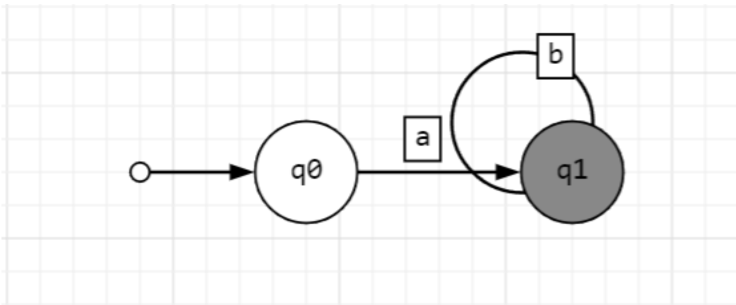
Automata Theory Homework. Dmitry Semenov, M3100, ISU 409537

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 $L\prime = \{w \in L(P)||w| \leq 5\}$. For “any” (.) and “negative” ([ˆ]) matches, assume that the alphabet is $\Sigma = \{a, b, c, d\}$

Dead states опущены, но подразумеваются

$(a) P_1 = ab^*$



$L' = \{a, ab, abb, abbb, abbbb\}$

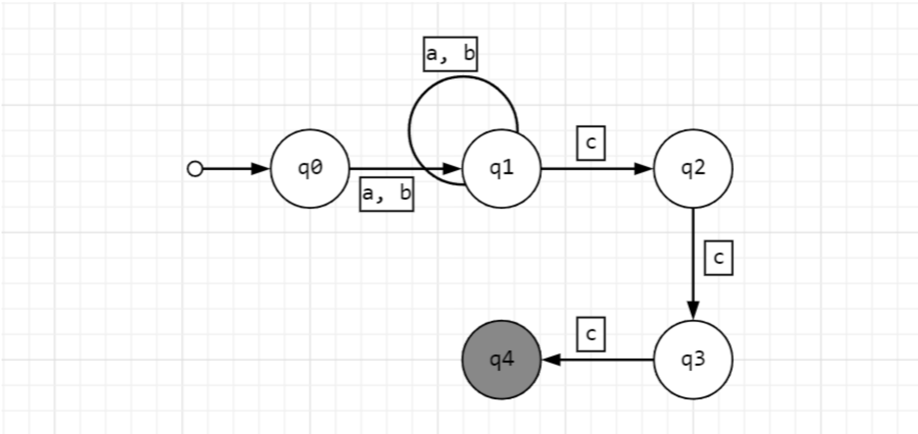
$\Rightarrow |L'| = 5$

$(b) P_2 = a + b?c$

$L' = \{ac, abc, aac, aabc, aaac, aaabc, aaaac\}$

$\Rightarrow |L'| = 7$

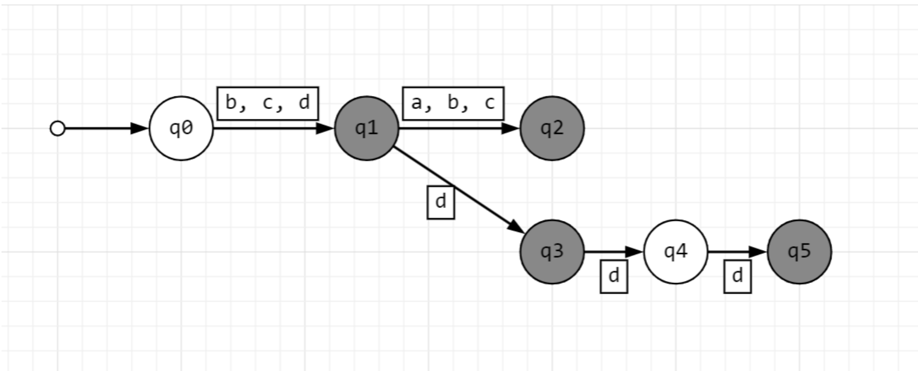
$(c) P_3 = [\hat{}cd] + c\{3\}$



$L' = \{accc, bccc, aaccc, abccc, bbccc, baccc\}$

$\Rightarrow |L'| = 6$

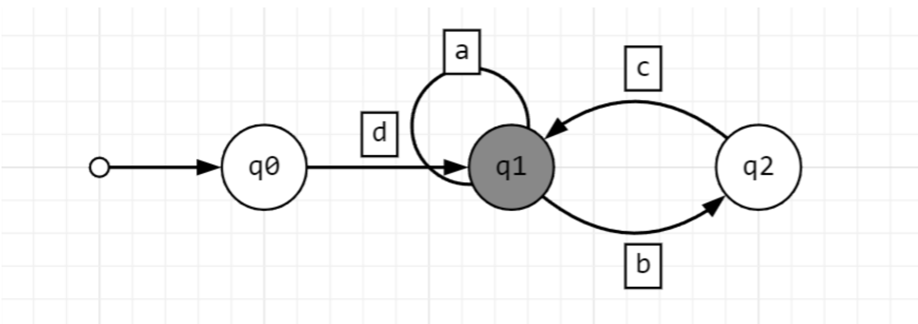
$(d) P4 = [\hat{\ } a](.|ddd)?$



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

$\Rightarrow |L'| = 18$

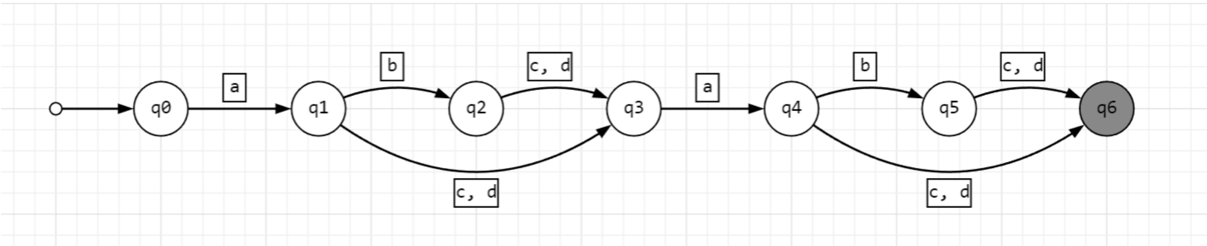
$(e) P5 = d(a|bc)*$



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

$\Rightarrow |L'| = 12$

$(f) P6 = ((a|ab)[cd])\{2\}$



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

$\Rightarrow |L'| = 12$

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

(a) $\langle string \rangle ::= \langle L \rangle + \langle D \rangle ? \langle L \rangle +$

$\langle L \rangle ::= a|b|c$

$\langle D \rangle ::= 0|1$

Это определение задает множество строк, которые начинаются с одного или более символов из множества $\{a, b, c\}$, за которыми может следовать один символ из множества $\{0, 1\}$, а затем заканчиваются одним или более символами из множества $\{a, b, c\}$.

Примеры строк из этого множества: $"ab"$, $"a1b"$, $"cccb"$, $"b0c"$, $"caaa"$

(b) $\langle string \rangle ::= \langle sign \rangle ? \langle N \rangle$

$\langle sign \rangle ::= `+' | `-'$

$\langle N \rangle ::= \langle D \rangle (\langle D \rangle | 0)^*$

$\langle D \rangle ::= 1|...|9$

Это определение задает множество строк, которые могут начинаться с знака $" + "$ или $" - "$, за которым следует целое число. Число может начинаться с любой цифры от 1 до 9, за ней могут следовать любые другие цифры, в том числе 0.

Примеры строк из этого множества: $" + 239"$, $" - 5"$, $"789"$, $" + 0"$, $" - 25"$

(c) $\langle string \rangle ::= \langle L \rangle^*(\langle D \rangle+)?\langle L \rangle^*$

$\langle L \rangle ::= x|y$

$\langle D \rangle ::= 0|1$

Это определение задает множество строк, которые могут содержать любое количество символов $'x'$ или $'y'$ (в том числе 0), за которыми может следовать одно или более вхождений символов $'0'$ или $'1'$, а затем опять любое количество символов $'x'$ или $'y'$ (в том числе 0).

Примеры строк из этого множества: $"xx011yx"$, $"x1y"$, $"y010x"$, $"xyx"$

(d) $\langle string \rangle ::= \langle C \rangle\langle R \rangle^*$

$\langle C \rangle ::= a|...|z|A|...|Z$

$\langle D \rangle ::= 0|...|9$

$\langle R \rangle ::= \langle C \rangle|\langle D \rangle|‘\_’$

Это определение задает множество строк, которые начинаются с любого символа из латинского алфавита (маленькая или заглавная буква), за которым может следовать любое количество символов из латинского алфавита, цифр или символов подчеркивания.

Примеры строк из этого множества: $"a123"$, "$B\_$", $"Zz\_9"$, $"C\_dE\_2"$

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

(a) Show that 111000 belongs to the language generated by G

Корректная последовательность шагов, приводящая к 111000:

$\mathbb{S} \to 1\mathbb{S} \to 11\mathbb{S} \to 111\mathbb{S} \to 11100A \to 111000$

(b) Show that 11001 does not belong to the language generated by G

Покажем конструктивно, что
G не может сгенерировать 11001.

Заметим, что 1 можно получить, только применением $\mathbb{S} \to 1\mathbb{S}$. То есть, с таким P мы не можем получить одиночный терминальный символ 1. Кроме того, после 0 не могут идти единицы в данном языке.

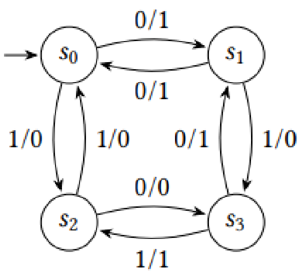
$\Rightarrow$ 11001 не принадлежит языку, который порождает G, так как он не содержит $.^{*}1$

(c) What is the language generated by G?

$$L(G) = \{000, 0000, 00000, ..., 1000, 11000, 110000, ...\}$$

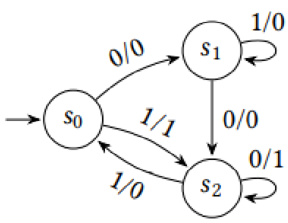
$$\Rightarrow L(G) = (1)^{*}(000)(0)^{*}$$

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



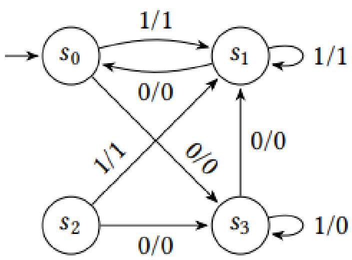
Input: 01110

Output: 10101



Input: 01110

Output: 00000



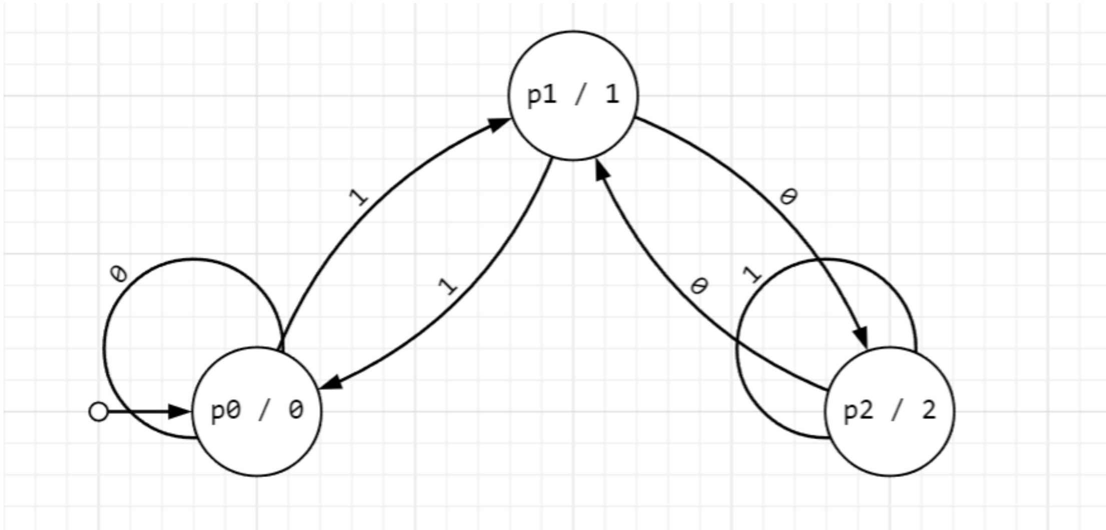
Input: 01110

Output: 00000

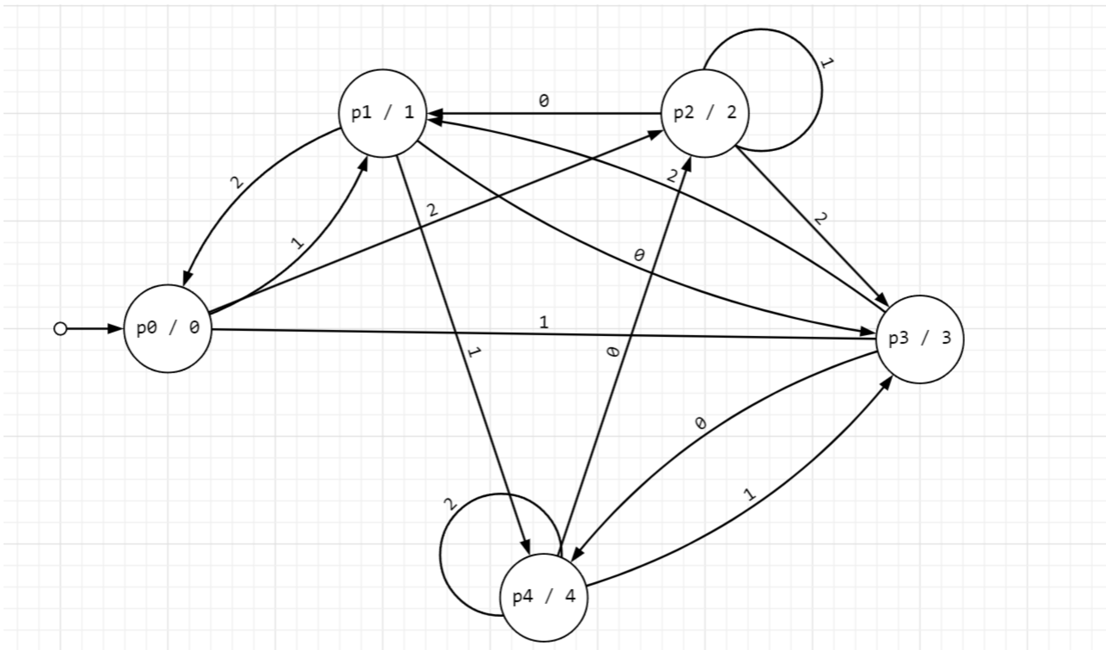
5. Construct a Moore machine for each of the following descriptions.

(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

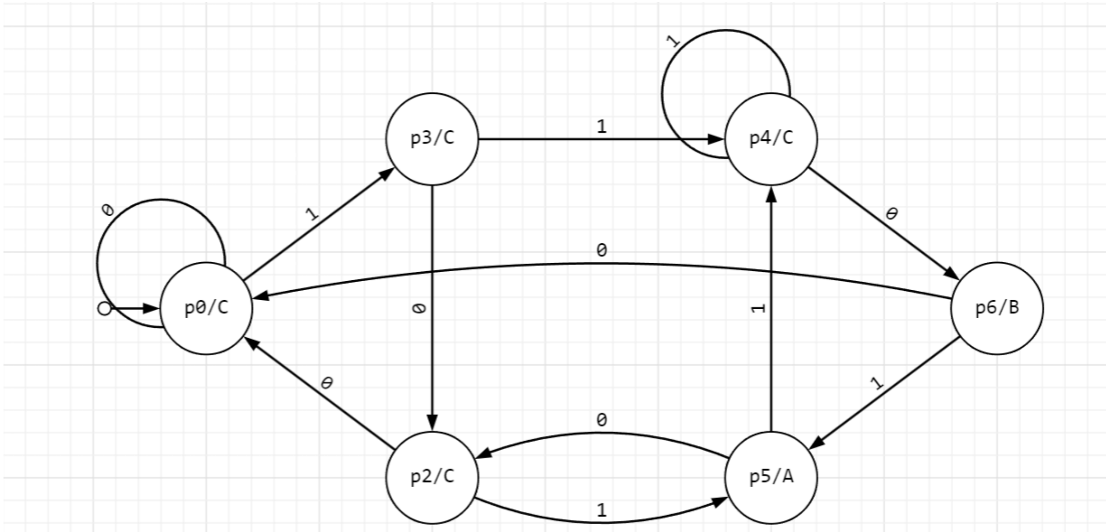
2; and
1010 (value 10) has residue 1.



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



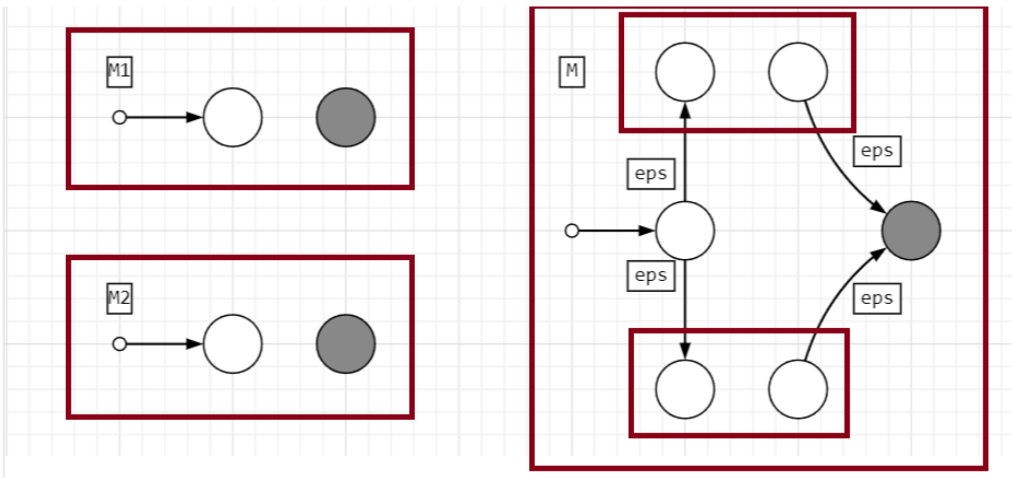
(c) Output A if the binary input ends with 101; output B if it ends with 110; otherwise output C.



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

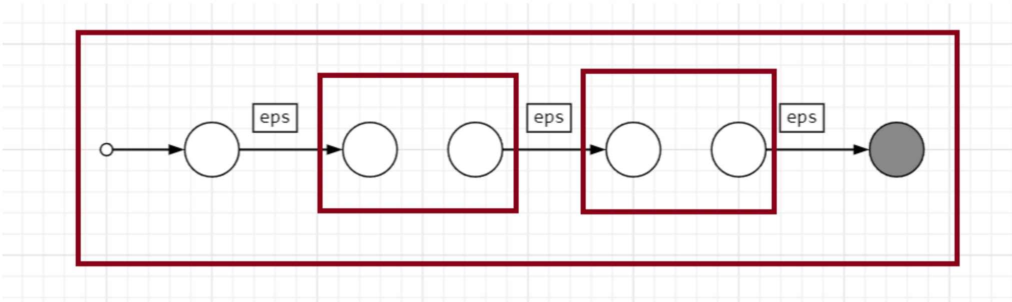
(a) Union, that is, if L_1 and L_2 are regular languages, then $L_1 \cup L_2$ is also regular.

Пусть L_1 и L_2 — регулярные языки над алфавитом Σ. По определению, существуют регулярные выражения R_1 и R_2, такие что $L(R_1) = L_1$ и $L(R_2) = L_2$. Тогда регулярное выражение $R = R_1 + R_2$ представляет объединение $L_1 \cup L_2$. Следовательно, $L(R)$ является регулярным языком, а значит, $L_1 \cup L_2$ также является регулярным



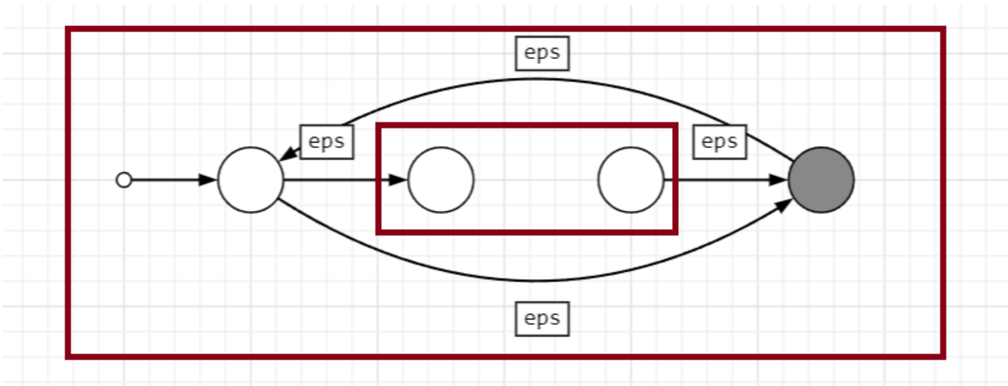
(b) Concatenation, that is, if L_1 and L_2 are regular languages, then $L_1 \cdot L_2$ is also regular.
Пусть
L_1 и L_2 - регулярные языки над алфавитом Σ. По определению, существуют

регулярные выражения R_1 и R_2, такие что $L(R_1) = L_1$ и $L(R_2) = L_2$. Тогда регулярное выражение $R = R_1 \cdot R_2$ представляет конкатенацию $L_1 \cdot L_2$. Следовательно, $L(R)$ является регулярным языком, а значит, $L_1 \cdot L_2$ также является регулярным



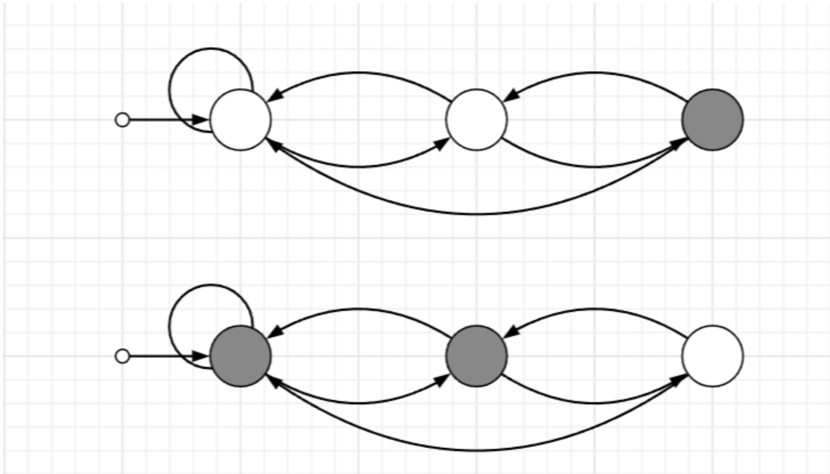
(c) Kleene star, that is, if L is a regular language, then L^* is also regular.

Пусть L — регулярный язык над алфавитом Σ. По определению, существует регулярное выражение R, такое что $L(R) = L$. Тогда регулярное выражение R^* представляет звезду Клини L^*. Следовательно, $L(R^*)$ является регулярным языком, а значит, L^* также является регулярным



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

Пусть L — регулярный язык над алфавитом Σ. Регулярные языки замкнуты относительно операции дополнения, что означает, что если язык L является регулярным, то его дополнение $\overline{L}$ также является регулярным. Регулярный язык может быть распознан конечным автоматом, значит $\overline{L}$ может быть получено простым инвертированием принимающих и не принимающих состояний автомата языка L, что также останется конечным автоматом. Таким образом, $\overline{L}$ также будет регулярным языком



(e) Intersection, that is, if L_1 and L_2 are regular languages, then $L1 \cap L2$ is also regular.

Пусть L_1 и L_2 — регулярные языки над алфавитом Σ. Мы знаем, что дополнение регулярного языка также является регулярным. Таким образом, $\overline{L_1}$ и $\overline{L_2}$ — регулярные. По закону Де Моргана, $\overline{L_1} \cup \overline{L_2} = \overline{L_1 \cap L_2}$. Поскольку регулярные языки замкнуты относительно объединения, $\overline{L_1} \cup \overline{L_2}$ является регулярным. Следовательно, $\overline{L_1 \cap L_2}$ является регулярным, что означает, что $L_1 \cap L_2$ также является регулярным

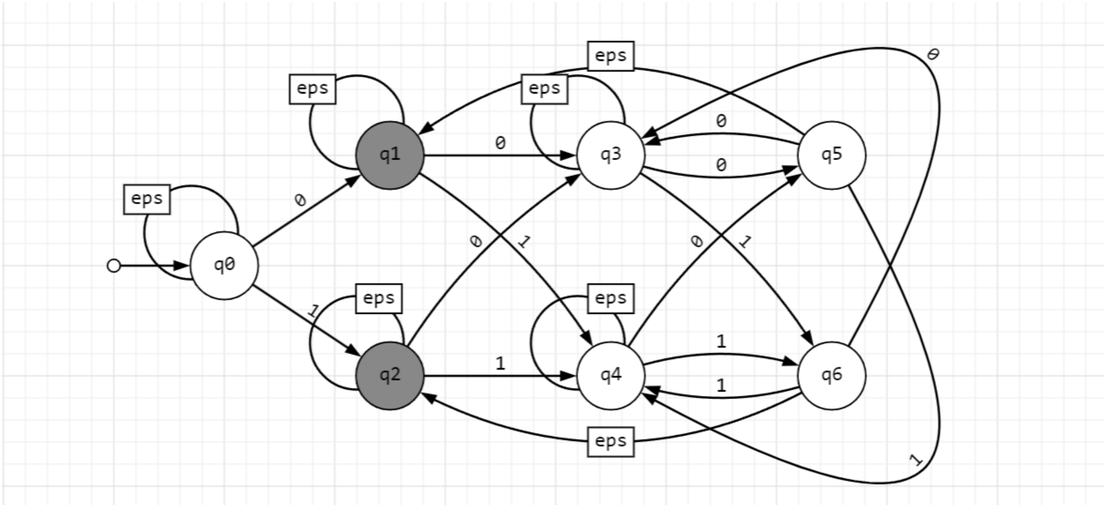
Автоматы для используемых в доказательстве замыканий уже построены.

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 $\varepsilon - NFA$.

Dead states опущены, но подразумеваются

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

Регулярное выражение для регулярного языка $L_1 = (0|1)(00|01|10|11)^*$



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

Предположим, что язык L_2 является регулярным. Тогда существует число накачки p для этого языка. Выберем слово $w = 0^p 1^p$, которое принадлежит языку L_2. $|w| \geq p$, поэтому оно удовлетворяет условиям леммы о накачке.

Согласно лемме о накачке, слово w можно разбить на три части $w = xyz$, где $\mid y \mid > 0$, $\mid xy \mid \leq p$ и $xy^i z$принадлежит языку L_2 $\forall i \geq 0$.

Так как $\mid xy \mid \leq p$, то y состоит только из нулей (так как первые p символов слова w – это нули).

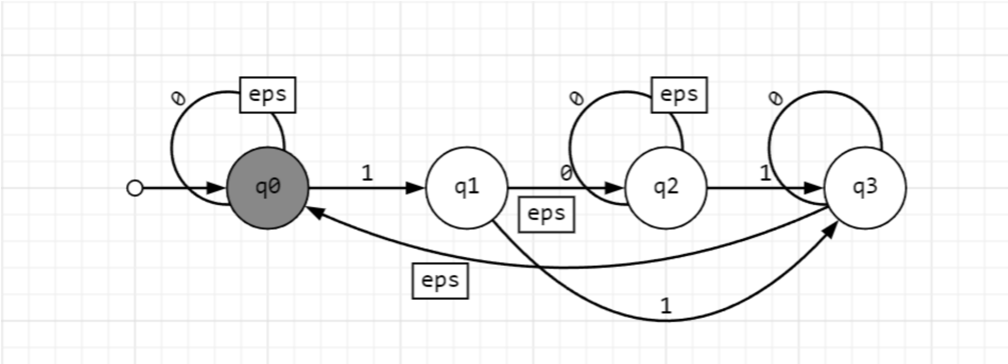
Если мы "накачаем" w, заменив y на y^2, мы получим слово $xy^2 z = 0^{p+|y|} 1^p$.

Однако, это слово не принадлежит языку L_2, так как количество нулей и единиц в нем не совпадает. Это противоречит лемме о накачке, которая утверждает, что $xy^i z$ должно принадлежать языку L_2 для всех $i \geq 0$.

$\Rightarrow L_2 = \{0^n 1^n | n \in N\}$ – нерегулярный

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

Регулярное выражение для регулярного языка $L_3 = 0^*(0^*10^*10^*)^*$



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

Предположим, что язык $L_4 = \{1^{n^2} \mid n \in N\}$ является регулярным и имеет константу накачки p. Рассмотрим слово $w = 1^{p^2}$, которое, принадлежит языку L_4.

Согласно лемме о накачке, w можно разбить на три части: $w = xyz$, где $\mid y \mid > 0$, $\mid xy \mid \leq p$, и xy^iz также принадлежит языку L_4 для любого $i \geq 0$.

Однако, если мы увеличим количество единиц в y, заменив y на y^2, мы получим слово $xy^2z = 1^{p^2+|y|}$. Это слово не принадлежит языку L_4, так как $p+ \mid y \mid$ не является полным квадратом для любого $\mid y \mid > 0$.

Это противоречит лемме о накачке.

$\Rightarrow L_4 = \{1^{n^2} | n \in N\} -$ нерегулярный

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 sR_kt 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 sR^*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) Show that for every non-negative integer k, R_k is an equivalence relation on S. Two states s and t are called $k-$equivalent if sR_kt.

Чтобы показать, что отношение R_k является отношением эквивалентности на множестве состояний S, нам нужно доказать, что оно обладает рефлексивностью, симметричностью и транзитивностью.

Рефлексивность: Для любого состояния s из S и для любой строки w с $\mid w \mid \leq k$, $\delta(s, w)$ будет либо конечным, либо не конечным состоянием. Таким образом, каждое состояние s будет $k-$эквивалентно самому себе

Симметричность: Если состояние s $k-$эквивалентно состоянию t, это значит, что для всех строк w с $\mid w \mid \leq k$, $\delta(s, w)$ и $\delta(t, w)$ оба либо конечные, либо не конечные состояния. Это же верно и в обратном направлении, если s $k-$эквивалентно t, то и t $k-$эквивалентно s

Транзитивность: Если состояние s $k-$эквивалентно состоянию t, и состояние t $k-$эквивалентно состоянию u, то для всех строк w с $\mid w \mid \leq k$, $\delta(s, w)$, $\delta(t, w)$ и $\delta(u, w)$ будут либо все конечными, либо все не конечными состояниями. Следовательно, s будет $k-$эквивалентно u

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

Полностью аналогично предыдущему пункту.

Рефлексивность: Для произвольного состояния s выполняется условие sR^*s, так как $\delta(s, w)$ будет либо конечным, либо не конечным состоянием, значит состояние s всегда эквивалентно самому себе

Симметричность: Если s R^*-эквивалентно t, то $\delta(s, w)$ и $\delta(t, w)$ либо оба финальные, либо оба не финальные для любой строки w, что означает, что t R^*-эквивалентно s, так как это верно и в другом направлении

Транзитивность: Если s R^*-эквивалентно t и t R^*-эквивалентно u, то $\delta(s, w)$, $\delta(t, w)$ и $\delta(u, w)$ либо оба финальные, либо оба не финальные для любой строки w, что означает, что s R^*-эквивалентно u

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

Если два состояния s и t $k-$эквивалентны, то для всех строк входа w с $\mid w \mid \leq k$, $\delta(s, w)$ и $\delta(t, w)$ оба либо конечные, либо не конечные состояния. Это означает, что для всех строк w с $\mid w \mid \leq k-1$, что является подмножеством строк с $\mid w \mid \leq k$, $\delta(s, w)$ и $\delta(t, w)$ также будут оба либо конечными, либо не конечными состояниями. То есть, если два состояния $k-$эквивалентны, то они также $(k-1)-$эквивалентны

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

Классы эквивалентности отношения R_k являются уточнением классов эквивалентности отношения R_{k-1}, потому что если два состояния эквивалентны относительно R_k, они также эквивалентны относительно R_{k-1}, как было показано в прошлом пункте. Это означает, что классы эквивалентности для R_k не могут быть шире, чем для R_{k-1}, *а могут только совпадать или иметь мощность меньше.*

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

Если два состояния s и t $k-$эквивалентны для каждого неотрицательного целого числа k, это означает, что для всех строк w, независимо от их длины, $\delta(s, w)$ и $\delta(t, w)$ оба либо конечные, либо не конечные состояния одновременно. Следовательно, s и t также R^*-эквивалентны по определению.

(f) Show that all states in a given R^*-equivalence class are final or all are not final.
Рассмотрим произвольный класс эквивалентности
R^*. Предположим, что в этом классе есть финальное состояние s и не финальное состояние t. Поскольку s и t R^*-эквивалентны, то для любой строки w, $\delta(s, w)$ и $\delta(t, w)$ либо оба финальные, либо оба не финальные. Но тогда $\delta(s, \varepsilon)$ и $\delta(t, \varepsilon)$ также либо оба финальные, либо оба не финальные, что противоречит тому, что s финальное, а t не финальное.

Таким образом, все состояния в классе эквивалентности R^* должны быть либо все финальные, либо все не финальные

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

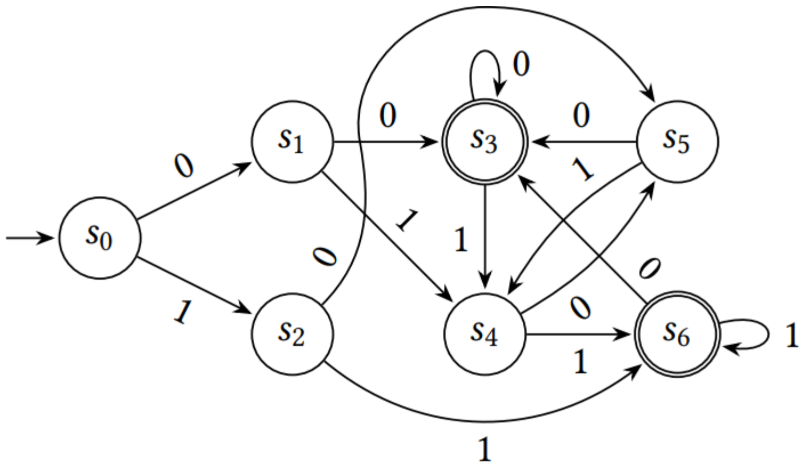
Пусть s и t $*-$эквивалентны. Это означает, что для любой строки w, $\delta(s, w)$ и $\delta(t, w)$ либо оба финальные, либо оба не финальные.

Теперь рассмотрим произвольный символ $a \in \Sigma$. Для любой строки wa, $\delta(s, wa) = \delta(\delta(s, w), a)$ и $\delta(t, wa) = \delta(\delta(t, w), a)$. Поскольку $\delta(s, w)$ и $\delta(t, w)$

либо оба финальные, либо оба не финальные, то и $\delta(s, wa)$ и $\delta(t, wa)$ будут либо оба финальные, либо оба не финальные.

Следовательно, $\delta(s, a)$ и $\delta(t, a)$ также $*-$эквивалентны

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



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

$k = 0 : \{s_0, s_1, s_2, s_4, s_5\}, \{s_3, s_6\}$

$k = 1, 2, 3 : \{s_0\}, \{s_1, s_5\}, \{s_2, s_4\}, \{s_3\}, \{s_6\}$

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

$[s_0]_{R*} = \{s_0\}$

$[s_1]_{R_*} = \{s_1, s_5\}$

$[s_2]_{R_*} = \{s_2, s_4\}$

$[s_3]_{R*} = \{s_3\}$

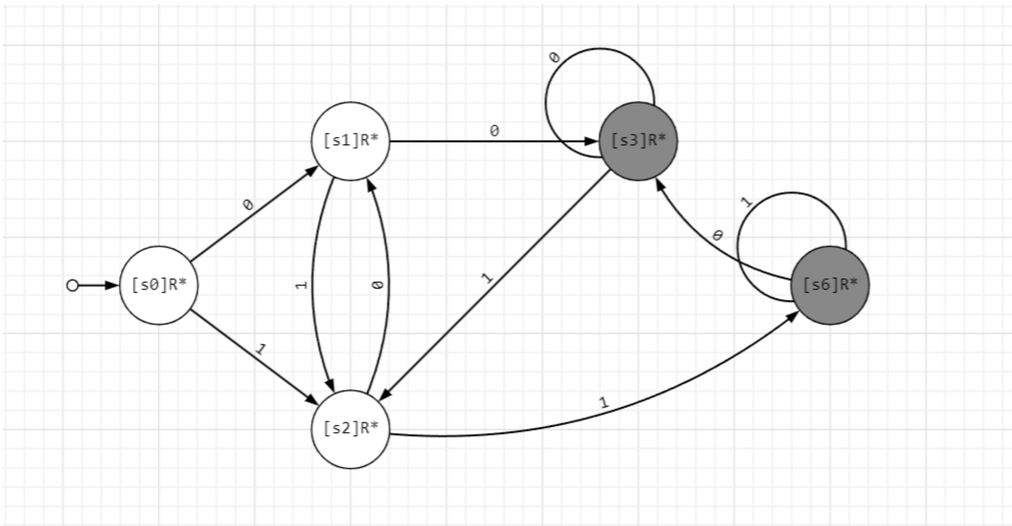
$[s_6]_{R*} = \{s_6\}$

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

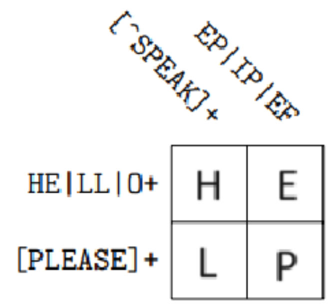
◅ The quotient automaton

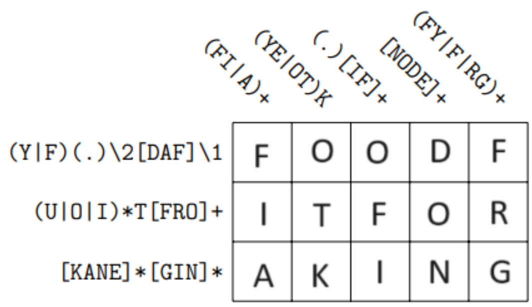
$\overline{M}$ of the deterministic finite-state automaton $M = (\Sigma, S, s_0, F, \delta)$ is the finite state automaton $M = (\Sigma, S, [s_0]_{R^*}, F, \delta)$, where the set of states S is the set of R^*-equivalence classes of S; the transition function δ is defined by $\delta([s]_{R^*}, a) =$

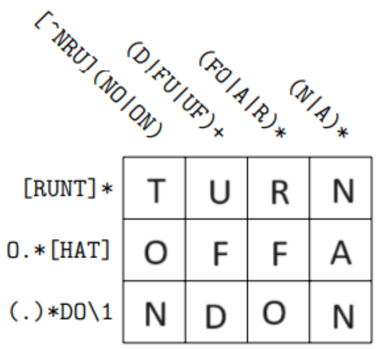
$[\delta(s,a)]_{R^*}$ for all states $[s]_{R^*}$ of M and input symbols $a \in \Sigma$; and F is the set consiting of R^*-equivalence classes of final states of M.

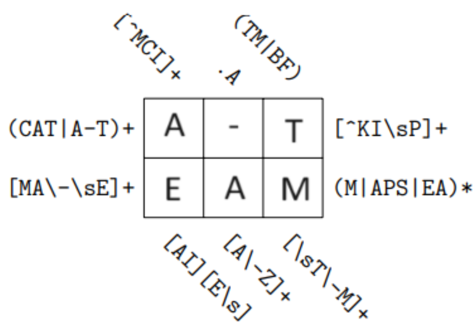


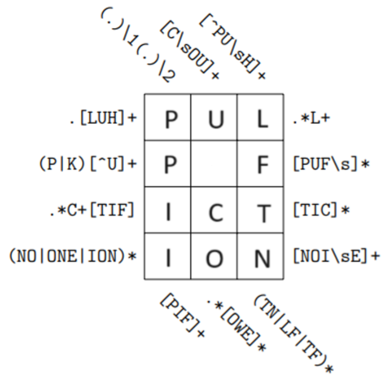
10. Solve the following regex crosswords. Fill each cell with a single ASCII character (an uppercase letter, a digit, a punctuation mark, or a space). Each row/column, when read left to right or top to bottom must match the regular expression(s) given for that row/column.



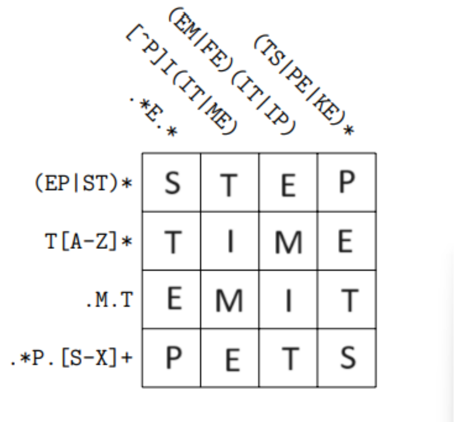








(.)\1(.)\2
[C\sOU]+
[^PU\sH]+
.[LUH]+
(P|K)[^U]+
.*C+[TIF]
(NO|ONE|ION)*
.*L+
[PUF\s]*
[TIC]*
[NOI\sE]+
[PIF]+
.*[OWE]*
(TN|LF|TF)*
P U L
P F
I C T
I O N



.*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