I.T.C.R. Compiladores & Intérpretes

Departamento de Computación — Scanner
Prof: Ing. Erika Marín Schumann. — II Semestre del 2014

Motivación

Muchas de las aplicaciones que se desarrollan actualmente requieren validar los datos de entrada, ya sea de interfaz o de archivos. Para esto, se necesita los mismos conocimientos básicos que son utilizados en la construcción de la etapa de análisis léxico de un compilador, de ahí que sea de suma importancia tener experiencia al respecto.

Objetivos del proyecto

- Aprender a desarrollar un scanner para un lenguaje de programación
- Utilizar herramientas ya diseñadas para facilitar el diseño del scanner (JFLEX)
- Diseñar la primera etapa del compilador del curso.

Definición general

Esta primera etapa del proyecto conocida formalmente como Scanner o Análisis Léxico es de vital importancia que sea desarrollada con cuidado y visión hacía el futuro, ya que el resto del proyecto se basará en una buena definición del lenguaje a compilar.

Para esta primera etapa del proyecto se debe entregar un programa que reciba un código fuente escrito en un lenguaje de alto nivel parecido a Pascal y realice el análisis léxico respectivo. Para esto se debe definir el lenguaje aceptado utilizando la herramienta JFLex. Por lo tanto, la tarea debe ser escrita en Java.

Al finalizar el scaneo el programa deberá desplegarle al usuario el resultado del Análisis Léxico que efectuó. Se esperan dos aspectos del resultado.

1. **Listado de errores léxicos encontrados:** El programa debe desplegar una lista de todos los errores léxicos que se encontraron en el código fuente. Debe desplegar la línea en la que se encontró el error. Es importante que el programa debe poder recuperarse del error y no desplegar los errores en cascada ni terminar de hacer el scaneo al encontrar el primer error.
2. **Listado de los tokens encontrados:** Durante la ejecución del análisis léxico se debe llevar el control de todos los tokens o palabras aceptadas que se encuentren en el código fuente. Al finalizar el análisis, el programa debe ser capaz de desplegar una lista con cada uno de estos tokens, el tipo de token que son y las líneas del código fuente donde se presentan y la cantidad de ocurrencias del token en cada línea. Los tokens que presentaron errores no deben ser listados. Se espera que esta lista sea lo más ordenada posible y que sea fácil de leer.

Ej:

Token	Tipo de Token	Línea
Hola	IDENTIFICADOR	5, 7, 50(2)
+	OPERADOR	6,8,15(3),36,45

Descripción Detallada

Se les sugiere tomar en cuenta los siguientes aspectos para asegurar la completitud del programa.

- El detalle de los tipos de Token será asignado por ustedes mismos. El tipo de Token no se refiere a los tokens que debe aceptar el programa sino que tipo de cada uno de estos puede diferir en cada tarea. Por ejemplo el token “+” puede ser de tipo “OPERADOR” o “OPERADOR ADITIVO”. Sin embargo, deben haber al menos 4 grandes grupos de Tokens:
    - IDENTIFICADORES
    - OPERADORES
    - PALABRAS RESERVADAS
    - LITERALES
    - COMENTARIOS

- El programa debe identificar los comentarios y omitir todos los tokens que se encuentran dentro de ellos. Se cuenta con dos tipos de comentarios. Los comentarios de línea inician con la sequencia '//' y los comentarios de bloque pueden ser de dos formas. Unas sería que comiencen con (* y terminan con *) y la otra son comentarios entre corchetes { }. Los comentarios no deben venir en el listado de tokens que presentará el sistema.

- Los identificadores son palabras que representan constantes, variables, tipos de datos, procedimientos, funciones y algunos otros datos. Un identificador es una secuencia de 1 a 127 caracteres, que inicia con una letra, no tienen espacios ni símbolos: &, !, *, etc. y no es alguna palabra reservada. No existen diferencias entre mayúsculas y minúsculas, así que a un identificador denominado "valor" se le puede referir como "VALOR" o "VaLoR".

- Las palabras reservadas a aceptar son las siguientes:

AND ARRAY BEGIN BOOLEAN BYTE CASE CHAR CONST DIV DO DOWNTO ELSE END FALSE FILE FOR FORWARD FUNCTION GOTO IF IN INLINE INT LABEL LONGINT MOD NIL NOT OF OR PACKED PROCEDURE PROGRAM READ REAL RECORD REPEAT SET SHORTINT STRING THEN TO TRUE TYPE UNTIL VAR WHILE WITH WRITE XOR

- Los literales deben permitir número enteros, números flotantes, caracteres y strings.

    - Los números reales deben llevar por fuerza al menos un dígito de cada lado del punto decimal así sea éste un cero. Como ejemplo, el número 5 debe representarse como: 5.0, el .5 como 0.5 , etc. También se puede utilizar la la notación científica, Ejemplo: 3.0E5 o 1.5E-4
    - Los Strings y los caracteres se representean entre ' ' (comillas simples). Los strings pueden ser de varias líneas.) Además los caracteres también se pueden representar con el signo de # seguido por un número entero. Ej: #65

- A continuación se detalla una lista con los operadores válidos en el lenguaje.

```
","   ";"   "++"  "--"  ">="  ">"   "<="  "<"   "<>"  "="   "+"   "-"   "*" "/"
"("   ")"   "["   "]"   ":="  "."   ":"   "+="  "-="  "*="  "/="  ">>" "<<" "<<="
">>="

Además las palabras reservadas NOT OR AND XOR DIV MOD corresponden a operadores.
```

- Recuerden que esta es la primera etapa del proyecto del curso, por eso entre más detallado y funcional esté, más facilidad van a tener en el resto de etapas del proyecto

Documentación

Se espera que sea un documento donde especifique el análisis de resultados del programa junto con unos casos de pruebas. El Análisis debe resumir el resultado de la programación, qué sirve, qué no sirve y aspectos que consideren relevantes. Para las pruebas se espera que definan claramente cada prueba, cuáles son los resultados esperados y cuáles fueron los resultados obtenidos. No es necesario que sean grandes pero deben evaluar la funcionalidad completa del programa. Deben especificar, además, como compilar y ejecutar su scanner.

Aspectos Administrativos

- El desarrollo de este programa debe de realizarse en grupos de exactamente dos personas salvo acuerdo con el profesor. Los grupos de trabajo deben permanecer iguales para los siguientes proyectos del curso.
- El trabajo se debe de entregar el día 5 de Octubre de 2014 antes de media noche, enviarlo por correo.
- Deben entregar el código fuente junto con el ejecutable.
- Se debe de entregar el documento IMPRESO para ser calificado, así como una copia del archivo de la documentación.
- Referencia: http://www.jflex.de/index.html