1 -

Dado:

a := FACT (4) // FACT acepta 0

a) Escribir gramatica para la sintaxis de esa sentencia

b) Escribir las funciones en polaca.

(RTA: Resolucion en IMG\_3554)

2 -

Escribir BNF tal que respete la gramatica siguiente: Cada sentencia puede tener un solo nodo o una sola lista. Un nodo es un SYM o un NUM (tomarlos como terminales). Una lista puede contener cualquier cantidad de nodos o listas adentro. Las listas se escriben entre parentesis y pueden ser vacias.

Ejemplos de sintaxis validas:

NUM

( NUM SYM )

( SYM ( ) NUM ( NUM SYM ) )

( ( NUM ) ( SYM NUM ( ) ) )

(

RTA:

PROG -> LINEA | PROG LINEA

LINEA -> NODO | LISTA

LISTA -> (ELEM) | ()

ELEM -> NODO | LISTA | ELEM NODO | ELEM LISTA

NODO -> NUM | SYM

)

3 - (Nota: El enunciado del parcial estaba un poco mas detallado y se entendia mejor)

Un compilador maneja constantes string. El automata lexico (o sea, el flex) llega a su estado final y al devolver el token STRING arroja: "CHAR invalido en la string".

?Es esto posible?

(RTA: No es posible, Resolucion en IMG\_3554

4 -

Hacer parsing ascendente de esta gramatica e indicar si hay algun conflicto

E -> E : E

E -> E + E

E -> id

E -> ( E )

(RTA: Gramatica es ambigua, Resolucion en IMG\_3554.)

5-

Dado:

integer a

ALIAS x%a

ALIAS b%c // Esto da error, porque c no esta definido

Decir que etapas se ven afectadas durante la compilacion y describir que pasa en cada una

a) AL

b) AS

c) TS

d) GCI

e) GCASM

(RTA: Resolucion en IMG\_3555)