Report on the solution of Oblig 2 in IN2031

Student mxyang (mxyang@ifi.uio.no)
Student annaasor (annaasor@ifi.uio.no)

November 15, 2024

1 Introduction

This report documents the reference solution of the Oblig3 assignment. The task was to implement a static type checker for AeroScript based on certain typing rules given in the handout. Section 2 will further explain some of our interpretations.

2 Implementation

Because our grammar was written differently than the handed out precode grammar, we decided to meet in the middle, tweak the precode grammar and change our initial implementation a little too. We changed AcMove to MOVE('to' POINT point OR 'by' expression) as it was more familiar and understandable.

The typechecker works similarly to the interpreter, iterating through the AST. However, instead of creating objects of various classes, the typechecker now checks for type (NUM, POINT) errors. For each relevant statement, we check if the type aligns with the defined typing rule.

3 Testing

We implemented the JUnit test similarly to the JUnit test in oblig2, and using the code examples from the handout to check that our typechecker actually works. We are testing for all well-typed and ill-typed statements.

4 Limitations and Potential Problems

Due to the way we wrote our grammar, we are expecting a "point" after a "move to" statement. Therefore our code will receive a parsing error instead of a typechecker error.

5 Conclusions

We have implemented a static typechecker and a fitting JUnit test for our solution. We judge the implementation to fulfill the necessary minimum requirements for this oblig.