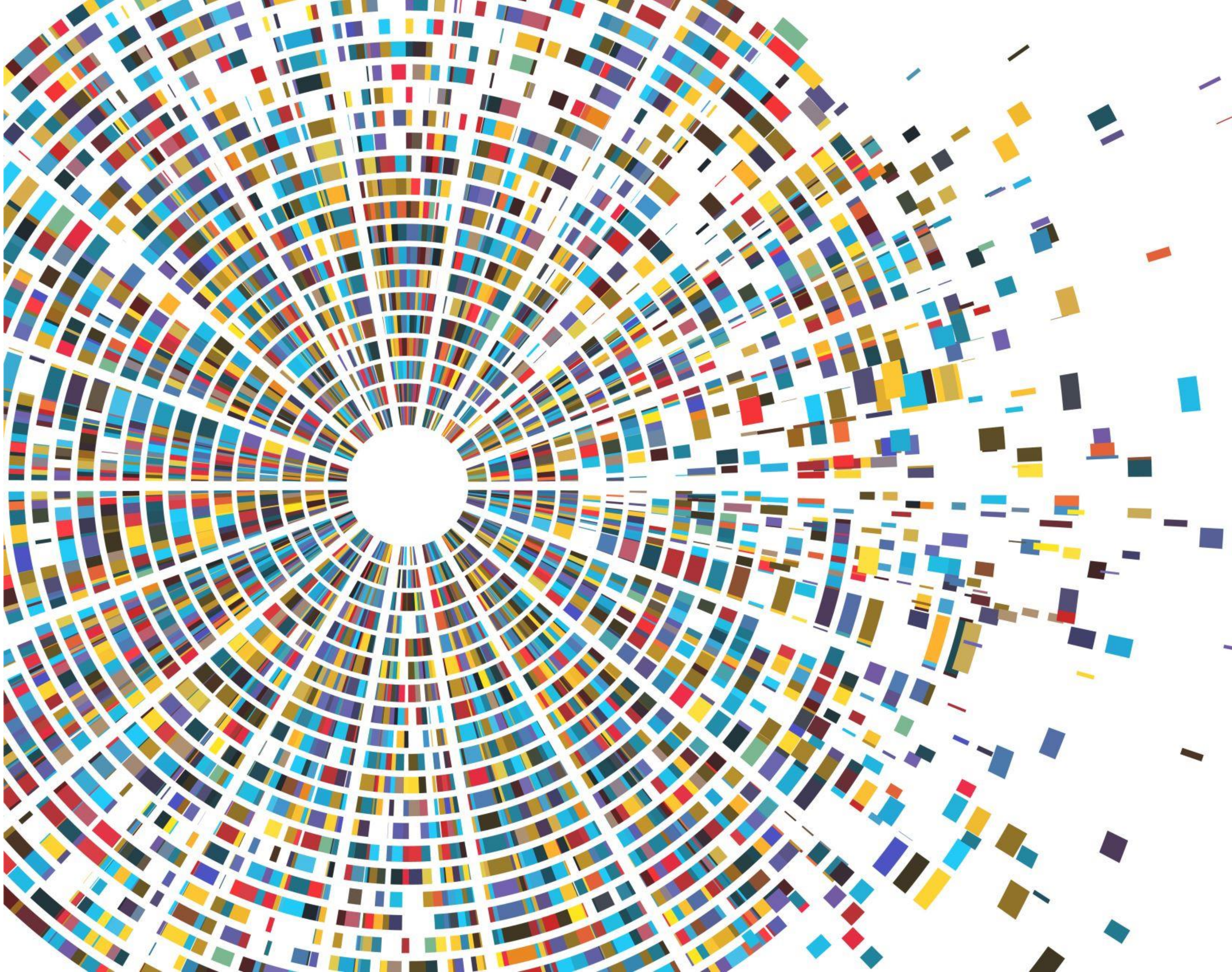
ТЕОРИЯ АВТОМАТОВ
Решенный тренировочный вариант контрольной работы
Задача 8



Направл.: Математика и компьютерные науки

к.ф.-м.н., доцент

Нагребецкая Ю.В.

Участвовали в проекте ст-ты МЕН-290201

Нечуговских А.А, Старков Э.Е.

ЗАДАЧА (вспомогательная)

Построить стандартный λ-НКА , распознающий язык, заданный регулярным выражением. Построить соответствующий приведенный ДКА.

Напомним, как строится автомат по данному регулярному выражению.

Пусть $\boldsymbol{A_1}, \boldsymbol{A_2}$ – автоматы, распознающие языки $\boldsymbol{L_1}, \boldsymbol{L_2}$, соответственно.

$\boldsymbol{q_1}, \boldsymbol{q_2}$ — начальные вершины автоматов $\boldsymbol{A_1}, \boldsymbol{A_2}$, соответственно.

$\boldsymbol{r_1}, \boldsymbol{r_2}, \ldots$ — заключительные вершины автомата $\boldsymbol{A_1}$.

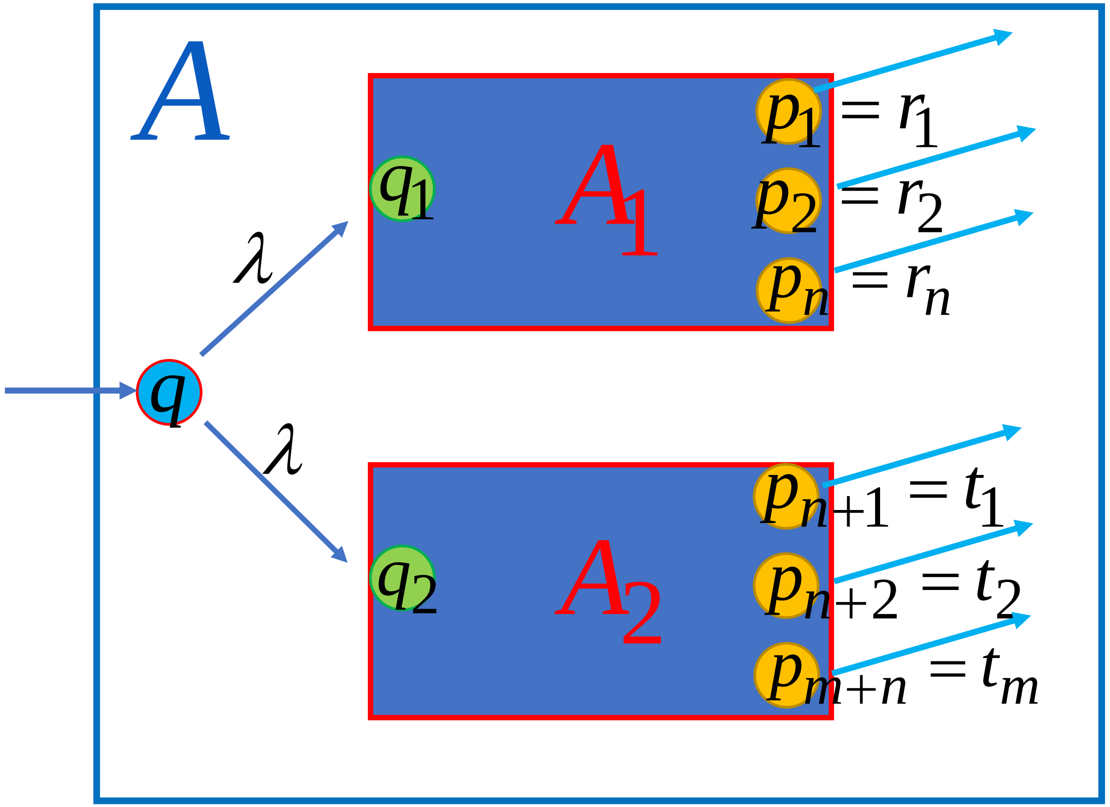
$\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots$ — заключительные вершины автомата $\boldsymbol{A_2}$.

По этим автоматам будет строиться автомат $\boldsymbol{A}$.

$\boldsymbol{q}$ — начальная вершина автомата $\boldsymbol{A}$.

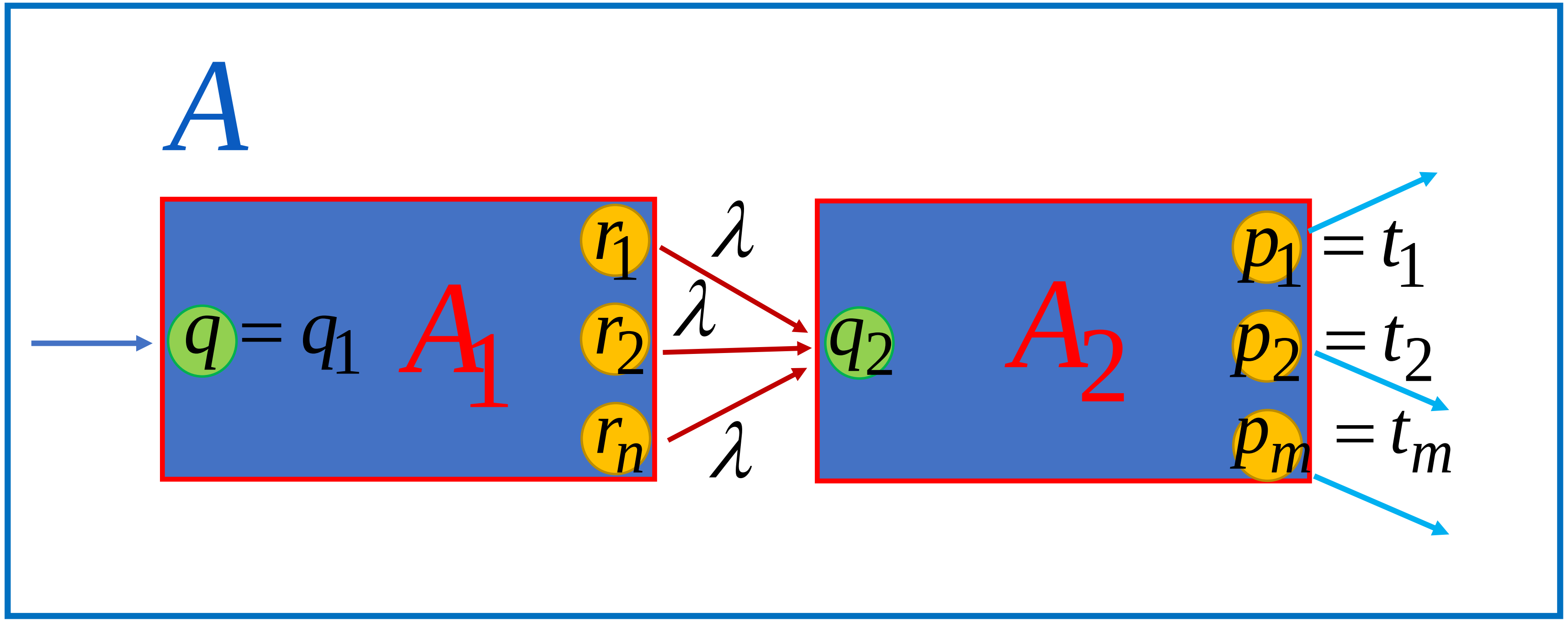
$\boldsymbol{p_1}, \boldsymbol{p_2}, \ldots$ — заключительные вершины автомата $\boldsymbol{A}$.

Построение λ-НКА для объединения языков



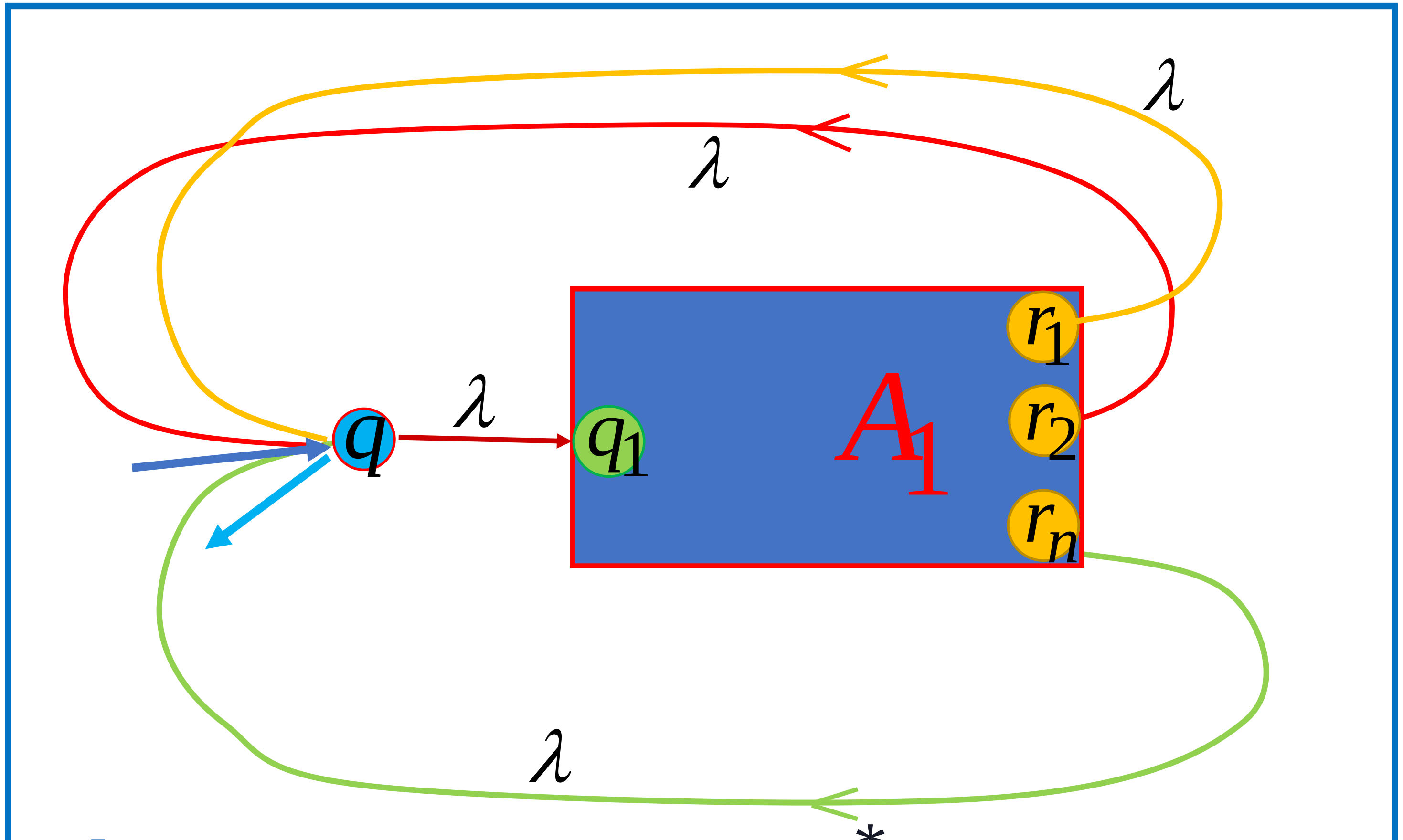
$$L(A) = L_1 \cup L_2 = L_1 + L_2$$

Построение λ-НКА для произведения языков



$$L(A) = L_1 L_2$$

Построение λ-НКА для операции * (итерации)

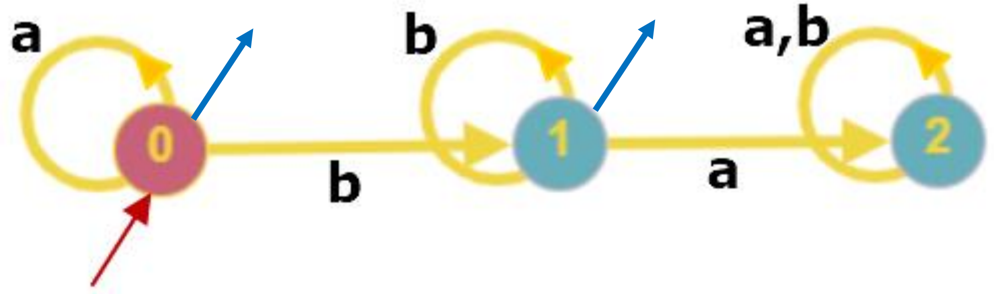


$$L(A) = (L_1)^*$$

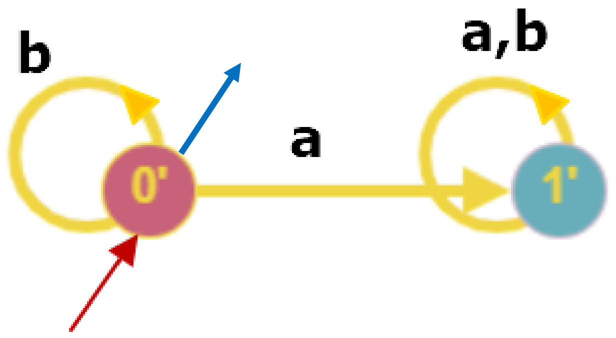
ЗАДАЧА 8а), I способ (построение ДКА для пересечения двух языков при помощи λ-НКА) и законов Де-Моргана.

Построить приведенный ДКА, задающий пересечение языков $L_1 = a^*b^*$ и $L_2 = b^*$

Языки L_1, L_2, задаваемые регулярными выражениями a^*b^*, b^* заданы соответственно автоматами:

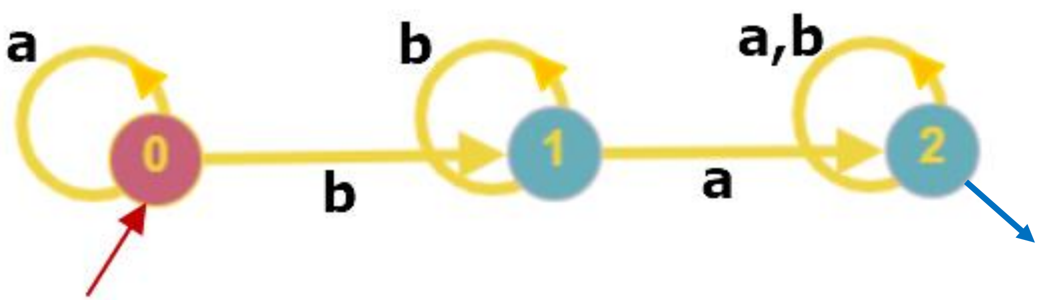


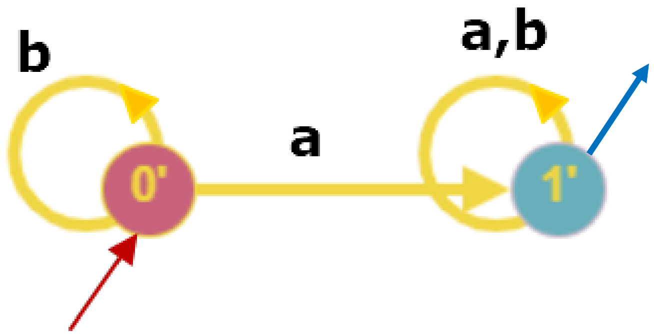
A_1 для $L_1 = a^*b^*$



A_2 для $L_2 = b^*$

Если язык L распознаётся ДКА $A = (\Sigma, Q, \delta, q_0, F)$, то $\bar{L}$ распознаётся $\bar{A} = (\Sigma, Q, \delta, q_0, Q \backslash F)$



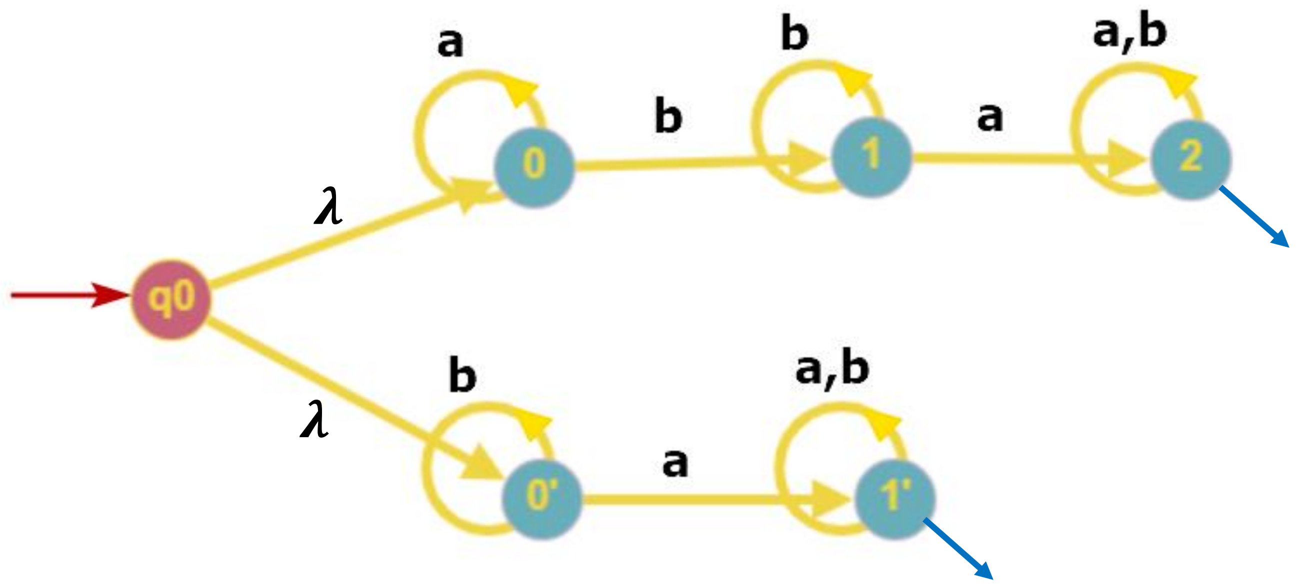


A_3 для $\overline{L_1}$

(видно, что $\overline{L_1} = a^*b^+a\Sigma^*$)

A_4 для $\overline{L_2}$

(видно, что $\overline{L_2} = b^*a\Sigma^*$)



A_5 – для $\overline{L_1} \cup \overline{L_2}$

Детерминируем его:

0) $Q_0 = Clousure\{q_0\} = \{q_0, 0, 0^{`}\}$

1) $Q_0 \to^{\alpha} \{0, 1^{`}\}, Clousure\{0, 1^{`}\} = \{0, 1^{`}\} \Rightarrow Q_0.a = Q_1 = \{0, 1^{`}\}$

$Q_0 \to^{b} \{0^{`}, 1\}, Clousure\{0^{`}, 1\} = \{0^{`}, 1\} \Rightarrow Q_0.b = Q_2 = \{0^{`}, 1\}$

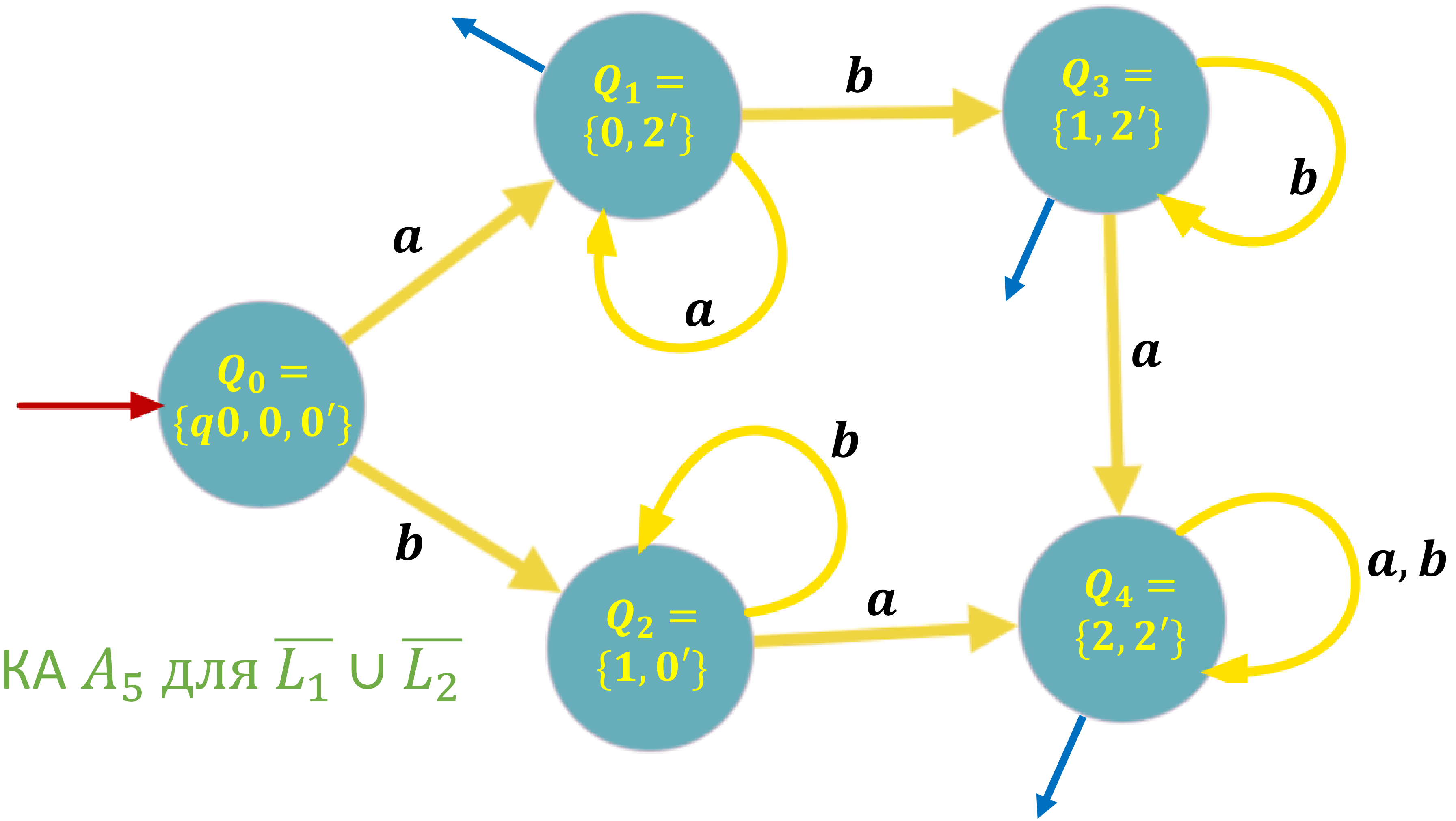
Аналогично, $Q_1.a = Q_1$ $\quad Q_1.b = Q_3 = \{1, 2^{`}\}$

$Q_2.a = Q_4 = \{2, 2^{`}\}$ $\quad Q_2.b = Q_2$

$Q_3.a = Q_4$ $\quad Q_3.b = Q_3$

$Q_4.a = Q_4$ $\quad Q_4.b = Q_4$

Построение ДКА для решения ЗАДАЧИ 8а), I способ



ДКА A_5 для $\overline{L_1} \cup \overline{L_2}$

Построение приведенного ДКА для решения ЗАДАЧИ 8б

Детерминируем A_5. Все вершины в A_5 достижимы.

$F = \{Q_1,\ Q_3,\ Q_4\}$, $\bar{F} = \{Q_0,\ Q_2\}$

$$\begin{array}{llll} & \quad I & \quad II & \\ 0) & Q_1\ Q_3\ Q_4 \mid Q_0\ Q_2 & & \\ 1) & & Q_0\ Q_2 \mid Q_1\ Q_3\ Q_4 & \\ 2) & & & Q_1\ Q_3\ Q_4 \mid Q_0\ Q_2 \end{array}$$

1)

	a	b
Q_1	I	I
Q_3	I	I
Q_4	I	I

$\{Q_1,\ Q_3,\ Q_4\}$ разделять не надо

2)

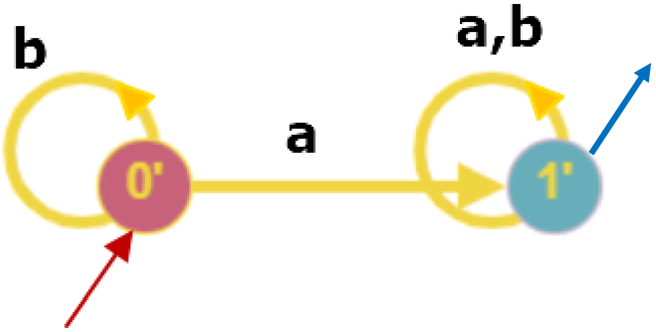
	a	b
Q_2	I	II
Q_4	I	II

$\{Q_0,\ Q_2\}$ тоже не надо разделять

Искомая конгруэнция $\approx$:

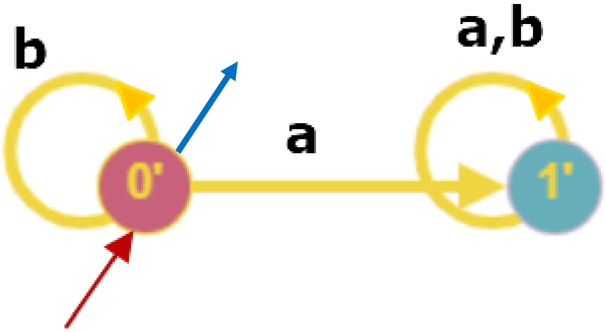
$\{\{Q_1\ Q_3\ Q_4\}, \{Q_0\ Q_2\}\}$.

Построение приведенного ДКА для решения ЗАДАЧИ 8а)



Фактор − автомат $A_5/\approx$ изоморфен автомату A_4, распознающего язык $\overline{L_1} \cup \overline{L_2} = b^*a\Sigma^*$

$0` = II = \{Q_0,\ Q_4\},\ 1` = I = \{Q_1,\ Q_3,\ Q_5\}$

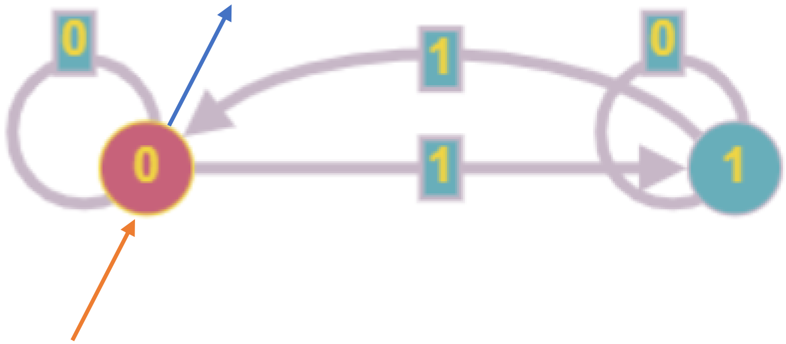


A_2 − автомат (минимальный) для $L = \overline{(\overline{L_1} \cup \overline{L_2})} = b^*$

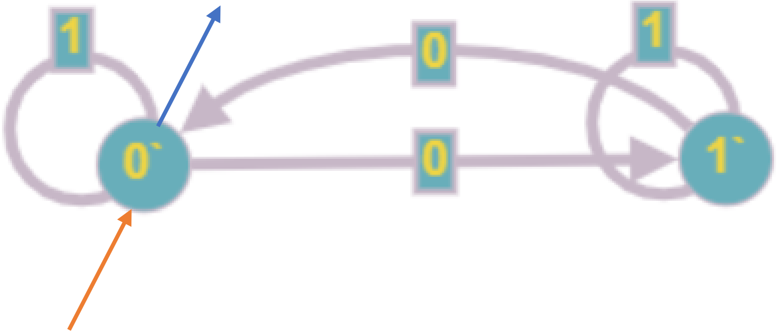
Задача 8б, I способ (Построение ДКА для пересечения двух языков при помощи λ-НКА) и законов де Моргана.

Построить приведённый ДКА, распознающий бинарные последовательности с чётным числом нулей и единиц

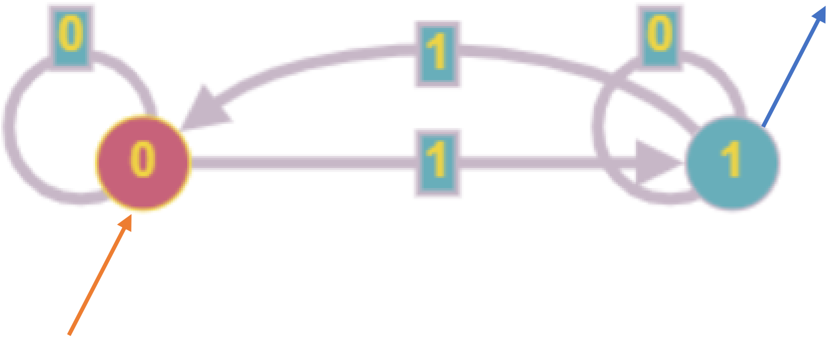
A_1 распознаёт язык L_1- множество последовательностей с чётным числом единиц



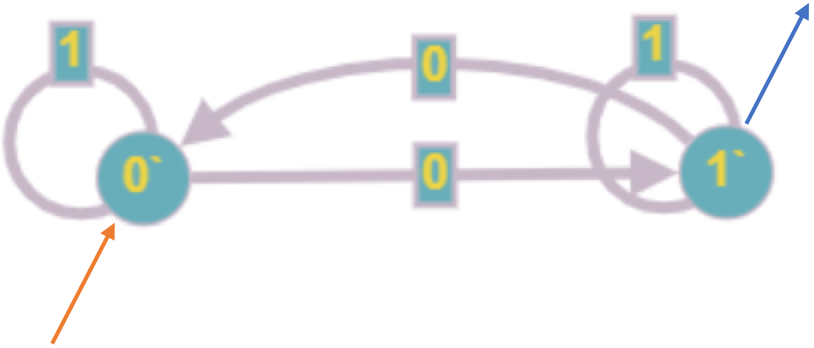
A_2 распознаёт L_2- множество последовательностей с чётным числом нулей



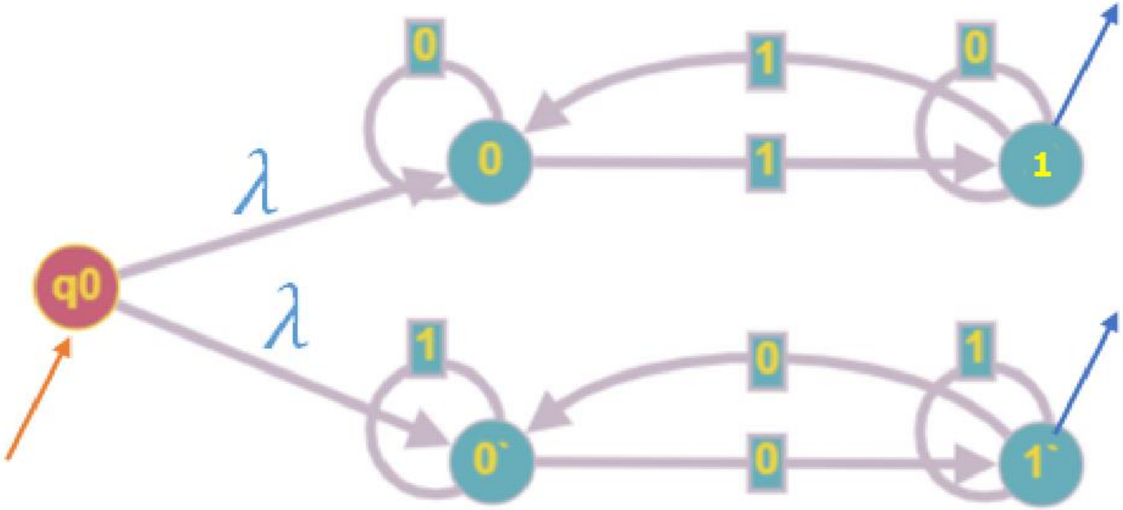
$\overline{A}_1$- автомат для дополнения $\overline{L}_1$ языка L_1



$\overline{A}_2$- автомат для дополнения $\overline{L}_2$ языка L_2

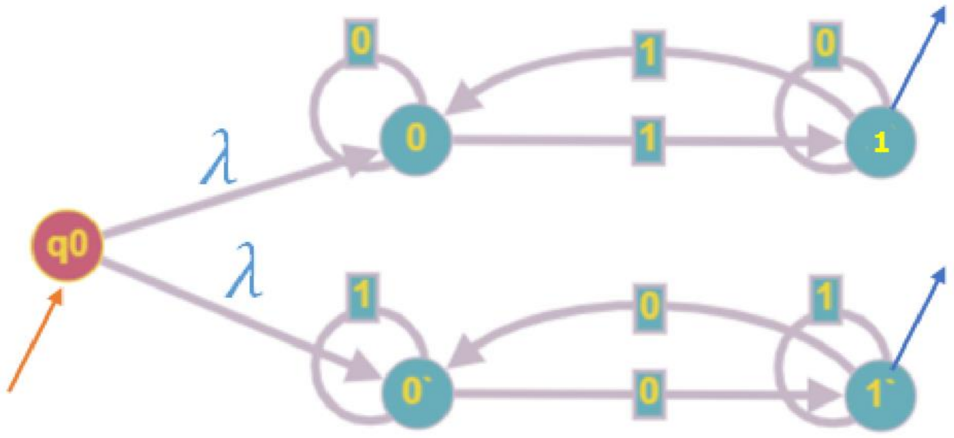


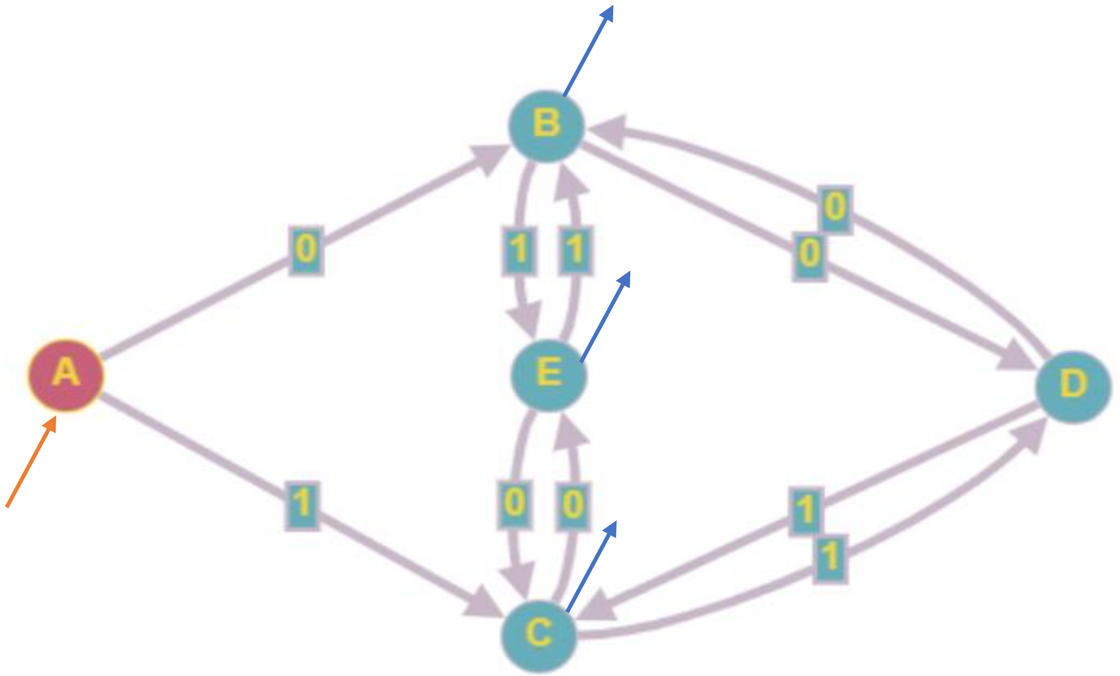
Автомат A_3, распознающий $\overline{L}_1 \cup \overline{L}_2$:



Детерминируем автомат, распознающий объединение дополнений

	0	1
$A=\{q_0 0,0^{`}\}$	B	C
$B=\{0,1^{`}\}\rightarrow$	D	E
$C=\{0^{`},1\}\rightarrow$	E	D
$D=\{0,0^{`}\}$	B	C
$E=\{1,1^{`}\}\rightarrow$	C	B

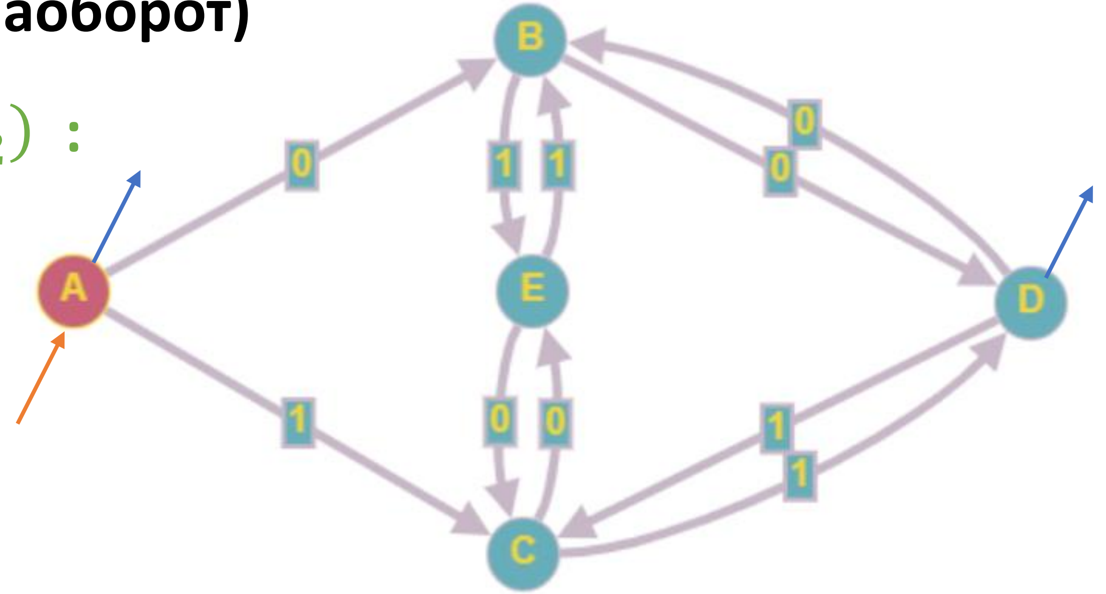




ДКА A_4, распознающий тот же язык, что и автомат НКА A_3.

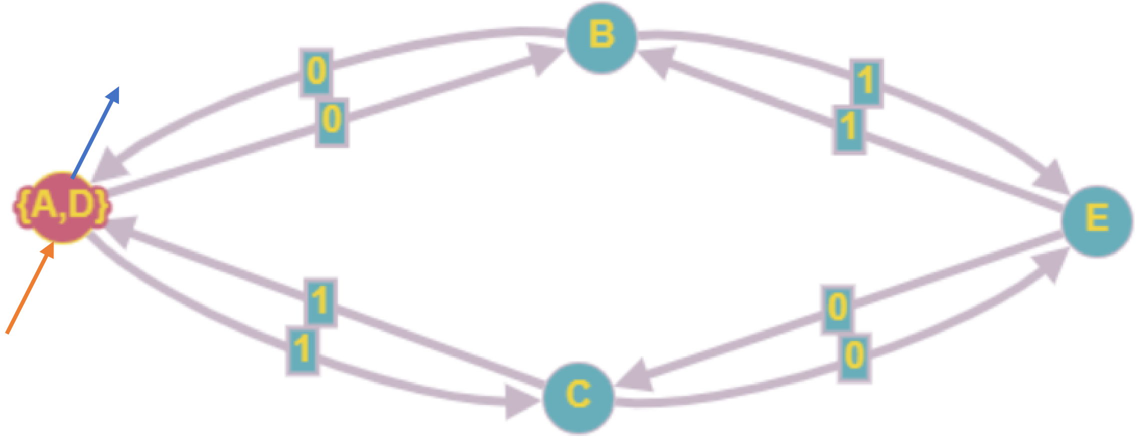
Построим дополнение для ДКА объединения дополнений
(делаем нетерминальные терминальными и наоборот)

Детерминирован. для $\overline{(\overline{L}_1 \cup \overline{L}_2)}$:



можно минимизировать

	B	C	E	A	D
0	II	I	I	I	I
1	I	II	I	I	I
0	IV	III	II	I	I
1	III	IV	I	II	II



Итоговый минимал. ДКА для $\overline{(\overline{L}_1 \cup \overline{L}_2)}$:

ТЕОРИЯ: построение ДКА для пересечения [объединения] двух языков

при помощи декартового произведения автоматов

Сначала построим автоматы, распознающие каждый язык:

Пусть у первого автомата:
всего n состояний: q0, q1, …, qn
F1- множество финальных состояний

Пусть у второго автомата:
всего k состояний: p0, p1, …, pk
F2 - множество финальных состояний

Тогда у искомого автомата всего n*k состояний, каждое из которых задается парой (qi, pj):
(q0, p0), (q0, p1), …, (q0, pk)
(q1, p0), (q1, p1), …, (q1, pk)
…
(qn, p0), (qn, p1), …, (qn, pk)

Функция переходов нового автомата задается как: δ((qi, pj), a) = (δ(qi, a), δ(pj, a))
т.е. допустим: δ(q0, a) = q1 ; δ(p0, a) = p1
Тогда: δ((q0, p0), a) = состояние (q1, p1)

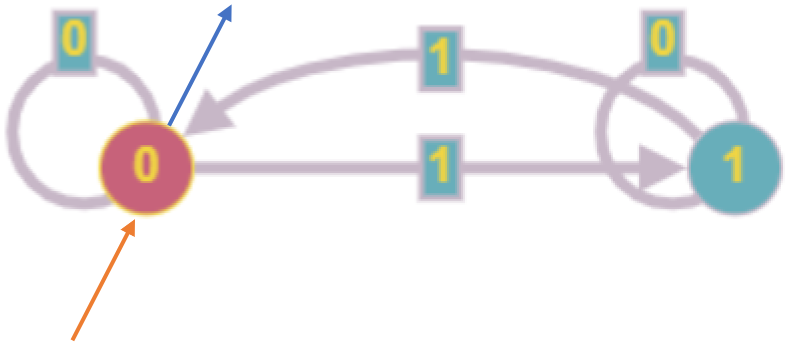
Различия между автоматами, распознающими пересечение и объединение, лишь во множестве финальных состояний:

В случае пересечения состояние (qi, pj) будет финальным, если $qi \in F1 \land pj \in F2$
В случае объединения состояние (qi, pj) будет финальным, если $qi \in F1 \lor pj \in F2$

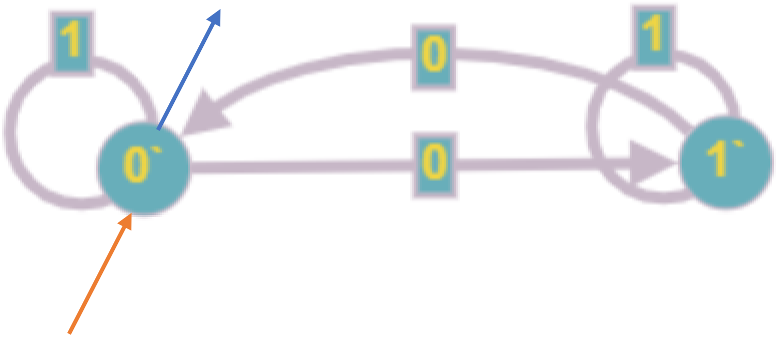
Задача 8б, II способ (Построение ДКА для пересечения двух языков при помощи декартового произведения автоматов)

Построить приведённый ДКА, распознающий бинарные последовательности с чётным числом нулей и единиц

A_1 распознаёт язык L_1- множество последовательностей с чётным числом единиц

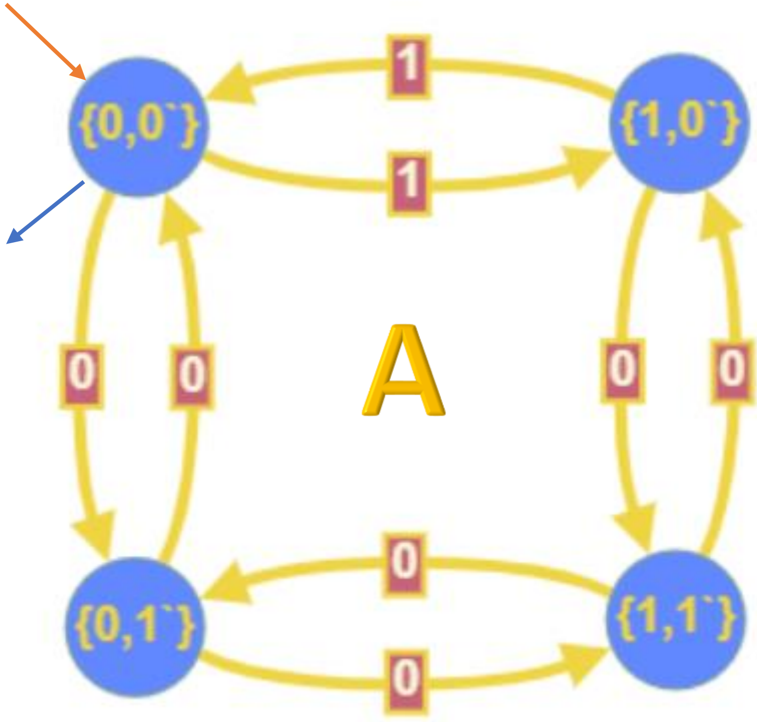


A_2 распознаёт L_2- множество последовательностей с чётным числом нулей



Задача 8б, II способ

Автомат *A* распознаёт язык *L* - множество последовательностей с чётным числом нулей и четным числом единиц



{0,0`}.0={0.0, 0`.0}={0,1`}
{0,0`}.1={0.1, 0`.1}={1,0`}
{1,0`}.0={1.0, 0`.0}={1,1`}
{1,0`}.1={1.1, 0`.1}={0,0`}

Аналогично,
{0,1`}.0= {0,0`}
{0,1`}.1= {1,1`}
{1,1`}.0= {1,0`}
{1,1`}.1= {1,1`}

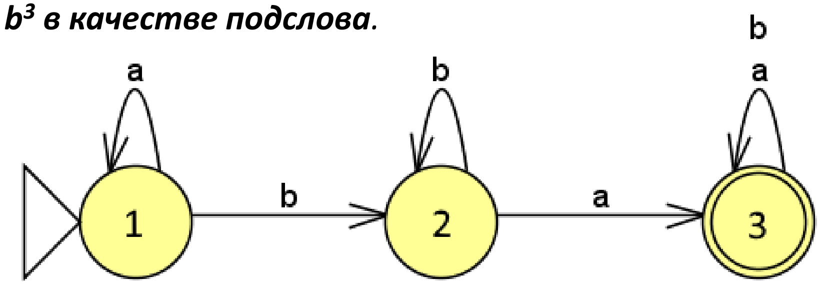
Вершина {0,0`} – является заключительной вершиной автомата ***A***,
вершина 0 – заключительная для автомата A_1 и
вершина 0` – заключительная для автомата A_2

Задача 8в (II способ)

Построить автомат, распознающий язык всех слов над алфавитом {a, b}, которые содержат ba, но не содержат b^3 в качестве подслова.

1. Построим автомат, распознающий слова с подсловом ba:

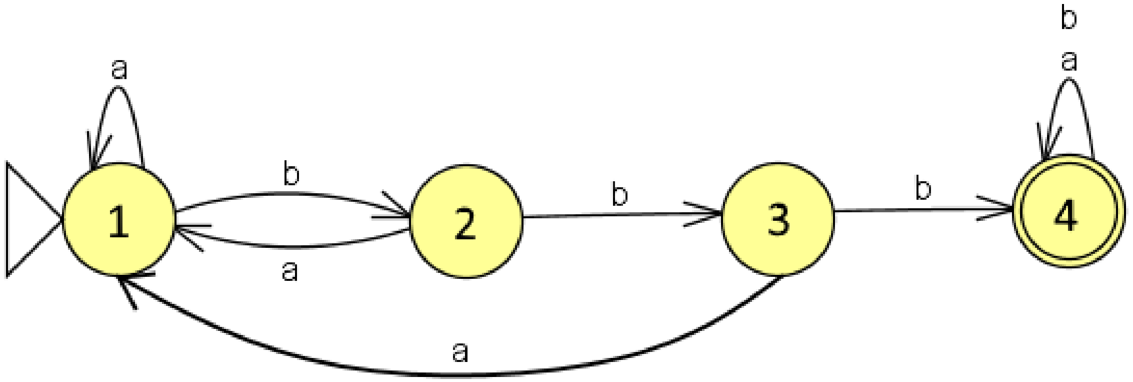
   $\left(\text{Язык } L_1 = \Sigma^* ba \Sigma^*\right)$



Подробно, как построить этот автомат, на двух СЛЕДУЮЩИХ слайдах

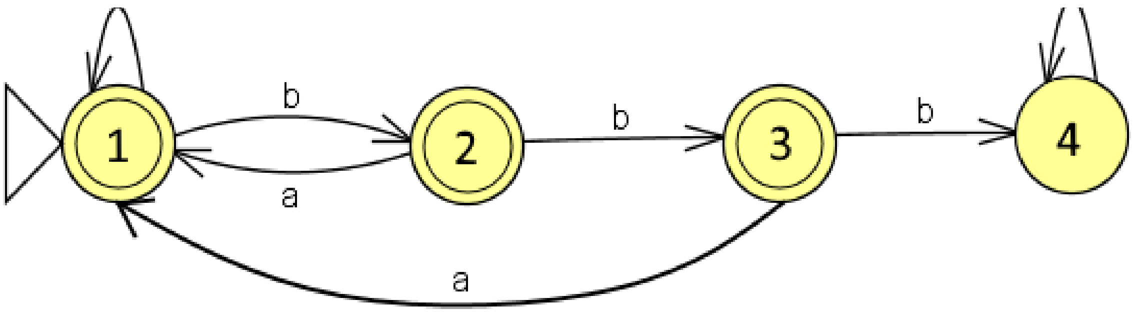
2. Построим автомат, распознающий слова с подсловом b^3:

   $\left(\text{Язык } L_2 = \Sigma^* b^3 \Sigma^*\right)$



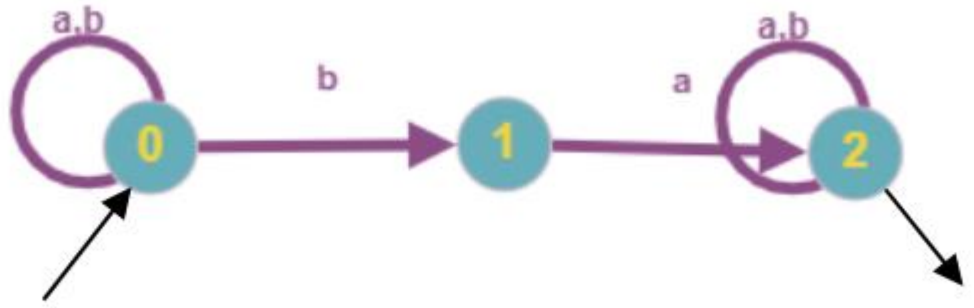
Автомат строится аналогично тому, как строился автомат в п.1

3. Построим автомат, распознающий язык $\overline{L_2}$. Достаточно изменить множество финальных состояний: $F_{new} = Q \setminus F$

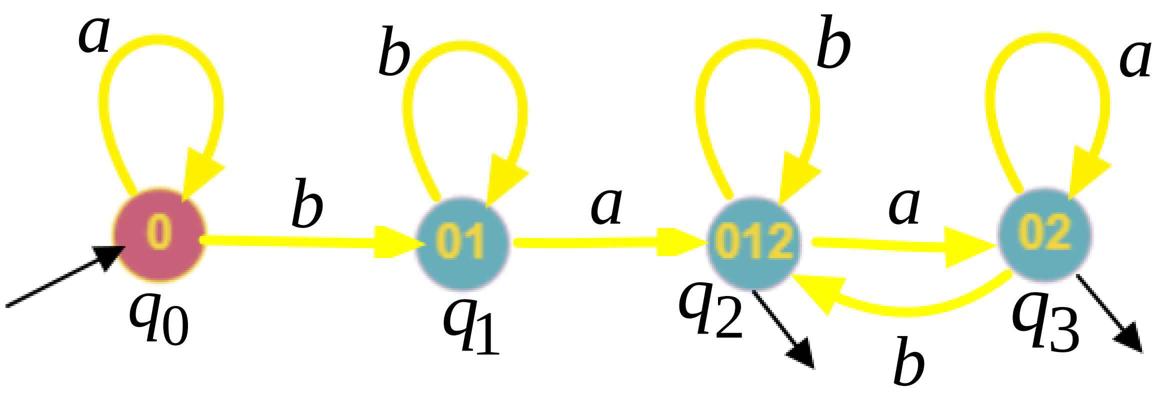


Задача 8в (II способ)

Построим НКА распознающий слова с подсловом *ba,* т.е. язык
$L = (a + b)^* ba(a + b)^*$

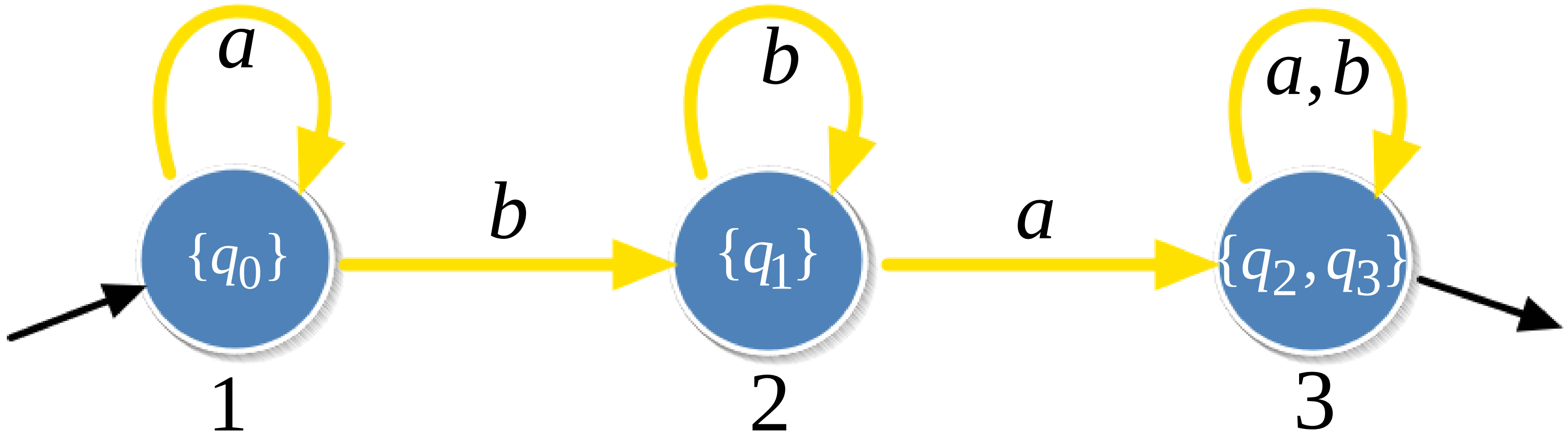


Построим ДКА распознающий тот же язык L.



Задача 8в (II способ)

Построим по нему приведенный ДКА

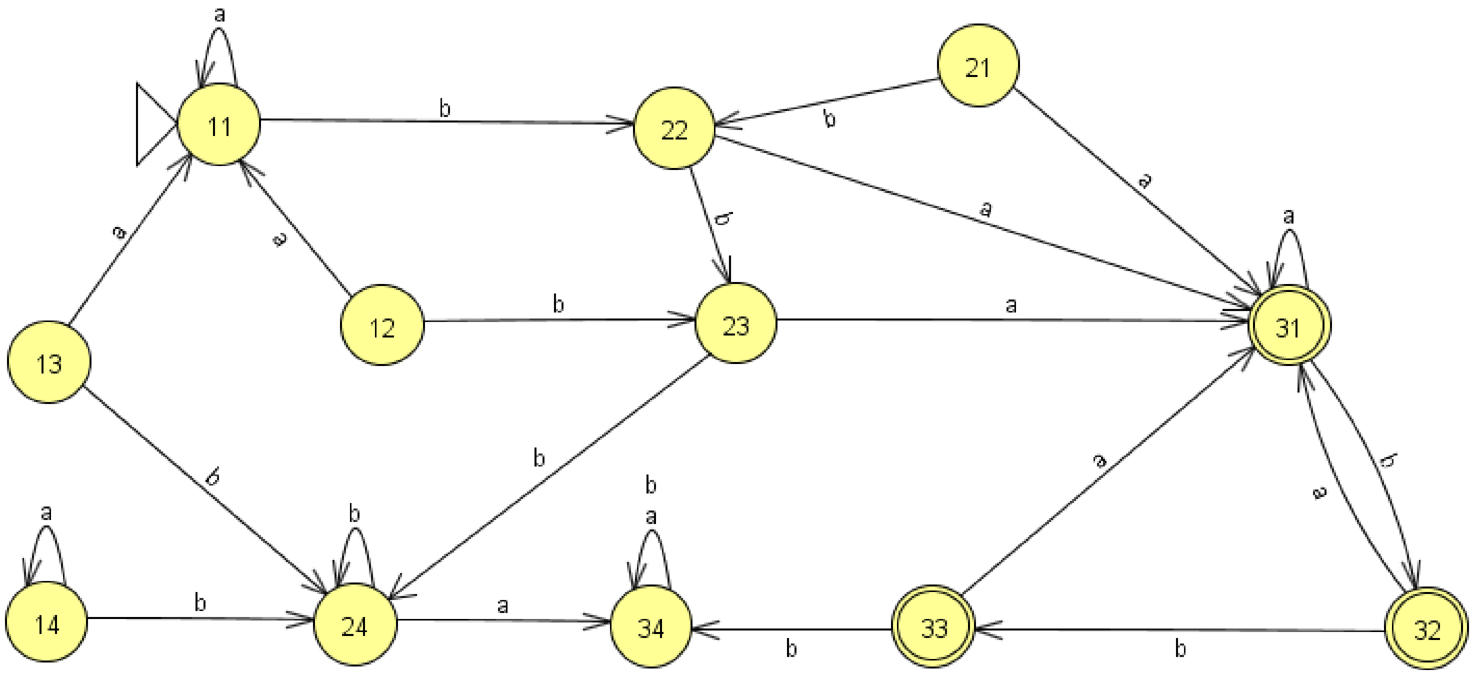


4. Построим автомат, распознающий пересечение языков автоматов 1 и 3.
В нем будет 3 * 4 = 12 состояний, задающихся парой (qi, pj), где

- qi - состояние из первого автомата
- pj - из второго.

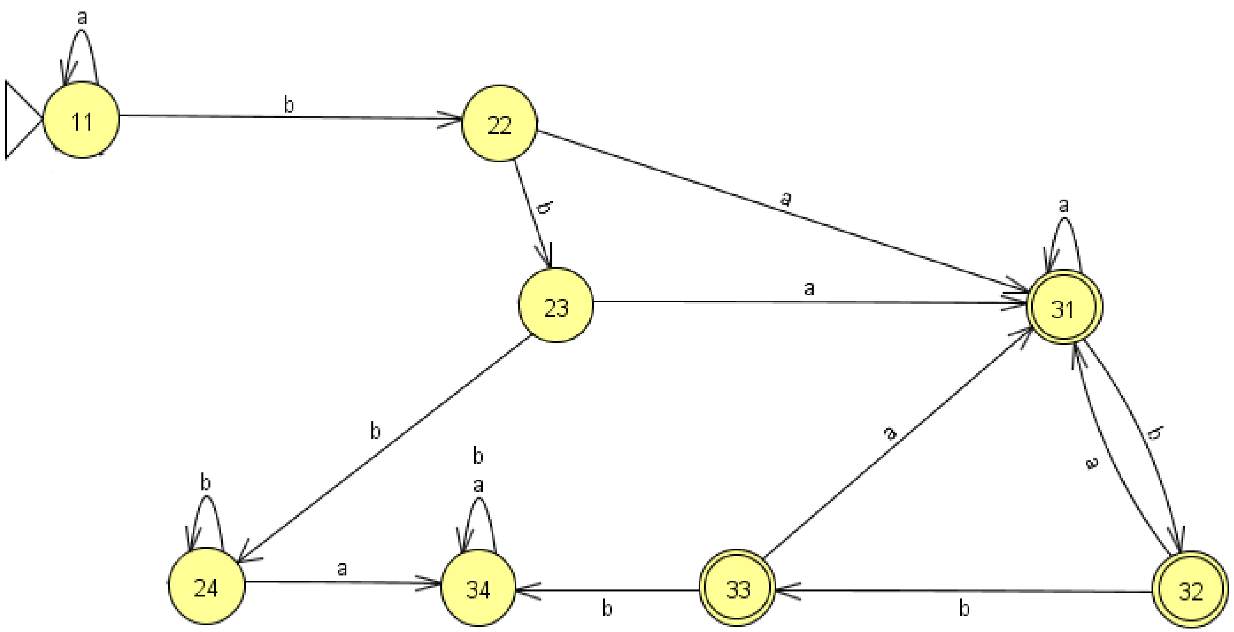
Состояние (qi, pj) - начальное, если qi, pj - начальные.
Состояние (qi, pj) - финальное, если qi, pj - финальные.

Построим таблицу переходов, по правилу: δ((qi, pj), a) = (δ(qi, a), δ(pj, a))
т.е. допустим: δ(q0, a) = q1 ; δ(p0, a) = p1
тогда: δ((q0, p0), a) = состояние (q1, p1)



Состояние	a	b
(1, 1)	(1, 1)	(2, 2)
(1, 2)	(1, 1)	(2, 3)
(1, 3)	(1, 1)	(2, 4)
(1, 4)	(1, 4)	(2, 4)
(2, 1)	(3, 1)	(2, 2)
(2, 2)	(3, 1)	(2, 3)
(2, 3)	(3, 1)	(2, 4)
(2, 4)	(3, 4)	(2, 4)
(3, 1)	(3, 1)	(3, 2)
(3, 2)	(3, 1)	(3, 3)
(3, 3)	(3, 1)	(3, 4)
(3, 4)	(3, 4)	(3, 4)

5. Заметим, состояния (12, 13, 14, 21) недостижимы, уберем их:



6. **Минимизируем** автомат, разбив вершины на классы:

класс I : {31, 32, 33} - финальные состояния
класс II : {11, 22, 23, 24, 34} - остальные

Построим таблицу, в вершину какого класса автомат попадет после чтения буквы.
Например: δ(22, a) = 31 - вершина I класса

	31	32	33	11	22	23	24	34
a	I	I	I	II	I	I	II	II
b	I	I	II	II	II	II	II	II

Не все вершины внутри классов переходят в одинаковые классы, значит можно выделить дополнительные, создав новое разбиение:
I - {31, 32} II - {33} III - {11, 24, 34} IV - {22, 23}
Построим аналогичную таблицу для новых классов:

	31	32	33	11	22	23	24	34
a	I	I	I	III	I	I	III	III
b	I	II	III	IV	IV	III	III	III

Можно выделить классы:

I - {31} II - {32} III - {33} IV - {11} V - {22} VI - {23} VII - {24, 34}

Построим аналогичную таблицу для новых классов:

	31	32	33	11	22	23	24	34
a	I	I	I	IV	I	I	VII	VII
b	II	III	VII	V	VI	VII	VII	VII

Вершины внутри класса эквивалентны, соответственно можно рассматривать целый класс в виде одной вершины.

Тогда предыдущая таблица - таблица переходов искомого минимизированного автомата: