Guía Práctica 1 - MII 779
Análisis Léxico en ANTLR
Desarrollo de Lenguajes Orientados a Objeto y Compiladores Avanzados

Profesor: Ricardo Soto

Ejercicio 1: Instalar plug-in ANTLR para Eclipse

- Compruebe si el plug-in está instalado, verificando en la carpeta `eclipse/plugins` si las carpetas `org.antlr.doc, org.antlr.core, org.antlr.ui y org.antlr` existen:
  - Si las carpetas no existen, acceda a `Help->Install New Software->Add` y agregue el siguiente sitio `http://antlreclipse.sourceforge.net/updates`
  - Seleccione el plug-in ANTLR e instale.

Ejercicio 2: Compruebe la instalación del plug-in

- Descargue el archivo `milePP-lexic.zip` desde `http://www.inf.ucv.cl/~rsoto/cursos/MII779/milePP-lexic.zip`
- Descomprima y copie en su workspace.
- Cree un nuevo proyecto Java llamado "MilePP" seleccionando como fuente del proyecto (Create project from existing source) la carpeta recientemente descomprimida.
- Dentro del proyecto acceda a `src->cl.ucv.inf.mileppcompiler->compilers` y compile el archivo `MilePPLexer.g` (posicionado sobre el archivo, botón derecho, Compile ANTLR Grammar)
- Se creará el archivo `MilePPLexer.java` (junto a 3 archivos más), el cual corresponde al lexer escrito en Java generado automáticamente a partir de la gramática milePPLexer.g.

Ejercicio 3: Implemente el analizador léxico para el lenguaje Mile++

- Complete la gramática `MilePPLexer.g` con los tokens y reglas faltantes.
- Compile la gramática.

Ejercicio 4: Comprenda la clase `Tool`

- Diríjase a `src->cl.ucv.inf.mileppcompiler->Tool`. Esta clase permitirá comprobar su analizador léxico.
- La clase Tool.java está compuesta de 4 métodos:
  - `main` se encarga de recibir el archivo a escánear.
  - `setSourceFile` almacena el nombre del archivo y genera un objeto `FileInputStream` a partir del archivo fuente para que el lexer pueda analizarlo.
  - `lexicTest` llama a nuestro lexer (`MilePPLexer`) y recorre los tokens reconocidos por el lexer utilizando `nextToken`. Por cada token se imprime el nombre del archivo escaneado, la línea y columna del token y el token.

Ejercicio 5: Pruebe el analizador léxico

- Ejecute la clase `Tool` utilizando como programa fuente el archivo `examples/test1.mile`. La ejecución debería entregar la lista de tokens encontrados en el archivo, incluyendo su número de línea y columna.
- Cree y ejecute un nuevo archivo con tokens NO pertenecientes al lenguaje Mile++.

Ejercicio 6: Extienda el analizador léxico

- Agregue una regla para comentarios. ¿Qué se debe incluir en la regla para que el analizador no los muestre en pantalla al ejecutar la clase `Tool`?
- Agregue una regla para strings.
- Modifique la clase `Tool` para mostrar en pantalla el tipo de token (`tok.getType()`). ¿Cómo se identifican los tokens en ANTLR?

Ejercicio 7: Implemente y pruebe el analizador léxico de su lenguaje.
