Trabajo Práctico

CFG – Formas Normales – PDA - ASDPNR

1. **Defina CFGs equivalentes a las siguientes ER:**
   1. a(a|bcc\*)\*b
   2. (ab|ba|aa|bb)\*(a|e)
   3. bab\*(c|ac)(cc|abc)
   4. ((a|bb)\*|cb)\*
2. **Escriba una derivación de al menos 4 pasos para cada una de las CFG siguientes. Exprese en el lenguaje natural cuales son los lenguajes generados por cada CFG:**
   1. A  aA | e

a\*

* 1. A  aA | a

aa\*

* 1. A  Aa | a

aa\*

* 1. A  AA | b

b2^n con n>= 0

* 1. A  Aa | Ab | e

(ab)\*

* 1. A  aA | bA | a
  2. A  Aa | bA | a

b\*aa\*

* 1. A  bAa | e

no es regular

* 1. A  bAa | a

no es regular

* 1. A  abAa | e
  2. A  aAa | a

a(aa)\*

* 1. A  bAa | Aa | Ab | e
  2. A  AaA | b
  3. A  AaA | AbA | c

1. **Considere la gramática:**

S  **(** L **)** | **a**

L  L **,** S | S

1. Demuestre, a través de derivaciones, que las siguientes cadenas son frases:
   1. (a, a, a)
   2. ((a, a))
2. Si las siguientes cadenas son frases, realice una demostración encontrando sus árboles de derivación. En caso contrario explique por qué no son frases:
   1. (a, a)
   2. (a, ((a, a, a))
   3. (a, (a, a a))
   4. (a, (a, (a), (a, a)))
3. **Defina CFGs para los siguientes lenguajes:**
4. Cadenas sobre {a, b} que contengan como subcadena a **aba** y cuya longitud sea impar
5. Cadenas sobre {a, b, c} de la forma cnambmcn , para n y m >0.
6. Cadenas sobre {a, b} de la forma am bn bm an , para n y m >=0.
7. Cadenas sobre {a, b} de longitud 2n, n>0.
8. Cadenas sobre {0, 1, #}, de la forma **x#yR**, con **x** e **y** en {0, 1}\*, |x|=|y|, donde el valor decimal de **y** es igual al valor decimal de **x** + 1. **yR** significa “reverso de y”.
9. Cadenas sobre {0, 1} con cantidad par de **0** y cantidad impar de **1**.
10. Cadenas sobre {a, b} que contengan la misma cantidad de **a** que de **b**.
11. Cadenas sobre {a, b, c} de la forma {aibjck | i <> j o j <> k }

1. **Defina a través de una CFG la sintaxis que represente un bloque de sentencias de asignación (con expresiones aritméticas) y condicionales (tipo if - then - else).**
2. Defina a través de una CFG la sintaxis de definiciones de constantes, tipos y variables del lenguaje Pascal, considerando los tipos INTEGER, CHAR, STRING, BOOLEAN, RECORD y ARRAY.
3. **Defina una CFG que genere un lenguaje con las siguientes características:**
4. Un programa es una secuencia de funciones
5. Una función contiene la palabra reservada FUNCION, el nombre, una lista de parámetros entre paréntesis y el cuerpo de la misma.
6. La lista de parámetros consiste en identificadores separados por espacios en blanco, sin definición de tipos
7. El cuerpo de la función es una secuencia de sentencias de la forma:

expresión condicional **:** expresión aritmética

1. Las expresiones aritméticas permitidas son: suma, resta, producto y cociente; y se realizan sobre variables, constantes e identificadores de función.
2. Los operadores relacionales son: menor, mayor, igual y distinto

Ejemplo:

FUNCION siguienteMayor (A B)

A > B+1 : A;

A < B : B\*2/(A+1);

A = B : B – minimo(A\*2, 5);

FIN

1. Defina una CFG que reconozca el cuerpo de un procedimiento o función en Pascal, considerando la definición de las estructuras WHILE, FOR e IF; sentencias de asignación; expresiones aritméticas, relacionales y lógicas.
2. Defina una gramática que reconozca un lenguaje orientado a conjuntos numéricos, con las siguientes características:
3. Un programa es una secuencia de expresiones
4. Cada expresión puede ser una asignación, una lectura de uno o más variables conjunto, o una escritura por pantalla de variables o constantes conjunto.
5. Las operaciones permitidas son: complementación, intersección y unión.
6. Una constante conjunto tiene dos formas:

- por extensión, que se expresa como una secuencia de números entre llaves

- por comprensión, que se expresa mediante un identificador de función entre llaves

1. Defina una CFG para un lenguaje imperativo creado por Ud. El lenguaje debe contener al menos una estructura de asignación, una cíclica y una condicional, así como alguna forma de entrada - salida. Puede ser dinámicamente tipado. Dé un ejemplo de programa válido, y explique su significado.
2. **Considere la gramática:**

S  S**a**S | T

T  **b**T | **b** | S

1. Demuestre que esta gramática es ambigua construyendo dos árboles de derivación para **bbab**
2. Construya dos árboles de derivación correspondiente a **babbabb**
3. ¿Qué lenguaje genera esta gramática?
4. Defina una CFG equivalente no ambigua
5. **Determine cuales de las CFG de definió en los puntos 5 al 10, son ambiguas. En tales casos, construya CFGs equivalentes no ambiguas.**
6. **Dadas las CFG que se expresan a continuación**
7. Elimine símbolos inútiles
8. Construya una gramática equivalente en CNF
9. Construya una gramática equivalente en GNF
10. Construya el PDA equivalente

1. C  C + C | C; E | C ^ C | E

D  C + E | a | b

E  a | b | C

2. S  abB | Sa

B  eBf | dBg |  | BSD

C  CaC | aCab | a | S

D  Da | Dab

3. S  AB | SBC | ABD | AS

A A\* | A+ | SD | #

G  GaS | Gsd | bDa

C  aF | Cb

D  Da | aDbY | SCD

B  e | B | aBc | bBdc

1. **Elimine ambigüedad, recursividad por izquierda y factorice las gramáticas del punto anterior. Construya la TAS correspondiente a cada una.**
2. **Construya la TAS para cada una de las siguientes gramáticas:**

1. V = { A, B, C, D, E }

T = { **a, b, c, &, $, #, @** }

P = { A  AB | CA | **&**A | D

B  **$**A| **#**A | E

C  A**@** | 

D  D**a** | D**b** | **a** | D | E

E  **c**E**c** | CE

}

S = A

2. V = { S, A, B, C, D, E }

T = { **a, op, t, b, c, (, )** }

P = { S S**a**S | BC | B**b**B | C

B  B | B**a** | E | **(**B**)** | **a**A**b**

C  **op** | **t** | **(**S**)** | D | C**d**C | C**c**C

D  **a**D | D**b** | **b** | B

E  E**b** | S**a**B | **ab**A

}

S = S

3. V = { S, B, E, A, X, C, D, F, G }

T = { **a, o, d, g, +, [, ]** }

S = S;

P = { S  SBC | AX | AFS

B  B**a** | E | **a**B | **a**

E  EA | EC | G

A  SD | SF | EFS | ****

X  C | FC

G  EG | E

C  **o** | **[**S**]** | E | S

D  **+** }

4. V = { S, B, E, A, H, V, X, C, D, G };

T = { **a, o, c, d, g, +, #, @** };

S = S;

P = { S  SVS | B | CES | AX | S**o**C

B  B**c** | E | **d**B | **c**

E  EA | E**d** | GE

A  S**o**AS | **d**H | AC | **d**E

V  **o**

X  X**a** | 

G  EG | **g**

C  **+** | X**#**S**@** }

5. V = { S, A, B, C, D, E, G }

T = { **a, op, t, b, c, (, )** }

P = { S  S**a**S | BC | CG | C

B  B | B**a** | E | **(** B **)**

C  **op** | **(** S **)** | D | B **b** C | B **c** C

D  DD | D**t** | **c** | B

E  E**b** | S**a**B | E**ab**A

G  **b**S| **

}

S = S

6. V = { A, G, T, C, X, F, H, V, F };

T = { **a, #, $, b, d, g, h, k, v** };

S = A;

P = { A  F**#**G | G**$**A | H | VH

G  G**b** | **d** |G**d** | A

C  **a**VCV | T**b** | X**v**H

J  VA | XF | XFF

H  H**h** | H**k** | TV | **h** | A | **g**F

V  **** | VCV | XV

F  F**g** | **g**F | FF | ****}

7. V = { A, G, T, C, X, F, H, V, F };

T = { **a, #, $, b, d, g, v** };

S = A;

P = { A  F**#**G | G**$**A | H | VH

G  G**b** | **g**J | **d** |G**d** | A

C  **a**VCV | T**b** | X**v**H

F  VA | XF | XFF

H  H**d** | H**b** | TV | **d** | A | **g**F

V  **** | VCV | XV

J  J**g** | **g**J | JJ | ****}

8.

F     FD |  QM |  M  
  D    QbF  | cF  |  cM  
  Q      | P  
  M    Mb  | dM | , | M |  P | R  
  P    hP | Pk | P

R   #R | % | ,

9.

S  CT | FG | H

F  FaGF | FbF | K

C Bc | aB | GSG

D Da | Ba | Dc | aEa

K  K# | /K | c | d

G  KB | D/ | baBS | 

E KE | D/

H  H! | !

T :C | DT

10.

S  K,A | T | T!K

T  FaF | TXT | F | TcF

B  Ba | Bc | a | b | c

X  a | Wb | c

W  EaW|  | WaE

F  FF | -F+ | H | FH | +#

K  SE | SW

H  + | -W | #

A  EA | WTW

E  WE | -SZ | EE

11.

X  XF | KM | CB | MX**c**X

E  E**b**G| G**a**X | **a**E**b**

M   | MG

C  C**r** | M**r** | **a**B | **u**C

D  **a**D | **b**D | **a**

K  **(** X **)** | CM

B  B | B**a** | E | **(** B **)**

F FB | **a**L | 