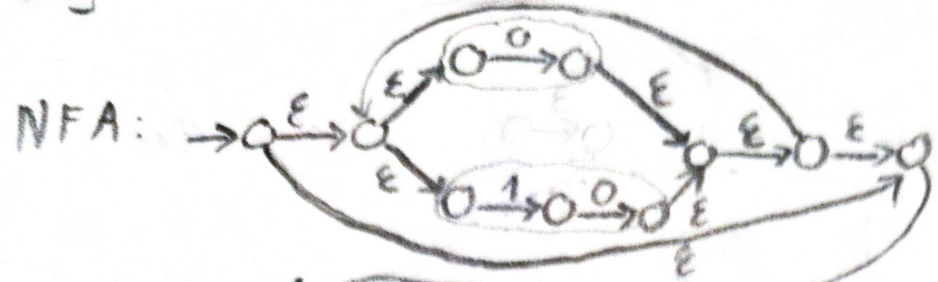
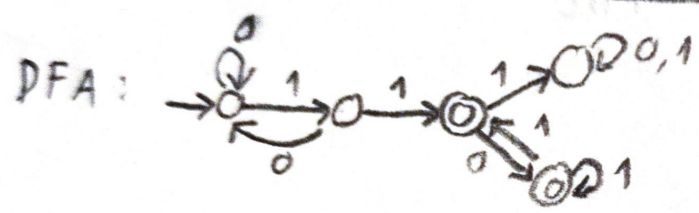
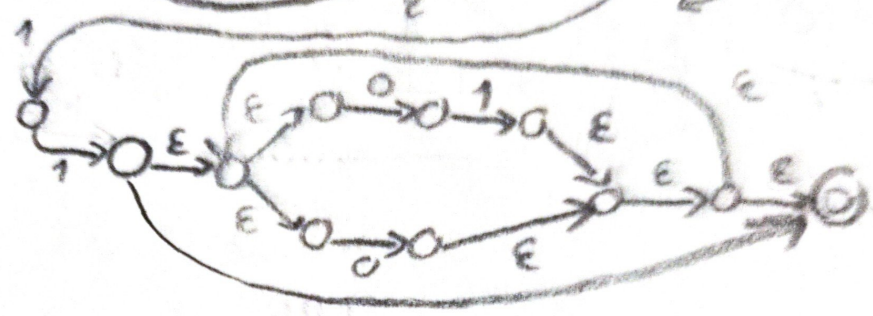


1 a) **Vou resolver os NFAs com o algoritmo de Thompson**
portanto vão ficar maiores que o necessário

Regex: $(0110)^* 11 (0110)^*$

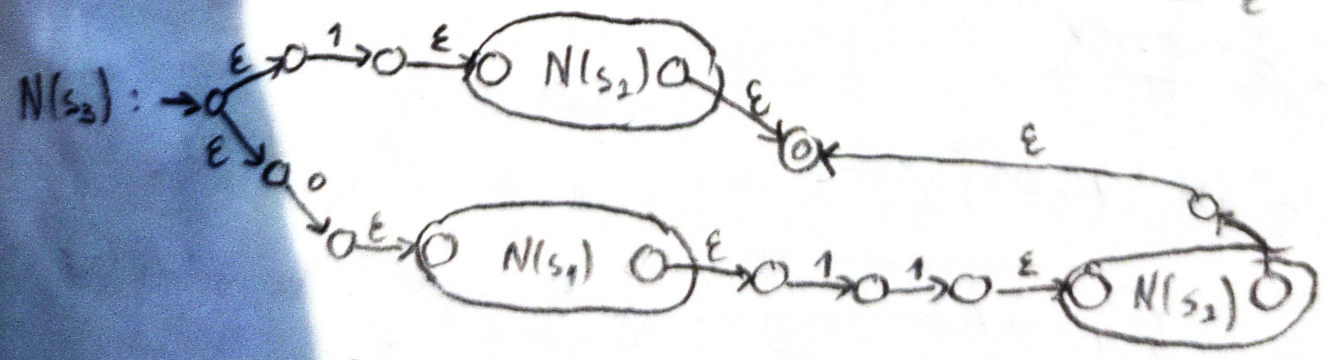
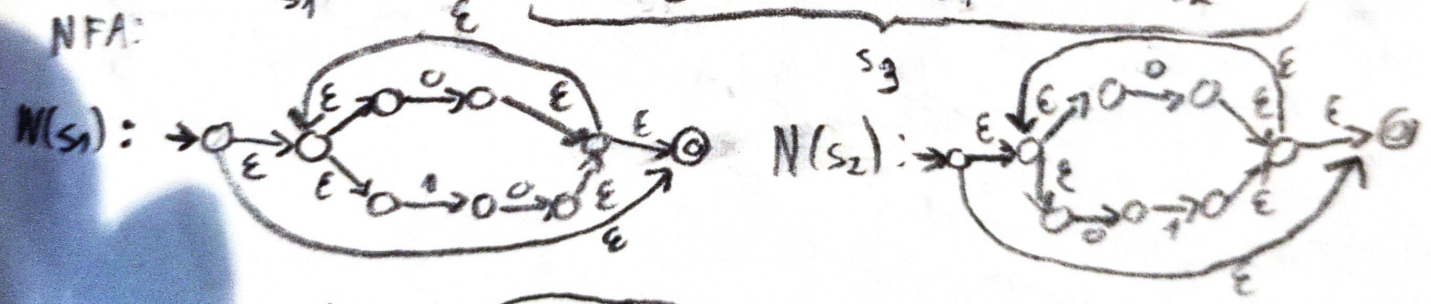


Algoritmo de Thompson

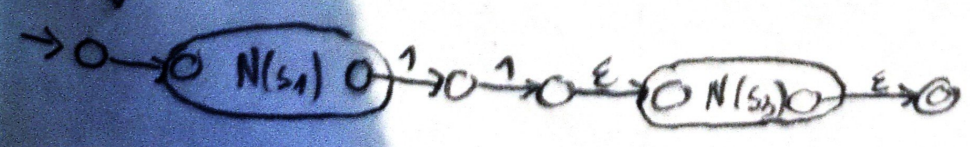


b) (111 são 2 ocorrências)

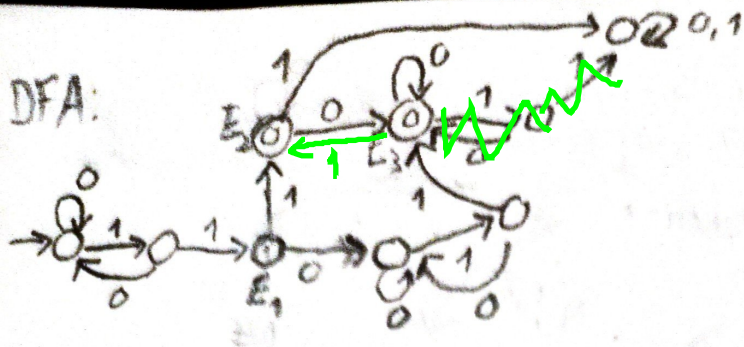
Regex: $(0110)^* 11 (1(0101)^* | (0(0110)^* 11(0101)^*))$



Final



DFA:



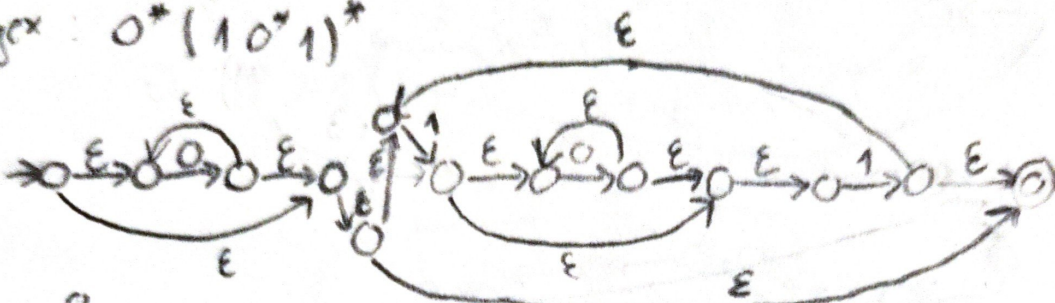
$$E_1: \#(11) = 1$$

$$E_2: \#(111) = 1$$

$$E_3: \#(11) = 2$$

c) Regex: $0^*(10^*1)^*$

NFA:

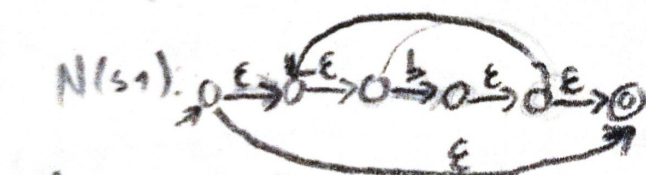


DFA:

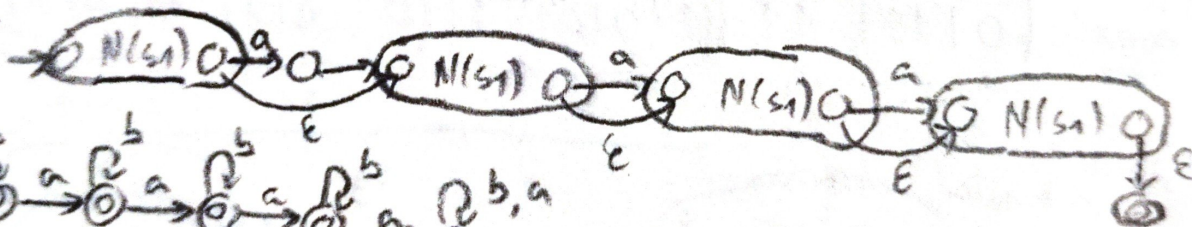


d) Regex: $b^*a^?b^*a^?b^*a^?b^*$

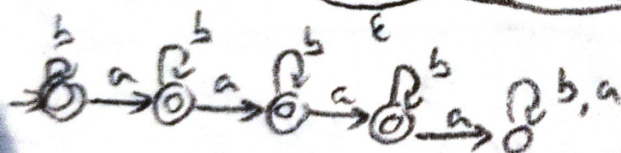
NFA:



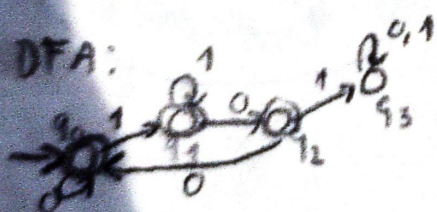
Final:



DFA:

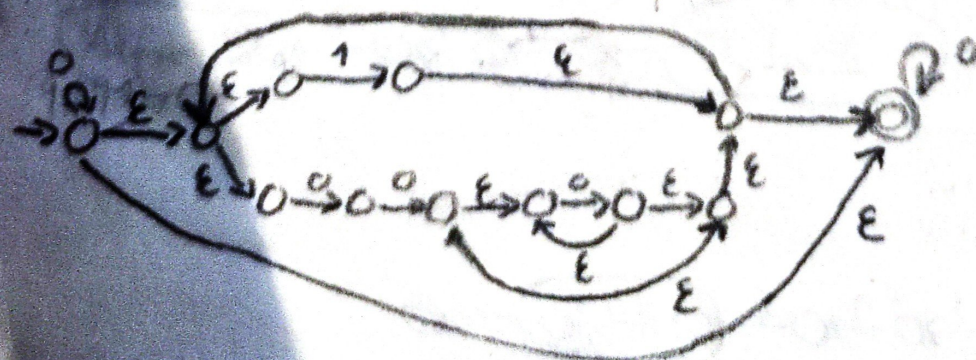


2 a) DFA:



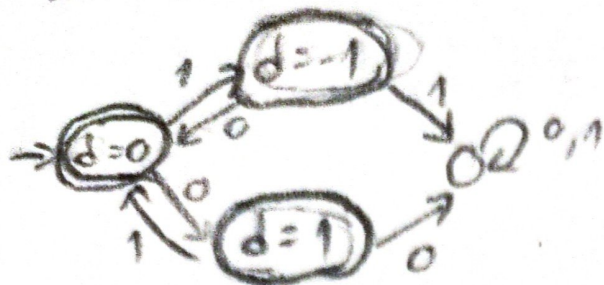
Regex: $0^*(11000^*)^*0^*$

NFA:



b) DFA

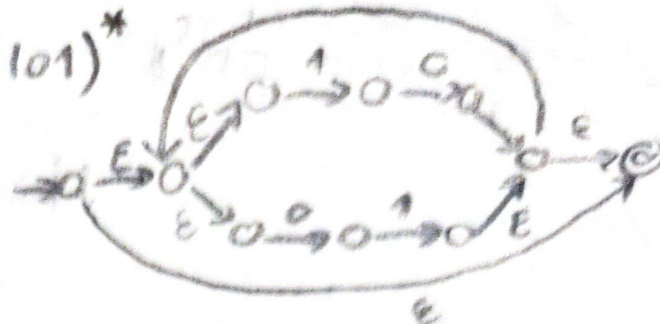
$$d = (\#0 - \#1)$$



Regex:

$$(10101)^*$$

NFA:



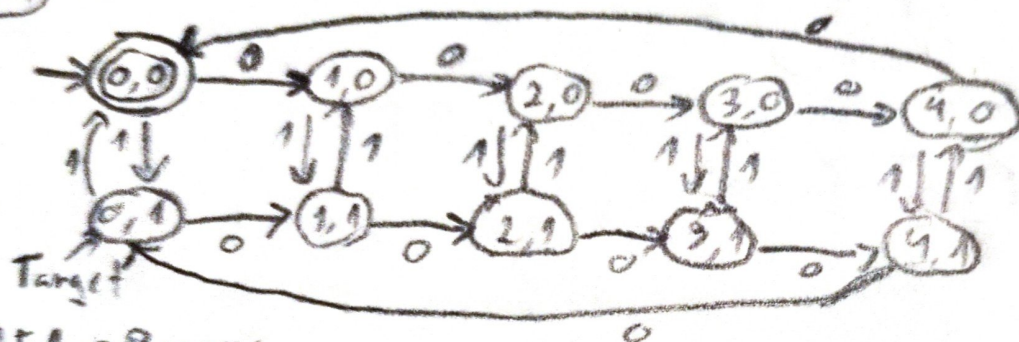
c)

$$m = \#0 \bmod 5$$

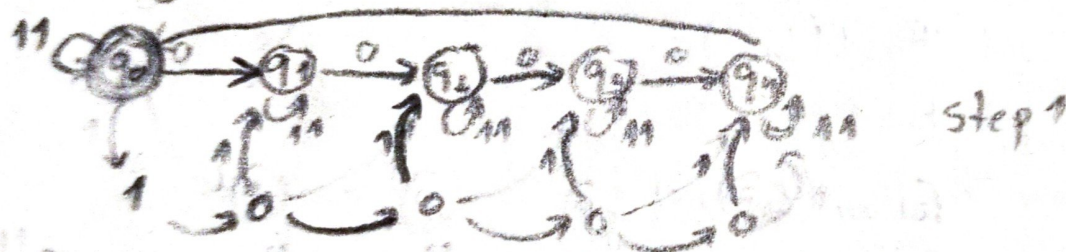
$$p = (\#1 \bmod 2 = 0 ? 0 : 1)$$

$$\text{node } (x, y) \Leftrightarrow m=x, p=y$$

DFA

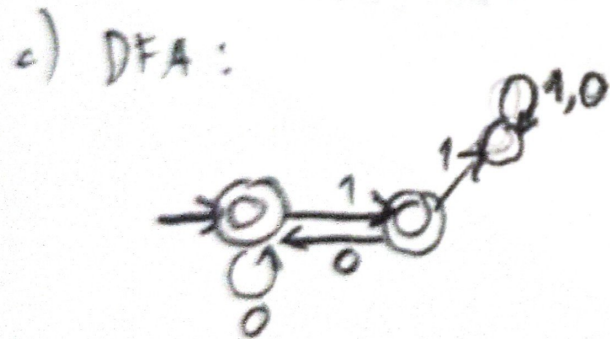
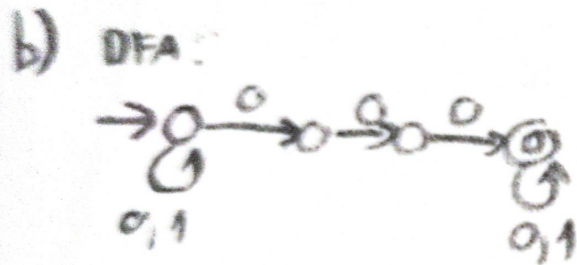
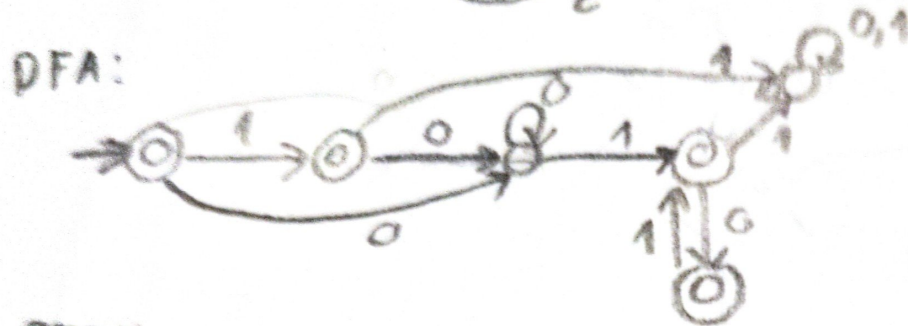
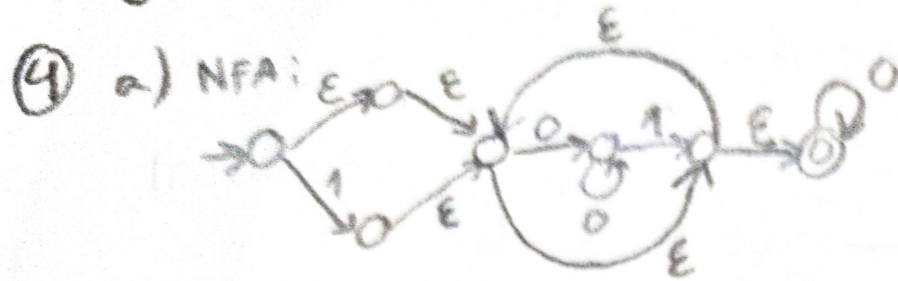


NFA → Regex:



Continuar eliminação de estados...
(não estou para isso)

- 3) a) set de strings binárias sem dois 0's seguidos
 b) set de strings binárias com a substring "000"
 c) igual ao a)



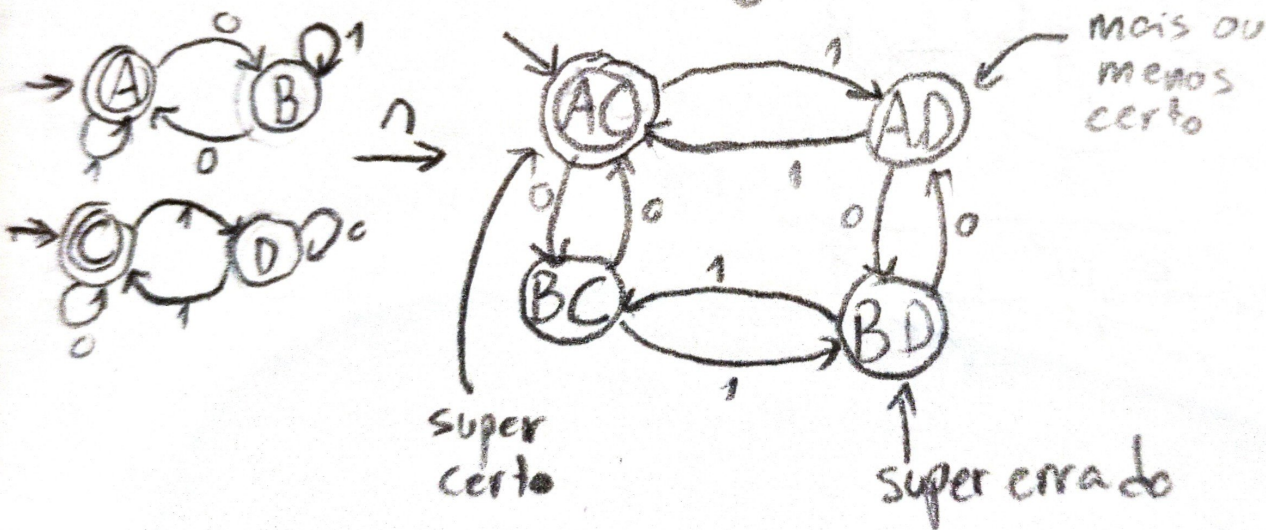
5) Prova que linguagens regulares são fechadas sob interseção

$L_1 = \{ \#0 \bmod 2 = 0 \}$ $M_1: \rightarrow \text{DFA with 2 states}$ $M_2: \rightarrow \text{DFA with 2 states}$
 $L_2 = \{ \#1 \bmod 2 = 0 \}$

Se L_1 é aceita por $M_1 (Q_1, \Sigma, q_1, T_1, \delta_1)$ e L_2 por $M_2 (Q_2, \Sigma, q_2, T_2, \delta_2) \Rightarrow L_1 \cap L_2$ é aceita pelo DFA com $(Q_1 \times Q_2, \Sigma, (q_1, q_2), T_1 \times T_2, \delta)$ com

$$\delta((r, s), x) = (\delta_1(r, x), \delta_2(s, x))$$

Ou seja, cada estado no produto é um par de estados dos automatos originais



6)

a) set de strings com 2 a's adjacentes unido com
 " " 2 b's adjacentes

b) $((b|a)^*(aa)(b|a)^*) \cup ((b|a)^*(bb)(b|a)^*)$