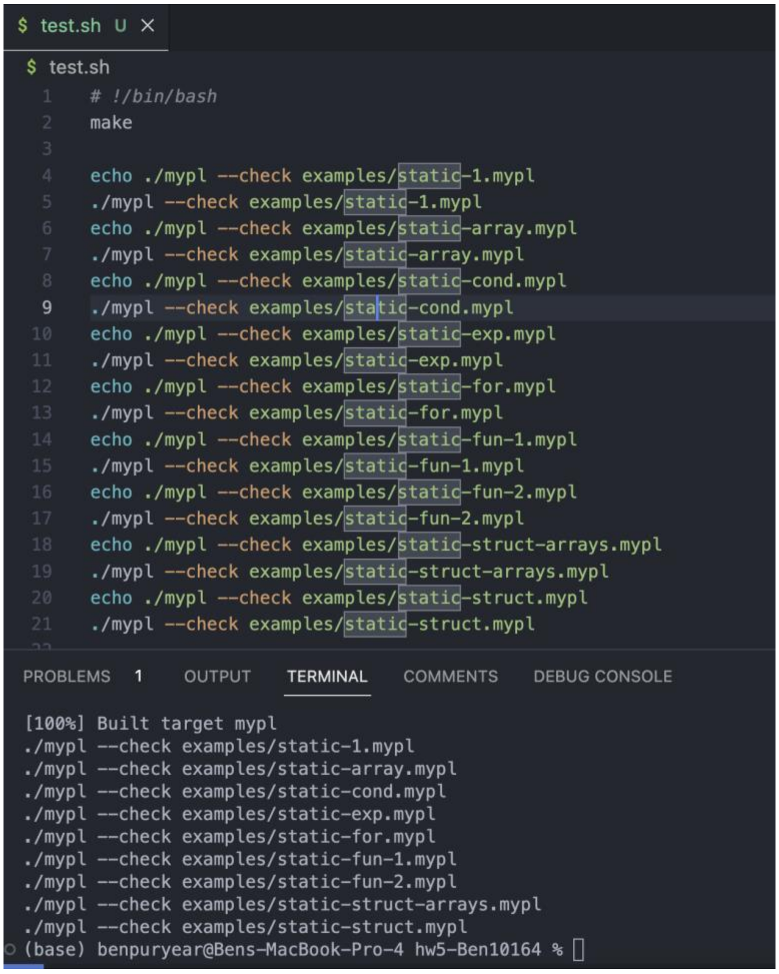
```
[       OK ] BasicSemanticCheckerTests.Ben9 (0 ms)
[ RUN      ] BasicSemanticCheckerTests.Ben10
[       OK ] BasicSemanticCheckerTests.Ben10 (0 ms)
[----------] 91 tests from BasicSemanticCheckerTests (5 ms total)

[----------] Global test environment tear-down
[==========] 91 tests from 1 test suite ran. (5 ms total)
[  PASSED  ] 91 tests.
○ (base) benpuryear@Bens-MacBook-Pro-4 hw5-Ben10164 % make && ./semantic_checker_tests
```

Additional Tests

My additional tests focused mostly on arrays and paths and what values can be set to what inside them. This is because my main problem with the assignment was arrays and paths, specifically type checking the different paths.

All Outputs of Examples/*

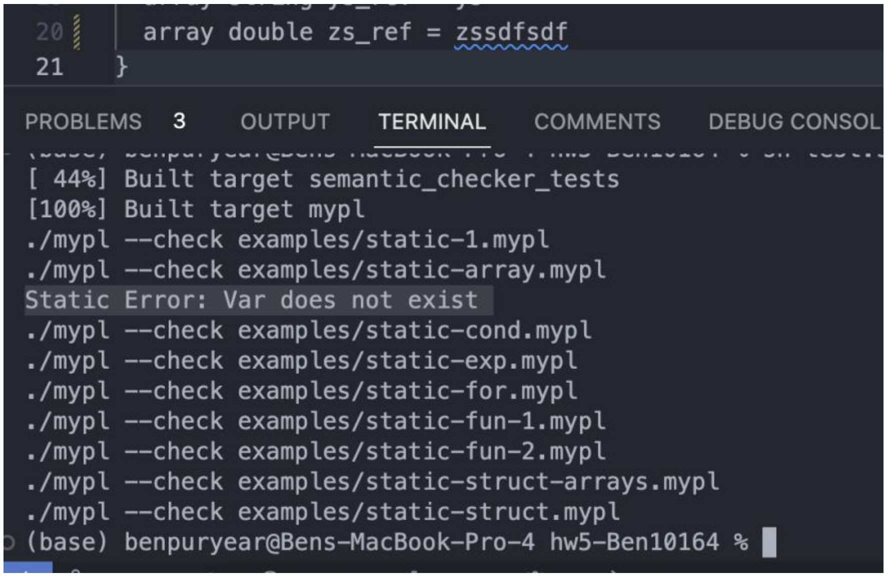
To do this more efficiently I created a bash file. (On next page)



```
# !/bin/bash
make

echo ./mypl --check examples/static-1.mypl
./mypl --check examples/static-1.mypl
echo ./mypl --check examples/static-array.mypl
./mypl --check examples/static-array.mypl
echo ./mypl --check examples/static-cond.mypl
./mypl --check examples/static-cond.mypl
echo ./mypl --check examples/static-exp.mypl
./mypl --check examples/static-exp.mypl
echo ./mypl --check examples/static-for.mypl
./mypl --check examples/static-for.mypl
echo ./mypl --check examples/static-fun-1.mypl
./mypl --check examples/static-fun-1.mypl
echo ./mypl --check examples/static-fun-2.mypl
./mypl --check examples/static-fun-2.mypl
echo ./mypl --check examples/static-struct-arrays.mypl
./mypl --check examples/static-struct-arrays.mypl
echo ./mypl --check examples/static-struct.mypl
./mypl --check examples/static-struct.mypl
```

```
[100%] Built target mypl
./mypl --check examples/static-1.mypl
./mypl --check examples/static-array.mypl
./mypl --check examples/static-cond.mypl
./mypl --check examples/static-exp.mypl
./mypl --check examples/static-for.mypl
./mypl --check examples/static-fun-1.mypl
./mypl --check examples/static-fun-2.mypl
./mypl --check examples/static-struct-arrays.mypl
./mypl --check examples/static-struct.mypl
(base) benpuryear@Bens-MacBook-Pro-4 hw5-Ben10164 %
```

As you can see, none of the example files output anything.
Here is an example of an error to prove that the bash script outputs errors.



```
    array double zs_ref = zssdfsdf
}
```

```
[ 44%] Built target semantic_checker_tests
[100%] Built target mypl
./mypl --check examples/static-1.mypl
./mypl --check examples/static-array.mypl
Static Error: Var does not exist
./mypl --check examples/static-cond.mypl
./mypl --check examples/static-exp.mypl
./mypl --check examples/static-for.mypl
./mypl --check examples/static-fun-1.mypl
./mypl --check examples/static-fun-2.mypl
./mypl --check examples/static-struct-arrays.mypl
./mypl --check examples/static-struct.mypl
(base) benpuryear@Bens-MacBook-Pro-4 hw5-Ben10164 %
```