Robert D. Carswell

5 March 2024

project 4 report

CMSC 430 – COMPILER THEORY AND DESIGN

# **Table of Contents**

[Executive Summary 3](#_Toc160097967)

[Testing 4](#_Toc160097968)

[Test Case Table 4](#_Toc160097969)

[Test Case Screenshots 14](#_Toc160097970)

[Test Case 0 14](#_Toc160097971)

[Test Case 1 14](#_Toc160097972)

[Test Case 2 15](#_Toc160097973)

[Test Case 3 15](#_Toc160097974)

[Test Case 4 16](#_Toc160097975)

[Test Case 5 16](#_Toc160097976)

[Test Case 6 16](#_Toc160097977)

[Test Case 7 17](#_Toc160097978)

[Test Case 8 17](#_Toc160097979)

[Test Case 9 17](#_Toc160097980)

[Test Case 10 18](#_Toc160097981)

[Test Case 11 18](#_Toc160097982)

[Test Case 11.a 18](#_Toc160097983)

[Test Case 11.b 19](#_Toc160097984)

[Test Case 12 19](#_Toc160097985)

[Test Case 12.a 19](#_Toc160097986)

[Test Case 12.b 20](#_Toc160097987)

[Test Case 13 20](#_Toc160097988)

[Test Case 13.a 20](#_Toc160097989)

[Test Case 13.b 21](#_Toc160097990)

[Test Case 14 21](#_Toc160097991)

[Test Case 14.a 22](#_Toc160097992)

[Test Case 14.b 22](#_Toc160097993)

[Test Case 15 23](#_Toc160097994)

[Test Case 15.a 23](#_Toc160097995)

[Test Case 15.b 24](#_Toc160097996)

[Test Case 16 24](#_Toc160097997)

[Test Case 16.a 24](#_Toc160097998)

[Test Case 16.b 25](#_Toc160097999)

[Test Case 17 25](#_Toc160098000)

[Test Case 17.a 26](#_Toc160098001)

[Test Case 17.b 26](#_Toc160098002)

[Test Case 17.c 27](#_Toc160098003)

[Test Case 17.d 27](#_Toc160098004)

[Test Case 18 28](#_Toc160098005)

[Test Case 18.a 28](#_Toc160098006)

[Test Case 18.b 28](#_Toc160098007)

[Test Case 18.c 29](#_Toc160098008)

[Test Case 19 29](#_Toc160098009)

[Test Case 19.a 29](#_Toc160098010)

[Test Case 19.b 30](#_Toc160098011)

[Test Case 20 30](#_Toc160098012)

[Test Case 20.a 30](#_Toc160098013)

[Test Case 20.b 31](#_Toc160098014)

[Test Case 21 31](#_Toc160098015)

[Test Case 21.a 32](#_Toc160098016)

[Test Case 22 32](#_Toc160098017)

[Approach 33](#_Toc160098018)

[Lessons Learned 34](#_Toc160098019)

[References 36](#_Toc160098020)

# **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 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 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  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.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  14.a | Read test file semantic11a.txt | File contents with  Line 10, 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 10, 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  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  Msg (Bottom of file):  Lexical Errors 0  Syntax Errors 0  Semantic Errors 2 | File contents with  Line 6, Error Msg:  Semantic Error, Type Mismatch on Variable Initialization  Line 9, Error Msg:  Semantic Error, Illegal Narrowing Function Return  Msg (Bottom of file):  Lexical Errors 0  Syntax Errors 0  Semantic Errors 2 | Pass |
| 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  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.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):  Lexical Errors 0  Syntax Errors 0  Semantic Errors 1 | File contents with  Line 6, Error Msg:  Semantic Error, Fold Requires A Numeric List  Msg (Bottom of file):  Lexical Errors 0  Syntax Errors 0  Semantic Errors 1 | Pass |
| 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  19.b | Read test file semantic16b.txt | File contents with:  Msg (Bottom of file):  Compilation Successful | File contents with:  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  22 | Read test file semantic19.txt | File contents with  Line 5, Error Msg:  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 | File contents with  Line 5, Error Msg:  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 | Pass |

# **Test Case Screenshots**

## **Test Case 0**

A screen shot of a computer

Description automatically generated

## **Test Case 1**

A screen shot of a computer

Description automatically generated

## **Test Case 2**

A screen shot of a computer

Description automatically generated

## **Test Case 3**

A screenshot of a computer program

Description automatically generated

## **Test Case 4**

A screenshot of a computer screen

Description automatically generated

## **Test Case 5**

A screenshot of a computer program

Description automatically generated

## **Test Case 6**

A screen shot of a computer program

Description automatically generated

## **Test Case 7**

A screenshot of a computer program

Description automatically generated

## **Test Case 8**

A screen shot of a computer

Description automatically generated

## **Test Case 9**

A computer screen shot of white text

Description automatically generated

## **Test Case 10**

A screen shot of a computer

Description automatically generated

## **Test Case 11**

A screenshot of a computer program

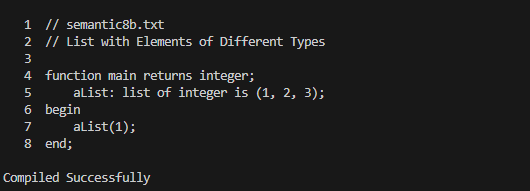
Description automatically generated

### **Test Case 11.a**

A computer screen shot of a black screen

Description automatically generated

### **Test Case 11.b**



## **Test Case 12**

A screenshot of a computer

Description automatically generated

### **Test Case 12.a**

A screenshot of a computer

Description automatically generated

### **Test Case 12.b**

A screen shot of a computer

Description automatically generated

## **Test Case 13**

A computer screen shot of white text

Description automatically generated

### **Test Case 13.a**

A computer screen shot of white text

Description automatically generated

### **Test Case 13.b**

A computer screen with white text

Description automatically generated

## **Test Case 14**

A computer screen with white text

Description automatically generated

### **Test Case 14.a**

A screenshot of a computer program

Description automatically generated

### **Test Case 14.b**

A screenshot of a computer program

Description automatically generated

## **Test Case 15**

A screen shot of a computer

Description automatically generated

### **Test Case 15.a**

A screen shot of a computer program

Description automatically generated

### **Test Case 15.b**

A screen shot of a computer

Description automatically generated

## **Test Case 16**

A screen shot of a computer

Description automatically generated

### **Test Case 16.a**

A screenshot of a computer program

Description automatically generated

### **Test Case 16.b**

A screenshot of a computer

Description automatically generated

## **Test Case 17**

A screenshot of a computer program

Description automatically generated

### **Test Case 17.a**

A screenshot of a computer program

Description automatically generated

### **Test Case 17.b**

A screenshot of a computer program

Description automatically generated

### **Test Case 17.c**

A computer screen shot of a black screen

Description automatically generated

### **Test Case 17.d**

A screenshot of a computer program

Description automatically generated

## **Test Case 18**

A computer screen shot of white text

Description automatically generated

### **Test Case 18.a**

A screenshot of a computer program

Description automatically generated

### **Test Case 18.b**

A computer screen with white text

Description automatically generated

### **Test Case 18.c**

A screenshot of a computer program

Description automatically generated

## **Test Case 19**

A computer screen with white text

Description automatically generated

### **Test Case 19.a**

A screen shot of a computer

Description automatically generated

### **Test Case 19.b**

A screen shot of a computer

Description automatically generated

## **Test Case 20**

A computer screen with white text

Description automatically generated

### **Test Case 20.a**

A computer screen shot of a black screen

Description automatically generated

### **Test Case 20.b**

A screenshot of a computer program

Description automatically generated

## **Test Case 21**

A computer screen shot of a black screen

Description automatically generated

### **Test Case 21.a**

A screenshot of a computer program

Description automatically generated

## **Test Case 22**

A screenshot of a computer program

Description automatically generated

# **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>