

Symbol Table

Design a symbol table appropriate for our language which permits function scopes. The symbol table should give the type information for identifiers. Entries for function names should be designed to accommodate types of parameters (in the order), return type (return type is always int. hence, you can mark if there is a return type or not, just.) as well. Our language do not allow to declare global variables, and all the names needs declaration at the beginning of the corresponding function, and the scope of such names are limited to the corresponding function. It is desired to have a common interface for lookups corresponding to every scopes, and the required interface is as follows.

Requirements Specification:

- Input : Concrete Parse Tree (Your Stage-1 output)
- Output : Symbol Table
- Side Effects : Entries for every symbol tables (type information)
- Exceptions : Undeclared, Multiply declared identifiers.

Interface Requirements

- **LookUp (String name)** --> looks if name is defined in the current scope. If yes, it returns the corresponding structure, else returns the structure indicating an error. Identifiers which are not declared are reported.
- **Insert (String name, nameStruct ns)** --> stores the ns (which has type and other relevant information) in the table at location **h(name)**. Multiple declarations of same identifier is reported here. h(.) is the has function

Abstract Syntax Tree

Requirements Specification:

- Input : Concrete Parse Tree (Your Stage-1 output)
- Output : AST ([Find the specification in the S-02 AST.docx/pdf](#))
- Side Effects : Nil
- Exceptions : Context Errors ([Find the specification in the S-02 Type.docx/pdf](#))

Type Checking

Requirements Specification:

- Input : AST
- Output : Type Correct AST ([Find the specification in the S-02 Type.docx/pdf](#))
- Side Effects : Nil
- Exceptions : Type Errors / Semantic Errors

Stage-2 Deliverable

- **C Code** for generating AST and for semantic analysis.
- **makefile** for compilation and execution
- Test cases and output (AST & Error Messages) files for test cases
- Brief **readme** file for directory/program structure, invocation command etc.