

**Define los siguientes conceptos formalmente o conceptualmente**

**Define que es un alfabeto**

Un alfabeto es un conjunto finito no vacío de símbolos y se denota como  $\Sigma$

**Define que es un lenguaje**

Un lenguaje  $L$  es un conjunto de cadenas sobre un alfabeto  $\Sigma$  definido

**Define que es una cadena**

Una cadena es una secuencia finita de un alfabeto

Una cadena es una secuencia finita de símbolos que pertenecen a un alfabeto y comúnmente se denota con la letra  $w$ . La cadena vacía se denota como  $\epsilon$  y es una secuencia vacía de símbolos tomados de cualquier alfabeto  $\Sigma$

**Define que es una expresión regular**

Cualquier elemento que pertenezca al alfabeto se le conoce como expresión regular  $c \in \Sigma$

**Define que es una gramática tipo 0 (VN=conjunto finito de símbolos no terminales y VT=conjunto finito de símbolos terminales)**

Llamadas gramáticas no restringidas o gramáticas con estructura de frase.

Su regla de derivación es  $\alpha \rightarrow \beta$  siendo  $\alpha \in (VN \cup VT)^+$  y  $\beta \in (VN \cup VT)^*$

**Define que es una gramática tipo 1**

Llamadas gramáticas sensibles al contexto, en esta es importante tomar en cuenta la ubicación de los símbolos no terminales en la regla de derivación.

Su regla de derivación es  $\alpha A \beta \rightarrow \alpha \gamma \beta$  siendo  $A \in VN$ ;  $\alpha, \beta \in (VN \cup VT)^*$  y  $\gamma \in (VN \cup VT)^+$

**Define que es una gramática tipo 2**

Llamadas gramáticas de contexto libre o libres de contexto. Solo permiten tener un símbolo no terminal en su parte izquierda.

Su regla de derivación es  $A \rightarrow \alpha$  siendo  $A \in VN$  y  $\alpha \in (VN \cup VT)^+$

**Define que es una gramática tipo 3**

Llamada gramáticas regulares, se comienza con un símbolo terminal y puede ser seguido o no seguido por un símbolo no terminal

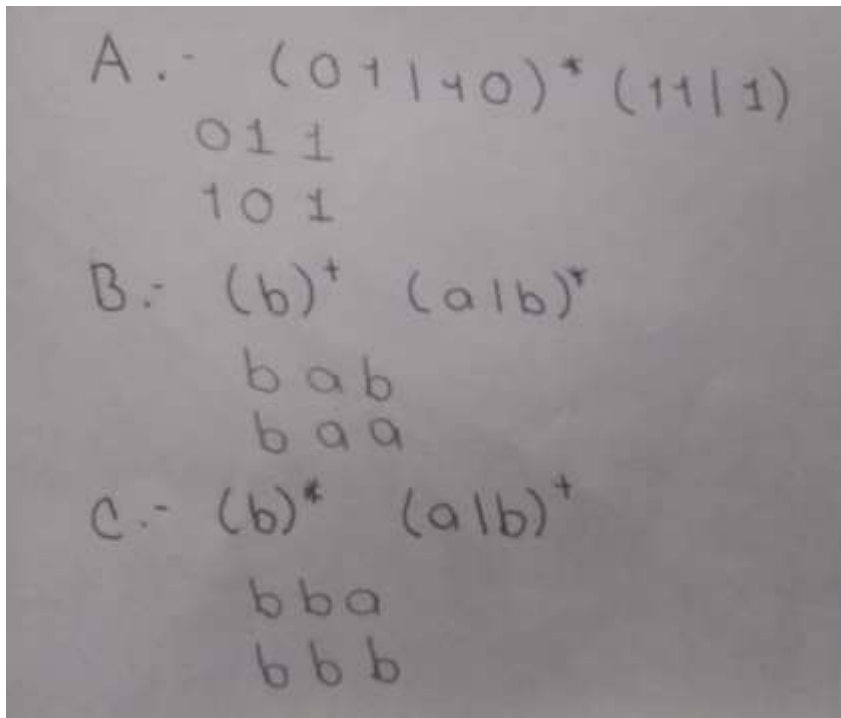
$$A \rightarrow \alpha B$$

$$A \rightarrow \alpha$$

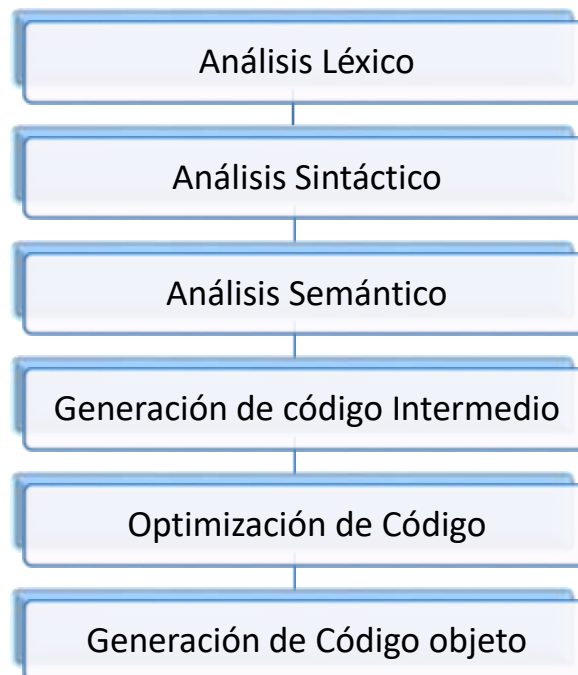
Donde  $A, B \in VN$  y  $\alpha \in VT$

Resuelve lo que se te pide

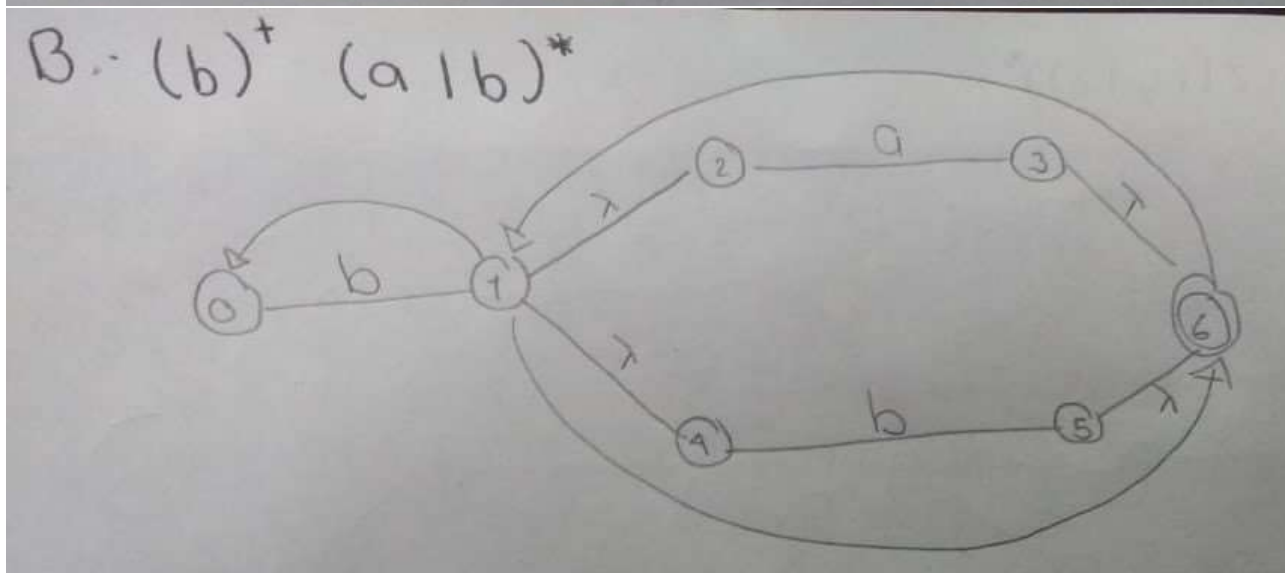
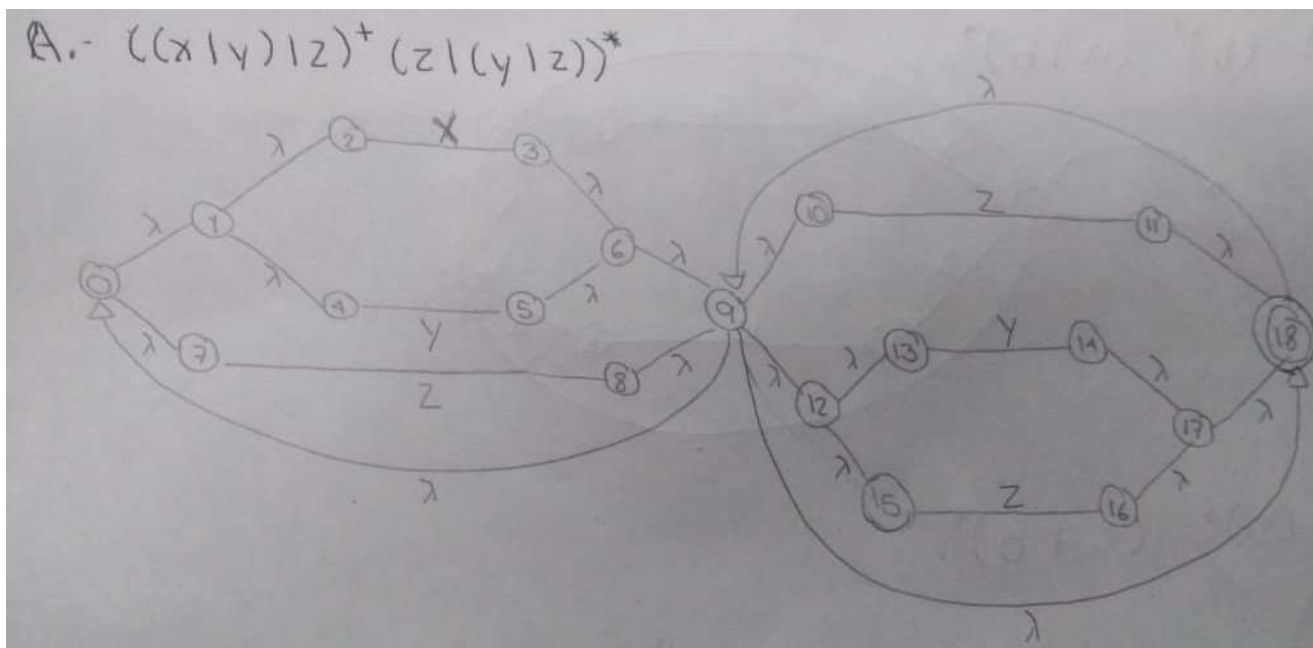
De las siguientes expresiones regulares, crea dos cadenas de no más de tres caracteres para cada expresión



Ordena las fases de un compilador



Convierte las siguientes expresiones regulares en un AF

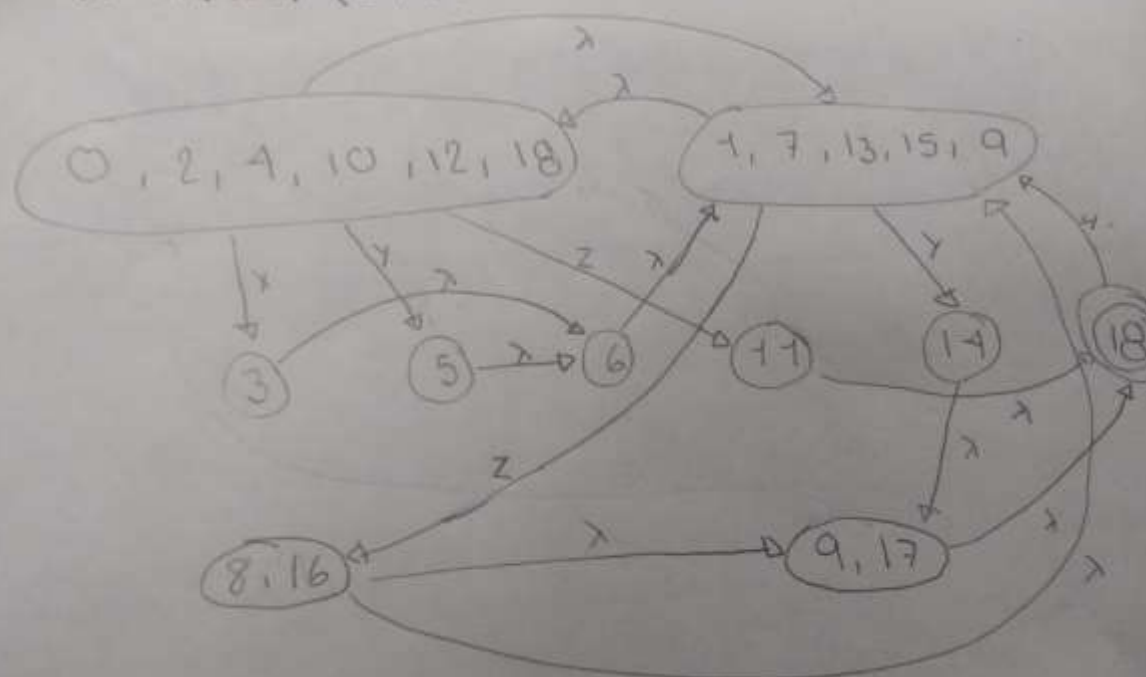


Convierte las expresiones del ejercicio anterior de un AFND a un AFD

$$A: ((x|y)|z)^+ (z|(y|z))^*$$

	x	y	z	λ
0	—	—	—	1, 7
1	—	—	—	2, 4
2	3	—	—	—
3	—	—	—	6
4	—	5	—	—
5	—	—	—	6
6	—	—	—	9
7	—	—	8	—
8	—	—	—	9
9	—	—	—	0, 10, 12, 18
10	—	—	11	—
11	—	—	—	18
12	—	—	—	13, 15
13	—	14	—	—
14	—	—	—	17
15	—	—	16	—
16	—	—	—	17
17	—	—	—	18
18	—	—	—	9
1, 7	—	—	8	2, 4
2, 4	3	5	—	—
0, 10, 12, 18	—	—	11	1, 7, 13, 15, 9
13, 15	—	14	16	—
1, 7, 13, 15, 9	—	14	8, 16	2, 4, 0, 10, 12, 18
2, 4, 0, 10, 12, 18	3	5	11	1, 7, 13, 15, 9
8, 16	—	—	—	9, 17
9, 17	—	—	—	0, 10, 12, 18

$$A = ((x|y|z)^*(z|(y|z))^*)^*$$



$$B: (b)^+ (a|b)^*$$

	a	b	$\lambda$
0	—	1	—
1	—	—	0, 2, 4, 6
2	3	—	6
3	—	—	—
4	—	5	—
5	—	—	6
6	—	—	1
0, 2, 4, 6	3	1, 5	1
1, 5	—	—	0, 2, 4, 6

$$B: (b)^+ (a+b)^*$$

