



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE IZTAPALAPA



INSTITUTO TECNOLÓGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

EJERCICIOS.

PRESENTAN:

RUEDA IMÁN SOFIA VIRIDIANA
CELISEO GOMEZ ADAN GAMALIEL
MEJIA RAMOS LUIS ENRIQUE
QUINTERO BOLIO ERIK EDUARDO

NO. DE CONTROL:

171080043

171080044

161080183

171080151

DOCENTE:

PARRA HERNANDEZABIEL TOMAS

CIUDAD DE MÉXICO

ENERO/2021



1. Lenguajes aceptados.

$$\Sigma = \{\text{una} | \text{segundo} | a, b\}$$

$$\Sigma = \{0, 1\}$$

$$\Sigma = \{\text{una} | \text{segundo} | a, b\}$$

Idioma $\{\text{una}, \text{segundo}\}$

cadenas validas.

Segundo segundo, una una, una segundo

Idioma $\{0, 1\}$

Cadenas validas.

II, 01, 011, 111, 1011

Idioma $\{\text{una antes de cristo}\}$

cadena valida.

antes de cristo antes de cristo
una una, una antes de cristo.

2. Construye una fa aceptando cada uno de los siguientes idiomas

1. $w \in \{una, segundo\}^*$

2. $w \in \{0,1\}$

3. $w \in \{una, antes de cristo\}^*$

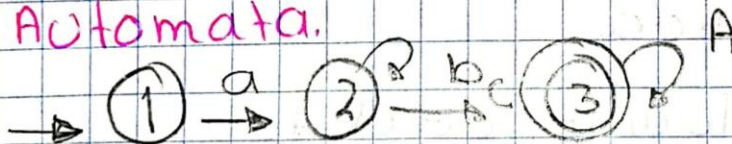


labl C.A Y Cadenas

C.A. ab. C.A C.A abab

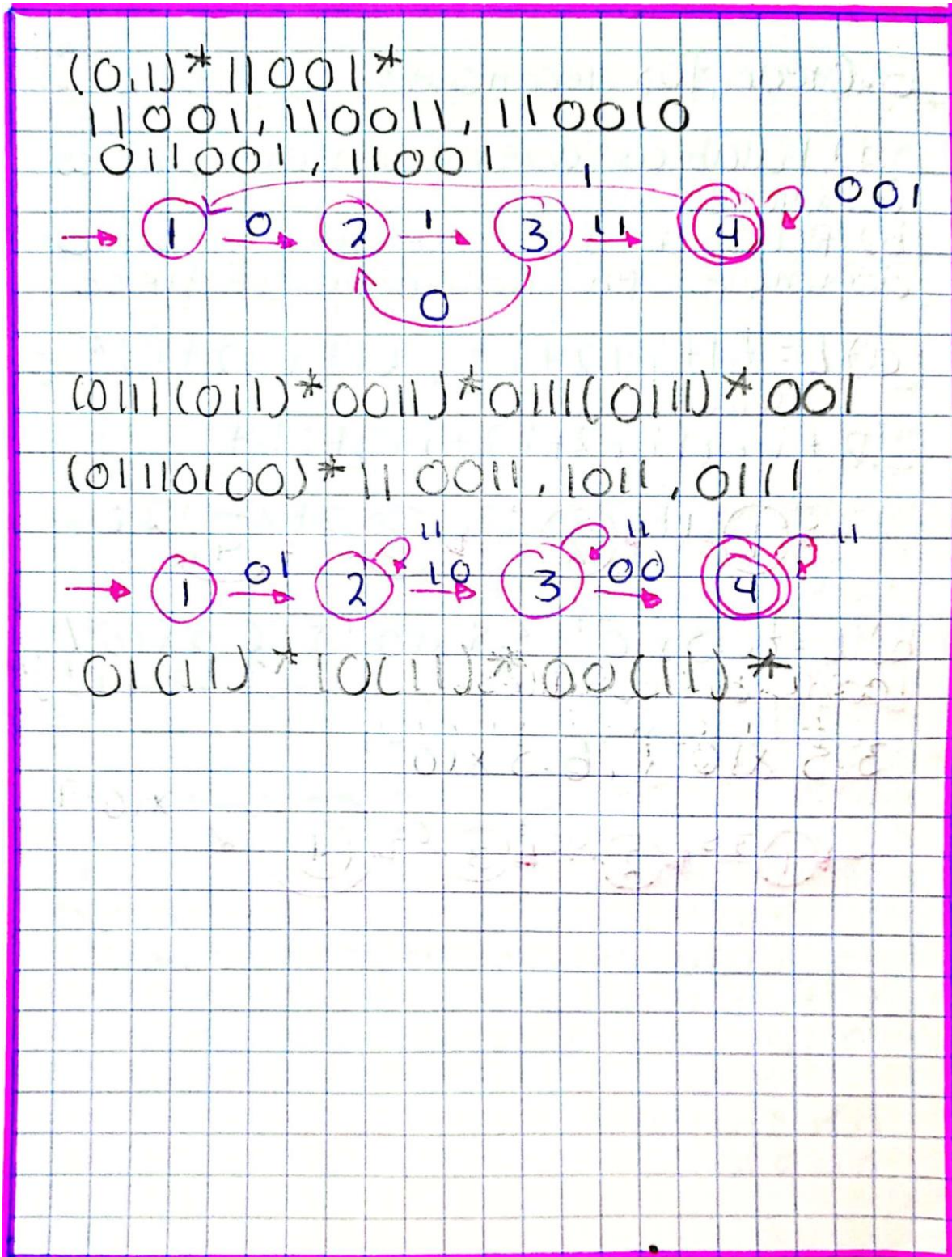
C.A C.A. C.A ababab

Automata.



E.R

$ab^*c(A)^*$



3. Crear Fas reconocer:

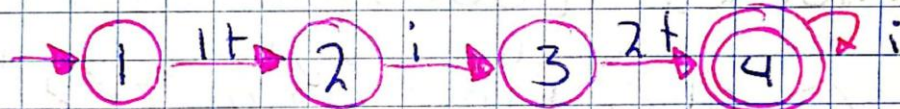
a). Palabras que representen números completos.

b). Palabras que representen números decimales en notación científica

a). $L = \{1+i, 1+2+i\}, (1+i | 1+2+i)^*$

cadena posibles

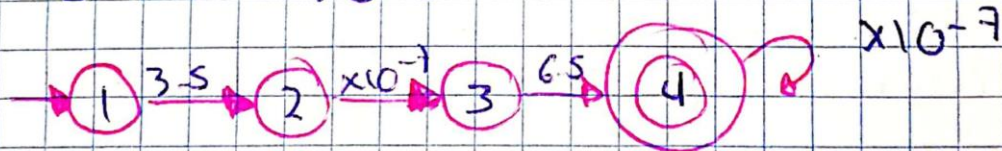
$1+i, 1+i, 1+2+i, 1+2+i, 1+i | 1+i$



b). $L = \{6.5 \times 10^{-7} | 3.5 \times 10^{-7}\}, (6.5 \times 10^{-7} | 3.5 \times 10^{-7})^*$

cadena posibles.

$3.5 \times 10^{-7}, 6.5 \times 10^{-7}$



Escribe una expresión regular para cada de los siguientes idiomas.

A. Dado un alfabeto $\Sigma = \{0,1\}$ L es el conjunto de todas las cadenas de pares alternas de ceros y pares de unos.

$L = \{0111, 1011, 0111, 1011\}$

B. Dado un alfabeto $\Sigma = \{0,1\}$ L es el conjunto de todas las cadenas V o Y que contiene un número par de 0 a un número par de 1.

$L = \{011, 001, 101, 010, 00, 11, 011, 100, 110\}$

C. Dado un alfabeto $\Sigma = \{a, b, c, d\}$, L es el conjunto de cadenas $xyzw$, donde xyw son cadenas de uno o mas caracteres en Σ , y es cualquier caracter vale o alfabeto. Cada cadena $xyzw$ contiene 2 palabras xy y wy contruido a partir de letras en Σ . Las palabras terminan en la misma letra y . Ellos estan separado por " z ".

$L = \{abcdba, bacdob, abc, bcd, cadbac\}$

D. Dado el alfabeto inglés en minúsculas, L es el conjunto de todas las cadenas en cuyas letras aparecen en el orden lexico gráfico.

$L = \{abcd, bcd, cd, ac, bd, ab, ad, bc\}$

E. Dado un alfabeto $\Sigma = \{+, -, \times, \div, ()\}$, L es el conjunto de expresiones algebraicas usando suma, resta, multiplicación, división y parentesis sobre enteros (factores).

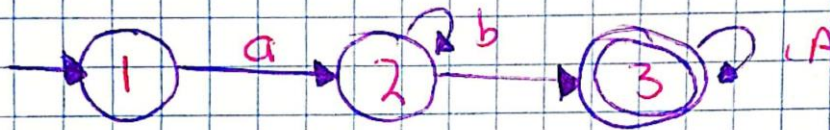
$L = \{2+3-4 \times 5 \div 6(7+7), 2-3+9 \div 5 \times 6, (7-7) \times \div 4 + (x + xy) - x, (4)(x) + x - y + x \div y\}$

7. Considere las 3 expresiones regulares.

$(ab|c.A)^*$

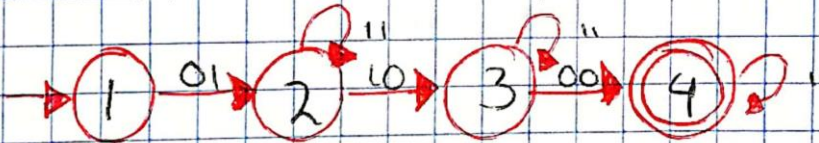
$C.A, ab, c.A, c.A$

$abab, c.A, c.A, ababab$



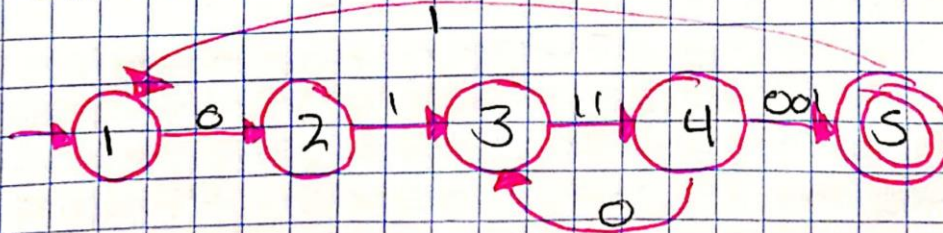
Ej. $ab^*c(A)^*$

$(0110100)^* 110011, 1011, 0111$



Ej. $01(11)^*10(11)^*00(11)^*$

$(0,1)^* 11001^* 11001, 110011, 110010$
 $011001, 11001$
 $(011(011)^*0011)^* 0111(011)^* 00$



Ejercicios

1 En este ejercicio se crea un subconjunto de la sintaxis de la E.R

Quantity $*, +, ?$

Choose: $|$

Group: $()$

$P \rightarrow P | P$
 $| PP$
 $| (P)Q$
 $| A$

$B \rightarrow B \text{ char}$
 $| B [S]$
 $| \text{char}$
 $| [S]$

$A \rightarrow ABQ$
 $\rightarrow B$

$S \rightarrow S \text{ char}$
 $\rightarrow \text{char}$

$Q \rightarrow *$
 $| ?$
 $| +$
 $| \varepsilon$

2 Escribir una gramática libre de contexto para la forma Backus Naur (bnf). notación para gramáticas libres de contexto.

$S \rightarrow \langle \text{word} \rangle ; = P$

$P \rightarrow P | A \text{ "word"}$
 $| A \langle \text{word} \rangle$
 $| \text{"word"}$
 $| \text{"word"}$

INSTITUTO TECNOLÓGICO DE IZTAPALAPA

3: Una gramática en la que cada oración puede tener un árbol gramatical único por inferencia del extremo izquierdo

- A grammar in which each sentence can get a unique grammar tree by far-left inference.

4: Primero se convertirá la recursividad izquierda $R \rightarrow Rbc$ en recursión por la derecha. La sintaxis convertida es la siguiente.

$L \rightarrow Ra$
 $| Qba$
 $R \rightarrow ab a R'$
 $| c a b a R'$
 $R' \rightarrow bc R'$
 $| \epsilon$
 $Q \rightarrow b b c$
 $| bc$

$L \rightarrow Ra$
 $| Qba$
 $R \rightarrow ab a R'$
 $| c a b a R'$
 $R' \rightarrow bc R'$
 $| \epsilon$

$\rightarrow Q \rightarrow | a'$
 $Q' \rightarrow bc$
 $| c$

Extraeremos el factor común b y nos resulta

$A \rightarrow Ba$
 $B \rightarrow d abd$
 $| c B t a c \epsilon$
 $D \rightarrow a c \epsilon$
 $| \epsilon$
 $E \rightarrow b F$
 $F \rightarrow a c \epsilon$
 $| \epsilon$

5 La gramática tiene recursión por la izquierda de manera indirecta.

La derivación provocará una recursividad directa a la izquierda.

$A \rightarrow Ba$
 $B \rightarrow dab$
 $\quad | cb$
 $c \rightarrow cBc'$
 $\quad | dabacc'$
 $c \rightarrow tacc$
 $\quad | \epsilon$

Gen $B \rightarrow cb$
 $A \rightarrow Ba$
 $B \rightarrow dab$
 $\quad | cBc'b$
 $\quad | aabacc'b$
 $c' \rightarrow bacc'$
 $\quad | \epsilon$
 Extraemos dab

$A \rightarrow Ba$
 $B \rightarrow dab$
 $\quad | cBbaacc't$
 $D \rightarrow acc'b$
 $\quad | \epsilon$
 $c \rightarrow bacc'$
 $\quad | \epsilon$

Encontramos $First + (c) \rightarrow tacc'$
 A TIERTA $(c' \rightarrow \epsilon) \{t\}$, no es un
 conjunto vacío entonces debemos o hay
 retroceso
 agregamos $\epsilon \rightarrow c't$ y eliminamos
 c'

$A \rightarrow Ba$
 $B \rightarrow dab$
 $\quad | cBbaacc'\epsilon$
 $D \rightarrow acc'\epsilon$
 $\quad | \epsilon$
 $E \rightarrow tacc'$
 $\quad | b$

6 FIRST: para la sintaxis a la declaración derivada que contiene solo el terminador es menor a k , agregue la declaración completa a FIRST(α) si la longitud de la declaración es mayor o igual que k agregue un prefijo con una longitud de k a FIRST(α) el conjunto de todas las declaraciones que se procesan es ITST(α)

FOLLOW(α): para el no terminador α , FOLLOW(α) es un prefijo de k para la longitud de todas las cadenas de símbolos posibles inmediatamente después de α .

La expresión es de la siguiente manera

$$\text{FOLLOW}(\alpha) = \{ \text{FIRST}(x) \mid \text{cualquier cadena de símbolos en forma de } \alpha x \text{ derivada de } S \}$$

Para facilitar las descripciones de las condiciones que cumplen con la sintaxis $u(\alpha)$, defina un operador definimos dos colecciones, S_1 y S_2 y una de las cadenas en S_1 en S_2 y la colección de prefijos de k para todas las cadenas posibles en S_1 o S_2

→

$$L_1 f + L_2 t \rightarrow L_1 f t + L_2 t$$

$$| \epsilon$$

$$L_1 f + \rightarrow \uparrow L_1 f$$

$$L_1 f t \rightarrow L_1 f t \downarrow$$

$$| \downarrow$$

La gramática no puede derivar de la secuencia $\uparrow \downarrow \uparrow \downarrow$, y la secuencia cumple con los requisitos de la pregunta por lo que no requiere que la gramática pueda derivar todos las secuencias posibles.

Si deseamos derivar $\uparrow \downarrow \uparrow \downarrow$, podemos transformar la gramática

$$Lift + List \rightarrow Lift + Lift + List$$

$$\uparrow \uparrow Lift + List$$

$$Lift \rightarrow \uparrow \uparrow Lift$$

$$Lift \rightarrow Lift \downarrow$$

8 Tomamos la sintaxis de la expresión de suma, resta, multiplicación y división y declaramos b+c+d+a como ejemplo.

$$E \rightarrow E + T$$

$$\downarrow E - T$$

$$\downarrow T$$

$$T \rightarrow T \times F$$

$$\downarrow T \div F$$

$$\downarrow F$$

$$F \rightarrow \text{name}$$

El análisis de sintaxis top-down (LL1) construye un árbol de sintaxis para la declaración b+c+d+a de la siguiente forma.



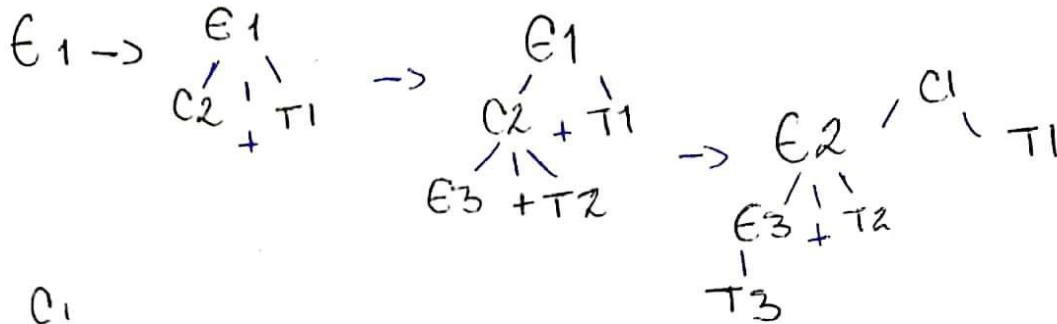
EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE IZTAPALAPA



C_1
 $E_2 + T_1$
 $/ \quad \backslash$
 $E_3 + T_2$
 T_3
 $|$
 T_1
 $|$
 b

CS Escaneado con CamScanner



Av. Telecomunicaciones S/N, Col.Chinampac de Juarez, C.P. 09208, Alcaldía de Iztapalapa,
Ciudad de México Tel.5773-8210, e-mail:division@iztapalapa.tecnm.mx www.tecnm.mx /
www.iztapalapa.tecnm.m