Project 4 Documentation

Antonio E Ramirez

University of Maryland Global Campus

10 May 2020

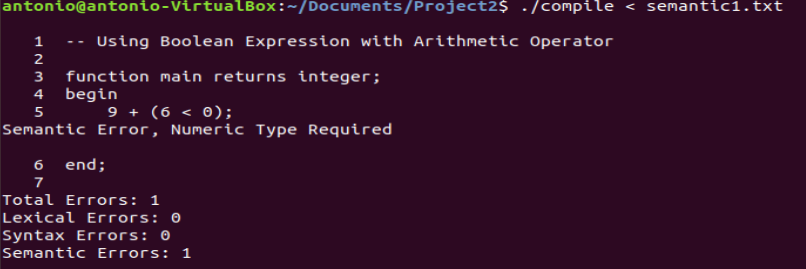
# Approach

This week I wanted to ensure I was prepared as the last project was exceptionally difficult for me. I read through the course content thoroughly and watched the video series for both weeks 7 and 8 a few times through to make sure I understood the content. I then read through the project requirements, emphasizing the rubric. This gave me a solid understanding of the final goal of the project and I started to develop my own milestones from the rubric. I then went and corrected my Project 3 code using the feedback provided, making sure I understood the changes. After this I read through the approach document. The first thing I noticed was that the Project 2 code was suggested as a starting point. I then took my Project 2 code and compared it to the Project 4 Skeleton Code. As I did this, I was also watching the 4th video from Week 8, that walks through some of the changes and why they are done. I then ran the Skeleton Code, again making sure I understood how everything worked. I finally began working on my Project 4 code, using Project 2 as a starting point. I implemented the changes from the Skeleton Code into my code, making some changes so the code would run and produce the same output as the video for the semantic1-6.txt files. I was now at a good point to compare milestones with the Project 4 Approach document. Thankfully they lined up nicely, so I went through the Approach document implementing the features step-by-step. I made sure that I ran each of the provided tests and got the same result before moving on. Upon running the final test, semantic19.txt, I went back and reran all of the tests to make sure that none of the previous working parts of the program were affected during the changes.

# Test Plan

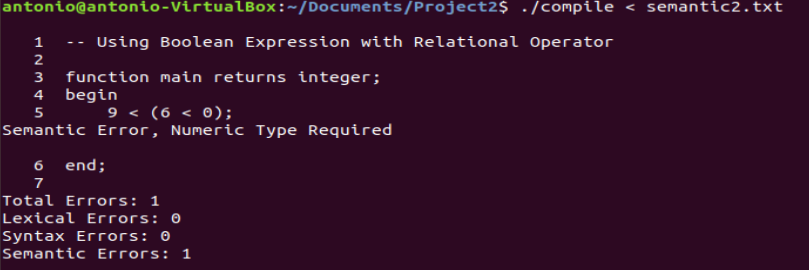
To test my program, I used the test cases included in the approach document, as I found them highly applicable and because they tested all the graded requirements from the rubric. Thankfully I was much more successful this week in being able to implement all the project requirements, with the test cases returning the desired results. I found it very helpful to stop at each step in the Approach document and ensure that the given test case is a pass before continuing. This is important because the steps seemed to build off the previous steps, and if those were not working properly it could affect the future requirements. I will include screenshots of all the test cases, what was being tested, and whether the program received a pass or fail for the test.

**Test Case 1:**



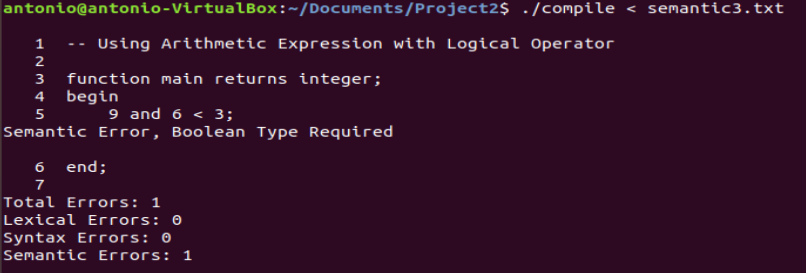
This test case was a pass. We tested a Boolean expression with an Arithmetic operator. This was a test case provided and was reviewed in the video series. Our program detected the semantic error correctly.

**Test Case 2:**



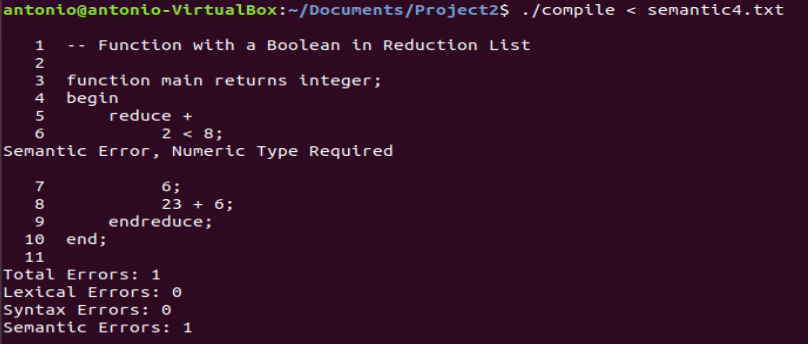
This test case was a pass. We tested a Boolean expression with a Relational operator. This was a test case provided and was reviewed in the video series. Our program detected the semantic error correctly.

**Test Case 3:**



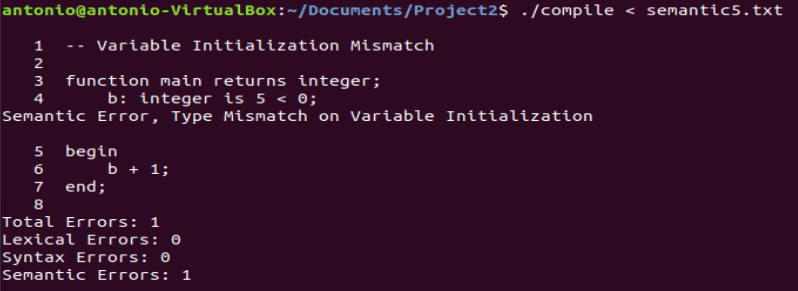
This test case was a pass. We tested an Arithmetic expression with a Logical operator. This was a test case provided and was reviewed in the video series. Our program detected the semantic error correctly.

**Test Case 4:**



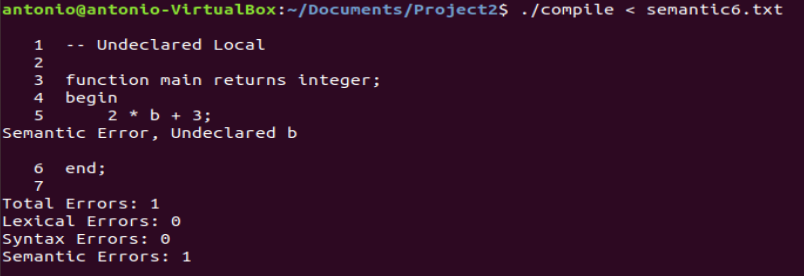
This test case was a pass. We tested a Boolean within a reduction. This was a test case provided and was reviewed in the video series. Our program detected the semantic error correctly.

**Test Case 5:**



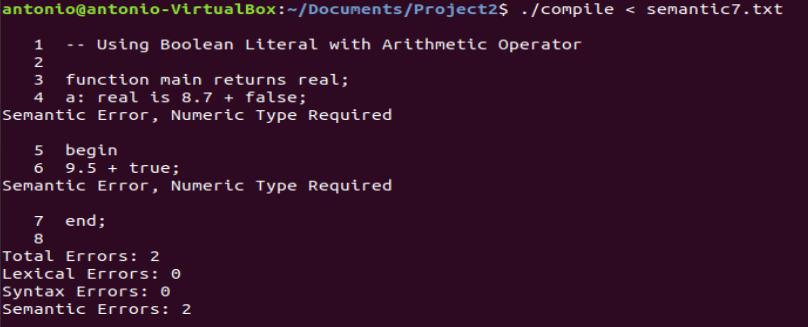
This test case was a pass. We tested variable initialization and a mismatch of type. This was a test case provided and was reviewed in the video series. Our program detected the semantic error correctly.

**Test Case 6:**



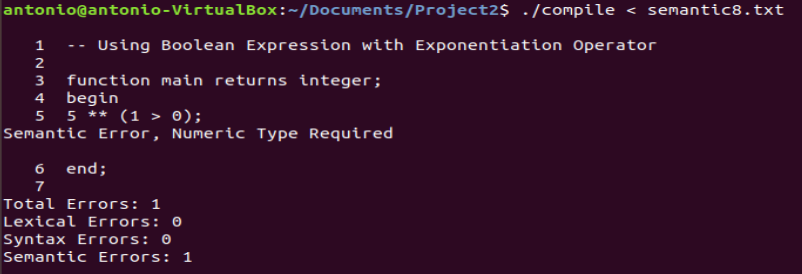
This test case was a pass. We tested an undeclared variable. This was a test case provided and was reviewed in the video series. Our program detected the semantic error correctly.

**Test Case 7:**



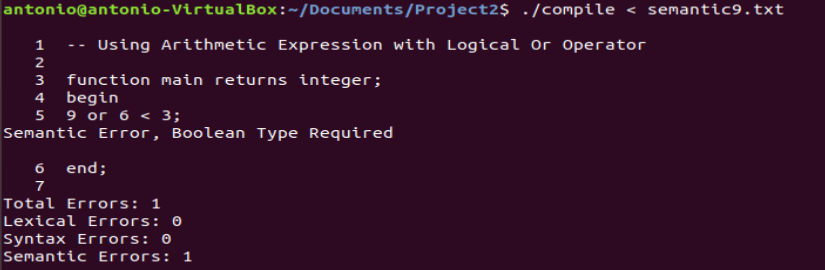
This test case was a pass. We tested using a Boolean literal with an arithmetic operator. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

**Test Case 8:**



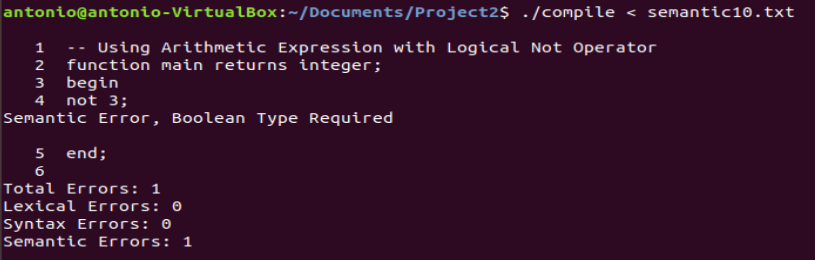
This test case was a pass. We tested using a Boolean expression with the exponentiation operator. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

**Test Case 9:**



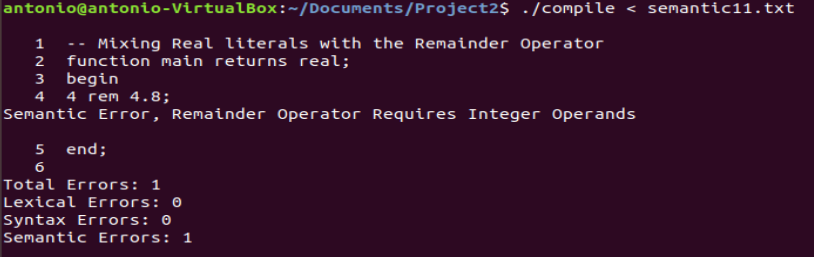
This test case was a pass. We tested using an arithmetic expression with a logical operator. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

**Test Case 10:**



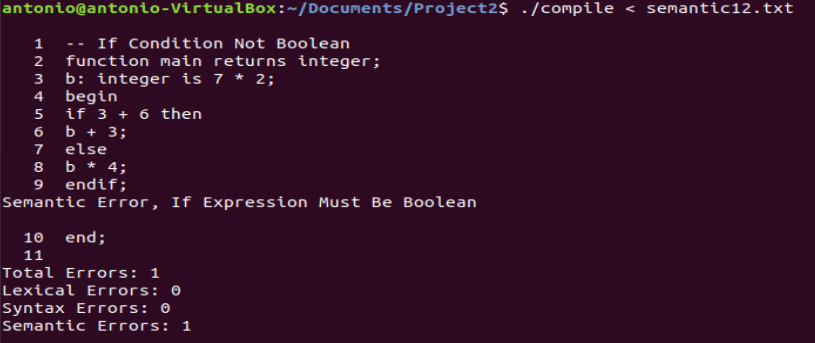
This test case was a pass. We tested using an arithmetic expression with the not operator, which is a logical operator. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

**Test Case 11:**



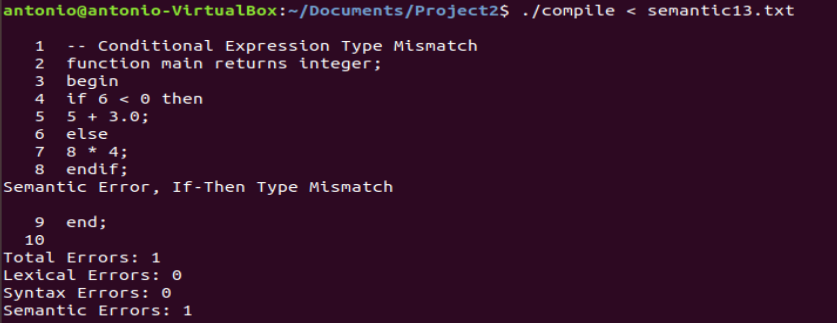
This test case was a pass. We tested using a real with the rem operator, which requires integers. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

**Test Case 12:**



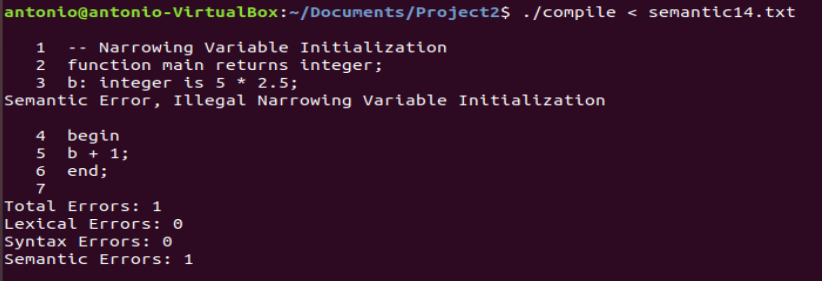
This test case was a pass. We tested using an arithmetic expression as the condition of the if statement, which requires a Boolean expression or value. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

**Test Case 13:**



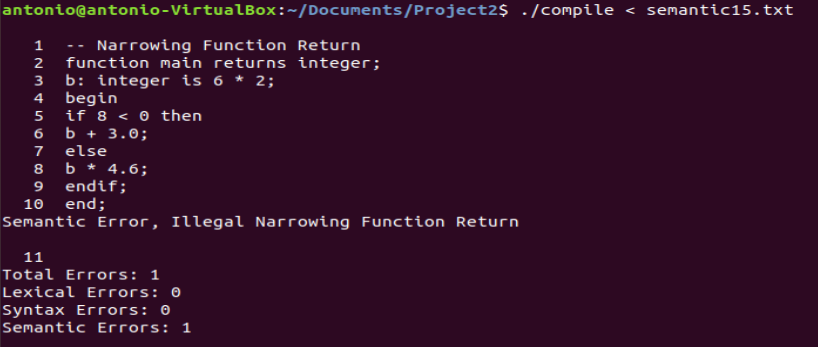
This test case was a pass. We tested having two different types returned for the then and else statements, real and integer. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

**Test Case 14:**



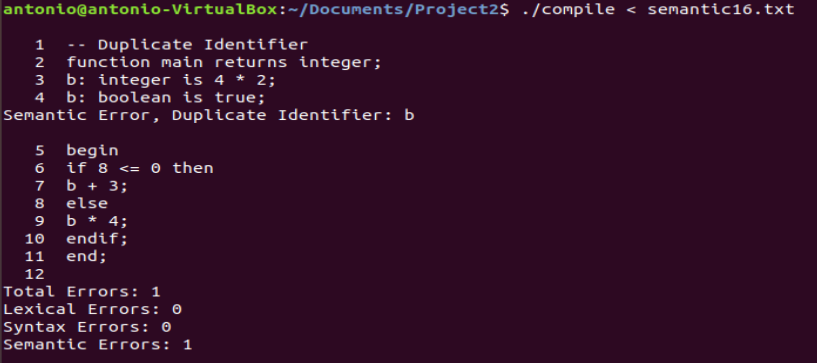
This test case was a pass. We tested illegal narrowing at the variable initialization. The variable is of type integer, but is fed a real value, thus causing the semantic error. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

**Test Case 15:**



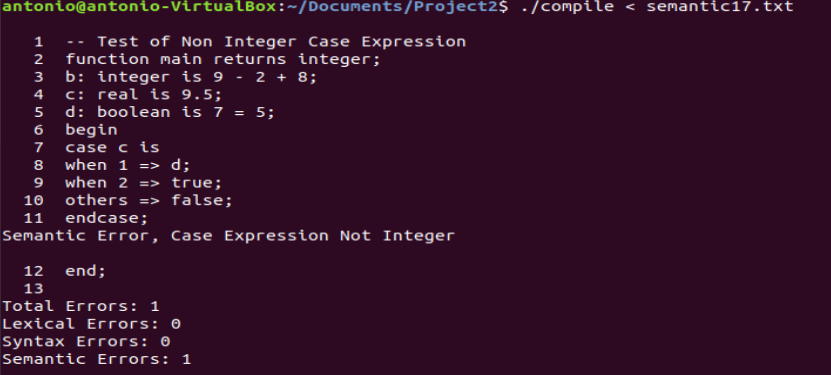
This test case was a pass. We tested illegal narrowing at the function return level. The return type is integer, with the body returning a real, thus throwing the semantic error. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

**Test Case 16:**



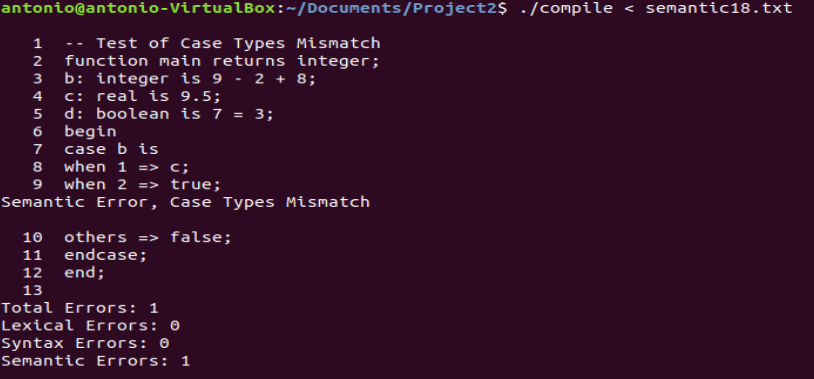
This test case was a pass. We tested having duplicate variable declarations. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

**Test Case 17:**



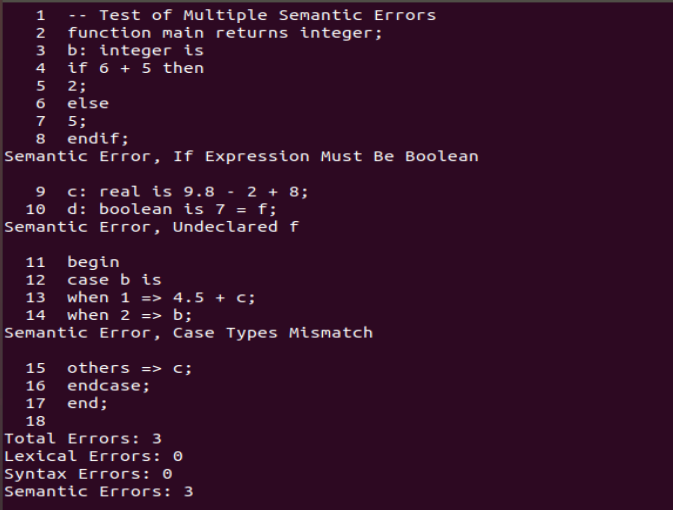
This test case was a pass. We tested having a different type than integer in our case expression, using a real. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

**Test Case 18:**



This test case was a pass. We tested having varying types of cases. We have when 1 resulting in a real type, and all other cases resulting in a Boolean type. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

**Test Case 19:**



This test case was a pass. We tested multiple semantic errors within the same test case as a final test. First, we have an arithmetic expression used for the conditional of the if statement, which must be Boolean. Then we have an undeclared variable used within a variable declaration. Lastly, we have mismatched types of case statements. This was a test case provided in the Project 4 Approach document. Our program achieved the same result as the document by detecting the semantic error(s) correctly.

# Lessons Learned

I enjoyed this project much more than the previous. However, I feel as though I was able to apply the newly learned knowledge to this project much easier than Project 3. The lessons learned from Project 3, which I struggled with, were applied to this project and made the experience much more enjoyable. I was able to make sure that I understood the content and requirements much more than before, and I was able to develop a sounder plan of attack. This was due to a combination of the course content, Approach document, Ask the Professor discussions, and feedback from Project 3. While I ended up using the Project 2 code as the foundation for this project, the feedback received from Project 3 helped immensely in understanding where my knowledge gaps were. I do not think I would have been as successful without the feedback. The importance of identifying knowledge gaps was a lesson learned during this project, and class as a whole. I also gained a much stronger understanding of the parse tree, specifically with the passing of values and recursion within the tree. I also understood the purpose of the empty “leaves” of the tree with regards to recursion much more. Previously I did not fully understand the purpose, but when implementing the case semantic checks this required me to build a deeper understanding of the recursion and the parse tree form a wholistic perspective. It was much more fulfilling to be able to complete the project with a pass on all test cases. I am sure there are areas that my code could be less ambiguous or more straightforward and efficient. But overall, I am very happy with the outcome of this project, not only for completing it, but for the concepts reinforced and learned.