

Escuela Politécnica Nacional
Compiladores y Lenguajes 2016-A
Definición de Proyecto – Segunda Etapa: Análisis Sintáctico

Este trabajo consiste en el diseño e implementación de un compilador funcional para un lenguaje de programación basado en el Lenguaje C, cuyas especificaciones léxicas ya fueron realizadas en la etapa anterior. En la segunda etapa del trabajo es necesario realizar un analizador sintáctico y llenar la tabla de símbolos.

1. Funcionalidades necesarias

- a. **Análisis Sintáctico.** El programa principal debe llamar al analizador sintáctico para el reconocimiento de líneas de código que formen parte del lenguaje.
- b. **Informe de errores sintácticos.** En caso que la entrada no sea reconocida, debe imprimirse un mensaje informando el número de línea donde se encontró el error sintáctico.
- c. **Enriquecimiento de la tabla de símbolos.** Las entradas almacenadas en la tabla de símbolos deben ser organizadas entre los tipos de símbolos almacenados (identificadores, literales, etc.)

2. Descripción del Lenguaje a analizar

Un programa en el Lenguaje a Analizar está compuesto por un conjunto opcional de declaraciones de variables globales y un conjunto opcional de funciones, que pueden aparecer intercaladamente y en cualquier orden. Todas las declaraciones globales son terminadas por punto-y-coma. Cada función es descrita por un encabezado y un cuerpo. Los comandos pueden ser de atribución, control de flujo o los comandos **input**, **output** y **return**.

a. Declaración de variables globales

Las variables son declaradas por su tipo, dos-puntos y son terminadas por su nombre. El lenguaje incluye también la declaración de vectores, hecha por la definición de su tamaño entero positivo entre corchetes, colocada a la derecha de su nombre. Todas las variables globales serán inicializadas con el valor de cero. Las variables pueden ser de tipos: **int**, **float**, **char**, **bool** y **string**.

b. Definición de Funciones

Cada función es definida por un encabezado, una lista de declaraciones locales y un cuerpo.

El encabezado consiste en el tipo de valor de retorno, seguido por dos-puntos y terminado por una lista. Esta lista, que puede también ser vacía, indica los parámetros de entrada de la función, separados por comas y donde cada parámetro es definido por su tipo, dos-puntos y su nombre.

La lista de declaraciones locales es un grupo de declaraciones de variables en el mismo formato de las declaraciones globales, donde cada una consiste en el tipo de la variable, dos-puntos, el nombre de la variable y el terminador punto-y-coma. Las declaraciones locales, al contrario de las globales, no permite vectores.

El cuerpo de la función está compuesto por un bloque, como se define a continuación. La función no debe ser terminada con punto-y-coma.

c. Bloque de comandos

Un bloque de comandos es definido entre llaves, y consiste en una secuencia, que puede ser vacía, de comandos simples, separados por punto-y-coma. Un bloque de comandos es considerado como un comando único simple, recursivamente, y puede ser utilizado en cualquier construcción que acepte un comando simple.

d. Comandos Simples

Los comandos simples del lenguaje pueden ser: atribución, construcciones de control de flujo, operaciones de entrada, de salida y de retorno, un bloque de comandos,

llamadas de función, y el comando vacío. En la atribución se usa una de las siguientes formas:

```
variable = expresión  
vector[expresión] = expresión
```

Los tipos correctos para los valores de las expresiones y los índices son solamente verificados en el análisis semántico.

e. Expresiones aritméticas y lógicas

Las expresiones aritméticas pueden ser formadas recursivamente con operadores aritméticos, y también permiten el uso de paréntesis para indicar asociación.

Las expresiones lógicas pueden ser formadas a través de los operadores relacionales aplicados a expresiones aritméticas, o de operadores lógicos aplicados a expresiones lógicas, recursivamente.

La descripción sintáctica debe aceptar cualquier operador y subexpresión como válida, dejando para el análisis semántico la tarea de verificar la validez de los operandos y operadores.

f. Comandos de Control de Flujo

Para el control de flujo, el lenguaje posee una única construcción condicional con el else opcional y dos construcciones de repetición, descritas así:

```
if (expresión) then comando  
if (expresión) then comando else comando  
while (expresión) do comando  
do comando while (expresión)
```

3. Recomendaciones

Se recomienda que se ejecuten varios tests. Asimismo que se verifique la conformidad con cada una de las reglas de este documento y de la especificación de la etapa.

Verificar si el trabajo puede ser compilado y ejecutado en otro sistema aparte del utilizado para el desarrollo.

4. Evaluación de las Etapas y del Proyecto

Cada etapa del proyecto será evaluada. Asimismo se evaluará la participación de los miembros del grupo por su conocimiento y capacidad para explicar el funcionamiento del proyecto.

5. Plazos de Entrega

La presentación del proyecto se realizará el día jueves 30 de junio de 2016.