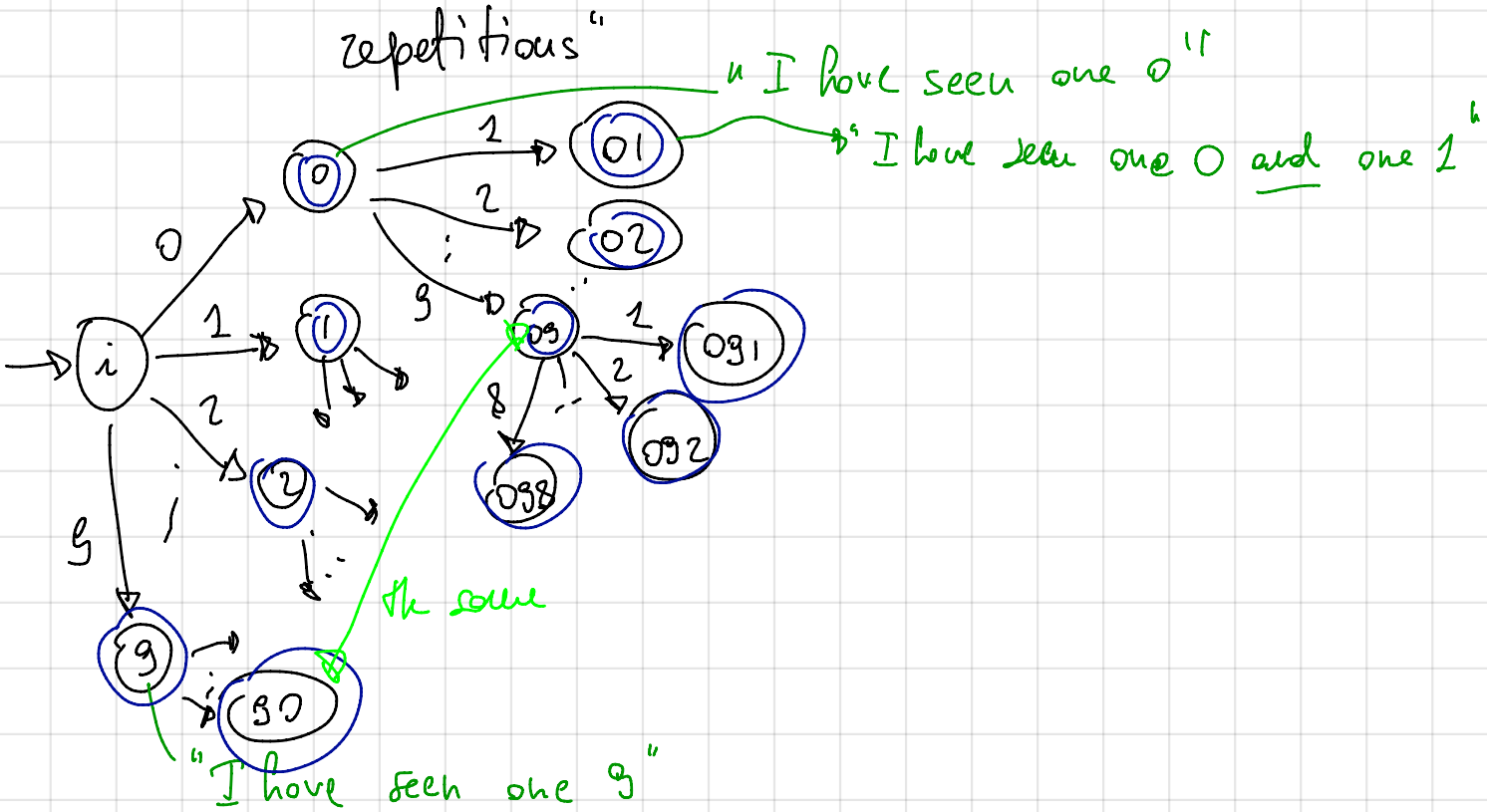
Language = "a string of digits in which there are not repetitious"



Counting. How many states if I don't save?

i 1 digit 2 digits ... 90 digit

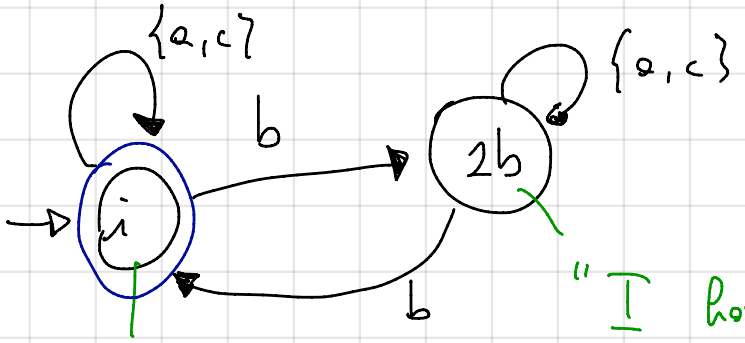
$1 + 10 + 10 \cdot 9 + 10 \cdot 9 \cdot 8 + \dots + 10 \cdot 9 \cdot 8 \dots 1$

$>> 10!$

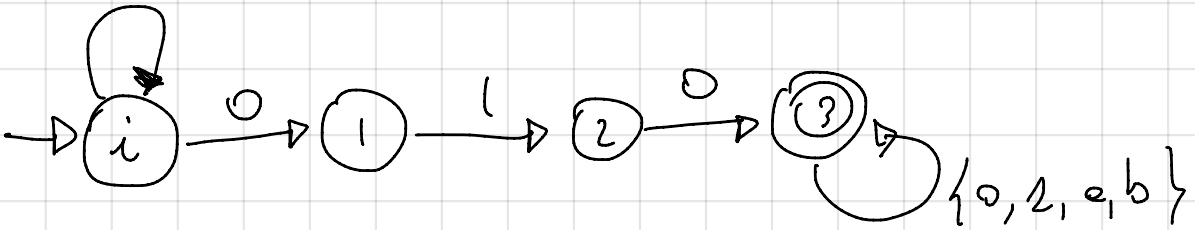
There is space for optimization

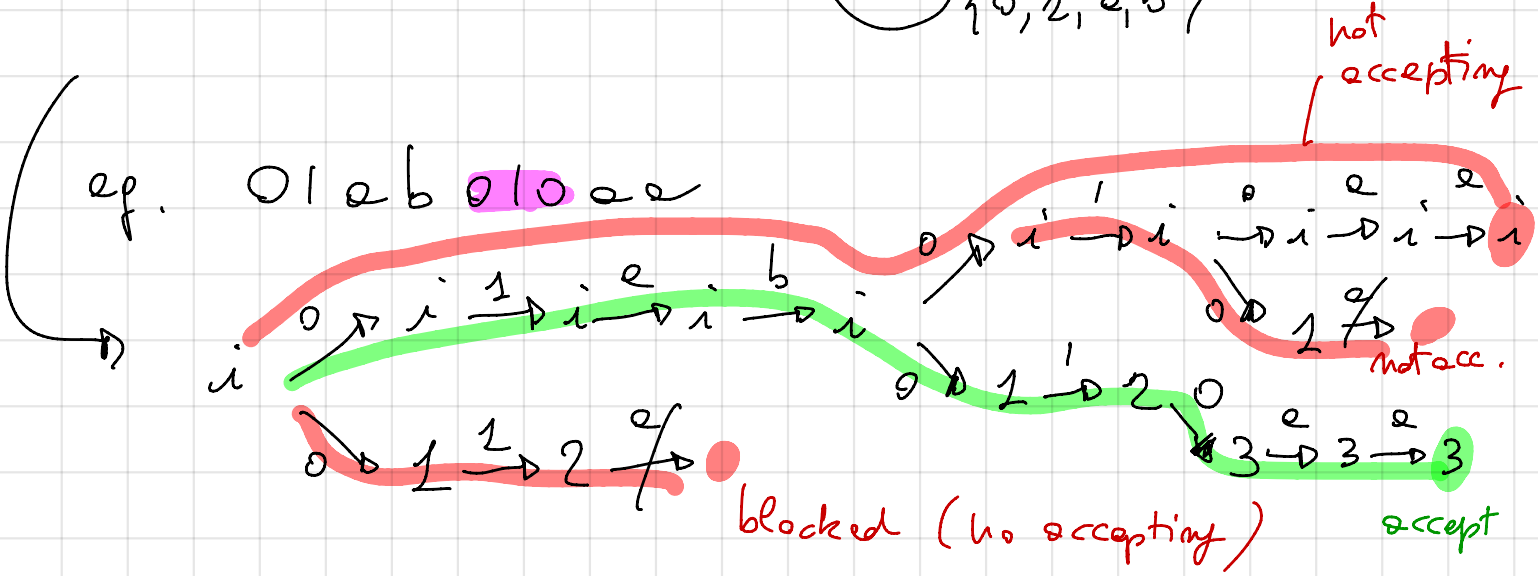
Language $L = \{ x \in \{a,b,c\}^* \mid x$ has an even number of b's $\}$

NOTE: Zero is an even number



$L = \{ x \in \{0,1,a,b\}^+ \mid x$ contains the substring '0 1 0'





The string is accepted

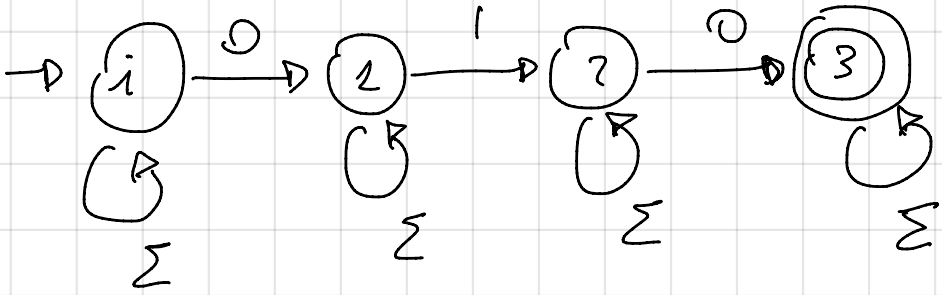
$L = \{ x \in \{0,1,a,b\}^* \mid x$ contains the subsequence '010' $\}$

010

eg $abab10ab$

010

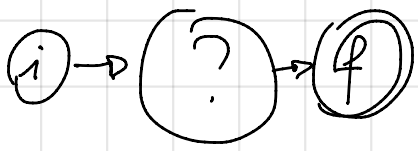
$\Sigma = \{0,1,a,b\}$



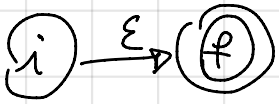
Thompson Alg.

Assumption: only one initial, only one final

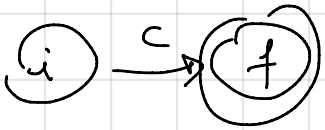
regexp r → NFA



$r = \varepsilon$ → NFA

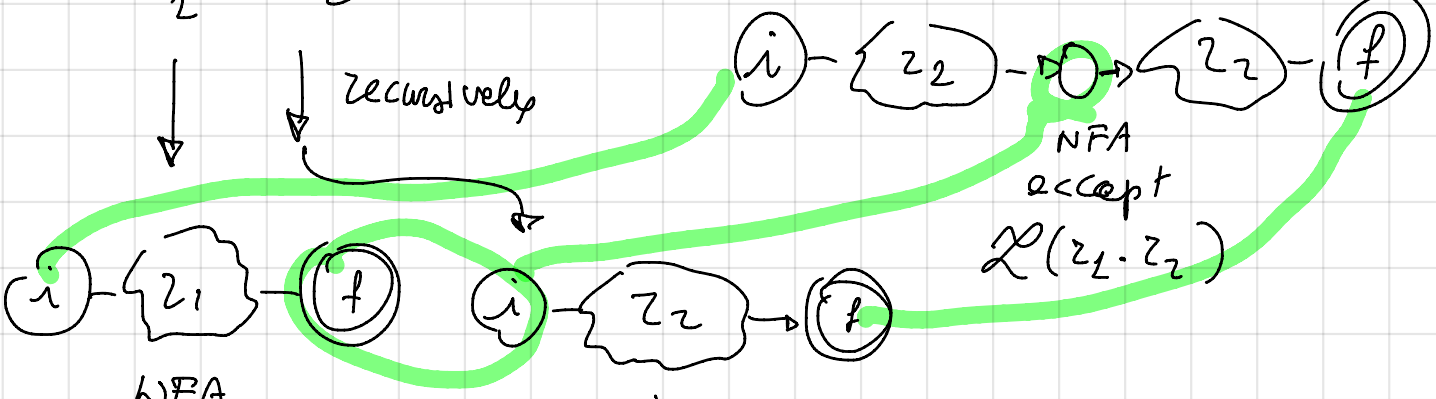


$r = c \in \Sigma$ → NFA



$r = r_1 \cdot r_2$ → NFA

recursively



NFA accepts $\mathcal{L}(r_1)$

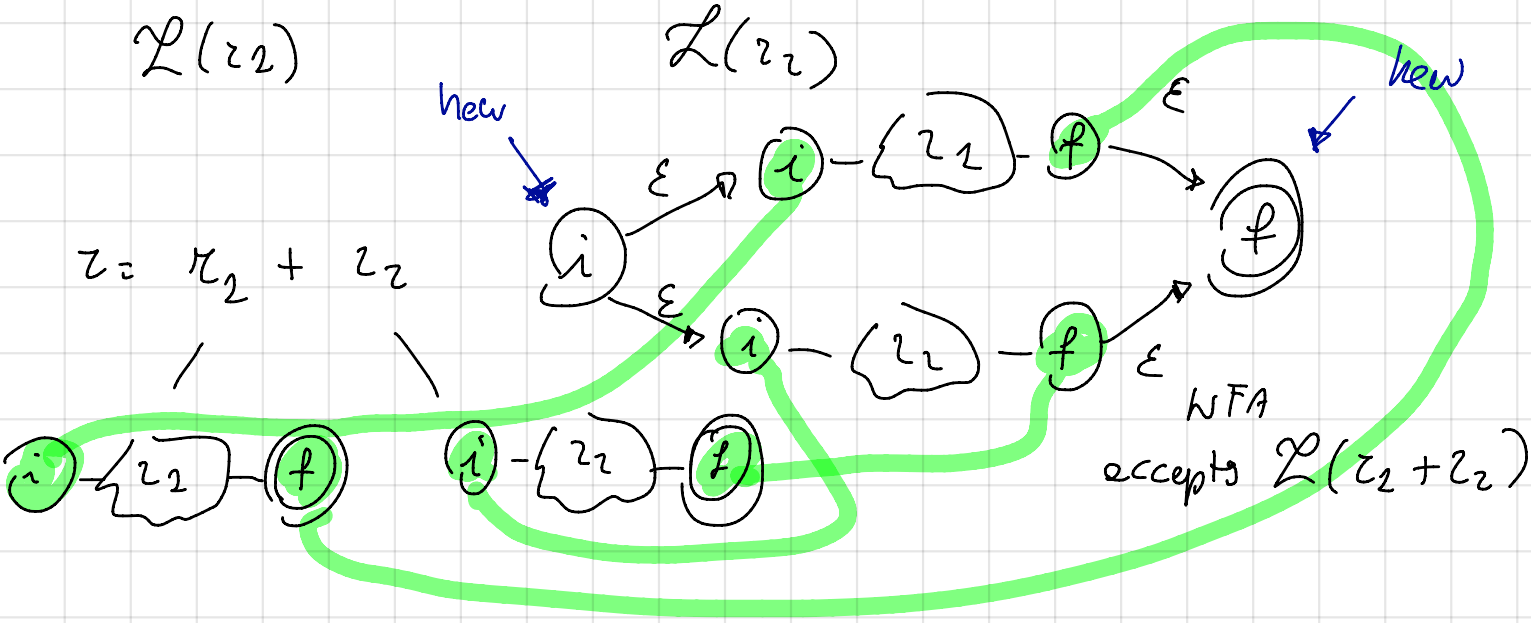
NFA accepts $\mathcal{L}(r_2)$

NFA accept $\mathcal{L}(r_1 \cdot r_2)$

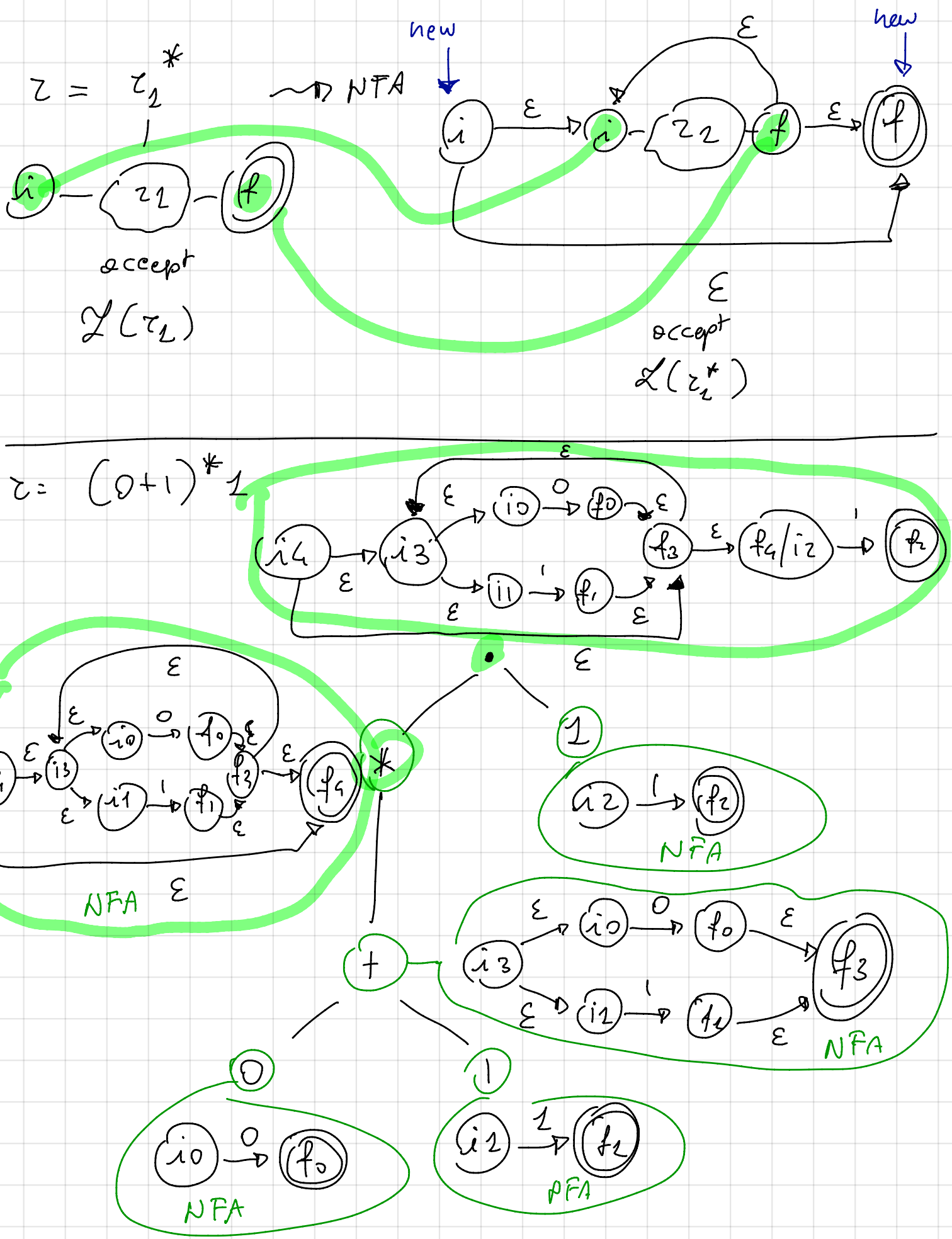
$r = r_1 + r_2$

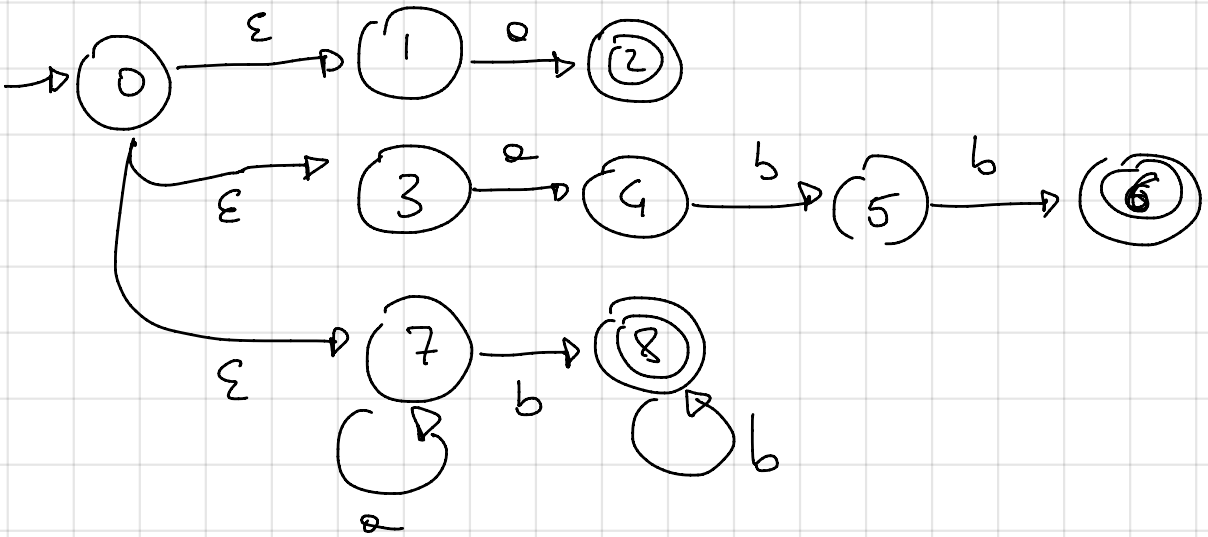
new

new



NFA accepts $\mathcal{L}(r_1 + r_2)$


r = r1*
NFA
new
new
ε
i
r1
f
accept
L(r1)
ε
accept
L(r1*)
r = (0+1)*1
i4
i3
i0
f0
i1
f1
f3
f4/i2
f2
i2
NFA
+
0
1
NFA
NFA
NFA



Last_final = ~~{ 8 }~~

Input_pos_at_Last_final = ~~3~~

INPUT = a a b a $

1 2 3 4

ε-closure({0}) = {0, 1, 3, 7}

δ({0, 1, 3, 7}, a) = {2, 4, 7} → update variables

ε-closure({2, 4, 7}) = = {2, 4, 7}

δ({2, 4, 7}, a) = {7}

ε-closure({7}) = {7}

δ({7}, b) = {8} → update variable

ε-closure({8}) = {8}

δ({8}, a) = { } BLOCKED → OUTPUT

OUTPUT the last seen pattern, that is, state 8, corresponding to TOKEN 3

→ ⟨TOKEN3, aab⟩

LEXEME

→ RESTART THE ALGORITHM BY RESETTING ALL VARIABLES AND CONSUMING THE LEXEME ON THE INPUT:

Last_Final = {2}

Input_pos_at_Last_final = 1

INPUT: a $

1

ε-closure({0}) = {0, 1, 3, 7}

δ({0,1,3,7}, a) = {2, 4, 7} → update variables

ε-closure({2,4,7}) = {2,4,7}

δ({2,4,7}, $)

→ BLOCKED

OUTPUT ⟨TOKEN1, a⟩ STOP