Teoría de la computación I

TP Integrador

Integrantes:
Nicolas Cavasin - Legajo #143501
Sofía Vázquez - Legajo #138224

Objetivo:

- Realizar un analizador lexicográfico utilizando la herramienta JFLEX.
- Realizar un analizador sintáctico utilizando la herramienta Java CUP.
- La aplicación realizada debe mostrar una interfaz gráfica que pueda utilizarse como IDE del compilador, el cual en este caso deberá mostrar por pantalla un texto aclaratorio identificando las reglas sintácticas que va analizando el parser, en base a un archivo de entrada (prueba.txt).
- Las impresiones deben ser claras.
- Las reglas que no realizan ninguna acción no deben generar salida.

Función take:

- Recibe un operador numérico básico (suma, resta, división o multiplicación), una constante entera (*n*) y una lista de constantes enteras.
- Devuelve el valor que resulta de aplicar el operador numérico a los primeros *n* elementos de la lista de enteros.
- Si recibe una lista vacía, devuelve 0 (cero).
- Signature: TAKE (OP_NUM; CTE_INT; [lista CTE_INTs])

```
- var := TAKE (+ ; 2 ; [2 ; 12 ; 24 ; 48] )    ==> 14.
- var := TAKE (/ ; 5 ; [])                     ==> 0.
```

Gramática libre de contexto

```
<programa> ::= <declaraciones> BEGIN_PROGRAM <lista_sentencias> END_PROGRAM
              | BEGIN_PROGRAM <funcion_print> END_PROGRAM

<declaraciones> ::= DECLARE <lista_declaraciones> ENDDECLARE

<lista_declaraciones> ::= <lista_declaraciones> <linea_declaracion>
                        | <linea_declaracion>

<linea_declaracion> ::= COR_ABRE <declaracion> COR_CIERRA

<declaracion> ::=  ID COMA <declaracion> COMA <tipo>
                 | ID COR_CIERRA ASIGNA COR_ABRE <tipo>

<tipo> ::= INT | FLOAT | STRING
```

Gramática libre de contexto

```
<lista_sentencias> ::= <lista_sentencias> <sentencia> | <sentencia>

<sentencia> ::= <asignacion> | <funcion_print> | <funcion_if>
              | <funcion_while>

<asignacion> ::= ID ASIGNA <expresion>

<expresion> ::= <expresion> SUMA <termino> | <expresion> RESTA <termino>
              | <termino>

<termino> ::= <termino> MULTIPLICA <factor> | <termino> DIVIDE <factor>
            | <factor>

<factor> ::= ID | CTE_INT | CTE_FLOAT | CTE_STRING
           | PAR_ABRE <expresion> PAR_CIERRA | <funcion_take>
```

Gramática libre de contexto

```
<funcion_take> ::= TAKE PAR_ABRE <op_num> PUNTO_COMA CTE_INT PUNTO_COMA
                   COR_ABRE <lista_take> COR_CIERRA PAR_CIERRA
                 | TAKE PAR_ABRE <op_num> PUNTO_COMA CTE_INT PUNTO_COMA
                   COR_ABRE COR_CIERRA PAR_CIERRA

<op_sum> ::= SUMA | RESTA | DIVIDE | MULTIPLICA

<lista_take> ::= <lista_take> PUNTO_COMA CTE_INT | CTE_INT

<funcion_print> ::= PRINT CTE_STRING

<funcion_if> ::= IF PAR_ABRE <condicion> PAR_CIERRA <bloque>
               | IF PAR_ABRE <condicion> PAR_CIERRA <bloque> ELSE <bloque>

<funcion_while> ::= WHILE PAR_ABRE <condicion> PAR_CIERRA <bloque>
```

Gramática libre de contexto

```
<condicion> ::= PAR_ABRE <condicion> PAR_CIERRA <op_bin> PAR_ABRE <condicion>
                PAR_CIERRA
          | PAR_ABRE <condicion> PAR_CIERRA <op_logico> PAR_ABRE <condicion>
            PAR_CIERRA
          | PAR_ABRE <condicion> PAR_CIERRA <op_logico> <expresion>
          | <expresion> <op_logico> PAR_ABRE <condicion> PAR_CIERRA
          | <expresion> <op_logico> <expresion>

<bloque> ::= LLAVE_ABRE <lista_sentencias< LLAVE_CIERRA

<op_logico> ::= IGUAL | DISTINTO | MENOR | MAYOR | MAYOR_IGUAL

<op_bin> ::= AND | OR
```

Link al proyecto:

https://github.com/ncavasin/teo1/blob/main/tp07-integrador/entrega_2/compilador.jar