

EJERCICIOS PREVIOS TEMA 3

1. Modifica las reglas de producción de la siguiente gramática para que genere números naturales sin ceros no-significativos:

GRAMÁTICA NÚMEROS NATURALES CON CEROS NO-SIGNIFICATIVOS

```

G = ({0,1,2,3,4,5,6,7,8,9}, {N,C}, N,P)
con P = {
N ::= NC | C,
C ::= 0|1|2|3|4|5|6|7|8|9
}

```

Solución:

Primero de todo, hay que aclarar que los ceros no-significativos son aquellos que van antes de la primera cifra, por ejemplo, 074, el primero 0 es no significativo. Por ello, modificamos la regla:

$$P = \{ N ::= C \mid CN \mid 0, \\ C ::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \}$$

Así, todos los números que se pueden generar son los que están en el 0 y el 9 si tienen una cifra, y tras la segunda cifra solo podrán ir 0 de manera intercalada o al final del número.

2. Dada la siguiente gramática:

V_T = ({, }, :, =, a, ..., z, 0, ..., 9, +, -, *, /) /*alfabeto de nuestro lenguaje*/

V_N = (<programa>, <lista_sentencias>, <sentencia>, <identificador>, <letra>, <dígito>, <expresión>, <operador>)

S = <programa>

con reglas de producción:

```

<programa> ::= { <lista_sentencias> }
<lista_sentencias> ::= <sentencia> | <sentencia> ;
<lista_sentencias>
<sentencia> ::= <identificador> = <expresión>
<identificador> ::= <letra> | <identificador> <dígito> |
<identificador> <letra>
<letra> ::= a | b | ... | z
<dígito> ::= 0 | 1 | ... | 9
<expresión> ::= <identificador> <operador> <identificador> |
<identificador>
<operador> ::= + | - | * | /

```

a) ¿Cuáles serían los tokens y los patrones que los describen?

- identificador → Letra + secuencia de letras y dígitos.
- dígito → 0, 1, 2, 3,..., 9
- letra → El abecedario (a,b,c,...,z)
- expresión → <identificador><operador><identificador>
- sentencia → <identificador>=<expresión>
- operador → +, -, *, /
- punto y coma → ;
- corchete izquierdo → {

- corchete derecho → }

b) Para la sentencia { suma = b * z }, ¿qué secuencia de tokens se generaría? ¿qué se almacenaría en la tabla de símbolos?

<sentencia> → <identificador>=<expresión> → <identificador><letra>=<expresión>
 → <identificador><letra><letra>=<expresión> →
 <identificador><letra><letra><letra>=<expresión> →
 <letra><letra><letra><letra>=<expresión> →
 <letra><letra><letra><letra>=<identificador><operador><identificador> →
 <letra><letra><letra><letra>=<letra><operador><letra> →

En la tabla de símbolos, se almacenarían las 4 letras, el símbolo = y la expresión que en este caso es un producto, es decir, suma = b*z.

c) Para las siguientes sentencias, ¿habría algún tipo de error? ¿Cuál?

A. { 8a = b + c }

La forma de esta sentencia es <identificador>=<expresión> y 8a no puede ser un identificador pues comienza por un dígito y en esta gramática cualquier identificador solo puede contener dígitos si empieza por alguna letra.

B. { suma + b = a * c }

Es un caso similar al anterior, una sentencia es de la forma <identificador>=<expresión>, pero en la parte izquierda de la igualdad no tenemos un identificador si no que tenemos una expresión pues es de la forma <identificador> <operador> <identificador> siendo + el operador.

C. { suma = a ^ c }

Aunque en este caso la estructura si es correcta, ^ no es un operador en esta gramática.