

Reporte Fase 3: Análisis de Contexto

Constanza Abarca 13-10000

Pedro Maldonado 13-10790

En esta fase del proyecto se extendió el AST devuelto por el parser construido en la fase 2, agregando los atributos Type, Arreglo y Lexeme necesarios para realizar la verificaciones estáticas relacionadas con los tipos de los terminales y de arreglos para generar expresiones válidas y correctas pertenecientes al lenguaje BasicTran.

En las reglas para terminales (p_terminal) y para asignaciones (p_asignacion) del parser se agregaron verificaciones adicionales para agregar en el árbol el tipo, valor de las variables y el tamaño del arreglo en caso de que se esté declarando uno.

Una vez generado el AST extendido, se procede a realizar el Análisis de Contexto a través de la definición una tabla de símbolos que lleva rastro de las variables declaradas en los distintos bloques del programa y de distintas funciones pertenecientes a la clase Context que verifican que se cumplan la correctitud entre los tipos de las expresiones y el alcance de las variables.

Las entradas en la tabla de símbolos pertenecen a la clase “Símbolo” que contiene el respectivo valor y tipo devueltos por el AST extendido.

La tabla de símbolos está implementada como una tabla de hash, en python se utilizó una lista (pila) de tablas de hash (diccionarios) para tener una tabla de símbolos por cada bloque donde se realicen declaraciones y así llevar registro del alcance de las variables y poder manejar los errores de contexto relacionados.

Las funciones definidas para el análisis de contexto son:

- contextCheck: Recorre el AST para buscar declaraciones de variables y agregar scopes (variables declaradas en cada bloque) a la pila de scopes. Si se encuentran asignaciones se chequea que el tipo del valor asignado concuerde con la declaración realizada. Si se encuentra un bloque, se llama nuevamente a la función y al terminar con el bloque se remueve el scope del tope de la pila.
- checkTipoArrOfArr, getTipold, getTipo: Se utilizan para verificar la validez de los arreglos declarados y utilizados en el programa.
- nuevoScope, agregarSimbolo: Agrega una entrada en la respectiva tabla de símbolos.
- checkExp: Chequea la validez de las expresiones booleanas, relacionales, aritméticas, de arreglos, de caracteres y terminales. Es decir, que las expresiones estén conformadas por términos del mismo tipo.
- checkVar: Chequea que una variable exista en los scopes.