



JFLEX y CUP

OLC1-A
Clase 3

Temas

01

Generadores de
Analizadores

02

JFlex

03

JCup

04

Ejemplo 1

05

Ejemplo 2



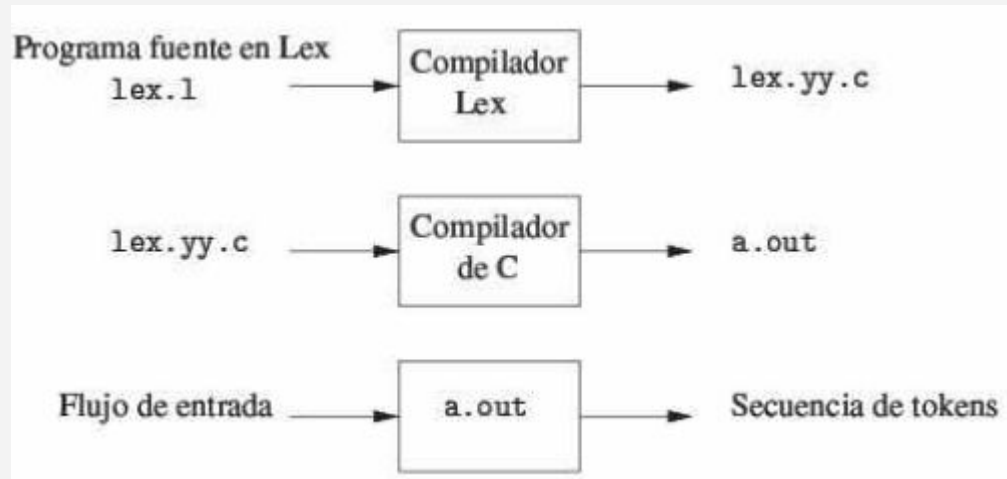
01

Generadores de
Analizadores



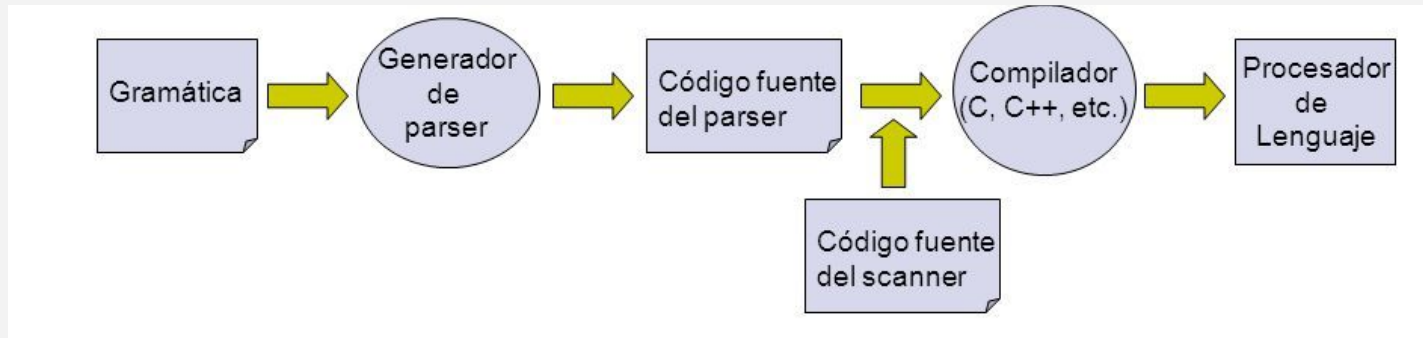
Generadores de Analizadores Léxicos

- Toman como entrada una especificación (expresiones regulares)
- Devuelven una implementación en código de un autómata finito que reconoce lenguajes regulares definidas.



Generadores de Analizadores Sintácticos

- Toman como entrada una especificación de la gramática
- Devuelven una implementación en código del parser



([a-z][a-

02

JFlex

- JLex es un generador de analizadores léxicos escrito en Java
- JLex se utiliza en Java
- Produce una secuencia de tokens
- Maneja errores léxicos

20-9/-1*10

Sintaxis

//package e importaciones

%%

//configuraciones para el análisis

%%

//reglas lexicas


```

package analizadores;
import java_cup.runtime.Symbol;

%%
%class Lexico
%public
%line
%char
%cup
%unicode
%ignorecase

%init{
    yyline = 1;
    yychar = 1;
%init}

BLANCOS=[ \r\t]+
D=[0-9]+
DD=[0-9]+("[" |0-9]+)?

%%

"Evaluar" {return new Symbol(sym.REVALUAR,yyline,yychar,
    yytext());}

";" {return new Symbol(sym.PTCOMA,yyline,yychar, yytext());}
"(" {return new Symbol(sym.PARIZQ,yyline,yychar, yytext());}
")" {return new Symbol(sym.PARDER,yyline,yychar, yytext());}
"[" {return new Symbol(sym.CORIZQ,yyline,yychar, yytext());}
"]" {return new Symbol(sym.CORDER,yyline,yychar, yytext());}

"+" {return new Symbol(sym.MAS,yyline,yychar, yytext());}
"-" {return new Symbol(sym.MENOS,yyline,yychar, yytext());}
"*" {return new Symbol(sym.POR,yyline,yychar, yytext());}
"/" {return new Symbol(sym.DIVIDIDO,yyline,yychar, yytext());}

\n {yychar=1;}

{BLANCOS} {}
{D} {return new Symbol(sym.ENTERO,yyline,yychar, yytext());}
{DD} {return new Symbol(sym.DECIMAL,yyline,yychar, yytext());}

. {
    System.out.println("Este es un error lexico: "+yytext()+
        ", en la línea: "+yyline+", en la columna: "+yychar);
}

```

03

Cup



- Cup es un generador de analizadores sintácticos
- LALR
- De gran utilizar al utilizarse con JFlex
- Maneja errores sintácticos

Sintaxis

//package e importaciones

parser code

{:

:}

//terminales

```
terminal String ENTERO;
```

//no terminales

```
non terminal instrucciones;
```

//precedencias

```
precedence left MAS,MENOS;
```

//producción de inicio

```
start with ini;
```

//producciones

```
ini::=instrucciones;
```

```

package analizadores;
import java_cup.runtime.*;

parser code
{
    /**
     * Método al que se llama automáticamente ante algún error sintáctico.
     */
    public void syntax_error(Symbol s){
        System.out.println("Error Sintáctico en la Línea " + (s.left) +
            " Columna "+s.right+ ". No se esperaba este componente: " +s.value+ ".");
    }

    /**
     * Método al que se llama automáticamente ante algún error sintáctico
     * en el que ya no es posible una recuperación de errores.
     */
    public void unrecovered_syntax_error(Symbol s) throws java.lang.Exception{
        System.out.println("Error sintáctico irrecuperable en la Línea " +
            (s.left)+ " Columna "+s.right+. Componente " + s.value +
            " no reconocido.");
    }
}

:}

terminal String PTCOMA,PARIZQ,PARDER,CORIZQ,CORDER;
terminal String MAS,MENOS,POR,DIVIDIDO;
terminal String ENTERO;
terminal String DECIMAL;
terminal String UMENOS;
terminal String REVALUAR;

non terminal ini;
non terminal instrucciones;
non terminal instruccion;
non terminal Double expresion;

precedence left MAS,MENOS;
precedence left POR,DIVIDIDO;
precedence right UMENOS;

start with ini;

ini::=instrucciones;

instrucciones ::=
    instruccion instrucciones
    | instruccion
    | error instrucciones

```

Muchas gracias por su atención!

Dudas?

<https://jflex.de/download.html>

<http://www2.cs.tum.edu/projects/cup/>

<https://www.cs.princeton.edu/~appel/modern/java/JLex/>