Ноябрьская контрольная по ТРЯП
задачи, решения и критерии

ФУПМ 2017

Разбалловка и общие положения

неуд	удовл	хорошо	отлично
$0 \leqslant \Sigma \leqslant 13$	$14 \leqslant \Sigma \leqslant 19$	$20 \leqslant \Sigma \leqslant 28$	$29 \leqslant \Sigma \leqslant 42$
1: 0-7, **2:** 8-13	**3:** 14-16, **4:** 17-19	**5:** 20-22, **6:** 23-25, **7:** 26-28	**8:** 29-32, **9:** 33-36, **10:** 37-42

В случае дробной суммы баллов, перед выставлением оценки происходит арифметическое округление.

Приведённые ниже критерии оценивания выработанны с учётом типовых ошибок и определяют общую политику проверки, однако заведомо не могут покрыть всевозможные случаи. При некритериальном случае, проверяющий оценивает решение исходя из здравого смысла и духа критериев. В случае несогласия с оценкой за работу, студент имеет право подать апелляцию. Апелляцию нужно подать в письменном виде во время показа работ (или в день показа до 23:59 только в случае невозможности это сделать во время показа). Жалобы и замечания, возникшие после показа работ можно направить письмом всем преподавателям курса (адреса сообщаются при желании подать таковые).

Внимание! подача апелляции преведёт к полному пересмотру работы апелляционной комиссией, в результате чего оценка может как повыситься, так и понизиться.

Напоминаем положения, указанные в преамбуле к контрольной.

1. Ответы, включая правильные, при отсутствии решений оцениваются в 0 (ноль) баллов.

2. Объекты, полученные «методом внимательного вглядывания», без доказательства корректности построения оцениваются в 0 (ноль) баллов.

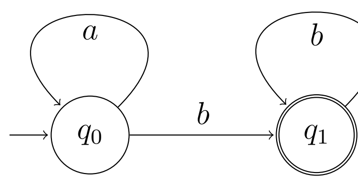
3. При формулировке вопроса «верно ли, что», в случае положительного ответа приведите доказательство, а в случае отрицательного – контрпример. Верное рассуждение без контрпримера оценивается в половину задачи.

Задачи, решения и критерии

Задача 1 (3+4).

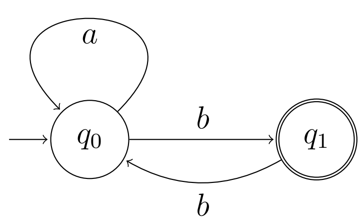
1 $\langle$i$\rangle$. 1. Постройте для РВ $a^*(b|bb)(a^*bb^*|b^*)^*$ над алфавитом $\{a, b\}$ эквивалентный минимальный всюду определённый ДКА $\mathcal{A}$.

2. Постройте праволинейную грамматику для языка из всех слов чётной длины, входящих в язык $\{a, b\}^* \setminus L(\mathcal{B})$, где $\mathcal{B}$ – автомат, заданный диаграммой



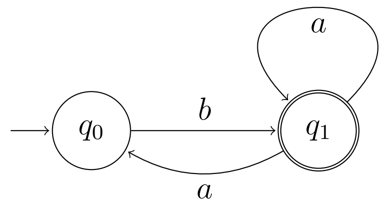
1 $\langle$ii$\rangle$. 1. Постройте для РВ $(a^*bb^*|b^*)^*a^*(b|bb)$ над алфавитом $\{a, b\}$ эквивалентный минимальный всюду определённый ДКА $\mathcal{A}$.

2. Постройте праволинейную грамматику для языка из всех слов нечётной длины, входящих в язык $\{a, b\}^* \setminus L(\mathcal{B})$, где $\mathcal{B}$ – автомат, заданный диаграммой



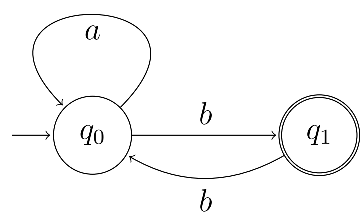
1 ⟨iii⟩. 1. Постройте для РВ $b^*(a|aa)(b^*aa^*|a^*)^*$ над алфавитом $\{a, b\}$ эквивалентный минимальный всюду определённый ДКА $\mathcal{A}$.

2. Постройте праволинейную грамматику для языка из всех слов чётной длины, входящих в язык $\{a, b\}^* \setminus L(\mathcal{B})$, где $\mathcal{B}$ – автомат, заданный диаграммой



1 ⟨iv⟩. 1. Постройте для РВ $(b^*aa^*|a^*)^*b^*(a|aa)$ над алфавитом $\{a, b\}$ эквивалентный минимальный всюду определённый ДКА $\mathcal{A}$.

2. Постройте праволинейную грамматику для языка из всех слов нечётной длины, входящих в язык $\{a, b\}^* \setminus L(\mathcal{B})$, где $\mathcal{B}$ – автомат, заданный диаграммой



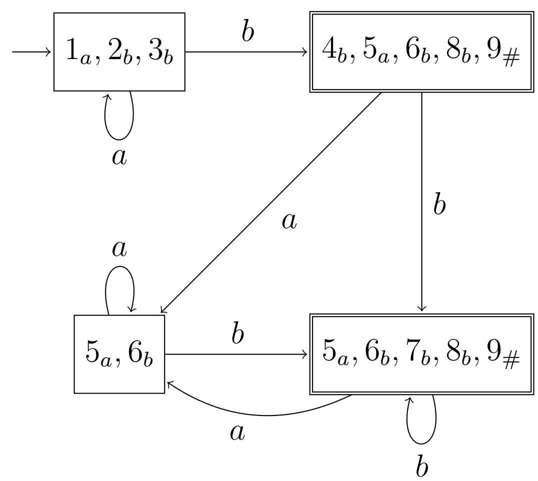
Решение. Мы не будем подробно описывать исполнение алгоритмов (например, мы опустим вычисление по дереву атрибутов nullable, firstpos и lastpos), однако приведём ключевые этапы решения, которые позволят проверить правильность исполнения алгоритмов.

1 ⟨i⟩. **1.** Занумеруем все символы, входящие в регулярное выражение, укажем результаты вычисления функции **followpos** и построим получившийся по алгоритму «РВ → ДКА» автомат и соответсвующий минимальный автомат.

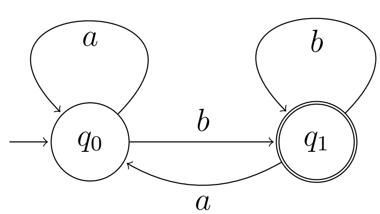
$$a_1^*(b_2|b_3b_4)(a_5^*b_6b_7^*|b_8^*)^*\#_9$$

№	**Followpos**
1	1,2,3
2	5,6,8,9
3	4
4	5,6,8,9
5	5,6
6	5,6,7,8,9
7	5,6,7,8,9
8	5,6,8,9

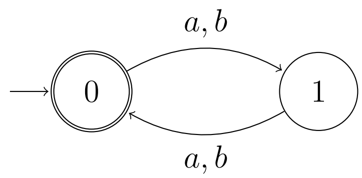
Автомат по РВ:



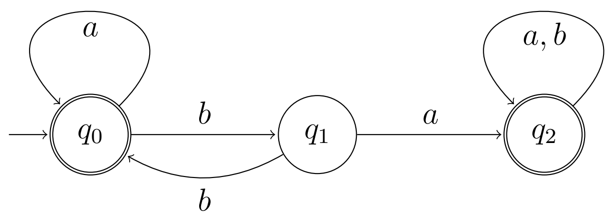
Минимальный автомат:



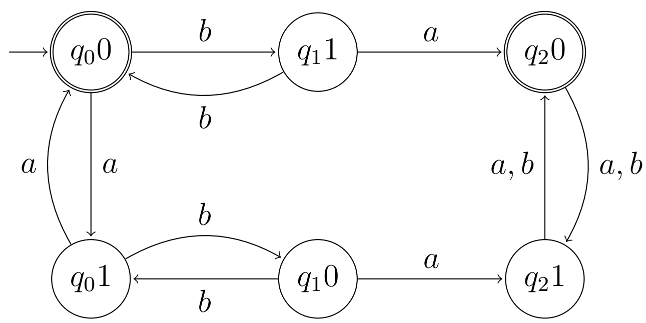
2. Автомат для языка всех слов чётной длины хорошо известен из курса:



Для построения автомата $\overline{\mathcal{B}}$, распознающего дополнение языка $L(\mathcal{B})$ необходимо сделать автомат $\mathcal{B}$ всюду определённым и поменять местами принимающие и непринимающие состояния:



Автомат $\mathcal{C}$, распознающий слова из $L(\overline{\mathcal{B}})$ чётной длины, строим по конструкции произведения:



Праволинейную грамматику построим по алгоритму «КА $\to$ ПГ»: $S = [q_00]$, грамматика задана правилами:

$$[q_00] \to a[q_01] \mid b[q_11] \mid \varepsilon; \quad [q_10] \to b[q_01] \mid a[q_21]; \quad [q_20] \to a[q_21] \mid b[q_21] \mid \varepsilon$$

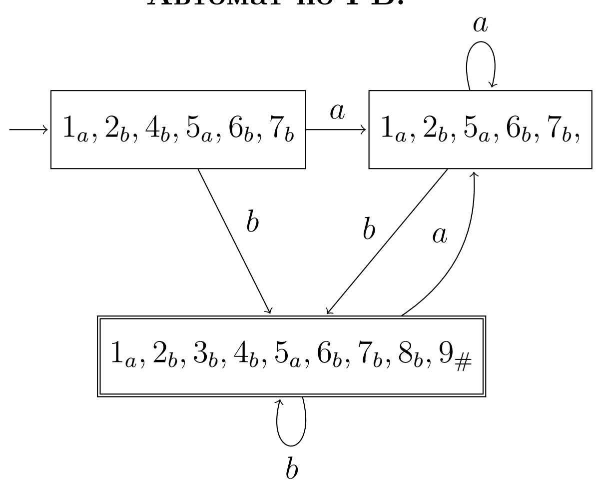
$$[q_01] \to a[q_00] \mid b[q_10]; \quad [q_11] \to b[q_00] \mid a[q_20]; \quad [q_21] \to a[q_20] \mid b[q_20].$$

1 $\langle$ii$\rangle$. **1.** Занумеруем все символы, входящие в регулярное выражение, укажем результаты вычисления функции **followpos** и построем получившийся по алгоритму «РВ $\to$ ДКА» автомат и соответсвующий минимальный автомат.

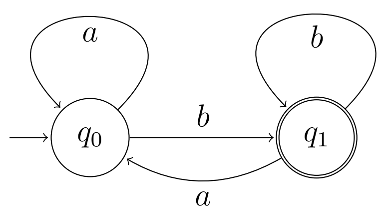
$$(a_1^* b_2 b_3^* | b_4^*)^* a_5^* (b_6 | b_7 b_8) \#_9$$

№	Followpos
1	1,2
2	1,2,3,4,5,6,7
3	1,2,3,4,5,6,7
4	1,2,4,5,6,7
5	5,6,7
6	9
7	8
8	9

Автомат по РВ:



Минимальный автомат:

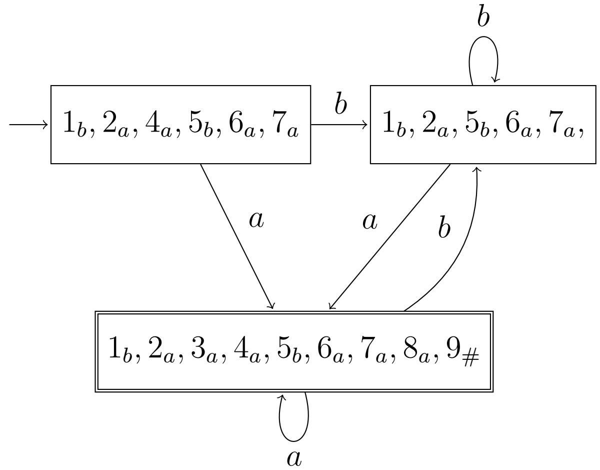


1 ⟨iii⟩. **1.** Занумеруем все символы, входящие в регулярное выражение, укажем результаты вычисления функции **followpos** и построем получившийся по алгоритму «РВ → ДКА» автомат и соответсвующий минимальный автомат.

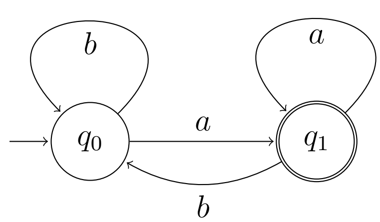
$$(b_1^* a_2 a_3^* | a_4^*)^* b_5^* (a_6 | a_7 a_8) \#_9$$

№	**Followpos**
1	1,2
2	1,2,3,4,5,6,7
3	1,2,3,4,5,6,7
4	1,2,4,5,6,7
5	5,6,7
6	9
7	8
8	9

Автомат по РВ:



Минимальный автомат:

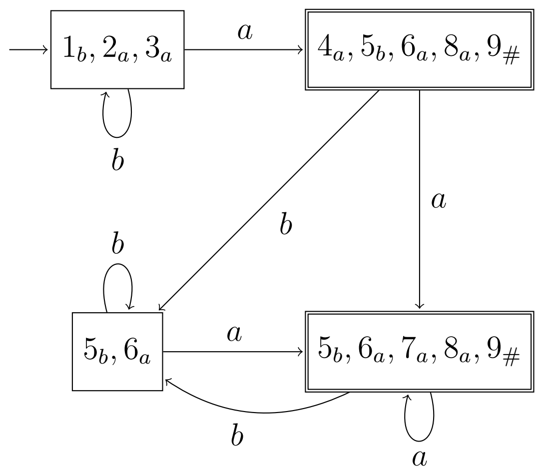


1 $\langle$iv$\rangle$. **1.** Занумеруем все символы, входящие в регулярное выражение, укажем результаты вычисления функции **followpos** и построим получившийся по алгоритму «РВ → ДКА» автомат и соответсвующий минимальный автомат.

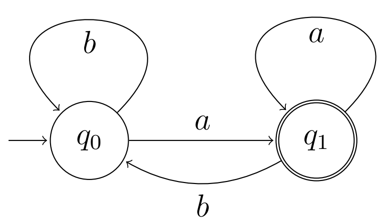
$$b_1^*(a_2|a_3a_4)(b_5^*a_6a_7^*|a_8^*)^*\#_9$$

№	**Followpos**
1	1,2,3
2	5,6,8,9
3	4
4	5,6,8,9
5	5,6
6	5,6,7,8,9
7	5,6,7,8,9
8	5,6,8,9

Автомат по РВ:



Минимальный автомат:



<u>Критерии.</u> стоимость алгоритма (в случае ошибки в решении)

Алгоритм	Σ
РВ $\to$ ДКА	2
Минимизация	1
Дополнение	1
Автомат для слов чётной длины	1
Конструкция произведения	1
ПГ $\to$ ДКА	1

Задача 2 (1+5+1).

2 $\langle$i$\rangle$. Даны языки $L = \{w \mid \exists x, y, z \in \Sigma^* : w = xyz,\ |y| > 0, xy = (xy)^R,\ yz = (yz)^R\}$ и $A = L(b^+a^+b^+a^+)$ над алфавитом $\Sigma = \{a, b\}$.

1. Укажите такое слово $w \in L \cap A$, что $|w| > 5$.

2. Является ли L регулярным языком?

3. Верно ли, что для произвольного регулярного языка R', язык $L \cap R'$ является нерегулярным?

2 $\langle$ii$\rangle$. Даны языки $L = \{w \mid \exists x, y, z \in \Sigma^* : w = xyz,\ |x| = |z|, xy = (yz)^R\}$ и $A = L(a^*ba^*)$ над алфавитом $\Sigma = \{a, b\}$. 1. Укажите такое слово $w \in L \cap A$, что $|w| > 5$.

2. Является ли L регулярным языком?

3. Верно ли, что для произвольного регулярного языка R', язык $L \cup R'$ является нерегулярным?

2 $\langle$iii$\rangle$. Даны языки $L = \{w \mid \exists x, y, z \in \Sigma^* : w = xyz,\ |y| > 0, xy = (xy)^R,\ yz = (yz)^R\}$ и $A = L(a^+b^+a^+b^+)$ над алфавитом $\Sigma = \{a, b\}$. 1. Укажите такое слово $w \in L \cap A$, что $|w| > 5$.

2. Является ли L регулярным языком?

3. Верно ли, что для произвольного регулярного языка R', язык $R' \setminus L$ является нерегулярным?

2 $\langle$iv$\rangle$. Даны языки $L = \{w \mid \exists x, y, z \in \Sigma^* : w = xyz,\ |x| = |z|, (xy)^R = yz\}$ и $A = L(b^*ab^*)$ над алфавитом $\Sigma = \{a, b\}$. 1. Укажите такое слово $w \in L \cap A$, что $|w| > 5$.

2. Является ли L регулярным языком?

3. Верно ли, что для произвольного регулярного языка R', язык $L \setminus R'$ является нерегулярным?

Решение. 2 $\langle$i$\rangle$. 1. $\forall n > 0 \exists x = a^n b^n, y = b^n a^n, z = a^n b^n$ (и слово $xyz \in L \cap R$ искомое)

2. Заметим, что для слов, выбранных в пункте 1, не выполняется лемма о расрастании. Действительно, $\forall n > 0 \exists uwv = a^n b^{2n} a^{2n} b^n, |uwv| > n : uw = a^{n-k}, n > k \geqslant 0$, но при этом для любого разбиения $uw = a^{n-k} : n > k \geqslant 0$ существует i:

$uw^i v = a^{4n+t} b^{2n} a^{2n} b^n$ – не представимо в виде $xyz : |y| > 1, xy = (xy)^R, yz = (yz)^R$

Покажем это явно. Если $x = a^k$, то $y = a^m$ либо $y = b^{2n} a^k$, тогда нарушется условие $yz = (yz)^R$. В случае если $x = a^{4n+t} b^k$, то $y = b^{2n-k} a^{4n+t}$, что также невозможно.

Отсюда получаем, из отрицании леммы о разрастании $\Rightarrow L \notin \mathsf{REG}$

2 $\langle$ii$\rangle$. $x = a^n, y = b, z = a^n$

2 $\langle$iii$\rangle$. $x = b^n a^n, y = a^n b^n, z = b^n a^n$

2 $\langle$iv$\rangle$. $x = b^n, y = a, z = b^n$

Критерии. 1. Если указано слово, но не доказано, что слово лежит в языке L – **0 очков** по пункту 1. преамбулы. 2. Приведена верная последовательность слов и верно использована лемма о накачке, но не доказано, что получившееся слово не принадлежит L (нет анализа всевозможных разбиений) – **1.5 очка**; незаконченное верное доказательство – **3 очка**. Типичный пример: при решении второго варианта сказано, что взяв

$i = 0$ в лемме о накачке получаем $a^{n-k}ba^n \in L$, однако из $|x| = |z|$ и $xy = (yz)^R$ следует, что разбиение невозможно, поскольку слева от b и справа от b разное количество букв a. На самом деле последнее утверждение является наблюдением и требует дополнительного обоснования: например, доказать, что слово y обязано быть палиндромом.

Задача 3 (4).

3 $\langle$i$\rangle$. Постройте для словаря $S = \{aac, acb, b, ac, c\}$ автомат Ахо-Корасик. Посчитайте с его помощью (или с помощью ДКА Ахо-Корасик) количество различных вхождений слов из словаря S в подслово $aacbacb$.

3 $\langle$ii$\rangle$. Постройте для словаря $S = \{abc, ca, bc, a, c\}$ автомат Ахо-Корасик. Посчитайте с его помощью (или с помощью ДКА Ахо-Корасик) количество различных вхождений слов из словаря S в подслово $abcabac$.

3 $\langle$iii$\rangle$. Постройте для словаря $S = \{ccb, cba, a, cb, b\}$ автомат Ахо-Корасик. Посчитайте с его помощью (или с помощью ДКА Ахо-Корасик) количество различных вхождений слов из словаря S в подслово $ccbacba$.

3 $\langle$iv$\rangle$. Постройте для словаря $S = \{cab, bc, ab, c, b\}$ автомат Ахо-Корасик. Посчитайте с его помощью (или с помощью ДКА Ахо-Корасик) количество различных вхождений слов из словаря S в подслово $cabcacb$.

Решение.

3 $\langle$i$\rangle$. На рис. 2 изображён автомат Ахо-Корасик для словаря из варианта $\langle$i$\rangle$.

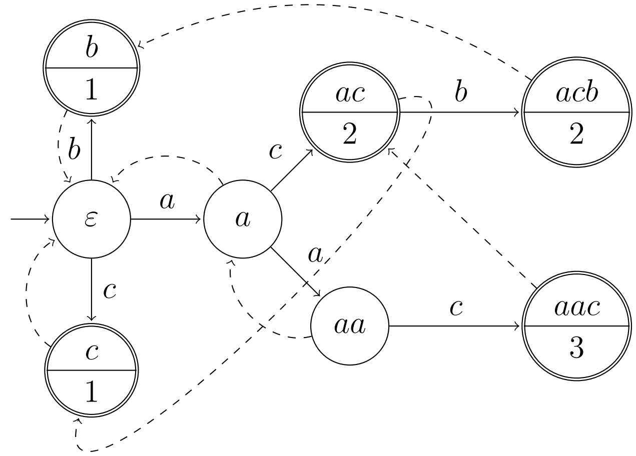


Рис. 1: ДКА $\mathcal{A}$ реализует словарь варианта $\langle \mathrm{i} \rangle$.

состояние	необработанная часть входа	$+k$
ε	$aacbacb$	0
a	$acbacb$	0
aa	$cbacb$	0
aac	$bacb$	+3
ac	$bacb$	0!
acb	acb	+2
b	acb	0!
ε	acb	0!
a	cb	0
ac	b	+2
acb	ε	+2

Запись «0!» означает, что счётчик не увеличился, поскольку переход был по суффиксной ссылке, а это значит, что вклад состояния v (в которое был переход) был учтён при попадании в состояние uv по прямому переходу. Суммируя все увеличения счётчика, получаем **численный ответ:** 9.

3 $\langle$ii$\rangle$.

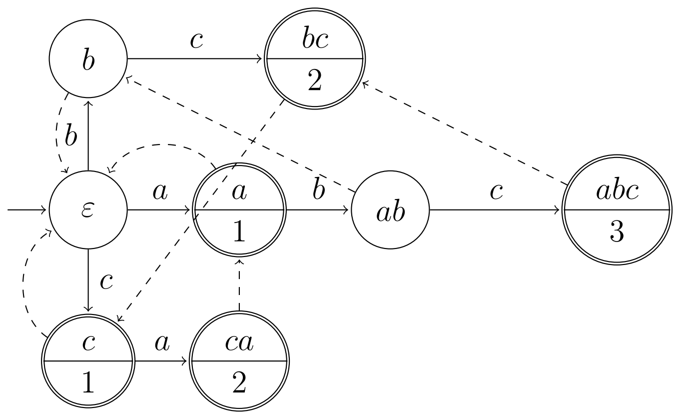


Рис. 2: ДКА $\mathcal{A}$ реализует словарь варианта $\langle$ii$\rangle$.

Численный ответ: 8.

3 $\langle$iii$\rangle$. Такой же граф как и в $\langle$i$\rangle$, **численный ответ:** 9.

3 $\langle$iv$\rangle$. такой же граф как и в $\langle$ii$\rangle$, **численный ответ:** 8.

Критерии. Построен автомат Ахо-Корасик (ДКА Ахо-Корасик): +**2** очка. Одна ошибка при демонстрации работы: **-0.5** очка. Ошибка при подсчёте кратности суффиксов в автомате Ахо-Корасик: **-1** очко.

Задача 4(2+4). Пусть $\Sigma = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$. Введём функцию $go\colon \Sigma^* \to (\mathbb{Z}\times\mathbb{Z})$, соответствующую операции перемещения в двумерной целочисленной плоскости из точки $(0,0)$ в финальную точку маршрута, заданного последовательностью стрелок (словом в алфавите Σ). Так, $go(\rightarrow\uparrow\rightarrow\uparrow\rightarrow) = (3,2)$, $go(\leftarrow\uparrow\rightarrow\downarrow) = (0,0)$.

Задача 4 $\langle$i$\rangle$. Язык L состоит из слов-путей w из точки $(0,0)$ в $(6,5)$ (формально $go(w) = (6,5)$), не выходящих за пределы квадрата $[0,10]\times[0,10]$ и не пересекающих прямую $y = x+1$.

1. Является ли язык L регулярным?

2. Опишите классы эквивалентности Майхилла-Нероуда для языка L. (Факт того, что описанные классы являются классами Майхилла-Нероуда для языка L, требует доказательства!)

Задача 4 $\langle$ii$\rangle$. Язык L состоит из слов-путей w из точки $(0,0)$ в $(7,3)$ (формально $go(w) = (7,3)$), не выходящих за пределы квадрата $[0,10]\times[0,10]$ и не пересекающих прямую $y = 11-x$.

1. Является ли язык L регулярным?

2. Опишите классы эквивалентности Майхилла-Нероуда для языка L. (Факт того, что описанные классы являются классами Майхилла-Нероуда для языка L, требует доказательства!)

Задача 4 $\langle$iii$\rangle$. Язык L состоит из слов-путей w из точки $(0,0)$ в $(6,7)$ (формально $go(w) = (6,7)$), не выходящих за пределы квадрата $[0,10]\times[0,10]$ и не пересекающих прямую $y = 2+x$.

1. Является ли язык L регулярным?

2. Опишите классы эквивалентности Майхилла-Нероуда для языка L. (Факт того, что описанные классы являются классами Майхилла-Нероуда для языка L, требует доказательства!)

Задача 4 $\langle$iv$\rangle$. Язык L состоит из слов-путей w из точки $(0,0)$ в $(8,2)$ (формально $go(w) = (8,2)$), не выходящих за пределы квадрата $[0,10]\times[0,10]$ и не пересекающих прямую $y = 12-x$.

1. Является ли язык L регулярным?

2. Опишите классы эквивалентности Майхилла-Нероуда для языка L. (Факт того, что описанные классы являются классами Майхилла-Нероуда для языка L, требует доказательства!)

Решение.

1. Язык является регулярным, поскольку существует распознающий его конечный автомат. Состояния автомата – пары целых чисел (x, y) из квадрата $[0, 10] \times [0, 10]$ ниже прямой и тупиковое состояние. Обрабатывая стрелку из состояния (x, y), автомат переходит в соответсвующее состояние $(x \pm 1, y)$ или $(x, y \pm 1)$, при условии что переход по стрелке не выводит за пределы квадрата или при переходе не происходит пересечение указанной прямой, иначе автомат переходит в тупиковое состояние. Начальное состояние автомата – пара $(0, 0)$, а единственное принимающее – целевая точка (x_f, y_f). По определению функции переходов ясно, что описанный автомат переходит по слову w из начального состояния $(0, 0)$ в целевую точку (x_f, y_f) тогда и только тогда, когда $go(w) = (x_f, y_f)$ и путь не покидал пределы квадрата и не пересекал прямую.

2. Решение пункта 1 можно продолжить до решения всей задачи, доказав, что построенный автомат минимальный и описав его классы. Мы приведём здесь другое решение – через явное описание классов эквивалентности Майхилла-Нероуда; из этого решения в свою очередь следует решение пункта 1.

Обозначим множество точек, для которых выполняется ограничение на слово-путь w из условия

$S = \{(x, y) | y < x + 1, (x, y) \in [0, 10] \times [0, 10]\}$

Введём следующие множества путей:

$C_{(x,y)} = \{w | go(w) = (x, y) \in S \wedge \forall i \rightarrow go(w[0 : i]) \in S\}$

$C_{NO} = \{w | \exists i : go(w[0 : i]) \notin S\}$

Заметим, что $\forall w \in \Sigma^*$ найдётся множество C из описанных, такое что $w \in C$ Действительно, если для w не выполняются ограничения на слова-пути (т.е. $\exists i : go(w[0 : i]) \notin S$), то оно автоматически попадает в C_{NO}. В противном случае w будет содержаться в $C_{(x,y)}$, где $go(w) = (x, y)$ по определению введённого семейства $C_{(x,y)}$

Покажем, что введённые множества путей не пересекаются.

$\forall (x, y) \in S \rightarrow C_{(x,y)} \cap C_{NO} = \emptyset$, т.к. w не может одновременно удовлетворять и не удовлетворять ограничениям на слова-пути.

Также $\forall (x_1, y_1), (x_2, y_2 \in S, (x_1, y_1) \neq (x_2, y_2) \rightarrow C_{(x_1,y_1)} \cap C_{(x_2,y_2)} = \emptyset$, иначе в противном случае $\exists w : go(w) = (x_1, y_1) \wedge go(w) = (x_2, y_2)$, что невозможно, т.к. go определена однозначно.

Докажем, что введённые множества $\{C_i\}$ есть классы эквивалентности для языка L. Выше мы получили, что множества покрывают Σ^* и не пересекаются.

1. Покажем, что $\forall w_1, w_2 \in C_{(x,y)} \to w_1 \sim_L w_2$

   Заметим, что прибавление некоторого суффикса $z \in \Sigma^*$ к $w_1, w_2 \in C_{(x,y)}$ означает продолжение пути из точки (x, y), и здесь возможно несколько случаев.

   Без ограничения общности рассмотрим w_1

   (a) $w_1 z \in L \Rightarrow$ путь $w_1 z$ удовлетворяет ограничениям на пути-слова, и при этом $go(w_1 z) = (5, 5)$ Тогда $go(w_2 z) = (5, 5)$, тоже, так как $go(w_1) = go(w_2)$, и для обоих путей ограничения выполняются. Получаем что $\forall z : w_1 z \in L \Rightarrow w_2 z \in L$

   (b) $w_1 z \notin L \Rightarrow$ путь $w_1 z$ не удовлетворяет ограничениям на пути-слова или $go(w_1 z) \neq (5, 5)$.

   Заметим, что изначально для w_1, w_2 ограничения выполнялись, то есть нарушение ограничений могло возникнуть лишь вследствие добавление продолжения пути как суффикса z, который в случае w_1, w_2 стартует и одной и той же точки $go(w_1) = go(w_2) = (x, y)$

   Отсюда, если путь $w_1 z$ не удовлетворяет ограничениям на пути-слова, то для w_2 они так же не выполняются.

   В случае $go(w_1 z) \neq (5, 5)$ заметим, что добавление приводит к детерминированном у продолжению пути (в данном случае из одной и той же точки $go(w_1) = go(w_2) = (x, y)$) т.е. $go(w_1 z) = go(w_2 z) = (x, y) + go(z) \neq (5, 5)$

   Отсюда, если $go(w_1 z) \neq (5, 5)$, то $go(w_2 z) \neq (5, 5)$

   Получаем что $\forall z : w_1 z \notin L \Rightarrow w_2 z \notin L$

   Так как мы брали произвольные $w_1, w_2 \in C_{(x,y)}$, то все полученные утверждения справедливы и в обратную сторону. Из определения эквивалентности получаем, что $\forall w_1, w_2 \in C_{(x,y)} \to w_1 \sim_L w_2$

2. Покажем, что $\forall w_1, w_2 \in C_{NO} \to w_1 \sim_L w_2$

   В данном случае эквивалентность выполняется автоматически и $\forall z \in \Sigma^* \to w_1 z \notin L, w_2 z \notin L$, так как для w_1, w_2 в любом случае не выполняются ограничения на пути-слова.

3. Покажем, что $\forall w_1 \in C_{(x_1,y_1)}, \forall w_2 \in C_{(x_2,y_2)} : (x_1, y_1) \neq (x_2, y_2) \to w_1 \not\sim_L w_2$

   Для этого заметим, что в этом случае $go(w_1) = (x_1, y_1) \neq go(w_2) = (x_2, y_2)$ Так как добавление суффикса z к w_1 означает детерминированное продолжение пути из (x_1, y_1), то $\forall z : go(w_1 z) = (5, 5) \wedge \forall i \to go(w_1 z[0 : i]) \in S \longrightarrow go(w_2 z) \neq (5, 5)$.

   То есть $\forall z : w_1 z \in L \Rightarrow w_2 z \notin L$, множества неэквивалентны.

4. Покажем, что $\forall w_1 \in C_{(x_1,y_1)}, \forall w_2 \in C_{NO} : (x_1, y_1) \neq (x_2, y_2) \to w_1 \not\sim_L w_2$

   Заметим, что с одной стороны $\forall z \to w_2 z \notin L$, так как для такого пути не будут выполняться ограничения, а с другой $\exists z : w_1 z \in L$

   Отсюда $\exists z : w_1 z \in L \wedge w_2 z \notin L$, множества неэквивалентны.

Таким образом, просуммировав всё вышесказанное можем сказать, что Σ^* было разбито на классы эквивалентности Майхилла-Нероуда по L, которые попарно неэквивалентны, причём покрывают Σ^* целиком, т.е. набор классов полон.

Заметим, что $|\{C_{(x,y)}\}| = |S| \leqslant 10 * 10$. Отсюда по теореме Майхилла-Нероуда получаем, что L – регулярен, а число состояний в минимальном ДКА, распознающем его есть $|S| + 1$

Критерии. В случае корректного описаний множеств слов, которые либо являются классами Майхилла-Нероуда, либо соответствуют состояниям неминимального автомата и обоснования конечности $+\mathbf{2}$ очка за пункт 1. При этом за доказательство выполнения каждого из следующих условий по $+\mathbf{1}$ очку:

- доказана полнота
- доказана эквивалентность внутри класса
- доказана попарная неэквивалентность

Задача 5 (2+3). Язык $\mathrm{PreSuf}(u)$ состоит из слов над алфавитом $\Sigma = \{a, b\}$, которые имеют и префикс u, и суффикс u.

1. Докажите, что для любого слова $u \in \Sigma^*$ язык $\mathrm{PreSuf}(u)$ является регулярным.

5 $\langle$i$\rangle$. 2. Запишите регулярное выражение для языка $\mathrm{PreSuf}(a^2b^3a^2)$.

5 $\langle$ii$\rangle$. 2. Запишите регулярное выражение для языка $\mathrm{PreSuf}(aba^3b)$.

5 $\langle$iii$\rangle$. 2. Запишите регулярное выражение для языка $\mathrm{PreSuf}(ab^2a^3b^2)$.

5 $\langle$iv$\rangle$. 2. Запишите регулярное выражение для языка $\mathrm{PreSuf}((ab)^4)$.

Решение. **5** $\langle$i$\rangle$. 1. Язык всех слов, имеющих префикс, равный w, ре-

гулярен: его можно задать регулярным выражением $w\Sigma^*$. Аналогично язык всех слов, имеющих суффикс, равный w, регулярен: его можно задать регулярным выражением $\Sigma^* w$. Тогда $\mathrm{PreSuf} = w\Sigma^* \cap \Sigma^* w$ и, следовательно, регулярен.

2. **Ответ:** $a^2b^3a^2\Sigma^*a^2b^3a^2 + a^2b^3a^3b^3a^2 + a^2b^3a^2b^3a^2 + a^2b^3a^2$.

Нужно быть осторожным: слово $a^2b^3a^2$ имеет нетривиальный перехлест с самим собой. Поэтому аккуратно рассмотрим, какие слова могут иметь суффикс и префикс, равный $a^2b^3a^2$:

- само слово $a^2b^3a^2$,
- слово $a^2b^3a^2b^3a^2$, в котором искомые префикс и суффикс пересекаются по a^2,
- слово $a^2b^3a^3b^3a^2$, в котором искомые префикс и суффикс пересекаются по a,
- слово $a^2b^3a^2wa^2b^3a^2$ для любого $w \in \Sigma^*$, в котором искомые префикс и суффикс не пересекаются.

Других слов мы не найдём, потому что в слове $u = a^2b^3a^2$ есть только две группы букв a. Если приложить к первой букве a первой группы слово u, мы получим само слово u, если приложить ко второй a первой группы – наложения не получится, а для каждого наложения слова u к буквам a из второй группы мы привели подходящее слово. Суммируем все это и получаем ответ.

5 $\langle$ii$\rangle$. **Ответ:** $aba^3b\Sigma^*aba^3b + aba^3ba^3b + aba^3b$.

5 $\langle$iii$\rangle$. **Ответ:** $ab^2a^3b^2\Sigma^*ab^2a^3b^2 + ab^2a^3b^2a^3b^2 + ab^2a^3b^2$

5 $\langle$iv$\rangle$. **Ответ:** $(ab)^4\Sigma^*(ab)^4 + (ab)^7 + (ab)^6 + (ab)^5 + (ab)^4$.

Критерии. 2. Если сказано, что язык $\mathrm{PreSuf}(u)$ состоит из языка $u\Sigma^*u$ и конечного числа перехлёстов u с собой, после чего выписано правильное РВ: **2 очка**. Забыт один перехлёст: **1.5 очка**.

Мини-задачи и вопросы

Общие критерии. В случае положительного ответа нужно привести доказательство, а в случае отрицательного – контрпример. Верное рассуждение без контрпримера оценивается в половину задачи (см. преамбулу). В задачах, оцениваемых в 3 балла, третий балл ставится, если в доказательстве не было огрехов.

Задача 6 (3).

6 $\langle$i$\rangle$. Является ли язык чисел в десятичной записи $L = \{w \mid w \equiv 0 \mod 3\}$ регулярным?

6 $\langle$ii$\rangle$. Является ли язык чисел в двоичной записи $L = \{w \mid w = 0 \mod 3\}$ регулярным?

6 $\langle$iii$\rangle$. Является ли язык чисел в десятичной записи $L = \{w \mid w = 0 \mod 9\}$ регулярным?

6 $\langle$iv$\rangle$. Является ли язык чисел в десятичной записи $L = \{w \mid w = 0 \mod 8, |w| \geqslant 3\}$ регулярным?

Решение. 6 $\langle$i$\rangle$. Да, данный язык будет регулярным, так как множество

всех слов из языка чисел в десятичной записи можно разбить на классы эквивалентности по L: $\Sigma^*\backslash_{\sim_L} = \{C_0, C_1, C_2\}$, где $C_k = \{x | x = k \mod 3\}$

Покажем, что $\forall w_1, w_2 \in C_k, \forall z \in \Sigma^* \to w_1z \in L \iff w_2z \in L$ Без ограничения общности пусть $w_1 = k \mod 3, z = m \mod 3$. Тогда по признаку делимости на 3 (признак Паскаля) получаем что $w_1z = m + k \mod 3$ и $w_2z = m + k \mod 3$ (так как $w_2 \in C_k$, а значит $w_2 = k \mod 3$) Отсюда $\forall z \in \Sigma^* \to w_1z \in L \Longleftrightarrow w_2z \in L$

Докажем, что данные классы эквивалентности попарно неэквивалентны. Действительно $\forall w_1 \neq w_2, \forall z \in \Sigma^* : z = m \mod 3$ будет выполняться $w_1z = m + k_1 \mod 3, w_2z = m + k_2 \mod 3, \; k_1 \neq k_2$, т.е. найдётся $z : w_1z \in L, w_2z \notin L$

Таким образом получили полную конечную систему попарно неэквивалентных классов эквивалентности. По теореме Майхилла-Нероуда L регулярен.

6 $\langle$ii$\rangle$. данном случае задача решается также через построение системы классов эквивалентности Майхилла-Нероуда $\{C_0, C_1, C_2\}$, куда входят слова, являющиеся числами, имеющими соответствующий остаток по модулю 3.

Ключевым отличием в данном случае выляется то что нельзя просто суммировать остаток от деления на 3 суммы цифр. Заметим, в результате дописывания к числу u суффикса z длины k получаем число $u * 2^k + z$.

Для доказательством эквивалентности внутри класса воспользуемся тем что $\forall u, v \in C_i : u = 3n + a, v = 3m + a \forall z$ выполняется $(3n + a) * 2^k + z \mod 3 = a * 2^k + z \mod 3$, и также $(3m + a) * 2^k + z \mod 3 = a * 2^k + z \mod 3$, т.е. $\forall z \in \Sigma^* \to uz \in L \Longleftrightarrow vz \in L$

Для доказательства попарной неэквивалентности заметим что $\forall a, b \in 0, 1, 2 : a \neq b \to a * 2^k \neq b * 2^k$ Тогда $\forall u, v \in C_i : u = 3n + a, v = 3m + b : a \neq b \forall z$ выполняется $(3n + a) * 2^k + z \mod 3 \neq (3n + b) * 2^k + z \mod 3$, т.е. найдётся $z : uz \in L, vz \notin L$

Таким образом получили полную конечную систему попарно неэквивалентных классов эквивалентности Майхилла-Нероуда по L. По теореме Майхилла-Нероуда L регулярен.

6 $\langle$iii$\rangle$. Задача сводится к использованию признака остатка от деления на 9 и решается аналогично $\langle 1 \rangle$, с той лишь разницей, что в процесс получается девять классов эквивалентности Майхилла-Нероуда.

6 $\langle$iv$\rangle$. В данном случае возможно задачу проще решать через построение КА для исходного языка и доказательства его корректности.

Заметим, что число делится на 8 если число, составленное из трёх его последних цифр делятся на 8 (для $w \in L : |w| \geqslant 3$). Так как множество таких числе L_8 конечно – можно построить КА, распознающий его.

Далее построим тривиальный автомат с единственным состоянием, который принимает любое слово и соединяем его ε-переходом с начальным состоянием автомата для языка L_8. начальным состоянием в данном случае будет единственное состояние тривиального атомата, а конечным состоянием – принимающее состояние автомата, распознающего L_8.

Получаем автомат $\mathcal{A} : L(\mathcal{A}) = \{yx \mid |x| = 3, x = 0 \mod 8\}$ (по построению). Тогда по признаку делимости на 8 $L(\mathcal{A}) = \{w \mid w, x = 0 \mod 8, |w| \geqslant 3\}$

Таким образом мы построили автомат для L и доказали его корректность $\Rightarrow L$ – регулярен.

Критерии. При проверке оценка не снижалась, в случае если в решении считалось, что пустое слово эквивалентно числу длины ноль. Также допускались обе трактовки: когда записи чисел могут иметь ведущие нули и когда нет. В случае, если построен верный автомат с попыткой обоснования **+1 очко**.

Задача 7 (3). Пусть $L \in \mathsf{REG}$. Верно ли, что язык L' является регулярным?

7 ⟨i⟩.

$$L' = \{w \in \Sigma^* \mid w = yx, xy \in L, \ |x| = 2\}$$

7 ⟨ii⟩.

$$L' = \{w \in \Sigma^* \mid w = y^R x, yx \in L, |x| = 1\}$$

7 ⟨iii⟩.

$$L' = \{w \in \Sigma^* \mid w = zyx, xyz \in L, |x| = |z| = 1\}$$

7 ⟨iv⟩.

$$L' = \{w \in \Sigma^* \mid w = zx, xyz \in L, |x| = |y| = 2\}$$

Решение.

7 ⟨i⟩. Так как язык L регулярен, то существует ДКА $\mathcal{A}$ распознающий его.

Произведём необходимые манипуляции над ним:

Найдём все вершины, которые достижимы из q_0 – начального состояния исходного автомата – по двум символам, обозначим полученное множество Q_m.

Добавим новое начальное состояние q_0' и ε-переходы из него во все вершины Q_m

Выделим все возможные пути по двум символам из q_0, получим конечное множество слов L_2. $L_2 = \{x \mid xy \in L, |x| = 2\}$

Построим КА для языка L_2, обозначив его начальное и конечное состояние как q_2 и q_f' соответственно.

Добавим ε-переходы из всех принимающих состояний исходного автомата в q_2.

В полученном преобразованном автомате обозначим q_0' – единственное начальное состояние, q_f' – единственное конечное состояние.

Докажем корректность построенного автомата $\mathcal{A}'$ для L'

$L' \subseteq L(\mathcal{A}')$

$\forall w \in L'$ слово w представимо в виде $yx : xy \in L, |x| = 2$ по определению L' Тогда существует $q_m \in Q_m : \delta(q_0, x) = q_m$ (из введения обозначений множества состояний Q_m) Также $x \in L_2$ (из определения L_2)

Так как $xy \in L$, то в исходном автомате $\delta(q_m, y) = q_f$, где q_f – одно из принимающих состояний исходного автомата $\mathcal{A}$

Отсюда для построенного автомата $\mathcal{A}'$ существует путь из q_0' в q_m по ε (по построению), из q_m в q_f по y (из исходного автомата), из q_f в q_2 по ε (по построению), и далее по x в q_f' (по построению автомата для L_2).

Таким образом $\forall w \in L' : w = yx, xy \in L, |x| = 2$ в построенном автомате $q_f' \in \delta(q_0', w)$, т.е. слово принимается автоматом $\Rightarrow L' \subseteq L(\mathcal{A}')$

$L(\mathcal{A}') \subseteq L'$

Рассмотрим слово $w \in L(\mathcal{A}')$. Из построения автомата $\mathcal{A}'$ это означает, что $w = yx : |x| \in L_2$, причём $\delta(q_0, x) = q_m \in Q_m$ в исходном автомате, а также $\exists q_2 \in \delta(q_0', y)$.

Тогда в исходном автомате $\mathcal{A}$ существует q_f – одно из финальных состояний: $\delta(q_m, y) = q_f$ (по построению)

Получили, что $\forall w \in L(\mathcal{A}') w = yx : |x| \in L_2$, причём для исходного автомата $\delta(q_0, xy) = q_f, ..xy \in L$. Отсюда $yx \in L' \Rightarrow L(\mathcal{A}') \subseteq L'$

Из равенства $L(\mathcal{A}') = L'$ следует корректность построенного автомата. Отсюда искомый язык L' регулярен, т.к. существует КА $\mathcal{A}'$,распознающий его.

7 $\langle$ii$\rangle$. Принцип доказательства через построение автомата аналогичен варианту $\langle$i$\rangle$.

В данном случае необходимо найти все состояния $\{q_m\}$ в исходном автомате $\mathcal{A}$ по которым можно попась по одному символу в $q_f \in F$ (F – множество принимающих состояний исходного автомата), сделать теперь уже данное множество $\{q_m\}$ принимающими состояниями и обернуть полученный автомат по стандартному алгоритму. Затем к полученному финальному состоянию q_0 (принимающее в исходном автомате $\mathcal{A}$) добавить переходы в новое финальное состояние q'_f по $a \in \Sigma : \delta(q_m, a) = q_f \in F\mathcal{A}$. Тогда добавив новое начальное состояние q'_0 и соединив его ε-переходами с $\{q_f\}$ получим автомат, распознающий L'

Подход к доказательству корректности аналогичен варианту $\langle$i$\rangle$.

7 $\langle$iii$\rangle$. Принцип доказательства через построение автомата аналогичен варианту $\langle$i$\rangle$.

В данном случае необходимо найти все состояния $\{q_z\}$ в исходном автомате $\mathcal{A}$ по которым можно попась по одному символу в $q_f \in F$ (F – множество принимающих состояний исходного автомата), выделив все символы в конечный язык однобуквенных слов L_z. Аналогично $\{q_x\}$ – состояния, в которые можно попасть по одному символу из q_0, причём F достижимо из q_x, множество таких символов образует L_x (условие достижимости в данном случае существенно).

Построив тривиальные автоматы для L_x, L_z соединим конечное состояние автомата для языка L_z ε-переходами с $\{q_x\}$ исходного автомата, а $\{q_z\}$ в исходном автомате соединим ε-переходами с начальным состоянием автомата для языка L_x.

В полученном автомате начальным состоянием станет начальное состояние автомата для языка L_z, а конечным – конечное состояние для языка L_x

Подход к доказательству корректности аналогичен варианту $\langle$i$\rangle$.

7 $\langle$iv$\rangle$. Принцип доказательства через построение автомата аналогичен варианту $\langle$i$\rangle$.

В данном случае необходимо найти все состояния $\{q_z\}$, достижимые за 4 перехода из q_0 – начального состояния. Также необходимо найти все состояния $\{q_x\}$, в которые можно попасть за 2 перехода из q_0, такие что F достижимо для каждого из них. Для таких двубуквенных переходов обозначим $L_x = \{x : |x| = 2, \exists xy \in L\}$. Достижимость F из $\{q_x\}$ в данном случае существенна для корректности строящегося автомата.

Построив тривиальный автомат для L_x соединим финальные состояния исходного автомата ε-переходами с начальным состоянием автомата для языка L_x. Единственным принимающим состоянием построенного автомата будет принимающее состояние для языка L_x Тогда добавив новое начальное состояние q_0' и соединив его ε-переходами с $\{q_z\}$ получим автомат, распознающий L'

Подход к доказательству корректности аналогичен варианту ⟨i⟩.

Критерии. Задача внезапно оказалась трудной для студентов. Поэтому решено в случае корректного построения промежуточного автомата (или грамматики), который можно было бы использовать дальше при правильном решении, ставить **1** очко.

Задача 8 (3). Пусть $\mathcal{A}$ и $\mathcal{B}$ минимальные всюду определённые ДКА с n и m состояниями соответственно над алфавитом $\Sigma = \{a, b\}$.

8 ⟨i⟩. Верно ли, что минимальный всюду определённый ДКА для языка $L(\mathcal{A}) \cap L(\mathcal{B})$ имеет не более $n + m$ состояний?

8 ⟨ii⟩. Верно ли, что минимальный всюду определённый ДКА для языка $L(\mathcal{A}) \setminus L(\mathcal{B})$ имеет не более $n - m$ состояний?

8 ⟨iii⟩. Верно ли, что минимальный всюду определённый ДКА для языка $L(\mathcal{A}) \cup L(\mathcal{B})$ имеет не более $n + m$ состояний?

8 ⟨iv⟩. Верно ли, что минимальный всюду определённый ДКА для языка $L(\mathcal{A}) \triangle L(\mathcal{B}) = L(\mathcal{A}) \setminus L(\mathcal{B}) \cup L(\mathcal{B}) \setminus L(\mathcal{A})$ имеет не более n/m состояний?

Решение. Во всех вариантах **ответ:** «неверно». В качестве контрпримера для варианта $\langle i \rangle$ возьмём минимальные автоматы, распознающие языки из слов чётной длины и слов длины, кратной трём, соответсвенно. Тогда автомат $\mathcal{A}$ имеет два состояния, автомат $\mathcal{B}$ имеет три состояния, а язык $L(\mathcal{A}) \cap L(\mathcal{B})$ состоит из слов длины кратной шести и минимальный автомат для него содержит 6 состояний, а не 5. В варианте 3 в качестве автоматов $\mathcal{A}$ и $\mathcal{B}$ возьмём автоматы распознающие языки слов нечётной длины и слов, длины не кратной трём – в них будет также 2 и 3 состояния, как и в случае варианта $\langle i \rangle$, а язык $L(\mathcal{A}) \cup L(\mathcal{B})$ будет языком из слов длины, не кратной шести, и в нём также будет 6 состояний (минимальный автомат, распознающий дополнение к языку L, содержат такие же состояния, как и минимальный автомат, распознающий L).

Для вариантов $\langle ii\rangle$ и $\langle iv\rangle$ в качестве автомата $\mathcal{A}$ возьмём автомат, распознающий все слова, а в качестве автомата $\mathcal{B}$ – автомат, распознающий все слова чётной длины (в нём два состояния). Тогда $L(\mathcal{A}) \setminus L(\mathcal{B}) = L(\mathcal{A}) \triangle L(\mathcal{B}) = \Sigma^* \setminus L(\mathcal{B})$, а минимальный автомат, распознающий $\Sigma^* \setminus L(\mathcal{B})$ очевидно имеет два состояния (а предполагалось, что имеет либо одно, либо не больше половины состояния, что бы это ни значило).

Критерии. Если доказана верхняя оценка $m \times n$ на число состояний автомата (например, явно выписана конструкция произведения, после чего указана эта оценка), то $+\mathbf{1}$. Если упомянута – $+\mathbf{0.5}$.

Задача 9 (2). **9** $\langle$i$\rangle$. Существует ли нерегулярный язык $X \subseteq \{a, b\}^*$, такой что $a^*X = Xb^*$?

9 $\langle$ii$\rangle$. Существует ли нерегулярный язык $X \subseteq \{a, b\}^*$, такой что $a^*Xb^* = a^*b^*$?

9 $\langle$iii$\rangle$. Существует ли нерегулярный язык $X \subseteq \{a, b\}^*$, такой что $X = aXa \mid bXb \mid \varepsilon$?

9 $\langle$iv$\rangle$. Существует ли нерегулярный язык $X \subseteq \{a, b\}^*$, такой что $a^*X = X$?

Решение. **9** $\langle$i$\rangle$. **Ответ:** существует. Возьмем $X = \{a^x b^y \mid x \neq y\}$. Во-первых, X нерегулярен. Это можно доказать разными способами, но самый простой заключается в том, что все слова a^k не эквивалентны по Майхилл-Нероду: для $m \neq n$ имеем $a^m b^n \in X$, но $a^n b^n \notin X$. Во-вторых, $a^*X = Xb^* = a^*b^* \setminus \{\varepsilon\}$: любое слово $a^x b^y$ можно получить либо как $\varepsilon \cdot a^x b^y$ или $a^x b^y \cdot \varepsilon$ для $a^x b^y \in X$, если $x \neq y$, либо как $a \cdot a^{x-1} b^x$ или $a^x b^{x-1} \cdot b$, если $x = y$; при этом все слова имеют длину хотя бы 1 и $a^*X = Xb^* \subset a^*b^*$, поэтому других слов не получится.

9 $\langle$ii$\rangle$. **Ответ:** существует. Возьмем $X = \{a^n b^n \mid n \geqslant\}$. Во-первых, X нерегулярен. Во-вторых, $a^*Xb^* = a^*b^*$: любое слово $a^x b^y$ можно получить как $a^x \cdot \varepsilon \cdot b^y$, а $a^*Xb^* \subseteq a^*b^*$.

9 $\langle$iii$\rangle$. **Ответ:** существует. Возьмём в качестве X язык палиндромов чётной длины.

9 $\langle$iv$\rangle$. **Ответ:** существует. Возьмем $X = \{a^x b^y \mid x > y\}$. Во-первых, X нерегулярен. Во-вторых, $a^* X = X$: любое слово $a^x b^y$ можно получить либо как $\varepsilon \cdot a^x b^y$ для $a^x b^y \in X$, а других слов не получится, ведь $a^* X \subseteq X$.

Задача 10 (2).

10 $\langle$i$\rangle$. Верно ли, что существует регулярный язык R, такой что каждый распознающий его всюду определённый ДКА имеет не меньше 2017 состояний?

10 $\langle$ii$\rangle$. Верно ли, что не существует регулярного языка R, такого что минимальный всюду определённый ДКА, распознающий R, имеет ровно 2017 состояний?

10 $\langle$iii$\rangle$. Верно ли, что для любого регулярного языка R, существует распознающий его всюду определённый ДКА, который имеет не более чем 2017 состояний?

10 $\langle$iv$\rangle$. Верно ли, что для любого регулярного языка R, не существует всюду определённого ДКА, распознающего R, который имеет более чем 2017 состояний?

Решение. В варианте $\langle$i$\rangle$ **ответ:** «да», в остальных вариантах **ответ:** «нет». В качестве языка R для варианта $\langle$i$\rangle$ достаточно взять язык из одного слова длины 2015 – во всюду определённом минимальном автомате будет 2017 состояний. Тот же язык (a^{2015}) служит контрпримером для варианта $\langle$ii$\rangle$. Для вариантов $\langle$iii-iv$\rangle$ подходит язык a^{2017}.