

Вариант 1

1. Построить левостолбчатую грамматику для языка десятичных чисел, делящихся на 4, и преобразовать её в правостолбчатую.
2. Построить минимальный автомат, распознающий регулярное выражение $(ab \mid a)b^*ab(a^* \mid ba)$.
3. Исследовать язык $\{w \mid |w|_a \neq 2 * |w|_b \ \& \ w \in \{a, b, c\}^*\}$.

Вариант 2

1. Построить регулярную грамматику для десятичных чисел, делящихся на 3, и регулярную грамматику для их реверсов (побуквенных обращений). Построить их формальное пересечение.
2. Построить минимальный автомат, распознающий регулярное выражение $a^*(bab^*)^*a$.
3. Исследовать язык $\{a^n b^m c^{n*m}\}$.

Вариант 3

1. Построить регулярную грамматику, распознающую строки, содержащие блоки, заключенные в $/^*$, $^*/$, и не содержащие знака $/$ (эти блоки могут и отсутствовать в строке). Построить регулярную грамматику, распознающую строки, содержащие блоки, заключенные в ** ** и не содержащие знака * . Построить их формальное пересечение. Алфавит $\{/, *, a\}$.
2. Построить минимальный автомат, распознающий регулярное выражение $1^*[0]^*(10)^+2^?$ в алфавите $\{0, 1, 2\}$.
3. Исследовать язык $\{w^R z^R w z \mid w, z \in \{a, b\}^* \ \& \ |w| > 1\}$.

Вариант 4

1. Построить регулярную грамматику, распознающую список идентификаторов. Список начинается и заканчивается квадратными скобками, разделитель элементов в списке — запятая. Идентификатор — последовательность латинских букв и цифр, начинающаяся со строчной буквы. Между элементами списка может быть сколько угодно пробелов.
2. Построить минимальный автомат, распознающий регулярное выражение $(abc \mid bbb)(ab)?a^*(a \mid b)$.
3. Исследовать язык $\{z_1 w z_2 w \mid w \in \{a, b\}^* \ \& \ |w| > 0 \ \& \ z_i \in \{b, c\}^*\}$.

Вариант 5

1. Построить регулярную грамматику или регулярное выражение, распознающие последовательность объявлений целых переменных в C-подобном синтаксисе. Объявление переменной начинается с объявления типа `int`, затем через минимум один пробел следует идентификатор — последовательность букв и цифр, начинающаяся не с цифры — и опционально объявление значения через знак равенства. Считаем, что переменным присваиваются лишь целочисленные значения. Элементы последовательности отделяются друг от друга точкой с запятой. Между ними могут быть переводы строки. Расстановка пробелов произвольна.
2. Построить минимальный автомат, распознающий регулярное выражение $http[s]? : // (a | b)^+ . (com | ru)$.
3. Исследовать язык $\{www \mid w \neq z_1vvz_2 (|v| > 0) \ \& \ w \in \{a, b\}^*\}$.

Вариант 6

1. Построить регулярную грамматику, распознающую следующую конструкцию: идентификатор, за которым следуют единственные фигурные скобки, внутри которых записано одно или более выражение, содержащее идентификаторы, пробелы и единственный знак равенства и заканчивающееся точкой с запятой и, возможно, переводом строки (код \$). Если выражение не заканчивается переводом строки, тогда после него обязан стоять пробел. Идентификатор — строка из латинских букв.
2. Построить минимальный автомат, распознающий регулярное выражение $(a(a \mid b)^+a) \mid (ba^*b)$.
3. Исследовать язык $\{w \mid |w|_{abb} = |w|_{bba} \text{ \& } w \in \{a, b\}^*\}$.

Вариант 7

1. Построить регулярную грамматику для следующей структуры перечислений. Перечисление начинается с тега `en_b`, заканчивается тегом `en_e` и содержит как минимум одно поле. Поле начинается тегом `en_i` и заканчивается переводом строки (код `$`). После тега поля может идти любая последовательность букв, пробелов и знаков подчёркивания, за исключением тега `en_b`, кроме того, от тега `en_i` и до перевода строки не должно встречаться и тегов конца или поля перечисления.
2. Построить минимальный автомат, распознающий регулярное выражение $([be]^* b [be]^+ e [be]^* |[be]^*)$ в алфавите $\{a, b, c, e\}$.
3. Исследовать язык $\{w \mid w \in \{(\,,)\}^+ \text{ \& } w \text{ — префикс правильной скобочной последовательности, отличный от правильной скобочной последовательности}\}$.

Вариант 8

1. Построить регулярное выражение, описывающее регулярные выражения не больше чем с одним уровнем вложенности скобок $[,]$. Входное регулярное выражение может содержать операции $*$, $+$, $?$, $|$ и латинские буквы.
2. Построить минимальный автомат, распознающий регулярное выражение `if $id=id$ then num (; | else num ;)` . Пробелы значащие, id — последовательность латинских букв, num — неотрицательное целое число.
3. Исследовать язык $\{w^{v|_a}v^{w|_a} \mid w, v \in \{a, b\}^* \ \& \ |w|_a > 1 \ \& \ |v|_a > 1\}$.
- 4.

Вариант 9

1. Построить регулярную грамматику для слов, начинающихся и кончающихся одной и той же буквой, и слов, не содержащих подстрок aa , bb , cc в алфавите $\{a, b, c\}$. Построить их формальное пересечение.
2. Построить минимальный автомат, распознающий троичные числа, делящиеся на 2. Распознавание числа с конца — от младших разрядов к старшим.
3. Исследовать язык $\{w \mid |w|_0 \neq |w|_1 \vee |w|_0 \neq |w|_2 \ \& \ w \in \{0, 1, 2\}^+\}$.
(Указание: рассмотрите язык $\{w \mid |w|_1 > |w|_0\}$)

Вариант 10

1. Построить регулярные грамматики для скобочных выражений, в которых ни одна закрывающая скобка не стоит впереди открывающей, и для сбалансированных скобочных выражений без вложенных скобок. Построить их формальное пересечение.
2. Построить минимальный автомат, распознающий троичные числа, делящиеся на 2. Распознавание числа с начала — от старших разрядов.
3. Исследовать язык $\{w_1cw_2 \mid w_i \in \{a, b, c\}^* \ \& \ |w_1|_a = |w_2|_b \ \& \ |w_1|_b = |w_2|_a\}$.

Вариант 11

1. Построить регулярную грамматику для слов, в которых не больше двух подслов ab , и регулярную грамматику для слов, в которых встречается минимум по две буквы a и b . Алфавит $\{a, b\}$. Построить их формальное пересечение.
2. Построить минимальный автомат, распознающий язык $(a^*b^*aa^*b^*ba^*b^*a)^*$.
3. Исследовать язык $\{a^{n!}\} \cup \{a^{n^2}\}$.

Вариант 12

1. Построить регулярную грамматику для тождественно истинных логических формул без скобок, со связками только \vee , $\&$ и \neg (с обычным приоритетом операций) и константами Т, F.
2. Построить минимальный автомат, распознающий язык $d[*d](+d[*d]^*)^* = d$. Здесь $*$ и $+$ не в верхних индексах — символы языка, так же как и буква d и знак равенства.
3. Исследовать язык $\{a^n cz \mid n = |z|_a \ \& \ z \in \{a, b\}^*\}$.

Вариант 13

1. Построить регулярную грамматику для арифметических выражений, принимающих значение 0. Выражения содержат числа от 0 до 9, а также знаки сложения и умножения (с обычным приоритетом операций). Скобок нет.
2. Построить минимальный автомат, распознающий язык списков неотрицательных целых чисел. Список начинается и заканчивается квадратной скобкой, разделитель — запятая. Список может быть пустым.
3. Исследовать язык таких SRS, записанных в одну строку: $(W_1 \rightarrow W_2\$)^+$ (где $W_1, W_2 \in \{[a - z]\}^+$), для которых существует перекрытие некоторой левой части W_1 подсловом некоторой другой левой части. Например, как в данной SRS: $bab \rightarrow b\$a \rightarrow a\$$. Здесь левая часть второго правила, а именно a , является подсловом левой части первого правила, а именно bab . Пробелы в языке SRS не допускаются, в примере они расставлены для лучшей читаемости SRS.

Вариант 14

1. Построить регулярное выражение или регулярную грамматику для арифметических выражений, принимающих значение 1. Выражения содержат только числа 0, 1, 2, а также знаки сложения и умножения (с обычным приоритетом операций). Скобок нет.
2. Построить минимальный автомат, распознающий правильно записанные неотрицательные числа с плавающей точкой в двоичной системе.
3. Исследовать язык $\{a^b b \mid a \in \{S\}^+ \text{ \& } b \in \{I\}^+\}$ и под a^b подразумевается слово, полученное b приписываниями слова a к самому себе $\}$.

Вариант 15

1. Построить регулярное выражение, принимающее рассортированные по возрастанию списки, содержащие только числа 0, 1, 2. Список начинается и заканчивается квадратными скобками, разделитель элементов в списке — запятая. Между элементами может быть произвольное число пробелов.
2. Построить минимальный автомат, распознающий язык $f(&f)^*(\vee f(&f)^*)^*$ в алфавите $\{\&, \vee, f\}$.
3. Исследовать язык перепутанных скобочных последовательностей. В нём встречаются два вида скобок: круглые и квадратные, причём если стереть все квадратные скобки, получится правильная скобочная последовательность из круглых скобок, и если стереть все круглые скобки, тоже останется правильная скобочная последовательность. Между собой скобки могут быть перепутаны, например, как в слове $([])]$.

Вариант 16

1. Построить регулярное выражение или грамматику для распознавания тождественно ложных логических формул в ДНФ, если известно, что формулы имеют один уровень вложенности скобок (т.е. вложенных скобочных конструкций нет). Формулы содержат символы $\&$, \vee , \neg , $(,)$, A , B , где A , B — логические переменные. Приоритет операций обычный. Конъюнкты могут быть заключены в скобки или нет.
2. Построить нормальную форму Грейбах для грамматики, распознающей правильные скобочные термы, содержащие только скобки двух типов: круглые и квадратные.
3. Исследовать язык $\{1(+1)^n - (2(+2)^m) = 0 \mid 1 * (n + 1) = 2 * (m + 1)\}$. Т.е. этот язык состоит из правильных равенств, где справа всегда нуль, а слева — сумма из $n + 1$ единицы, из которой вычитается сумма из $m + 1$ двоек.

Вариант 17

1. Построить регулярное выражение, описывающее плоские образцы рефала. Рефал-образец — это последовательность констант, строк и переменных, разделенных пробелами. Константы — это последовательности латинских букв и цифр, начинающиеся с буквы, без кавычек. Строки — последовательности латинских букв, цифр, знаков препинания и пробелов в одинарных кавычках (апострофах). Апострофы внутри строк считаем запрещенными. Переменные — это символы e , t или s , после которых следует точка и непустая последовательность латинских букв и цифр. Например, $e.A0$, $s.01s$ — переменные.
2. Построить нормальную форму Грейбах для грамматики правильных скобочных последовательностей, распознающих два вида скобок (круглые и фигурные). Причём внутри круглых скобок запрещены фигурные. Т.е. выражения вида $\{\}$ считаются некорректными.
3. Исследовать язык $\{w_1w_2 \mid w_1 \neq w_2 \ \& \ |w_i| > 2\}$. Алфавит не унарный.

Вариант 18

1. Построить регулярное выражение, описывающее строку из латинских букв, точек и пробелов, устроенную следующим образом. Строка начинается с заглавной буквы; пробелы не могут идти друг за другом, а также не могут начинать или заключать строку; за точкой всегда идёт либо конец строки, либо пробел, и после этого пробела — заглавная буква.
2. Построить нормальную форму Грейбах для грамматики истинных выражений вида $M \cdot N = T$, где M, T, N — унарные числа вида $s^n z$. Т.е. 0 — это z , $n+1$ — это выражение для числа n , к которому приписана в начале буква s . Примеры таких выражений: $sz - sz = z$, $ssssz - ssz = ssz$. Примеры выражений, не принадлежащих языку: $ssz - s = sz$, $sss - ss = s$, $ssz - sz = z$.
3. Исследовать язык $\{w_1 w_2 \mid w_2 = h(w_1) \ \& \ w_1 \in \{a, b\}^*\}$, где h — это гомоморфизм, определенный как $h(a) = \varepsilon$, $h(b) = b$.

Вариант 19

1. Построить регулярное выражение, описывающее тексты в \LaTeX , имеющие корректную структуру математических блоков (в том числе таких блоков может и не быть в тексте). Блок либо начинается последовательностью $\$$ и заканчивается ею же, либо начинается знаком $\$$ и заканчивается им же. Блоки не должны содержать незэкранированных знаков $\$$. Экранирование делается с помощью символа \backslash . Считаем, что кроме вышеописанных знаков блоков, в тексте не встречается ничего, кроме пробелов и латинских букв, а в математических блоках могут быть ещё и цифры.
2. Построить нормальную форму Хомского для грамматики логических формул с кванторами $\forall \dots(\dots)$ и $\exists \dots(\dots)$, связкой $\&$, единственным предикатом $P(1)$ и переменными x и y . Переменная не может входить в формулу не как аргумент предиката, но может входить как аргумент предиката, не будучи связанной квантором. Т.е. выражение $x \& P(x)$ некорректно, выражение $\forall x(\forall x(P(y) \& P(x))) \& P(x)$ корректно.
3. Исследовать язык $\{a^{2*n}a^{3*k} \mid k > n\}$.

Вариант 20

1. Построить регулярное выражение, описывающее корректные скобочные логические формулы. Формула считается корректной, если её смысл однозначно определяется исходя из приоритета операций. В формулах могут быть связки \vee , \neg , $\&$, \Rightarrow , а также заглавные латинские буквы — булевские переменные. Поскольку \Rightarrow — не ассоциативная связка, считаем, что больше чем одно её вхождение в скобочную формулу приводит к неоднозначности.
2. Построить нормальную форму Грейбах для грамматики правильно записанных арифметических выражений, содержащих скобки, знаки $+$ и $-$, унарный минус и литеру n , обозначающую любое число.
3. Исследовать язык скобочных последовательностей таких, что они не могут быть префиксами правильно построенных скобочных последовательностей. Скобки одного типа — круглые.

Вариант 21

1. Построить регулярную грамматику для выражений, содержащих произвольно расставленные круглые скобки и правильно расставленные квадратные скобки, не вложенные в другие квадратные скобки. Построить аналогичную грамматику для произвольно стоящих квадратных скобок и правильно стоящих невложенных круглых скобок. Найти их пересечение.
2. Построить нормальную форму Хомского для языка арифметических выражений над двоичными числами, равных 0. Допустимые операции — сложение и умножение.
3. Исследовать язык логических формул с кванторами от переменных вида x_i , где $i \in \mathbb{N}$, в которых каждая переменная, входящая в формулу, должна быть связана квантором. Переменные под кванторами запрещается объединять, т.е. записи $\forall x_1 x_2(\dots)$ и $\forall x_1, x_2(\dots)$ считаем некорректными. Кроме кванторных выражений и скобок, формулы могут содержать только унарный предикатный символ $P(\bullet)$ и связку \Rightarrow .

Вариант 22

1. Построить регулярную грамматику для правильно записанных арифметических выражений без скобок с единственной операцией $+$ и натуральными числами, и грамматику для выражений в алфавите $\{0, 1, +\}$, в которых нет подстрок, состоящих из двух одинаковых символов. Построить их пересечение.
2. Построить нормальную форму Грейбах для грамматики палиндромов, не содержащих подслов вида aa , bb , cc , в алфавите $\{a, b, c\}$.
3. Исследовать язык программ, вычисляющихся без ошибок, вида $[<\text{операция}>]^*$, где операция — это присваивание $<\text{переменная}> = <\text{выражение}>$; переменная — последовательность латинских букв; выражение — арифметическое выражение, содержащее натуральные числа, $+$, $*$ и переменные. Программа считается вычисляющейся без ошибок, если правая часть любой операции содержит только те переменные, которым уже были присвоены значения какой-либо предшествующей операцией.

Вариант 23

1. Построить регулярную грамматику для регулярных выражений без скобок над алфавитом $\{a, b\}$, которые не принимают пустую строку. Допустимые операции — альтернатива, положительная итерация $(+)$, итерация Клини, знак вопроса.
2. Построить нормальную форму Грейбах для грамматики языка $\{a^n b^{m+1} = c^m d^{n+1}\}$, где n и m могут быть и нулями. Знак равенства входит в символы языка.
3. Исследовать язык арифметических выражений без вычитания, в которых нет деления на 0. То есть в языке не допускаются выражения, например, вида $1/(1 * 0)$. Арифметическое выражение содержит любые натуральные числа, операции умножения, деления и сложения, а также скобки. Делитель не заключается в скобки лишь тогда, когда состоит из единственного числа.

Вариант 24

1. Построить регулярную грамматику для регулярных выражений без скобок, обязательно допускающих пустую строку. Алфавит констант: $\{0, 1\}$, допустимые операции — итерация Клини, альтернатива, знак вопроса, положительная итерация (+). Примеры выражений, которые не допускаются таким языком — $0|1$, 1^*1 , $1?00+$. Примеры допускаемых: $1^*1?$, $1^*|0+$.
2. Построить нормальную форму Грейбах для грамматики $S \rightarrow aSc \mid B$, $B \rightarrow bBc \mid Sc \mid \varepsilon$.
3. Исследовать язык $\{a^{k_1}b^{k_2} \dots a^{k_{n-1}}b^{k_n}c^{k_1+k_2+\dots+k_n} \mid \text{для всех возможных } n, k_i > 0\}$.

Вариант 25

1. Построить регулярную грамматику для правильно записанных неотрицательных чисел с плавающей точкой. Построить регулярную грамматику для реверсов (обращенных записей) таких чисел. Построить их пересечение.
2. Построить нормальную форму Грейбах для грамматики

$$S \rightarrow (V) \mid S \Rightarrow S \quad V \rightarrow T \mid V \& T \quad T \rightarrow \neg T \mid a \mid b$$

3. Исследовать язык $\{a^{k_1}b^{k_2+k_1}\dots a^{k_{99}+k_{98}}b^{k_{100}+k_{99}} \mid k_i > 0\}$.

Вариант 26

1. Построить регулярную грамматику для правильно построенных арифметических выражений, содержащих скобки одного уровня вложенности, знак $+$ и переменные x и y . Построить регулярную грамматику реверсов таких выражений. Найти их пересечение.
2. Построить нормальную форму Грейбах для грамматики

$$S \rightarrow S; S \mid T = T \quad T \rightarrow T + T \mid n$$

3. Исследовать язык $\{wa^n b^n w \mid w \in \{a, b\}^*\}$.

Вариант 27

1. Построить регулярную грамматику для правильно построенных логических формул над константами Т и F без скобок и со связками $\&$, \neg , принимающих значение F. Построить грамматику всех бесскобочных логических формул со связками $\&$, \neg и константой Т. Построить их пересечение.
2. Построить нормальную форму Грейбах для грамматики

$$S \rightarrow SbSa \mid SaSb \mid \varepsilon$$

3. Исследовать язык $\{w \mid \forall w_1, z, w_2 (w = w_1 z w_2 \ \& \ z = a^k \ \& \ w_1 = w_3 b \ \& \ w_2 = b w_4 \Rightarrow |w_1|_a = |w_2|_a = k)\}$. Можно ограничиться алфавитом $\{a, b\}$.

Вариант 28

1. Построить регулярную грамматику для слов, содержащих подслово aba . Построить регулярную грамматику для слов, содержащих подслово baa . Построить их пересечение. Алфавит $\{a, b, c\}$.
2. Построить нормальную форму Грейбах для грамматики

$$S \rightarrow A \Rightarrow B \quad A \rightarrow 0 \mid (B \Rightarrow A) \quad B \rightarrow 1 \mid (S)$$

3. Исследовать язык правильных регулярных выражений, обязательно принимающих, в числе прочего, пустое слово. Регулярные выражения могут содержать буквы $A-z$, а также операции альтернативы, итерации Клини, положительной итерации $(+)$ и квадратные скобки, формирующие группы.

Вариант 29

1. Построить регулярную грамматику для слов, не содержащих подслов ab . Построить грамматику для слов, не содержащих подслова aa в конце. Построить их формальное пересечение.
2. Построить нормальную форму Грейбах для грамматики

$$S \rightarrow S [] S | T(T) \quad T \rightarrow TT | [T] | ()$$

3. Исследовать язык слов в алфавите $\{a, b, c\}$ таких, что перестановкой букв в них можно получить палиндром.

Вариант 30

1. Построить регулярную грамматику для слов, на чётных позициях которых стоит буква a . Построить регулярную грамматику для слов, у которых в каждом подслове из трёх символов совпадают первая и последняя буквы. Построить их формальное пересечение. Алфавит $\{a, b\}$.
2. Построить нормальную форму Грейбах для грамматики регулярных выражений. Регулярные выражения могут содержать букву a , квадратные скобки, знак альтернативы и итерацию Клини.
3. Исследовать язык $\{w \mid \forall w_1, z, w_2 (w = w_1 z w_2 \ \& \ z = a^k \ \& \ w_1 = w_3 b \ \& \ w_2 = b w_4 \Rightarrow |w_1|_a + |w_2|_a = k)\}$. Можно ограничиться алфавитом $\{a, b\}$.

Вариант 31

1. Построить регулярную грамматику для слов, либо начинающихся и заканчивающихся на a , либо не содержащих букв a . Построить регулярную грамматику для слов, в которых нет подслов w длины 2 и больше, которые не содержат букв a . Построить их формальное пересечение. Алфавит $\{a, b, c\}$.
2. Построить нормальную форму Грейбах для грамматики

$$S \rightarrow S_1 S_1 \quad S_1 \rightarrow S_2 S_2 \quad S_2 \rightarrow ab \mid S$$

3. Исследовать язык всех возможных подслов правильных скобочных последовательностей. Допускаются два типа скобок — круглые и квадратные.

Вариант 32

1. Построить регулярную грамматику для слов, содержащих подслово aba . Построить регулярную грамматику для слов, содержащих подслово baa . Построить их пересечение. Алфавит $\{a, b\}$.
2. Построить нормальную форму Грейбах для грамматики

$$S \rightarrow S[S]S \mid S(S)S \mid [] \mid ()$$

3. Исследовать язык всех возможных подслов правильных скобочных последовательностей. Тип скобок единственный — круглые.

Вариант 33

1. Построить регулярную грамматику для распознавания списка правил регулярной (право- или леволинейной) грамматики. Правила могут содержать знак альтернативы, нетерминалы A-Z и терминалы a-z. Правила разделяются знаком конца строки (\$). Начальный нетерминал S.
2. Построить минимальный автомат, распознающий выражения вида $A_1 = A_2$, где A_i — это натуральное число или запись суммы нескольких натуральных чисел.
3. Исследовать язык $\{w \mid \exists v, z_1 (|v| > 0 \ \& \ (w = vvz_1 \vee w = z_1vv))\}$. Алфавит $\{a, b\}$.

Вариант 34

1. Построить регулярную грамматику для чисел в троичной системе, на чётных позициях которых стоят нули. Построить регулярную грамматику для чисел в *двоичной* системе, делящихся на 3. Построить их формальное пересечение.
2. Построить нормальную форму Грейбах для регулярной грамматики, содержащей нетерминалы S (стартовый), A , B , C , терминал a и порождающей хотя бы одну бесконечную цепочку. Правила грамматики имеют синтаксис $N \rightarrow tN\$$ или $N \rightarrow t$ ($\$$ — конец строки, N — нетерминал, t — терминал).
3. Исследовать язык списков натуральных чисел, рассортированных по неубыванию, причём если некоторое число входит в такой список, то оно входит в него минимум дважды. Считаем, что список заключён в квадратные скобки, и элементы списка разделяются только запятой.

Вариант 35

1. Построить регулярную грамматику для распознавания списка правил контекстно-свободной грамматики в форме Грейбах. Правила могут содержать знак альтернативы, нетерминалы A-Z и терминалы a-z, и представляют собой выражение «нетерминал -> правая часть». Правила разделяются знаком конца строки (\$). Начальный нетерминал S.
2. Построить нормальную форму Грейбах для грамматики дизъюнктивных нормальных форм без переменных (с константами T и F). Скобки допускаются только вокруг конъюнктов, но их может и не быть.
3. Исследовать язык $\{w \mid |w|_{aba} = |w|_{baa} = |w|_{aab} \ \& \ |w|_{aaa} = 0 \ \& \ w \in \{a, b, c\}^*\}$.