Universidad LaSalle Arequipa

Facultad de Ingenieria

Escuela Profesional de Ingenieria de Software

PRACTICA 13

Curso:

Compiladores

Semestre:

VI

Integrantes:

Elmerson Ramith Portugal Carpio

Aron Josue Hurtado Cruz

Frank Jhoseph Duarte Oruro

2022

Ejercicios 5

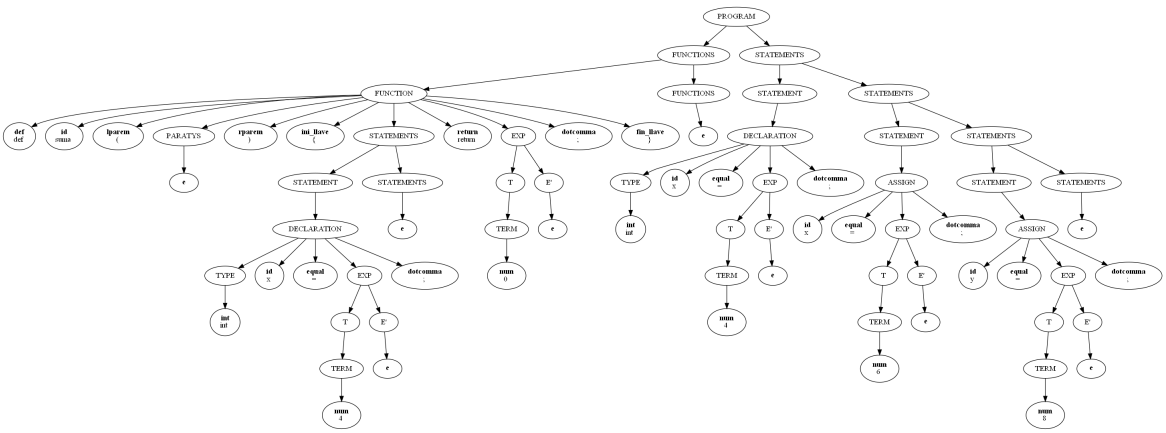
Implemente la verificacion e inferencia de tipos de datos del analizador semantico. El trabajo debe estar integrado con el analizador lexico y sint actico.

```
# Creacion de variables-------------------------------------------------
  if (root.symbol.symbol == 'DECLARATION'):
      variable = root.children[1]
      nodo_tipo = root.children[0]
      expresion = identificado(root)
      aux = root

      # Vamos a comprovar si esta o no es una funcion------------------
      while aux.symbol.symbol != 'FUNCTION':
        if aux.father == None:
          break
        aux = aux.father

      padre_asigando = "LIBRE"
      ## tomanos el nombre de la funcion perteneciente
      if aux.symbol.symbol == 'FUNCTION':
        padre_asigando = aux.children[1].lexeme

      if encontrar(variable.lexeme):
        print("VARIABLE YA CREADA -> ERROR EN LINEA ->", variable.line)
      else:

        if nodo_tipo.children[0].lexeme == "bool" and  expresion == "BOOLEAN" :
            print("VARIABLE CREADA -> EN LINEA ->", variable.line)
            tipo = "id"
            categoria = expresion
            padre = padre_asigando
            agregar(variable.lexeme, tipo, categoria, padre)

        elif nodo_tipo.children[0].lexeme == "int" and  expresion == "num":
            print("VARIABLE CREADA -> EN LINEA ->", variable.line)
            tipo = "id"
            categoria = expresion
            padre = padre_asigando
            agregar(variable.lexeme, tipo, categoria, padre)

        else:
          print("ERROR DE TIPOS -> EN LINEA ->", variable.line)
```

```
# Asigacion de variables-------------------------------------------------

  if root.symbol.symbol == 'ASSIGN' and root.father.symbol.symbol != 'DECLARATION' and
    root.father.symbol.symbol != 'FUNCTION':
    sub_valor = root.children[0]

    valor = identificado(root)
    flag = False
    for i in array:

      if i.lexema == sub_valor.lexeme and i.categoria == valor:
        print('VARIABLE EN USO -> EN LINEA ->', sub_valor.line )
        flag = True
      elif i.lexema == sub_valor.lexeme and i.categoria != valor:
        print('ERROR DE ASIGACION -> EN LINEA ->', sub_valor.line )
        flag = True

    if not flag:
        print('VARIABLE NO CREADA -> ERROR EN LINEA ->',  sub_valor.line )
      #return


## Eliminacion de los valores de las funciones ------------------------------------

  if root.symbol.symbol == 'fin_llave' and root.father.symbol.symbol == 'FUNCTION':

    count = 0
    for i in array:
      if i.funcion_padre == root.father.children[1].lexeme:
        count = count + 1

    while count > 0:
      for j in array:
        if j.funcion_padre == root.father.children[1].lexeme:
          array.remove(j)
      count = count - 1
```

Output

```
def suma (){
    int x = 4;
    return 0;
}

int x = 4;
x = 6;
y = 8;
Todo bien!
FUNCION CREADA CORRECTAMENTE
VARIABLE CREADA -> EN LINEA -> 2
VARIABLE CREADA -> EN LINEA -> 6
VARIABLE EN USO -> EN LINEA -> 7
VARIABLE NO CREADA -> ERROR EN LINEA -> 8
suma FUNCION None LIBRE
x id num LIBRE
```



Repositorio del trabajo

Aqui esta el repositorio Link