Week 6: Project 3

John Kucera

Prof. Renee McDonald

CMSC 430 Compiler Theory and Design

22 April 2021

Week 6: Project 3 Documentation

Modifying the Program Interpreter (C++ with Flex and Bison)

Approaching the Project

Like with the previous projects, I first made sure I fully understood everything in the skeleton code before attempting to modify it or add anything to it. As usual, Dr. Duane Jarc's videos were an incredible help to me understanding each line of the code. I then went to make additions and modifications to the skeleton code that I already had from previous projects, like the error checking and additional tokens in scanner.l. I also added the proper precedence of operators and tree branches to parser.y, as I did in project 2.

It was fairly simple to add more arithmetic and relational operators. I first used yylval in scanner.l to properly assign tokens to values/operators so that they would be recognized for having a meaning. Then, I made additions to values.cc that holds the evaluation functions. That file is completely in C++ code, so I just had to write C++ code that represented what the operators are supposed to do. For example, if the input program has an exponent ** operator, values.cc says to perform the following C++ code: result = pow(left * 1.0, right * 1.0), which is how to perform exponent operations in C++.

Week 6: Project 3

It got trickier when it came to figuring out the conditional expressions, but when thinking about the operations in just C++, I was able to figure out simple solutions. I realized that I could perform the if-else entirely in C++ and then return the chosen statement. I created multiple functions to handle the case statements. In the first function, I stored the value of the variable in question (for example, storing ‘a’ from “case a is…”). In the second function, I evaluated a case and returned the resulting statement IF it was a matching case. In the third function, I evaluated “others” by seeing if a case had matched. If a case was not matched, then the result from “others” is returned instead. All was done in values.cc.

Finally, I converted all values to double. Integer and real literals are doubles. Boolean literals are also double, true being equal to 1 and false being equal to 0. I also made sure to store multiple parameters in an array to be referenced later.

Week 6: Project 3

Test Cases

(Test Case 1)

Aspect tested: BOOLEAN FALSE. An input file containing Boolean literals, multiple variables, and an if expression

Input: test1.txt, input 23.8.

Expected Output: 0 (represents false)

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/D
ocuments/'CMSC 430'/JohnKucera-Project3-Tests/test1.txt 23.8

   1  -- This tests boolean literal evaluation, if expression evaluation.
   2
   3  function test1 a: real returns boolean;
   4    b: boolean is true;
   5    c: boolean is false;
   6  begin
   7    if a > 50.1 then
   8            b;
   9    else
  10            c;
  11    endif;
  12  end;

Compiled Successfully.

Result = 0
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. “if” statement is not met, so it goes into “else”. This returns variable c which equals false, represented by 0.

Week 6: Project 3

(Test Case 2)

Aspect tested: BOOLEAN TRUE. An input file containing Boolean literals, multiple variables, and an if expression

Input: test1.txt, input 87.1

Expected Output: 1 (represents true)

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/D
ocuments/'CMSC 430'/JohnKucera-Project3-Tests/test1.txt 87.1

   1  -- This tests boolean literal evaluation, if expression evaluation.
   2
   3  function test1 a: real returns boolean;
   4    b: boolean is true;
   5    c: boolean is false;
   6  begin
   7    if a > 50.1 then
   8            b;
   9    else
  10            c;
  11    endif;
  12  end;

Compiled Successfully.

Result = 1
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. “if” statement is met. This returns variable d which equals true, represented by 1.

Week 6: Project 3

(Test Case 3)

Aspect tested: ADDITION/MULTIPLICATION, CASE. An input file containing real literals, arithmetic operators, case expression.

Input: test2.txt, input 1

Expected Output: (13.1 + 7.77) * 4.5 = **93.915**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera
-Project3-Tests/test2.txt 1

   1  -- This tests real evaluation, arithmetic operators, case expression evaluation, multiple variables.
   2
   3  function test2 a: integer returns real;
   4    b: real is 13.1 + 7.77;
   5    c: real is 87.2 - 10.1;
   6    d: real is 55.5 / 11.1;
   7  begin
   8    case a is
   9            when 1 => b * 4.5;
  10            when 2 => c * (17.5 rem 6.0);
  11            others => d ** 2;
  12    endcase;
  13  end;

Compiled Successfully.

Result = 93.915
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case statement is met, addition and multiplication are properly evaluated.

Week 6: Project 3

(Test Case 4)

Aspect tested: SUBTRACTION, REM, CASE. An input file containing real literals, arithmetic operators, case expression.

Input: test2.txt, input 2

Expected Output: (87.2 – 10.1) * (17.5 rem 6.0) = **424.05**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera
-Project3-Tests/test2.txt 2

   1  -- This tests real evaluation, arithmetic operators, case expression evaluation, multiple variables.
   2
   3  function test2 a: integer returns real;
   4    b: real is 13.1 + 7.77;
   5    c: real is 87.2 - 10.1;
   6    d: real is 55.5 / 11.1;
   7  begin
   8    case a is
   9            when 1 => b * 4.5;
  10            when 2 => c * (17.5 rem 6.0);
  11            others => d ** 2;
  12    endcase;
  13  end;

Compiled Successfully.

Result = 424.05
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case statement is met, subtraction and rem are properly evaluated.

Week 6: Project 3

(Test Case 5)

Aspect tested: DIVISION, EXPONENT, CASE. An input file containing real literals, arithmetic operators, case expression.

Input: test2.txt, input 5

Expected Output: (55.5 / 11.1) ^ 2 = **25**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera
-Project3-Tests/test2.txt 5

  1  -- This tests real evaluation, arithmetic operators, case expression evaluation, multiple variables.
  2
  3  function test2 a: integer returns real;
  4    b: real is 13.1 + 7.77;
  5    c: real is 87.2 - 10.1;
  6    d: real is 55.5 / 11.1;
  7  begin
  8    case a is
  9            when 1 => b * 4.5;
 10            when 2 => c * (17.5 rem 6.0);
 11            others => d ** 2;
 12    endcase;
 13  end;

Compiled Successfully.

Result = 25
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case statement is met, division and exponent are properly evaluated.

Week 6: Project 3

(Test Case 6)

Aspect tested: LESS/GREATER, AND, MULTIPLE PARAMETERS, NESTED CASE/IF. An input file containing both case and if statements, relational operators, logical operators, multiple parameters, and reductions.

Input: test3.txt, input 1 12

Expected Output: when 1 => if a < b AND b > c then **1.1**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera-Project3-Tests/test3.txt 1 12

  1  -- This tests nested conditional statements, relation operators, logical operators, multiple parameters, and reduction evaluation.
  2
  3  function test3 a: integer, b: integer returns integer;
  4    c: integer is 10;
  5  begin
  6    case a is
  7            when 1 =>
  8                    if a < b and b > c then
  9                            1.1;
 10                    else
 11                            2.2;
 12                    endif;
 13            when 2 =>
 14                    if b >= c or b <= a then
 15                            3.3;
 16                    else
 17                            4.4;
 18                    endif;
 19            when 3 =>
 20                    if a /= b then
 21                            5.5;
 22                    else
 23                            6.6;
 24                    endif;
 25            when 4 =>
 26                    if not (a = b) then
 27                            7.7;
 28                    else
 29                            8.8;
 30                    endif;
 31            when 5 =>
 32                    reduce +
 33                            20;
 34                            6;
 35                            10;
 36                    endreduce;
Text Editor
           when 6 =>
                    reduce *
 39                            2;
 40                            90;
 41                            10;
 42                    endreduce;
 43            others =>
 44                    100;
 45    endcase;
 46  end;

Compiled Successfully.

Result = 1.1
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case and if statements are met, multiple parameters are both stored properly, “less”, “greater” and “and” operators are properly evaluated.

Week 6: Project 3

(Test Case 7)

Aspect tested: LESS/GREATER, AND, MULTIPLE PARAMETERS, NESTED CASE/IF. An input file containing both case and if statements, relational operators, logical operators, multiple parameters, and reductions.

Input: test3.txt, input 1 7

Expected Output: when 1 => else **2.2**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera-Project3-Tests/test3.txt 1 7

  1  -- This tests nested conditional statements, relation operators, logical operators, multiple parameters, and reduction evaluation.
  2
  3  function test3 a: integer, b: integer returns integer;
  4    c: integer is 10;
  5  begin
  6    case a is
  7            when 1 =>
  8                    if a < b and b > c then
  9                            1.1;
 10                    else
 11                            2.2;
 12                    endif;
 13            when 2 =>
 14                    if b >= c or b <= a then
 15                            3.3;
 16                    else
 17                            4.4;
 18                    endif;
 19            when 3 =>
 20                    if a /= b then
 21                            5.5;
 22                    else
 23                            6.6;
 24                    endif;
 25            when 4 =>
 26                    if not (a = b) then
 27                            7.7;
 28                    else
 29                            8.8;
 30                    endif;
 31            when 5 =>
 32                    reduce +
 33                            20;
 34                            6;
 35                            10;
 36                    endreduce;
 37            when 6 =>
 38                    reduce *
 39                            2;
 40                            90;
 41                            10;
 42                    endreduce;
 43            others =>
 44                    100;
 45    endcase;
 46  end;

Compiled Successfully.

Result = 2.2
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case and if statements are met, multiple parameters are both stored properly, “less”, “greater” and “and” operators are properly evaluated.

Week 6: Project 3

(Test Case 8)

Aspect tested: LESS/GREATER OR EQUAL TO, OR, MULTIPLE PARAMETERS, NESTED CASE/IF. An input file containing both case and if statements, relational operators, logical operators, multiple parameters, and reductions.

Input: test3.txt, input 2 1

Expected Output: when 2 => if b >= c or b <= a then **3.3**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera-Project3-Tests/test3.txt 2 1

  1  -- This tests nested conditional statements, relation operators, logical operators, multiple parameters, and reduction evaluation.
  2
  3  function test3 a: integer, b: integer returns integer;
  4    c: integer is 10;
  5  begin
  6    case a is
  7            when 1 =>
  8                    if a < b and b > c then
  9                            1.1;
 10                    else
 11                            2.2;
 12                    endif;
 13            when 2 =>
 14                    if b >= c or b <= a then
 15                            3.3;
 16                    else
 17                            4.4;
 18                    endif;
 19            when 3 =>
 20                    if a /= b then
 21                            5.5;
 22                    else
 23                            6.6;
 24                    endif;
 25            when 4 =>
 26                    if not (a = b) then
 27                            7.7;
 28                    else
 29                            8.8;
 30                    endif;
 31            when 5 =>
 32                    reduce +
 33                            20;
 34                            6;
 35                            10;
 36                    endreduce;
 37            when 6 =>
 38                    reduce *
 39                            2;
 40                            90;
 41                            10;
 42                    endreduce;
 43            others =>
 44                    100;
 45    endcase;
 46  end;

Compiled Successfully.

Result = 3.3
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case and if statements are met, multiple parameters are both stored properly, “less than or equal to”, “greater than or equal to” and “or” operators are properly evaluated.

Week 6: Project 3

(Test Case 9)

Aspect tested: LESS/GREATER OR EQUAL TO, OR, MULTIPLE PARAMETERS, NESTED CASE/IF. An input file containing both case and if statements, relational operators, logical operators, multiple parameters, and reductions.

Input: test3.txt, input 2 5

Expected Output: when 2 => else **4.4**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera-Project3-Tests/test3.txt 2 5
  1  -- This tests nested conditional statements, relation operators, logical operators, multiple parameters, and reduction evaluation.
  2
  3  function test3 a: integer, b: integer returns integer;
  4    c: integer is 10;
  5  begin
  6    case a is
  7            when 1 =>
  8                    if a < b and b > c then
  9                            1.1;
 10                    else
 11                            2.2;
 12                    endif;
 13            when 2 =>
 14                    if b >= c or b <= a then
 15                            3.3;
 16                    else
 17                            4.4;
 18                    endif;
 19            when 3 =>
 20                    if a /= b then
 21                            5.5;
 22                    else
 23                            6.6;
 24                    endif;
 25            when 4 =>
 26                    if not (a = b) then
 27                            7.7;
 28                    else
 29                            8.8;
 30                    endif;
 31            when 5 =>
 32                    reduce +
 33                            20;
 34                            6;
 35                            10;
 36                    endreduce;
 37            when 6 =>
 38                    reduce *
 39                            2;
 40                            90;
 41                            10;
 42                    endreduce;
 43            others =>
 44                    100;
 45    endcase;
 46  end;

Compiled Successfully.

Result = 4.4
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case and if statements are met, multiple parameters are both stored properly, “less than or equal to”, “greater than or equal to” and “or” operators are properly evaluated.

Week 6: Project 3

(Test Case 10)

Aspect tested: NOT EQUAL, MULTIPLE PARAMETERS, NESTED CASE/IF. An input file containing both case and if statements, relational operators, logical operators, multiple parameters, and reductions.

Input: test3.txt, input 3 80

Expected Output: when 3 => if a /= b then **5.5**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera-Project3-Tests/test3.txt 3 80

  1  -- This tests nested conditional statements, relation operators, logical operators, multiple parameters, and reduction evaluation.
  2
  3  function test3 a: integer, b: integer returns integer;
  4    c: integer is 10;
  5  begin
  6    case a is
  7            when 1 =>
  8                    if a < b and b > c then
  9                            1.1;
 10                    else
 11                            2.2;
 12                    endif;
 13            when 2 =>
 14                    if b >= c or b <= a then
 15                            3.3;
 16                    else
 17                            4.4;
 18                    endif;
 19            when 3 =>
 20                    if a /= b then
 21                            5.5;
 22                    else
 23                            6.6;
 24                    endif;
 25            when 4 =>
 26                    if not (a = b) then
 27                            7.7;
 28                    else
 29                            8.8;
 30                    endif;
 31            when 5 =>
 32                    reduce +
 33                            20;
 34                            6;
 35                            10;
 36                    endreduce;
 37            when 6 =>
 38                    reduce *
 39                            2;
 40                            90;
 41                            10;
 42                    endreduce;
 43            others =>
 44                    100;
 45    endcase;
 46  end;

Compiled Successfully.

Result = 5.5
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case and if statements are met, multiple parameters are both stored properly, “not equal” operator is properly evaluated.

Week 6: Project 3

(Test Case 11)

Aspect tested: NOT EQUAL, MULTIPLE PARAMETERS, NESTED CASE/IF. An input file containing both case and if statements, relational operators, logical operators, multiple parameters, and reductions.

Input: test3.txt, input 3 3

Expected Output: when 3 => else **6.6**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera-Project3-Tests/test3.txt 3 3

  1  -- This tests nested conditional statements, relation operators, logical operators, multiple parameters, and reduction evaluation.
  2
  3  function test3 a: integer, b: integer returns integer;
  4    c: integer is 10;
  5  begin
  6    case a is
  7            when 1 =>
  8                    if a < b and b > c then
  9                            1.1;
 10                    else
 11                            2.2;
 12                    endif;
 13            when 2 =>
 14                    if b >= c or b <= a then
 15                            3.3;
 16                    else
 17                            4.4;
 18                    endif;
 19            when 3 =>
 20                    if a /= b then
 21                            5.5;
 22                    else
 23                            6.6;
 24                    endif;
 25            when 4 =>
 26                    if not (a = b) then
 27                            7.7;
 28                    else
 29                            8.8;
 30                    endif;
 31            when 5 =>
 32                    reduce +
 33                            20;
 34                            6;
 35                            10;
 36                    endreduce;
 37            when 6 =>
 38                    reduce *
 39                            2;
 40                            90;
 41                            10;
 42                    endreduce;
 43            others =>
 44                    100;
 45    endcase;
 46  end;

Compiled Successfully.

Result = 6.6
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case and if statements are met, multiple parameters are both stored properly, “not equal” operator is properly evaluated.

Week 6: Project 3

(Test Case 12)

Aspect tested: EQUAL, NOT, MULTIPLE PARAMETERS, NESTED CASE/IF. An input file containing both case and if statements, relational operators, logical operators, multiple parameters, and reductions.

Input: test3.txt, input 4 76

Expected Output: when 4 => if not (a = b) then **7.7**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera-Project3-Tests/test3.txt 4 76
  1  -- This tests nested conditional statements, relation operators, logical operators, multiple parameters, and reduction evaluation.
  2
  3  function test3 a: integer, b: integer returns integer;
  4    c: integer is 10;
  5  begin
  6    case a is
  7            when 1 =>
  8                    if a < b and b > c then
  9                            1.1;
 10                    else
 11                            2.2;
 12                    endif;
 13            when 2 =>
 14                    if b >= c or b <= a then
 15                            3.3;
 16                    else
 17                            4.4;
 18                    endif;
 19            when 3 =>
 20                    if a /= b then
 21                            5.5;
 22                    else
 23                            6.6;
 24                    endif;
 25            when 4 =>
 26                    if not (a = b) then
 27                            7.7;
 28                    else
 29                            8.8;
 30                    endif;
 31            when 5 =>
 32                    reduce +
 33                            20;
 34                            6;
 35                            10;
 36                    endreduce;
 37            when 6 =>
 38                    reduce *
 39                            2;
 40                            90;
 41                            10;
 42                    endreduce;
 43            others =>
 44                    100;
 45    endcase;
 46  end;

Compiled Successfully.

Result = 7.7
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case and if statements are met, multiple parameters are both stored properly, “equal” and “not” operators are properly evaluated.

Week 6: Project 3

(Test Case 13)

Aspect tested: EQUAL, NOT, MULTIPLE PARAMETERS, NESTED CASE/IF. An input file containing both case and if statements, relational operators, logical operators, multiple parameters, and reductions.

Input: test3.txt, input 4 4

Expected Output: when 4 => else **8.8**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera-Project3-Tests/test3.txt 4 4

  1  -- This tests nested conditional statements, relation operators, logical operators, multiple parameters, and reduction evaluation.
  2
  3  function test3 a: integer, b: integer returns integer;
  4    c: integer is 10;
  5  begin
  6    case a is
  7            when 1 =>
  8                    if a < b and b > c then
  9                            1.1;
 10                    else
 11                            2.2;
 12                    endif;
 13            when 2 =>
 14                    if b >= c or b <= a then
 15                            3.3;
 16                    else
 17                            4.4;
 18                    endif;
 19            when 3 =>
 20                    if a /= b then
 21                            5.5;
 22                    else
 23                            6.6;
 24                    endif;
 25            when 4 =>
 26                    if not (a = b) then
 27                            7.7;
 28                    else
 29                            8.8;
 30                    endif;
 31            when 5 =>
 32                    reduce +
 33                            20;
 34                            6;
 35                            10;
 36                    endreduce;
 37            when 6 =>
 38                    reduce *
 39                            2;
 40                            90;
 41                            10;
 42                    endreduce;
 43            others =>
 44                    100;
 45    endcase;
 46  end;

Compiled Successfully.

Result = 8.8
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case and if statements are met, multiple parameters are both stored properly, “equal” and “not” operators are properly evaluated.

Week 6: Project 3

(Test Case 14)

Aspect tested: REDUCE (addition), MULTIPLE PARAMETERS. An input file containing both case and if statements, relational operators, logical operators, multiple parameters, and reductions.

Input: test3.txt, input 5 1

Expected Output: when 5 => 20 + 6 + 10 = **36**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera-Project3-Tests/test3.txt 5 1

  1  -- This tests nested conditional statements, relation operators, logical operators, multiple parameters, and reduction evaluation.
  2
  3  function test3 a: integer, b: integer returns integer;
  4    c: integer is 10;
  5  begin
  6    case a is
  7            when 1 =>
  8                    if a < b and b > c then
  9                            1.1;
 10                    else
 11                            2.2;
 12                    endif;
 13            when 2 =>
 14                    if b >= c or b <= a then
 15                            3.3;
 16                    else
 17                            4.4;
 18                    endif;
 19            when 3 =>
 20                    if a /= b then
 21                            5.5;
 22                    else
 23                            6.6;
 24                    endif;
 25            when 4 =>
 26                    if not (a = b) then
 27                            7.7;
 28                    else
 29                            8.8;
 30                    endif;
 31            when 5 =>
 32                    reduce +
 33                            20;
 34                            6;
 35                            10;
 36                    endreduce;
 37            when 6 =>
 38                    reduce *
 39                            2;
 40                            90;
 41                            10;
 42                    endreduce;
 43            others =>
 44                    100;
 45    endcase;
 46  end;

Compiled Successfully.

Result = 36
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case statement is met, multiple parameters are both stored properly, addition reduction is properly evaluated.

Week 6: Project 3

(Test Case 15)

Aspect tested: REDUCE (multiplication), MULTIPLE PARAMETERS. An input file containing both case and if statements, relational operators, logical operators, multiple parameters, and reductions.

Input: test3.txt, input 6 2

Expected Output: when 5 => 2 * 90 * 10 = **1800**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera-Project3-Tests/test3.txt 6 2

  1  -- This tests nested conditional statements, relation operators, logical operators, multiple parameters, and reduction evaluation.
  2
  3  function test3 a: integer, b: integer returns integer;
  4    c: integer is 10;
  5  begin
  6    case a is
  7            when 1 =>
  8                    if a < b and b > c then
  9                            1.1;
 10                    else
 11                            2.2;
 12                    endif;
 13            when 2 =>
 14                    if b >= c or b <= a then
 15                            3.3;
 16                    else
 17                            4.4;
 18                    endif;
 19            when 3 =>
 20                    if a /= b then
 21                            5.5;
 22                    else
 23                            6.6;
 24                    endif;
 25            when 4 =>
 26                    if not (a = b) then
 27                            7.7;
 28                    else
 29                            8.8;
 30                    endif;
 31            when 5 =>
 32                    reduce +
 33                            20;
 34                            6;
 35                            10;
 36                    endreduce;
 37            when 6 =>
 38                    reduce *
 39                            2;
 40                            90;
 41                            10;
 42                    endreduce;
 43            others =>
 44                    100;
 45    endcase;
 46  end;

Compiled Successfully.

Result = 1800
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct case and if statements are met, multiple parameters are both stored properly, multiplication reduction is properly evaluated.

Week 6: Project 3

(Test Case 16)

Aspect tested: OTHERS. An input file containing both case and if statements, relational operators, logical operators, multiple parameters, and reductions.

Input: test3.txt, input 70 5

Expected Output: others => **100**

```
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$ ./compile < /home/john/Documents/'CMSC 430'/JohnKucera-Project3-Tests/test3.txt 70 5

  1  -- This tests nested conditional statements, relation operators, logical operators, multiple parameters, and reduction evaluation.
  2
  3  function test3 a: integer, b: integer returns integer;
  4    c: integer is 10;
  5  begin
  6    case a is
  7            when 1 =>
  8                    if a < b and b > c then
  9                            1.1;
 10                    else
 11                            2.2;
 12                    endif;
 13            when 2 =>
 14                    if b >= c or b <= a then
 15                            3.3;
 16                    else
 17                            4.4;
 18                    endif;
 19            when 3 =>
 20                    if a /= b then
 21                            5.5;
 22                    else
 23                            6.6;
 24                    endif;
 25            when 4 =>
 26                    if not (a = b) then
 27                            7.7;
 28                    else
 29                            8.8;
 30                    endif;
 31            when 5 =>
 32                    reduce +
 33                            20;
 34                            6;
 35                            10;
 36                    endreduce;
 37            when 6 =>
 38                    reduce *
 39                            2;
 40                            90;
 41                            10;
 42                    endreduce;
 43            others =>
 44                    100;
 45    endcase;
 46  end;

Compiled Successfully.

Result = 100
john@uwubuntu:~/Documents/CMSC 430/JohnKucera-Project3$
```

Compilation output: SUCCESS. Correct others statement is met.

Week 6: Project 3

Lessons Learned

The biggest lesson I learned in this project was to be conscious exactly of what languages you are working with and how all the pieces fit together. I initially got stumped a lot because I did not register how the tokens would be interpreted. I thought there would be some bison or flex function that has its own way of interpreted the tokens, and I felt like I was in the dark because of that. Once I really started to read the code and understand what was happening, I realized it was mostly the C++ code at work to determine the meanings and evaluations of this custom language. With that in mind, figuring my way through the modifications and additions became much simpler because it was all C++. I already knew how to evaluate every one of these tokens, I just blinded myself by forgetting the presence of C++ in this program. Again, I needed to step back and look at the big picture.

A more specific I learned were with the use of dollar signs $ in the code. This is the first time I really figured out and understood how they are used with identifiers and object-oriented programming languages. Understanding them helped me write many interpretations in parser.y.

Overall, I am getting a lot of learning out of this arrangement of projects (lexical analyzer, syntactic analyzer, interpreter). Before this class, all “compile” meant to me was the translation of languages so that the computer can understand what the program does. Now, understanding how the inside of a compiler works, as well as writing parts of it myself, gives me more confidence in running any other program in an IDE now.