PROJECT 4 REPORT

CMSC 430 – COMPILER THEORY AND DESIGN

Robert D. Carswell

5 March 2024

Table of Contents

Executive Summary

The project involved a comprehensive enhancement of the semantic analysis phase within a compiler framework, necessitating the incorporation of robust checks to identify various semantic errors. These errors were governed by a set of stringent rules that defined the language's static semantics, encompassing regulations related to scoping and type correspondence. Within this context, meticulous attention was given to ensuring the integrity of variable names within their respective scopes and enforcing uniformity in type usage across expressions and statements.

An essential aspect of the project revolved around conducting additional assessments to bolster the compiler's semantic analysis capabilities. This involved verifying type mismatches, scrutinizing the proper usage of arithmetic operators, and examining list declarations and accesses for adherence to language specifications. Of particular importance was the strict enforcement of type coercion rules, which required careful consideration during implementation to ensure compliance with language standards.

Key semantic checks were aimed at detecting inconsistencies in variable initialization types, ensuring uniformity in types within constructs like when statements and switch cases, and validating arithmetic operations' adherence to numeric type requirements. Furthermore, the project called for comprehensive measures to address issues such as undeclared and duplicate identifiers, which could potentially lead to erroneous program behavior if left unchecked.

To achieve the project's objectives, significant modifications were made to existing functions, particularly checkAssignment and checkArithmetic, to accommodate the handling of mixed-type assignments and facilitate coercion of integer types to real types. These adjustments were critical for aligning the compiler's behavior with the language's semantic specifications and ensuring robust error detection and reporting. Overall, the project aimed to enhance the compiler's semantic analysis capabilities, thereby improving the reliability and correctness of the compiled code produced by the system.

Testing

Test Case Table

Test Cases	Description	Expected Output	Actual Output	Pass/Fail
Test Case 0	Compile program	Compiles with no shift/reduce conflicts	Compiles with no shift/reduce conflicts	Pass
Test Case 1	Read test file semantic1.txt	File contents with Line 5, Error Msg: Semantic Error, Type Mismatch on Variable Initialization Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 5, Error Msg: Semantic Error, Type Mismatch on Variable Initialization Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 2	Read test file semantic2.txt	File contents with Line 6, Error Msg: Semantic Error, When Types Mismatch Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 6, Error Msg: Semantic Error, When Types Mismatch Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 3	Read test file semantic3.txt	File contents with Line 11, Error Msg: Semantic Error, Switch Expression Not Integer Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 11, Error Msg: Semantic Error, Switch Expression Not Integer Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 4	Read test file semantic4.txt	File contents with Line 9, Error Msg: Semantic Error, Case Types Mismatch Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 9, Error Msg: Semantic Error, Case Types Mismatch Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass

Test Cases	Description	Expected Output	Actual Output	Pass/Fail
Test Case 5	Read test file semantic5.txt	File contents with Line 7, Error Msg: Semantic Error, Integer Type Required Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 7, Error Msg: Semantic Error, Integer Type Required Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 6	Read test file semantic6.txt	File contents with Line 6, Error Msg: Semantic Error, Undeclared Scalar b Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 6, Error Msg: Semantic Error, Undeclared Scalar b Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 7	Read test file semantic7.txt	File contents with Line 6, Error Msg: Semantic Error, Undeclared List primes Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 6, Error Msg: Semantic Error, Undeclared List primes Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 8	Read test file valid1.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case 9	Read test file valid2.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case 10	Read test file valid3.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass

Test Cases	Description	Expected Output	Actual Output	Pass/Fail
Test Case 11	Read test file semantic8.txt	File contents with Line 5, Error Msg: Semantic Error, List Element Types Do Not Match Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 5, Error Msg: Semantic Error, List Element Types Do Not Match Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 11.a	Read test file semantic8a.txt	File contents with Line 5, Error Msg: Semantic Error, List Element Types Do Not Match Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 5, Error Msg: Semantic Error, List Element Types Do Not Match Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 11.b	Read test file semantic8b.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	
Test Case 12	Read test file semantic9.txt	File contents with Line 5, Error Msg: Semantic Error, List Type Does Not Match Element Types Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 5, Error Msg: Semantic Error, List Type Does Not Match Element Types Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 12.a	Read test file semantic9a.txt	File contents with Line 5, Error Msg: Semantic Error, List Type Does Not Match Element Types Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0	File contents with Line 5, Error Msg: Semantic Error, List Type Does Not Match Element Types Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0	Pass

Test Cases	Description	Expected Output	Actual Output	Pass/Fail
		Semantic Errors 1	Semantic Errors 1	
Test Case 12.b	Read test file semantic9b.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case 13	Read test file semantic10.txt	File contents with Line 7, Error Msg: Semantic Error, List Subscript Must Be Integer Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 7, Error Msg: Semantic Error, List Subscript Must Be Integer Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 13.a	Read test file semantic10a.txt	File contents with Line 7, Error Msg: Semantic Error, List Subscript Must Be Integer Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 7, Error Msg: Semantic Error, List Subscript Must Be Integer Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 13.b	Read test file semantic10b.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case 14	Read test file semantic11.txt	File contents with Line 8, Error Msg: Semantic Error, Character Literals Cannot be Compared to Numeric Expressions Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 8, Error Msg: Semantic Error, Character Literals Cannot be Compared to Numeric Expressions Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case	Read test file semantic11a.txt	File contents with Line 10, Error Msg:	File contents with Line 10, Error Msg:	Pass

Test Cases	Description	Expected Output	Actual Output	Pass/Fail
14.a		Semantic Error, Character Literals Cannot be Compared to Numeric Expressions Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Semantic Error, Character Literals Cannot be Compared to Numeric Expressions Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	
Test Case 14.b	Read test file semantic11b.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case 15	Read test file semantic12.txt	File contents with Line 6, Error Msg: Semantic Error, Arithmetic Operator Requires Numeric Types File contents with Line 8, Error Msg: Semantic Error, Arithmetic Operator Requires Numeric Types Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 2	File contents with Line 6, Error Msg: Semantic Error, Arithmetic Operator Requires Numeric Types File contents with Line 8, Error Msg: Semantic Error, Arithmetic Operator Requires Numeric Types Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 2	Pass
Test Case 15.a	Read test file semantic12a.txt	File contents with Line 6, Error Msg: Semantic Error, Type Mismatch on Variable Initialization Line 9, Error Msg: Semantic Error, Illegal Narrowing Function Return	File contents with Line 6, Error Msg: Semantic Error, Type Mismatch on Variable Initialization Line 9, Error Msg: Semantic Error, Illegal Narrowing Function Return	Pass

Test Cases	Description	Expected Output	Actual Output	Pass/Fail
		Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 2	Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 2	
Test Case 15.b	Read test file semantic12b.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case 16	Read test file semantic13.txt	File contents with Line 6, Error Msg: Semantic Error, Remainder Operator Requires Integer Operands Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 6, Error Msg: Semantic Error, Remainder Operator Requires Integer Operands Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 16.a	Read test file semantic13a.txt	File contents with Line 6, Error Msg: Semantic Error, Remainder Operator Requires Integer Operands Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 6, Error Msg: Semantic Error, Remainder Operator Requires Integer Operands Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 16.b	Read test file semantic13b.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case 17	Read test file semantic14.txt	File contents with Line 12, Error Msg: Semantic Error, If-Elsif-Else Type Mismatch Msg (Bottom of file): Lexical Errors 0	File contents with Line 12, Error Msg: Semantic Error, If-Elsif-Else Type Mismatch Msg (Bottom of file): Lexical Errors 0	Pass

Test Cases	Description	Expected Output	Actual Output	Pass/Fail
		Syntax Errors 0 Semantic Errors 1	Syntax Errors 0 Semantic Errors 1	
Test Case 17.a	Read test file semantic14a.txt	File contents with Line 12, Error Msg: Semantic Error, If-Elsif-Else Type Mismatch Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 12, Error Msg: Semantic Error, If-Elsif-Else Type Mismatch Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 17.b	Read test file semantic14b.txt	File contents with Line 14, Error Msg: Semantic Error, If-Elsif-Else Type Mismatch Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 14, Error Msg: Semantic Error, If-Elsif-Else Type Mismatch Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 17.c	Read test file semantic14c.txt	File contents with Line 10, Error Msg: Semantic Error, If-Elsif-Else Type Mismatch Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 10, Error Msg: Semantic Error, If-Elsif-Else Type Mismatch Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 17.d	Read test file semantic14d.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case 18	Read test file semantic15.txt	File contents with Line 6, Error Msg: Semantic Error, Fold Requires A Numeric List Msg (Bottom of file):	File contents with Line 6, Error Msg: Semantic Error, Fold Requires A Numeric List Msg (Bottom of file):	Pass

Test Cases	Description	Expected Output	Actual Output	Pass/Fail
		Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	
Test Case 18.a	Read test file semantic15a.txt	File contents with Line 6, Error Msg: Semantic Error, List Element Types Do Not Match Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 6, Error Msg: Semantic Error, List Element Types Do Not Match Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 18.b	Read test file semantic15b.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case 18.c	Read test file semantic15c.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case 19	Read test file semantic16.txt	File contents with Line 5, Error Msg: Semantic Error, Illegal Narrowing Variable Initialization Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 5, Error Msg: Semantic Error, Illegal Narrowing Variable Initialization Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 19.a	Read test file semantic16a.txt	File contents with Line 5, Error Msg: Semantic Error, Illegal Narrowing Variable Initialization Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 5, Error Msg: Semantic Error, Illegal Narrowing Variable Initialization Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case	Read test file semantic16b.txt	File contents with:	File contents with:	

Test Cases	Description	Expected Output	Actual Output	Pass/Fail
19.b		Msg (Bottom of file): Compilation Successful	Msg (Bottom of file): Compilation Successful	
Test Case 20	Read test file semantic17.txt	File contents with Line 12, Error Msg: Semantic Error, Illegal Narrowing Function Return Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	File contents with Line 12, Error Msg: Semantic Error, Illegal Narrowing Function Return Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 1	Pass
Test Case 20.a	Read test file semantic17a.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case 20.b	Read test file semantic17b.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case 21	Read test file semantic18.txt	File contents with Line 6, Error Msg: Semantic Error, Duplicate Scalar scalar Line 8, Error Msg: Semantic Error, Duplicate List a_list Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 2	File contents with Line 6, Error Msg: Semantic Error, Duplicate Scalar scalar Line 8, Error Msg: Semantic Error, Duplicate List a_list Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 2	Pass
Test Case 21.a	Read test file semantic18a.txt	File contents with: Msg (Bottom of file): Compilation Successful	File contents with: Msg (Bottom of file): Compilation Successful	Pass
Test Case	Read test file semantic19.txt	File contents with Line 5, Error Msg:	File contents with Line 5, Error Msg:	Pass

Test Cases	Description	Expected Output	Actual Output	Pass/Fail
22		Semantic Error, Illegal Narrowing Variable Initialization Line 6, Error Msg: Semantic Error, List Type Does Not Match Element Types Line 7, Error Msg: Semantic Error, Type Mismatch on Variable Initialization Line 10, Error Msg: Semantic Error, Fold Requires A Numeric List Line 11, Error Msg: Semantic Error, Undeclared Scalar name Line 12, Error Msg: Semantic Error, List Element Types Do Not Match Line 14, Error Msg: Semantic Error, When Types Mismatch Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 7	Semantic Error, Illegal Narrowing Variable Initialization Line 6, Error Msg: Semantic Error, List Type Does Not Match Element Types Line 7, Error Msg: Semantic Error, Type Mismatch on Variable Initialization Line 10, Error Msg: Semantic Error, Fold Requires A Numeric List Line 11, Error Msg: Semantic Error, Undeclared Scalar name Line 12, Error Msg: Semantic Error, List Element Types Do Not Match Line 14, Error Msg: Semantic Error, When Types Mismatch Msg (Bottom of file): Lexical Errors 0 Syntax Errors 0 Semantic Errors 7	

Test Case Screenshots

Test Case 0

```
PS C:\cygwin64\home\rober\Project4> make
flex scanner.l
mv lex.yy.c scanner.c
g++ -c scanner.c
bison -d -v parser.y
mv parser.tab.c parser.c
cp parser.tab.h tokens.h
g++ -c parser.c
g++ -c listing.cc
g++ -c types.cc
```

Test Case 1

```
   1  // semantic1.txt
   2  // Variable Initialization Mismatch
   3
   4  function main returns integer;
   5      value: integer is 'A';
Semantic Error, Type Mismatch on Variable Initialization
   6  begin
   7      1;
   8  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 2

```
   1  // semantic2.txt
   2  //  When Types Mismatch
   3
   4  function main returns integer;
   5  begin
   6      when 2 < 1, 1 : 'a';
Semantic Error, When Types Mismatch
   7  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 3

```
  1  // semantic3.txt
  2  // Non Integer Switch Expression
  3
  4  function main returns integer;
  5      b: character is 'A';
  6  begin
  7      switch b is
  8          case 1 => 2;
  9          case 2 => 4;
 10          others => 6;
 11      endswitch;
Semantic Error, Switch Expression Not Integer
 12  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 4

```
  1  // semantic4.txt
  2  // Case Types Mismatch
  3
  4  function main returns integer;
  5      b: character is 'b';
  6  begin
  7      switch 1 is
  8          case 1 => 2;
  9          case 2 => b;
Semantic Error, Case Types Mismatch
 10          others => 6;
 11      endswitch;
 12  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 5

```
  1  // semantic5.txt
  2  // Using Character Variable with Arithmetic Operator
  3
  4  function main returns integer;
  5      b: character is 'b';
  6  begin
  7      b + 10;
Semantic Error, Integer Type Required
  8  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 6

```
  1  // semantic6.txt
  2  // Undeclared Scalar Variable
  3
  4  function main returns integer;
  5  begin
  6      2 * b + 3;
Semantic Error, Undeclared Scalar b
  7  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 7

```
  1  // semantic7.txt
  2  // Undeclared List Variable
  3
  4  function main returns integer;
  5  begin
  6      primes(1) + 1;
Semantic Error, Undeclared List primes
  7  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 8

```
1  -- valid1.txt
2  -- Program with a Real Variable
3
4  function main returns real;
5      a: real is 4.5;
6  begin
7      a;
8  end;

Compiled Successfully
```

Test Case 9

```
1  -- valid2.txt
2  -- Program with a Hexadecimal Literals
3
4  function main returns integer;
5      a: integer is #A;
6  begin
7      a + #a;
8  end;

Compiled Successfully
```

Test Case 10

```
1  -- valid3.txt
2  -- Program with a Real Variable
3
4  function main returns real;
5      a: real is 4 + 4.5;
6  begin
7      a;
8  end;

Compiled Successfully
```

Test Case 11

```
   1  // semantic8.txt
   2  // List with Elements of Different Types
   3
   4  function main returns integer;
   5      aList: list of integer is (1, 2, 3.5);
Semantic Error, List Element Types Do Not Match
   6  begin
   7      aList(1);
   8  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 11.a

```
   1  // semantic8a.txt
   2  // List with Elements of Different Types
   3
   4  function main returns integer;
   5      aList: list of integer is (1.5, 2, 3);
Semantic Error, List Element Types Do Not Match
   6  begin
   7      aList(1);
   8  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 11.b

```
   1  // semantic8b.txt
   2  // List with Elements of Different Types
   3
   4  function main returns integer;
   5      aList: list of integer is (1, 2, 3);
   6  begin
   7      aList(1);
   8  end;

Compiled Successfully
```

Test Case 12

```
   1  // semantic9.txt
   2  // List Type Does Not Match Element Types
   3
   4  function main returns character;
   5      aList: list of character is (1, 2, 3);
Semantic Error, List Type Does Not Match Element Types
   6  begin
   7      aList(1);
   8  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 12.a

```
   1  // semantic9a.txt
   2  // List Type Does Not Match Element Types
   3
   4  function main returns integer;
   5      aList: list of integer is (1.5, 2.5, 3.5);
Semantic Error, List Type Does Not Match Element Types
   6  begin
   7      aList(1);
   8  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 12.b

```
   1  // semantic9b.txt
   2  // List Type Does Not Match Element Types
   3
   4  function main returns integer;
   5      aList: list of integer is (1, 2, 3);
   6  begin
   7      aList(1);
   8  end;

Compiled Successfully
```

Test Case 13

```
  1  // semantic10.txt
  2  // List Subscript is not Integer
  3
  4  function main returns integer;
  5      aList: list of integer is (1, 2, 3);
  6  begin
  7      aList(1.5);
Semantic Error, List Subscript Must Be Integer
  8  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 13.a

```
  1  // semantic10a.txt
  2  // List Subscript is not Integer
  3
  4  function main returns integer;
  5      aList: list of integer is (1, 2, 3);
  6  begin
  7      aList('a');
Semantic Error, List Subscript Must Be Integer
  8  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 13.b

```
  1  // semantic10b.txt
  2  // List Subscript is not Integer
  3
  4  function main returns integer;
  5      aList: list of integer is (1, 2, 3);
  6  begin
  7      aList(1);
  8  end;

Compiled Successfully
```

Test Case 14

```
   1  -- semantic11.txt
   2  -- Mixing Numeric and Character Types with Relational Operator
   3
   4  function main returns integer;
   5  begin
   6      if 'b' < 'c' then
   7          1;
   8      elsif 1 < 'b' then
Semantic Error, Character Literals Cannot be Compared to Numeric Expressions
   9          2;
  10      else
  11          3;
  12      endif;
  13  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 14.a

```
   1  -- semantic11a.txt
   2  -- Mixing Numeric and Character Types with Relational Operator
   3
   4  function main returns integer;
   5  begin
   6      if 'b' < 'c' then
   7          1;
   8      elsif 1 < 2 then
   9          2;
  10      elsif 'b' < 1 then
Semantic Error, Character Literals Cannot be Compared to Numeric Expressions
  11          2;
  12      else
  13          3;
  14      endif;
  15  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 14.b

```
  1  -- semantic11b.txt
  2  -- Mixing Numeric and Character Types with Relational Operator
  3
  4  function main returns integer;
  5  begin
  6      if 'b' < 'c' then
  7          1;
  8      elsif 1 < 2 then
  9          2;
 10      else
 11          3;
 12      endif;
 13  end;

Compiled Successfully
```

Test Case 15

```
  1  // semantic12.txt
  2  // Using Character Literal with Exponentiation Operator
  3  //     and Negation Operator
  4
  5  function main returns integer;
  6      c: character is ~'c';
Semantic Error, Arithmetic Operator Requires Numeric Types
  7  begin
  8      5 ^ 'P';
Semantic Error, Arithmetic Operator Requires Numeric Types
  9  end;
 10

Lexical Errors 0
Syntax Errors 0
Semantic Errors 2
```

Test Case 15.a

```
  1  // semantic12a.txt
  2  // Using Character Literal with Exponentiation Operator
  3  //     and Negation Operator
  4
  5  function main returns integer;
  6      c: character is ~1;
Semantic Error, Type Mismatch on Variable Initialization
  7  begin
  8      5 ^ 1.5;
  9  end;
Semantic Error, Illegal Narrowing Function Return
 10

Lexical Errors 0
Syntax Errors 0
Semantic Errors 2
```

Test Case 15.b

```
  1  // semantic12b.txt
  2  // Using Character Literal with Exponentiation Operator
  3  //     and Negation Operator
  4
  5  function main returns real;
  6      c: real is ~1.0;
  7  begin
  8      5 ^ 1.5;
  9  end;
 10

Compiled Successfully
```

Test Case 16

```
   1  // semantic13.txt
   2  // Mixing Real Literals with the Remainder Operator
   3
   4  function main returns integer;
   5  begin
   6      4 % 4.8;
Semantic Error, Remainder Operator Requires Integer Operands
   7  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 16.a

```
   1  // semantic13a.txt
   2  // Mixing Real Literals with the Remainder Operator
   3
   4  function main returns real;
   5  begin
   6      4.5 % 4;
Semantic Error, Remainder Operator Requires Integer Operands
   7  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 16.b

```
   1  // semantic13b.txt
   2  // Mixing Real Literals with the Remainder Operator
   3
   4  function main returns integer;
   5  begin
   6      10 % 4;
   7  end;

Compiled Successfully
```

Test Case 17

```
  1  -- semantic14.txt
  2  -- If Elsif Else Mismatch
  3
  4  function main returns integer;
  5  begin
  6      if 9 < 10 then
  7          1;
  8      elsif 8 = 1 then
  9          2;
 10      else
 11          3.7;
 12      endif;
Semantic Error, If-Elsif-Else Type Mismatch
 13  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 17.a

```
  1  -- semantic14a.txt
  2  -- If Elsif Else Mismatch
  3
  4  function main returns integer;
  5  begin
  6      if 9 < 10 then
  7          1;
  8      elsif 8 = 1 then
  9          2.8;
 10      else
 11          3;
 12      endif;
Semantic Error, If-Elsif-Else Type Mismatch
 13  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 17.b

```
  1  -- semantic14b.txt
  2  -- If Elsif Else Mismatch
  3
  4  function main returns integer;
  5  begin
  6      if 9 < 10 then
  7          1;
  8      elsif 8 = 1 then
  9          7;
 10      elsif 7 = 10 then
 11          1.8;
 12      else
 13          3;
 14      endif;
Semantic Error, If-Elsif-Else Type Mismatch
 15  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 17.c

```
  1  -- semantic14c.txt
  2  -- If Elsif Else Mismatch
  3
  4  function main returns integer;
  5  begin
  6      if 9 < 10 then
  7          1;
  8      else
  9          3.5;
 10      endif;
Semantic Error, If-Elsif-Else Type Mismatch
 11  end;
 12

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 17.d

```
  1  -- semantic14d.txt
  2  -- If Elsif Else Mismatch
  3
  4  function main returns integer;
  5  begin
  6      if 9 < 10 then
  7          1;
  8      elsif 8 = 1 then
  9          7;
 10      elsif 7 = 10 then
 11          8;
 12      else
 13          3;
 14      endif;
 15  end;
 16

Compiled Successfully
```

Test Case 18

```
  1  // semantic15.txt
  2  // Folding a nonnumeric List
  3
  4  function main returns integer;
  5  begin
  6      fold left + ('a', 'b', 'c') endfold;
Semantic Error, Fold Requires A Numeric List
  7  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 18.a

```
   1  // semantic15a.txt
   2  // Folding a nonnumeric List
   3
   4  function main returns integer;
   5  begin
   6      fold left + (5, 1.5, 2) endfold;
Semantic Error, List Element Types Do Not Match
   7  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 18.b

```
   1  // semantic15b.txt
   2  // Folding a nonnumeric List
   3
   4  function main returns integer;
   5  begin
   6      fold left + (5, 1, 2) endfold;
   7  end;

Compiled Successfully
```

Test Case 18.c

```
   1  // semantic15c.txt
   2  // Folding a nonnumeric List
   3
   4  function main returns real;
   5  begin
   6      fold left + (5.3, 1.5, 2.4) endfold;
   7  end;

Compiled Successfully
```

Test Case 19

```
   1  -- semantic16.txt
   2  -- Narrowing Variable Initialization
   3
   4  function main returns integer;
   5      b: integer is 5 * 2.5;
Semantic Error, Illegal Narrowing Variable Initialization
   6  begin
   7      b + 1;
   8  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 19.a

```
   1  -- semantic16a.txt
   2  -- Narrowing Variable Initialization
   3
   4  function main returns integer;
   5      b: integer is 5.5 * 2;
Semantic Error, Illegal Narrowing Variable Initialization
   6  begin
   7      b + 1;
   8  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 1
```

Test Case 19.b

```
   1  -- semantic16b.txt
   2  -- Narrowing Variable Initialization
   3
   4  function main returns real;
   5      b: real is 5.5 * 2;
   6  begin
   7      b + 1;
   8  end;

Compiled Successfully
```

Test Case 20

```
  1  -- semantic17.txt
  2  -- Narrowing Function Return
  3
  4  function main returns integer;
  5      b: integer is 6 * 2;
  6  begin
  7      if 8 < 0 then
  8          b + 3.0;
  9      else
 10          b * 4.6;
 11      endif;
 12  end;
Semantic Error, Illegal Narrowing Function Return
```

Test Case 20.a

```
  1  -- semantic17a.txt
  2  -- Narrowing Function Return
  3
  4  function main returns integer;
  5      b: integer is 6 * 2;
  6  begin
  7      if 8 < 0 then
  8          b + 3;
  9      else
 10          b * 4;
 11      endif;
 12  end;

Compiled Successfully
```

Test Case 20.b

```
  1  -- semantic17b.txt
  2  -- Narrowing Function Return
  3
  4  function main returns real;
  5      b: real is 6.0 * 2;
  6  begin
  7      if 8 < 0 then
  8          b + 3.0;
  9      else
 10          b * 4.6;
 11      endif;
 12  end;

Compiled Successfully
```

Test Case 21

```
  1  -- semantic18.txt
  2  -- S18 Duplicate Scalar and List Variables
  3
  4  function main returns integer;
  5      scalar: integer is 4 * 2;
  6      scalar: character is 'b';
Semantic Error, Duplicate Scalar scalar
  7      a_list: list of integer is (4, 2);
  8      a_list: list of real is (2.3, 4.4);
Semantic Error, Duplicate List a_list
  9  begin
 10      1;
 11  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 2
```

Test Case 21.a

```
  1  -- semantic18a.txt
  2  -- S18a Duplicate Scalar and List Variables
  3
  4  function main returns integer;
  5      scalar: integer is 4 * 2;
  6      b_scalar: character is 'b';
  7      c_scalar: integer is 4 * 2;
  8      a_list: list of integer is (4, 2);
  9      b_list: list of real is (2.3, 4.4);
 10      c_list: list of real is (2.3, 4.4);
 11  begin
 12      1;
 13  end;

Compiled Successfully
```

Test Case 22

```
  1  // semantic19.txt
  2  // Multiple Semantic Errors
  3
  4  function main returns integer;
  5      value: integer is 4.5;
Semantic Error, Illegal Narrowing Variable Initialization
  6      numbers: list of real is (1, 2, 3);
Semantic Error, List Type Does Not Match Element Types
  7      one: integer is '1';
Semantic Error, Type Mismatch on Variable Initialization
  8  begin
  9      if value > 0 then
 10          fold left + ('a', 'b') endfold;
Semantic Error, Fold Requires A Numeric List
 11      elsif name = 'N' then
Semantic Error, Undeclared Scalar name
 12          fold right * (1, 2.5) endfold;
Semantic Error, List Element Types Do Not Match
 13      else
 14          when value < 10, 1 : 1.5;
Semantic Error, When Types Mismatch
 15      endif;
 16  end;

Lexical Errors 0
Syntax Errors 0
Semantic Errors 7
```

Approach

In initiating the project, I thoroughly reviewed the requirements and the accompanying make file. This foundational step entailed meticulously examining the production elements to understand the system's data flow comprehensively. I then proceeded to transfer pertinent components from Project 2 to Project 4, a pivotal move aimed at dissecting the test execution process. Leveraging C++ references, I meticulously scrutinized the underlying structures to ensure seamless integration and prevent potential compilation errors arising from conflicting types.

Upon completing the integration process, I conducted an exhaustive testing phase to validate the program's behavior across diverse scenarios. I meticulously evaluated the program's responses using semantic and valid test cases, including (semantic1.txt–semantic7.txt) and (valid1.txt–valid3.txt). This rigorous testing regimen was instrumental in preempting any unforeseen issues or faults that could arise during the integration phase. Additionally, I diligently documented all steps taken during integration and testing to provide a comprehensive record for future reference, including detailed descriptions of code modifications and test case results.

With the integration complete, my focus shifted towards incorporating concrete data types into the system architecture. This endeavor necessitated meticulous adjustments to various components, including the type enumeration in types.h and attribute assignments in the scanner.l. Following these modifications, I conducted thorough testing to ensure the program's seamless handling of genuine data types. This phase involved running various scenarios to validate the system's ability to accurately

process and manipulate the new data types, update documentation, and provide clear instructions for future development.

I addressed specific enhancements to improve the compiler's functionality and efficiency by identifying hexadecimal literals as integer types, enforcing type coercion in arithmetic expressions, and ensuring uniformity in list element types. These enhancements required intricate modifications in both the scanner and parser files, possibly necessitating the introduction of additional functions to facilitate robust type-checking. By enhancing the accuracy and efficiency of the compiler's type-checking process, the program can now handle a broader range of input scenarios and produce more reliable output.

Throughout the iterative development process, I diligently employed the provided test cases to rigorously evaluate the program's behavior and verify the accurate reporting of semantic errors for each implemented patch. While promptly resolving any identified type clashes to ensure the seamless integration and smooth operation of all program components. Upon completing all revisions and successful testing with the provided and personal test cases, I deemed the project finalized and ready for submission, confident in its robustness and adherence to specified requirements.

Lessons Learned

Throughout the project, I gained invaluable insights and honed my skills in various aspects of software development. One notable instance occurred during the integration of the two projects, where I observed that the skeleton functions in types.cc lacked brackets, prompting me to delve into their behavior. This investigation revealed that the conditional if statement only applied to the following line, deepening my

understanding of program flow control mechanisms. Continuing through this phase, I meticulously examined the existing Bison functions within the skeleton, leveraging their mechanics to devise a function for detecting duplicate variables. However, I overlooked a crucial detail regarding the "type of list" when adding semantic actions to the primary production for subscripted lists.

The oversight spurred me to develop a validation function that compares the subscripted list expression to an integer type but fails to account for list type mismatches. Although this oversight did not immediately manifest in errors during initial testing, it became apparent when implementing check assignments in the parser's main function. Subsequently, the inclusion of test9.txt revealed an unexpected error message at the program's conclusion: "Type Mismatch on Function Return." While technically accurate, the discrepancy between the list type and the list subscript type should have been detected earlier. To address this issue, I utilized the find function from a preceding semantic action to relay the error to the checkListsSubscript function. This adjustment enabled the function to bypass the check and propagate the mismatch flag if a previous error had occurred, thus providing the check assignment function with the appropriate mismatch flag to prevent the error message.

Despite encountering challenges, I navigated them by continuing to leverage resources such as the C++ reference and IBM's UNIX System Services Programming Tools. These resources played a pivotal role in surmounting obstacles and enriching my skill set, ultimately contributing to the successful completion of the project.

References

cppreference.com. (n.d.). https://en.cppreference.com/

IBM Documentation. (n.d.). https://www.ibm.com/docs/en/zos/3.1.0