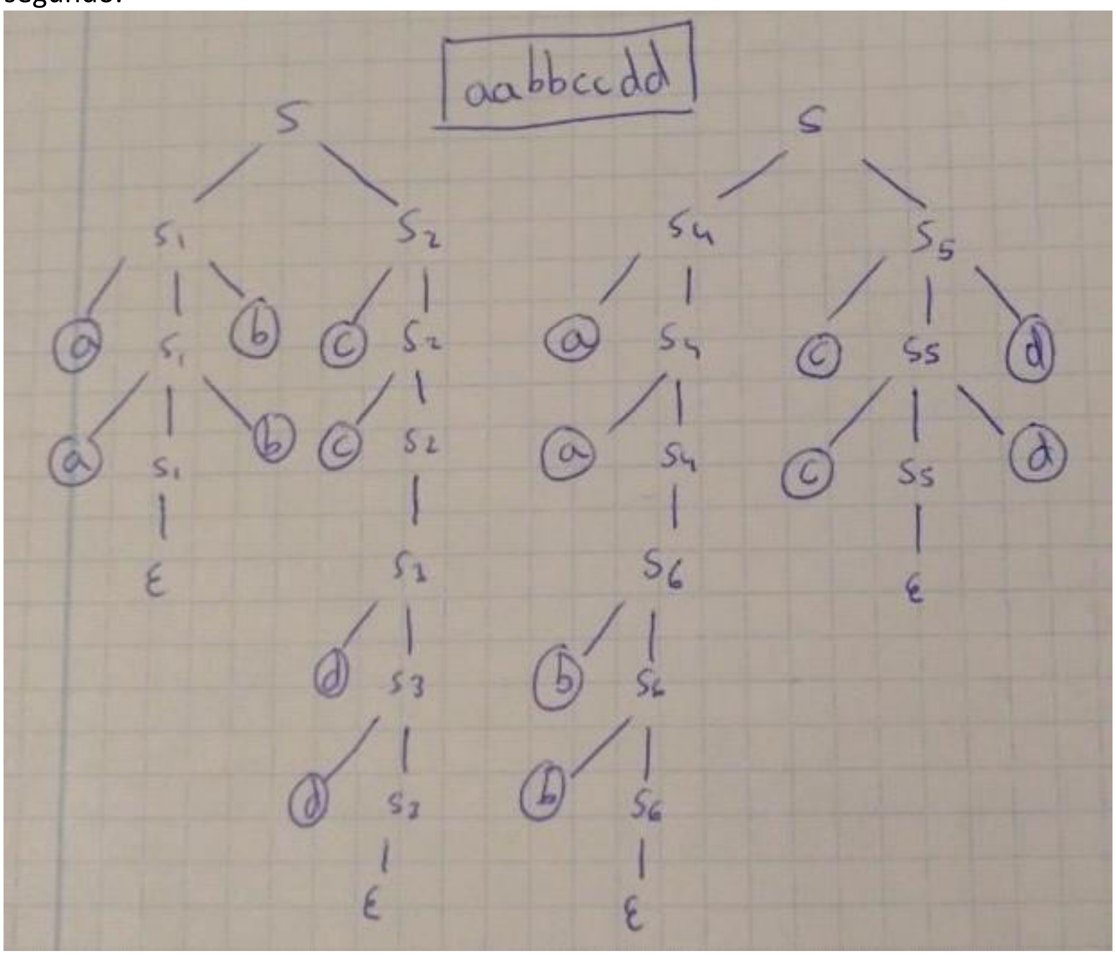
# David Infante Casas 3º C - C1
# Modelos de Computación
# Relación ejercicios 2

-Todos los ejercicios han sido hechos a papel, y, posteriormente, pasados a ordenador-

## 1.- Demuestra que la siguiente gramática libre de contexto es ambigua.

Para demostrar que la gramática es ambigua usaremos como ejemplo la palabra "aabbccdd". Obtendremos dicha palabra usando la derivación de S->$S_1S_2$ en el primer árbol y S->$S_4S_5$ en el segundo.




--------------------------------------------------------------------------------

a) L = { $a^ib^jc^kd^l$ | (i=j) OR (k=l) }

--------------------------------------------------------------------------------

b) Como no he sabido pensar la gramática no ambigua directamente, he pensado en primero construir un autómata con pila que produjese cadenas aceptadas por el lenguaje, quedando así:

El autómata sigue el siguiente razonamiento:
Cuando leo a, meto X en la pila.
Cuando leo b, quito X de la pila.
Cuando leo c, meto X en la pila.
Cuando leo d, quito X de la pila.

Estados = {q1}
Alfabeto de entrada = {a, b, c, d}
Alfabeto de pila = {R, X}
Funcion de transición = δ
Estado inicial = {q1}
Símbolo inicial de la pila = {R}
Estados finales = {Ø}

1- δ(q1, a, R) = {(q1, XR)}
2- δ(q1, b, R) = {(q1, R)}
3- δ(q1, c, R) = {(q1, XR)}
4- δ(q1, a, X) = {(q1, XX)}
5- δ(q1, b, X) = {(q1, ε)}
6- δ(q1, c, X) = {(q1, XX)}
7- δ(q1, ε, R) = {(q1, ε)}
8- δ(q1, d, X) = {(q1, ε)}
9- δ(q1, ε, X) = {(q1, ε)}

S->[q1,R,q1]
1- [q1,R,q1]->a[q1,X,q1][q1,R,q1]
2- [q1,R,q1]->b[q1,R,q1]
3- [q1,R,q1]->c[q1,X,q1][q1,R,q1]
4- [q1,X,q1]->a[q1,X,q1][q1,X,q1]
5- [q1,X,q1]->b
6- [q1,X,q1]->c[q1,X,q1][q1,X,q1]
7- [q1,R,q1]->ε
8- [q1,X,q1]->d
9- [q1,X,q1]->ε

Damos alias para que sea más legible:

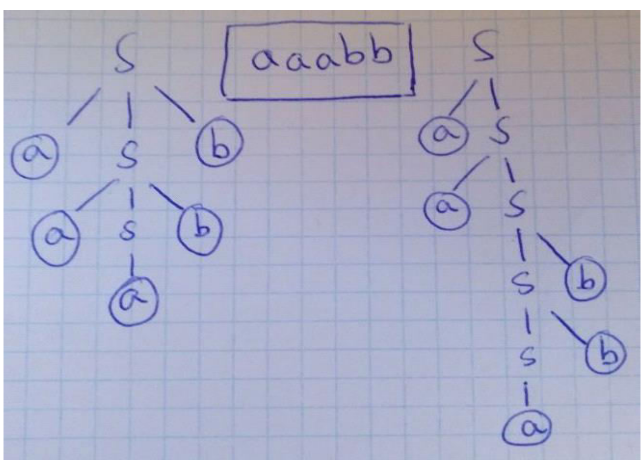
S = S
A = [q1,R,q1]
B = [q1,X,q1]

(Quitando producciones y símbolos inútiles):
S->bS|aAS|cAS|ε
A->aAA|cAA|b|d|ε

Probamos por ejemplo la cadena aabbc:
S->
aAS->
aaAAS->
aabAS->
aabbS->
aabbcAS->
aabbc

## 2. - Determinar cuáles de las siguientes gramáticas son ambiguas y, en su caso, comprobar si los lenguajes generados son inherentemente ambiguos:

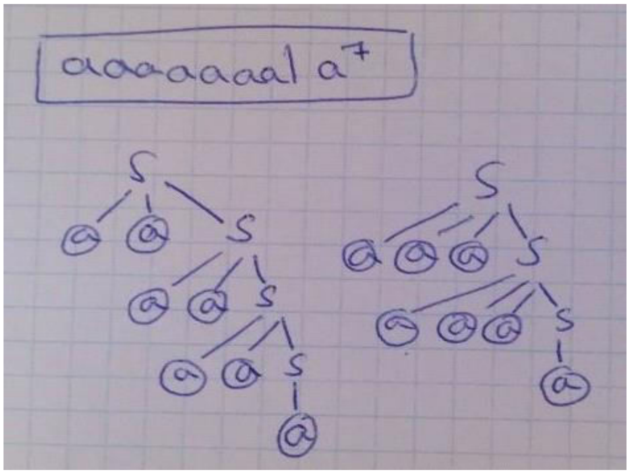
a) Es ambigua.




Genera el lenguaje:
L = { $a^ib^j$ | i > 0; j >= 0 }
No es inherentemente ambigua porque la siguiente gramática genera el mismo lenguaje y no es ambigua:
S -> aS | Sb | a

--------------------------------------------------------------------------------

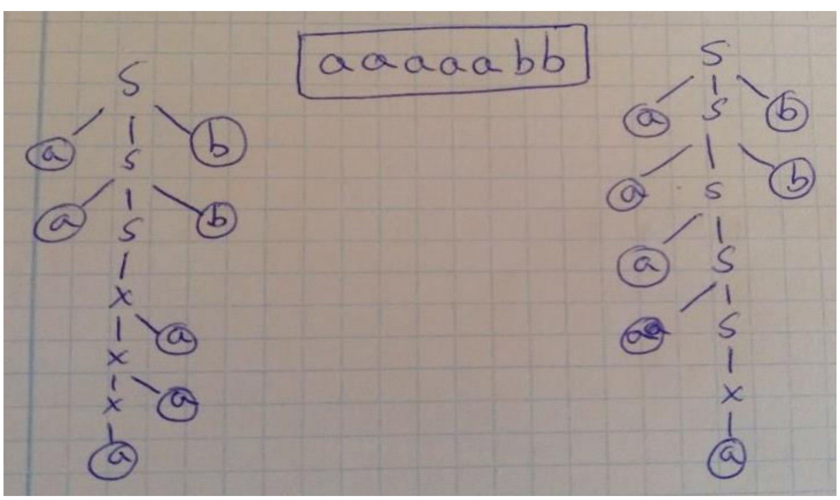
b) Es ambigua.




Genera el lenguaje:
L = { $a^i$ | i > 0 AND i != 2 }
No es inherentemente ambigua porque la siguiente gramática genera el mismo lenguaje y no es ambigua:
S -> $aS_1$ | a
$S_1$ -> $aS_2$
$S_2$ -> $aS_2$ | a

--------------------------------------------------------------------------------

c) Es ambigua.




Genera el lenguaje:
L = { $a^ib^j$ | i > j AND i > 0 }
No es inherentemente ambigua porque la siguiente gramática genera el mismo lenguaje y no es ambigua:

S -> aS | aSb | a

## 3. - Pasa a Forma Normal de Chomsky la siguiente gramática libre de contexto:

Primero haremos uso de los algoritmos de eliminación de símbolos y producciones inútiles, eliminación de producciones nulas y eliminación de producciones unitarias:

Paso 1

Eliminamos producciones inútiles:

$V \backslash V_t$ = {E}

S -> A|BCa|aDcd

A -> aAb|c

B -> CD|Ad|ε

C -> Cc|Bb|c

D -> aDd|Dd|ε

F -> aFd|d

Paso 2

Se elimina F porque es inalcanzable desde S

S -> A|BCa|aDcd

A -> aAb|c

B -> CD|Ad|ε

C -> Cc|Bb|c

D -> aDd|Dd|ε

Paso 3

Eliminamos producciones nulas:

H = {B, D}

S -> A|BCa|Ca|aDcd|acd

A -> aAb|c

B -> CD|C|Ad

C -> Cc|Bb|b|c

D -> aDd|ad|Dd|d

Paso 4

Eliminamos producciones unitarias:

S -> aAb|c|BCa|Ca|aDcd|acd

A -> aAb|c

B -> CD|Cc|Bb|b|c|Ad

C -> Cc|Bb|b|c

D -> aDd|ad|Dd|d

Paso 5

Forma Normal de Chomsky:

$S \rightarrow C_aE_1|c|BE_2|CC_a|C_aE_3|C_aE_4$

$A \rightarrow C_aE_1|c$

$B \rightarrow CD|CC_c|BC_b|b|c|AC_d$

$C \rightarrow CC_c|BC_b|b|c$

$D \rightarrow C_aF_1|C_aC_d|DC_d|d$

$E_1 \rightarrow AC_b$

$E_2 \rightarrow CC_a$

$E_3 \rightarrow DE_4$

$E_4 \rightarrow C_cC_d$

$F_1 \rightarrow DC_d$

$C_a \rightarrow a$

$C_b \rightarrow b$

$C_c \rightarrow c$

$C_d \rightarrow d$

## 4. - Pasa a Forma Normal de Greibach la siguiente gramática:

Para poder aplicar Greibach primero pasaremos la gramática a la forma normal de Chomsky:

Quitamos las producciones unitarias:

$S_1 \rightarrow S_1S_2c|S_2e|S_3bS_3$

$S_2 \rightarrow S_1S_1|d$

$S_3 \rightarrow S_2e$

Forma Normal de Chomsky:

$S_1 \rightarrow S_1D_1|S_2C_e|S_3D_2$

$S_2 \rightarrow S_1S_1|d$

$S_3 \rightarrow S_2C_e$

$D_1 \rightarrow S_2C_c$

$D_2 \rightarrow C_bS_3$

$C_b \rightarrow b$

$C_c \rightarrow c$

$C_e \rightarrow e$

Asignamos nombres a las variables para tener que hacer el mínimo de Elimina1 posibles:

$C_e > C_c > D_2 > D_1 > C_b > S_3 > S_1 > S_2$

$C_e$ = A1

$C_c$ = A2

$D_2$ = A3

$D_1$ = A4

$C_b$ = A5

$S_3$ = A6

$S_1$ = A7

$S_2$ = A8

**Paso 1 de Greibach:**

| | | | | | |
|---|---|---|---|---|---|
| A7->A7A4 | A7->A8A1 | A7->A6A3 | A8->A7A7 | | A8->d |
| A6->A8A1 | A4->A8A2 | A3->A5A6 | A5->b | A2->c | A1->e |

Elimina1:

Se elimina: A7->A6A3

Se añade: A7->A8A1A3

Queda:

| | | | | | |
|---|---|---|---|---|---|
| A7->A7A4 | A7->A8A1 | A7->A8A1A3 | A8->A7A7 | | A8->d |
| A6->A8A1 | A4->A8A2 | A3->A5A6 | A5->b | A2->c | A1->e |

Elimina2:

Se añade B7 y las producciones: B7->A4 B7->A4B7

Se elimina: A7->A7A4

Se añade: A7->A8A1B7 A7->A8A1B7

Queda:

| | | | | | |
|---|---|---|---|---|---|
| A7->A8A1 | A7->A8A1A3 | A8->A7A7 | A8->d | | |
| A6->A8A1 | A4->A8A2 | A3->A5A6 | A5->b | A2->c | A1->e |
| B7->A4 | B7->A4B7 | A7->A8A1B7 | A7->A8A1A3B7 | | |

Elimina1:

Se elimina: A8->A7A7

Se añade: A8->A8A1A7 A8->A8A1A3A7 A8->A8A1B7A7

A8->A8A1A3B7A7

Queda:

| | | | | | |
|---|---|---|---|---|---|
| A7->A8A1 | A7->A8A1A3 | A8->d | | | |
| A6->A8A1 | A4->A8A2 | A3->A5A6 | A5->b | A2->c | A1->e |
| B7->A4 | B7->A4B7 | A7->A8A1B7 | A7->A8A1A3B7 | | |
| A8->A8A1A7 | A8->A8A1A3A7 | A8->A8A1B7A7 | A8->A8A1A3B7A7 | | |

Elimina2:

Se añade B8 y las producciones: B8->A1A7 B8->A1A7B8

B8->A1A3A7 B8->A1A3A7B8

B8->A1B7A7 B8->A1B7A7B8

B8->A1A3B7A7 B8->A1A3B7A7B8

Se eliminan: A8->A8A1A7 A8->A8A1A3A7 A8->A8A1B7A7

A8->A8A1A3B7A7

Se añade: A8->dB8

Queda:

A7->A8A1 A7->A8A1A3 A8->d

A6->A8A1 A4->A8A2 A3->A5A6 A5->b A2->c A1->e

B7->A4 B7->A4B7 A7->A8A1B7 A7->A8A1A3B7

B8->A1A7 B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8

B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8

A8->dB8

**Paso 2 de Greibach:**

A7->A8A1 A7->A8A1A3 A8->d

A6->A8A1 A4->A8A2 A3->A5A6 A5->b A2->c A1->e

B7->A4 B7->A4B7 A7->A8A1B7 A7->A8A1A3B7

B8->A1A7 B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8

B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8

A8->dB8

Elimina1:

Se elimina: A7->A8A1

Se añade: A7->dA1 A7->dB8A1

Queda:

A7->A8A1A3 A8->d A6->A8A1 A4->A8A2 A3->A5A6 A5->b A2->c A1->e

B7->A4 B7->A4B7 A7->A8A1B7 A7->A8A1A3B7

B8->A1A7 B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8

B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8

A8->dB8 A7->dA1 A7->dB8A1

Elimina1:

Se elimina: A7->A8A1A3

Se añade: A7->dA1A3 A7->dB8A1A3

Queda:

A8->d A6->A8A1 A4->A8A2 A3->A5A6 A5->b A2->c A1->e

B7->A4 B7->A4B7 A7->A8A1B7 A7->A8A1A3B7
B8->A1A7 B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8
B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3

Elimina1:
Se elimina: A6->A8A1
Se añade: A6->dA1 A6->dB8A1
Queda:
A8->d A4->A8A2 A3->A5A6 A5->b A2->c A1->e
B7->A4 B7->A4B7 A7->A8A1B7 A7->A8A1A3B7
B8->A1A7 B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8
B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1

Elimina1:
Se elimina: A4->A8A2
Se añade: A4->dA2 A4->dB8A2
Queda:
A8->d A3->A5A6 A5->b A2->c A1->e
B7->A4 B7->A4B7 A7->A8A1B7 A7->A8A1A3B7
B8->A1A7 B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8
B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1 A4->dA2 A4->dB8A2

Elimina1:
Se elimina: A3->A5A6
Se añade: A3->bA6
Queda:
A8->d A5->b A2->c A1->e
B7->A4 B7->A4B7 A7->A8A1B7 A7->A8A1A3B7
B8->A1A7 B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8
B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1 A4->dA2 A4->dB8A2 A3->bA6

Elimina1:

Se elimina: A7->A8A1B7
Se añade: A7->dA1B7 A7->dB8A1B7
Queda:
A8->d A5->b A2->c A1->e
B7->A4 B7->A4B7 A7->A8A1A3B7
B8->A1A7 B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8
B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1 A4->dA2 A4->dB8A2 A3->bA6
A7->dA1B7 A7->dB8A1B7

Elimina1:
Se elimina: A7->A8A1A3B7
Se añade: A7->dA1A3B7 A7->dB8A1A3B7
Queda:
A8->d A5->b A2->c A1->e
B7->A4 B7->A4B7
B8->A1A7 B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8
B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1 A4->dA2 A4->dB8A2 A3->bA6
A7->dA1B7 A7->dB8A1B7 A7->dA1A3B7 A7->dB8A1A3B7

-------------------------------------------------------------------

Marcamos las producciones que restan:
A8->d A5->b A2->c A1->e B7->A4 B7->A4B7
B8->A1A7 B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8
B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1 A4->dA2 A4->dB8A2 A3->bA6
A7->dA1B7 A7->dB8A1B7 A7->dA1A3B7 A7->dB8A1A3B7

Elimina1:
Se elimina: B7->A4
Se añade: B7->dA2 B7->dB8A2
Queda:
A8->d A5->b A2->c A1->e B7->A4B7
B8->A1A7 B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8
B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8

A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1 A4->dA2 A4->dB8A2 A3->bA6
A7->dA1B7 A7->dB8A1B7 A7->dA1A3B7 A7->dB8A1A3B7
B7->dA2 B7->dB8A2

Elimina1:
Se elimina: B7->A4B7
Se añade: B7->dA2B7 B7->dB8A2B7
Queda:
A8->d A5->b A2->c A1->e
B8->A1A7 B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8
B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1 A4->dA2 A4->dB8A2 A3->bA6
A7->dA1B7 A7->dB8A1B7 A7->dA1A3B7 A7->dB8A1A3B7
B7->dA2 B7->dB8A2 B7->dA2B7 B7->dB8A2B7

Elimina1:
Se elimina: B8->A1A7
Se añade: B8->eA7
Queda:
A8->d A5->b A2->c A1->e
B8->A1A7B8 B8->A1A3A7 B8->A1A3A7B8
B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1 A4->dA2 A4->dB8A2 A3->bA6
A7->dA1B7 A7->dB8A1B7 A7->dA1A3B7 A7->dB8A1A3B7
B7->dA2 B7->dB8A2 B7->dA2B7 B7->dB8A2B7
B8->eA7

Elimina1:
Se elimina: B8->A1A7B8
Se añade: B8->eA7B8
Queda:
A8->d A5->b A2->c A1->e
B8->A1A3A7 B8->A1A3A7B8
B8->A1B7A7 B8->A1B7A7B8 B8->A1A3B7A7 B8->A1A3B7A7B8
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1 A4->dA2 A4->dB8A2 A3->bA6

A7->dA1B7   A7->dB8A1B7   A7->dA1A3B7   A7->dB8A1A3B7
B7->dA2   B7->dB8A2   B7->dA2B7   B7->dB8A2B7
B8->eA7   B8->eA7B8

Elimina1:
Se elimina: B8->A1A3A7
Se añade: B8->eA3A7
Queda:
A8->d   A5->b   A2->c   A1->e
B8->A1A3A7B8
B8->A1B7A7   B8->A1B7A7B8   B8->A1A3B7A7   B8->A1A3B7A7B8
A8->dB8   A7->dA1   A7->dB8A1   A7->dA1A3   A7->dB8A1A3
A6->dA1   A6->dB8A1   A4->dA2   A4->dB8A2   A3->bA6
A7->dA1B7   A7->dB8A1B7   A7->dA1A3B7   A7->dB8A1A3B7
B7->dA2   B7->dB8A2   B7->dA2B7   B7->dB8A2B7
B8->eA7   B8->eA7B8   B8->eA3A7

Elimina1:
Se elimina: B8->A1A3A7B8
Se añade: B8->eA3A7B8
Queda:
A8->d   A5->b   A2->c   A1->e
B8->A1B7A7   B8->A1B7A7B8   B8->A1A3B7A7   B8->A1A3B7A7B8
A8->dB8   A7->dA1   A7->dB8A1   A7->dA1A3   A7->dB8A1A3
A6->dA1   A6->dB8A1   A4->dA2   A4->dB8A2   A3->bA6
A7->dA1B7   A7->dB8A1B7   A7->dA1A3B7   A7->dB8A1A3B7
B7->dA2   B7->dB8A2   B7->dA2B7   B7->dB8A2B7
B8->eA7   B8->eA7B8   B8->eA3A7   B8->eA3A7B8

Elimina1:
Se elimina: B8->A1B7A7
Se añade: B8->eB7A7
Queda:
A8->d   A5->b   A2->c   A1->e
B8->A1B7A7B8   B8->A1A3B7A7   B8->A1A3B7A7B8
A8->dB8   A7->dA1   A7->dB8A1   A7->dA1A3   A7->dB8A1A3
A6->dA1   A6->dB8A1   A4->dA2   A4->dB8A2   A3->bA6
A7->dA1B7   A7->dB8A1B7   A7->dA1A3B7   A7->dB8A1A3B7
B7->dA2   B7->dB8A2   B7->dA2B7   B7->dB8A2B7

B8->eA7 B8->eA7B8 B8->eA3A7 B8->eA3A7B8
B8->eB7A7

Elimina1:
Se elimina: B8->A1B7A7B8
Se añade: B8->eB7A7B8
Queda:
A8->d A5->b A2->c A1->e
B8->A1A3B7A7 B8->A1A3B7A7B8
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1 A4->dA2 A4->dB8A2 A3->bA6
A7->dA1B7 A7->dB8A1B7 A7->dA1A3B7 A7->dB8A1A3B7
B7->dA2 B7->dB8A2 B7->dA2B7 B7->dB8A2B7
B8->eA7 B8->eA7B8 B8->eA3A7 B8->eA3A7B8
B8->eB7A7 B8->eB7A7B8

Elimina1:
Se elimina: B8->A1A3B7A7
Se añade: B8->eA3B7A7
Queda:
A8->d A5->b A2->c A1->e
B8->A1A3B7A7B8
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1 A4->dA2 A4->dB8A2 A3->bA6
A7->dA1B7 A7->dB8A1B7 A7->dA1A3B7 A7->dB8A1A3B7
B7->dA2 B7->dB8A2 B7->dA2B7 B7->dB8A2B7
B8->eA7 B8->eA7B8 B8->eA3A7 B8->eA3A7B8
B8->eB7A7 B8->eB7A7B8 B8->eA3B7A7

Elimina1:
Se elimina: B8->A1A3B7A7B8
Se añade: B8->eA3B7A7B8
Queda:
A8->d A5->b A2->c A1->e
A8->dB8 A7->dA1 A7->dB8A1 A7->dA1A3 A7->dB8A1A3
A6->dA1 A6->dB8A1 A4->dA2 A4->dB8A2 A3->bA6
A7->dA1B7 A7->dB8A1B7 A7->dA1A3B7 A7->dB8A1A3B7
B7->dA2 B7->dB8A2 B7->dA2B7 B7->dB8A2B7
B8->eA7 B8->eA7B8 B8->eA3A7 B8->eA3A7B8

B8->eB7A7 B8->eB7A7B8 B8->eA3B7A7 B8->eA3B7A7B8

**Resultado:**

| | | | | |
|---|---|---|---|---|
| A8->d A5->b | A2->c A1->e | | | |
| A8->dB8 | A7->dA1 | A7->dB8A1 | A7->dA1A3 | A7->dB8A1A3 |
| A6->dA1 | A6->dB8A1 | A4->dA2 | A4->dB8A2 | A3->bA6 |
| A7->dA1B7 | A7->dB8A1B7 | A7->dA1A3B7 | A7->dB8A1A3B7 | |
| B7->dA2 | B7->dB8A2 | B7->dA2B7 | B7->dB8A2B7 | |
| B8->eA7 | B8->eA7B8 | B8->eA3A7 | B8->eA3A7B8 | |
| B8->eB7A7 | B8->eB7A7B8 | B8->eA3B7A7 | B8->eA3B7A7B8 | |

## 5. - Dar un autómata con pila que acepte las cadenas del siguiente lenguaje por el criterio de pila vacía: L = {aibjckdl / (i=l) ∨ (j=k)}

Para encontrar el autómata, he construido primero una gramática que genera el lenguaje:

$S \to S_1|S_4S_5S_6$

$S_1 \to aS_1d|S_2$

$S_2 \to bS_2|S_3$

$S_3 \to cS_3|\varepsilon$

$S_4 \to aS_4|\varepsilon$

$S_5 \to bS_5c|\varepsilon$

$S_6 \to dS_6|\varepsilon$

Hecho esto, paso la gramática a autómata con pila:

Estados = {q}

Alfabeto de entrada = {a, b, c, d}

Alfabeto de pila = $\{a, b, c, d, S, S_1, S_2, S_3, S_4, S_5, S_6\}$

Funcion de transición = δ

Estado inicial = {q}

Símbolo inicial de la pila = {S}

Estados finales = {Ø}

$\delta(q, \varepsilon, S) = \{(q, S_1), (q, S_4S_5S_6)\}$

$\delta(q, \varepsilon, S_1) = \{(q, aS_1d), (q, S_2)\}$

$\delta(q, \varepsilon, S_2) = \{(q, bS_2), (q, S_3)\}$

$\delta(q, \varepsilon, S_3) = \{(q, cS_3), (q, \varepsilon)\}$

$\delta(q, \varepsilon, S_4) = \{(q, aS_4), (q, \varepsilon)\}$

$\delta(q, \varepsilon, S_5) = \{(q, bS_5c), (q, \varepsilon)\}$

$\delta(q, \varepsilon, S_6) = \{(q, dS_6), (q, \varepsilon)\}$

$\delta(q, a, a) = \{(q, \varepsilon)\}$
$\delta(q, b, b) = \{(q, \varepsilon)\}$
$\delta(q, c, c) = \{(q, \varepsilon)\}$
$\delta(q, d, d) = \{(q, \varepsilon)\}$

Probamos el autómata con la cadena aaabddd, por ejemplo:
Pila -> ]

| | |
|---|---|
| aaabddd | S] |
| aaabddd | $S_1$] |
| aaabddd | a$S_1$d] |
| aaabddd | $S_1$d] |
| aaabddd | a$S_1$dd] |
| aaabddd | $S_1$dd] |
| aaabddd | a$S_1$ddd] |
| aaabddd | $S_1$ddd] |
| aaabddd | $S_2$ddd] |
| aaabddd | b$S_2$ddd] |
| aaabddd | $S_2$ddd] |
| aaabddd | $S_3$ddd] |
| aaabddd | ddd] |
| aaabddd | dd] |
| aaabddd | d] |
| aaabddd | ] |

## 6. - Dar un autómata con pila determinista que acepte las cadenas definidas sobre el alfabeto A de los siguientes lenguajes por el criterio de pila vacía, si no es posible encontrarlo por ese criterio entonces usar el criterio de estados finales:

a)El autómata sigue el siguiente razonamiento:
Cuando leo 0, meto X en la pila.
Cuando leo 1, meto X de la pila.
Cuando leo 2, meto X de la pila.
Cuando leo 3, quito X de la pila.

Estados = {q1}
Alfabeto de entrada = {0, 1, 2, 3}
Alfabeto de pila = {R, X}
Funcion de transición = δ

Estado inicial = {q1}
Símbolo inicial de la pila = {R}
Estados finales = {Ø}

δ(q1, 0, R) = {(q1, XR)} - leer 0 al empezar y meter X
δ(q1, 1, R) = {(q1, XR)} - leer 1 al empezar y meter X
δ(q1, 2, R) = {(q1, XR)} - leer 2 al empezar y meter X
δ(q1, 0, X) = {(q1, XX)} - leer 0 y meter X
δ(q1, 1, X) = {(q1, XX)} - leer 1 y meter X
δ(q1, 2, X) = {(q1, XX)} - leer 2 y meter X
δ(q1, 3, X) = {(q1, ε)} - leer 3 y quitar X
δ(q1, ε, R) = {(q1, ε)} - quitar el símbolo inicial (R)

Las tres primeras funciones de transición permiten i, j, k >= 0.

------------------------------------------------------------------------------------

b)El autómata sigue el siguiente razonamiento:
Cuando leo 0, meto X en la pila.
Cuando leo 1, meto X de la pila.
Cuando leo 2, meto X de la pila.
Cuando leo 3, quito X de la pila.
Cuando leo 4, quito R de la pila.

Estados = {q1}
Alfabeto de entrada = {0, 1, 2, 3, 4}
Alfabeto de pila = {R, X}
Funcion de transición = δ
Estado inicial = {q1}
Símbolo inicial de la pila = {R}
Estados finales = {Ø}

δ(q1, 0, R) = {(q1, XR)} - leer 0 al empezar y meter X
δ(q1, 1, R) = {(q1, XR)} - leer 1 al empezar y meter X
δ(q1, 2, R) = {(q1, XR)} - leer 2 al empezar y meter X
δ(q1, 0, X) = {(q1, XX)} - leer 0 y meter X
δ(q1, 1, X) = {(q1, XX)} - leer 1 y meter X
δ(q1, 2, X) = {(q1, XX)} - leer 2 y meter X
δ(q1, 3, X) = {(q1, ε)} - leer 3 y quitar X

δ(q1, 4, R) = {(q1, ε)} - quitar el símbolo inicial (R)

Las tres primeras funciones de transición permiten i, j, k = 0.
La última función de transición permite poner 4 siempre al final.

## 7. - Construir un autómata con pila que acepte el siguiente lenguaje:

El autómata sigue el siguiente razonamiento:
Cuando leo a, meto X en la pila.
Cuando leo b, quito X de la pila hasta llegar a R.
Si me faltan b por leer, meto X en la pila.
Cuando leo c, quito X de la pila.

Estados = {q1, q2}
Alfabeto de entrada = {a, b, c}
Alfabeto de pila = {R, X}
Funcion de transición = δ
Estado inicial = {q1}
Símbolo inicial de la pila = {R}
Estados finales = {Ø}

1- δ(q1, a, R) = {(q1, XR)} - leer a al empezar y meter X
2- δ(q1, a, X) = {(q1, XX)} - leer a y meter X
3- δ(q1, b, R) = {(q2, XR)} - leer b, si llego a R, meter X y pasar a q2
4- δ(q1, b, X) = {(q1, ε)} - leer b y quitar X
5- δ(q1, c, X) = {(q1, ε)} - leer c y quitar X
6- δ(q1, ε, R) = {(q1, ε)} - quitar estado inicial
7- δ(q2, c, X) = {(q1, ε)} - leer c, quitar X y pasar a q1
8- δ(q2, b, X) = {(q2, XX)} - leer b y meter X si no había a

-----------------------------------------------------------------------------

a) S->[q1,R,q1]            S->[q1,R,q2]

1- [q1,R,q1]->a[q1,X,q1][q1,R,q1]
[q1,R,q2]->a[q1,X,q1][q1,R,q2]
[q1,R,q1]->a[q1,X,q2][q2,R,q1]
[q1,R,q2]->a[q1,X,q2][q2,R,q2]

2- [q1,X,q1]->a[q1,X,q1][q1,X,q1]
[q1,X,q2]->a[q1,X,q1][q1,X,q2]
[q1,X,q1]->a[q1,X,q2][q2,X,q1]
[q1,X,q2]->a[q1,X,q2][q2,X,q2]

3- [q1,R,q1]->b[q2,X,q1][q1,R,q1]
[q1,R,q2]->b[q2,X,q1][q1,R,q2]
[q1,R,q1]->b[q2,X,q2][q2,R,q1]
[q1,R,q2]->b[q2,X,q2][q2,R,q2]

4- [q1,X,q1]->b

5- [q1,X,q1]->b

6- [q1,R,q1]->ε

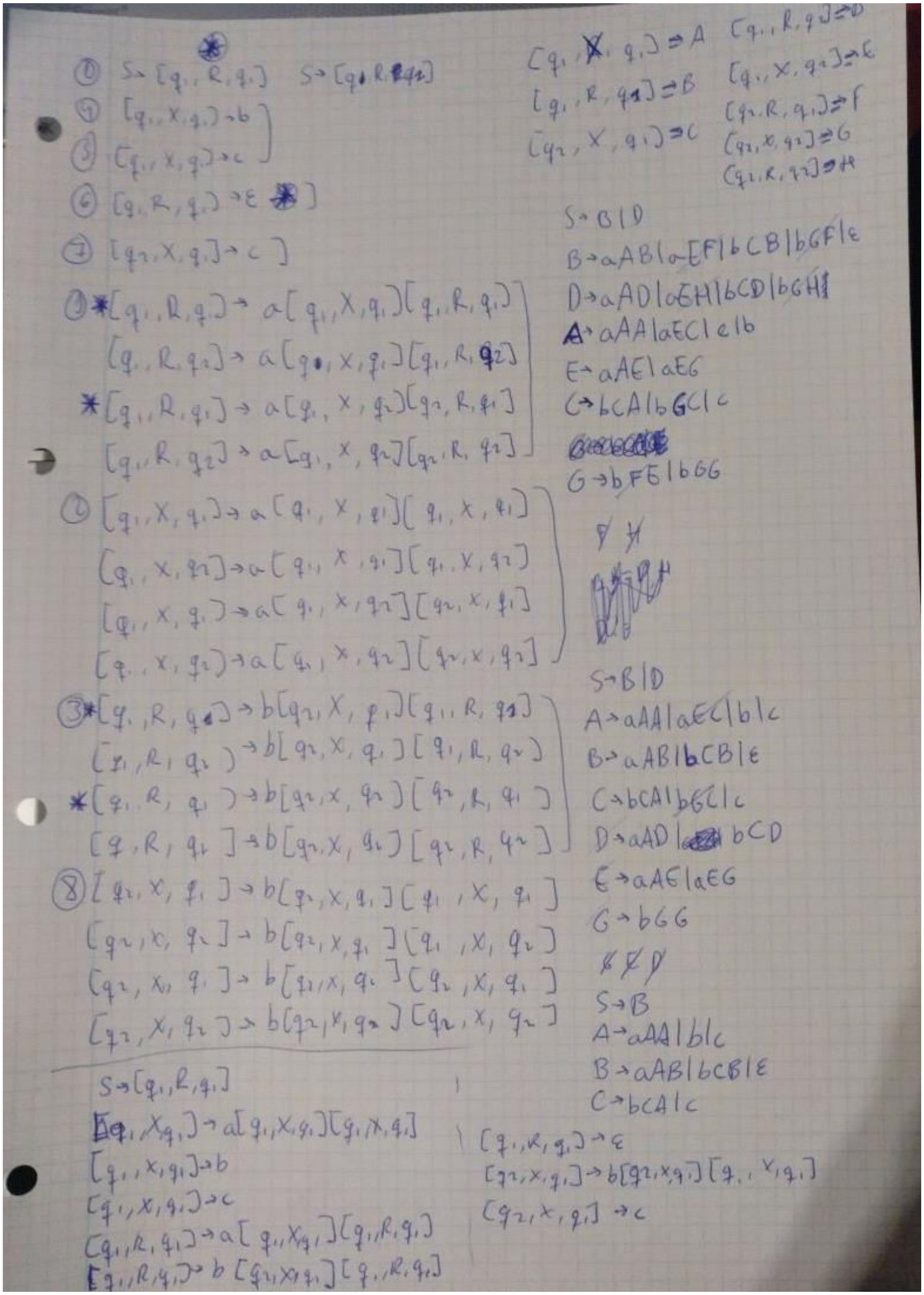
7- [q2,X,q1]->c

8- [q2,X,q1]->b[q2,X,q1][q1,X,q1]
[q2,X,q2]->b[q2,X,q1][q1,X,q2]
[q2,X,q1]->b[q2,X,q2][q2,X,q1]
[q2,X,q2]->b[q2,X,q2][q2,X,q2]

--------------------------------------------------------------------------------

b) Al eliminar producciones y símbolos inútiles, la gramática resultante es:
S->a[q1,X,q1]S
S->b[q2,X,q1]S
S->ε
[q1,X,q1]->a[q1,X,q1][q1,X,q1]
[q1,X,q1]->b
[q1,X,q1]->c
[q2,X,q1]->b[q2,X,q1][q1,X,q1]
[q2,X,q1]->c

Para realizar la simplificación de la gramática di alias a las variables para que fuese más sencillo, adjunto una imagen donde hice la simplificación:



(1) $S \to [q_1, R, q_1]$ $\quad S \to [q_1, R, q_2]$

(4) $[q_1, X, q_1] \to b$

(5) $[q_1, X, q_1] \to c$

(6) $[q_1, R, q_1] \to \varepsilon$

(7) $[q_2, X, q_1] \to c$

(1) $[q_1, R, q_1] \to a[q_1, X, q_1][q_1, R, q_1]$

$[q_1, R, q_2] \to a[q_1, X, q_1][q_1, R, q_2]$

$[q_1, R, q_1] \to a[q_1, X, q_2][q_2, R, q_1]$

$[q_1, R, q_2] \to a[q_1, X, q_2][q_2, R, q_2]$

(2) $[q_1, X, q_1] \to a[q_1, X, q_1][q_1, X, q_1]$

$[q_1, X, q_2] \to a[q_1, X, q_1][q_1, X, q_2]$

$[q_1, X, q_1] \to a[q_1, X, q_2][q_2, X, q_1]$

$[q_1, X, q_2] \to a[q_1, X, q_2][q_2, X, q_2]$

(3) $[q_1, R, q_1] \to b[q_2, X, q_1][q_1, R, q_1]$

$[q_1, R, q_2] \to b[q_2, X, q_1][q_1, R, q_2]$

$[q_1, R, q_1] \to b[q_2, X, q_2][q_2, R, q_1]$

$[q_1, R, q_2] \to b[q_2, X, q_2][q_2, R, q_2]$

(8) $[q_2, X, q_1] \to b[q_2, X, q_1][q_1, X, q_1]$

$[q_2, X, q_2] \to b[q_2, X, q_1][q_1, X, q_2]$

$[q_2, X, q_1] \to b[q_2, X, q_2][q_2, X, q_1]$

$[q_2, X, q_2] \to b[q_2, X, q_2][q_2, X, q_2]$

$[q_1, X, q_1] \Rightarrow A$ $\quad [q_1, R, q_2] \Rightarrow D$

$[q_1, R, q_1] \Rightarrow B$ $\quad [q_1, X, q_2] \Rightarrow E$

$[q_2, X, q_1] \Rightarrow C$ $\quad [q_2, R, q_1] \Rightarrow F$

$[q_2, X, q_2] \Rightarrow G$

$[q_2, R, q_2] \Rightarrow H$

$S \to B \mid D$

$B \to aAB \mid aEF \mid bCB \mid bGF \mid \varepsilon$

$D \to aAD \mid aEH \mid bCD \mid bGH$

$A \to aAA \mid aEC \mid c \mid b$

$E \to aAE \mid aEG$

$C \to bCA \mid bGC \mid c$

$G \to bFE \mid bGG$

$\cancel{F}$ $\cancel{H}$

$S \to B \mid D$

$A \to aAA \mid aEC \mid b \mid c$

$B \to aAB \mid bCB \mid \varepsilon$

$C \to bCA \mid bGC \mid c$

$D \to aAD \mid bCD$

$E \to aAE \mid aEG$

$G \to bGG$

$\cancel{E}$ $\cancel{G}$ $\cancel{D}$

$S \to B$

$A \to aAA \mid b \mid c$

$B \to aAB \mid bCB \mid \varepsilon$

$C \to bCA \mid c$

$S \to [q_1, R, q_1]$

$[q_1, X, q_1] \to a[q_1, X, q_1][q_1, X, q_1]$

$[q_1, X, q_1] \to b$

$[q_1, X, q_1] \to c$

$[q_1, R, q_1] \to a[q_1, X, q_1][q_1, R, q_1]$

$[q_1, R, q_1] \to b[q_2, X, q_1][q_1, R, q_1]$

$[q_1, R, q_1] \to \varepsilon$

$[q_2, X, q_1] \to b[q_2, X, q_1][q_1, X, q_1]$

$[q_2, X, q_1] \to c$

Probamos la gramática con, por ejemplo, la palabra abbbcc, y para que resulte más legible, daré alias a las variables:

S = S

A = [q1,X,q1]
B = [q2,X,q1]

entonces la gramática queda:
S->aAS|bBS|ε
A->aAA|b|c
B->bBA|c

probamos abbbcc:
S->
aAS->
abS->
abbBS->
abbbBAS->
abbbcAS->
abbbccS->
abbbcc

## 8. - Determinar qué lenguajes son regulares y/o libres de contexto

a) $z = 0^n1^n1^n0^n$
$z = 0^n1^n1^n0^n$
<u>000...0</u> <u>111...1</u> <u>111...1</u> <u>000...0</u> (___ = n)

El lenguaje no es libre de contexto, al bombear, la palabra resultante no pertenece al lenguaje.

--------------------------------------------------------------------------------

b) $z = 0^n10^n10^{n+n}$
Aplicando el lema de bombeo, nos damos cuenta que el lenguaje es libre de conexto ya que hay una relación binaria entre $0^n$ y $0^n$ con $0^{n+m}$ y por tanto al bombear, las palabras resultantes siguien perteneciendo al lenguaje.
Sin embargo, si aplicamos el lema de bombeo para lenguajes regulares, obtenemos que el lenguaje no es regular porque, al haber una relación binaria, si bombeamos, las palabras resultantes no pertenecen al lenguaje, por ejemplo, al bombear $0^n$, rompemos $0^{n+n}$

--------------------------------------------------------------------------------

c) Para representar este lenguaje podemos dar la expresión regular:

$(1(0+1))^*(1+\varepsilon)$

De forma que $(1(0+1))^*$ nos permite colocar un 1 siempre en la posición impar, y, $(1+\varepsilon)$ nos da la posibilidad de que la cadena tenga una longitud impar.

Por tanto, el lenguaje es regular.

## 9. - Encuentra una gramática libre de contexto en forma normal de Chomsky que genere el siguiente lenguaje definido sobre el alfabeto {a, 0, 1}:

Para encontrar la gramática, he construido primero un autómata con pila no determinista que acepta el lenguaje propuesto siguiendo el siguiente razonamiento:

Leo a al empezar (R).
Leo 0 y 1 y meto X e Y en la pila.
Leo a y paso a q2.
Leo 0 y 1 mientras vacío la pila.
Leo a al acabar (R).
Retiro el símbolo inicial de la pila (R).

Estados = {q1, q2}
Alfabeto de entrada = {0, 1, a}
Alfabeto de pila = {R, X, Y}
Funcion de transición = δ
Estado inicial = {q1}
Símbolo inicial de la pila = {R}
Estados finales = {Ø}

1- δ(q1, a, R) = {(q1, R)} - leer la primera a
2- δ(q1, 0, R) = {(q1, XR)} - leer 0 y meter X
3- δ(q1, 1, R) = {(q1, YR)} - leer 1 y meter Y
4- δ(q1, 0, X) = {(q1, XX)} - leer 0 y meter X
5- δ(q1, 1, X) = {(q1, YX)} - leer 1 y meter Y
6- δ(q1, 0, Y) = {(q1, XY)} - leer 0 y meter X
7- δ(q1, 1, Y) = {(q1, YY)} - leer 1 y meter Y
8- δ(q1, a, R) = {(q2, R)} - leer la a del medio
9- δ(q1, a, X) = {(q2, X)} - leer la a del medio
10- δ(q1, a, Y) = {(q2, Y)} - leer la a del medio
11- δ(q2, 0, X) = {(q2, ε)} - leer 0 y quitar X
12- δ(q2, 1, Y) = {(q2, ε)} - leer 1 y quitar Y
13- δ(q2, a, R) = {(q2, R)} - leer la última a
14- δ(q2, ε, R) = {(q2, ε)} - retirar el símbolo inicial de la pila

A continuación paso el autómata a gramática:
(En la parte de arriba a la derecha se pueden ver los alias dados a las Producciones para su

posterior simplificación mediante el algoritmo de producciones y símbolos inútiles)

⑫ $\delta(q_2, 1, Y) = \{(q_2, \varepsilon)\}$ ⑭ $\delta$ …

⓪ $S \to [q_1, R, q_1]$ $\quad S \to [q_1, R, q_2]$

⑪ $[q_2, X, q_2] \to 0$

⑫ $[q_2, Y, q_2] \to 1$

⑭ $[q_2, R, q_2] \to \varepsilon$

① $[q_1, R, q_2] \to a[q_1, R, q_2]$

⑧ $[q_1, R, q_2] \to a[q_2, R, q_2]$

⑨ $[q_1, X, q_2] \to a[q_2, X, q_2]$

⑩ $[q_1, Y, q_2] \to a[q_2, Y, q_2]$

⑬ $[q_2, R, q_2] \to a[q_2, R, q_2]$

② $[q_1, R, q_1] \to 0[q_1, X, q_1][q_1, R, q_1]$
$[q_1, R, q_2] \to 0[q_1, X, q_1][q_1, R, q_2]$
$[q_1, R, q_1] \to 0[q_1, X, q_2][q_2, R, q_1]$
$[q_1, R, q_2] \to 0[q_1, X, q_2][q_2, R, q_2]$

③ $[q_1, R, q_1] \to 1[q_1, Y, q_1][q_1, R, q_1]$
$[q_1, R, q_2] \to 1[q_1, Y, q_1][q_1, R, q_2]$
$[q_1, R, q_1] \to 1[q_1, Y, q_2][q_2, R, q_1]$
$[q_1, R, q_2] \to 1[q_1, Y, q_2][q_2, R, q_2]$

④ $[q_1, X, q_1] \to 0[q_1, X, q_1][q_1, X, q_1]$
$[q_1, X, q_2] \to 0[q_1, X, q_1][q_1, X, q_2]$
$[q_1, X, q_1] \to 0[q_1, X, q_2][q_2, X, q_1]$
$[q_1, X, q_2] \to 0[q_1, X, q_2][q_2, X, q_2]$

⑤ $[q_1, X, q_1] \to 1[q_1, Y, q_1][q_1, X, q_1]$
$[q_1, X, q_2] \to 1[q_1, Y, q_1][q_1, X, q_2]$
$[q_1, X, q_1] \to 1[q_1, Y, q_2][q_2, X, q_1]$
$[q_1, X, q_2] \to 1[q_1, Y, q_2][q_2, X, q_2]$

⑥ $[q_1, Y, q_1] \to 0[q_1, X, q_1][q_1, Y, q_1]$
$[q_1, Y, q_2] \to 0[q_1, X, q_1][q_1, Y, q_2]$
$[q_1, Y, q_1] \to 0[q_1, X, q_2][q_2, Y, q_1]$
$[q_1, Y, q_2] \to 0[q_1, X, q_2][q_2, Y, q_2]$

⑦ $[q_1, Y, q_1] \to 1[q_1, Y, q_1][q_1, Y, q_1]$
$[q_1, Y, q_2] \to 1[q_1, Y, q_1][q_1, Y, q_2]$
$[q_1, Y, q_1] \to 1[q_1, Y, q_2][q_2, Y, q_1]$
$[q_1, Y, q_2] \to 1[q_1, Y, q_2][q_2, Y, q_2]$

$A = [q_1, R, q_1]$ $\quad G = [q_2, R, q_1]$
$B = [q_1, R, q_2]$ $\quad H = [q_2, R, q_2]$
$C = [q_1, X, q_1]$ $\quad I = [q_1, Y, q_1]$
$D = [q_1, X, q_2]$ $\quad J = [q_1, Y, q_2]$
$E = [q_2, X, q_1]$ $\quad K = [q_2, Y, q_1]$
$F = [q_2, X, q_2]$ $\quad L = [q_2, Y, q_2]$

Simplificamos:

S→A|B

A→0CA|0DA|1IA|1JG

B→0CB|0DH|1IB|1JH|aB|aH

C→0CC|0DE|1IC|1JE

D→0CD|0DF|1ID|1JF|aF

I→0CI|0DK|1II|1Jk

J→0CJ|0DL|1IJ|1JL|aL

H→aH|ε

F→0

L→1

E, G, K

S→A|B

A→0CA|0DA|1IA

B→0CB|0DH|1IB|1JH|aB|aH

C→0CC|1IC

D→0CD|0DF|1ID|1JF|aF

I→0CI|1II

J→0CJ|0DL|1IJ|1JL|aL

H→aH|ε

F→0

L→1

A, C, I

a001a100a

aS

a0DH

a00DFH

a001JFFH

a001aLFFH

a001a100a

~~S→B~~

~~B→0CB|0DH|1J~~

S→B

B→0DH|1JH|aB|aH

D→0DF|1JF|aF

J→0DL|1JL|aL

H→aH|ε

F→0

L→1

B

S→0DH|1JH|aS|aH

D→0DF|1JF|aF

J→0DL|1JL|aL

H→aH|ε

F→0

L→1

La gramática que genera el lenguaje es:

S->0DH|1JH|aS|aH
D->0DF|1JF|aF
J->0DL|1JL|aL
H->aH|ε
F->0
L->1

Antes de continuar la probamos con, por ejemplo, la palabra a001a100a:
S->
aS->
a0DH->
a00DFH->
a001JFFH->
a001aLFFH->
a001a1FFH->
a001a10FH->
a001a100H->
a001a100a

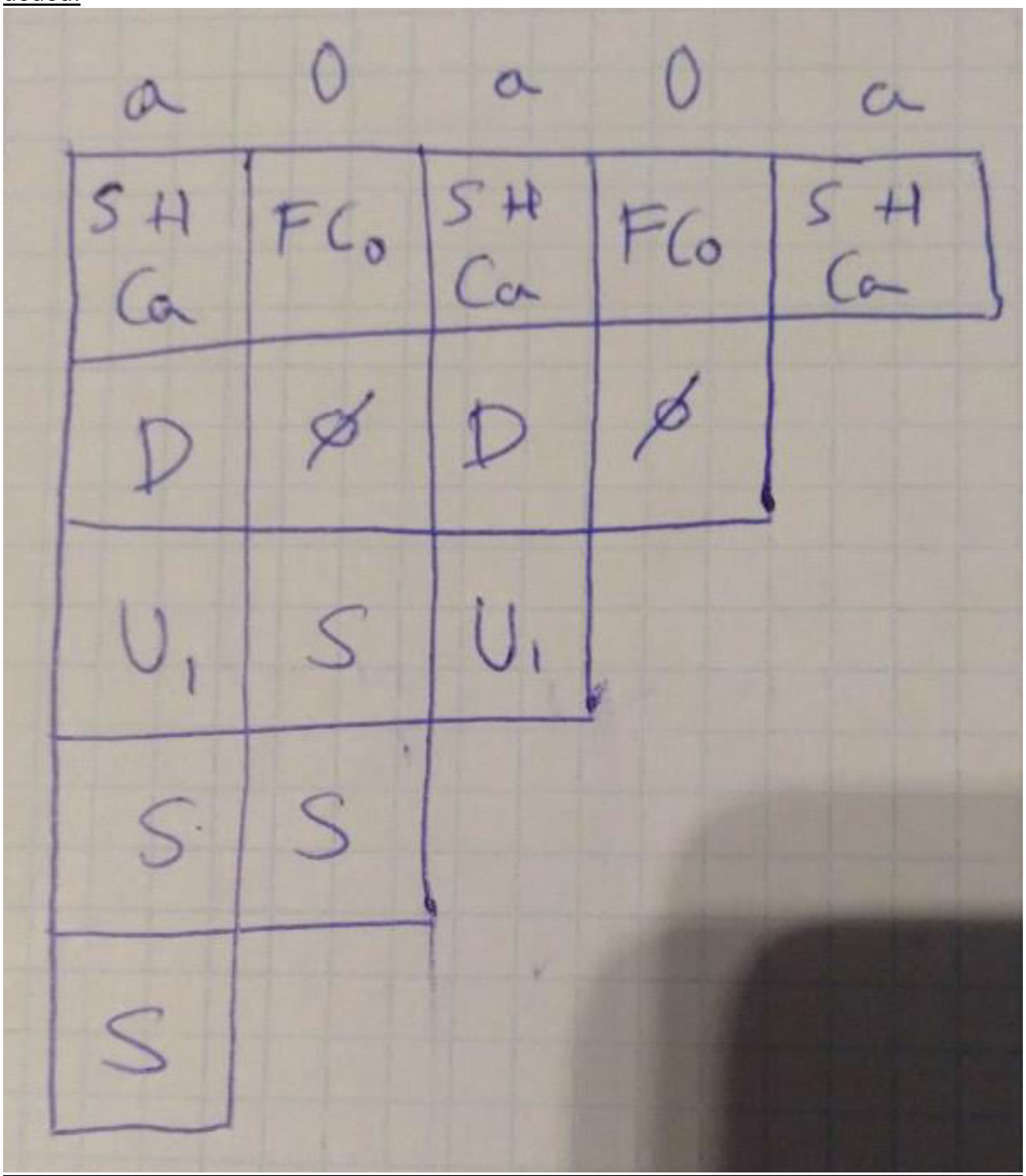
Como funciona, la pasamos a Forma Normal de Chomsky.

<u>Producciones nulas:</u>
S->0DH|1JH|aS|aH|0D|1J|a
D->0DF|1JF|aF
J->0DL|1JL|aL
H->aH|a
F->0
L->1

<u>Forma Normal de Chomsky:</u>
S->$C_0U_1$|$C_1U_2$|$C_aS$|$C_aH$|$C_0D$|$C_1J$|a
D->$C_0V_1$|$C_1V_2$|$C_aF$
J->$C_0W_1$|$C_1W_2$|$C_aL$
H->$C_aH$|a
F->0
L->1
$U_1$->DH
$U_2$->JH
$V_1$->DF
$V_2$->JF
$W_1$->DL
$W_2$->JL
$C_a$->a
$C_0$->0
$C_1$->1

Usamos el algoritmo de Cocke-Younger-Kasami para probar las palabras:

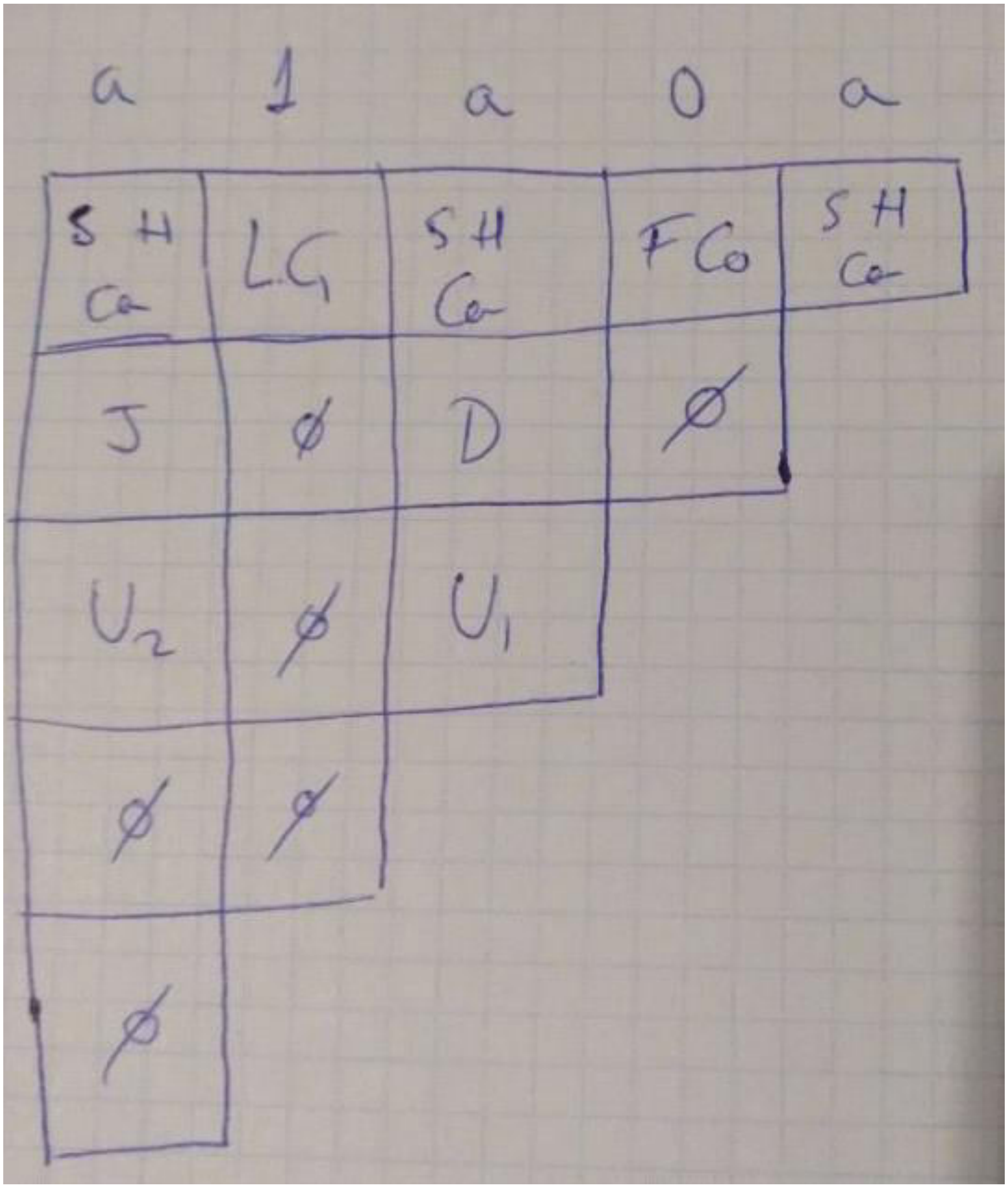
<u>a0a0a:</u>



| a | 0 | a | 0 | a |
|---|---|---|---|---|
| S H Ca | F $C_0$ | S H Ca | F $C_0$ | S H Ca |
| D | $\emptyset$ | D | $\emptyset$ | |
| $U_1$ | S | $U_1$ | | |
| S | S | | | |
| S | | | | |

La palabra a0a0a es generada.

<u>a1a0a:</u>




La palabra a1a0a no es generada.

## 10. - Comprobar, usando el algoritmo de Early si las palabras bba0d1 y cba1d1 pertenecen al lenguaje generado por la gramática:

a) bba0d1

Aplicamos el algoritmo de Early y, finalmente, en REGISTROS[6], aparece el registro de la forma (0, n, S, α, ε) -> (0, 6, S, AaC, ε) que confirma que la palabra bba0d1 pertenece al lenguaje generado por la gramática.

REGISTROS[0] = (0, 0, S, ε, AaB), (0, 0, S, ε, AaC), (0, 0, A, ε, Ab), (0, 0, A, ε, Ac),
(0, 0, A, ε, b), (0, 0, A, ε, c)
REGISTROS[1] = (0, 1, A, b, ε), (0, 1, S, A, aB), (0, 1, S, A, aC), (0, 1, A, A, b), (0, 1, A, A, c)
REGISTROS[2] = (0, 2, A, Ab, ε), (0, 2, S, A, aB), (0, 2, S, A, aC), (0, 2, A, A, b), (0, 2, A, A, c)
REGISTROS[3] = (0, 3, S, Aa, B), (0, 3, S, Aa, C), (3, 3, B, ε, BdC), (3, 3, B, ε, 0),
(3, 3, C, ε, CeB), (3, 3, C, ε, 1)
REGISTROS[4] = (3, 4, B, 0, ε), (3, 4, B, B, dC), (0, 4, S, AaB, ε)
REGISTROS[5] = (3, 5, B, Bd, C), (5, 5, C, ε, CeB), (5, 5, C, ε, 1)
REGISTROS[6] = (5, 6, C, 1, ε), (3, 6, C, 1, ε), (3, 6, B, BcD, ε), (3, 6, C, C, eB),
(0, 6, S, AaC, ε)

--------------------------------------------------------------------------------

b) cba1d1
Aplicamos el algoritmo de Early y al llegar a REGISTROS[5], no existe ningún registro que nos pueda proporcionar d, por tanto, se puede continuar el algoritmo, pero no se genera ningún registro, y así, la palabra cba1d1 no se genera.
Analizando la gramática observamos que esto tiene sentido, ya que para poder acceder a d, tendríamos que aplicar "Terminación" en B->BdC, cosa que nunca sucede.

REGISTROS[0] = (0, 0, S, ε, AaB), (0, 0, S, ε, AaC), (0, 0, A, ε, Ab), (0, 0, A, ε, Ac),
(0, 0, A, ε, b), (0, 0, A, ε, c)
REGISTROS[1] = (0, 1, A, c, ε), (0, 1, S, A, aB), (0, 1, S, A, aC), (0, 1, A, A, b), (0, 1, A, A, c)
REGISTROS[2] = (0, 2, A, Ab, ε), (0, 2, S, A, aB), (0, 2, S, A, aC), (0, 2, A, A, b), (0, 2, A, A, c)
REGISTROS[3] = (0, 3, S, Aa, B), (0, 3, S, Aa, C), (3, 3, B, ε, BdC), (3, 3, B, ε, 0),
(3, 3, C, ε, CeB), (3, 3, C, ε, 1)
REGISTROS[4] = (3, 4, C, 1, ε), (0, 4, S, AaC, ε), (3, 4, C, C, eB)
REGISTROS[5] =
REGISTROS[6] =

## Ejercicio extra: Comprueba mediante el algoritmo CYK si la cadena *a10100a00101a* pertenece al lenguaje generado por la gramática del ejercicio 9:

(Este ejercicio ha sido inventado por mi)

La gramática era:
$S \to C_0U_1|C_1U_2|C_aS|C_aH|C_0D|C_1J|a$
$D \to C_0V_1|C_1V_2|C_aF$
$J \to C_0W_1|C_1W_2|C_aL$
$H \to C_aH|a$

F->0
L->1
$U_1$->DH
$U_2$->JH
$V_1$->DF
$V_2$->JF
$W_1$->DL
$W_2$->JL
$C_a$->a
$C_0$->0
$C_1$->1

<u>a10100a00101a:</u>

| a | 1 | 0 | 1 | 0 | 0 | a | 0 | 0 | 1 | 0 | 1 | a |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S H Ca | L C1 | F C0 | L C1 | F C0 | F C0 | S H Ca | F C0 | F C0 | L C1 | F C0 | L C1 | S H Ca |
| J | Ø | Ø | Ø | Ø | Ø | D | Ø | Ø | Ø | Ø | Ø | |
| V2 | Ø | Ø | Ø | Ø | S | V1 | Ø | Ø | Ø | Ø | | |
| Ø | Ø | Ø | Ø | Ø | D | Ø | Ø | Ø | Ø | | | |
| Ø | Ø | Ø | Ø | S | W1 | Ø | Ø | Ø | | | | |
| Ø | Ø | Ø | Ø | J | Ø | Ø | Ø | | | | | |
| Ø | Ø | Ø | S | V2 | Ø | Ø | | | | | | |
| Ø | Ø | Ø | D | Ø | Ø | | | | | | | |
| Ø | Ø | S | W1 | Ø | | | | | | | | |
| Ø | Ø | J | Ø | | | | | | | | | |
| Ø | S | U2 | | | | | | | | | | |
| S | S | | | | | | | | | | | |
| S | | | | | | | | | | | | |

Como aparece S en la última casilla, la cadena a10100a00101a es generada.