UNIDAD Nº 2 GRAMÁTICAS Y LENGUAJES FORMALES

Ejercicios propuestos de gramática limpia

Para cada una de las siguientes gramáticas obtener una gramática limpia equivalente (su definición formal) indicando: reglas innecesarias, símbolos inaccesibles terminales y no terminales y símbolos superfluos, si los hubiera.

<u>Ejercicio 12</u>: G1 = ({ 0, 1, 2, 3 }, { S, A, B, C, D, E }, S, P1)
P1 = {S:= 0A | 1B | 01, A:= A | 1B | 0, B:= 0C | 0E | 10, C:= 1, E:= 0E, D:=0A|1B|0}

Reglas innecesarias: A:=A (recordar que son reglas donde está el mismo símbolo no terminal a ambos lados de :=)

Símbolos terminales inaccesibles: 2,3
Símbolos No terminales inaccesibles: D
(son aquellos símbolos que si bien están en los alfabetos nunca intervienen en ninguna derivación, ya que no están en las producciones)

Símbolos superfluos: E (son símbolos que no permiten generar una cadena solo de terminales o vacía)

Para los símbolos no terminales inaccesibles y superfluos se puede confeccionar las tablas que se muestran en la bibliografía de la cátedra o corroborar revisando el listado de producciones directamente.

Gramática Limpia

Se define nuevamente la gramática eliminando de los alfabetos los símbolos inaccesibles y superfluos y las producciones que contienen a los símbolos no terminales superfluos:

$G_{1(Limpia)}$ = ({ 0, 1 }, { S, A, B, C, }, S, P_{1L})
P_{1L} = {S:= 0A | 1B | 01, A:= 1B | 0, B:= 0C | 10, C:= 1}

UNIDAD Nº 2 GRAMÁTICAS Y LENGUAJES FORMALES

Ejercicio 12: G1 = ({ 0, 1, 2, 3 }, { S, A, B, C, D, E }, S,P1)

P1 =

{S:= 0A
S:= 1B
S:= 01
A:= A
A:= 1B
A:= 0
B:= 0C
B:= 0E
B:= 10
C:= 1
E:= 0E
D:= 0A
D:= 1B
D:= 0}

REGLAS INNECESARIAS

Son producciones que tienen el mismo símbolo no terminal a ambos lados y no están acompañados de otros símbolos . .

SÍMBOLOS INACCESIBLES TERMINALES

Si bien los terminales están incluidos en el alfabeto, en las producciones no se los menciona.

SÍMBOLOS INACCESIBLES NO TERMINALES

En las partes derechas de las producciones no se menciona al no terminal en cuestión.

SÍMBOLOS SUPERFLUOS

Son aquellos no terminales que en el proceso de derivación, no es posible eliminarlos:

S → 0A → 01B → 010E → 0100E → 01000E y así seguiríamos sucesivamente. Se elimina el símbolo del alfabeto y las producciones que lo contienen tanto en la parte izquierda como derecha.

UNIDAD Nº 2 GRAMÁTICAS Y LENGUAJES FORMALES

Aplique el procedimiento para limpiar la gramática en los siguientes grupos:

Ejercicio 13:

G2 = ({ 0, 1, 2 }, { S, A, B, C }, S, P2)

P2= { S:= 0A|1|S, A:= 1B0|01, C:= 0|1B|1|C, B:= 1A|A0|1B }

SOLUCIÓN: G_{13L}= ({0, 1}, {S, A, B}, S, P_{13L})

P_{13L}={S:=0A|1 , A :=1B0|01 , B:=1A|A0|1B}

Ejercicio 14:

G3 = ({ a, b, c }, { S, A, B, C, D, E }, S, P3)

P3 = { S:= aBb | λ, A:= bB | Ca, B:= bA | b | a | bE | B, C:= a | bB | aD, D:= a, E:= aE | E}

SOLUCIÓN: G_{14L}= ({a, b}, {S, A, B, C, D}, S, P_{14L})

P_{14L} = {S:=aBb | λ, A:= bB | Ca, B:= bA | b | a , C:= a | bB | aD, D:= a}

Ejercicio 15:

G4 = ({ 0, 1, 2 }, { Q, R, S, T }, Q, P4)

P4 = { Q:= 1R0 | λ, R:= 0S1 | 0T | 1, T:= 0R | RT1, S:= 0 }

SOLUCIÓN: G_{15L}= ({ 0, 1}, {Q, R, S, T}, Q, P_{15L})

P_{15L} = {Q:= 1R0 | λ, R:= 0S1 | 0T | 1, T:= 0R | RT1, S:= 0}

UNIDAD Nº 2 GRAMÁTICAS Y LENGUAJES FORMALES

Ejercicios propuestos de gramática bien formada.

Para cada una de las siguientes gramáticas, generar la gramática bien formada equivalente, eliminando, si las hay, r*eglas no generativas* y r*eglas de redenominación.* (recuerde que primero deben estar limpias).

<u>Ejercicio 16</u>: G1 = ({ 0, 1 }, { S, A, B }, S, P1)   P1= {S:= AB1 |λ, A:= BA| λ, B:= 0A|λ }

La gramática ya esta limpia

<u>REGLAS NO GENERATIVAS</u>

Son producciones donde su lado derecho es de menor longitud que el lado izquierdo (regla compresora). Por ejemplo A:= λ ; para eliminarla se debe reemplazar en todas las que contengan en la parte derecha al no terminal “A” por λ

→ S:= B1
→ A:= B
→ B:= 0

Entonces las producciones quedan: P1’={S:= AB1 |λ | B1, A:= BA|B , B:= 0A|λ | 0 }

Para B:= λ → S:= A1
→ A:= A Se elimina por regla innecesaria
→ S:= 1
→ A:= λ Se debe volver a comprobar si hay alguna producción para reemplazar y luego eliminarla.

Entonces las producciones quedan: P1’’={S:= AB1 |λ | B1 | A1 | 1, A:= BA|B , B:= 0A | 0 }

UNIDAD Nº 2 GRAMÁTICAS Y LENGUAJES FORMALES

REGLAS DE REDENOMINACIÓN

Son producciones donde hay un solo símbolo no terminal diferente a cada lado del símbolo ":="
Por ejemplo la regla A:=B es de redenominación; su lectura dice que el no terminal A puede ser reescrito como el no terminal B o también que el no terminal A debe producir todo lo que produce el no terminal B, entonces si B:= 0A | 0, se agregan las producciones:

A:= 0A
A:= 0

Entonces las producciones quedan:

P1'''={S:= AB1 |λ | B1 | A1 | 1, A:= BA|0A | 0 , B:= 0A | 0 }

Y la Gramática bien formada queda:

G1 = ({ 0, 1 }, { S, A, B }, S, P1''')
P1'''={S:= AB1 |λ | B1 | A1 | 1, A:= BA|0A | 0 , B:= 0A | 0 }

Resuelva:

Ejercicio 17:

G2 = ({ 0,1, 2, 3 }, {S, A, B, C, D }, S, P2)
P2 = {S := C0 | λ | D10, A:= 1C3, B:= B, C:= 1 | λ | 0, D:= 1D}

Ejercicio 18:

G3 = ({ a, b, c, d }, { A, B, C, D }, A, P3)
P3 = {A:= bBa, B:=bDa | aC | b | λ, C:= BB | A, D:= λ | a | b}

Soluciones a los ejercicios propuestos gramática bien formada

Ejercicio 17

G_2 = ({ 0,1, 2, 3 }, {S, A, B, C, D }, S, P_2)

P_2 = {S:= C0 | λ | D10, A:= 1C3, B:= B, C:= 1 | λ | 0, D:= 1D}

Reglas innecesarias: B:=B

Símbolos inaccesibles No terminales: A, B

Símbolos inaccesibles terminales: 2,3

Símbolos superfluos: D

Gramática limpia:

G_{2L} = ({0,1}, {S, C}, S, P_{2L})

P_{2L} = {S:= C0 | λ, C:= 1 | λ | 0}

Regla no generativa C:= λ, se elimina agregando aquellas producciones que surgen de reemplazar al no terminal "C" por su parte derecha:

P'_{2L} = {S:= C0 | λ | 0, C:= 1 | 0}

Reglas de redenominación: No hay

Gramática bien formada:

G'_{2L} = ({0,1}, {S, C}, S, P'_{2L})

P'_{2L} = {S:= C0 | λ | 0, C:= 1 | 0}

Ejercicio 18:

G_3 = ({a, b, c, d}, {A, B, C, D}, A, P_3)
P_3 = {A:= bBa, B:=bDa | aC | b | λ, C:= BB | A, D:= λ | a | b}

Reglas innecesarias: no hay
Símbolos inaccesibles No terminales: No hay
Símbolos inaccesibles terminales: c,d
Símbolos superfluos: No hay

Gramática limpia: G_{3L} = ({ a, b}, {A, B, C, D}, A, P_{3L})
P_{3L} = {A:= bBa, B:=bDa | aC | b | λ, C:= BB | A, D:= λ | a | b}

Se elimina regla no generativa B:= λ
P'_{3L} = {A:= bBa | ba, B:=bDa | aC | b , C:= BB | A | B, D:= λ | a | b}
Se elimina regla no generativa D:= λ
P''_{3L} = {A:= bBa | ba, B:=bDa | aC | b | ba, C:= BB | A | B, D:= a | b}
Se elimina regla de redenominación C:=A
P'''_{3L} = {A:= bBa | ba, B:=bDa | aC | b | ba, C:= BB | bBa | ba | B, D:= a | b}
Se elimina regla de redenominación C:=B
P''''_{3L} = {A:= bBa | ba, B:=bDa | aC | b | ba, C:= BB | bBa | ba | bDa | aC | b, D:= a | b}

Gramática bien formada: G_{3BF} = ({ a, b }, { A, B, C, D }, A, P''''_{3L})
P''''_{3L} = {A:= bBa | ba, B:=bDa | aC | b | ba, C:= BB | bBa | ba | bDa | aC | b , D:= a | b}

UNIDAD Nº 2 GRAMÁTICAS Y LENGUAJES FORMALES

Ejercicios propuestos de eliminación de recursividad por izquierda en un paso

<u>Ejercicio 19</u>:

G1 = ({a, b}, {P, Q, R}, P, P1)

P1 = {P:= abP | aQ, Q:= a | bR, R:= Ra | b}

Recordando el procedimiento para identificar los α_i, β_j en aquellas producciones con recursividad por izquierda: A := Aα_1 | Aα_2 | … | Aα_n | β_1 | β_2 | … | β_m con $\alpha_i, \beta_j \in (\Sigma T \cup \Sigma N)^+$

transformamos cada producción de este tipo en dos producciones agregando el no terminal X:

A := β_1X | β_2X | … | β_mX | β_1 | β_2 | … | β_m

X := α_1X | α_2X | … | α_nX | α_1 | α_2 | … | α_n

Entonces en nuestro caso la producción recursiva por izquierda es **R:= Ra** de forma que **α= a β = b**

por lo tanto la producción **R:= Ra** se reescribe como las producciones: X:= aX |a , R:= bX |b

y la gramática finalmente queda:

G1 = ({a, b}, {P, Q, R, X}, P, P1 ‘)

P1’ = {P:= abP | aQ, Q:= a | bR, R:= bX | b X:= a | aX }

Ejercicio 20:
G2 = ({a, b, c}, {S, A, B, C, D}, S, P2)
P2 = {S:= AB | c , A:= aC , B:= aD , C:= Ca | b , D:= b}

Reglas recursivas: C:= Ca |Cab se agrega el no terminal X
C:= b| bX
X:= a | ab | aX | abX
G_2 = ({a, b, c}, {S, A, B, C, D,X}, S, P'_2)
P'_2 = {S:= AB | c , A:= aC , B:= aD , C:= b| bX , X:= a | ab | aX | ab X , D:= b}

Ejercicio 21:
G3 = ({a, b, c},{S, A, B}, S, P3)
P3 = {S:= aAb, A:= aB | a | Ac, B:=c}

Regla recursiva: A:= Ac se agrega el no terminal X
A:= aB | a | aBX | aX
X:= c | cX
G_3 = ({a, b, c},{S, A, B, X}, S, P'_3)
P'_3 = {S:=aAb, A:= aB | a | aBX | aX, X:= c | cX, B:=c}

Ejercicio 22:

G4 = ({a, b},{M, N, P}, M, P4)

P4 = {M:= Ma | aP | b , N:= aP | a , P:= b | aN | Pb}

Regla recursiva M:= Ma se agrega el no terminal X

M:=aP | b| aPX| bX

X:= a | aX

Regla recursiva: P:= Pb se agrega el no terminal Y

P:= b | aN | bY | aNY

Y:= b | bY

G_4 = ({a, b},{M, N, P, X, Y}, M, P'_4)

P'_4 = { M:= aP | b| aPX| bX, X:= a | aX, N:= aP|a , P:= b|aN | bY|aNY, Y:= b|bY}

Ejercicio 23:

G5 = ({a, b},{M, P}, M, P5)

P5 = {M:= Pa | b , P:= Mb | b}

Regla recursiva M:= Ma se agrega el no terminal X

M:= b | bX

X:= a | aX

G_5 = ({a, b},{M, P, X}, M, P'_5)

P'_5 = { M:= b | bX, X:= a | aX , P:= Mb | b}

UNIDAD Nº 2 GRAMÁTICAS Y LENGUAJES FORMALES

Ejercicios propuestos de Formas Normales

Ejercicio 24:

A las cuatro gramáticas obtenidas de los ejercicios de gramática limpia, llevarlas a gramáticas bien formadas, expresarlas en **Forma Normal de Chomsky** y derivar dos palabras con la gramática antes y después de haber aplicado la FNC.

$G_{1(Limpia)}$ = ({ 0, 1 }, { S, A, B, C, }, S, P_{1L}) (EJERCICIO 12 de Gramáticas Limpias)
P_{1L} = {S:= 0A | 1B | 01, A:= 1B | 0, B:= 0C | 10, C:= 1}
Esta gramática está limpia y bien formada.

Vamos a analizar producción por producción para determinar si están en FNC; recordemos que en la FNC las producciones tienen la siguiente forma ⟶ A:= BC A:=a S:= λ con S,A,B,C $\in \Sigma_N$ y a $\in \Sigma_T$

S:=0A no está en FNC agregamos el no terminal X:= 0 y convertimos la producción a S:=XA
S:=1B no está en FNC agregamos el no terminal Y:= 1 y convertimos la producción a S:=YB
S:=01 no está en FNC convertimos S:=XY
A:=1B no está en FNC convertimos A:=YB
A:=0 está en FNC
B:= 0C no está en FNC convertimos B:= XC
B:= 10 no está en FNC convertimos B:=YX
C:= 1 está en FNC

La gramática en FNC queda:
$G_{1(FNC)}$ = ({ 0, 1 }, { S, A, B, C,X,Y }, S, $P_1´$)
$P_1´$= {S:= XA | 1B | XY, A:= YB | 0, B:= XC | YX, C:= 1, X:=0 , Y:= 1 }

ANTES
S-> 0A->01B->0110
S-> 1B->10C->101

DESPUÉS
S->XA->0A->0YB->01B->01YX->011X->0110
S->1B->1XC->10C ->101

A partir de estos grupos de gramáticas limpias, llevarlas a bien formadas y luego a la FNC:

Ejercicio 13

G_{13L}= ({ 0, 1 }, { S, A, B }, S, P_{13L})

P_{13L}={S:=0A|1 , A :=1B0|01 , B:=1A|A0|1B}

S->0A->001

S->0A->01B0->01A00->010100

G_{13FNC}= ({ 0, 1 }, { S, A, B, X, Y, Z }, S, P_{13FNC})

P_{13FNC}={S:=XA|1 , A :=ZX|XY , Z:= YB, B:=YA|AX|YB, X:=0 , Y:= 1}

S->XA->0A->0XY->00Y->001

S->XA->0A->0ZX->0YBX->01BX->01AXX->01XYXX->010YXX->0101XX->01010X->010100

Ejercicio 14

G_{14L}= ({ a, b }, { S, A, B, C, D }, S, P_{14L})

P_{14L} = {S:=aBb | λ, A:= bB | Ca, B:= bA | b | a , C:= a | bB | aD, D:= a }

S->aBb->abAb->abbBb->abbab

S->aBb->abb

G_{14FNC}= ({ a, b }, { S, A, B, C, D, X,Y }, S, P_{14FNC})

P_{14FNC} = {S:= XZ | λ, X:=a, Y:=b, Z:=BY, A:= YB | CX, B:= YA | b | a , C:= a | YB | XD, D:= a }

S->XZ->aZ->aBY->aYAY->abAY->abYBY->abbBY->abbaY->abbab

S->XZ->aZ->aBY->abY->abb

Ejercicio 15

G_{15L}= ({ 0, 1 }, { Q, R, S, T }, Q, P_{15L})

P_{15L} = { Q:= 1R0 | λ, R:= 0S1 | 0T | 1, T:= 0R | RT1, S:= 0 }

Q->1R0->10T0->100R0->10010

Q->1R0->10T0->10RT10->101T10->1010R10->1010110

G_{15FNC}= ({ 0, 1 }, { Q, R, S, T, X, Y, Z,W, V}, Q, P_{15FNC})

P_{15FNC} = { Q:= XZ | λ, X:=1, Y:=0, Z:=RY, W:=SX, V:=TX, R:= YW | YT | 1, T:= YR | RV, S:= 0 }

Q->XZ->1Z->1RY->1YWY->10WY->10SXY->100XY->1001Y->10010

Q->XZ->1Z->1RY->1YTY->10TY->10RVY->101VY->101TXY->101YRXY->1010RXY->10101XY->101011Y->
1010110

UNIDAD Nº 2 GRAMÁTICAS Y LENGUAJES FORMALES

Expresar las siguientes gramáticas en **Forma Normal de Greibach**.

Recordemos que en la FNG las producciones tienen la forma A:= aη S:=λ A,S $\in \Sigma_N$, a $\in \Sigma_T$ y $\eta \in \Sigma_N^*$

Ejercicio 25:

G_1 = ({0, 1}, { S, A, B, C}, S, P_1)

P_1 = {S:= A0 | 1 | C1, A:= 0B | 1, B:= A0 | 0, C:= 1}

Pasos:

1° Paso: Asignar un orden cualquiera a los símbolos No Terminales de la gramática: S, A, B, C

2° Paso: Separar las producciones del conjunto P en tres grupos:

Grupo 1: todas las producciones que comienzan con un símbolo Terminal A:= aα siendo α ε $(\Sigma_T \cup \Sigma_N)^*$ y si existe S:= λ .

S:= 1 , A:= 0B | 1 , B:= 0 , C:= 1

Grupo 2: producciones A_i := A_j α con α ε $(\Sigma_T \cup \Sigma_N)^+$ y con el símbolo A_i anterior A_j en el ordenamiento dado (i < j).

S:= A0 | C1

Grupo 3: producciones A_i := A_j α con α ε $(\Sigma_T \cup \Sigma_N)^+$ y con el símbolo A_i posterior A_j en el ordenamiento dado (i > j).

B:= A0

El caso i = j no puede producirse porque se ha eliminado la recursión por izquierda anteriormente.

UNIDAD Nº 2 GRAMÁTICAS Y LENGUAJES FORMALES

3° Paso: Para cada producción del **grupo 3** $A_i := A_j\alpha$ reemplazar por todos los lados derechos de las producciones de A_j .
Al terminar este proceso todas las producciones pertenecerán al grupo 1 ó 2.

B:= A0 → B:=0B0
B:= A0 → B:=10

4° Paso: Repetir el proceso anterior para las producciones del **Grupo 2.**
Al terminar este proceso todas las producciones pertenecerán al grupo 1.

S:= A0 → S:= 0B0
S:= A0 → S:= 10

S:= C1 → S:= 11

5° Paso: Para cada símbolo terminal que esté en el lado derecho de las producciones resultantes, pero no al inicio, crear un nuevo No terminal, una nueva producción para él y reemplazar todas las producciones que contengan estos símbolos terminales por nuevas producciones reemplazando ese símbolo Terminal.

B:= 0B0 creamos Y:=0 queda B:=0BY
B:= 10 queda B:=1Y
S:= 0B0 queda S:=0BY
S:= 10 queda S:=1Y
S:=11 creamos X:=1 queda S:=1X
S:= 1 está en FNG
A:= 0B está en FNG
A:= 1 está en FNG
B:= 0 está en FNG
C:= 1 está en FNG

G_1 Forma Normal de Greibach = ({0, 1}, { S, A, B, C, X, Y}, S, P_1)
P_1 = { X:= 1, Y:= 0, S:= 1 | 0BY | 1Y | 1X, A:= 0B | 1, B:= 0 | 0BY | 1Y, C:= 1}

Ejercicio 26:

G_2 = ({0, 1},{S, A, B, C, D}, S, P_2)

P_2 = {S:= DA | 1, A:= B0 | 1, B:= B1 | 0, C:= 0, D:= C0 | 01}

La gramática está limpia y bien formada pero con recursividad por izquierda: B:=B1

Se levanta la recursividad por izquierda **B:= 0 | 0X , X:= 1 | 1X**

la gramática queda entonces: G_2 = ({0, 1},{S, A, B, C, D, X}, S, P'_2)

P'_2 = {S:= DA | 1, A:= B0 | 1, B:= 0 | 0X, X:= 1 | 1X, C:= 0, D:= C0 | 01}

Pasos:

1° Paso: Asignar un orden cualquiera a los símbolos No Terminales de la gramática: S, A, B, C, D, X

2° Paso: Separar las producciones del conjunto P en tres grupos:

Grupo 1: todas las producciones que comienzan con un símbolo Terminal A:= aα siendo $\alpha \in (\Sigma_T \cup \Sigma_N)^*$ y si existe S:= λ .

S:= 1 , A:= 1 , B:= 0 | 0X , X:= 1 | 1X, C:= 0 , D:= 01

Grupo 2: producciones $A_i := A_j \alpha$ con $\alpha \in (\Sigma_T \cup \Sigma_N)^+$ y con el símbolo A_i anterior A_j en el ordenamiento dado ($i < j$).

S:= DA , A:=B0

Grupo 3: producciones $A_i := A_j \alpha$ con $\alpha \in (\Sigma_T \cup \Sigma_N)^+$ y con el símbolo A_i posterior A_j en el ordenamiento dado ($i > j$).

D:= C0

El caso i = j no puede producirse porque se ha eliminado la recursión por izquierda anteriormente.

3° Paso: Para cada producción del **grupo 3** $A_i := A_j\alpha$ reemplazar por todos los lados derechos de las producciones de A_j .

Al terminar este proceso todas las producciones pertenecerán al grupo 1 ó 2.

4° Paso: Repetir el proceso anterior para las producciones del **Grupo 2.**

Al terminar este proceso todas las producciones pertenecerán al grupo 1.

5° Paso: Para cada símbolo Terminal que esté en el lado derecho de las producciones resultantes, pero no al inicio, crear un nuevo No Terminal, una nueva producción para él y reemplazar todas las producciones que contengan estos símbolos Terminales por nuevas producciones reemplazando ese símbolo Terminal.

Para cada producción del **grupo 3**

D:=C0 no está en FNG se reemplaza "C" por su parte derecha y se aplica Y:=0

D:= 0Y

Para cada producción del **grupo 2**

S:=DA no está en FNG por lo tanto se reemplaza a "D" por sus partes derechas:

S:= C0A se reemplaza a "C" por su parte derecha S:= 00A y se agrega Y:= 0 quedando

S:= 0YA

S:= 01A se agrega Z:=1quedando

S:= 0ZA

A:=B0 no está en FNG se reemplaza "B" por sus partes derechas y se aplica Y:=0 quedando

A:= 0Y

A:= 0XY

Grupo 1

S:= 1 está en FNG

A:=1 está en FNG

B:=0 está en FNG

B:= 0X está en FNG

X:= 1 está en FNG

X:= 1X está en FNG

C:=0 está en FNG

D:=01 no está en FNG se aplica Z:=1

D:= 0Z

La gramática en FNG queda

G_2 = ({0, 1},{S, A, B, C, D, X, Y, Z }, S, P'_2)

P'_2 = {S:= 0YA |0ZA | 1, Y:= 0, Z:= 1, A:= 0Y | 0XY | 1, B:= 0 | 0X, X:= 1 | 1X, C:= 0, D:= 0Y | 0Z}

Ejercicio 27:

G_3 = ({a, b, c}, {S, A, B, C, D, E, F}, S, P_3)

P_3 = {S:= Aa | bB | bC , A:= Da | D , B:= Ba | b | a | B , C:= a | Db |bE , D:= a, E:= bF, F:= aF }

La gramática no está limpia

Reglas inncesarias: B:=B

Símbolos inaccesibles terminales: c

Símbolos inaccesibles No terminales: No hay

Símbolos superfluos: E,F

Gramática Limpia

G_3 = ({a, b}, {S, A, B, C, D }, S, P'_3)

P'_3 = {S:= Aa | bB | bC , A:= Da | D , B:= Ba | b | a , C:= a | Db , D:= a }

Esta gramática no está bien formada

se elimina regla de redenominación A:=D agregando A:=Da | a

Gramática bien formada G_3 = ({a, b}, {S, A, B, C, D }, S, P''_3)

P'_3 = {S:= Aa | bB | bC , A:= Da | a , B:= Ba | b | a , C:= a | Db , D:= a }

Esta gramática presenta recursividad por izquierda B:=Ba

Se levanta la recursividad por izquierda B:= b | a |bX| aX X:= a | aX

La gramática queda entonces: G_3 = ({a, b, c}, {S, A, B, C, D, X }, S, P''_3)

P''_3 = {S:= Aa | bB | bC , A:= Da | a , B:= b | a |bX | aX, X:= a | aX , C:= a | Db , D:= a }

La gramática queda entonces: G_3 = ({a, b, c}, {S, A, B, C, D, X }, S, P''_3)

P''_3 = {S:= Aa | bB | bC , A:= Da | a , B:= b | a |bX | aX, X:= a | aX , C:= a | Db , D:= a }

Pasos:

1° Paso: Asignar un orden cualquiera a los símbolos No Terminales de la gramática: S, A, B, C, D, X

2° Paso: Separar las producciones del conjunto P en tres grupos:

Grupo 1: todas las producciones que comienzan con un símbolo Terminal A:= aα siendo α ε $(\Sigma_T \cup \Sigma_N)^*$ y si existe S:= λ

S:= bB | bC , A:= a , B:= b |a | bX | aX , X:= a | aX, C:= a , D:= a

Grupo 2: producciones $A_i := A_j\ \alpha$ con α ε $(\Sigma_T \cup \Sigma_N)^+$ y con el símbolo A_i anterior A_j en el ordenamiento dado (i < j).

S:= Aa , A:=Da , C:= Db

Grupo 3: producciones $A_i := A_j\ \alpha$ con α ε $(\Sigma_T \cup \Sigma_N)^+$ y con el símbolo A_i posterior A_j en el ordenamiento dado (i > j).

El caso i = j no puede producirse porque se ha eliminado la recursión por izquierda anteriormente.

3° Paso: Para cada producción del **grupo 3** $A_i := A_j\alpha$ reemplazar por todos los lados derechos de las producciones de A_j .

Al terminar este proceso todas las producciones pertenecerán al grupo 1 ó 2.

4° Paso: Repetir el proceso anterior para las producciones del **Grupo 2.**

Al terminar este proceso todas las producciones pertenecerán al grupo 1.

5° Paso: Para cada símbolo terminal que esté en el lado derecho de las producciones resultantes, pero no al inicio, crear un nuevo No terminal, una nueva producción para él y reemplazar todas las producciones que contengan estos símbolos terminales por nuevas producciones reemplazando ese símbolo terminal.

Para cada producción del **grupo 3** **No hay**

Para cada producción del **grupo 2**

S:= Aa no está en FNG entonces se reemplaza a “A” por sus partes derechas:

S:= Daa aún no está en FNG se reemplaza a “D” por su parte derecha S:= aaa y aplicando Y:= a

S:= aYY

S:= aa aún no está en FNG se agrega Y:= a

S:= aY

A:= Da no está en FNG se reemplaza a “D” por su parte derecha y se aplica Y:= a

A:= aY

C:= Db no está en FNG se reemplaza a “D” por su parte derecha y se agrega Z:= b

C:=aZ

grupo 1

S:=bB está en FNG

S:=bC está en FNG

B:= bX está en FNG

B:= aX está en FNG

X:= a está en FNG

X:= aX está en FNG

A:= a está en FNG

C:= a está en FNG

D:= a está en FNG

Entonces la gramática en FNG queda:

G3 = ({a, b, c}, {S, A, B, C, D, X, Y, Z }, S, P’’3)

P’’3 = {S:= aYY |aY |bB | bC , A:= aY | a ,B:= b | a |bX | aX , X:= a | aX , Y:= a, Z:= b, C:= a |aZ , D:= a }

UNIDAD Nº 2 GRAMÁTICAS Y LENGUAJES FORMALES

Ejercicios propuestos de expresiones regulares

Ejercicio 28

Construya ocho cadenas correspondientes a cada uno de los lenguajes regulares representados por:

a) L_1 ((11+0)*) El lenguaje que denota la expresión regular puede determinarse aplicando la definición paso por paso:

$L((11+0)^*) = [L((11+0))]^*$ (por ser una expresión regular que denota al lenguaje formado por la estrella de kleene del lenguaje denotado)

$= [L(11+0)]^*$ (por ser una expresión regular que denota al mismo lenguaje denotado)

$= [L(11) \cup L(0)]^*$ (por ser una expresión regular que denota al lenguaje unión de los lenguajes denotados)

$= [(L(1) \,.\, L(1)) \cup L(0)]^*$ (por ser una expresión regular que denota al lenguaje concatenación de los lenguajes denotados)

$= [\{1\} \,.\, \{1\} \cup \{0\}]^*$ (por ser una expresión regular que denota al lenguaje cuya única palabra es la de largo unitario)

$= [\{11,0\}]^*$ (por concatenación y unión)

$= \{11,0\}^0 \cup \{11,0\}^1 \cup \{11,0\}^2 \cup \{11,0\}^3 \cup \ldots\ldots\ldots$ (por estrella de kleene)

$= \{\lambda, 11, 0, 110, 011, 11110, 0011, 1111110, 000011, \ldots\ldots\}$ (por unión)

UNIDAD Nº 2 GRAMÁTICAS Y LENGUAJES FORMALES

b) L_2 ((a+bb)*+ab)

$$
\begin{aligned}
L_2\ ((a+bb)^*+ab) &= [L((a+bb))]^* \cup L(ab) \\
&= [L(a+bb)]^* \cup ((L(a).\ L(b)) \\
&= [L(a) \cup L(bb)]^* \cup \{a\}.\{b\} \\
&= [L(a) \cup (L(b)\ .\ L(b))]^* \cup \{ab\} \\
&= [\{a\} \cup \{b\}\ .\ \{b\}]^* \cup \{ab\} \\
&= [\{a,bb\}]\ ^* \cup \{ab\} \\
&= \{a,bb\}^0 \cup \{a,bb\}^1 \cup \{\ a,bb\ \}^2 \cup \{\ a,bb\ \}^3 \cup \ldots\ldots\ldots\ \cup \{ab\} \\
&= \{\lambda,a,\ bb,\ aa,\ abb,bba,bbbb,\ \ldots..\ \} \cup \{ab\} \\
&= \{ab,\lambda,a,\ bb,\ aa,\ abb,\ bba,\ bbbb,\ \ldots..\ \}
\end{aligned}
$$

<u>Ejercicio 29</u>:

Determine una expresión regular para cada uno de los siguientes conjuntos palabras:

a) Cadenas de bits que empiezan con 1 y terminan con 0 (números binarios pares)

$1(0+1)^*0$

b) Cadenas de bits que se expresen como la unidad seguida de ceros (potencias de dos escritas en sistema binario)

$1(0)^*$