YAFSM - Phase 3

CS3423-DSL
Group 6

Mahin Bansal - Project Manager
Satpute Tukaram Aniket - System Architect/Integrator
Harshit Pant - Tester
Vishal Vijay Devadiga - Language Guru

Overview

Phase 3 of DSL design involves the semantic analysis related to the language specification

The semantic checks implemented in our language are:

- LHS - RHS type checking/type coherence
- Function return type checks
- ID redeclaration tests (function names can be repeated by variables though and vice-versa)

Overview(2)

- Division by zero (constant expressions evaluating to zero)
- Finite state machine checks (such as non-inclusion of epsilon transitions in DFAs)
- Declaration before definition checks
- Existence of main function checks
- Initialization of global variables with non-constants checks
- Basic template functions implemented for ease of programming for user
- ....

Symbol Tables

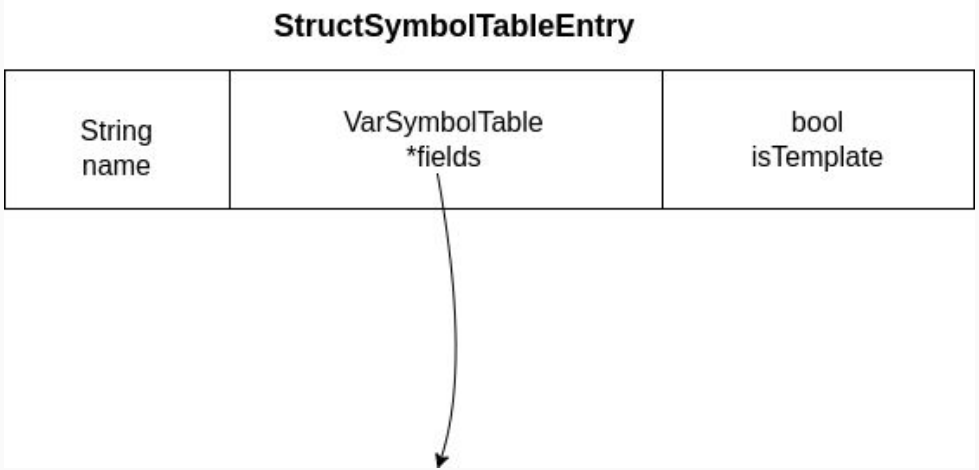
We used three symbol tables

- StructSymbolTable
- FunctionSymbolTable
- VarSymbolTable

Also for nested scopes we use chain of VarSymbolTable implemented as a list (VarSymbolTableList)

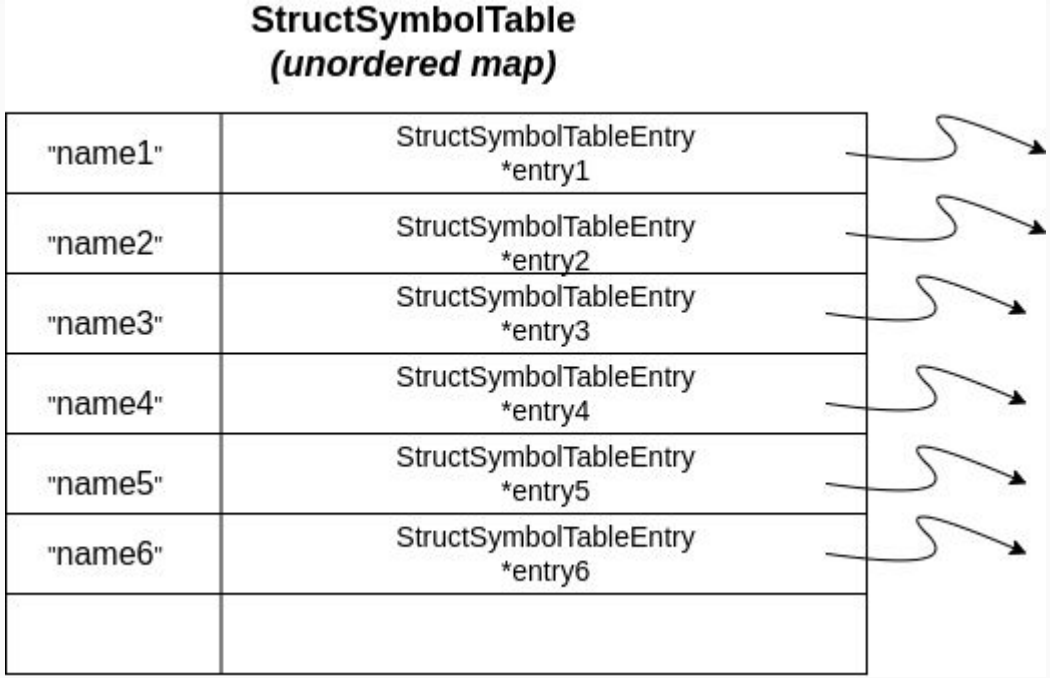
Since function/struct definitions are allowed only in global scope therefore no need for chaining FunctionSymbolTable and StructSymbolTable

StructSymbolTableEntry



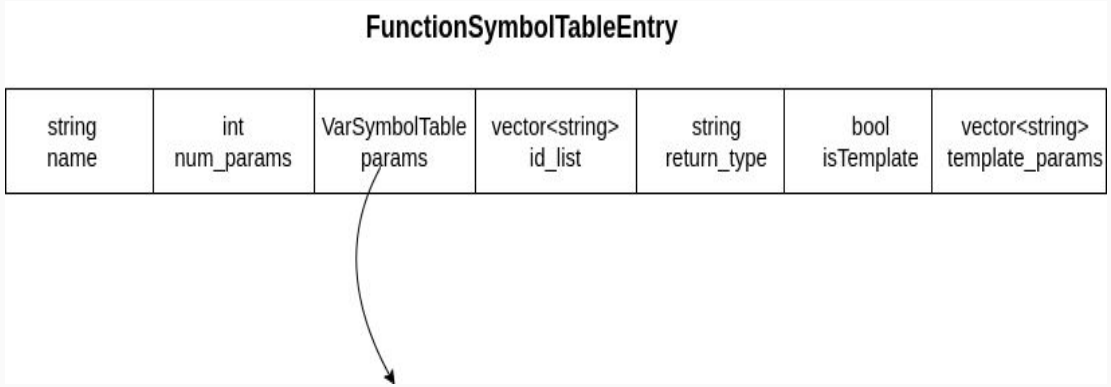
- Struct Table Entry contains a VarSymbolTable *field for information about members of struct
- Typenames for a template function are inserted as structs in struct ST with isTemplate as true and an empty struct variables ST.

StructSymbolTable



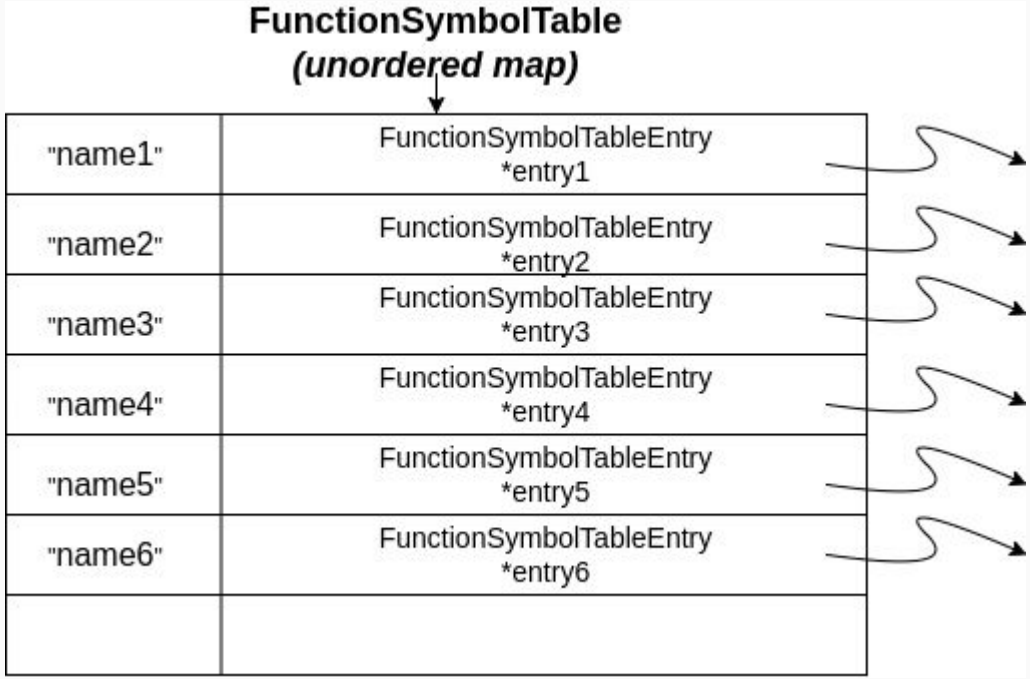
- StructSymbolTable is an unordered map of struct names and struct table entry pointers
- Functions for inserting, removing and initializing struct table at start of compilation are also implemented

FunctionSymbolTableEntry



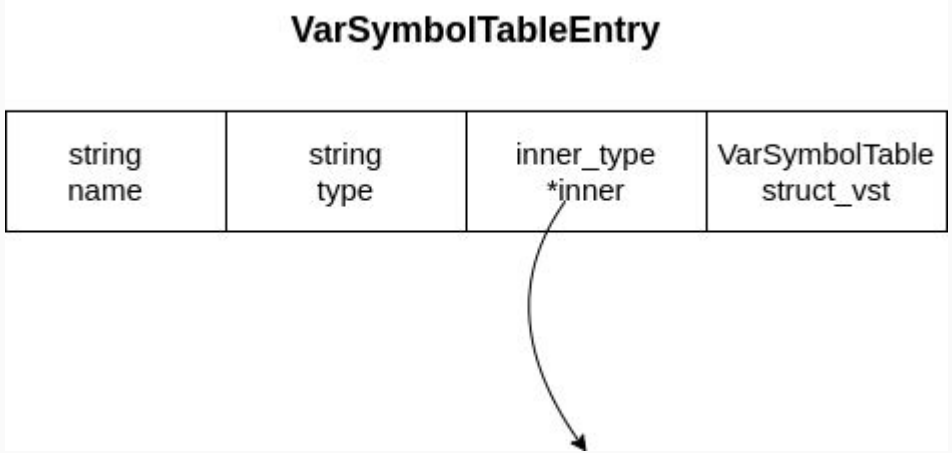
- Variable Symbol Table for parameter information.
- Ordered list of all identifiers of the parameter list
- isTemplate is true for template functions
- Ordered list of all types in case of a template function

FunctionSymbolTable



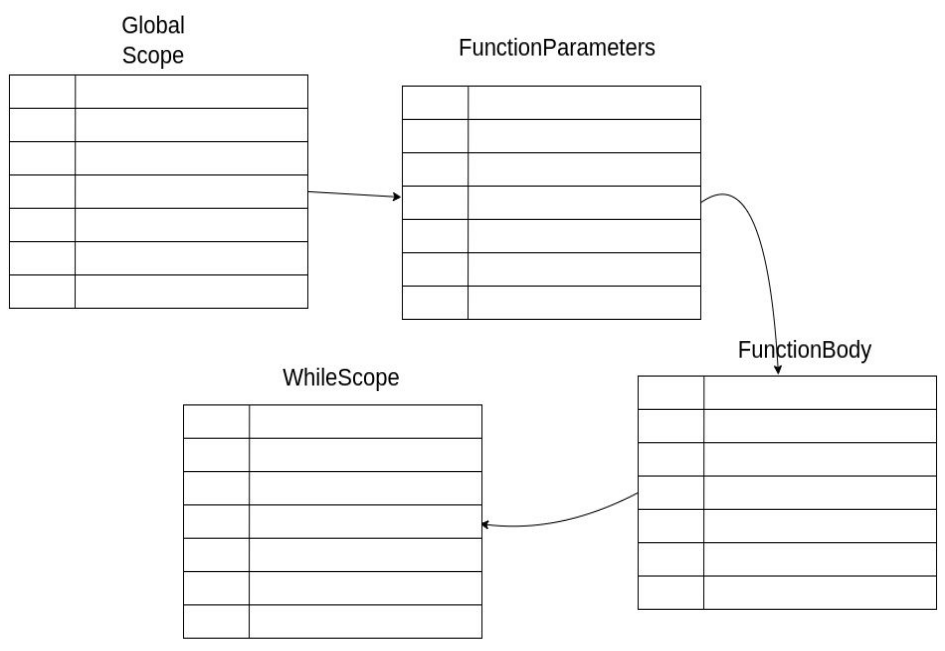
- Unordered map of function names and corresponding FunctionSymbolTableEntry pointers

VarSymbolTableEntry



- Name of variable
- Outer type of variable (int_8...,struct..,o_set...,cfg...)
- Inner type of variable (in case of sets)
- VarSymbolTable* in case it is a struct

VarSymbolTableList



- List of Variable symbol tables in case of nested blocks
- One Global Variable Symbol Table is always present
- Lookup is done from outer scope to inner