

Static Semantics

Due April 24, 2018 at 11:59 pm

Total 50 points.

Invocation:

> statSem [*file*]

Wrong invocations will not be graded.

Graded 90% execution 10% structure/standards.

Stat Semantics Definition

- The only static semantics we impose that can be processed by the compiler (static) are proper use of variables.
- **Variables**
 - Variables have to be defined before used first time (must satisfy syntax too)
 - Variable name can only be defined once in a scope but can be reused in another scope
- **Global**
 - There is only one scope, the global scope, in the case

Suggestions

Modify the main function so that after calling parser and receiving the tree, the main will call the static semantic function on the tree.

Software support

- Use any container (called ST here) for names such as array, list, etc. with the following interface. It shows String as the parameter, which is the ID token instance, but it could include line number or the entire token for more detailed error reporting. This container will process identifier tokens only.
 - insert(String) - insert the string if not already there or error if already there (you may return fail indication or issue detailed error here and exit)

- Bool verify(String) - return true if the string is already in the container variable and false otherwise (Suggest you return false indicator rather than issue detailed error here with exit but either way could possibly work if you assume that no one checks verify() unless to process variable use)

Static semantics

- Instantiate ST as STV for variables
- Traverse the tree and perform the following (looks like preorder traversal) based on the subtree you are visiting
 - If visiting <vars> or <mvars> and you find identifier token then call STV.insert(String) // this is variable definition
 - Otherwise (you are not in <vars> nor <mvars>) if you find identifier token call STV.verify(String)