

Etapa III: Análisis de Contexto de GCL

Para esta etapa del proyecto se necesita realizar un análisis de contexto sobre programas en su lenguaje imperativo. 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 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) Errores de tipo (como por ejemplo, sumar enteros y booleanos o asignar una expresión de un tipo a una variable declarada de otro tipo).

Es recomendable usar diccionarios para implementar las tablas de símbolo. Aunque no se prohíbe el uso de cualquier otra estructura de datos.

Para cubrir los temas de bloques anidados, 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. Los errores de 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

```
{  
  int x, y, p;  
  function[..1] z;
```

```

x := 1;
y := 3;
z := x, y;
if x == y -->
  x := z.y + 1
[] x <> y -->
{
  int z;
  function[..2] q;
  z := y-x+1;
  print z
}
fi
}

```

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: p | type: int
--variable: z | type: function[..1]
-Sequencing
--Sequencing
---Sequencing
----Asig
-----Ident: x | type: int
-----Literal: 1 | type: int
----Asig
-----Ident: y | type: int
-----Literal: 3 | type: int
---Asig
----Ident: z | type: function[..1]
----Comma | type: function with length=2
-----Ident: x | type: int
-----Ident: y | type: int
--If
---Guard
----Then
-----Equal | type: bool
-----Ident: x | type: int
-----Ident: y | type: int
-----Asig
-----Ident: x | type: int
-----Plus | type: int
-----App | type: int
-----Ident: z | type: function[..1]
-----Ident: y | type: int
-----Literal: 1 | type: int
----Then
-----NotEqual | type: bool

```

```

-----Ident: x | type: int
-----Ident: y | type: int
-----Block
-----Symbols Table
-----variable: z | type: int
-----variable: q | type: function[..2]
-----Sequencing
-----Asig
-----Ident: z | type: int
-----Plus | type: int
-----Minus | type: int
-----Ident: y | type: int
-----Ident: x | type: int
-----Literal: 1 | type: int
-----Print
-----Ident: z | type: int

```

3. Detalles de la entrega

La entrega del proyecto es el miércoles de la semana 10. Su entrega debe incluir lo siguiente:

- Un archivo comprimido **tar.gz** con el código fuente de su proyecto, debidamente documentado. El nombre del archivo debe ser **Etapal-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.