

$R \vee S$

$$(R \wedge \neg S) \vee (\neg R \wedge S)$$

$$(R \vee S) \wedge (\neg R \vee \neg S)$$

$$A + 0 = A$$

$$A \cdot 1 = A$$

$\wedge \equiv \cdot$
$\vee \equiv +$

$$0 \vee A = A$$

$$A: B^n \rightarrow \{0, 1\}$$

$$A + \overline{A} = 1$$

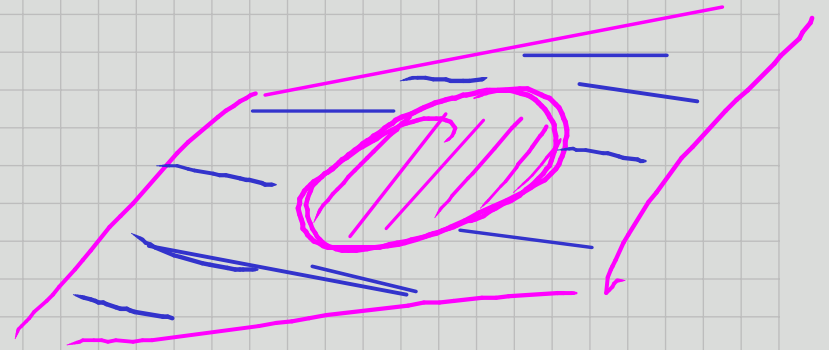
$$A \cdot \overline{A} = 0$$

$$A \vee \neg A = 1$$

$$A \wedge \neg A = 0$$

$$\neg(\neg A) = A$$

$$\overline{A} \equiv \neg A$$



$$\overline{A \cdot B} \equiv \neg(A \wedge B)$$

$$\overline{A} + \overline{B} \equiv \neg A \vee \neg B$$

$$\overline{A + B} \equiv \neg(A \vee B)$$

$$\overline{A} \cdot \overline{B} \equiv (\neg A) \wedge (\neg B)$$

$$\neg(A \wedge B)$$

0	0	1
0	1	1
1	0	1
1	1	0

$$\neg A \vee \neg B$$

1	1	1
1	0	1
0	1	1
0	0	0

$$1 \quad (R \wedge \neg S) \vee (\neg R \wedge S)$$

$$2 \quad (R \vee S) \wedge (\neg R \vee \neg S)$$

$$3 \quad A \wedge (B \vee C)$$

$$4 \quad (A \wedge B) \vee (A \wedge C)$$

$$5 \quad (R \vee S) \wedge \neg R \quad \vee \quad (R \vee S) \wedge \neg S$$

$$6 \quad \neg R \wedge (R \vee S)$$

$$7 \quad (\neg R \wedge R) \vee (\neg R \wedge S) \vee (\neg S \wedge R) \vee (\neg S \vee S)$$

$$\text{III} \quad \textcircled{1} \quad \vee \boxed{(\neg R \wedge S) \vee (\neg S \wedge R)} \quad \vee \quad \textcircled{0}$$

1) ~ ~ ~
 2) ~ ~ ~
 3) ~ ~ ~
 4) ~ ~ ~

① 1 ② 1 ③ 1 ④ 74

truth table

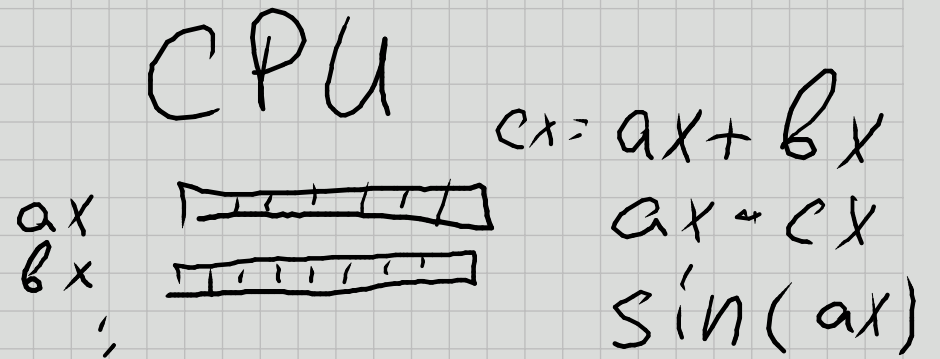
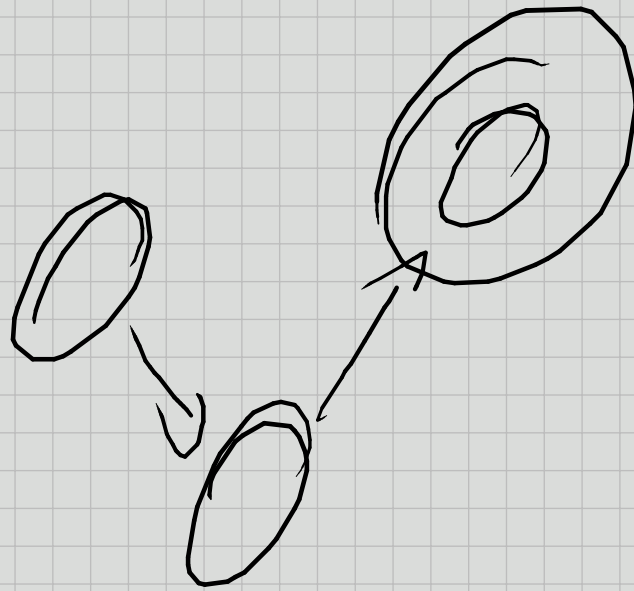
① ① ①
 ① ① 1
 1

1 1 1

$\alpha_1 \wedge \beta_1 \wedge \gamma_1$
 $\alpha_2 \wedge \beta_2 \wedge \gamma_2$

$\alpha_8 \wedge \beta_8 \wedge \gamma_8$

Automatons and Grammars



max

$a_1 \quad a_2 \quad a_3 \quad \dots \quad a_n$

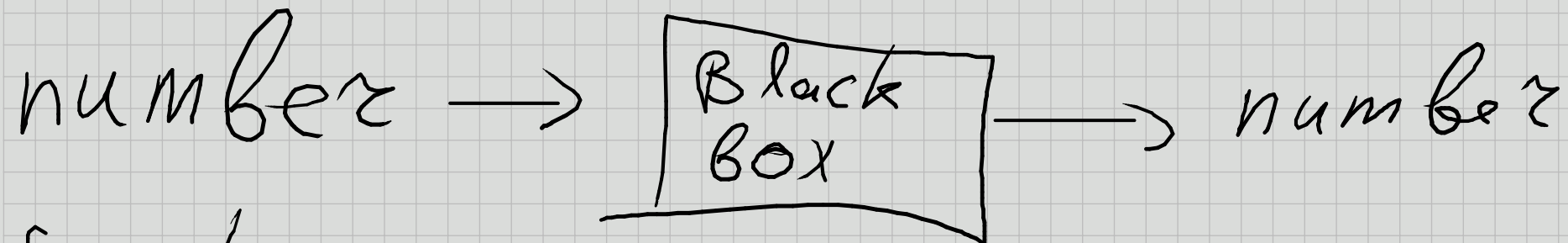
5, 8, 7, ..., 1

$T \sim \# \text{operations}$

Q: what is operation?

Q: what is computer

compute = $\overbrace{\text{вычислитель}}$
 число



function: mapping $A \rightarrow B$

in some sense, computer implements a function

$$\Sigma = \{a, b, \dots\}$$

$$\Sigma_0 = \{0, 1\} \equiv \{a, b\}$$

$$\text{word} = x_1 x_2 x_3 x_4 \dots x_n$$

$$x_i \in \Sigma = \{0, 1\}$$

$$\{\text{word}\}_0^\infty$$

$$\text{word} \equiv \text{string}$$

$$|W| = n, \text{ \# of different words}$$

$|W| = 10$ - all words
length 10

int 32
int 64
uint 10

$\{w \mid |w| = 10\}$
 $\Sigma = \{0, 1\}$

1) 000000 000000

2) 000000 000001

3) 000000 000010

\vdots

\vdots

k) 111111 111111

$\#W \sim 2^n$, $n = |W|$

$\{0, 1, 2\} \Rightarrow 3^n$

Lemma: # all words $|W| = n$
 $= |\Sigma|^n$ $\forall n$
 proof by induction

$n = 1 \quad 2 \quad 3 \quad 4 \quad \dots$

$n = 1, \Sigma = \{a, b\} \hookrightarrow W_1 = \{a, b\}, |W_1| = 2 = 2^1$

$n = k$
 $n = k+1$

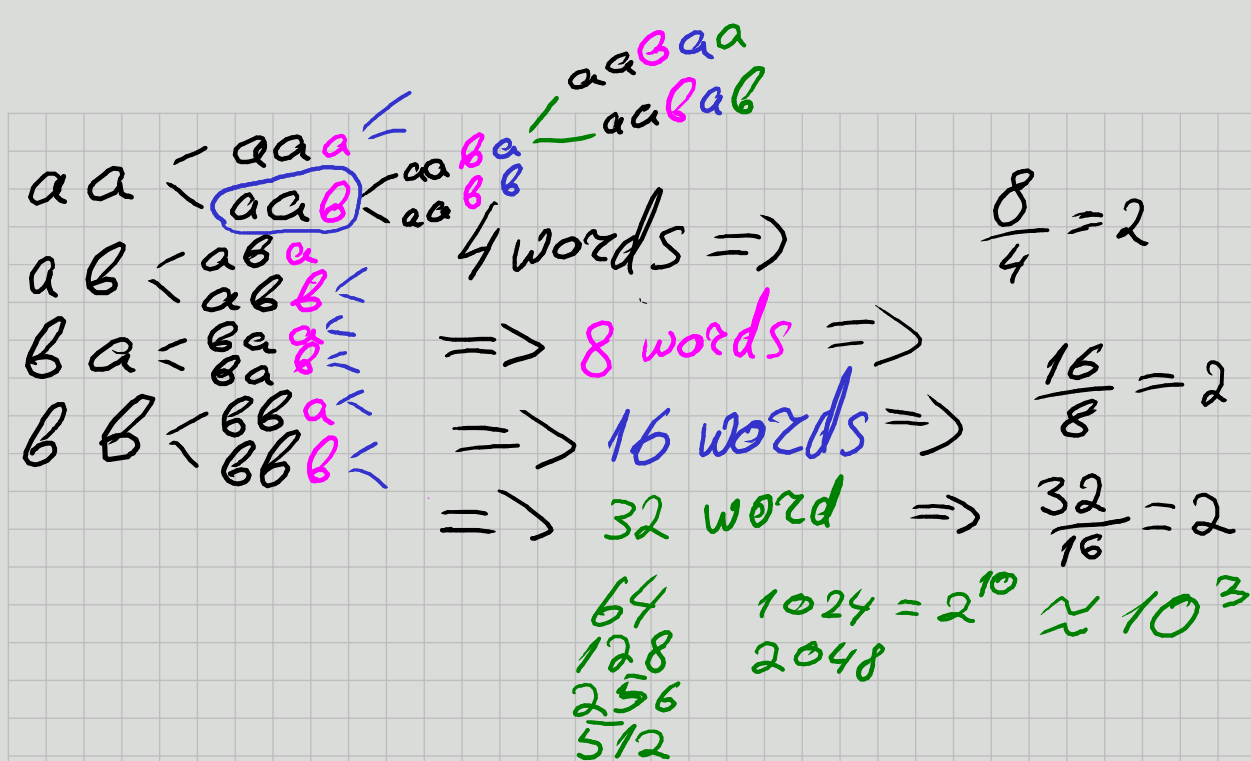
$\hookrightarrow W_k = \{w_1^k, w_2^k, \dots, w_{2^k}^k\}, w_i^k \neq w_j^k \quad \forall i \neq j$

$W_{k+1} = \{w_1^k a, w_1^k b, \dots, w_{2^k}^k a, w_{2^k}^k b, \dots\}$

$d_1 d_2 \dots d_k \beta, \beta \in \Sigma$

$\forall i \quad w_i^k \Rightarrow \begin{cases} w_i^k a \\ w_i^k b \end{cases}$

2 new words for each one old word



$$\frac{8}{4} = 2$$

$$\frac{16}{8} = 2$$

$$\frac{32}{16} = 2$$

$$\begin{array}{l} 1024 = 2^{10} \approx 10^3 \\ 2048 \end{array}$$

$\approx 10^3$

$$\begin{aligned} \# \text{ new words} &= 2^0 \cdot \# \text{ old words} \\ \# \text{ new words} &= 2 \cdot 2^k = 2^{k+1} \end{aligned}$$

$$\{\text{word}\}_0^\infty \rightarrow \{0, 1\}$$

this function defines a Language
finite or infinite

Language - set of words that belong to it.

Language can include words of different length.

Examples of languages:

1: aaabbb

↘ string length = 6

$$\{0, 1\} \equiv \{a, b\}$$

$L = \{^n a a b b ^n\}$ — language that has only one word

2. $L = \{^n a b ^n, ^n b a ^n\}$

3. $L = \{a, b\} = \{\alpha \mid \alpha \in \Sigma\}$
all possible words length = 1

4. $L = \{\epsilon\}$
 ϵ -word that contains no symbols

5. $L = \{\emptyset\}$

What problem are we going to solve?

given a language and a word
figure out if this word belongs
to the language

$$f(\text{word}) = \begin{cases} 0 & \text{if } w \notin L \\ 1 & \text{if } w \in L \end{cases}$$

example 1: bad language filter
for group chats.

example 2: find all occurrences
of a word in a book

example 3: compiler.

encodes, corresponds to

Each binary string generates a subset of a set (of size N).

The opposite is True as well, for each subset there exists a binary string that generates this subset.

We say that there exists isomorphisme between set of all possible binary strings (of length N) and set of all possible subsets of finite set of size N .

The above statement is True for infinite sets too.

M - our set
 $N = |M|$

For finite case: $2^N \longleftrightarrow \{\text{all possible subsets of } M\}$

For both finite and infinite case we write it the following way:

$\{\text{all possible subsets of } M\} \equiv 2^M$

Example: every Language is a subset of a set of all possible words in Σ

Regular Expressions

$$A, B \subseteq \{w \mid w \in \Sigma^*\} = 2^{\Sigma^*}$$

A, B - some regular Languages $A, B \in REG$

2^M
-
set of
all
subsets
of M

1. Union

$$\forall A, B \quad A \cup B = \{w \mid w \in A \text{ or } w \in B\}$$

2. Concatenation

$$A \circ B = \{xy \mid x \in A, y \in B\}$$

3. Star (a.k.a. Iteration)

$$A^* = \{x_1 x_2 \dots x_k \mid \text{each } x_i \in A \text{ for } k \geq 0\}$$

$|M| = n$

1	2	...	n
0	0		0
1	1		1
<hr/>			
0	1	0	1
2	4	6	
0	0	...	0
0	0	...	0
0	0	...	0
...
1	1	...	1

$k=3 \rightarrow X_1 X_2 X_3$ - 3 words

$k=0 \rightarrow \Sigma$

$$A = \{\emptyset\}$$

~~$$A^* = \{\emptyset\}$$~~

$$A^* = \{\epsilon\}$$

By definition, empty Language is considered to be Regular.

$$L = \{\emptyset\}$$

By definition, Languages that consist of one symbol from Alphabet are considered to be Regular. (word has length one)

$$L = \{a\}, L = \{b\}$$

By definition, All Languages that can be derived from Regular Languages by applying Regular Operations are considered to be Regular too.

$$L_1 = \{a\}, L = L_1^* \in REG$$

$$(\{a\}\{b\})^* \cup \{b\}$$

$$\varepsilon \quad (aa/b)^* \varepsilon (a|aa)$$

\parallel \parallel
 $\{a\} \cdot \{a\}$ $\{b\}$

$\varepsilon \in REG$

Let A be a Language that consists of finite number of words.
Is A Regular or Irregular?

Can we prove it?

Statement 1: any word alone represents a Regular Language.

$$W = a_1 a_2 a_3 \dots a_n$$

$$L_1 = \{a_1\}, L_2 = \{a_2\}, \dots, L_n = \{a_n\}$$

$$L_w = \underbrace{L_1 \cdot L_2 \cdot \dots \cdot L_{n-1}}_{L_{n-1}} \cdot L_n = \{a_1 a_2 \dots a_n\}$$

Statement 2: $L = \{w_1, w_2, \dots, w_k\} = L_{w_1} \cup L_{w_2} \cup \dots \cup L_{w_k}$

What happens if we apply Regular Operations infinite number of times?

$$\bigcup_{n=1}^K \{a^n\} \equiv a \cup aa \cup aaa \cup \dots \cup a^K$$

K - some fixed number

\uparrow REG

$$a \mid aa \mid aaaa \in REG$$

$a \mid a^2 \mid a^3 \mid$

$$\{aa\} \in REG$$

$$\{a\} \circ \{a\} = \{aa\}$$

$aaaa$
Proof by construction

$$\{aba\} \in REG$$

$$\{a\} \circ \{b\} \circ \{a\} = \{aba\}$$

$$(\{a\} \circ \{b\}) \circ \{a\} = \{ab\} \circ \{a\} = \{aba\}$$

$$(x \circ y) \circ f(x, y)$$

$$\begin{array}{r} 513 \times \\ \times 271 \times \\ \hline \times 380 \end{array}$$

$$\begin{array}{r} 513 \\ + 35910 \\ + 102600 \\ \hline 139023 \end{array}$$

$$a \times b \times c = (a \times b) \times c = d \times c = e$$

$$\{a|b\} \stackrel{?}{\in} REG$$

$$L = \{a, b\}$$

$$\{a\} \cup \{b\} = \{a, b\}$$

$$\{aa|ba\} \stackrel{?}{\in} REG$$

$$L = \{aa, ba\} =$$

$$= \{aa\} \cup \{ba\}$$

$$\{a\} \circ \{a\} \quad \{b\} \circ \{a\}$$

$$\{ab|b|aaa\} \stackrel{?}{\in} REG$$

$$L = \{ab, b, aaa\} =$$

$$= (\{ab\} \cup \{b\}) \cup \{aaa\} =$$

$$= \{ab|b\} \cup \{aaa\}$$

$$\{bb|ba|bab|aaa\}$$

$$L = \{bb|ba|bab\} \cup \{aaa\}$$

$$= L_3 \cup \{aaa\} = L_3 \cup L_1 = L_4$$

$$L_5 = \{w_1|w_2|w_3|w_4|w_5\} =$$

$$= \{w_1|w_2|w_3|w_4\} \cup \{w_5\} =$$

$$= L_4 \cup L_1$$

$$L_{100} = L_{99} \cup L_1$$

n *fixed number*

$$\bigcup_{k=1}^n W_k = W_1 \cup W_2 \cup W_3 \cup \dots \cup W_{n-1} \cup W_n$$

Formally speaking we define a family of sets parametrized by index "k", then we unite by this index

$$\bigcup_{k=1}^{\infty} W_k = W_1 \cup W_2 \cup \dots \cup W_n \cup \dots$$

$$\bigcup_{k \in \mathbb{N}}$$

Infinite Union ALWAYS yields a Regular Language. True or False?

$\forall L \Rightarrow L = \bigcup_{w \in L} \{w\}$ Any Language is a union of words that make this language

Property that either holds or not holds for finite/infinite cases.

Example why proof by induction is not always applicable

Set is bounded if:

$$M = \{m_1, m_2, \dots\}, m_i \in \mathbb{R}$$

satisfied
violated

$$\exists x \in \mathbb{R} : \forall m \in M \hookrightarrow |m| \leq x$$

example

Bounded ~ There exists Maximum Element

$$M = \{1, 2, 3, 4, 5\} \quad \begin{array}{l} x=5 \\ x=5.5 \end{array}, \quad x=6$$

Any set that contains finite number of elements IS bounded. True or False?

Base

$$1. N = |M| = 1; M = \{m_1\}; |m_1| \leq |m_1| + 1 \equiv X$$

step

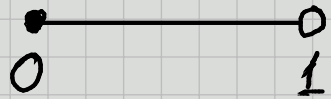
2. Assume $N = |M| = n$ statement is true.

$$M_{n+1} = M_n \cup \{m_{n+1}\}; \exists X_n : \forall m \in M_n \hookrightarrow |m| \leq X_n$$

$$X_{n+1} = \max \{X_n, |m_{n+1}|\}$$

$$[0; 1) \equiv [0; 1] \setminus \{1\}$$

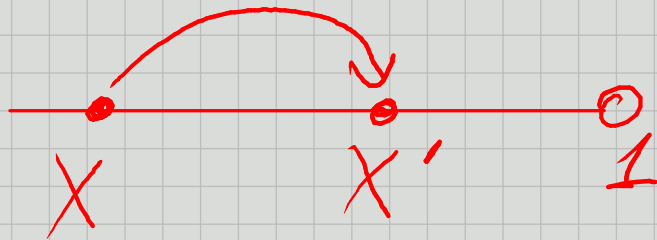
$$\text{Max}[M] \in M$$



$$x = 0,999$$

$$\Delta = 1 - x = 0,001$$

$$x' = x + \frac{1}{2} \Delta = 0,999 + 0,0005 = 0,9995$$



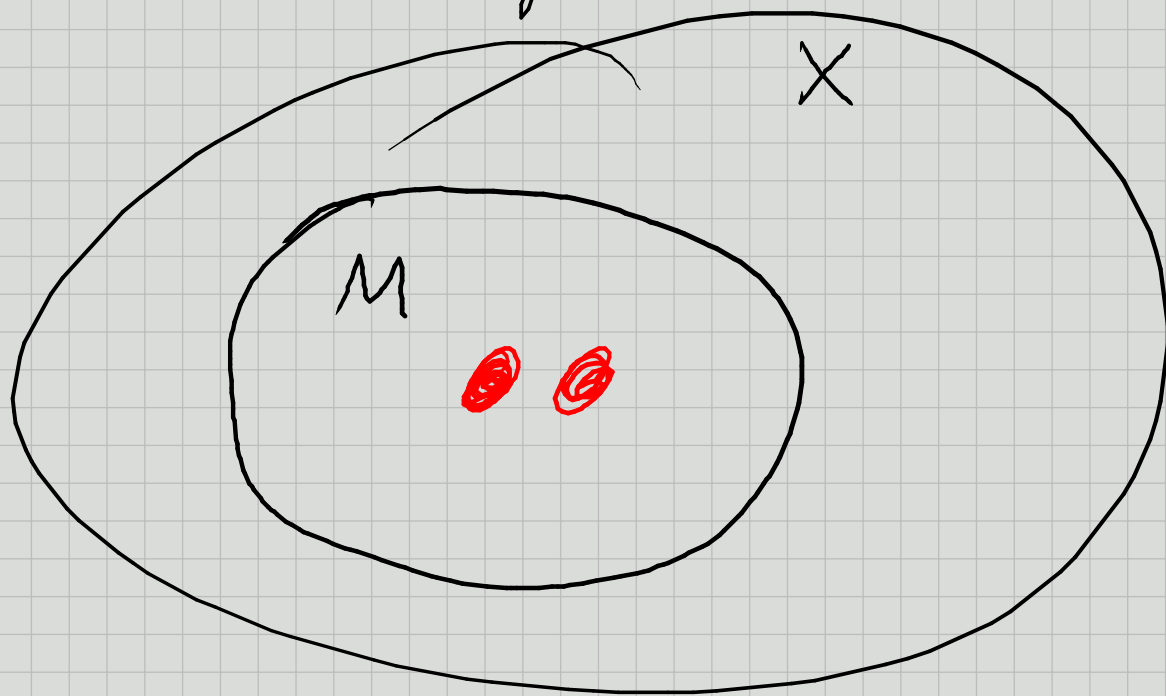
Set has infinite number of elements $\stackrel{?}{\implies}$ Bounded

~~We know that every infinite set is bounded.~~

$$M = \mathbb{N}$$

X - entire space

M



$$f(\cdot) \quad \begin{array}{l} x \rightarrow x \\ x, x \rightarrow x \\ x, x, x \rightarrow x \end{array}$$

$$f: M \rightarrow ?$$

$$M \rightarrow M$$

$$M \rightarrow X$$

$$L_1 \cup L_2 \quad U \equiv f(\cdot, \cdot) = f(x, y) \rightarrow z$$

$$\forall x, y \in M \Rightarrow f(x, y) \in M$$

f is closed in M ; M is closed under f

$M, |M| = \infty \Rightarrow M \text{ not bounded}$

$$[0; 1] \subset \mathbb{R}$$

Файл
Инфинит

Expression
Language
Automaton
Grammar

Technically these
are different

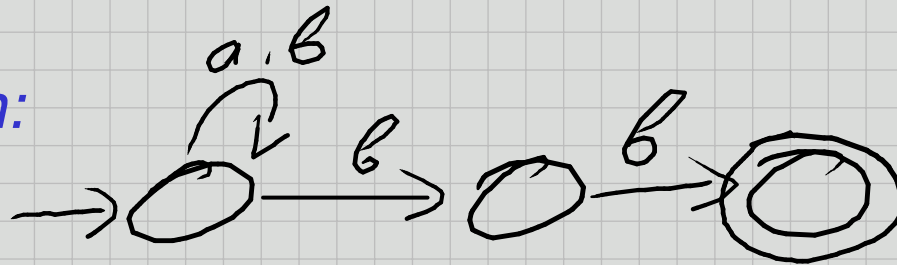
Expression:

$(a|b)^*bb$

Language:

$\{w \mid \text{ends with "bb"}\}$

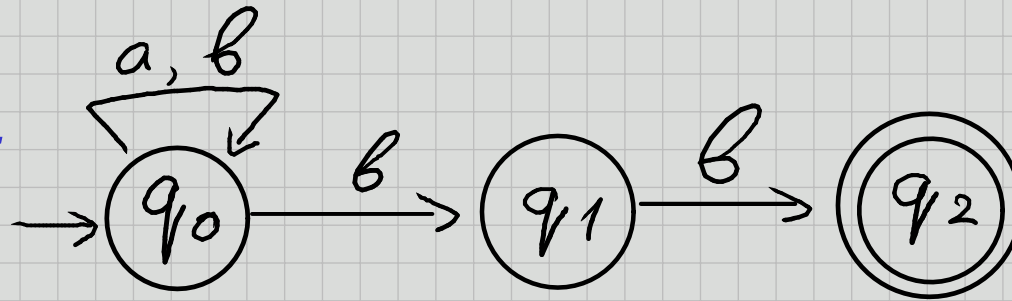
Automaton:



Grammar:

$S \rightarrow aS \mid bS \mid A$
 $A \rightarrow bb$

Automaton:



aa**bb**
✓

ba**b**
✗

abb**a**

Finite Automaton $(Q, \Sigma, q_0, F, \delta)$

$Q = \{q_1, \dots, q_n\}$ - Finite set of all states

$\Sigma = \{a, b, \dots\}$ - Alphabet

δ - Transition Matrix $\delta: Q \times \Sigma \rightarrow Q$

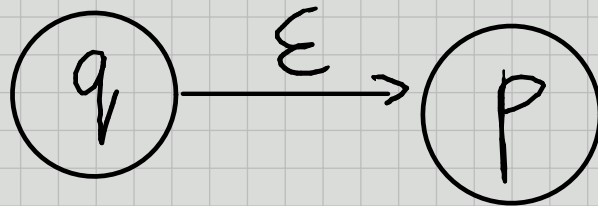
q_0 - starting pos.
 F - accept ✓ states

Automaton accepts a word: we say automaton accepts a word if there exists a path from "starting state" to some "accepts state" that consumes the word and where each transition $q_i \rightarrow q_j$ is accompanied by reading the next symbol of the word.

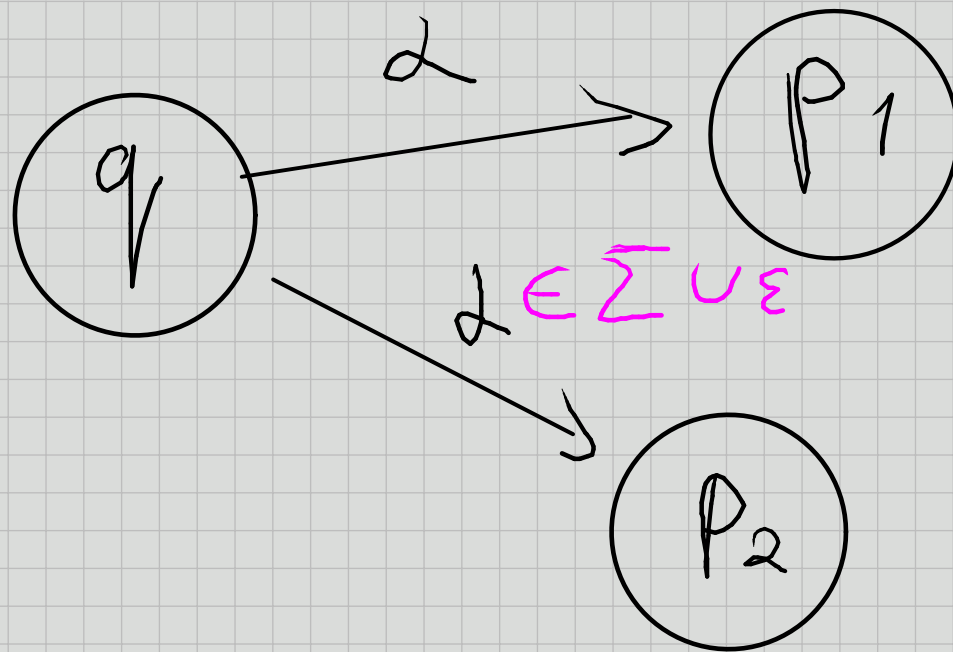
$$\delta(q_i, a) = q_j$$

Empty-word transition:

$$\delta(q_i, \epsilon) = p_j$$



Empty word transitions don't require reading any symbols from the word and don't consume any symbols of the word.



Non-deterministic transitions

$$\forall q \in Q, \forall a \in \Sigma$$

$$|\delta(q, a)| \leq 1$$

Automatons can be deterministic and non-deterministic

DFA - Determ. Finite Automaton

NFA - Nondeter. Finite Automaton

FA - Finite Automaton (any)

$$\delta : Q \times \Sigma \rightarrow 2^Q$$

$L \subset REG$

DFA NFA

$$L(M) = \{ w \mid M \text{ accepts } w \}$$

$$M = \text{DFA} \iff L(M) \subset REG$$

$$M = \text{NFA} \iff L(M) ?$$

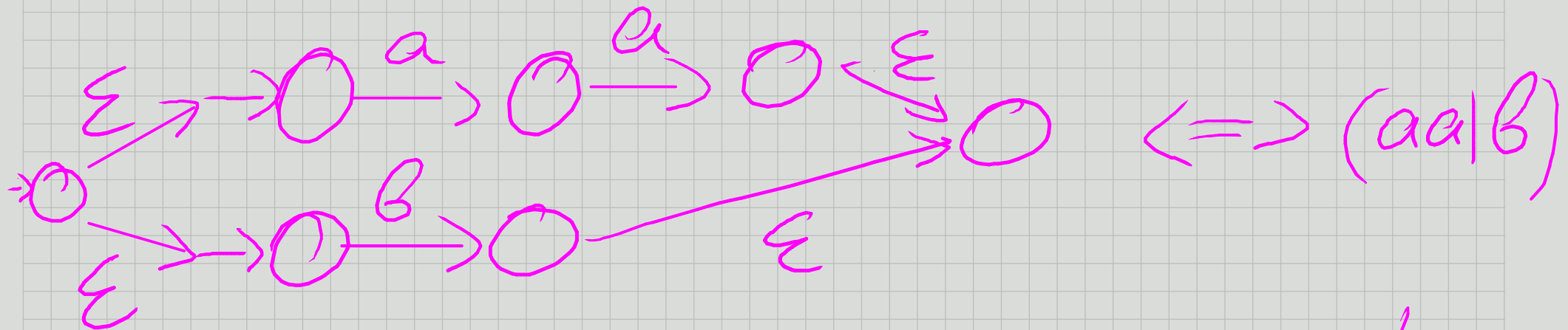
$$A(w) \rightarrow \{0, 1\}$$

$$\{w \mid A(w) = 1\} = L(A)$$

PB \rightarrow DKA (zus)
 RE \rightarrow DFA (eng)

$\Delta (a a | b)^* b (a | b)^* a \Delta$
 0 1 2 3 4 5 6 7 8

$\Delta a a b b b b a a \Delta$
 0 1 2 5 6 7 8



firstpos, lastpos, nullable, followpos

RE = {w} - finite or infinite

RE = aa|b {aa, b} fp = {0, 2} lp = {1, 2}

firstpos - set of symbols that can be the first symbol of words of the language

$$\{\alpha \mid \alpha x \in L; \alpha \in \Sigma, x \in \Sigma^*\}$$

lastpos - ...

$$\{\alpha \mid x \alpha \in L; \dots\}$$

nullable - $\varepsilon \in L$

$$\begin{cases} 1, & \varepsilon \in L \\ 0, & \text{otherwise} \end{cases}$$

followpos -

$$\text{followpos}[\alpha] = \{\beta \mid x \alpha \beta y \in L; x, y \in \Sigma^*, \alpha, \beta \in \Sigma\}$$

$$RE = (aba \mid bab \mid ba^*) \quad \Sigma = \{a, b\}$$

$$fp = \{id \mid (\{id\} w) \in L\} \quad \Sigma_{id} = \{0, 1, 2, \dots, 7\}$$

$$L = \{aba, bab, b, ba, baa, baac, \dots\}$$

$$L = \{[012], [345], [6], [67], [677], \dots\}$$

$$fp = \{0, 3, 6\} \quad nullable = false$$

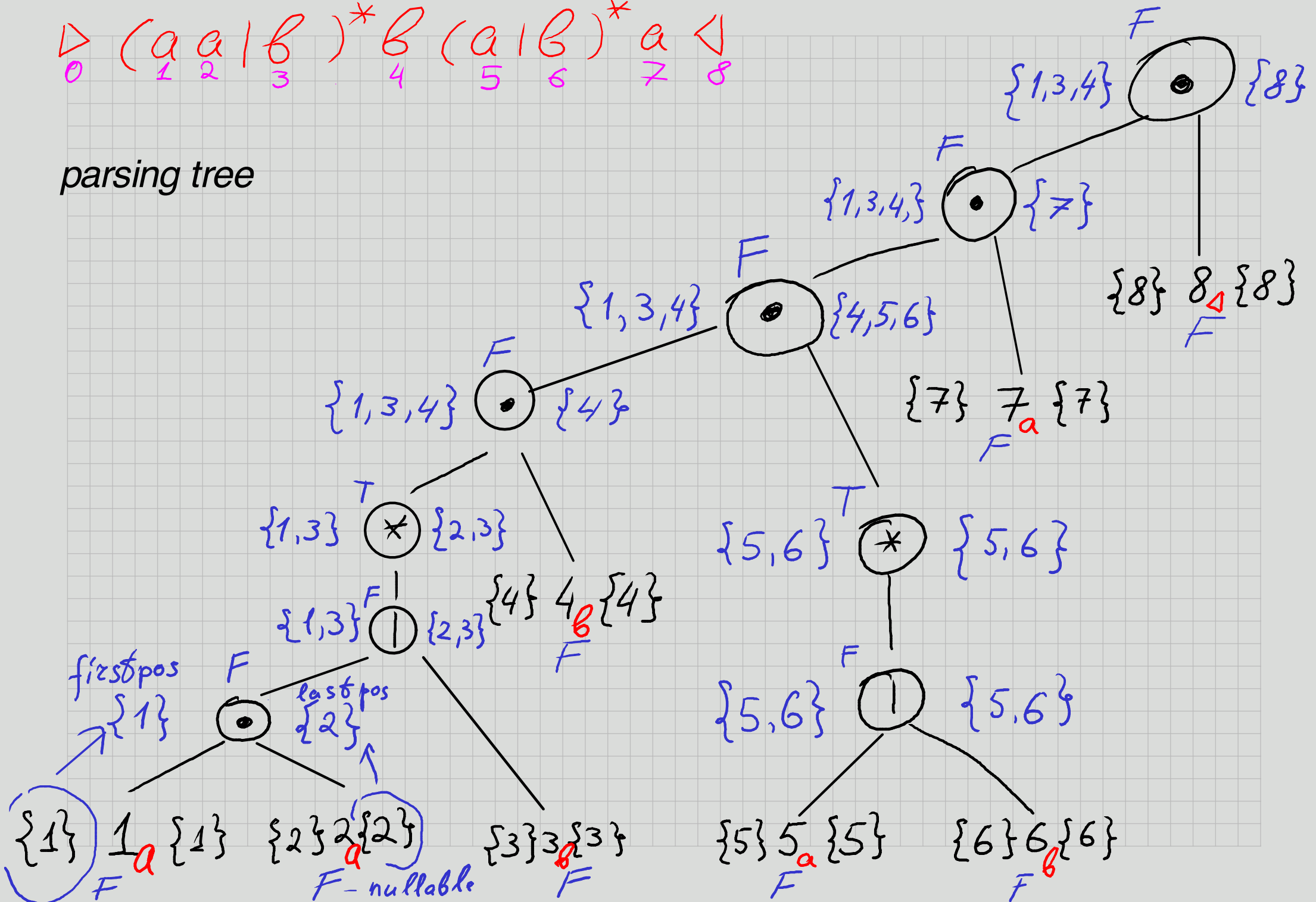
$$lp = \{2, 5, 6, 7\}$$

$$followpos[i] = \{j \mid x\{i\}\{j\}y \in L\}$$

0	1	3	4	6	△, 7
1	2	4	5	7	△, 7
2	△	5	△		

$\triangleright (a a b)^* b (a b)^* a \triangleleft$
 0 1 2 3 4 5 6 7 8

parsing tree



followpos table

0 \triangleright 1_a 3_b 4_b

1_a 2_a

2_a 4_b 1_a 3_b

3_b 4_b 1_a 3_b

4_b 5_a 6_b 7_a

5_a 7_a 5_a 6_b

6_b 7_a 5_a 6_b

7_a 8 \triangleleft

$\triangleright (a a b)^* b (a b)^* a \triangleleft$

followpos table

0 \triangleright 1_a 3_b 4_b

1_a 2_a

2_a 4_b 1_a 3_b

3_b 4_b 1_a 3_b

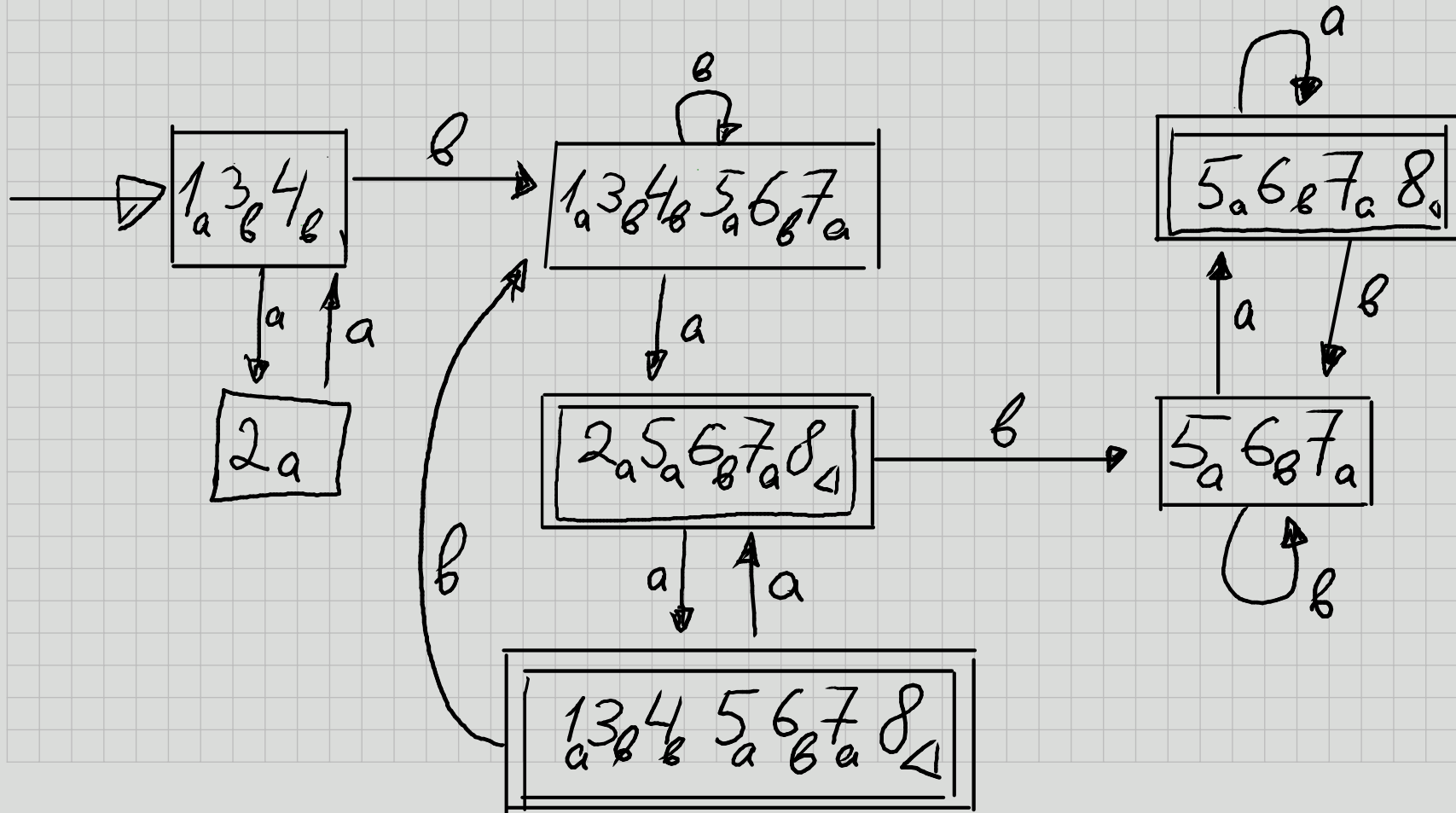
4_b 5_a 6_b 7_a

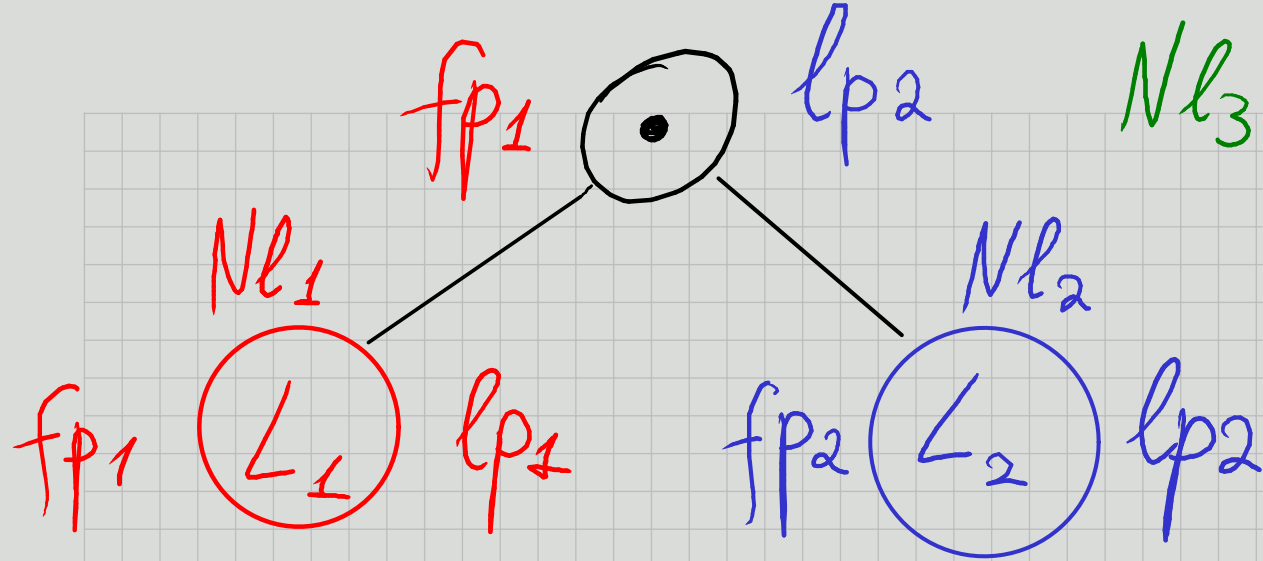
5_a 7_a 5_a 6_b

6_b 7_a 5_a 6_b

7_a 8_a

aa bbb aa





$$Nl_3 = Nl_1 \wedge Nl_2$$

$$fp_3 = \begin{cases} fp_1, & Nl_1 = F \\ fp_1 \cup fp_2 & T \end{cases}$$

$$lp_3 = \begin{cases} lp_2, & Nl_2 = F \\ lp_1 \cup lp_2 & T \end{cases}$$

$$L_1 = \{ fp_1 \cdot x \cdot lp_1 \} \quad L_2 = \{ fp_2 \cdot y \cdot lp_2 \}$$

$$L_3 = L_1 \cdot L_2 = \{ \underbrace{fp_1 \cdot x \cdot lp_1}_{\text{red}} \underbrace{fp_2 \cdot y \cdot lp_2}_{\text{blue}} \}$$

$$W_1 W_2 = \varepsilon \iff \begin{cases} W_1 = \varepsilon \\ W_2 = \varepsilon \end{cases}$$

both words
have to be
empty

$$\varepsilon \in L_1 \hookrightarrow L_1 \cup \{ \varepsilon \} = L_1$$

$$L_1 L_2 = (L_1 \cup \{ \varepsilon \}) L_2 = L_1 L_2 \cup \{ \varepsilon \} L_2 = L_1 L_2 \cup L_2$$

$\{fp_1, fp_2\}$ \bigcirc $\{lp_1, lp_2\}$

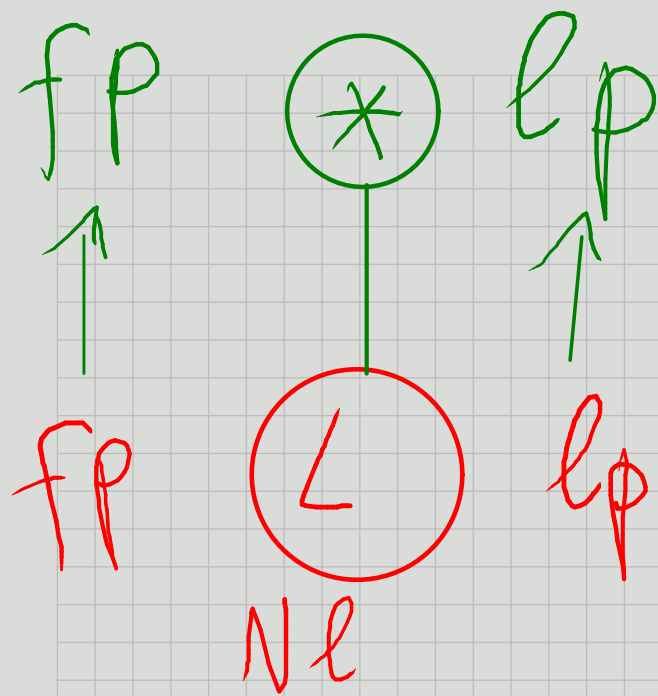
$$NL_3 = NL_1 \vee NL_2$$

fp_1 \bigcirc lp_1

fp_2 \bigcirc lp_2

$$L_1 = \{fp_1 \cdot x \cdot lp_1\} \quad L_2 = \{fp_2 \cdot y \cdot lp_2\}$$

$$L_3 = L_1 \cup L_2 = \{fp_1 \cdot x \cdot lp_1, fp_2 \cdot y \cdot lp_2\}$$



$$L^* = \varepsilon \cup L \cup LL \cup LLL$$

$$Nl = True$$

$$LLL = \{ \underbrace{fp}_{\text{red circle}} x lp, \quad fp y lp, \quad fp z \underbrace{lp}_{\text{red circle}} \}$$

$$L_n = \Sigma^* b \Sigma^n$$

$$P < 2^n$$

$aaaa \rightarrow a$
 $baaa \rightarrow a$
 $abaa \rightarrow a$

Language that consists of all words that have symbol "b" n positions before the last.

Lemma:

DFA that recognizes this language has at least 2^n states.

$$(abbb|b)^* | b^*) aa$$

$\triangleright [(abb|b)^* | b^*] aa \triangleleft$
 0 1 2 3 4 5 6 7 8

followpos

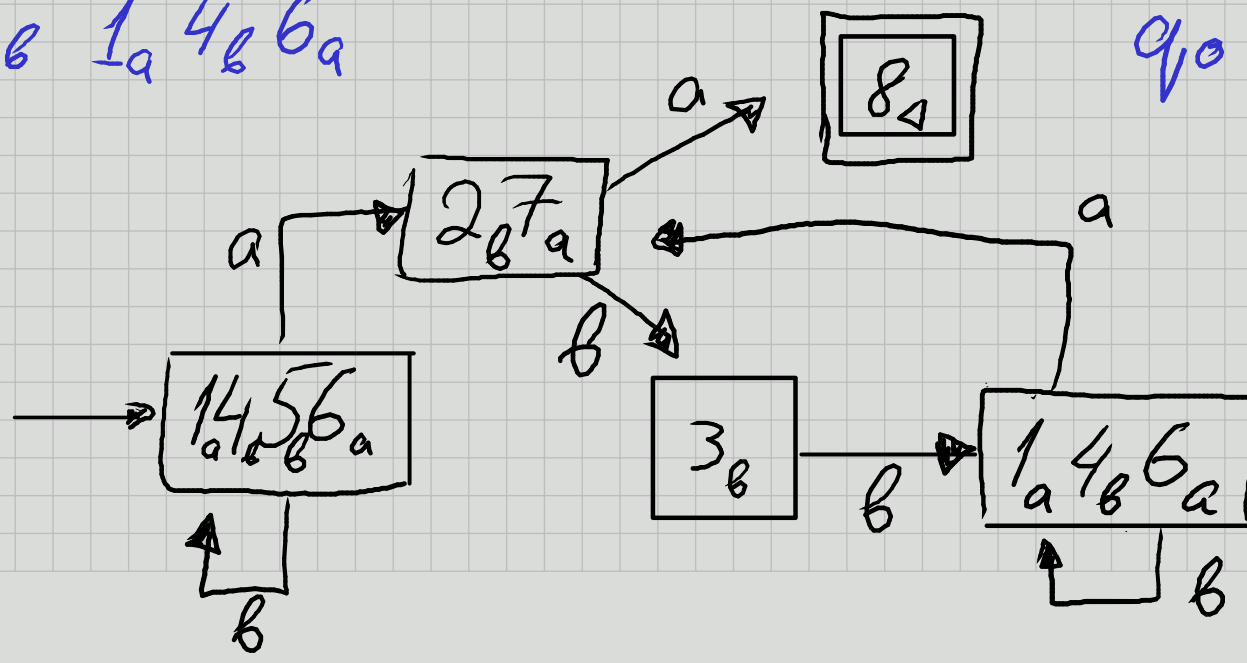
0 $1_a 4_b 5_b 6_a$
 1 $a 2_b$
 2 $b 3_b$
 3 $b 1_a 4_b 6_a$
 4 $b 1_a 4_b 6_a$

5 b 5 $b 6_a$
 6 a 7 a
 7 a 8 a

$\rightarrow abb$
 $abbaabb$

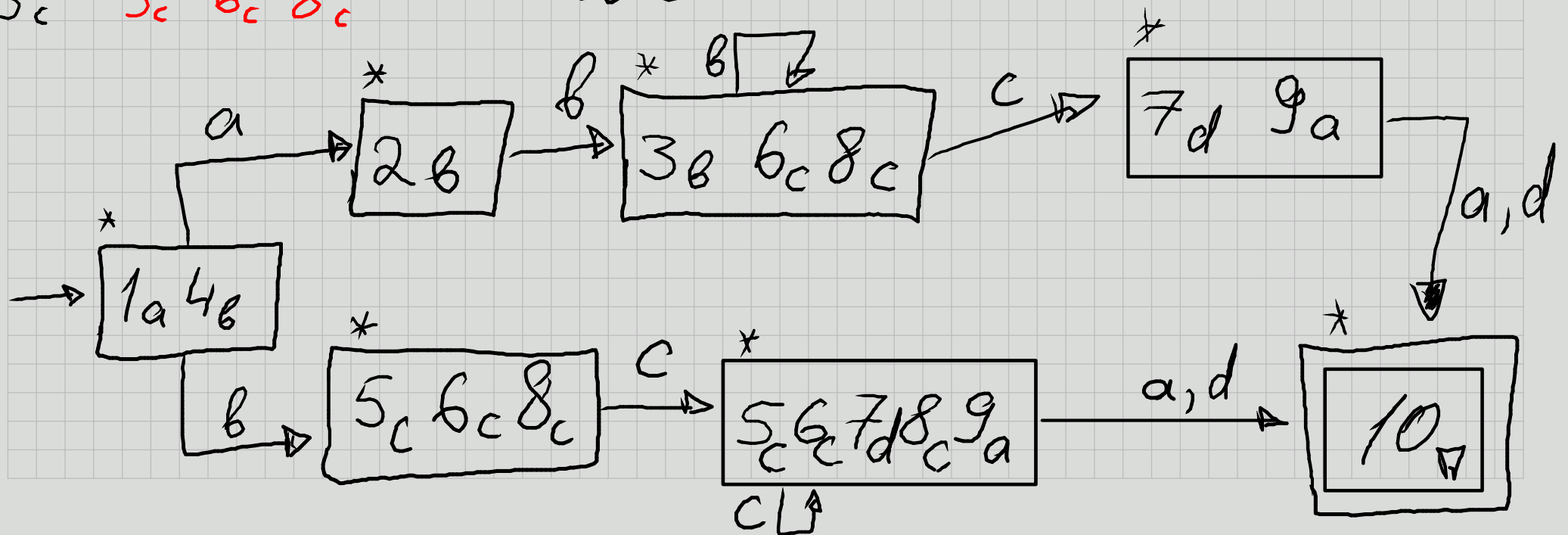
$q_0 \xrightarrow{abb} 146$
 $q_0 \xrightarrow{abbaabb} 146$

$q \xrightarrow{w} p$



0 \triangleright 1_a 4_e 6_c 7_d
 1_a 2_b 7_d 10 _{Δ}
 2_b 3_b 6_c 8_c 8_c 9_a
 3_b 3_b 6_c 8_c 9_a 10 _{Δ}
 4_e 5_c 6_c 8_c
 5_c 5_c 6_c 8_c

a b c d



$a^n b^n$ ww ww^R

$$|a| = |b|$$

~~REG~~

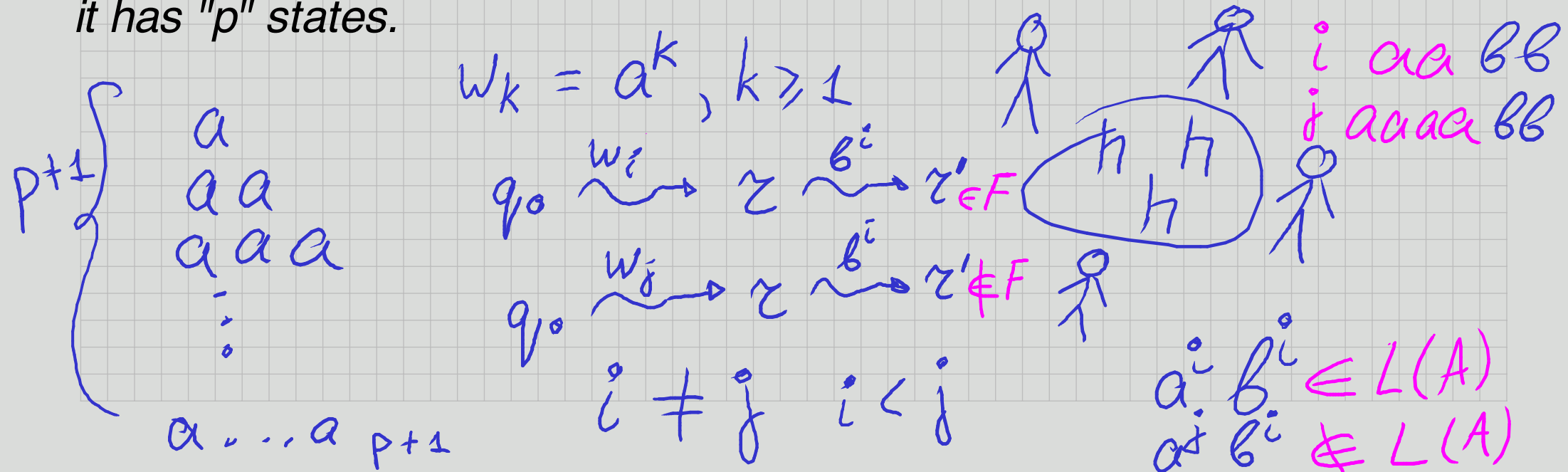
$a^n b^n$ is not regular.

Proof:

все переходы однозначные

F - set of accepting states

Lets assume the language IS regular. Then there exists DFA that recognizes this language. DFA has finite number of states, lets say it has "p" states.



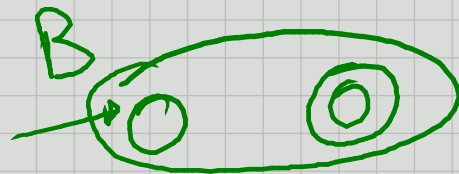
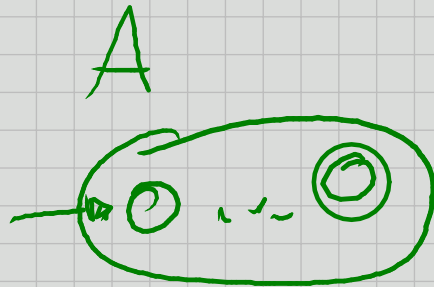
RE \rightarrow DFA
NFA \rightarrow DFA
minimization of DFA
NFA \rightarrow RE

Grammars
classification (0-3)
NFA \leftrightarrow Type(3) G
PDA \leftrightarrow Type(2) G
TM \leftrightarrow Type(0, 1)

RE \rightarrow NFA

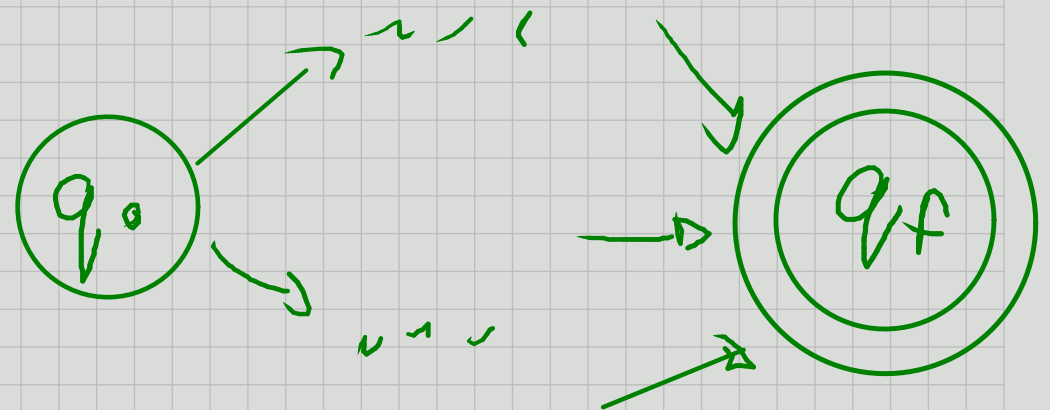
Inductive construction of NFA

Union, Concatenation, Star



A, B

$A \cup B$
 $A \mid B$
 A^*



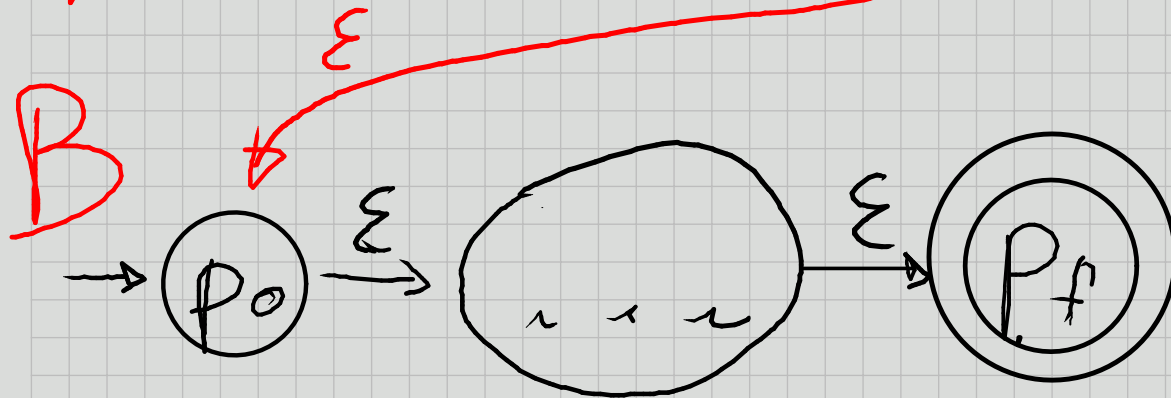
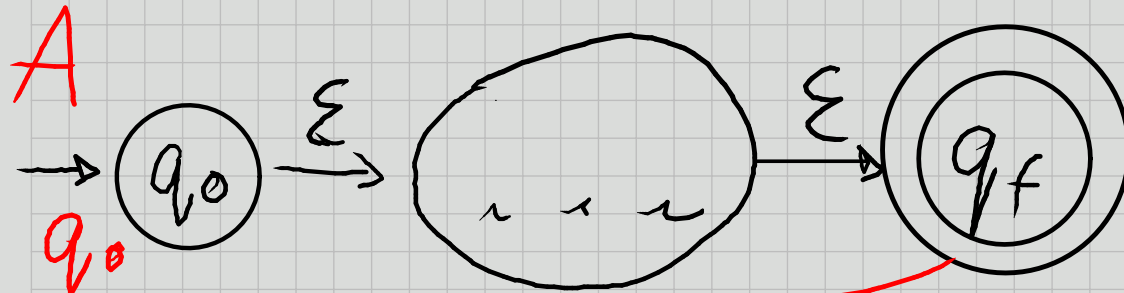
1. $|F| = 1$

2. input power of " q_0 " is zero

3. output power of " q_f " is zero

Concatenation

$$C = AB ; W_C = W_A W_B$$



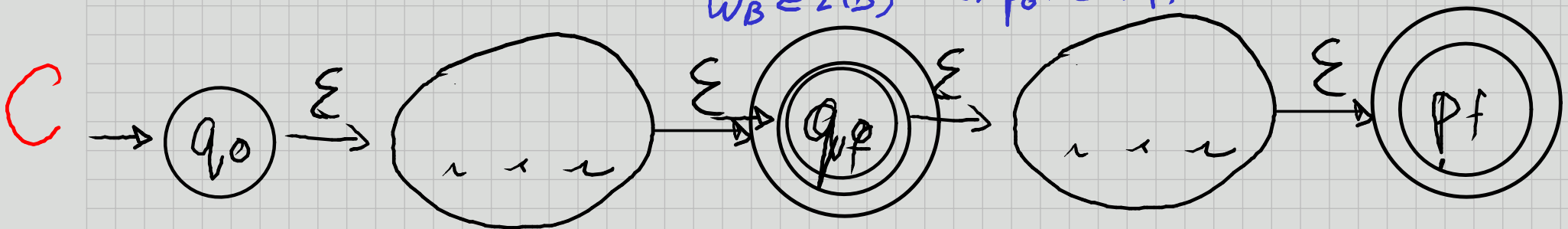
A, B, C - automata
 $L(A), L(B), L(C)$

$$L(C) \supseteq L(A) \circ L(B)$$

$$1) \forall w \in L(A) \circ L(B) \Rightarrow$$

$$\boxed{w \in C} : \exists z_0 z_1 z_2 \dots z_n : \\ z_i \xrightarrow{w_i} z_{i+1} \quad w_i \in L(A) \text{ or } L(B) \\ z_0 = q_0, z_n = p_f$$

$$w \in L(A) \circ L(B) \Rightarrow w = w_A w_B, \quad w_A \in L(A) \Rightarrow \exists q_0 \dots q_f : q_0 \xrightarrow{w_A} q_f \\ w_B \in L(B) \Rightarrow p_0 \xrightarrow{w_B} p_f$$



$$2) L(C) \subset L(A) \circ L(B)$$

$$\forall w \in C \stackrel{?}{\Rightarrow} w \in L(A) \circ L(B)$$

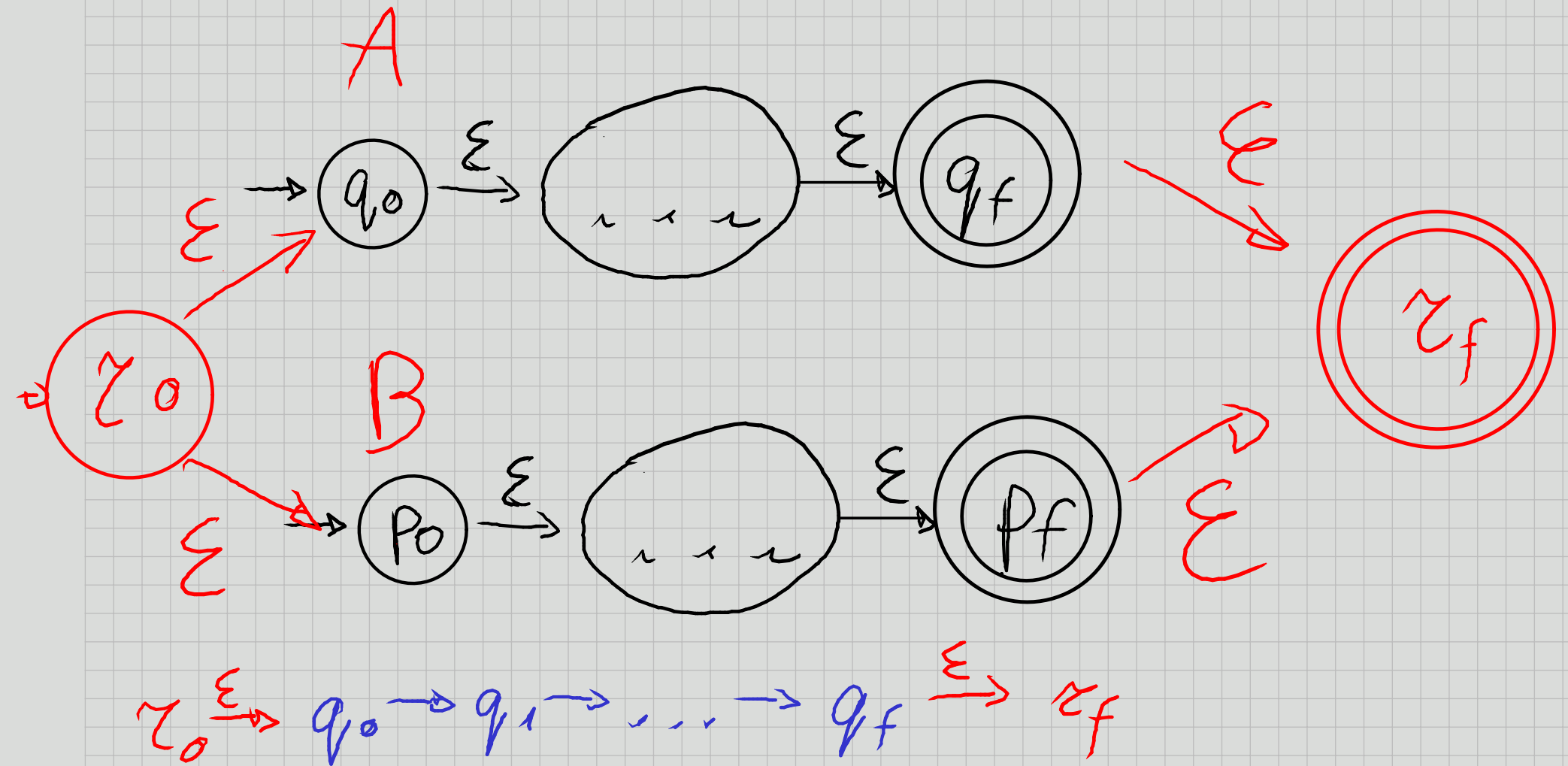
$$w \in C \Rightarrow q_0 \xrightarrow{w} p_f \Leftrightarrow q_0 \rightarrow z_1 \rightarrow z_2 \rightarrow \dots \rightarrow z_n \rightarrow p_f$$

We know that automata A and B do not intersect $\exists i : z_i = q_f, z_{i+1} = p_0$

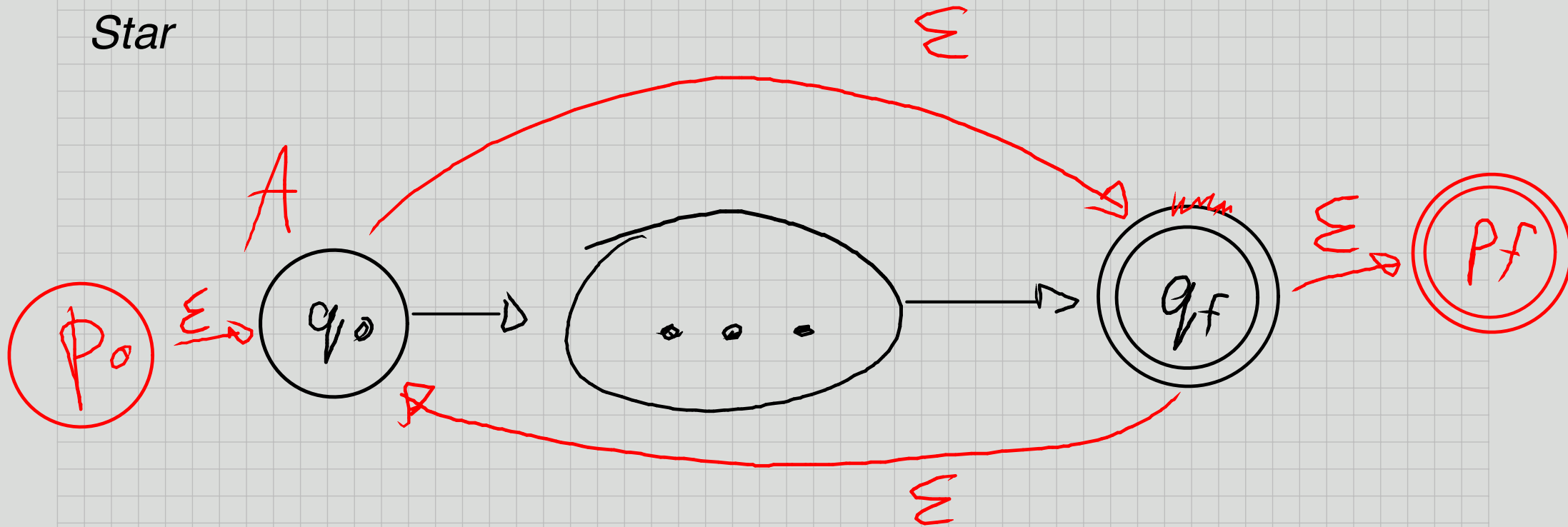
$$q_0 \xrightarrow{w} q_f \xrightarrow{\epsilon} p_0 \xrightarrow{w'} p_f$$

Union

$$C = A \cup B$$



Star



$$\Sigma \in A' \Rightarrow \exists q_0 \xrightarrow{\epsilon} \dots \xrightarrow{\epsilon} q_f$$

Empty word is recognized by automaton \Rightarrow there exists a path from starting state to accepting state where all transitions are empty word transitions

NFA \rightarrow DFA

$$\varepsilon\text{-closure}(q) = \{p \mid q \xrightarrow{\varepsilon} p\}$$

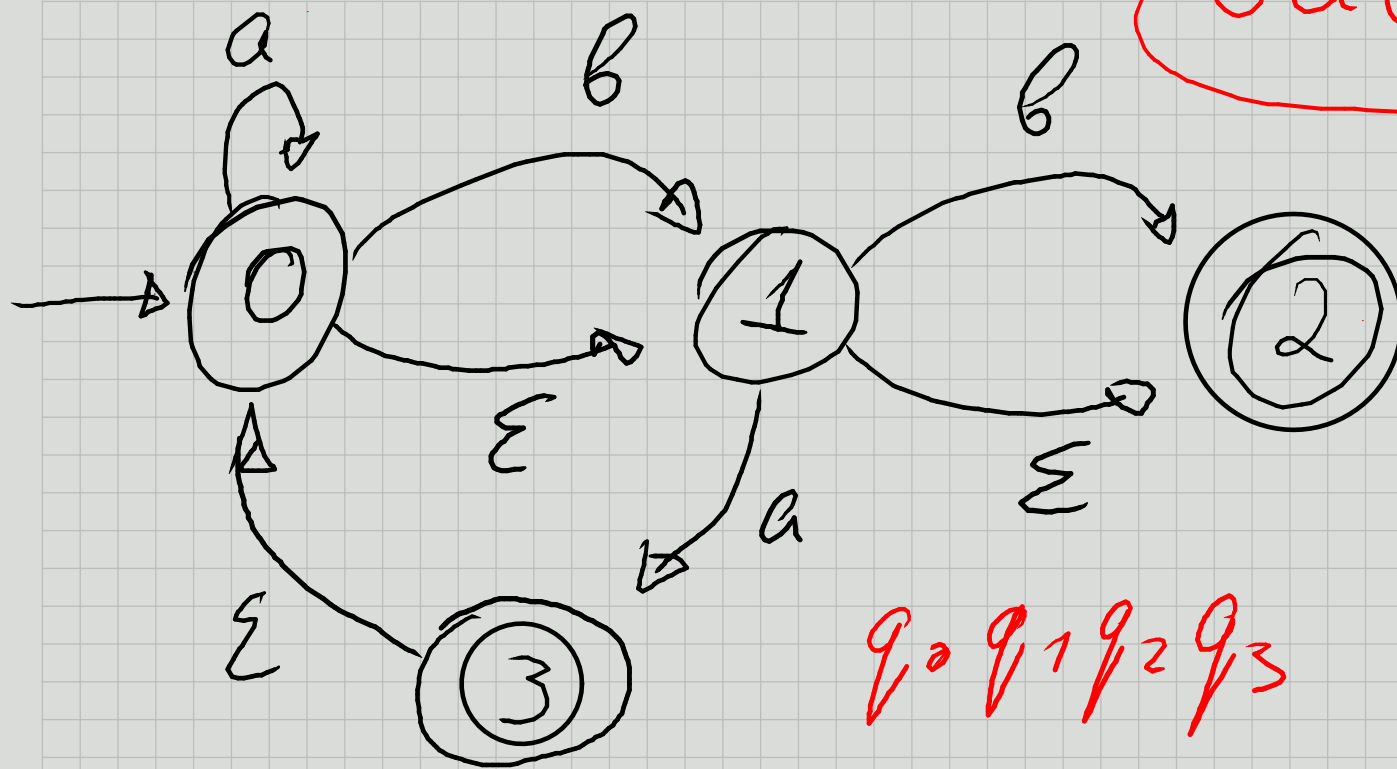
$\varepsilon\text{-cl}(q)$

$$\varepsilon\text{-cl}(Q) = \bigcup_{q \in Q} \varepsilon\text{-cl}(q)$$

$$Q_{\text{DFA}} = 2^{Q_{\text{NFA}}} \text{ - powerset}$$

$$Q_0 = \varepsilon\text{-cl}(q_0) \quad F_{\text{DFA}} = \{Q_i \mid Q_i \cap F_{\text{NFA}} \neq \emptyset\}$$

$$Q_i \xrightarrow{\alpha} Q_j = \varepsilon\text{-cl}(\{p \mid q \xrightarrow{\alpha} p, q \in Q_i\})$$



baab

$$\varepsilon\text{-cl}(0) = \{0, 1, 2\}$$

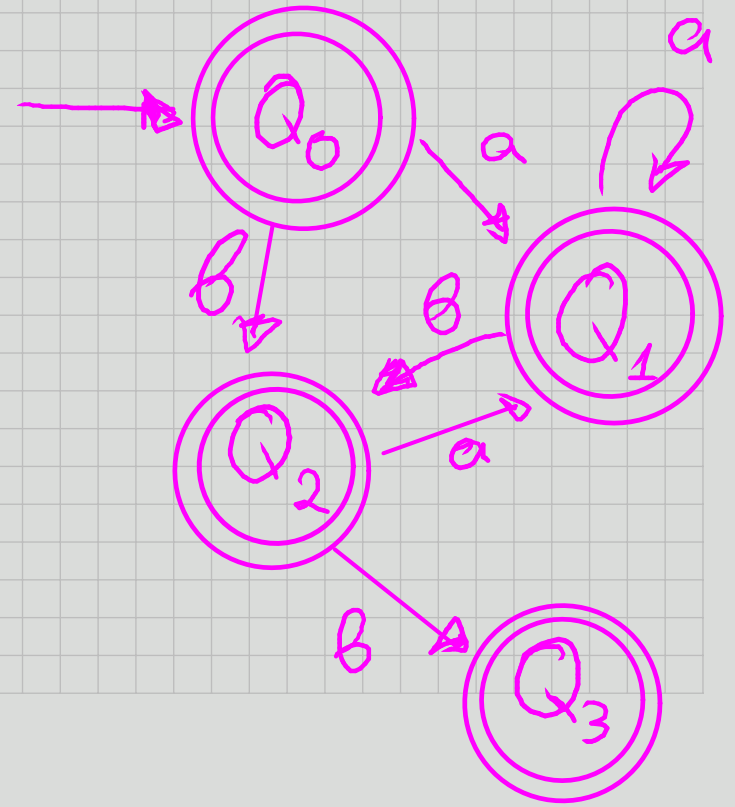
$$\varepsilon\text{-cl}(1) = \{1, 2\}$$

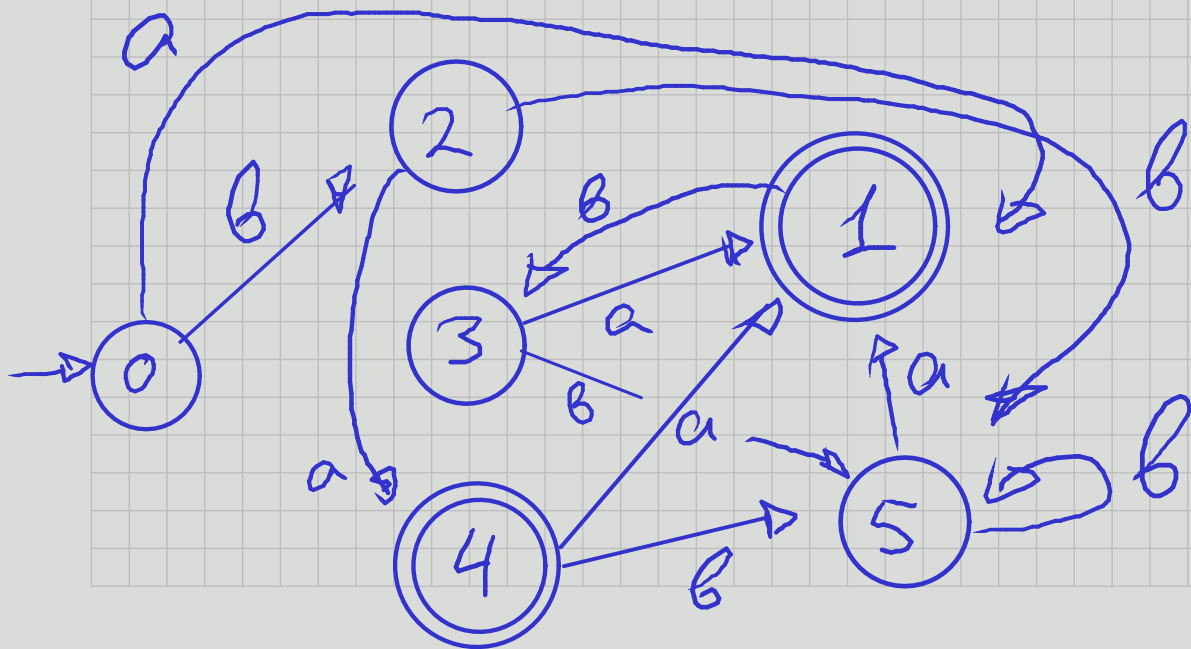
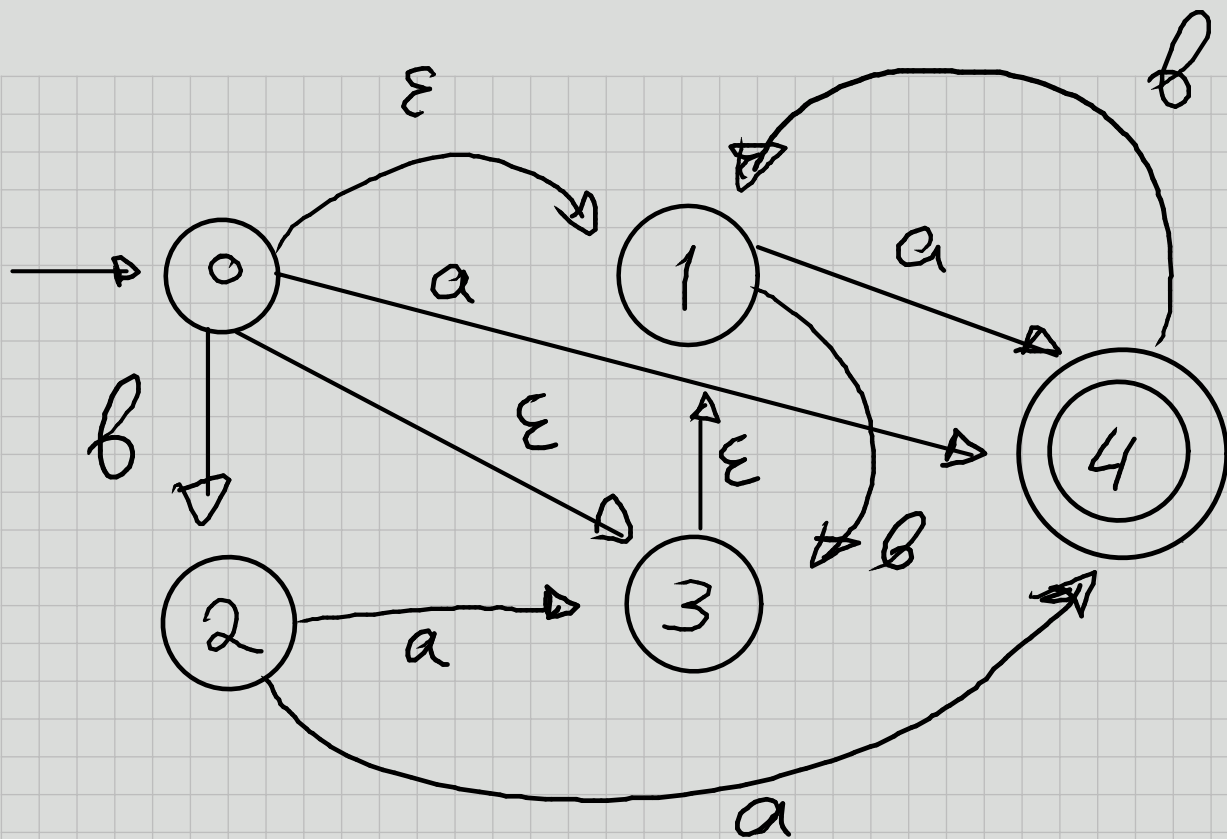
$$\varepsilon\text{-cl}(2) = \{2\}$$

$$\varepsilon\text{-cl}(3) = \{3, 0, 1, 2\}$$

$q_0 q_1 q_2 q_3$

DFA state	NFA state	a	b
Q_0^*	$q_0^* q_1^* q_2^*$	Q_1	Q_2
Q_1^*	$q_0^* q_3^* q_1^* q_2^*$	Q_1	Q_2
Q_2^*	$q_1^* q_2^*$	Q_1	Q_3
Q_3^*	q_2^*	—	—





DFA
state

NFA
state

a b

Q_0

$0^*1^*3^*$

Q_1

Q_2

Q_1^*

4^*

—

Q_3

Q_2

$2^*3^*1^*$

Q_4

Q_5

Q_3

1^*

Q_1

Q_5

Q_4^*

$3^*4^*1^*$

Q_1

Q_5

Q_5

3^*1^*

Q_1

Q_5

0

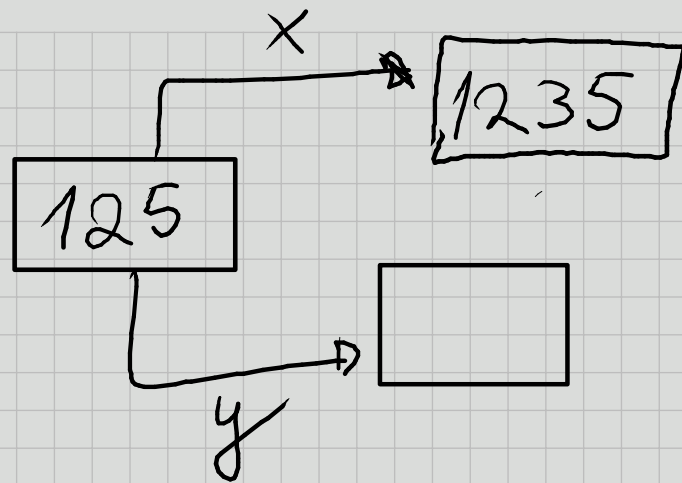
2

4

5

1

3



	x	y	z
125	1235	67	—
1235	1235	67	47
47	7	—	4
67	7	—	6
7	7	—	—
4	—	—	4

$$\triangleright x^*(xz^+ \mid yz^*)(xy)^+ \triangleleft$$

0 1 2 3 4 5 6 7 8

followpos

0, 1_x 2_x 4_y
 1_x 1_x 2_x 4_y
 2_x 3_z
 3_z 3_z 6_x
 4_y 5_z 6_x
 5_z 5_z 6_x
 6_x 7_y
 7_y 6_x 8_y

1_x 2_x 4_y

1_x 2_x 3_z 4_y

5_z 6_x

3_z 6_x

7_y

6_x 8_y

	x	y	z
1 _x 2 _x 4 _y	1 _x 2 _x 3 _z 4 _y	5 _z 6 _x	—
1 _x 2 _x 3 _z 4 _y	1 _x 2 _x 3 _z 4 _y	5 _z 6 _x	3 6
5 _z 6 _x	7	—	5 6
3 _z 6 _x	7	—	3 6
7 _y	—	6 8	—
6 _x 8 _y	7	—	—

1. Построить регулярное выражение (РВ) для языка из слов, содержащих в качестве под слова слово aab

examples of words:

baaba

aaabbb , aabbb
baabbba

$\Sigma^* aab \Sigma^*$

2. Построить РВ для языка, слова которого не содержат под-
слово ab,

examples of words:

aaaaa
↓
 a^*

$b^* a^*$

~~baba~~ ~~bbab~~
~~bbba~~
bbba aa a a^*
↓
 b^*

baaa

3. Построить регулярное выражение для языка из слов, содержащих в качестве под слова ровно одно слово ab

$babbb$

$aaabaaaa$

We need to have "ab" inside our word. What can we add before that and after that?

$\underbrace{\text{"not ab"}}_{b^*a^*} \underbrace{ab}_{ab} \underbrace{\text{"not ab"}}_{b^*a^*}$

$b^*a^*ab^*a^*$

$a^*a = aa^* = a^+$

$b^*a^+b^+a^*$

4. Построить РВ для языка всех слов чётной длины

$\Sigma = a, b$

$\varepsilon, aa, bb, ab, ba$
 $l=0$ $l=2$

$l=4$: aaaa, aabb

$\forall w \in L \Rightarrow w \circ w_2 \in L$, w_2 - word with length = 2

$\forall x, y \in L \Rightarrow x \circ y \in L$

inductive generation of language

Set $\{aa, bb, ab, ba\}$ is a generating set for the language

$w \in L$; $w \begin{matrix} \nearrow waa \\ \rightarrow wab \\ \searrow wba \\ \quad wbb \end{matrix}$

$(ba|bb|aa|ab)^*$
 $[(a|b)(a|b)]^* = (\Sigma\Sigma)^* = (\Sigma^2)^*$

$4 = 3 + 1$
 $3 = 2 + 1$

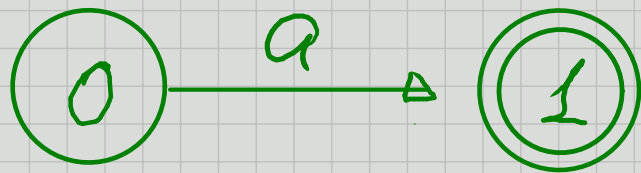
$N_{0,1}$ $2 = 1 + 1$

5. Постройте НКА А по РВ $(a(ab))^*b$.

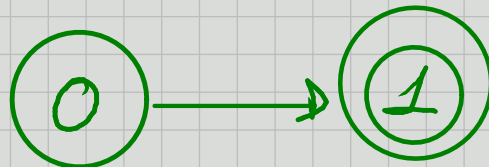
$RE \rightarrow NFA$

$[a(ab)]^*b$

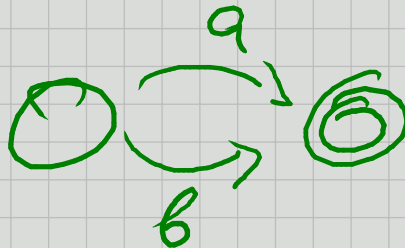
1. $A_1: \{a\}$



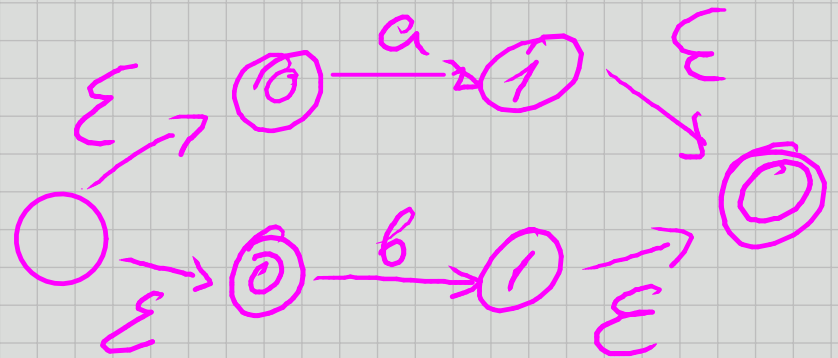
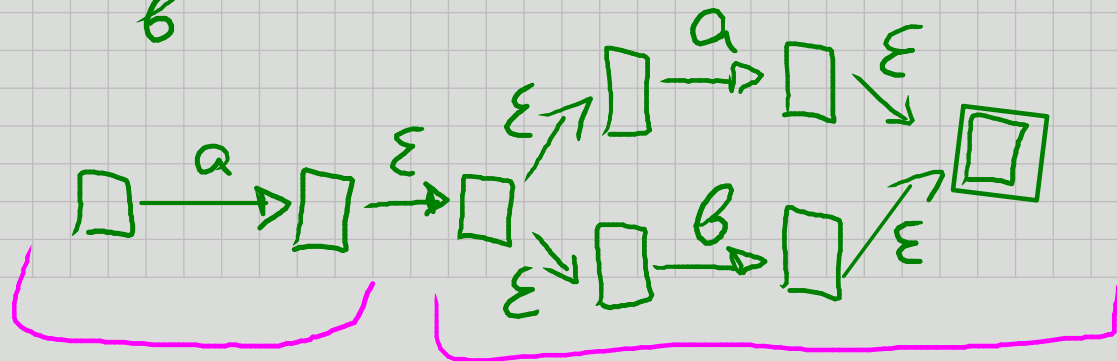
2. $A_2: \{b\}$



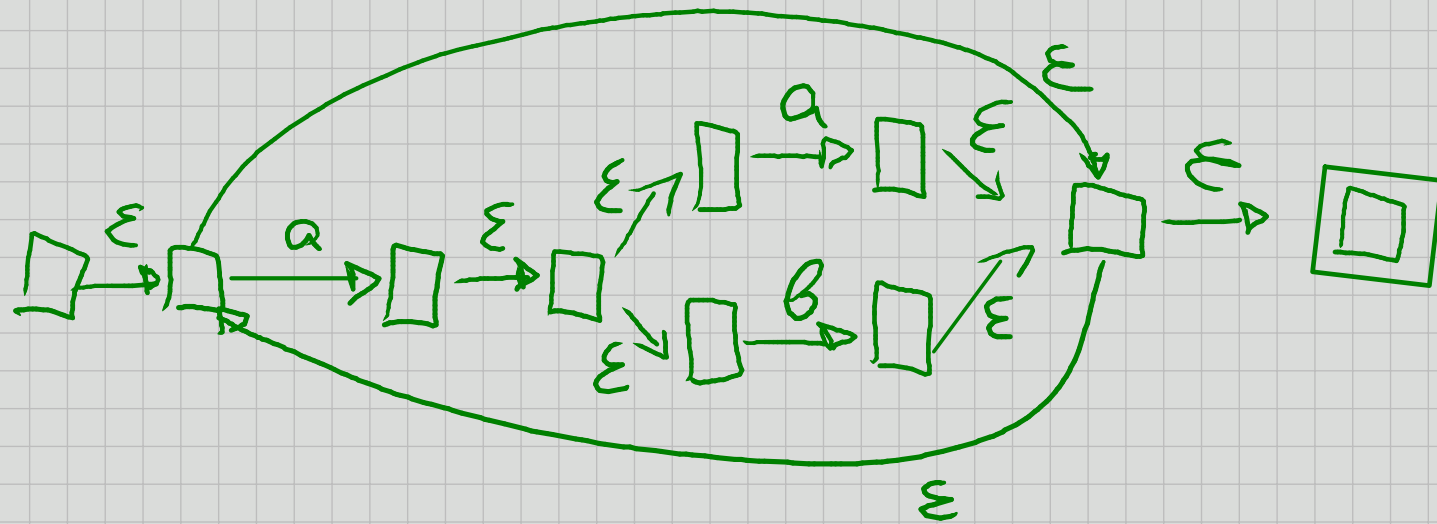
3. $\{a|b\}$



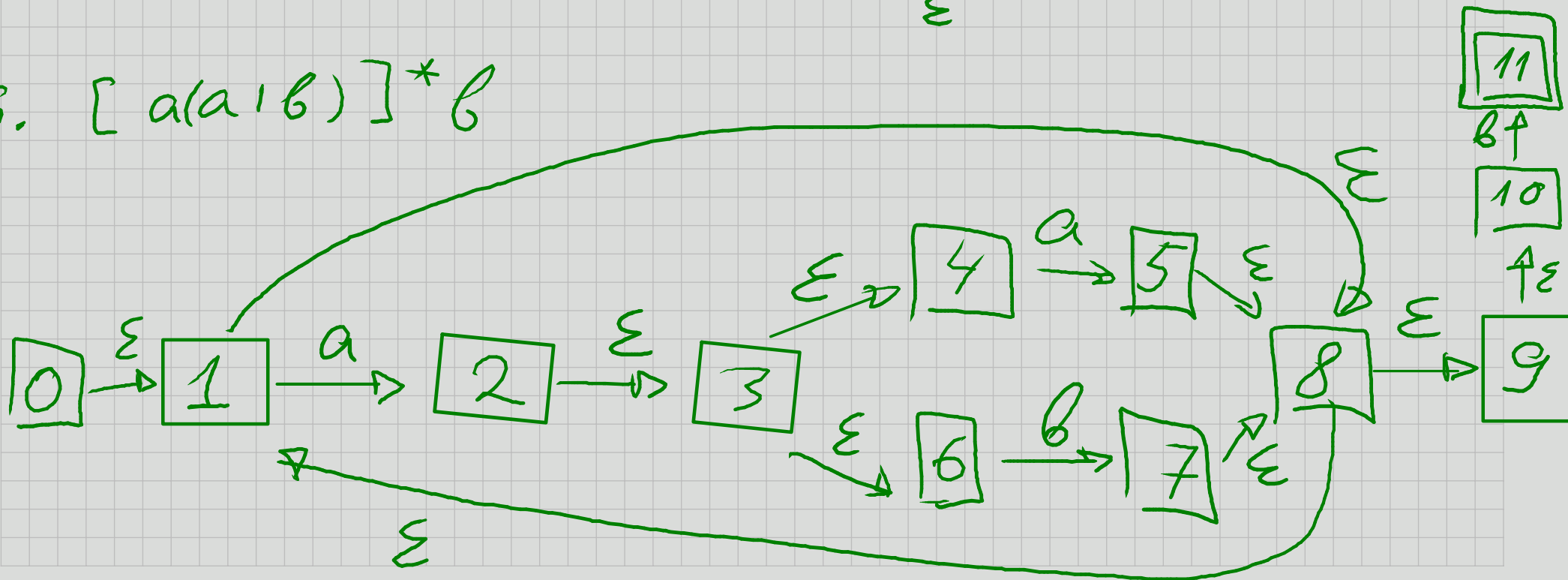
4. $a(ab)$

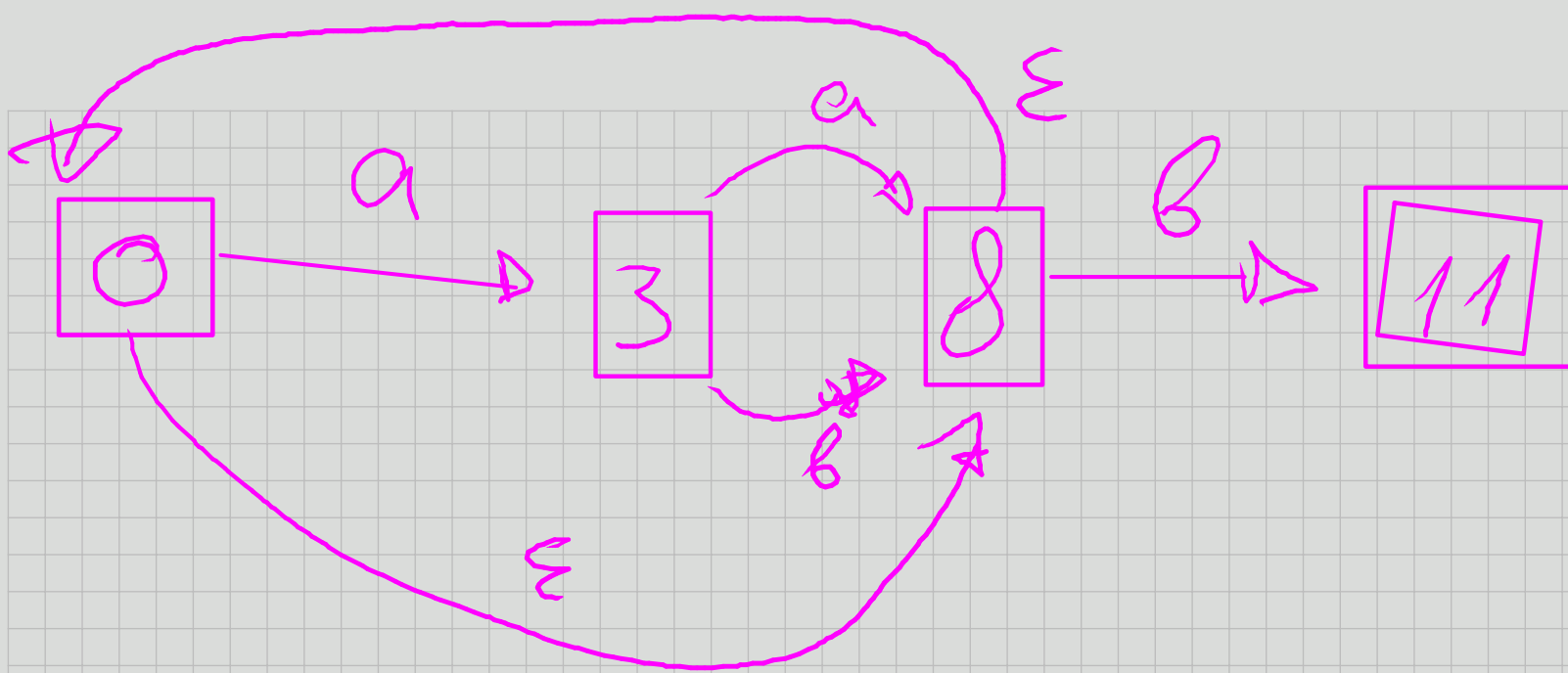


5. $[a(ab)^*]^*$

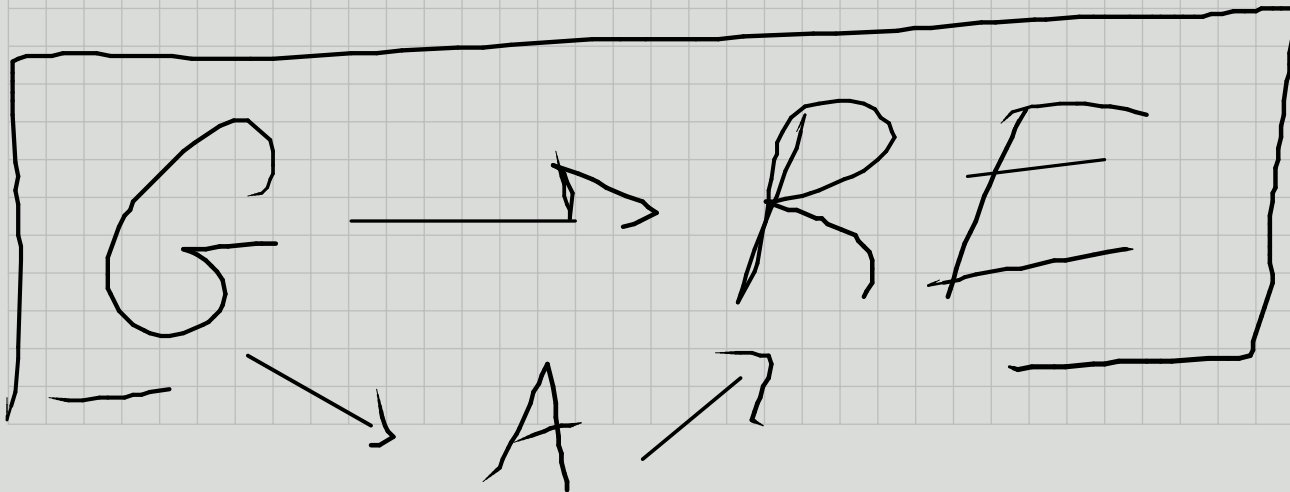
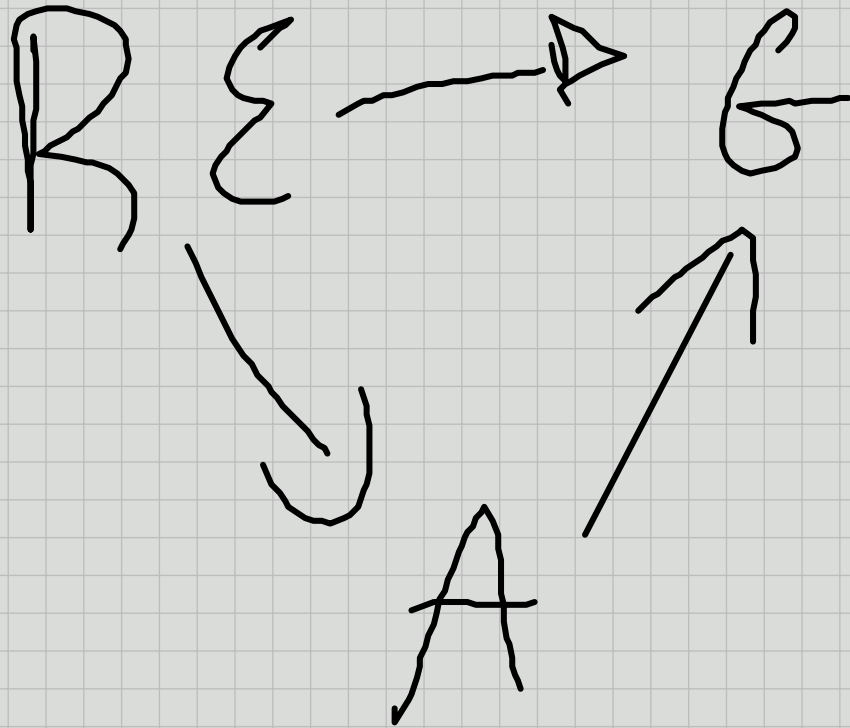


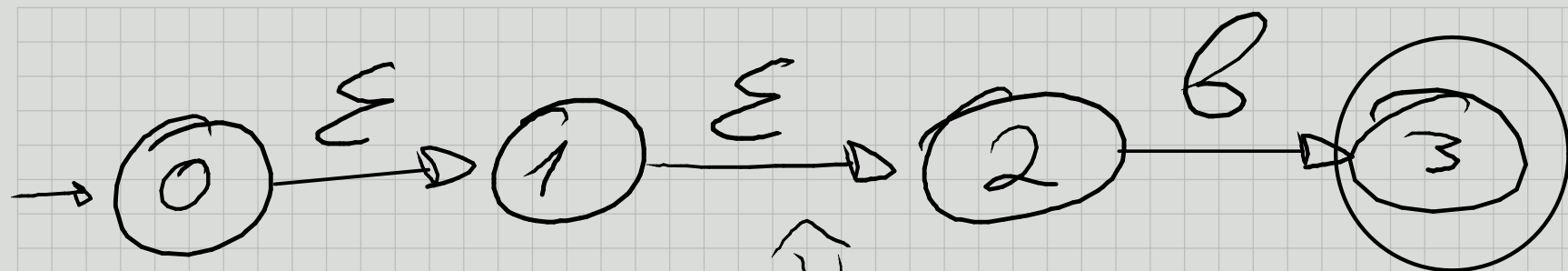
6. $[a(ab)^*]^*b$



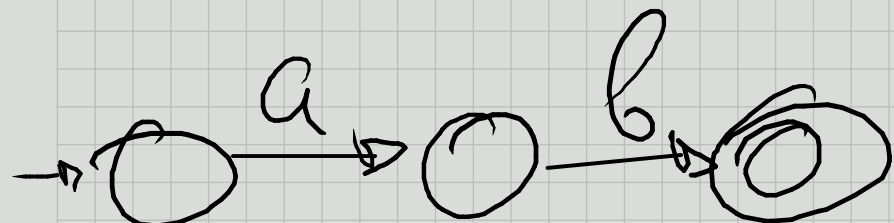
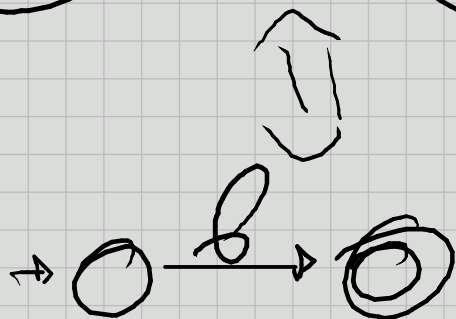


$NFA \rightarrow RE$

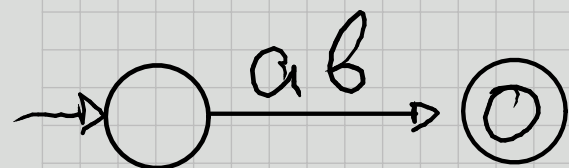




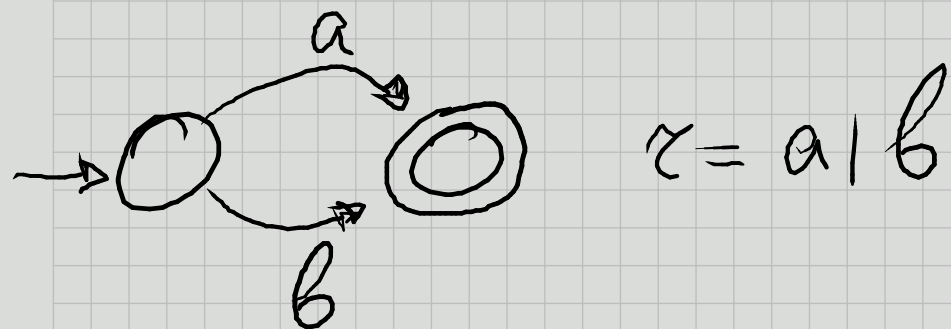
$w = b$



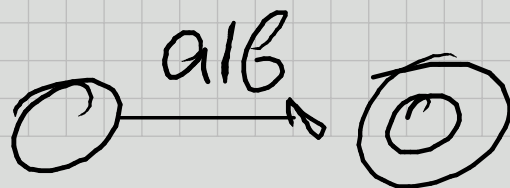
$w = ab$ $L = \{ab\}$ $r = ab$

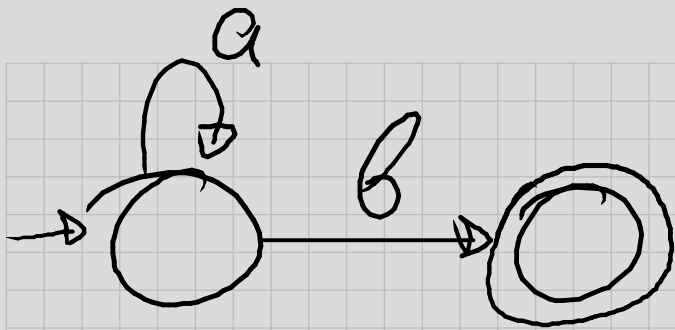


$d(p, w) \quad w = r$

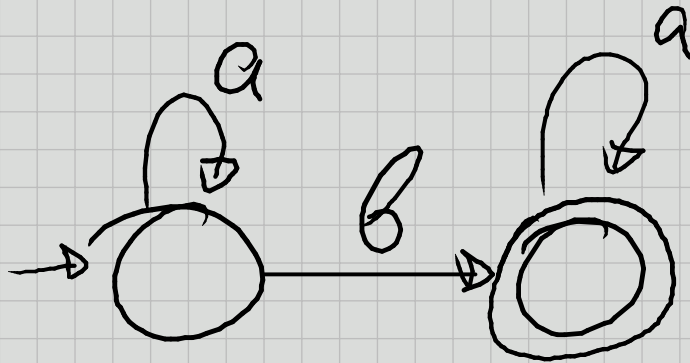
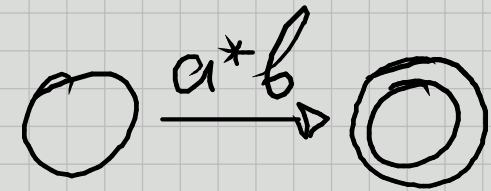


$r = a|b$

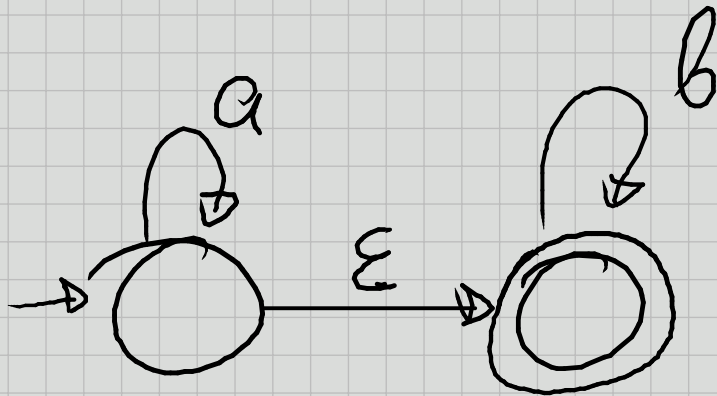
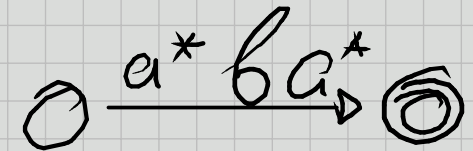




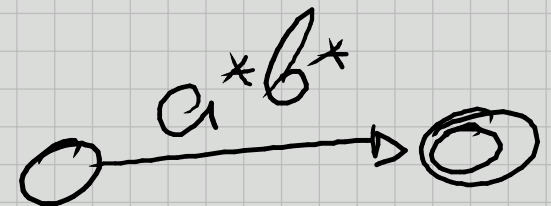
$$r = a^*b$$

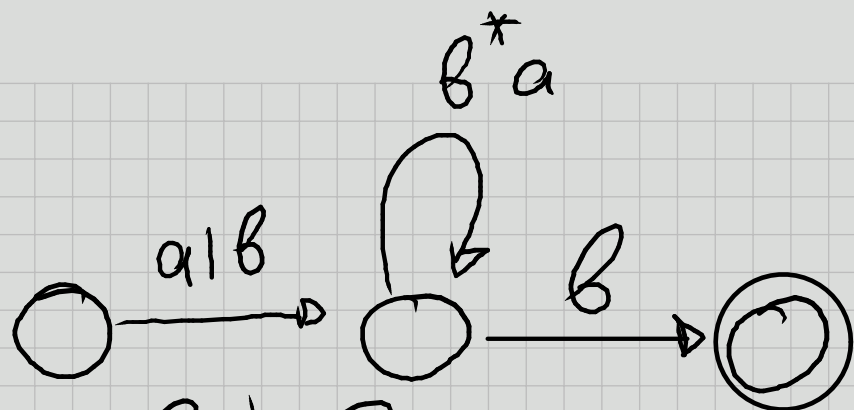


$$r = a^*ba^*$$



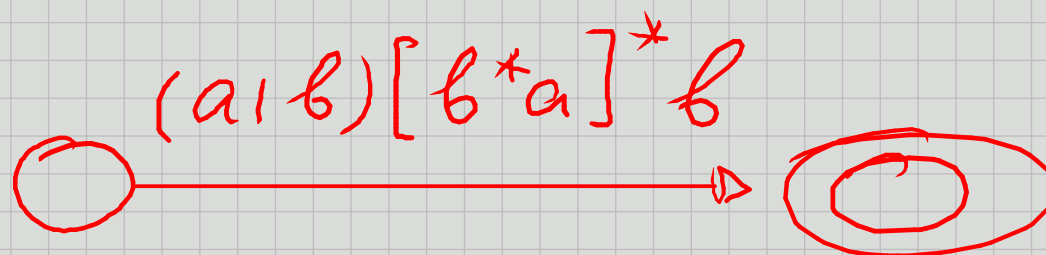
$$r = a^*b^*$$

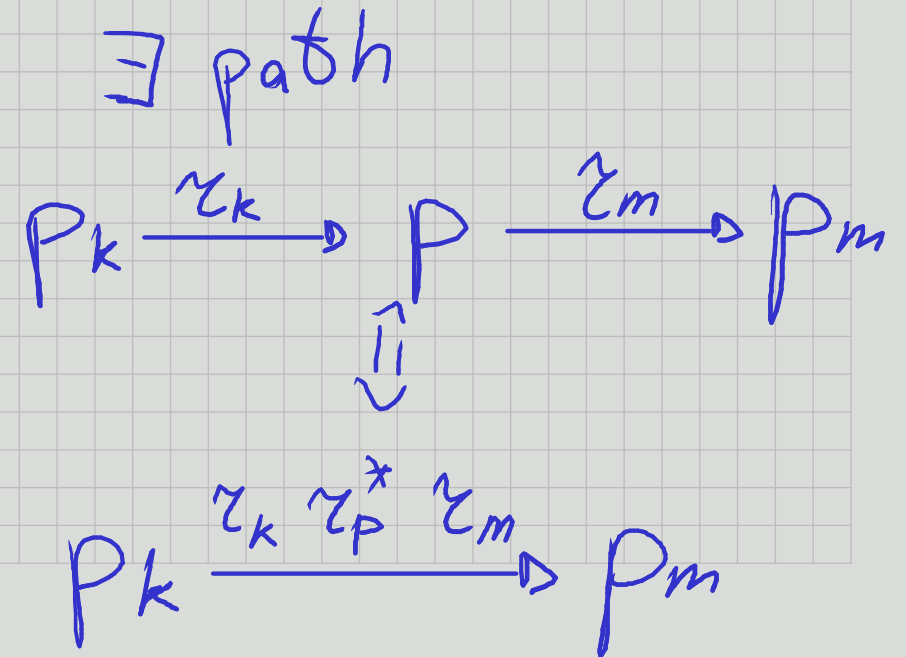
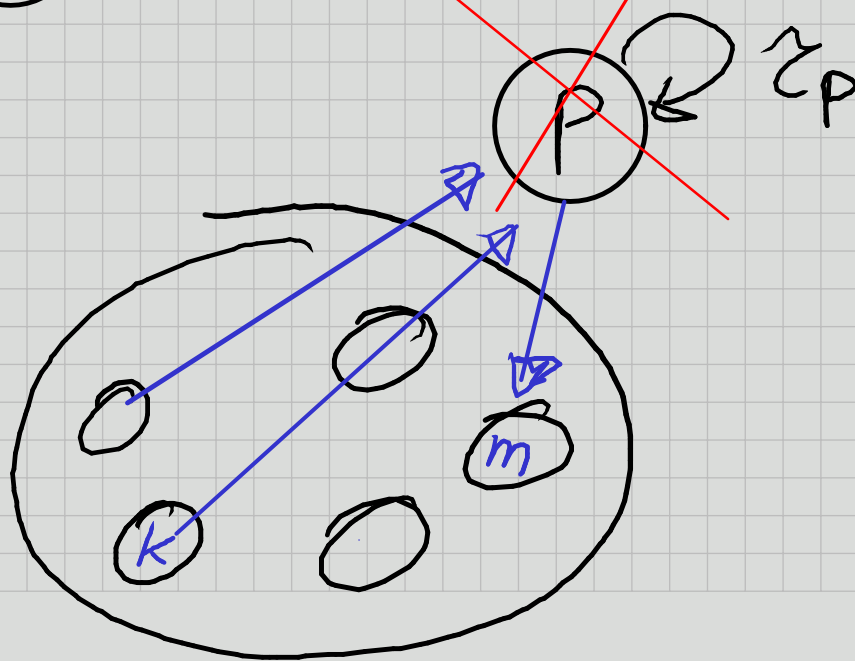
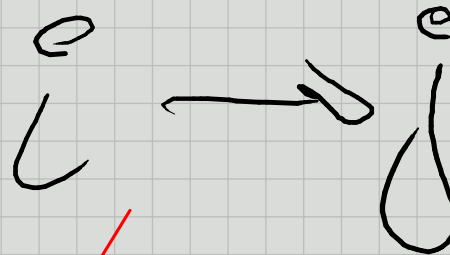
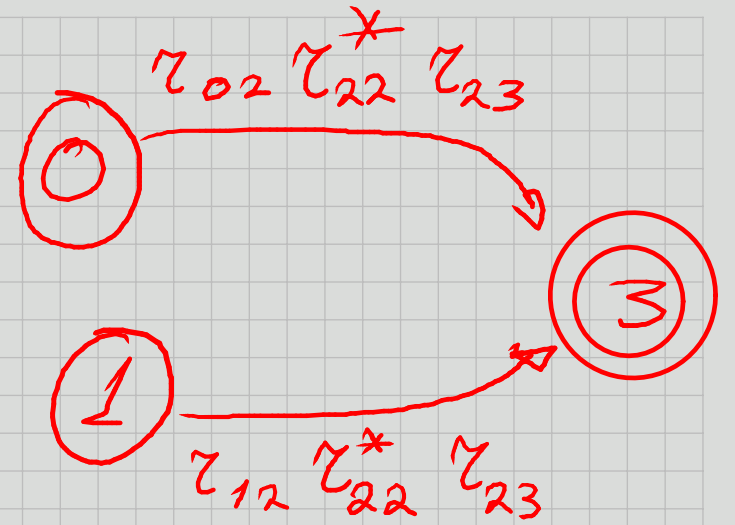
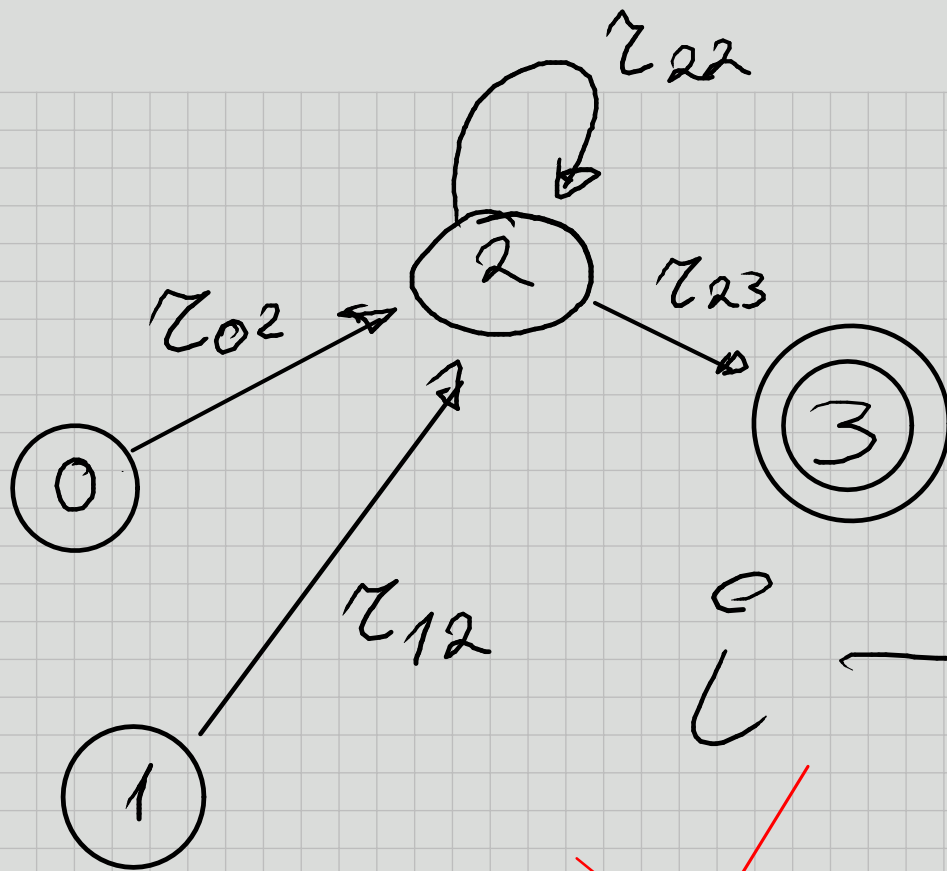


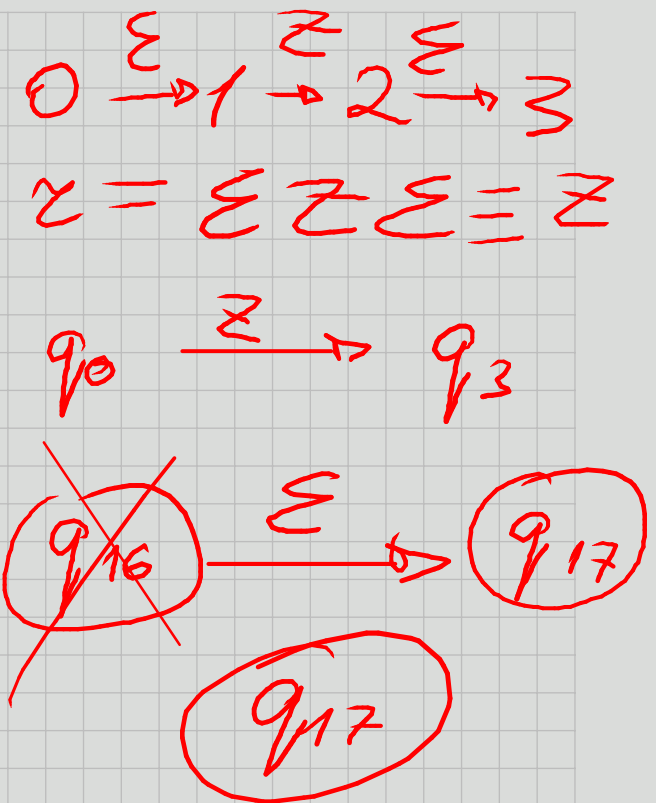
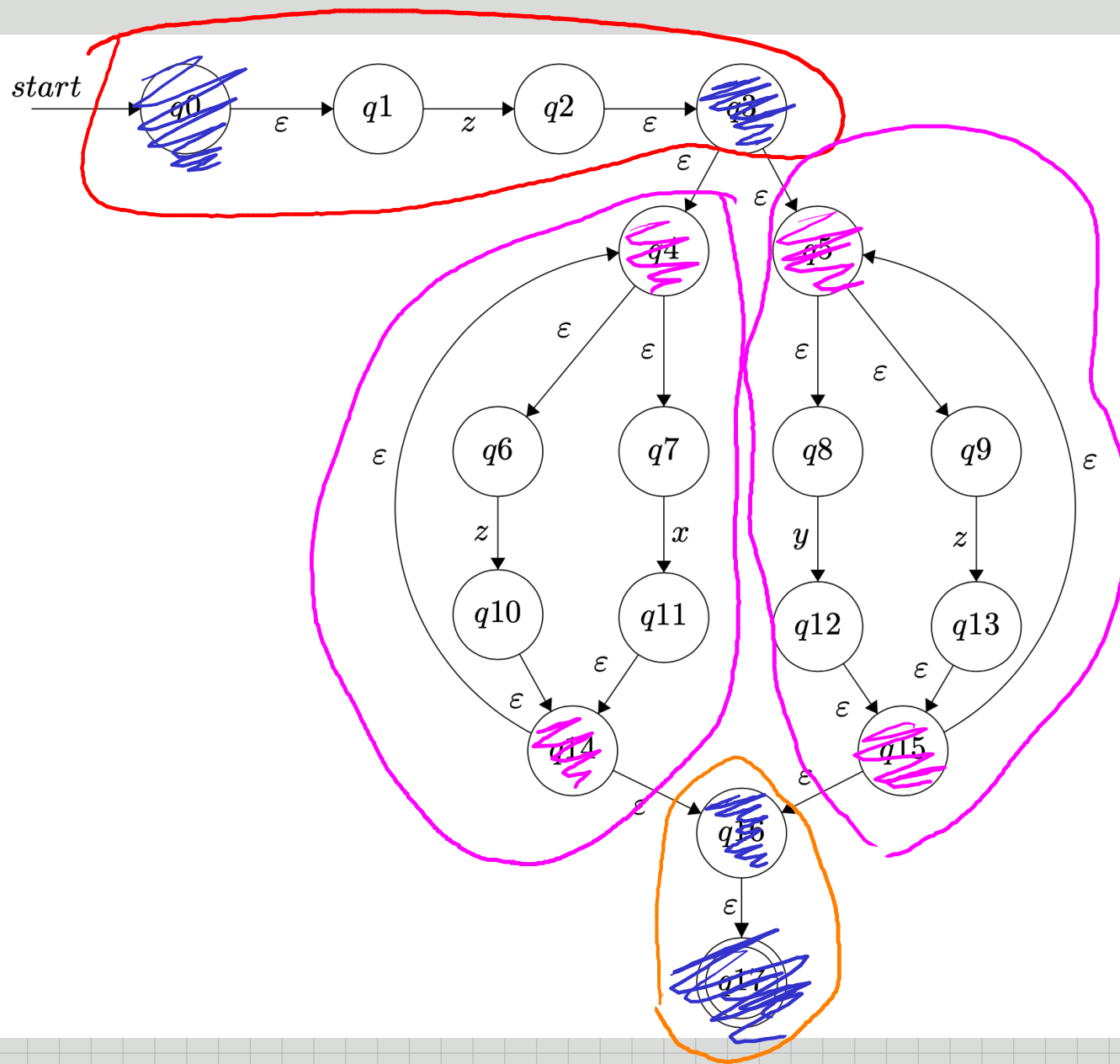


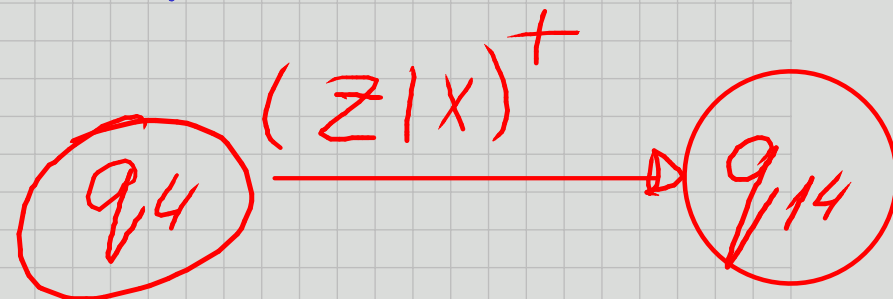
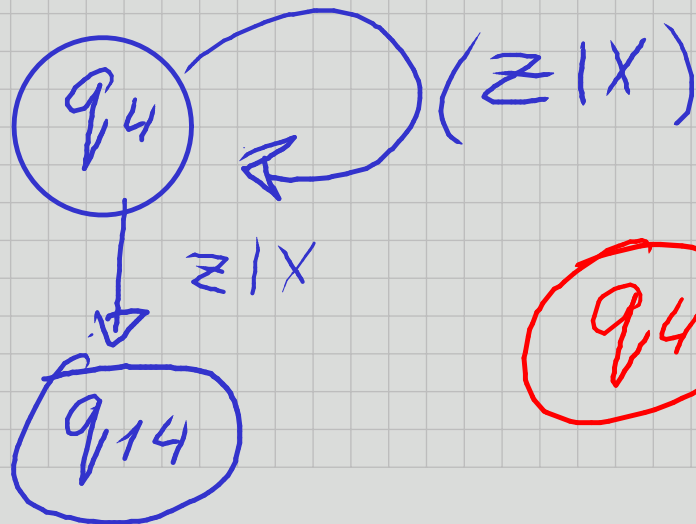
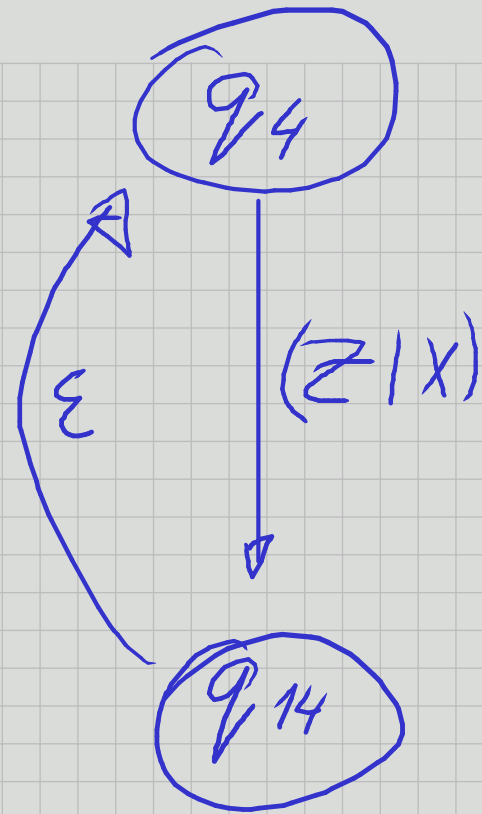
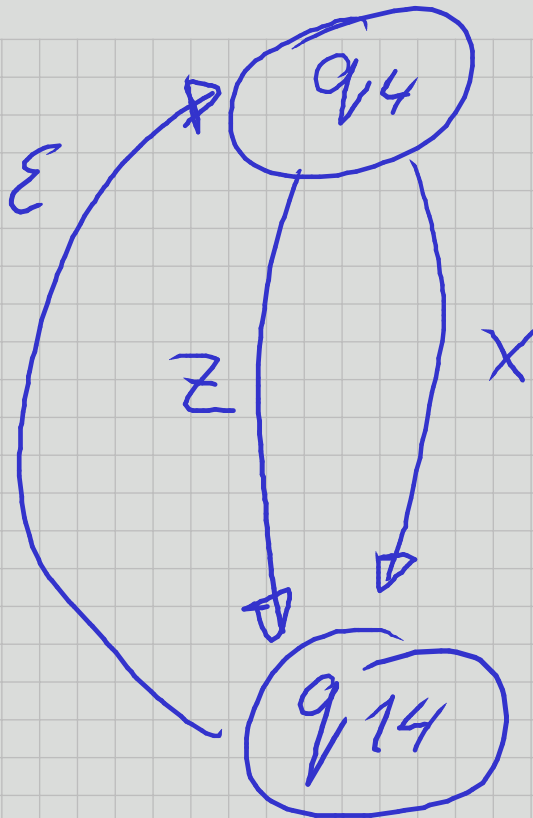
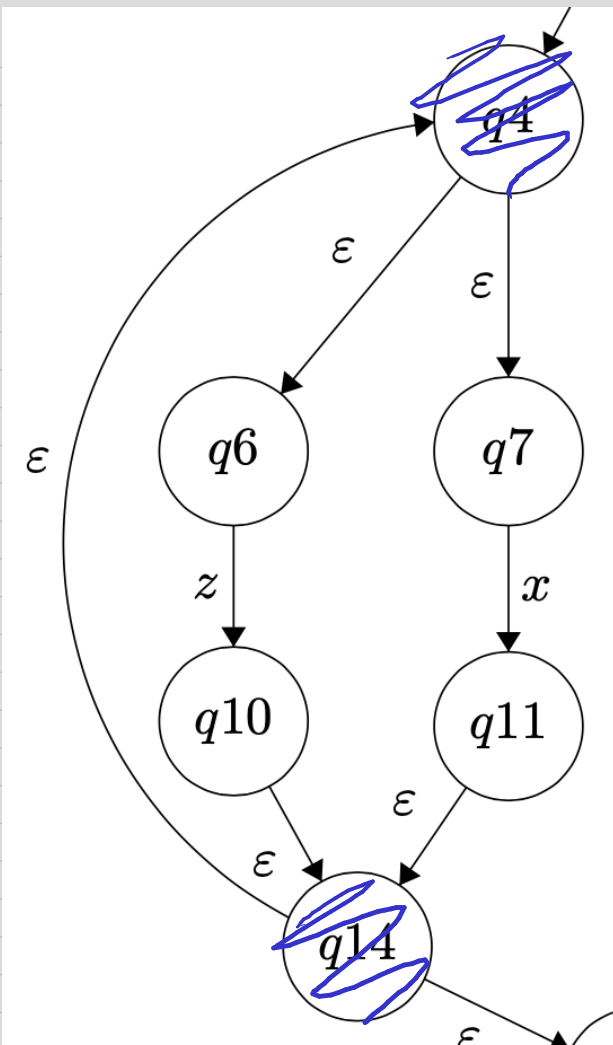
$$z = (a|b)[b^*a]^*b$$

$$\left[\begin{array}{l} (a|b)b \\ (a|b)(b^*a)b \\ (a|b)(b^*a)(b^*a)b \\ \vdots \end{array} \right]$$

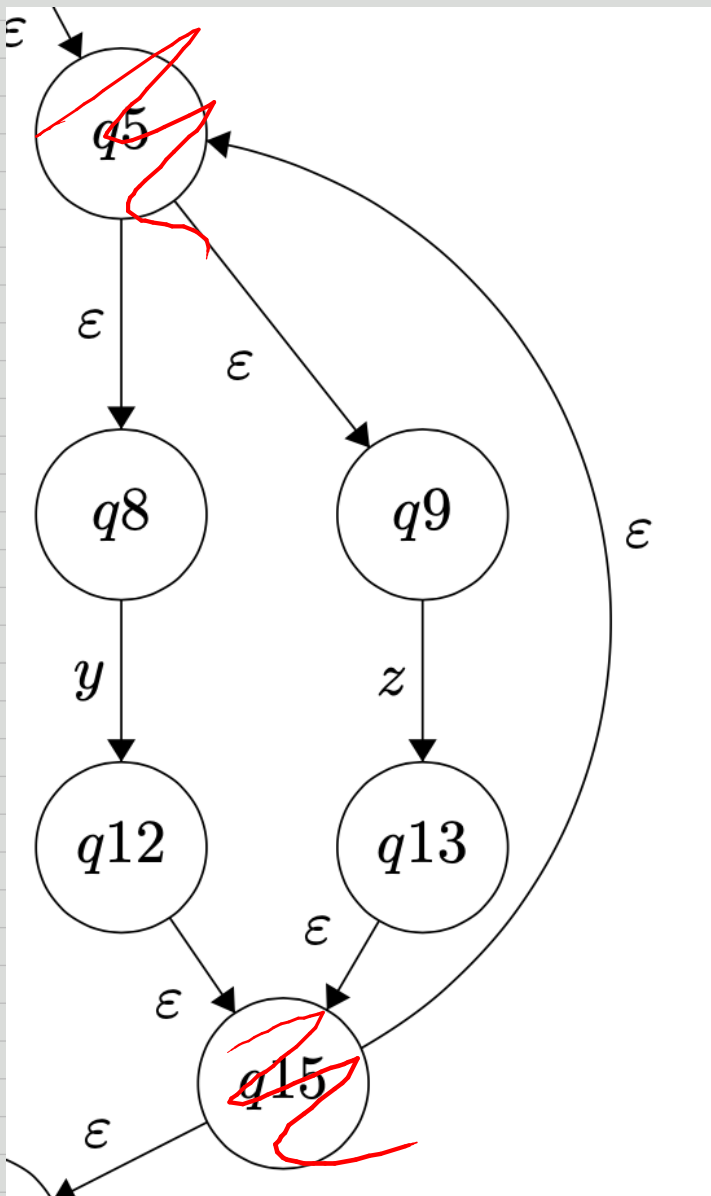






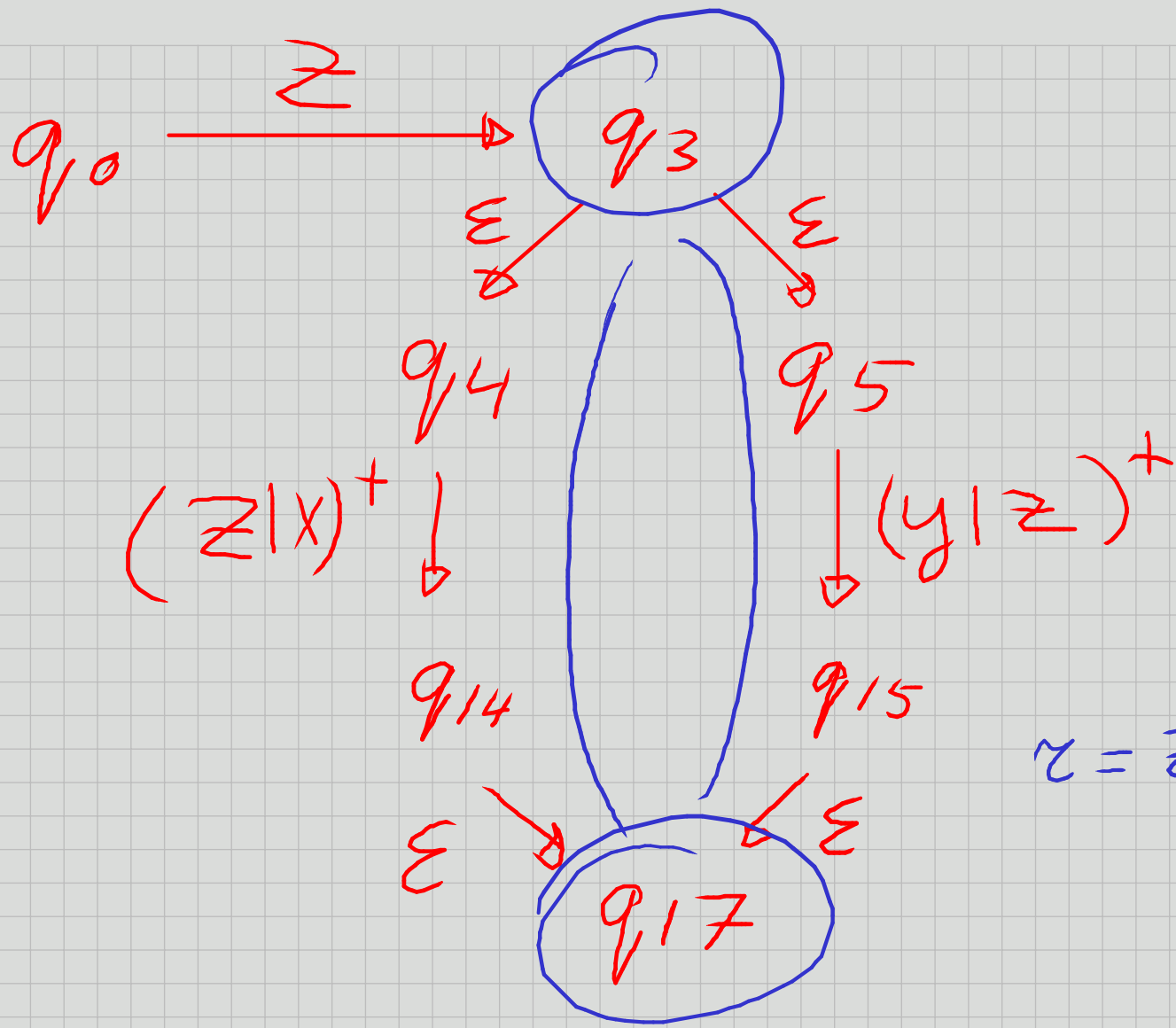


$$r = (z|x)^*(z|x)$$



$(y|z)^+$

$q_5 \xrightarrow{(y|z)^+} q_{15}$



$$r = z \left[(z|x)^+ \mid (y|z)^+ \right]$$

$$q_0 \xrightarrow{z} q_3 \xrightarrow{(z|x)^+ \mid (y|z)^+} q_{17}$$

