

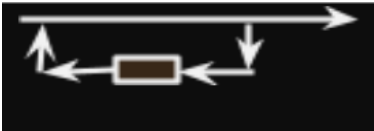


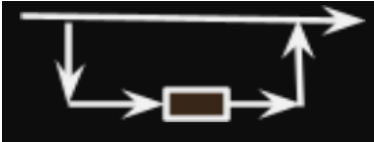




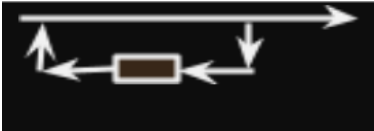

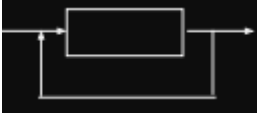

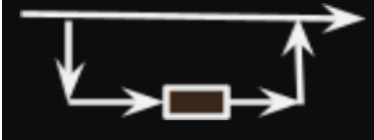
Ejercicio 1: Complete el siguiente cuadro:.....	2
Rta:.....	2
Ejercicio 2: ¿Cuál es la importancia de la sintaxis para un lenguaje? ¿Cuáles son sus elementos?.....	3
Ejercicio 3: ¿Explique a qué se denomina regla lexicográfica y regla sintáctica?.....	4
Ejercicio 4: ¿En la definición de un lenguaje, a qué se llama palabra reservada? ¿A qué son equivalentes en la definición de una gramática? De un ejemplo de palabra reservada en el lenguaje que más conozca. (Ada,C,Ruby,Python,..).....	4
Ejercicio 5: Dada la siguiente gramática escrita en BNF:.....	4
a- Identifique las componentes de la misma.....	4
b- Indique porqué es ambigua y corríjala.....	5
Ejercicio 6: Defina en BNF (Gramática de contexto libre desarrollada por Backus-Naur) la gramática para la definición de una palabra cualquiera.....	6
Ejercicio 7: Defina en EBNF la gramática para la definición de números reales. Inténtelo desarrollar para BNF y explique las diferencias con la utilización de la gramática EBNF.....	6
Ejercicio 8: Utilizando la gramática que desarrolló en los puntos 6 y 7, escriba el árbol sintáctico de:.....	8
a. Conceptos.....	8
b. Programación.....	9
c. 1255869.....	10
d. 854,26.....	11
e. Conceptos de lenguajes.....	11
Ejercicio 9: Definir utilizando diagramas sintácticos la gramática para la definición de un identificador de un lenguaje de programación. Tenga presente como regla que un identificador no puede comenzar con números.....	12
Ejercicio 10:.....	13
a) Defina con EBNF la gramática para una expresión numérica, dónde intervienen variables y números. Considerar los operadores +, -, * y / sin orden de prioridad. No considerar el uso de paréntesis.....	13
b) A la gramática definida en el ejercicio anterior agregarle prioridad de operadores.....	13
c) Describa con sus palabras los pasos y decisiones que tomó para agregarle prioridad de operadores al ejercicio anterior.....	14
Ejercicio 11: La siguiente gramática intenta describir sintácticamente la sentencia for de ADA, indique cuál/cuáles son los errores justificando la respuesta.....	14
Ejercicio 12: Realice en EBNF la gramática para la definición de un tag div en html 5. (Puede ayudarse con el siguiente enlace (https://developer.mozilla.org/es/docs/Web/HTML/Elemento/div)).....	15
Ejercicio 13: Defina en EBNF una gramática para la construcción de números primos.¿Qué debería agregar a la gramática para completar el ejercicio?.....	15
Ejercicio 14: Sobre un lenguaje de su preferencia escriba en EBNF la gramática para la definición de funciones o métodos o procedimientos (considere los parámetros en	

Ejercicio 1: Complete el siguiente cuadro:

Meta símbolos utilizados por Símbolo utilizado en		Símbolo utilizado en Diagramas sintácticos	Significado
BNF	EBNF		
palabra terminal	palabra terminal		Definición de un elemento terminal
		rectángulo 	Definición de un elemento no terminal
::=	::=	diagrama con rectángulos, óvalos y flechas	
	()	flecha que se divide en dos o más caminos	
< p > < p1 >			Repetición
			Repetición de 0 o más veces
	+		Repetición de 1 o más veces
	[]		

Rta:

Meta símbolos utilizados por Símbolo utilizado en		Símbolo utilizado en Diagramas sintácticos	Significado
BNF	EBNF		
palabra terminal	palabra terminal		Definición de un elemento terminal

<>	<>	rectángulo 	Definición de un elemento no terminal
::=	::=	diagrama con rectángulos, óvalos y flechas	Definición de una producción.
	()	flecha que se divide en dos o más caminos	Selección de una alternativa
< p > < p1 >	{ }		Repetición
	*		Repetición de 0 o más veces
	+		Repetición de 1 o más veces
	[]		Opcional, está presente o no lo está.

Ejercicio 2: ¿Cuál es la importancia de la sintaxis para un lenguaje? ¿Cuáles son sus elementos?

La sintaxis es un conjunto de reglas que definen como componer letras, dígitos y otros caracteres para formar los programas

La sintaxis establece reglas que definen cómo deben combinarse las componentes básicas, llamadas "word", para formar sentencias y programas.

En resumen la sintaxis importa para determinar que tan valido es un programa y cómo corregirlo si esta mal.

Elementos de la sintaxis:

- Alfabeto o conjunto de caracteres identificadores.
- Operadores.
- Palabra clave y palabra reservada.
- Comentarios y uso de blancos.

Ejercicio 3: ¿Explique a qué se denomina regla lexicográfica y regla sintáctica?

Lexicográfica = léxica.

Reglas léxicas: Conjunto de reglas para formar las "word", a partir de los caracteres del alfabeto.

Reglas sintácticas: Conjunto de reglas que definen cómo formar las "expresiones" y "sentencias".

Ejercicio 4: ¿En la definición de un lenguaje, a qué se llama palabra reservada? ¿A qué son equivalentes en la definición de una gramática? De un ejemplo de palabra reservada en el lenguaje que más conozca. (Ada,C,Ruby,Python,..)

En un lenguaje de programación una palabra reservada es aquella a que el lenguaje le asigna un significado específico y que el usuario no puede usar para definir otro significado.

Por ejemplo: if, for, else, elif, def, class, with, while, etc.

Ejercicio 5: Dada la siguiente gramática escrita en BNF:

$G = (N, T, S, P)$

$N = \{ \langle \text{numero_entero} \rangle, \langle \text{digito} \rangle \}$

$T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$S = \langle \text{numero_entero} \rangle$

$P = \{$

$\langle \text{numero_entero} \rangle ::= \langle \text{digito} \rangle \langle \text{numero_entero} \rangle \mid \langle \text{numero_entero} \rangle \langle \text{digito} \rangle \mid \langle \text{digito} \rangle$

$\langle \text{digito} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$\}$

a- Identifique las componentes de la misma.

$G = (N, T, S, P)$: 4-tupla donde se define el "Conjunto de reglas finitas que define un conjunto infinito de posibles sentencias válidas en el lenguaje."

N = "Conjunto de símbolos no terminales".

T = "Conjunto de símbolos terminales".

S = "Símbolo distinguido de la gramática que pertenece a N".

P = "Conjunto de producciones".

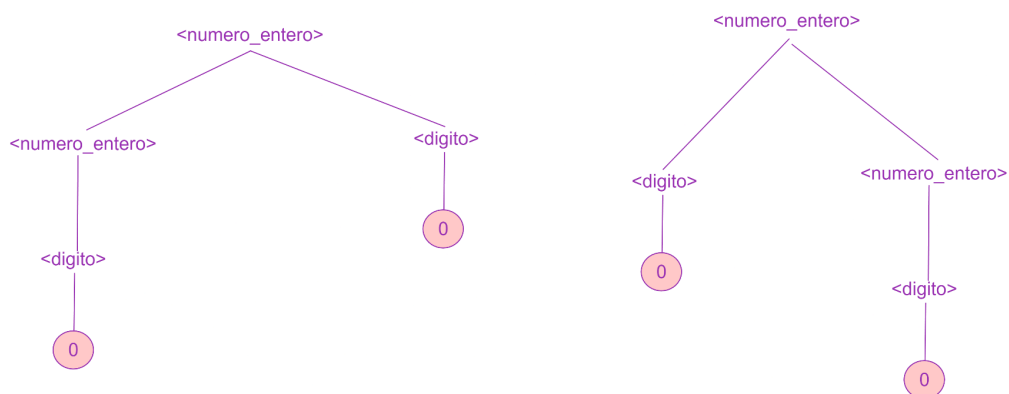
b- Indique porqué es ambigua y corrijala.

"Una gramática es ambigua si una sentencia puede derivarse de más de una forma".

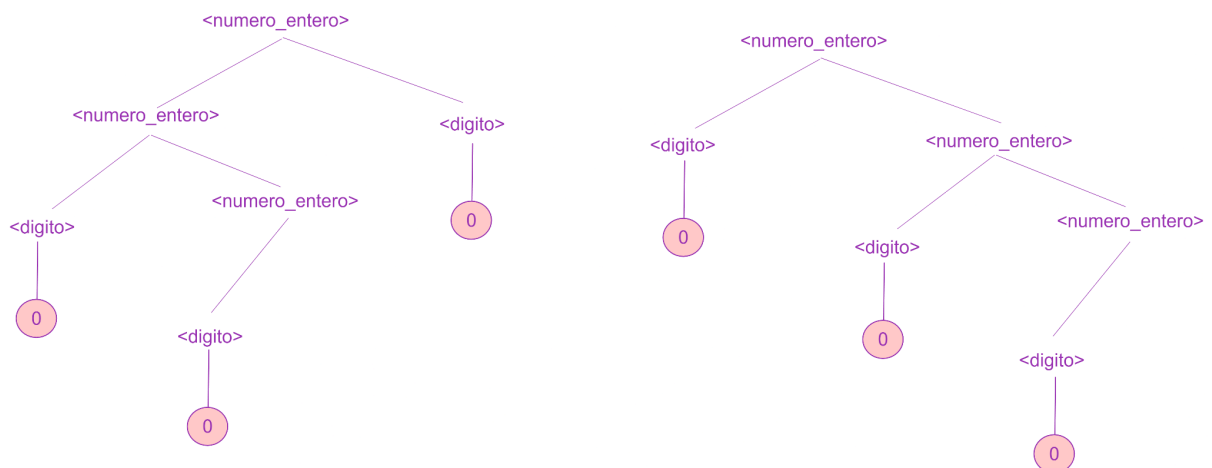
A simple vista es obvio, ya que hay 2 formas de definir el cuerpo del numero no terminal: $\langle \text{digito} \rangle \langle \text{numero_entero} \rangle$ | $\langle \text{numero_entero} \rangle \langle \text{digito} \rangle$

Representan al final lo mismo. Vamos a verlo con un arbol de derivación:

número 00:



número 000:



En este caso sabemos que se da porque las producciones recursivas deben tener un sentido de asociatividad, por derecha o por izquierda pero no los 2.

"Regla recursiva por la izquierda: La asociatividad es por la izquierda. El símbolo no terminal de la parte izquierda de una regla de producción aparece al comienzo de la parte derecha"

Regla recursiva por la derecha: La asociatividad es por la derecha. El símbolo no terminal de la parte izquierda de una regla de producción aparece al final de la parte derecha."

Versión piola:

$G = (N, T, S, P)$

$N = \{ \langle \text{numero_entero} \rangle, \langle \text{digito} \rangle \}$

$T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$S = \langle \text{numero_entero} \rangle$

$P = \{$

$\langle \text{numero_entero} \rangle ::= \langle \text{digito} \rangle \langle \text{numero_entero} \rangle \mid \langle \text{digito} \rangle$

$\langle \text{digito} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$\}$

Ejercicio 6: Defina en BNF (Gramática de contexto libre desarrollada por Backus-Naur) la gramática para la definición de una palabra cualquiera.

$G = (N, T, S, P)$

$N = \{ \langle \text{palabra} \rangle, \langle \text{caracter} \rangle \}$

$T = \{A, \dots, Z, a, \dots, z\}$

$S = \langle \text{palabra} \rangle$

$P = \{$

$\langle \text{palabra} \rangle ::= \langle \text{caracter} \rangle \langle \text{palabra} \rangle \mid \langle \text{caracter} \rangle$

$\langle \text{carácter} \rangle ::= A \mid \dots \mid Z \mid a \mid \dots \mid z$

$\}$

Ejercicio 7: Defina en EBNF la gramática para la definición de números reales. Inténtelo desarrollar para BNF y explique las diferencias con la utilización de la gramática EBNF.

$G = (N, T, S, P)$

$N = \{ \langle \text{real} \rangle, \langle \text{dígito} \rangle, \}$

$T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, "+", "-", ".", "\}$

$S = \langle \text{real} \rangle$

$P = \{$

$\langle \text{real} \rangle ::= [(+ \mid -)] \{ \langle \text{dígito} \rangle \}^+ [\{ \langle \text{dígito} \rangle \}^+]$

$\langle \text{dígito} \rangle ::= (0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9)$

}

Versión de pobre:

$G = (N, T, S, P)$

$N = \{ \langle \text{real} \rangle, \langle \text{dígito} \rangle, \langle \text{decimal} \rangle, \langle \text{número} \rangle \}$

$T = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, "+", "-", ",", " " \}$

$S = \langle \text{real} \rangle$

$P = \{$

$\langle \text{real} \rangle ::= \langle \text{signo} \rangle \langle \text{número} \rangle \langle \text{decimal} \rangle \mid \langle \text{signo} \rangle \langle \text{número} \rangle \mid \langle \text{número} \rangle$

$\langle \text{signo} \rangle ::= + \mid -$

$\langle \text{decimal} \rangle ::= , \langle \text{número} \rangle$

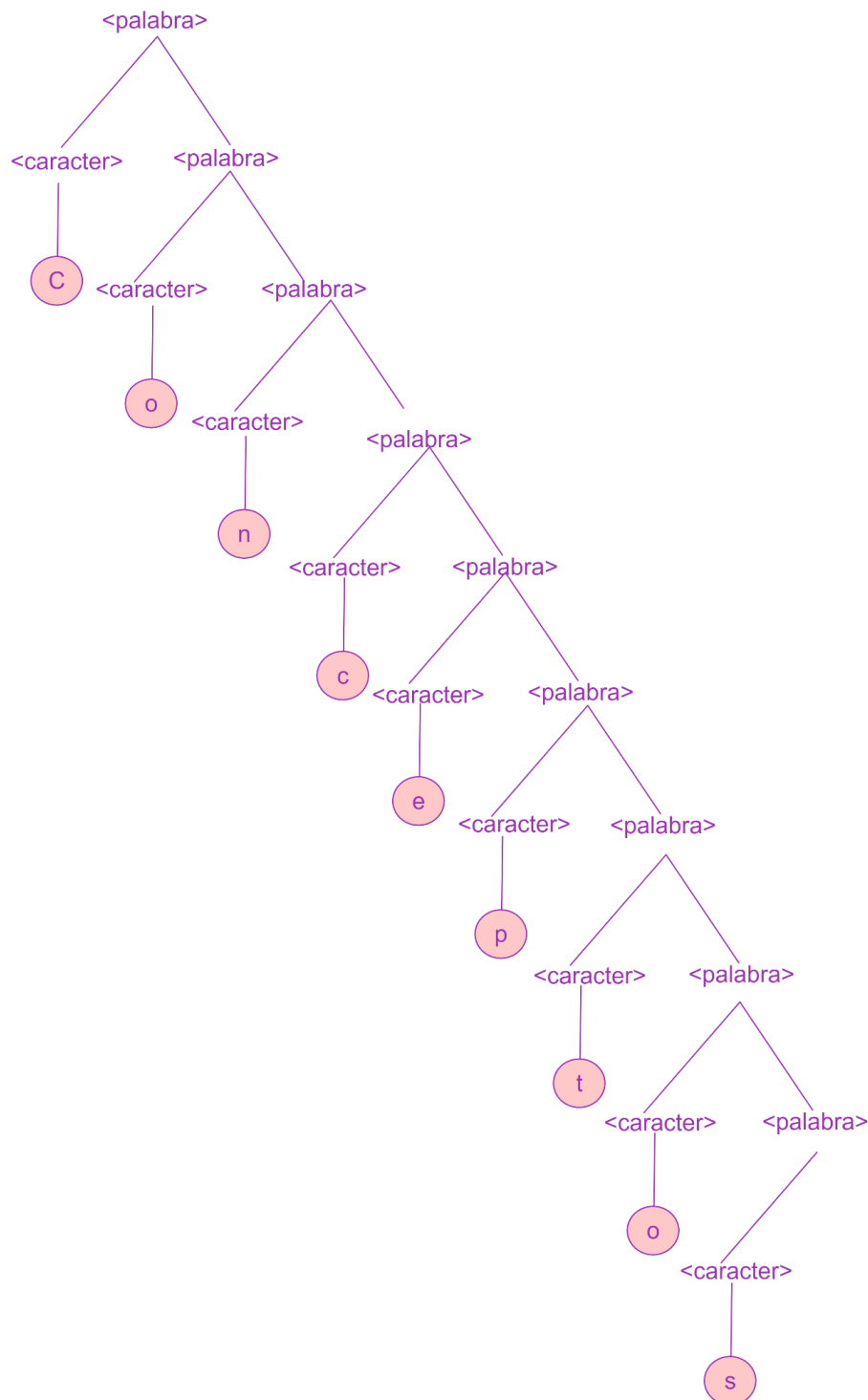
$\langle \text{número} \rangle ::= \langle \text{dígito} \rangle \mid \langle \text{dígito} \rangle \langle \text{número} \rangle$

$\langle \text{dígito} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

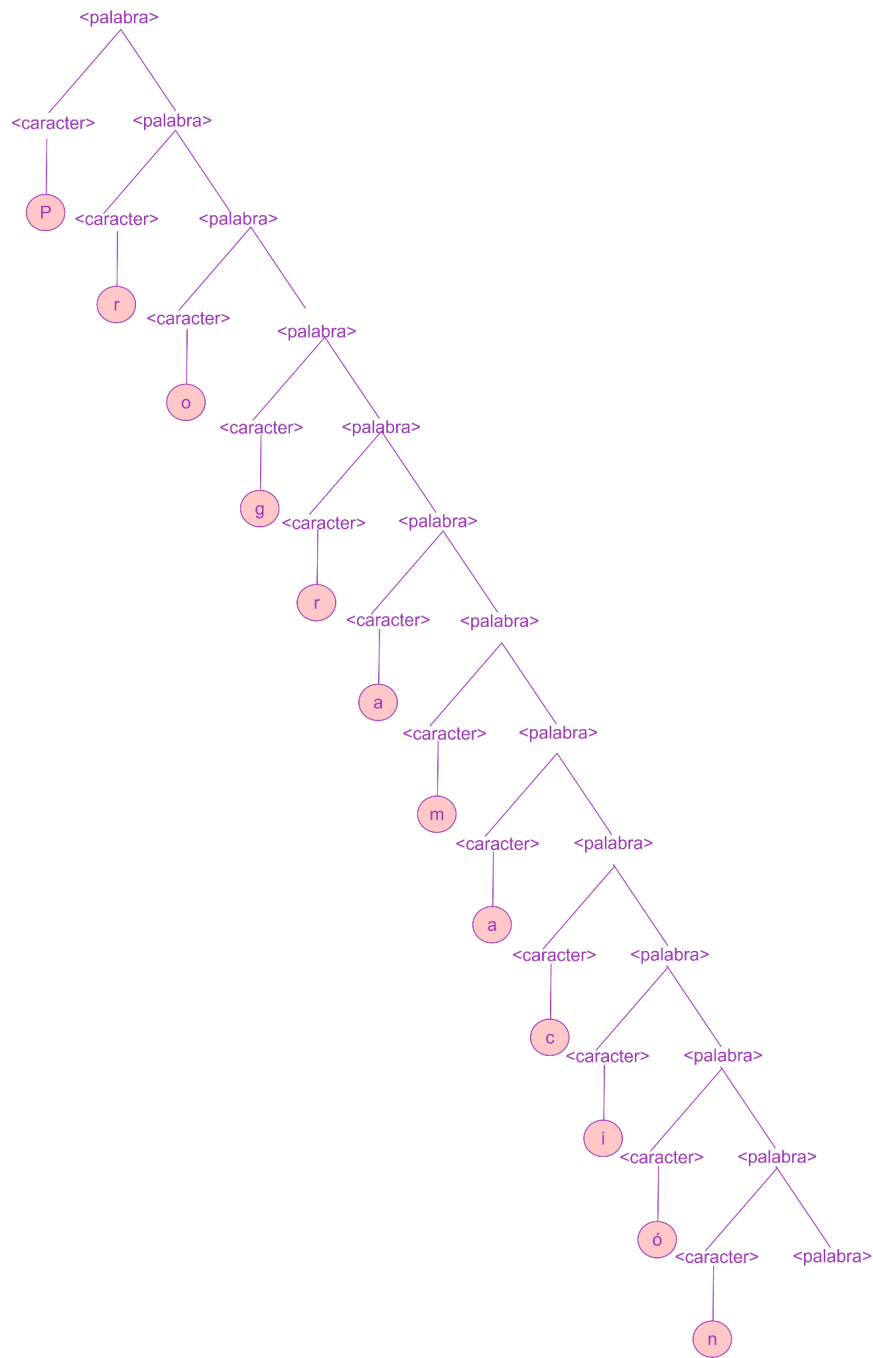
$\}$

Ejercicio 8: Utilizando la gramática que desarrolló en los puntos 6 y 7, escriba el árbol sintáctico de:

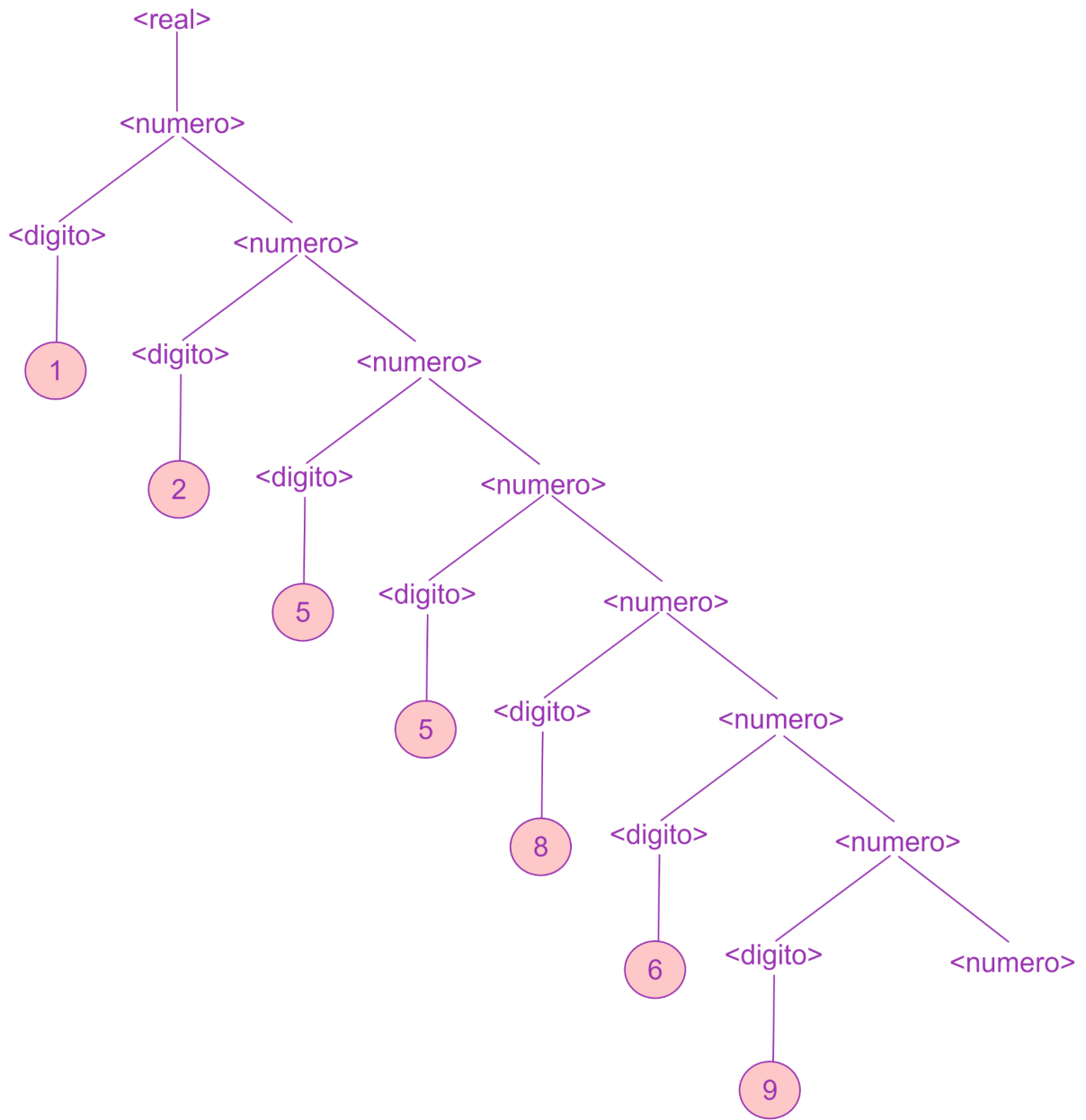
a. Conceptos



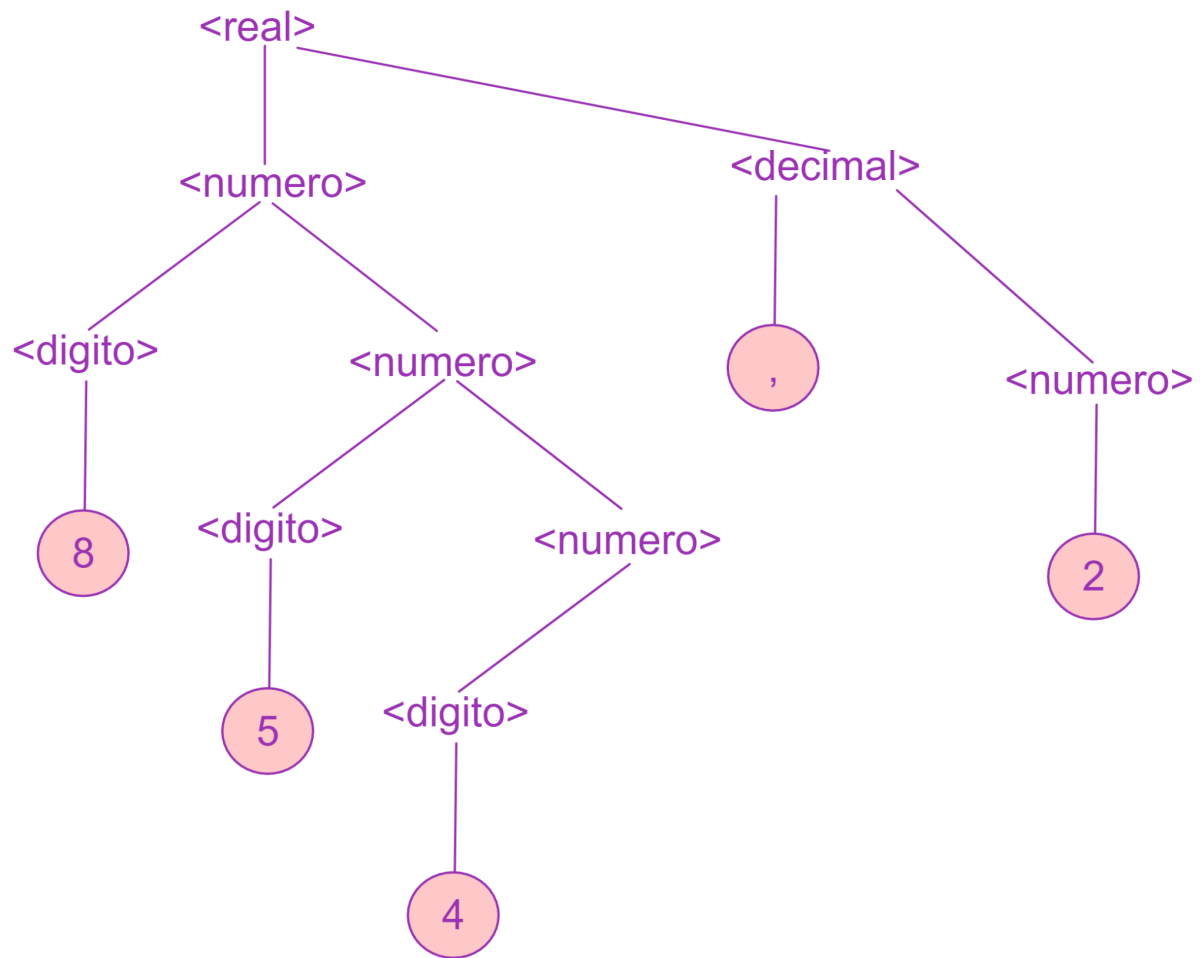
b. Programación



c. 1255869



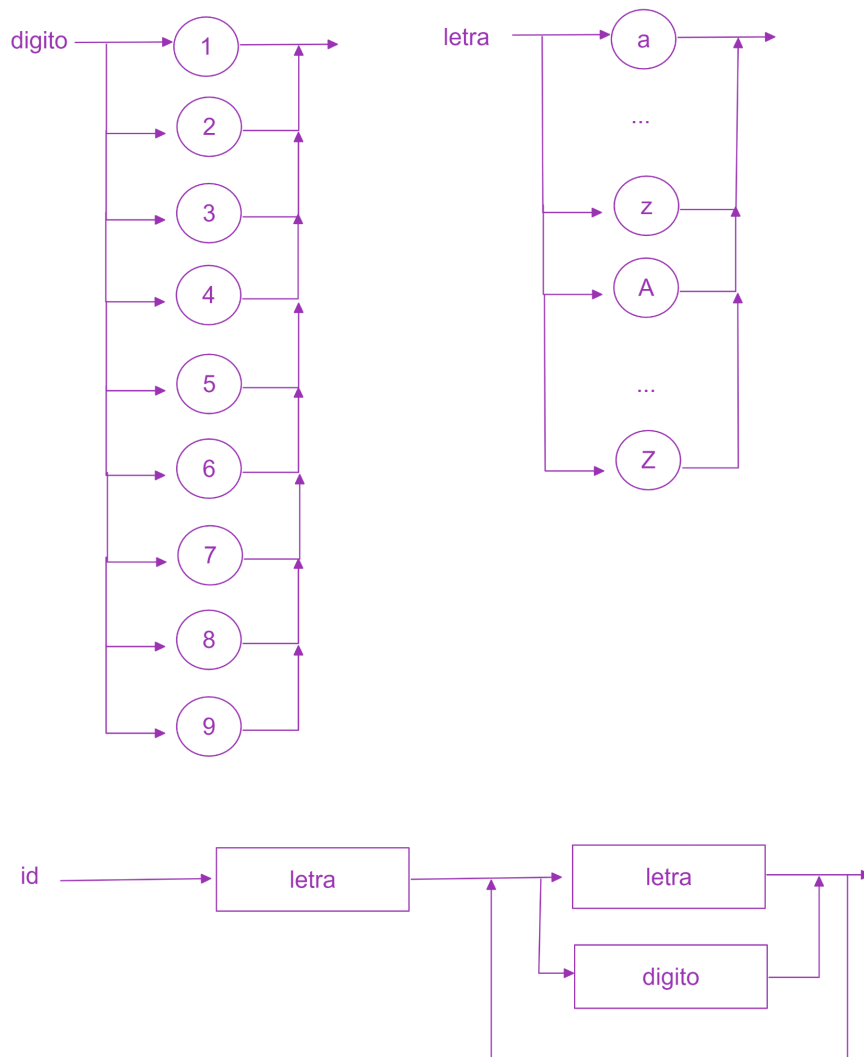
d. 854,26



e. Conceptos de lenguajes

Esto es un string, no una palabra.

Ejercicio 9: Definir utilizando diagramas sintácticos la gramática para la definición de un identificador de un lenguaje de programación. Tenga presente como regla que un identificador no puede comenzar con números.



Nota: Los circulitos son los símbolos terminales que en la explicación son óvalos.

Ejercicio 10:

a) Defina con EBNF la gramática para una expresión numérica, dónde intervienen variables y números. Considerar los operadores +, -, * y / sin orden de prioridad. No considerar el uso de paréntesis.

```
G = ( N, T, S, P )
N = { <variable>, <expresión>, <dígito>, <letra>, <operador>, <número> }
T = { 0, ..., Z, A, ..., Z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, "+", "-", "/", "*" }
S = <expresión>
P = {
  <expresión> ::= ( <variable> | <número> ) { <operador> ( variable | número ) } *
  <variable> ::= <letra> { ( <letra> | <número> ) } *
  <número> ::= [ ( + | - ) ] { <dígito> } +
  <letra> ::= ( a | ... | z | A | ... | Z )
  <dígito> ::= ( 1 | 2 | 3 | ... | 9 )
  <operador> ::= ( + | - | / | * )
}
```

b) A la gramática definida en el ejercicio anterior agregarle prioridad de operadores.

```
G = ( N, T, S, P )
N = { <variable>, <expresión>, <dígito>, <letra>, <operador_prioritario>,
  <operador_no_prioritario>, <número>, <termino> }
T = { 0, ..., Z, A, ..., Z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, "+", "-", "/", "*" }
S = <expresión>
P = {
  <expresión> ::= <término> { <operador_no_prioritario> <término> } *
  <término> ::= [ ( + | - ) ] ( <variable> | <número> ) { <operador_prioritario> ( <variable> |
  <número> ) } *
  <variable> ::= <letra> { ( <letra> | <número> ) } *
  <número> ::= [ ( + | - ) ] { <dígito> } +
  <letra> ::= ( a | ... | z | A | ... | Z )
  <dígito> ::= ( 1 | 2 | 3 | ... | 9 )
  <operador_prioritario> ::= ( / | * )
  <operador_no_prioritario> ::= ( + | - )
}
```

c) Describa con sus palabras los pasos y decisiones que tomó para agregarle prioridad de operadores al ejercicio anterior.

Para representar la prioridad de operadores, la idea es unir todas las posibles operaciones prioritarias en un término (si es que hay en ese término operaciones prioritarias), para luego sumarlas o restarlas entre sí:

Por eso surge el <término>, representa una multiplicación, división, o número, o varios de ellos.

Por ejemplo:

```
juan * pedro / perez
10
10 * 20
-10 * 20
roman
```

Luego en la expresión, es simplemente una concatenación de términos (como una fórmula real):

```
juan * pedro / perez + 10 - 10*20 + -10*20 - roman
```

Ejercicio 11: La siguiente gramática intenta describir sintácticamente la sentencia for de ADA, indique cuál/cuáles son los errores justificando la respuesta.

N= {<sentencia_for>, <bloque>, <variable>, <letra>, <cadena>, <dígito>, <otro>, <operacion>, <llamada_a_funcion>, <numero>, <sentencia> }

P= { <sentencia_for>::= for (i= IN 1..10) loop <bloque> end loop;

<variable>::= <letra> | <cadena>

<cadena>::= { (<letra> | <dígito> | <otro>) }+

<letra>::= (a | .. | z | A | .. | Z)

<dígito>::= (1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0)

<bloque>::= <sentencia> | <sentencia> <bloque> | <bloque> <sentencia> ;

<sentencia>::= <sentencia_asignacion> | <llamada_a_funcion> | <sentencia_if> | <sentencia_for> | <sentencia_while> | <sentencia_switch> }

Errores:

1. Faltan 2 elementos de la 4-tupla de definición de gramática y además la 4-tupla en sí.
2. Nunca define <sentencia_while>, <sentencia_switch>, <sentencia_if>, <llamada_a_funcion>, y <sentencia_asignacion>.
3. <bloque> es ambiguo porque una misma producción puede derivarse desde izquierda o derecha.

Ejercicio 12: Realice en EBNF la gramática para la definición de un tag div en html 5. (Puede ayudarse con el siguiente enlace (<https://developer.mozilla.org/es/docs/Web/HTML/Elemento/div>))

G = (N, T, S, P)

N = { <div>, <style>, <class>, <cadena>, <carácter>, <tag> }

T = { a, ..., z, A, ..., Z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, "+", "-", "/", "*", "<", ">", ";", ",", ".", ":", "{", "}", "[", "]", "_", " " }

S = <div>

P = {

<div> ::= <div> { (<style> | <class>)* } { <cadena> | <div> | <tag> }* /div>

<style> ::= style=" [<cadena>] "

<class> ::= class=" [<cadena>] "

<cadena> ::= { <carácter> } +

<carácter> ::= (a | ... | z | A | ... | Z | 1 | ... | 9 | | < | > | ; | ' | . | { | } | [|] | - | _ | + | - | / | *)

}

Nota: En esta implementación de div asumí que las otras tags ya existían.

Ejercicio 13: Defina en EBNF una gramática para la construcción de números primos. ¿Qué debería agregar a la gramática para completar el ejercicio?

Faltaría una forma de chequear que el número no sea divisible más que por 1 y el mismo.

Ejercicio 14: Sobre un lenguaje de su preferencia escriba en EBNF la gramática para la definición de funciones o métodos o procedimientos (considere los parámetros en caso de ser necesario)

G = (N, T, S, P)

N = { <div>, <style>, <class>, <cadena>, <carácter> }

T = { a, ..., z, A, ..., Z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, "(", ")", ",", " " }

S = <funcion>

P = {

<funcion> ::= def <nombre> <parámetro>: {<sentencia>}* [<return>]

<parámetros> ::= ({nombre,*})

<nombre> ::= <letra>{(<dígito> | <letra>)}*

<dígito> ::= (1 | ... | 9)

<letra> ::= (a | ... | z | A | ... | Z)

}

Nota: Asumo la existencia de <sentencia> y return.