

### **Правила сдачи ДЗ по РК-1**

Номер вашего варианта — это номер варианта РК-1, который вы взяли на каникулах. Уже решённые задачи из варианта перерешивать не надо, но лучше послать заново вместе с другими. Если вы не переписывали РК-1 на каникулах, тогда нужно написать на почту [a\\_nevod@mail.ru](mailto:a_nevod@mail.ru) и запросить номер варианта для ДЗ.

Стоимость полностью решенного ДЗ — 6 баллов. Решенные задачи можно высылать на почту, желательно все в одном письме. После приёма решённых задач нужно будет их защитить на лабе (ответить на несколько вопросов по работе). В случае защиты через зум потребуется включённая камера.

### Вариант 1

1. Построить левостолбчатую грамматику для языка десятичных чисел, делящихся на 4, и преобразовать её в правостолбчатую.
2. Построить регулярное выражение, распознающее текст в Markdown с правильными блоками курсива. Считаем, что основной текст состоит только из латинских букв, пробелов и точек, причём после точки должен стоять пробел или конец текста, а до точки — буква, и двойных пробелов нет.
3. Построить минимальный автомат, распознающий регулярное выражение  $(ab \mid a)b^*ab(a^* \mid ba)$ .
4. Построить нормальную форму Грейбах для грамматики тождественно истинных логических выражений только со связками  $\&$  и  $\Rightarrow$ , переменной  $A$  и скобками.
5. 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 регулярное выражение, распознающее логические формулы в ДНФ и без вложенных скобок. Переменные —  $A, B, C$ .
3. Построить минимальный автомат, распознающий регулярное выражение  $a^*(bab^*)^*a$ .
4. Построить нормальную форму Грейбах для грамматики не тождественно истинных логических выражений только со связками  $\&$  и  $\Rightarrow$ , переменной  $A$  и скобками.
5. 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 регулярное выражение, распознающее правила праволинейной грамматики, записанные через точку с запятой. Нетерминалы —  $An$ , где  $n$  — неотрицательное натуральное число, терминалы — только  $a, b$ .  $\varepsilon$ -правил нет, прочие символы языка — только стрелка  $\rightarrow$ .
3. Построить минимальный автомат, распознающий регулярное выражение  $1^*[0]^*(10)^+2^?$  в алфавите  $\{0, 1, 2\}$ .
4. Построить нормальную форму Грейбах для грамматики арифметических выражений с  $+$ ,  $-$ ,  $*$ , скобками и унарным минусом. Токен числа единственный — это  $n$ , расстановка пробелов произвольная.
5. 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 регулярное выражение, распознающее арифметические выражения без скобок, имеющие строго положительное значение. Выражения включают двоичные числа от 0 и больше, а также сложение и умножение.
3. Построить минимальный автомат, распознающий регулярное выражение  $(abc|bbb)(ab)?a^*(a|b)$ .
4. Построить нормальную форму Грейбах для языка правил праволинейной грамматики без  $\epsilon$ -правил над нетерминалами  $S, M, N, L$  и единственным терминалом  $a$ , таких что порождаемый ими язык не пуст. Стартовым всегда считается нетерминал  $S$ , правила перечисляются через точку с запятой.  
Примеры: слово  $N \rightarrow a$ ;  $S \rightarrow aN$ ;  $N \rightarrow aM$  принадлежит искомому языку, а слово  $S \rightarrow aS$  и слово  $M \rightarrow a$  не принадлежат.
5. 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 С-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 регулярную грамматику правильно записанных неотрицательных десятичных чисел с плавающей точкой. Построить регулярную грамматику реверсов правильно записанных неотрицательных десятичных чисел с плавающей точкой. Построить их формальное пересечение.
3. Построить минимальный автомат, распознающий регулярное выражение  $http[s]? : // (a | b)^+ . (com | ru)$ .
4. Построить нормальную форму Грейбах следующей грамматики.
$$S \rightarrow BSA | \varepsilon$$
$$A \rightarrow ASb | BB | a | b$$
$$B \rightarrow SBa | AA$$
5. Исследовать язык  $\{www \mid w \neq z_1vvz_2 (|v| > 0) \ \& \ w \in \{a, b\}^*\}$ .

### Вариант 6

1. Построить конечный автомат или регулярную грамматику, распознающую правильные арифметические выражения, содержащие только операции  $+$  и  $*$  (без структурных скобок) над кольцом  $\mathbb{Z}_4$ .
2. 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, за исключением, возможно, одного совпадают первая и вторая буквы. Построить их формальное пересечение. Алфавит  $\{a, b, c\}$ .
3. Построить минимальный автомат, распознающий регулярное выражение  $((abb)^*a)^*(ab)^*$ .
4. Построить нормальную форму Грейбах для следующей грамматики
$$S \rightarrow SS \mid A$$
$$A \rightarrow Sb \mid BB \mid ac$$
$$B \rightarrow Ba \mid AA$$
5. Исследовать язык  $\{w \mid w = w_1zw_2 \text{ \& } z = a^k \text{ \& } |w_1|_a + |w_2|_a = k\}$ . Алфавит  $\{a, b\}$ .

### Вариант 7

1. Построить конечный автомат или регулярную грамматику, распознающую чётные числа в системе счисления по основанию 5.
2. Построить регулярную грамматику для языка строк, у которых между каждыми двумя вхождениями буквы 2 есть буква 0. Построить регулярную грамматику для языка строк, содержащих чётное количество единиц. Построить их формальное пересечение. Алфавит  $\{0, 1, 2\}$ .
3. Построить минимальный автомат, распознающий регулярное выражение  $((q|er|re)^*(e|r)^*)^*$ .
4. Построить нормальную форму Грейбах для следующей грамматики
$$S \rightarrow ASA \mid z$$
$$A \rightarrow AcB \mid ad \mid d \mid a$$
$$B \rightarrow Sa$$
5. Исследовать язык  $\{w \mid w = w_1z_1w_2z_2 \text{ \& } z_1 = a^k \text{ \& } z_2 = b^s \text{ \& } |w_1|_a + |w_2|_a = k \text{ \& } |w_1|_b + |w_2|_b = s\}$ . Алфавит  $\{a, b\}$ .



### 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 регулярное выражение, распознающее язык последовательностей правил контекстно-свободной грамматики без  $\varepsilon$ -правил. Правило может содержать нетерминалы  $A$ ,  $B$ ,  $C$ , терминалы  $a$ ,  $b$ ,  $c$  и стрелку  $\rightarrow$ , правила друг от друга отделяются запятой, после которой может также стоять сколько угодно пробелов.
3. Построить минимальный автомат, распознающий троичные числа, делящиеся на 2. Распознавание числа с конца — от младших разрядов к старшим.
4. Построить нормальную форму Грейбах для грамматики
$$S \rightarrow SS \mid AaS \mid \varepsilon$$
$$A \rightarrow ASb \mid B \mid \varepsilon$$
$$B \rightarrow bBa \mid S$$
5. 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 регулярное выражение, распознающее слова в алфавите  $\{a, b, c\}$ , в которых можно переставить буквы так, чтобы получился палиндром.
3. Построить минимальный автомат, распознающий троичные числа, делящиеся на 2. Распознавание числа с начала — от старших разрядов.
4. Построить нормальную форму Грейбах для грамматики
$$S \rightarrow BSA \mid SS \mid \varepsilon$$
$$A \rightarrow ASb \mid S \mid ab$$
$$B \rightarrow SBa \mid \varepsilon$$
5. 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 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 регулярное выражение для текстов, содержащих не вложенные друг в друга теги  $\langle i \rangle$ ,  $\langle /i \rangle$  и  $\langle b \rangle$ ,  $\langle /b \rangle$ . Если в тексте присутствует открывающий тег, за ним обязательно должен следовать закрывающий, причём без закрытия тега  $\langle i \rangle$  не может появиться новый тег  $\langle i \rangle$ , и  $\langle b \rangle$  аналогично. Считаем, что в остальном тексте есть только пробелы и латинские буквы.  
Пример правильной расстановки тегов в тексте: Abb  $\langle i \rangle$ aa $\langle b \rangle$  $\langle /b \rangle$  $\langle /i \rangle$ A. Примеры неправильной расстановки:  $\langle i \rangle \langle i \rangle \langle /i \rangle$ aaa $\langle /i \rangle$ ,  $\langle i \rangle$ A $\langle /b \rangle$ .
3. 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.
4. Построить нормальную форму Грейбах для грамматики
$$S \rightarrow SSA \mid AB \mid \varepsilon$$
$$A \rightarrow SbA \mid \varepsilon \mid aaaA$$
$$B \rightarrow BaS \mid \varepsilon$$
5. Исследовать язык  $\{a^n cz \mid n = |z|_a \ \& \ z \in \{a, b\}^*\}$ .

### 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 регулярную грамматику для бесскобочных логических формул над переменной  $A$  со связками  $\&$  и  $\neg$ . Построить регулярную грамматику для слов в алфавите  $\{A, \&, \neg\}$ , в которых нет двух подряд идущих одинаковых символов. Построить их формальное пересечение.
3. Построить минимальный автомат, распознающий правильно записанные неотрицательные числа с плавающей точкой в двоичной системе.
4. Построить нормальную форму Грейбах для грамматики языка арифметических выражений в стиле С над переменной  $x$ , содержащих операцию сложения, предваряющий инкремент  $++x$ , инкремент  $x++$  и структурные скобки.
5. Исследовать язык  $\{a^b b \mid a \in \{S\}^+ \& b \in \{I\}^+\}$  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 регулярную грамматику для языка сумм натуральных чисел (без скобок), с произвольной расстановкой пробелов между элементами суммы. Построить регулярную грамматику для языка над  $\{0, 1, +, \_ \}$  (где  $\_$  — пробел), в которой все буквы на чётных позициях одинаковые. Построить их формальное пересечение.
3. Построить минимальный автомат, распознающий язык  $f(\&f)^*(\vee f(\&f)^*)^*$  в алфавите  $\{\&, \vee, f\}$ .
4. Построить нормальную форму Грейбах для грамматики многочленов второй степени от переменных  $x, y, z$  с коэффициентами в  $\mathbb{Z}[2]$ . Многочлены могут содержать операции сложения и вычитания, знаки умножения внутри мономов не указываются, степеней нет (т.е.  $x^2$  пишется с помощью явного умножения:  $xx$ ).
5. 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 регулярную грамматику для языка сумм натуральных чисел (без скобок), с произвольной расстановкой пробелов между элементами суммы. Построить регулярную грамматику для языка над  $\{0, 1, +, \_ \}$  (где  $\_$  — пробел), в которой во всех подстроках из трёх идущих подряд букв две крайние одинаковы. Построить их формальное пересечение.
3. Найти минимальный автомат, распознающий выражение  $[a|b]^*[aa|bb]^+(aba|b)a^*$ .
4. Построить нормальную форму Грейбах для грамматики, распознающей правильные скобочные термы, содержащие только скобки двух типов: круглые и квадратные.
5. 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 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 регулярную грамматику для языка сумм двоичных чисел (без скобок), с произвольной расстановкой пробелов между элементами суммы, но без сдвоенных пробелов. Построить регулярную грамматику для языка над  $\{0,1,+,\_ \}$  (где  $\_$  — пробел), в которой во всех подстроках из трёх идущих подряд букв хотя бы две одинаковы. Построить их формальное пересечение.
3. Построить минимальный автомат, распознающий язык выражения  $(0(00|11)^*1)|(1(01|10)^*0)1^*$ .
4. Построить нормальную форму Грейбах для грамматики истинных выражений вида  $M-N=T$ , 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:  $ssz - s = sz$ ,  $sss - ss = s$ ,  $ssz - sz = z$ .
5. 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  $\text{\LaTeX}$ , 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 регулярную грамматику для языка логических формул без отрицания и без скобок над переменными  $A, B$  (связки  $\&, \vee$ ). Построить регулярную грамматику для слов в алфавите  $\{A, B, \&, \vee\}$ , в которых встречается не больше чем две различные буквы (ограничений на число вхождений одинаковых букв нет). Построить их формальное пересечение.
3. Построить минимальный автомат, распознающий язык выражения  $1^*0^*(1^+|0^+)(00|11)^+$ .
4. 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.
5. 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 регулярную грамматику для языка арифметических выражений, содержащих сложение, вычитание, унарный минус и натуральные числа. Построить регулярную грамматику для слов в алфавите  $\{0, 2, +, -\}$ , в которых после каждого 0 обязательно идёт 2 и нет подстрок из двух одинаковых символов. Построить их формальное пересечение.
3. Построить минимальный автомат, распознающий язык выражения  $a^*(b|ab|abb)^*(a|ba|baa)^*$ .
4. Построить нормальную форму Грейбах для грамматики правильно записанных арифметических выражений, содержащих скобки, знаки  $+$  и  $-$ , унарный минус и литеру  $n$ , обозначающую любое число.
5. Исследовать язык скобочных последовательностей таких, что они не могут быть префиксами правильно построенных скобочных последовательностей. Скобки одного типа — круглые.

## Вариант 21

1. Построить регулярное выражение, распознающее все (не обязательно приведённые) полиномы степени 1 от переменной  $x$ , содержащие только операцию сложения и неотрицательные коэффициенты. Знак умножения в мономах опускается.
2. 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.
3. Построить минимальный автомат, распознающий язык выражения  $0((+1(0|1)^*)^*(+0)^* + 1^+)^+$ . Здесь  $+$  не в верхнем индексе — символ алфавита.
4. Построить нормальную форму Хомского для языка арифметических выражений над двоичными числами, равных 0. Допустимые операции — сложение и умножение.
5. Исследовать язык логических формул с кванторами от переменных вида  $xi$ , 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  $\Rightarrow$ .