Comp 442 A4

Alexander De Laurentiis

March 20, 2022

1 List of Semantic Rules Implemented

- Symbol creation phase
  - ☑ 1 Each scope is considered a table.
  - ☑ 2 Each declared class gets its own scope.
  - ☑ 3 For each scope, if there is a parameter, var, or else declared it will be recored in that scope.
  - ☑ 4 This is done for every declaration when traversing a scope.
  - ☑ 5 Will give warning before using *asserta*/1 on a new variable from a scope if its already been asserted.
  - ☑ 6 Will check if there is no body of a function then give the warning when parsing it.
  - ☑ 7 Done.
  - ☑ 8 Will give warning before using *asserta*/1 on a new variable from a scope if its already been asserted.
  - ☒ 9 Gives warning if 2 functions have the same name but separate parameters.
- Semantic Checking phase
  - ☒ 10 Type Check expressions and the sub parts to all be 1 type, and same for assignment
  - ☐ 11 Identifiers must be defined in a scope it is used.
  - ☐ 12 Function calls use the right number of parameters, expressions as parameters are of the right type.
  - ☐ 13 Array variables referred to have the correct quantity of dimmensions.
  - ☐ 14 Circular clss dependencies reported.
  - ☐ 15 The dot operator only being used on variables of the class type, and the right part must be a member of that class.

2 Design

So with the programming language tools at my disposal with prolog, I decided to redo my entire ast output to be a uniform node(Name,[Children]) and terminal(Name,[Value]) format. This way it would be iteratable. Then I proceeded to make a set of predicates that would recursively iterate through the tree structure in a depth first left to right manner assuming full single threaded use.

For my symbol table I use prolog predicates which are asserted into the current session. Prolog is a language meant to traverse search spaces and make logical connections so it seemed most appropriate especially since I could mimic the behavior of a DB or table this way as well. Most entries are of the fomat symbol/4, symbol/5, or symbol/6 depending on the content it contains, but for each type of vardecl, function, or so on they are uniform in the order of info presented, such as a function being symbol(Scope,function,Returns,ID).

For the separate tests I made a predicate for each after the table was generated which would be used to test them then. That way it would pass through each, write to the output stream, then be all good.

3 Tools Used

For this assignment the only tools technically used was emacs as the programming environment with the ediprolog as a package to compile and run my code within buffers, and scryer-prolog set as the prolog distribution to be used in my configuration file. This was done for the conveniences and efficiency offered with scryer-prolog despite its lack of being a fully stable release, it is a recognized release and comparable to sisctus prolog which is completely proprietary while scryer-prolog is open source.

Sidenote of issues with the assert predicate occured and will be opening a github issue most likely or switching prolog dialects.