Tarea 2 Compiladores I

1. En una cadena de longitud n, ¿cuántas subcadenas y subsecuencias hay? Justifique su

respuesta.

R// Ya que simboliza todas las subcadenas formadas por cada caractér, adicionando todas las combinaciones que existen entre los mismos.

2. Encontrar expresiones regulares que representen los siguientes lenguajes:

* Todas las cadenas sobre {a, b, c} que no contienen la subcadena baa
* Todas las cadenas sobre {a, b} que no contienen la subsecuencia baa
* El lenguaje sobre {0, 1} en que las palabras contienen un número de ceros diferente de 3, por ejemplo: “010”, “1111”, “00100”, etc.

1. (a|c)\*b\*(b|c|ba|ab)\*
2. (a)\*b(ba|ab|b)\*
3. ((01\*01\*|1\*01\*0)|1\*)\*|1\*01\*

3. Descríbanse correctamente los lenguajes representados por las siguientes expresiones regulares:

a) (aa|ab|ba|bb)\*(a|b)

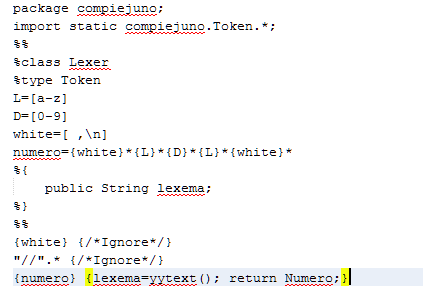
b) b\*(ab\*a)\*b\*

c) (00|11)\*((01|10)(00|11)\*(01|10)(00|11)\*)\*

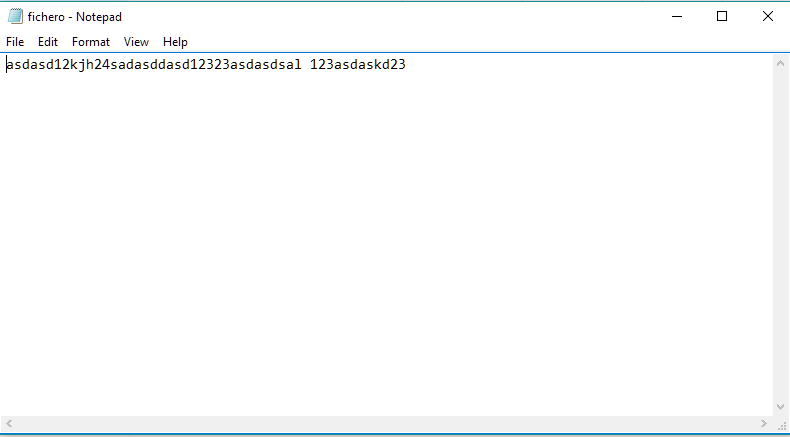
1. Los lenguajes sobre {a,b} tal que ab sea impar.
2. Las cadenas que tienen numero par de a’s.
3. Las cadenas binarias cuyo tamaño es múltiplo de 2.

4. Hacer un programa en JFlex que lea un archivo y determine cuántos números hay en él.

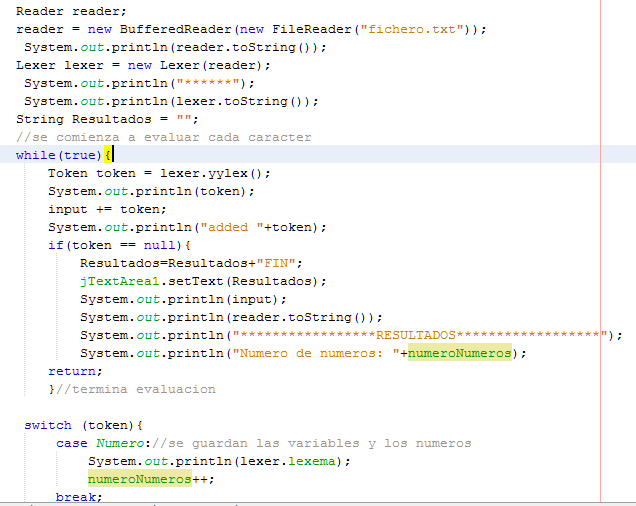
Flex File:



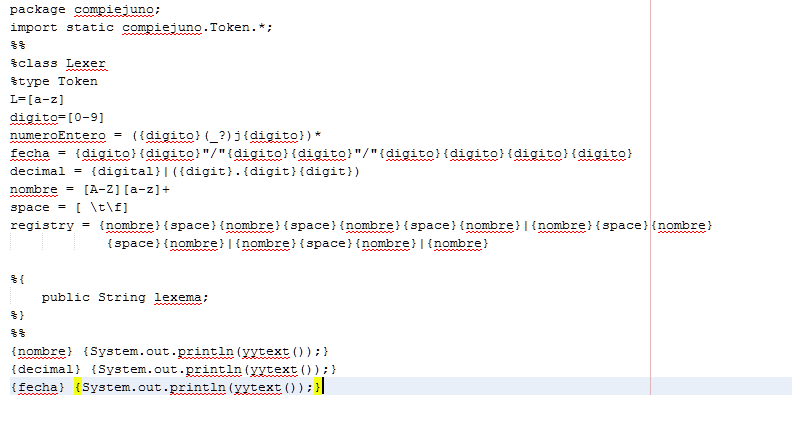
Archivo de Entrada:



Main:



5. Se tiene un archivo con datos de personas que se desea interpretar correctamente. La información de cada persona puede constar de: nombre, salario y fecha de nacimiento. Los campos no se encuentran separados por espacios ni por ningún otro carácter, y los registros se encuentran uno seguido del otro secuencialmente. Cada registro puede contener o no el salario y la fecha de nacimiento por lo que se necesita crear un analizador léxico que lea el archivo e imprima un registro por línea indicando el nombre, salario y fecha de nacimiento.



6. Considere la siguiente gramática:

S -> S S + | S S \* | a

Y la cadena aa+a\*

a) Proporcione una derivación por la izquierda para la cadena

b) Proporcione una derivación por la derecha para la cadena

c) Proporcione un árbol sintáctico para la cadena

d) ¿La gramática es ambigua o no? Justifique su respuesta

e) Describa el lenguaje generado por esta gramática y compruébelo.

1. aa+a\*

S => SS\*

S => SS+a\*

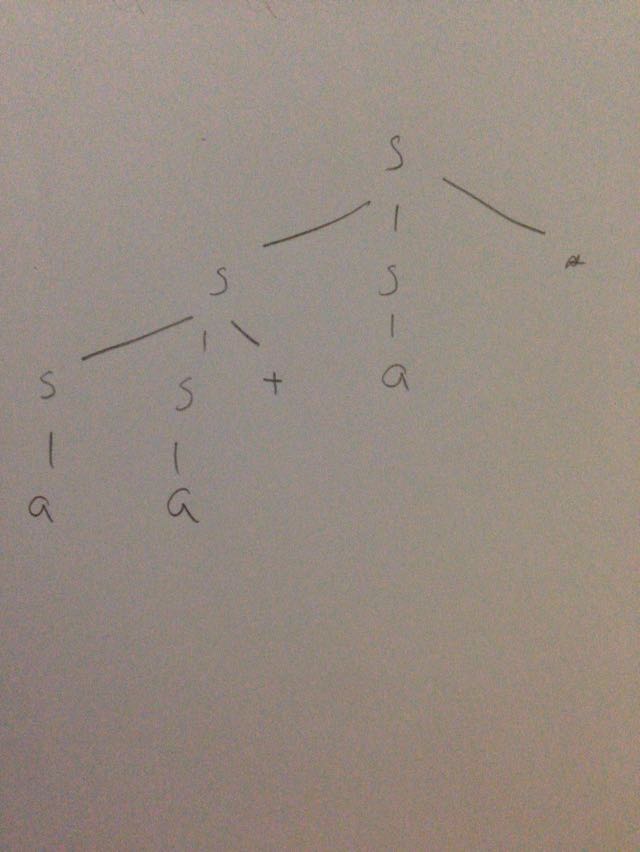
S => aa+a\*

1. aa+a\*

S => SS\*

S => SS+a\*

S => aa+a\*

1. 
2. No es ambigua porque no existen dos derivaciones por la izquierda o derecha que puedan ser validas.
3. Notación posfix de suma y multiplicación

7. Escriba una gramática no ambigua para los siguientes lenguajes:

i. El conjunto de todas las cadenas de 0 y 1 tales que todo 0 va seguido inmediatamente de al menos un 1

ii. Las cadenas de símbolos a y b en las que aba no es una subcadena.

iii. Instrucciones if para el lenguaje Ada de la forma:

If E1 then

S1

Elsif E2 then

S2

…

Elsif En then

Sn

Else

Sn+1

Endif

1. S → 01S | 1S | 3
2. R → XRX | S

S → a T b | b T a

T → XTX | X | ε

X → a | b

1. If\_statement → “if” condition “then” sequence\_of\_statements { “elseif” condition “then” sequence\_of\_statements }\* [ “else” sequence\_of\_statements ] “end” “if” “;”

Condition → bool\_expression

Sequence\_of\_statement → statement {statement}

Statement → ( {label}\* simple\_statement | {label}\* compound\_statement)

Bool\_expression →