Ejemplo 1

Expresiones Regulares

Sea $\Sigma = \{a, b, c\}$

Escribir una expresion regular para el lenguaje de todos los strings que contengan exactamente una b.

$\times\ a^* b c^*$ $\quad$ $\times\ a^* c^* b a^* c^*$

$(a|b)^* b (a|c)^*$

Ejemplo 2 (mismo alfa 1)

Todas las cadenas que tiene como max una b

$$(a|c)^* (\varepsilon|b) (a+c)^*$$

$$(a|c)^* b (a|c)^* \mid (a|c)^*$$

Ejemplo 3

$\Sigma = \{a, b\}$

Todas las cadenas con una b rodeada del mismo num de a

$$\{a^n b a^n, n \geq 0\} \longleftarrow$$

NO ES ER

Ejemplo 4

$\Sigma = \{a, b, c\}$

Todas las strings que no contienen 2 b consecutivas,

$$(a|c)^*(b(a|c)^+)^*(\varepsilon|b)$$

$$(b|\varepsilon)(a|c|ab|cb)^*$$

$$(a|c|ba|bc)^*(b|\varepsilon)$$

Expresiones regulares para Tokens de Lenguajes

1: ER para las palabras reservadas de C-

(if | then | else | while | ..)

2: ER para poder representar números

Clase	libro
digitos = [0 - 9]	
nat, = digitos$^+$	numeros : enteros ("." nat)? (Enteros)
enteros : (+, -)? nat	
float : enteros "." nat	

3 Identificadores

* Siempre iniciar con una letra

$$\boxed{[a\text{-}z\,A\text{-}Z]\,[a\text{-}z\,A\text{-}Z\,0\text{-}9]^*}$$

4 Comentarios en Pascal ({ .... })

$$\{(\wedge\}\,)^*\}$$

↑ no sean ese, todos los demas

/* * / * /

? - C++ (/* .. */ ó //)

$$/*\,(\,[\wedge *]^*\,|\,*^*\,|\,/^*\,|\,[\wedge /]^*\,)\,*/$$

Tarea 1

Hector Mauricio Gonzalez Coello
A01328258

1.

a) Lenguaje con vocabulario $\{0, 1\}$ que todos los strings que inician y terminan con un "0".

b) Lenguaje con vocabulario $\{a, b\}$ que todos los strings tienen al menos una "a".

c) Lenguaje con vocabulario $\{0, 1\}$ que todos los strings tienen al menos un "1".

d) Lenguaje con vocabulario $\{A \ldots Z\ a \ldots z\}$ con string compuesto por cualquier combinación de caracteres.

e) Lenguaje compuesto por un vocabulario $\{0, 1\}$ con strings que tienen al menos un "0" en la antepenúltima posición.

2.

a) $a(a|b|\ldots|y|z)^*a$

b) $(1|2|\cdots|8|9)^*(0|1|\ldots|8|9)^*$

c) $(\epsilon|1|\ldots|8|9)^*(0|2|4|6|8)^+$

d) $\epsilon/*([\wedge *]^*|*^*|/^*|[\wedge /]^*)*/\epsilon$

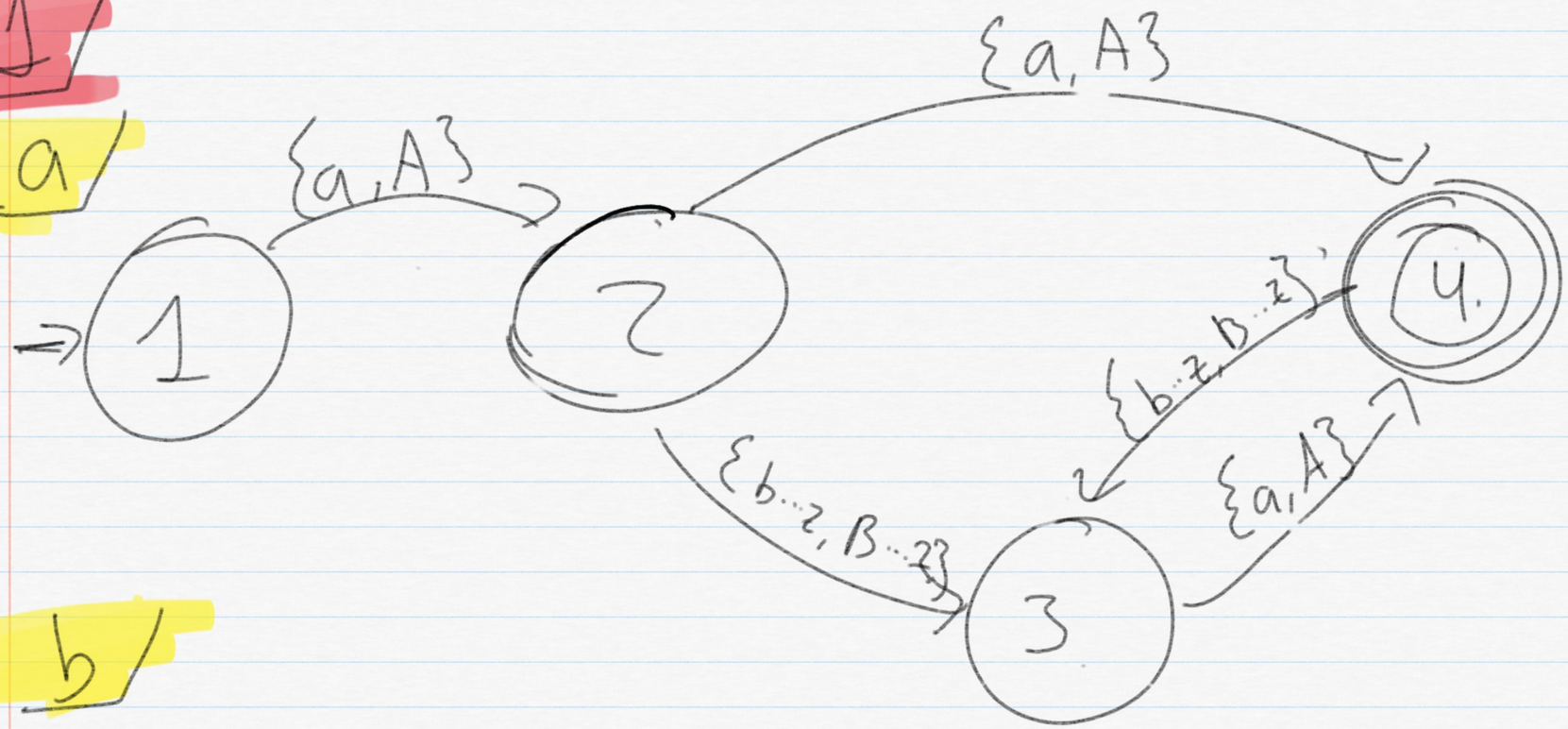
e) $\dfrac{(+|-)(0-9)^{+}(\in|(\because''(0-9)^{+}}{(\in|E((+|-)(0-9)^{+}))}$

Tarea 2

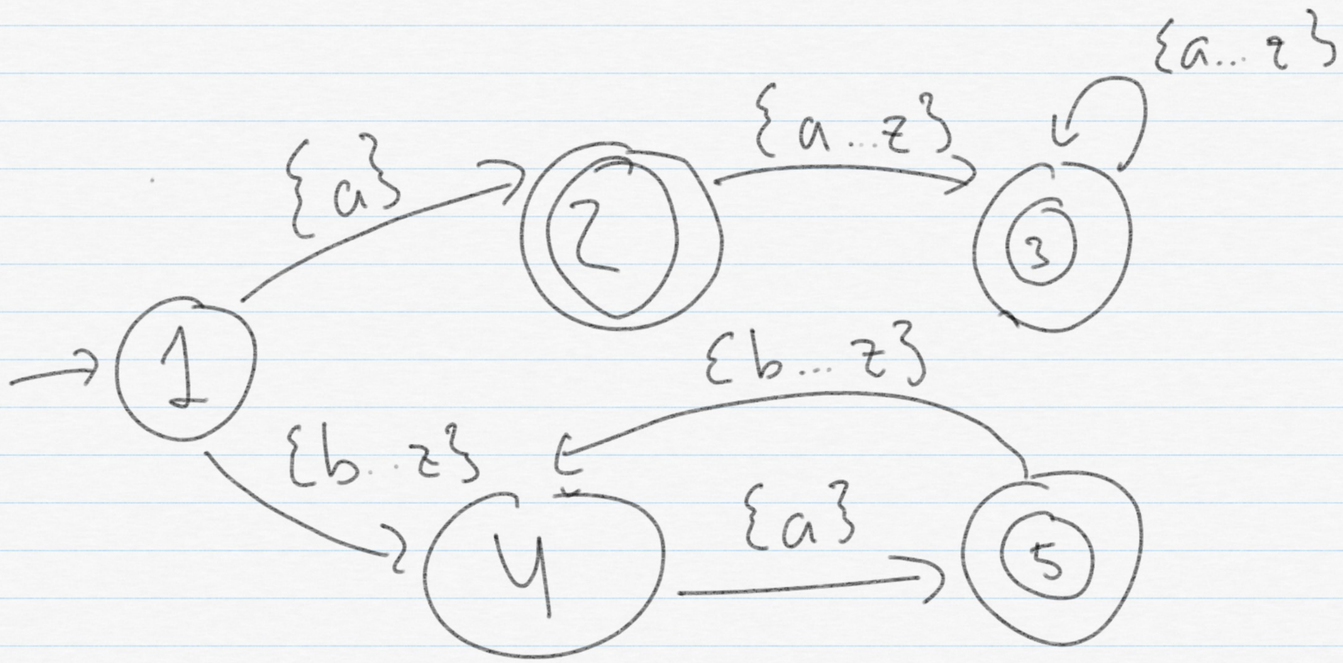
A01328258
Héctor Mauricio González Coello

1

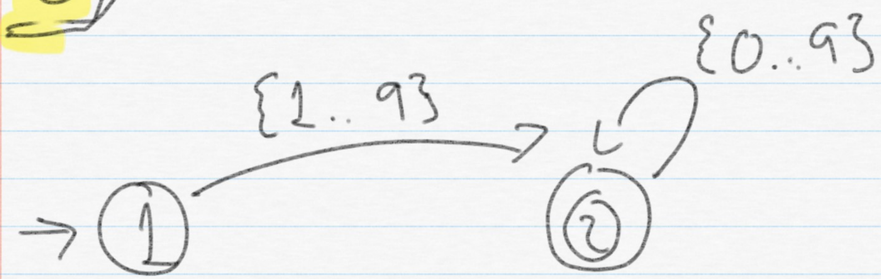
a



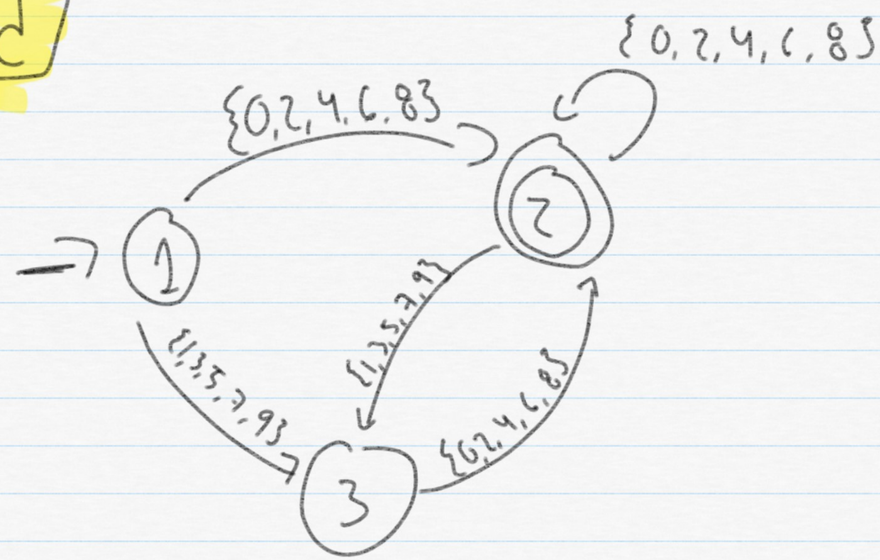
b



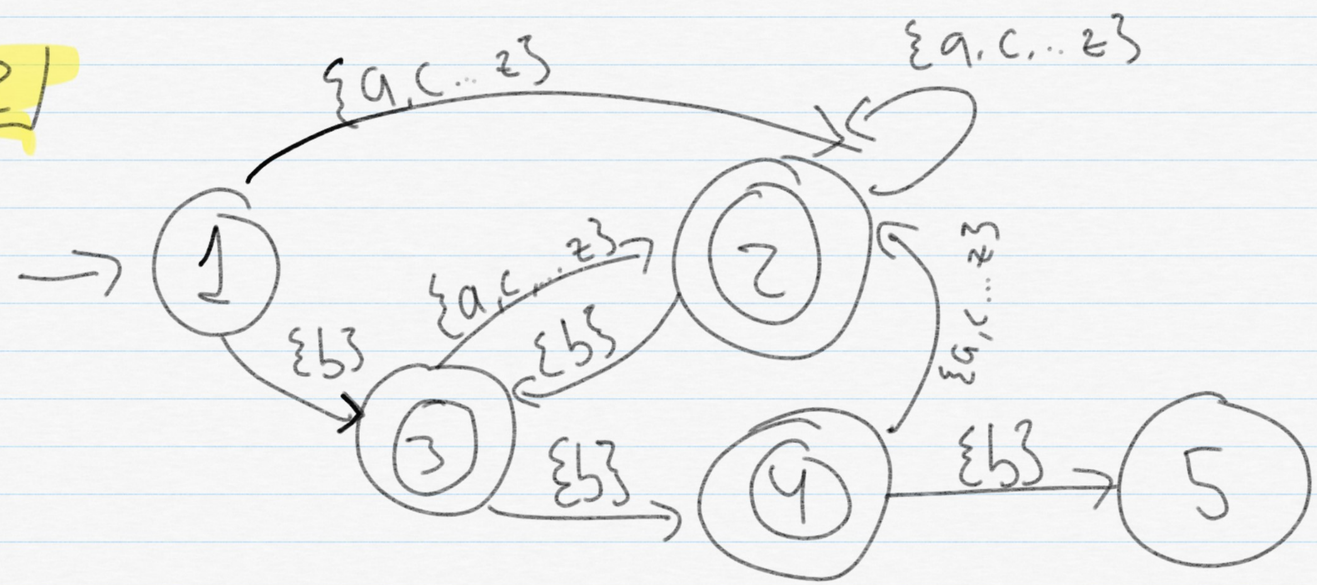
c


{1..9}
{0..9}
1
2

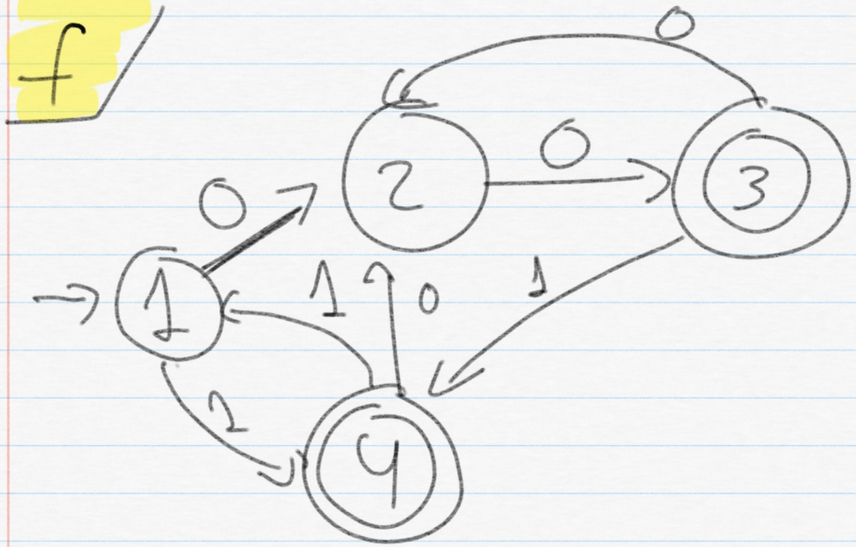
d


{0,2,4,6,8}
{0,2,4,6,8}
{1,3,5,7,9}
{1,3,5,7,9}
{0,2,4,6,8}
1
2
3

e

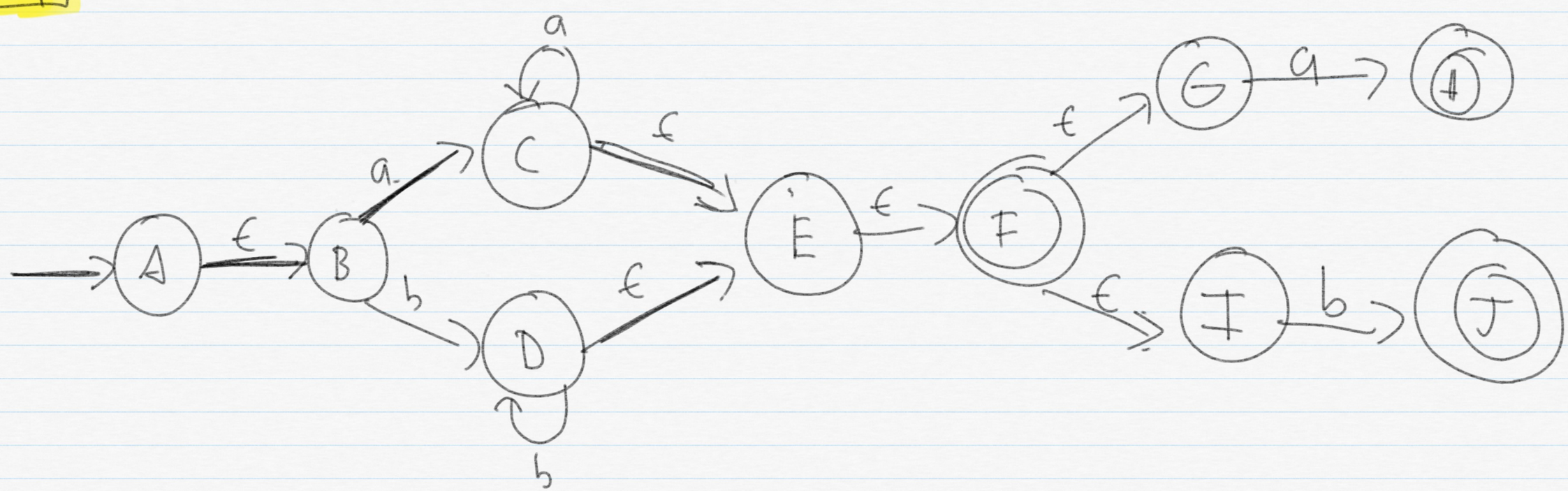

{a,c...z}
{a,c,..z}
{a,c,...z}
{b}
{b}
{a,c,...z}
{b}
{b}
1
2
3
4
5

f

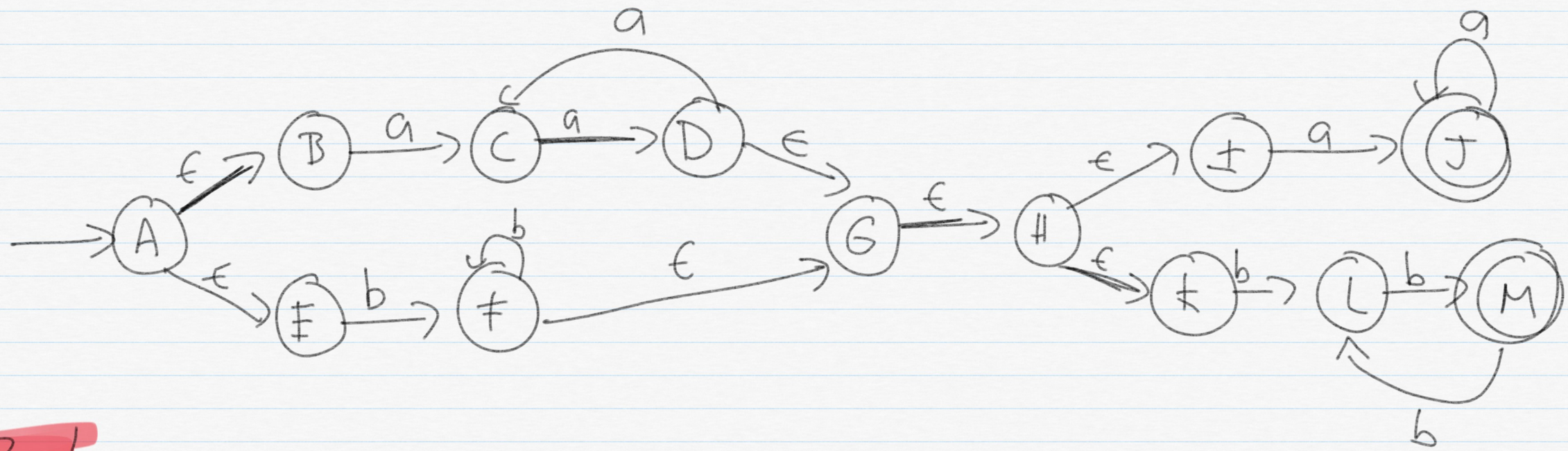


2

a

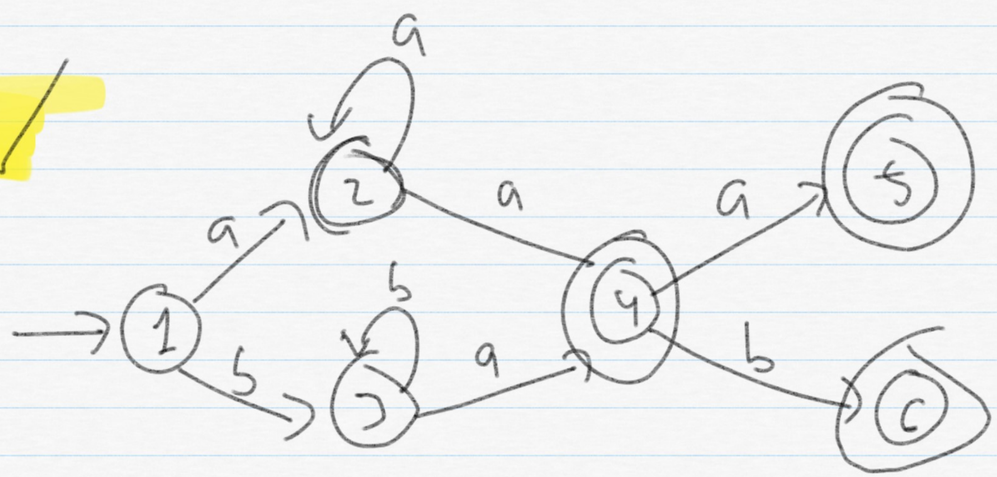


b/

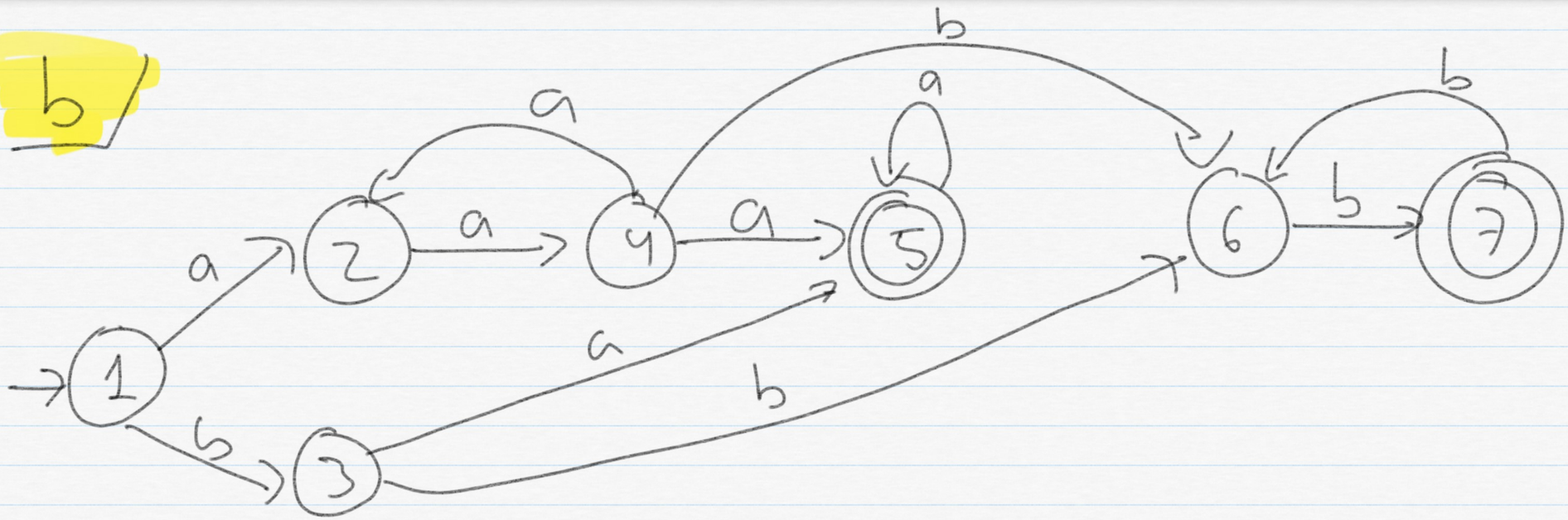


3/

a/

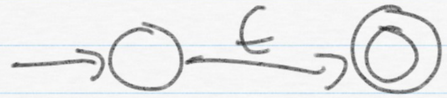


b

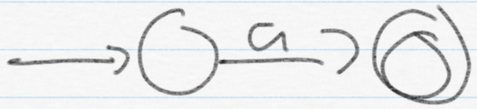


RE → NFA

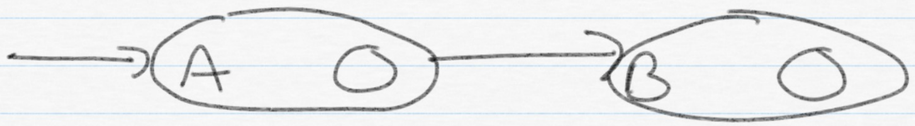
Para ϵ



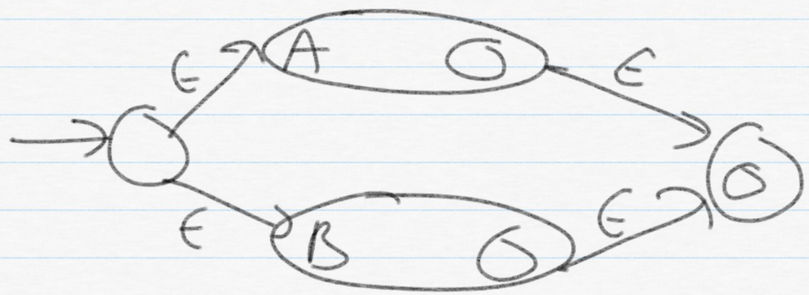
Para a



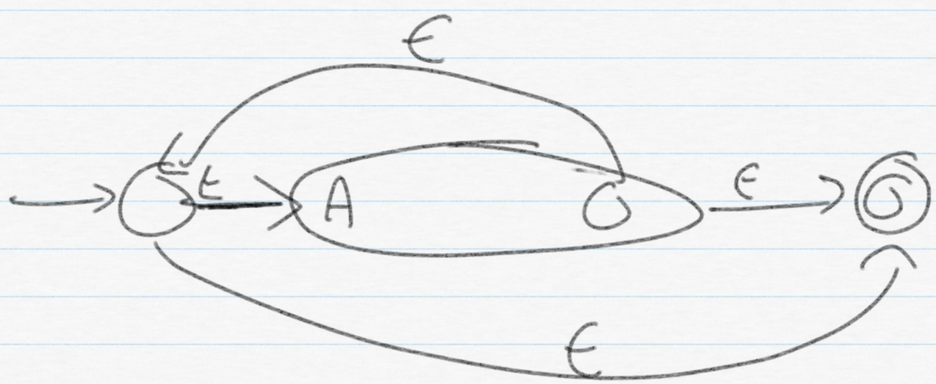
Para AB



Para A+B



Para A^*



Descente Recursivo

lexer

* Procedimiento que obtiene un token y regresa TRUE si hace match con el que indica la gramática.
También le incrementa el apuntador a los tokens

```
bool term (Token tok){
    return *next++ == tok; }
```

lexer

pide el siguiente Token.

* Un procedimiento por cada producción de un no terminal

E → T 1er Proc , E1

E → T+E 2da Proc E2.

* Un proc por cada no terminal de la izq.

E

+

$E \rightarrow T \mid T + E$

$T \rightarrow int \mid int * T \mid (E)$

$E \rightarrow T$

```
bool E1(){
    return T();}
```

$E \rightarrow T + E$

```
bool E2(){
    return T() && term(Plus) && E();}
```

$T \rightarrow int$

```
bool T1(){
    return term(int);}
```

$T \rightarrow int * T$ $\quad$ $* \rightarrow$ Mult

```
bool T2{
    return term(int) && term(*) && term(Close);}
```

Semestre Séptimo

Genesis A

pluma de gel

pluma de gel

pluma de gel automóvil

gorditas en gel

gel