

Report - Phase 3: Semantic Analysis

Yongun Seong 2017-19937

For the this third phase of the project, I implemented semantic analysis to reject syntatically correct yet semantically incorrect programs. I first implemented the type Match and Compare methods that determine whether two types are identical or compatible, respectively. Then, I implemented the type checker, which checked each possible AST node to determine whether its statements and expressions are correctly typed. Finally, I added code to validate compile-time constants, such as constant initialization and array dimensions. I did not handle replacing other constant expressions to the evaluated constants, however.

I also modified the procedure parser to silently replace all array parameters to accept pointers to arrays instead, as the language was defined to pass arrays by reference. Similarly, I modified the function and procedure call parsing code to silently insert an opAddress node instead of the array itself. Furthermore, I noticed that my code did not accept empty returns, even though these were allowed by the specification. I fixed this issue, and the parser now handles return statements correctly.

Currently, my parser accepts a few programs that may be considered invalid, depending on the interpretation of the specification.

For example, it does not require a function to actually contain any return statements. One way to implement this check would have been to require the last statement in a procedure to be a return statement, but this would have rejected arguably correct function bodies such as `if (condition) then return 1 else return 2 end`. For now, I have not implemented check, but I may implement this later on depending on what I learn during the later phases.

I also accept programs that assign integers that are not strictly the same type, as long as they are “compatible”. This only affects numeric types. That is, `longint = int` or `int = longint` are allowed by my parser. Depending on one’s interpretation of the specification, this may be seen as incorrect, but I understood the specification to allow these statements.