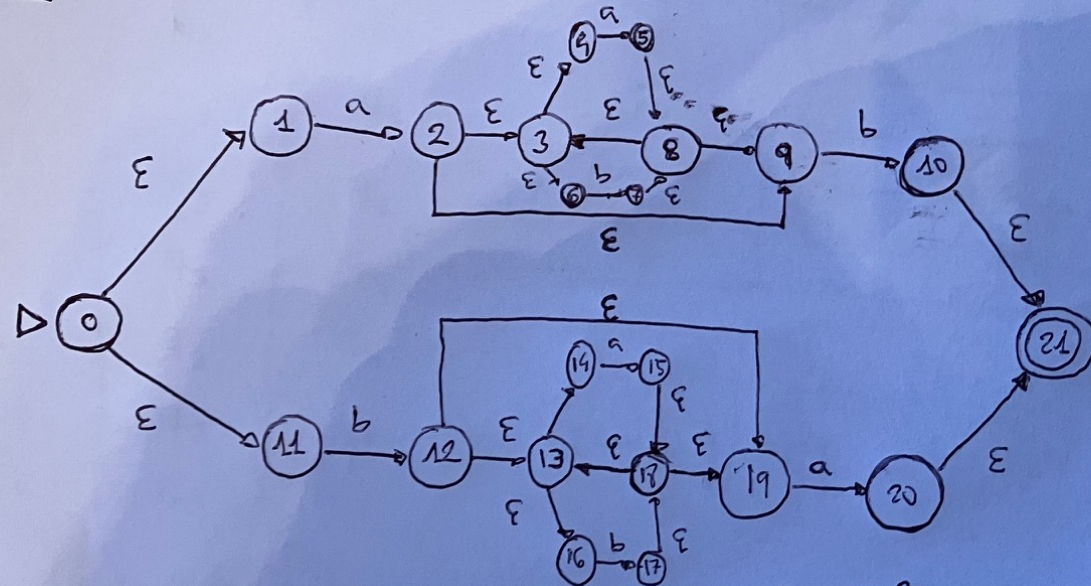


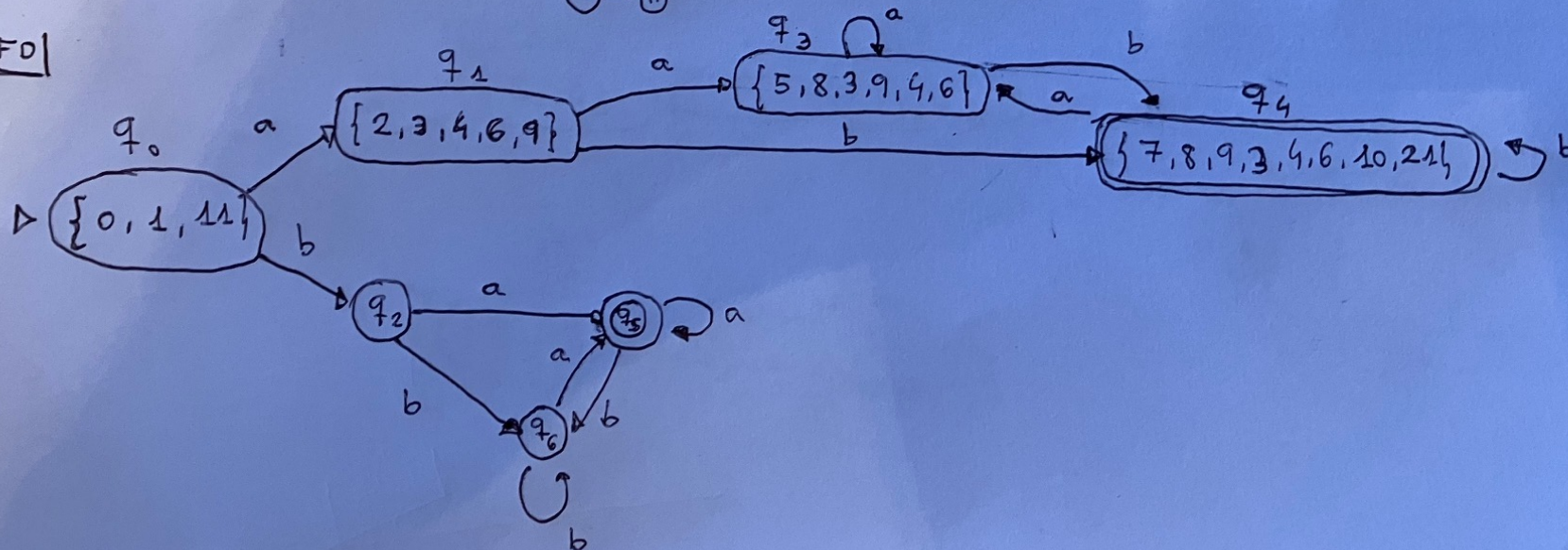
# EJERCICIO 3 DE LA HOJA 1

① REGEX  $(a(ab)^*b \mid b(ab)^*a)$

② AFN



③ AFD





# ④ AFD - MINIMAL

ESTADOS	a	b
q <sub>0</sub>	q <sub>1</sub>	q <sub>2</sub>
q <sub>1</sub>	q <sub>3</sub>	q <sub>4</sub>
q <sub>2</sub>	q <sub>5</sub>	q <sub>6</sub>
q <sub>3</sub>	q <sub>3</sub>	q <sub>4</sub>
q <sub>4</sub>	q <sub>3</sub>	q <sub>4</sub>
q <sub>5</sub>	q <sub>5</sub>	q <sub>6</sub>
q <sub>6</sub>	q <sub>5</sub>	q <sub>6</sub>

GRUPOS	ESTADOS	a	b
G <sub>1</sub> (NO FINAL)	q <sub>0</sub>	G <sub>1</sub>	G <sub>1</sub>
	q <sub>1</sub>	G <sub>1</sub>	G <sub>2</sub>
	q <sub>2</sub>	G <sub>2</sub>	G <sub>1</sub>
	q <sub>3</sub>	G <sub>1</sub>	G <sub>2</sub>
	q <sub>6</sub>	G <sub>2</sub>	G <sub>1</sub>
G <sub>2</sub> (FINAL)	q <sub>4</sub>	G <sub>1</sub>	G <sub>2</sub>
	q <sub>5</sub>	G <sub>2</sub>	G <sub>1</sub>

GRUPOS	ESTADOS	a	b
G <sub>1</sub>	G <sub>11</sub>	q <sub>0</sub>	G <sub>12</sub>
	G <sub>12</sub>	q <sub>1</sub> q <sub>3</sub>	G <sub>12</sub>
	G <sub>13</sub>	q <sub>2</sub> q <sub>6</sub>	G <sub>22</sub>
	G <sub>21</sub>	q <sub>4</sub>	G <sub>12</sub>
G <sub>2</sub>	G <sub>22</sub>	q <sub>5</sub>	G <sub>22</sub>
			G <sub>13</sub>

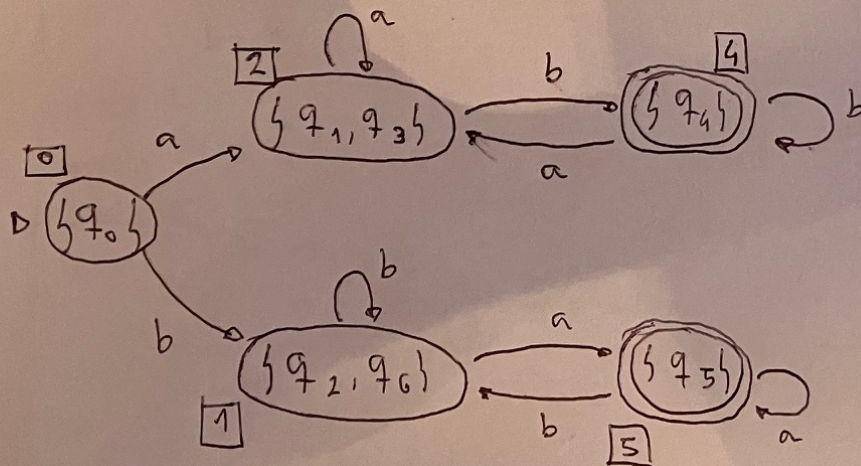
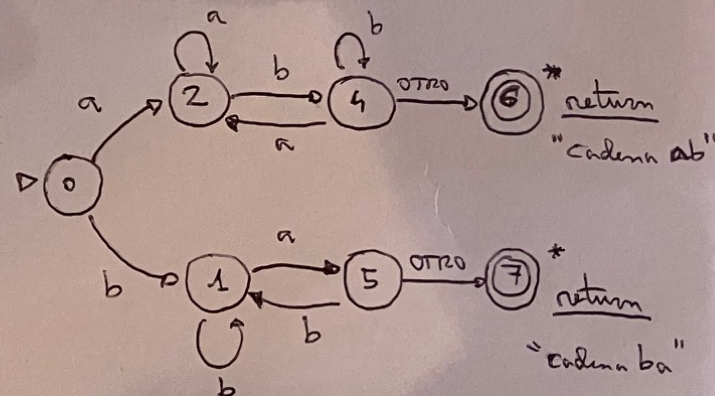


DIAGRAMA DE TRANSICIÓN DE ESTADOS (DTE)





## ⑤ ANALIZADOR LÉXICO (CONTROLADOR): C++

TOKEN obtenerCadena() {

TOKEN tokenCadena = new (Cadena);

while (~~estado~~<sup>1</sup>) { /\* repite el procesamiento de caracteres hasta que  
ocurre un retorno o un fallo.

switch (estado) {

case 0:

c = sigCar();

if (c == 'a') estado = 2

else if (c == 'b') estado = 1

else fallo(); /\* el lexema no es la cadena buscada \*/

break;

case 1:

...

case 6:

retractar();

tokenCadena. atributo = "cadena ab";

tokenCaden. valor = yytext;

...

return tokenCadena;

case 7:

...

}

}



# ⑥ BUFFER DE ENTRADA (ESQUEMA DE 2 BUFFERS)

switch (\*avance++) {

case <sup>\$</sup> eof :

if ('avance' está al final del BUFFER 1) {

recargar BUFFER 2

avance = inicio BUFFER 2;

}

else if ('avance' está al final del BUFFER 2) {

recargar BUFFER 1;

avance = inicio BUFFER 1;

}

else /\* 'eof' dentro de un buffer es el final de la entrada \*/

Terminar el análisis léxico

break;

case ... /\* CASOS PARA EL RESTO DE CARACTERES (DTE'S)

token = obtenerCadena();

token = obtenerNumero();

...

