DM2 | GAVRILOV MISHA

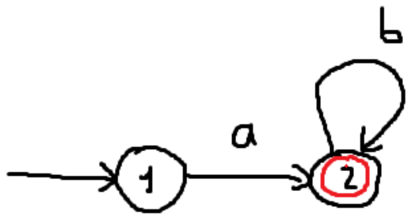
Опять же альтернативная ссылка на просмотр <u>решений</u>(из-за проблем с форматом при конвертации из notion в pdf).

Task 1

1. For each given regular expression P, construct a DFA (Deterministic Finite Automaton), and find the number of accepted word of length at most 5, i.e. the size of the set $\mathcal{L}' = \{w \in \mathcal{L}(P) \mid |w| \leq 5\}$. For "any" (`.`) and "negative" (`[^.]`) matches, assume that the alphabet is $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$.

a)

(a) $P_1 = \mathtt{ab*}$

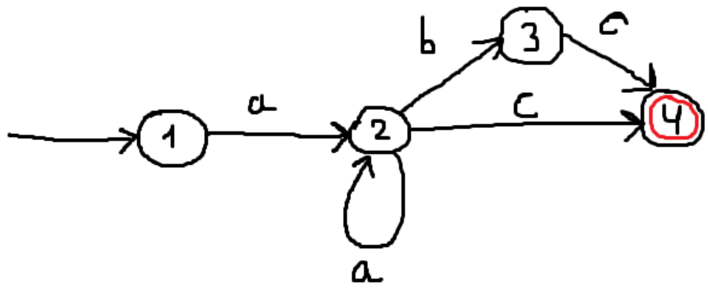


$ab^* = \{a, ab, abb, abbb, abbbb\}$

Size = 5

b)

(b) $P_2 = \mathtt{a+b?c}$

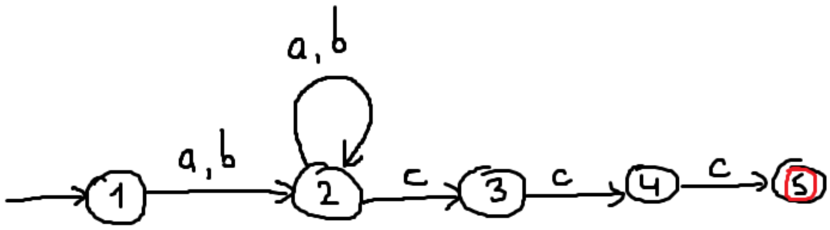


$a + b?c = \{ac, aac, aaac, aaaac, abc, aabc, aaabc\}$

`Size = 7`

c)

(c) $P_3 =$ `[^cd]+c{3}`

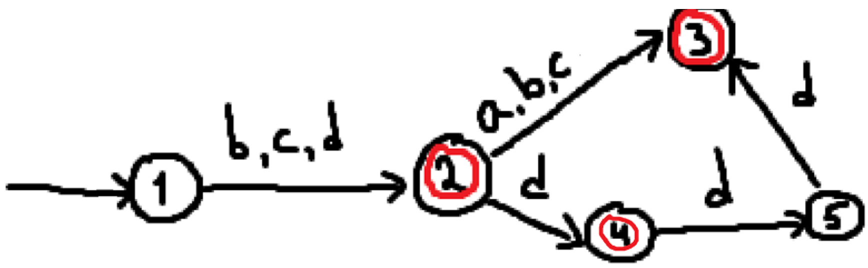


`[^cd]+c{3}` = $\{accc, aaccc, bccc, bbccc, abccc, baccc\}$

`Size = 6`

d)

(d) $P_4 =$ `[^a](.|ddd)?`

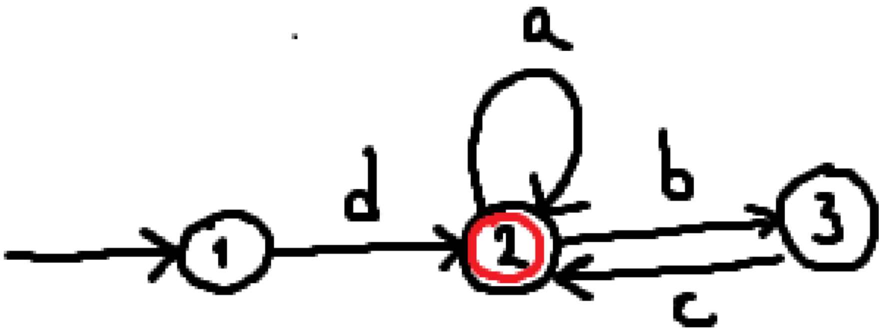


$P4 = \{b, ba, bb, bc, bd, bddd, c, ca, cb, cc, cd, cddd, d, da, db, dc, dd, ddddd\}$

Size = 18

e)

(e) P_5 = d(a|bc)*

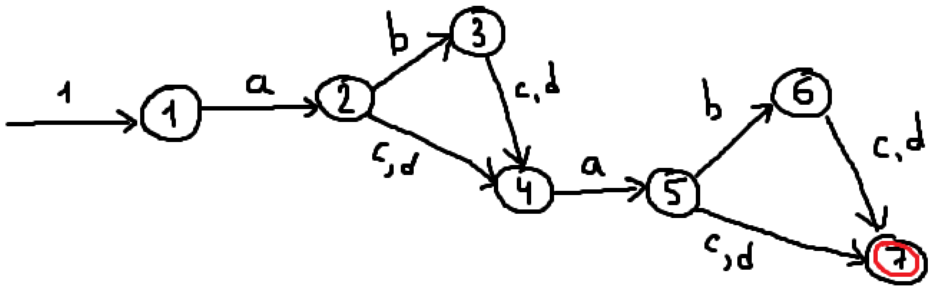


$P5 = \{d, da, daaa, daaaa, dbc, dbcbc, dabc, dbca, dabca, dbcaa, daabc\}$

Size = 11

f)

(f) P_6 = ((a|ab)[cd]){2}



$P6 =$
$\{acac, adad, acad, adac, abcac, abcad, abdac, abdad, acabc, acabd, adabc, adabd\}$

Size = 12

Task 2

. Describe the set of strings defined by each of these sets of productions in EBNF (extended Backus-Naur form).

a)

$$
\begin{aligned}
\text{(a)}\quad \langle \textit{string} \rangle &::= \langle L \rangle + \ \langle D \rangle ? \ \langle L \rangle + \\
\langle L \rangle &::= \mathsf{a} \mid \mathsf{b} \mid \mathsf{c} \\
\langle D \rangle &::= \mathsf{0} \mid \mathsf{1}
\end{aligned}
$$

$(a|b|c)+(0|1)?(a|b|c)+$

Строка состоящая из одной или нескольких букв a или b или c, за которыми может следовать или не следовать двоичная цифра, за которой следует одна или несколько букв a или b или c.

b)

$$
\begin{aligned}
\text{(b)}\quad \langle \textit{string} \rangle &::= \langle \textit{sign} \rangle ? \ \langle N \rangle \\
\langle \textit{sign} \rangle &::= \text{‘}+\text{’} \mid \text{‘}-\text{’} \\
\langle N \rangle &::= \langle D \rangle \ (\langle D \rangle \mid \mathsf{0})^* \\
\langle D \rangle &::= \mathsf{1} \mid \ldots \mid \mathsf{9}
\end{aligned}
$$

$(+|-)?(1|...|9)(1|...|9|0)^*$

Строка, которая может начинаться с символа $+$ или $-$(а может и не начинаться с этого), далее следует какое-то десятичное положительное число(не начинается с нуля).

c)

$$
\begin{aligned}
\text{(c)}\quad \langle \textit{string} \rangle &::= \langle L \rangle^* \ (\langle D \rangle +)? \ \langle L \rangle^* \\
\langle L \rangle &::= \mathrm{x} \mid \mathrm{y} \\
\langle D \rangle &::= \mathsf{0} \mid \mathsf{1}
\end{aligned}
$$

$(x|y)^*((0|1)+)?(x|y)^*$

Строка состоит из любого количества букв из $\{x,y\}$, за которыми возможно следует одна или несколько цифр из $\{0,1\}$, за которыми следует любое количество букв из $\{x,y\}$.

d)

(d) $\langle string \rangle$ ::= $\langle C \rangle$ $\langle R \rangle^*$
$\langle C \rangle$ ::= a | . . . | z | A | . . . | Z
$\langle D \rangle$ ::= 0 | . . . | 9
$\langle R \rangle$ ::= $\langle C \rangle$ | $\langle D \rangle$ | '_'

$(a|...|z||A|...|Z)(a|...|z|A|...|Z|0|...|9|\_|)^*$

Строка, которая начинается с большой или маленькой буквы латинского алфавита($a...z$ $A...Z$), после которой идет любое количество символов из $\{0...9\ a...z\ A...Z\ \_\}$.

Task 3

3. Let $\mathcal{G} = \langle V, T, S, P \rangle$ be the phrase-structure grammar with vocabulary $V = \{\mathtt{A}, \mathtt{S}\}$, terminal symbols $T = \{0, 1\}$, start symbol $S = \mathtt{S}$, and set of productions P: $\mathtt{S} \to 1\mathtt{S}$, $\mathtt{S} \to 00\mathtt{A}$, $\mathtt{A} \to 0\mathtt{A}$, $\mathtt{A} \to 0$.

a)

(a) Show that 111000 belongs to the language generated by $\mathcal{G}$.

$S \to 1S \to 11S \to 111S \to 11100A \to 111000$

Значит 111000 принадлежит языку, порожденному G

b)

(b) Show that 11001 does not belong to the language generated by $\mathcal{G}$.

Рассмотрим множество продукций : $S \to 1S$, $S \to 00A$, $A \to 0A$, $A \to 0$

Заметим, что любое слово оканчивается на 0. Значит 11001 не принадлежит языку, порожденному G.

c)

(c) What is the language generated by $\mathcal{G}$?

$S \to 1S$ продуцирует строки $1^n S$

$S \to 00A$ совместно с $A \to 0A$ и $A \to 0$ продуцирует строки $000A$

Тогда $L(G) = \{1^n 0^m | n \geq 0, m \geq 3\}$

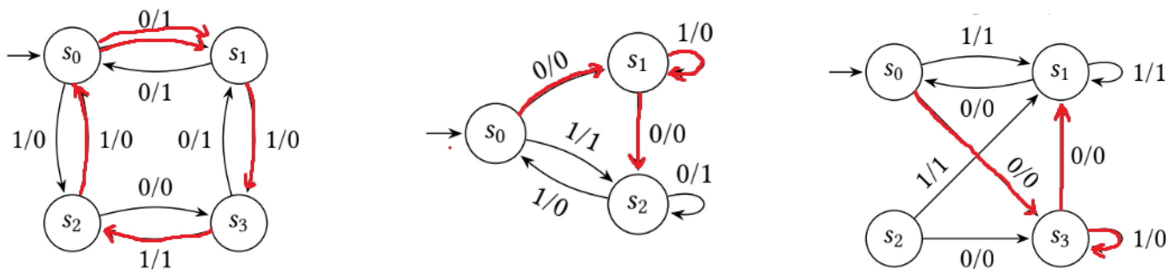
Task 4

4. Find the output generated from the input string 01110 for each of the following Mealy machines.

a) **10101**

b) **00000**

c) **00000**

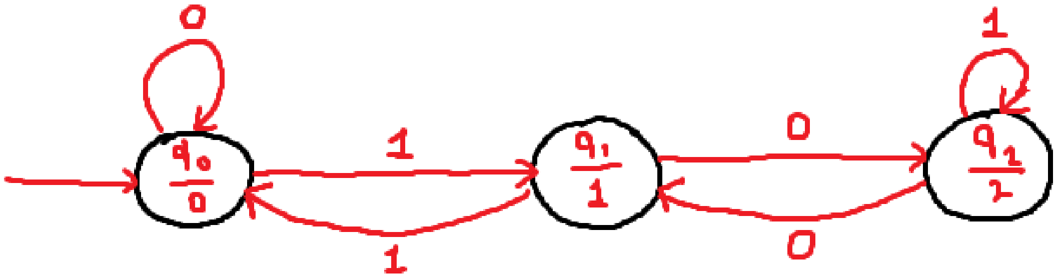


Task 5

5. Construct a Moore machine for each of the following descriptions.

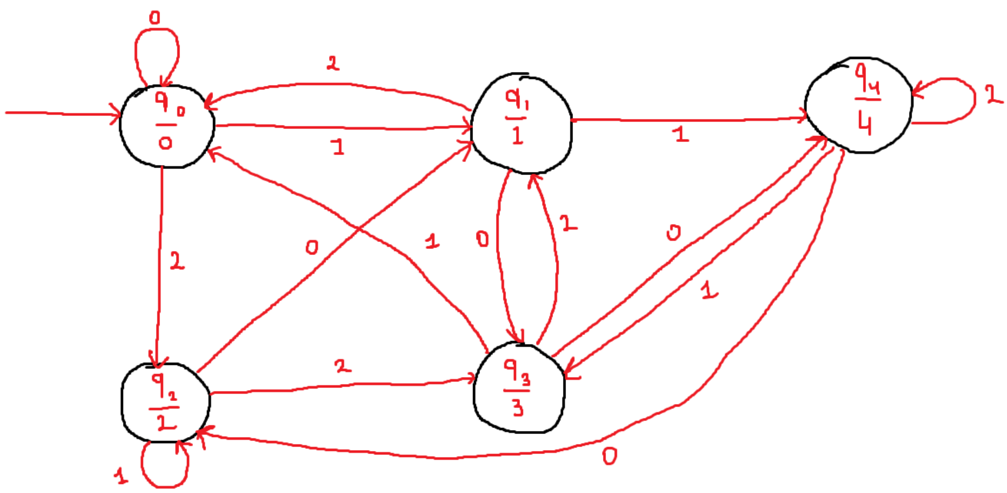
a)

(a) Determine the residue modulo 3 of the input treated as a binary number. For example, for input ε (which corresponds to "value" 0) the residue is 0; 101 (5 in decimal) has residue 2; and 1010 (value 10) has residue 1.



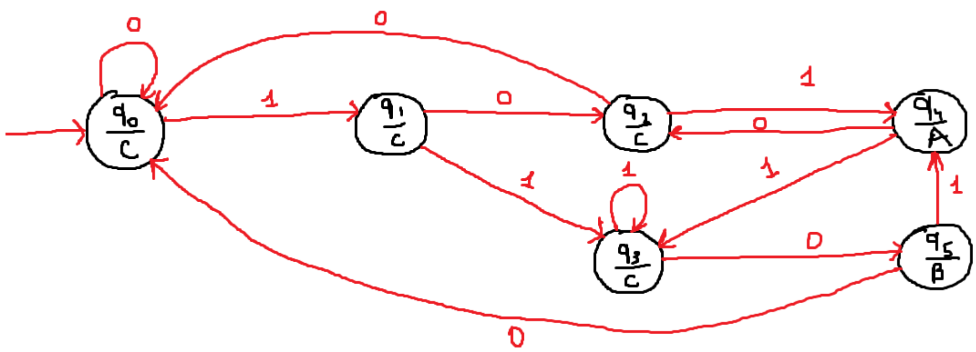
b)

(b) Output the residue modulo 5 of the input from $\{0, 1, 2\}^*$ treated as a ternary (base 3) number.



c)

(c) Output A if the binary input ends with 101; output B if it ends with 110; otherwise output C.



Task 6

Show that regular languages are *closed* under the following operations.

a)

(a) Union, that is, if L_1 and L_2 are regular languages, then $L_1 \cup L_2$ is also regular.

Если L_1 и L_2 - регулярные языки, то $L_1 \cup L_2$ - регулярный, по определению регулярного языка

b)

(b) Concatenation, that is, if L_1 and L_2 are regular languages, then $L_1 \cdot L_2$ is also regular.

Если L_1 и L_2 - регулярные языки, то $L_1 L_2$ - регулярный, по определению регулярного языка

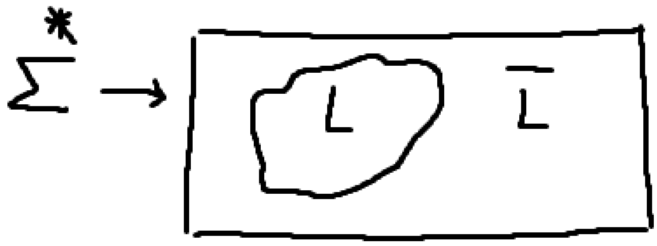
c)

(c) Kleene star, that is, if L is a regular language, then L^* is also regular.

Если L - регулярный язык, то L^* - регулярный, по определению регулярного языка

d)

(d) Complement, that is, if L is a regular language, then $\overline{L} = \Sigma^* - L$ is also regular.



Пусть $M = \{Q, \Sigma, \delta, q_0, F\}$ - автомат для L.

Если $\omega \in L$, то M допускает ω, то есть результирующее состояние $\in F$.

Если $\omega \notin L$, то M не допускает ω, то есть результирующее состояние $\notin F$.

Обозначим $\overline{M} = \{Q, \Sigma, \delta, q_0, Q \backslash F\}$ и $L(\overline{M})$ это $\overline{L}$.

Если $\omega \in L$, то M допускает ω, то есть результирующее состояние $\in F$, значит $\overline{M}$ не допускает ω.

Если $\omega \notin L$, то M не допускает ω, то есть результирующее состояние $\notin F$, значит $\overline{M}$ допускает ω.

Таким образом $\overline{L}$ - регулярный язык.

e)

(e) Intersection, that is, if L_1 and L_2 are regular languages, then $L_1 \cap L_2$ is also regular.

Если L_1 и L_2 - регулярные языки, а $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$, то $L_1 \cap L_2$ - регулярный(так как в пунктах $a - d$ мы доказали замкнутость остальных операций).

Task 7

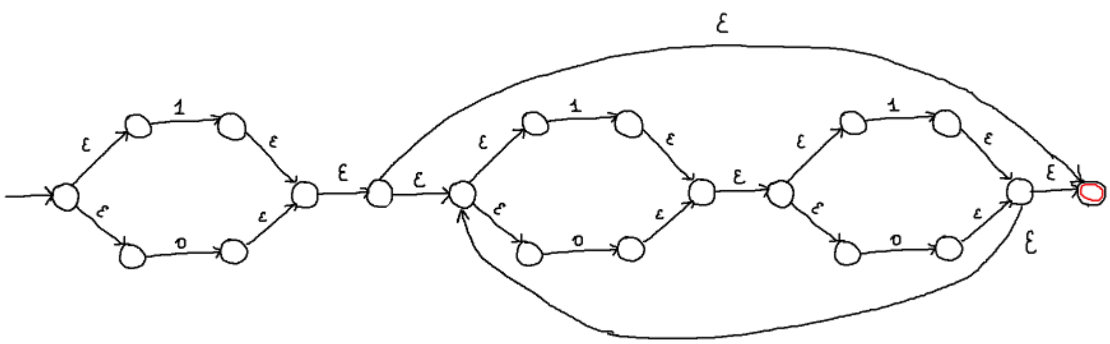
7. Determine whether the following languages are regular or not. For non-regular languages, use Pumping lemma to prove that they are not regular. For each regular language, provide a regular expression and construct an ε-NFA.

a)

(a) $L_1 = \{w \in \{0,1\}^* \mid \text{length of } w \text{ is odd}\}$

Этот язык регулярный : $(0|1)((0|1)(0|1))^*$.

Построим $\epsilon - NFA$ с помощью **Thompson's construction.**



b)

$$\text{(b) } L_2 = \{0^n 1^n \mid n \in \mathbb{N}\}$$

Допустим язык L_2 - регулярный.

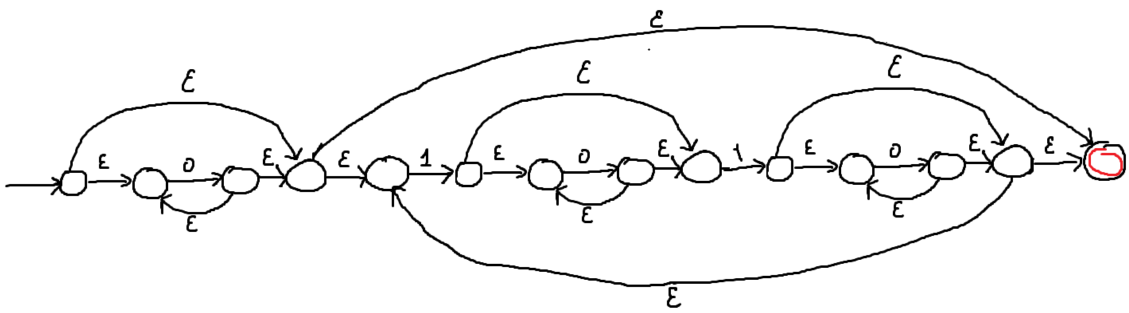
Тогда существует такое n, что для любого слова $\omega \in L_2$ длины не меньше n найдутся слова $x, y, z \in \Sigma^*$, для которых верно: $xyz = \omega, y \neq \varepsilon, |xy| \leqslant n$ и $\forall i \geqslant 0 \; xy^i z \in L_2$ (pumping).

Тогда возьмем $0^n 1^n = xyz$, при этом $|xy| \leq n$. Значит $y = 0^k$, где $0 < k \leq n$. Но тогда $xz = 0^{n-k}1^n$, то есть получилось, что $xz \notin L_2$, а это противоречие, так как xz должно принадлежать $L_2 (\forall i \geqslant 0 \; xy^i z \in L_2)$. Следовательно L_2 - не регулярный.

c)

$$L_3 = \{w \in \{0,1\}^* \mid w \text{ contains an even number of 1s}\}$$

Этот язык регулярный : $0^*(10^*10^*)^*$.



d)

$$\text{(d) } L_4 = \{1^{n^2} \mid n \in \mathbb{N}\}$$

Допустим язык L_4 - регулярный.

Проделаем аналогичные рассуждения пункту b, используем Pumping Lemma.

Тогда рассмотрим $\omega = 1^{k^2}$, $\omega = xyz$. Тогда $x = 1^\lambda, y = 1^\beta, z = 1^{k^2 - \lambda - \beta}$, при этом $\lambda + \beta \leq k$ и $\beta > 0$, $\lambda \geq 0$, значит $k - \beta \geq 0$, то есть $\beta \leq k$. Так как $\forall i \geqslant 0 \; xy^i z \in L_4$, то возьмем $i = 2$.

Рассмотрим $xy^2z = 1^{k^2+\beta} \in L_4$. Так как $\beta > 0$, то $k^2 + \beta > k$. А так как $\beta \leq k$, то $k^2 + \beta \leq k^2 + k < k^2 + 2k + 1 = (k+1)^2$. То есть $k^2 < k^2 + \beta < (k+1)^2$, при этом $k^2 + \beta$ должно быть точным квадратом. Заметим, что это невозможно, так как $k^2 + \beta$ зажато между двумя последовательными точными квадратами, то есть мы пришли к противоречию и L_4 не является регулярным.

Task 8

8. Consider a finite-state automaton $M = (\Sigma, Q, q_0, F, \delta)$ and a non-negative integer k. Let R_k be the relation on the set of states of M such that $s\ R_k\ t$ if and only if for every input string $w \in \Sigma^*$ with $|w| \leq k$, $\delta(s, w)$ and $\delta(t, w)$ are both final states or both not final states. Furthermore, let R^* be the relation on the set of states of M such that $s\ R^*\ t$ if and only if for every input string $w \in \Sigma^*$, regardless of length, $\delta(s, w)$ and $\delta(t, w)$ are both final states or both not final states.

a)

(a) Show that for every nonnegative integer k, R_k is an equivalence relation on S. Two states s and t are called k-equivalent if $s\ R_k\ t$.

Отношение эквивалентности : рефлексивно, симметрично, транзитивно.

1) Рефлексивность

$\forall \omega\ |\omega| \leq k, \delta(s, \omega)$ - либо допускающее, либо нет состояние, значит : sR_ks для $\forall \omega\ |\omega| \leq k$, то есть рефлексивно.

2) Симметричность

Если sR_kt, значит $\forall \omega\ |\omega| \leq k$, $\delta(s, \omega)$ и $\delta(t, \omega)$ - оба являются допускающими состояниями или оба не являются допускающими состояниями, значит $\delta(t, \omega)$ и $\delta(s, \omega)$ - оба являются допускающими состояниями или оба не являются допускающими состояниями, значит tR_ks для $\forall \omega\ |\omega| \leq k$, то есть симметрично.

3) Транзитивность

Если sR_km, то $\forall \omega\ |\omega| \leq k$, $\delta(s, \omega)$ и $\delta(m, \omega)$ - оба являются допускающими состояниями или оба не являются допускающими состояниями.

Если mR_kt, то $\forall \omega\ |\omega| \leq k$, $\delta(m, \omega)$ и $\delta(t, \omega)$ - оба являются допускающими состояниями или оба не являются допускающими состояниями.

Тогда $\delta(s,\omega)$, $\delta(m,\omega)$ и $\delta(t,\omega)$ - единовременно являются допускающими состояниями или не являются допускающими состояниями для $\forall\omega\ |\omega| \leq k$, значит sR_kt, то есть мы получили транзитивность.

b)

(b) Show that R^* is an equivalence relation on S.
Two states s and t are called $*$-equivalent if $s\ R^*\ t$.

Доказательство для R^* следует той же логике, что и для R_k, но без ограничения на длину входной строки ω. Таким образом, R^* также удовлетворяет рефлексивности, симметричности и транзитивности.

c)

(c) Show that if two states s and t are k-equivalent ($k > 0$), then they are also $(k-1)$-equivalent.

Если sR_kt, то это значит, что не существует ω, $|\omega| \leq k$, при котором $\delta(s,\omega)$ и $\delta(t,\omega)$ - не оба являются допускающими состояниями или не оба не являются допускающими состояниями, то есть не существует ω', $|\omega'| \leq k-1$, при котором $\delta(s,\omega')$ и $\delta(t,\omega')$ - не оба являются допускающими состояниями или не оба не являются допускающими состояниями, значит $sR_{k-1}t$.

d)

(d) Show that the equivalence classes of R_k are a *refinement* of the equivalence classes of R_{k-1}.

Если sR_kt для некоторого k, то для $\forall\omega$, $|\omega| \leq k$, $\delta(s,\omega)$ и $\delta(t,\omega)$ оба допускающие или оба не допускающие. Это означает, что s и t находятся в одном классе эквивалентности над R_k. Если s и t также находятся в одном классе в R_{k-1}, то для любого ω при $|\omega| < k$, $\delta(s,\omega)$ и $\delta(t,\omega)$ также оба допускающие или оба не допускающие, однако если они не в одном классе эквивалентности для R_{k-1}, то они не могут находится в одном классе эквивалентности для R_k, так как в R_{k-1} существует ω' $|\omega'| \leq k-1$, такое что $\delta(s,\omega')$ и $\delta(t,\omega')$ - оба не является допускающими или не допускающими.

e)

(e) Show that if two states s and t are k-equivalent for every non-negative integer k, then they are $*$-equivalent.

Если s и t k-эквивалентны для $\forall k \geq 0$, то для любой входной строки ω, независимо от ее длины, $\delta(s, \omega)$ и $\delta(t, \omega)$ оба допускающие или оба не допускающие. Это означает, что s и t находятся в одном классе эквивалентности R^* по определению, поэтому s и t $*$-эквивалентны.

f)

(f) Show that all states in a given R^*-equivalence class are final or all are not final.

Допустим это утверждение не верно и нашелся R^*-эквивалентный класс, в котором есть как допускающие, так и не допускающие состояния. Рассмотрим *-эквивалентный класс порядка 0, в таком случае наши исходные состояния поделятся на 2 множества, в каждом из которых будут либо только допускающие, либо только не допускающие. Но так как мы знаем, что если два состояния s и t не находились в одном классе эквивалентности на шаге меньше k, то они не могут находиться в одном классе эквивалентности на шаге k (мы можем взять ω' длины меньше k для которого не верно что $\delta(s, \omega')$ и $\delta(t, \omega')$ - оба допускающие или не допускающие), тогда мы пришли к противоречию, а это значит что данное утверждение верно.

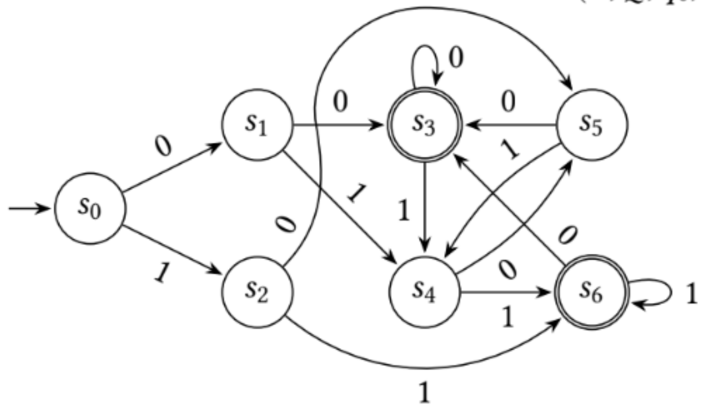
g)

(g) Show that if two states s and t are $*$-equivalent, then $\delta(s, a)$ and $\delta(t, a)$ are also $*$-equivalent for all $a \in \Sigma$.

Если s и t $*$-эквивалентны, то для $\forall \omega$, $\delta(s, \omega)$ и $\delta(t, \omega)$ оба допускающие или оба не допускающие(не зависимо от длины $|\omega|$). Тогда рассмотрим все ω с начальным символом a(так как нам нужно пройти через $\delta(s, a)$ и $\delta(t, a)$), мы знаем, s и t $*$-эквивалентны, то есть $\delta(s, \omega)$ и $\delta(t, \omega)$ приводят оба либо в допускающие, либо не допускающие, при этом мы пройдем через состояния $s' = \delta(s, a)$ и $t' = \delta(t, a)$. Тогда так как мы из s' и t' при любом $\omega' = \omega$ без начального символа a придем в те же состояния, что при $\delta(s, \omega)$ и $\delta(t, \omega)$, а мы знаем, что они оба либо допускающие, либо не допускающие, то s' и t' - $*$-эквивалентные.

Task 9

9. Consider the finite-state automaton $M = (\Sigma, Q, q_0, F, \delta)$ depicted below.



a)

(a) Find the k-equivalence classes of M for $k = 0, 1, 2, 3$.

k = 0

$k = 0$

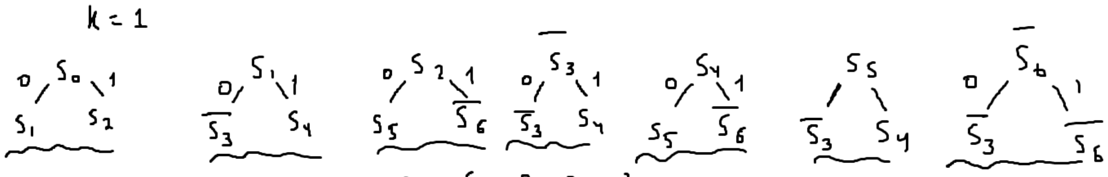
$\{s_0, s_1, s_2, s_4, s_5\}$ $\{\bar{s}_3, \bar{s}_6\}$

$\{s_0, s_1, s_2, s_4, s_5\}, \{s_3, s_6\}$

k = 1

Будем строить деревья, показывающие состояния по определенному переходу, тогда два состояния будут в одном классе эквивалентности, если при отбрасывании номера вершины в дереве они будут равны, то есть мы смотрим на то, является ли состояние допускающим или нет.

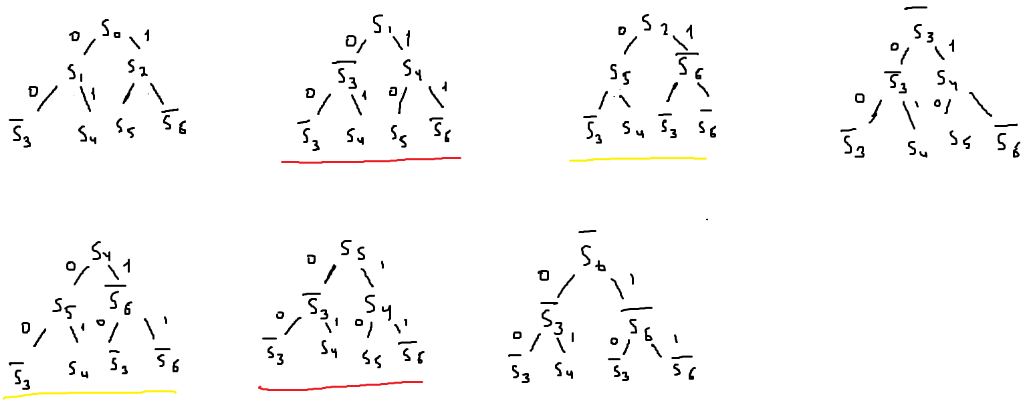
$k = 1$



$\{s_0\}, \{s_1, s_5\}, \{s_2, s_4\}, \{s_3\}, \{s_6\}$

$\{s_0\}, \{s_1, s_5\}, \{s_2, s_4\}, \{s_3\}, \{s_6\}$

k = 2



$\{s_0\}, \{s_1, s_5\}, \{s_2, s_4\}, \{s_3\}, \{s_6\}$

$\{s_0\}, \{s_1, s_5\}, \{s_2, s_4\}, \{s_3\}, \{s_6\}$

k = 3

Мы знаем, что если s_i и s_j не были в одном классе эквивалентности на шаге $k-1$, то они не могут оказаться в одном классе эквивалентности на шаге k. Поэтому проверим останутся ли $\{s_1, s_5\}$ и $\{s_2, s_4\}$ в одном классе эквивалентности.

Рассмотрим s_1 и s_5. Заметим, что при шаге $k = 1$ и $k = 2$, деревья с корнем в вершинах s_1 и s_5 были равны(кроме самого корня), поэтому на шаге $k = 3$, они также будут равны.

Аналогично с s_2 и s_4.

Тогда мы получаем классы аналогичные шагу $k = 2$.

$\{s_0\}, \{s_1, s_5\}, \{s_2, s_4\}, \{s_3\}, \{s_6\}$

b)

(b) Find the $*$-equivalence classes of M.

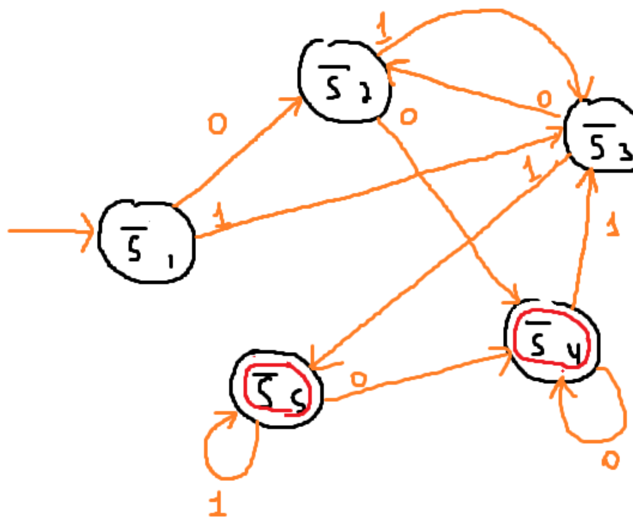
Аналогично шагу при $k = 3$ мы можем проделывать такие же заключения и получим следующее разбиение на классы:

$\{s_0\}, \{s_1, s_5\}, \{s_2, s_4\}, \{s_3\}, \{s_6\}$

c)

(c) Construct the quotient automaton $\overline{M}$ of M.

▹ The quotient automaton $\overline{M}$ of the deterministic finite-state automaton $M = (\Sigma, S, s_0, F, \delta)$ is the finite state automaton $\overline{M} = (\Sigma, \overline{S}, [s_0]_{R^*}, \overline{F}, \overline{\delta})$, where the set of states $\overline{S}$ is the set of R^*-equivalence classes of S; the transition function $\overline{\delta}$ is defined by $\overline{\delta}([s]_{R^*}, a) = [\delta(s, a)]_{R^*}$ for all states $[s]_{R^*}$ of $\overline{M}$ and input symbols $a \in \Sigma$; and $\overline{F}$ is the set consiting of R^*-equivalence classes of final states of M.



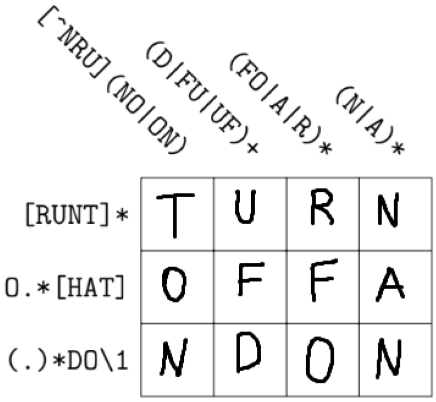
$$\{s_0\} = \overline{S_1}, \{s_1, s_5\} = \overline{S_2}, \{s_2, s_4\} = \overline{S_3}, \{s_3\} = \overline{S_4}, \{s_6\} = \overline{S_5}$$

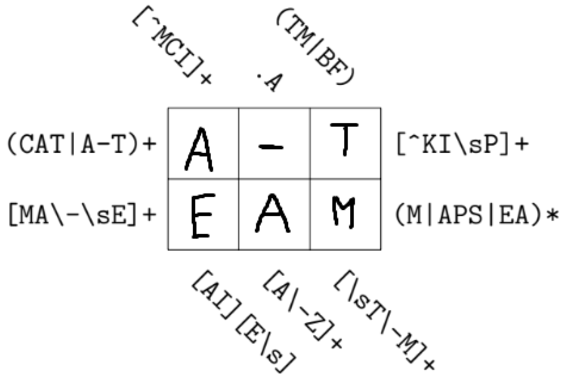
Task 10

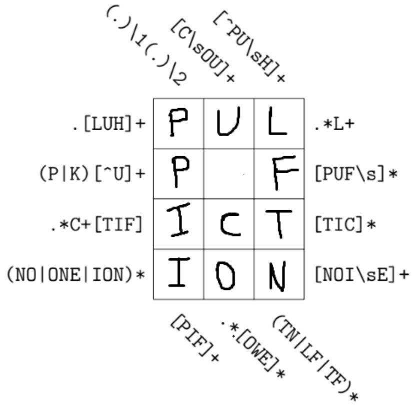
10. Solve the following regex crosswords. Fill each cell with a single ASCII character (an uppercase letter, a digit, a punctuation mark, or a space). Each row/column, when read left to right or top to bottom must match the regular expression(s) given for that row/column.

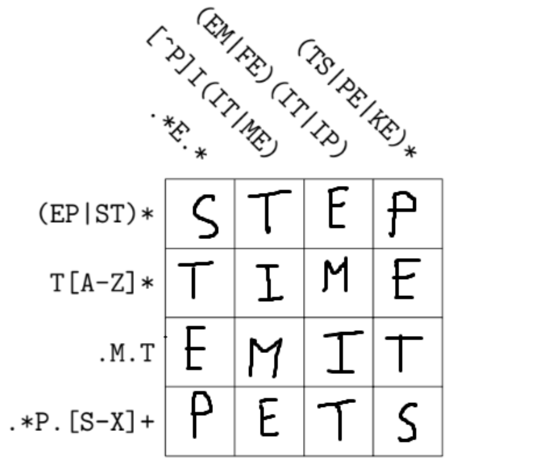
	[^SPEAK]+	EP|IP|EF
HE|LL|O+	H	E
[PLEASE]+	L	P

	(FI|A)+	(YE|OT)K	(.)[IF]+	[NODE]+	(FY|F|RG)+
(Y|F)(.)\2[DAF]\1	F	O	O	D	F
(U|O|I)*T[FRO]+	I	T	F	O	R
[KANE]*[GIN]*	A	K	I	N	G


[^NRU](NO|ON)
(D|FU|UF)+
(FO|A|R)*
(N|A)*
[RUNT]*
O.*[HAT]
(.)*DO\1
T U R N
O F F A
N D O N


[^MCI]+
.A
(TM|BF)
(CAT|A-T)+
[^KI\sP]+
[MA\-\sE]+
(M|APS|EA)*
[AI][E\s]
[A\-Z]+
[\sT\-M]+
A - T
E A M


(.)\1(.)\2
[C\sOU]+
[^PU\sH]+
.[LUH]+
.*L+
(P|K)[^U]+
[PUF\s]*
.*C+[TIF]
[TIC]*
(NO|ONE|ION)*
[NOI\sE]+
[PIF]+
.*[OWE]*
(TN|LF|TF)*
P U L
P F
I C T
I O N


.*E.*
[^P]I(IT|ME)
(EM|FE)(IT|IP)
(TS|PE|KE)*
(EP|ST)*
T[A-Z]*
.M.T
.*P.[S-X]+
S T E P
T I M E
E M I T
P E T S