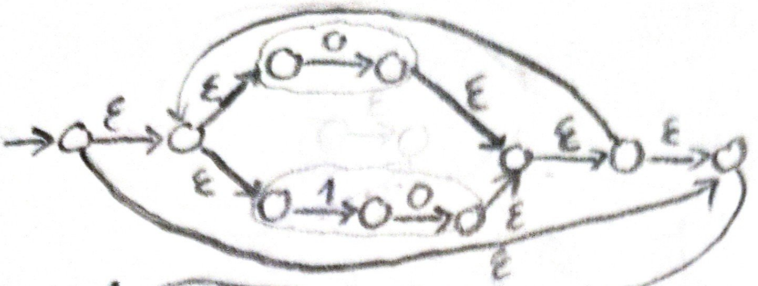
TP1 - Comp

Vou resolver os NFAs com o algoritmo de Thompson portanto vão ficar maiores que o necessário

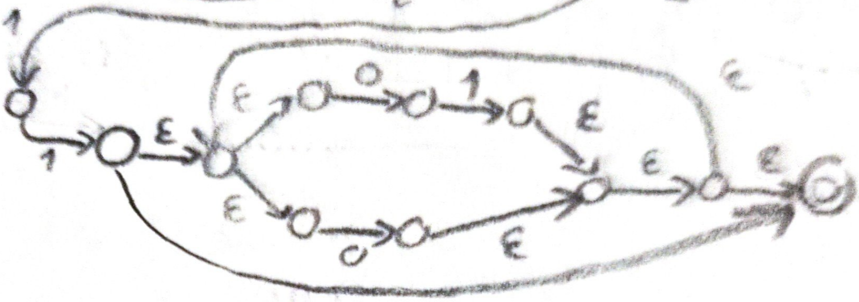
1 a)

Regex: $(0 \mid 10)^* \, 11 \, (01 \mid 0)^*$

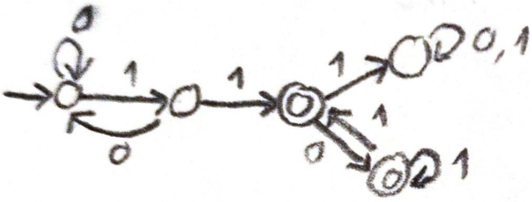
NFA:



Algoritmo de Thompson



DFA:

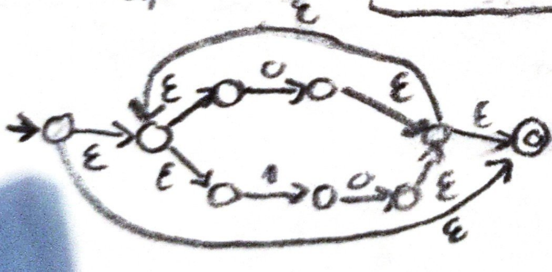


b) (111 são 2 ocorrências)

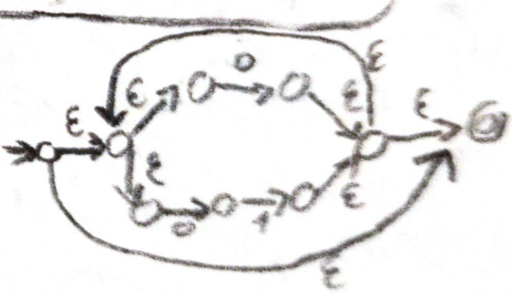
Regex: $\underbrace{(0\mid 10)^*}_{s_1} \, 11 \, \underbrace{\left(\left(1 \underbrace{(0\mid 01)^*}_{s_2}\right) \Big| \left(0 \underbrace{(0\mid 10)^*}_{s_1} 11 \underbrace{(0\mid 01)^*}_{s_2}\right) \right)}_{s_3}$

NFA:

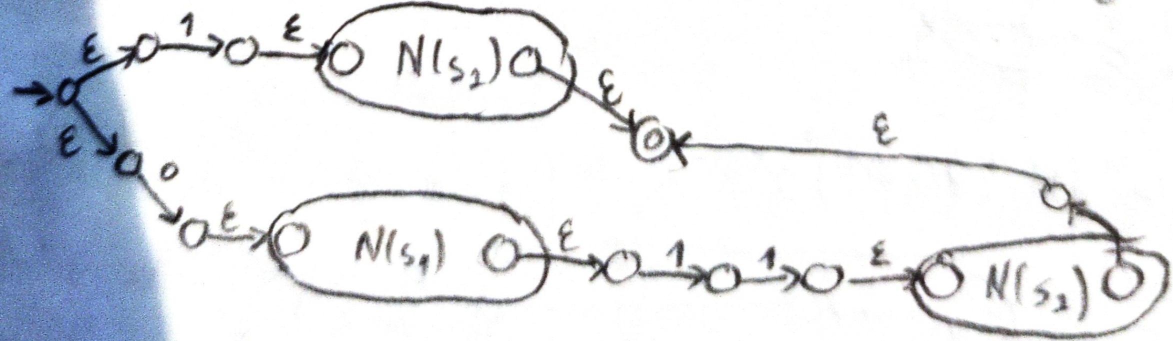
$N(s_1)$:



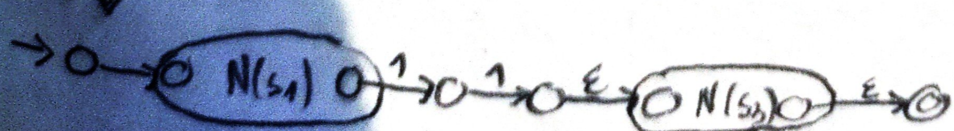
$N(s_2)$:



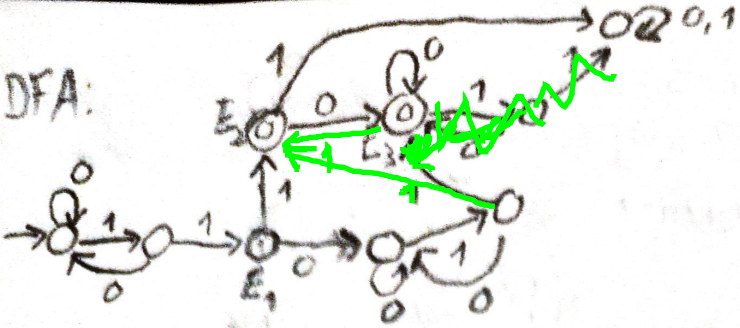
$N(s_3)$:



Final:



DFA:



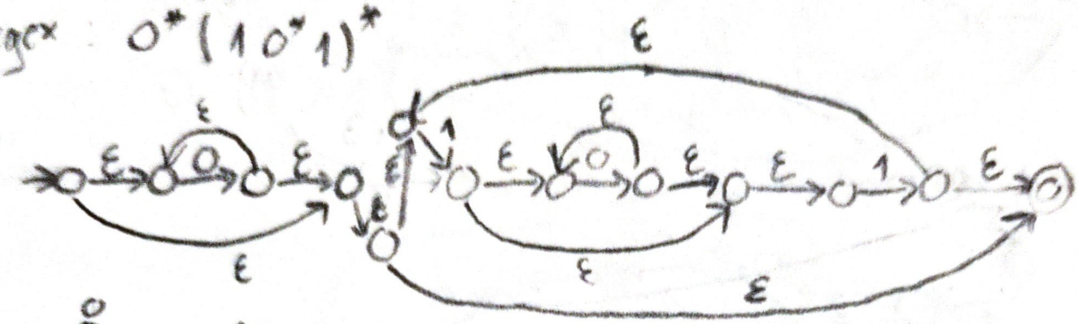
$E_1 : \#(11) = 1$

$E_2 : \#(111) = 1$

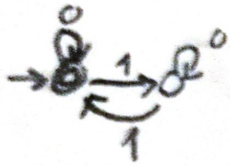
$E_3 : \#(11) = 2$

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

NFA:

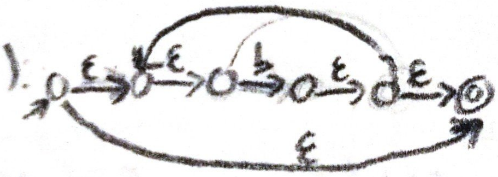


DFA:

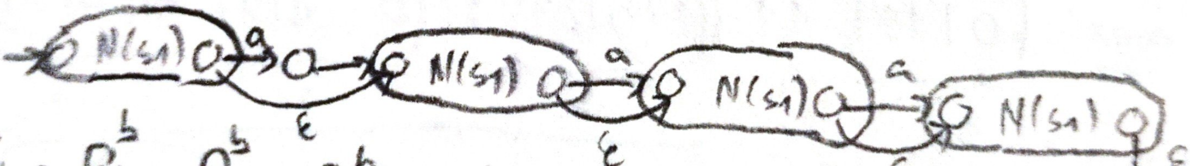


d) Regex: $\underbrace{b^*}_{s_1} a^? b^* a^? b^* a^? b^*$

NFA: $N(s_1)$:

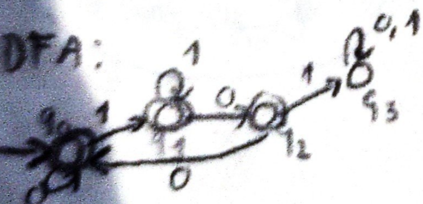


Final:



DFA:

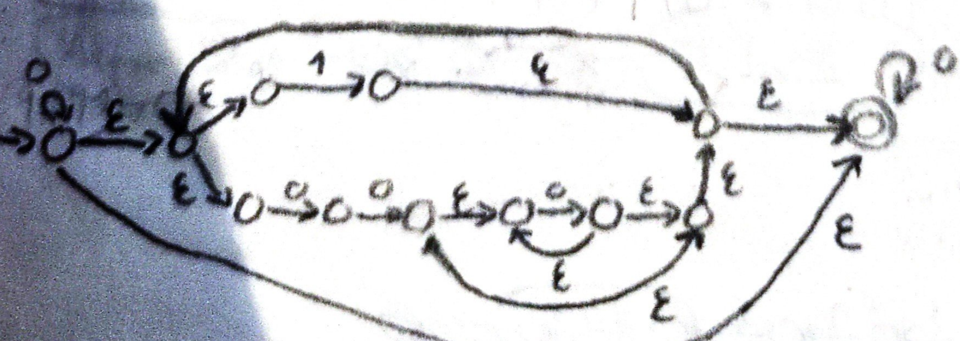
2 a) DFA:



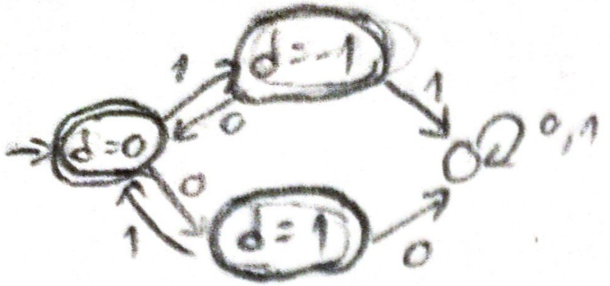
Regex: $0^*(1|000^*)^*0^*$

$E(q_0)$

NFA:



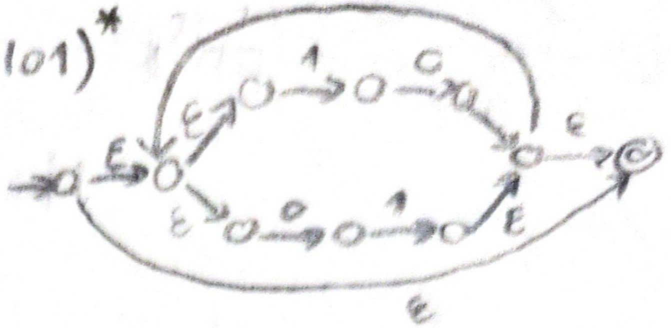
b) DFA $d = (\#0 - \#1)$



Regex:

$(10|01)^*$

NFA:

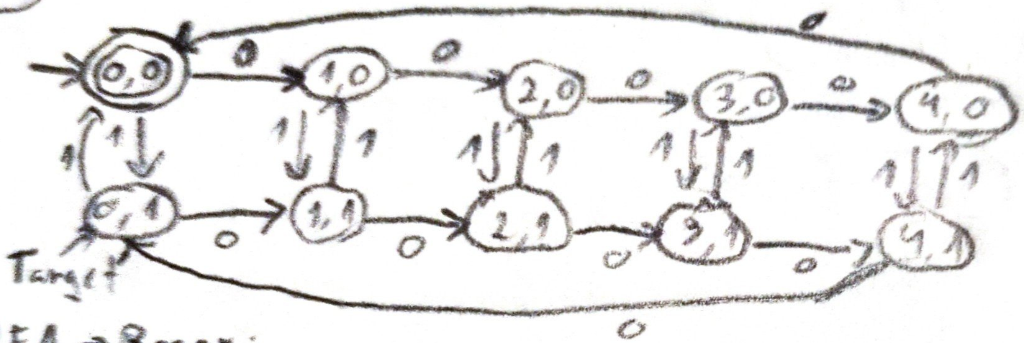


c) $m = \#0 \bmod 5$

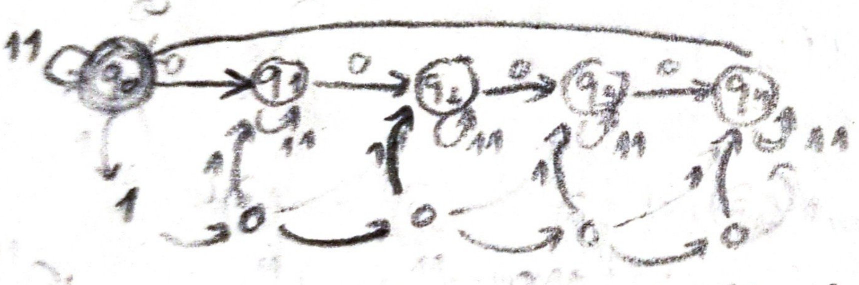
$p = (\#1 \bmod 2 = 0 \; ? \; 0 : 1)$

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

DFA



NFA → Regex:



step 1

Continuar eliminação de estados...

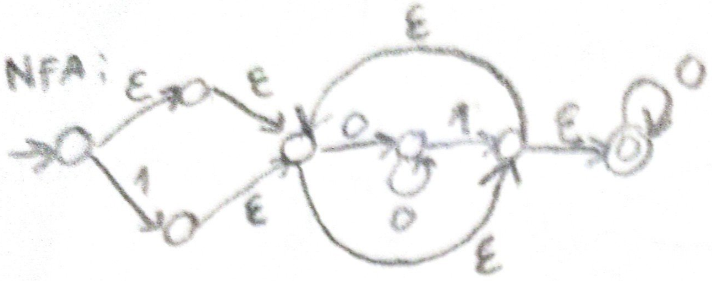
(não estou para isso)

③ a) set de strings binárias sem dois ou mais 1's seguidos

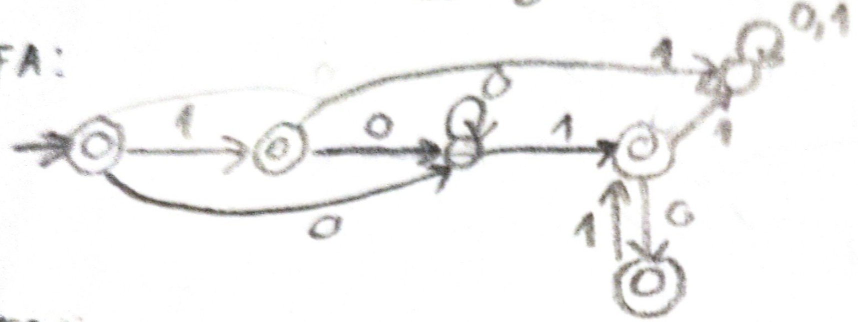
b) set de strings binárias com a substring "000"

c) igual ao a)

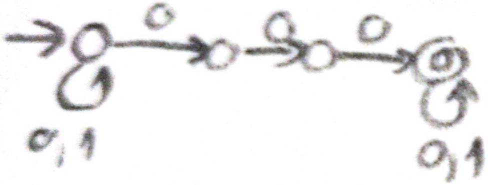
④ a) NFA:



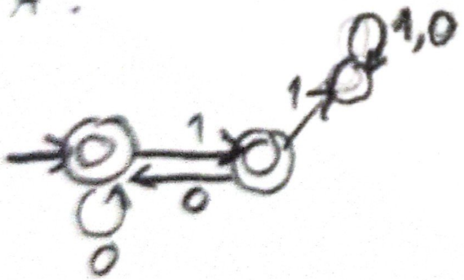
DFA:



b) DFA:

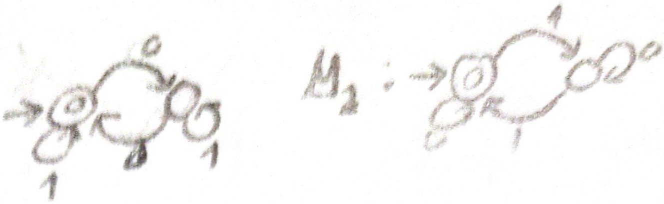


c) DFA:



5) Prova que linguagens regulares são fechadas sob interseção

$L_1 = \{ \#0 \bmod 2 = 0 \}$ M_1: M_2:

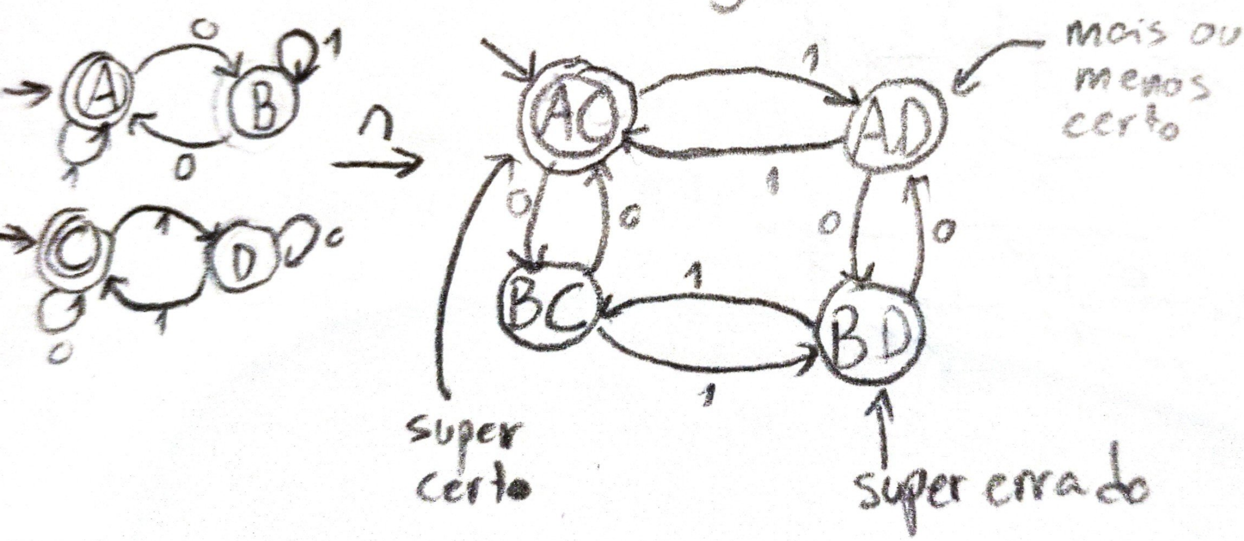


$L_2 = \{ \#1 \bmod 2 = 0 \}$

Se L_1 é aceite 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 aceite 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 2a's adjacentes unido com
" 2b's adjacentes

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