

Etapa III: Análisis de Contexto de PascalUSB

Para esta etapa del proyecto se necesita realizar un análisis de contexto sobre programas en PascalUSB. Se necesita tomar el Árbol Sintáctico Abstracto, implementado en la fase anterior, y aumentarlo, enriqueciéndolo con información de contexto (por ejemplo, variables y sus tipos). Así, también, se deben verificar errores estáticos (por ejemplo, tratar de sumar un entero con un booleano).

1. Requerimientos

Específicamente para esta fase se debe implementar la verificación de errores estáticos. La verificación de errores estáticos involucra:

1. Crear una tabla de símbolos en la cual, se van a almacenar las variables que fueron declaradas y su tipo
2. Reportar los errores estáticos en nuestro programa, estos errores serán los siguientes:
 - a) Utilizar variables sin haberlas declarado.
 - b) Declarar dos o más veces la misma variable (redeclaración de variables).
 - c) Asignar un valor a una variable de control para un `for`.

Por ejemplo:

```
Error in row x, column y: It is changing the variable "k", which is a control  
variable of a 'for' statement
```

- d) Errores de tipo (Como por ejemplo, sumar enteros y booleanos).

La tabla de símbolos debe ser implementada usando tablas de hash, es recomendable usar las tablas de hash que provee el lenguaje de su elección.

Para cubrir los temas de alcance, y repetición determinada, la tabla de símbolos tiene que ser jerárquica; por lo tanto, si un nombre no se encuentra en la tabla de símbolos local, debe buscarse en la tabla de símbolos inmediatamente superior. Si al recorrer todas las tablas de símbolos no se encuentra el nombre, entonces debe reportarse que el identificador no ha sido declarado. Las redeclaraciones de variables sólo ocurren en un mismo nivel (donde no se han incorporado nuevos alcances). Si un nombre es usado en diferentes niveles, el del nivel más interno esconde la definición del nivel más externo en la extensión de su alcance.

Como en la entrega pasada, si el programa analizado presenta errores léxicos debe mostrarlos todos. Si presenta un error sintáctico o un error de contexto debe imprimir sólomente el primero que encuentre (sintáctico o de contexto). Si el programa analizado no presenta errores, debe imprimir las tablas de símbolos y el árbol sintáctico abstracto decorado asociado al mismo.

2. Ejemplo

Para el siguiente programa escrito en pascalUSB

```
program
begin
  declare
    x, y, z, p as int, int, inter, int

  x := 1;
  y := 1;
  z := x..y;
  if x == y then
    x := itoi(z) + 1
  else
    begin
      declare
        z as int;
        q as inter
        z := itoi(x..y);
        println z
      end
    end
end
```

Usted debe imprimir el siguiente árbol sintáctico abstracto decorado con el siguiente formato

```
Block
  Symbols Table
    variable: x | type: int
    variable: y | type: int
    variable: z | type: inter
    variable: p | type: int

  Asig
    Ident: x
  Exp
    Literal: 1
  Sequencing
    Asig
      Ident: y
    Exp
      Literal: 1
  Sequencing
    Asig
      Ident: z
    Exp
      SoForth
        Ident: x
        Ident: y
  Sequencing
    If
      Exp
        Equal
          Ident: x
          Ident: y
```

```

Then
  Asig
    Ident: x
  Exp
    IntPlus
      Itoi
        Ident: z
      Literal: 1
Else
  Block
    Symbols Table
      variable: z | type: int
      variable: q | type: inter

  Asig
    Ident: z
  Exp
    Itoi
      SoForth
        Ident: x
        Ident: y
  Sequencing
    Println
      Ident: z

```

3. Entrega

La entrega del proyecto es el lunes de la semana 10 (26 de Noviembre) por Aula Virtual. Su entrega debe incluir lo siguiente:

- Un archivo comprimido **tar.gz** con el código fuente de su proyecto, debidamente documentado, colocado en el Aula Virtual. El nombre del archivo deber ser **Etap3-XX-YY.tar.gz** donde **XX-YY** son los carné de los integrantes del grupo.

El no cumplimiento de los requerimientos podría resultar en el rechazo de su entrega.