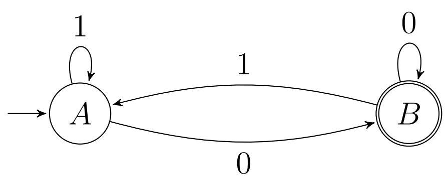
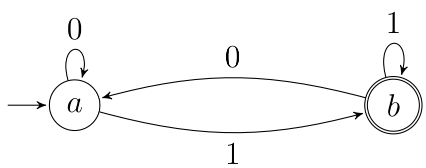
Формальные языки

домашнее задание до 23:59 05.03

1. Доказать или опровергнуть утверждение: произведение двух минимальных автоматов всегда дает минимальный автомат (рассмотреть случаи для пересечения, объединения и разности языков).

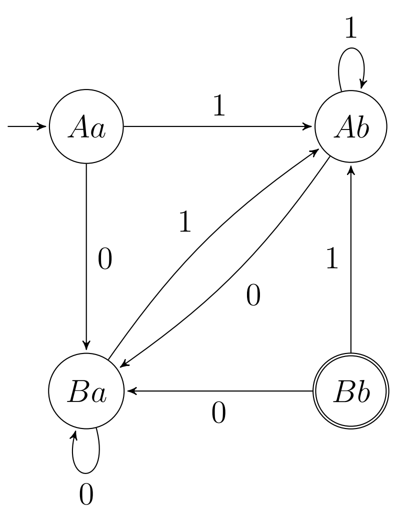
   Решение: Возьмем два автомата, один из которых принимает бинарные слова, оканчивающиеся на 0, а другой — оканчивающиеся на 1. Их минимальные автоматы, очевидно, выглядят так:

   (a) 

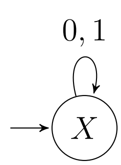
   (b) 

   Возьмем их произведение:

   (a) Пересечение:

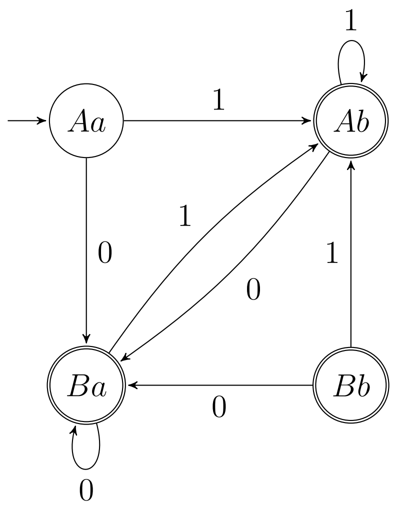
   

   Но минимальный автомат должен быть таким (то есть у нас пересечение пустое и

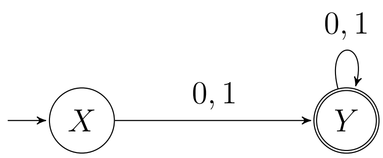
   

   нет терминальной вершины):

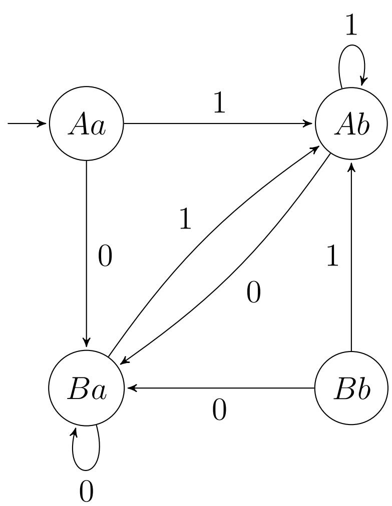
(b) Объединение:



Но минимальный автомат должен быть таким (то есть у нас объединение это все непустые бинарные слова):



(c) Разность первого и второго языка — это пустое множество, поэтому минимальный автомат как в пересечении, а произведение без терминальных вершин, то есть такое:

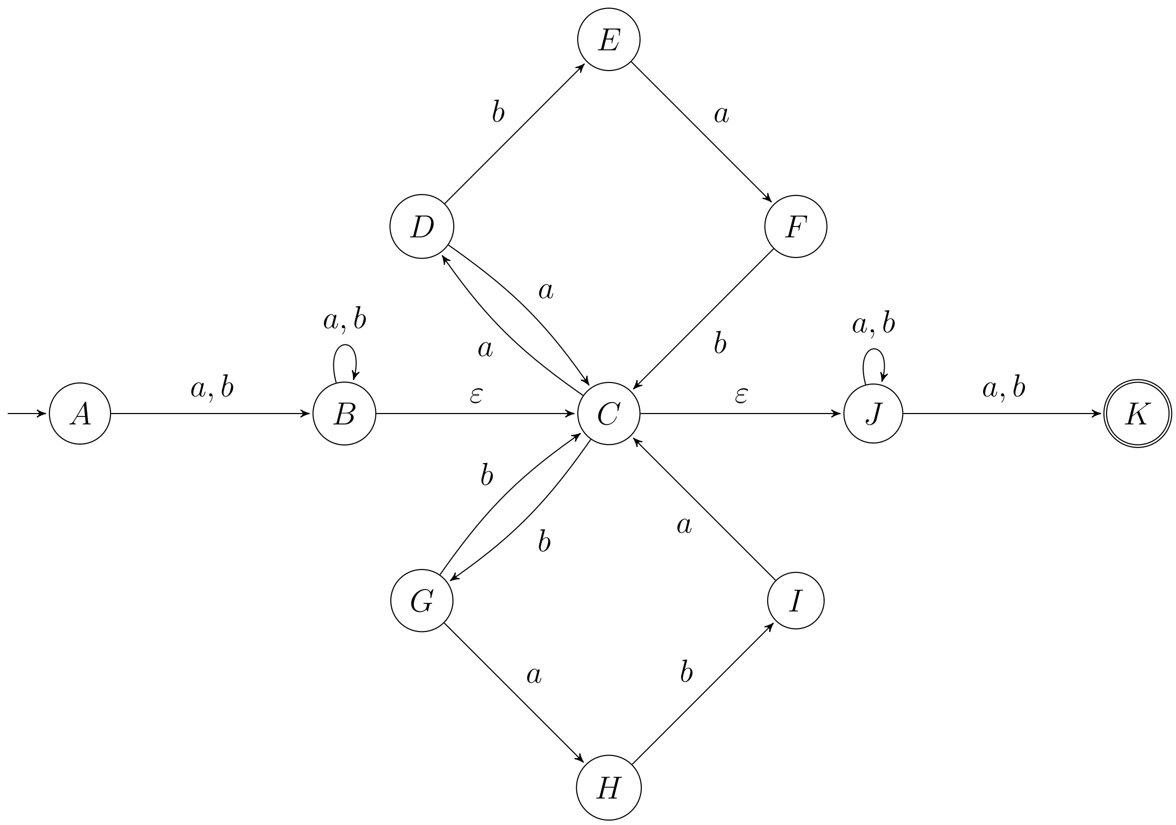


2. Для регулярного выражения:

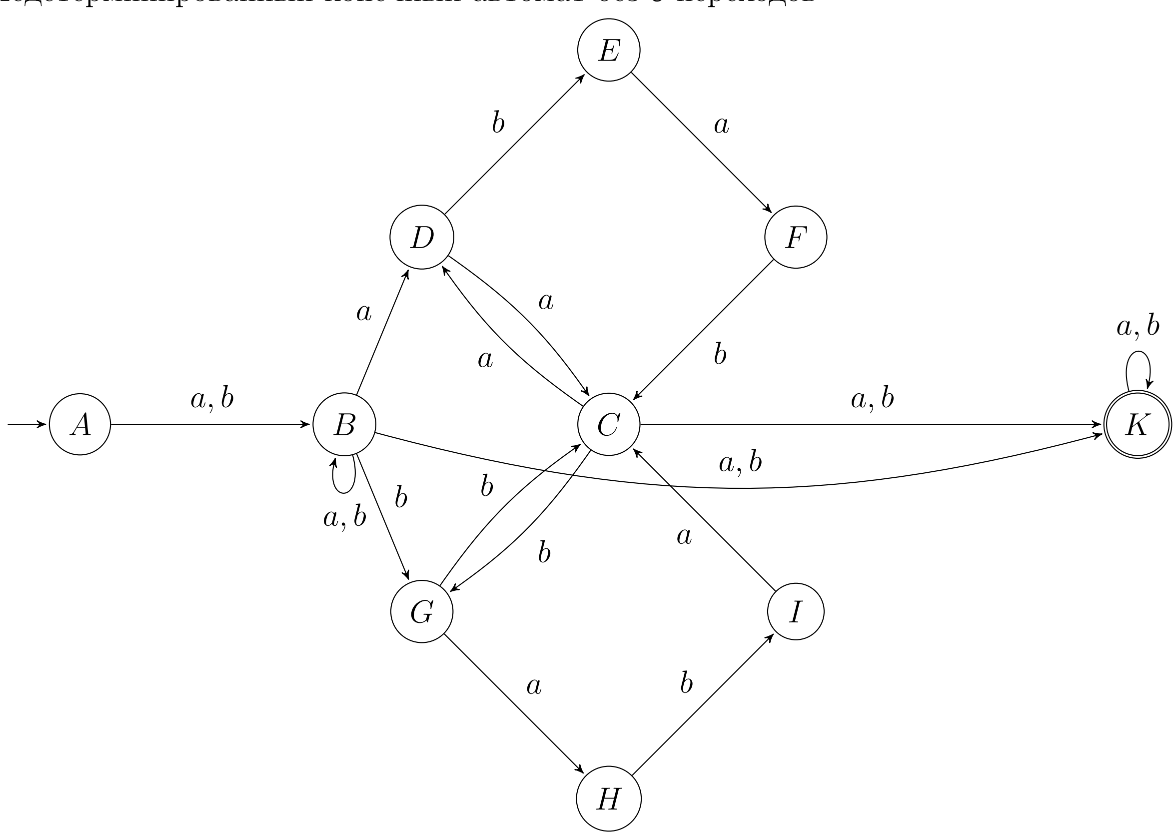
$$(a \mid b)^+(aa \mid bb \mid abab \mid baba)^*(a \mid b)^+$$

Построить эквивалентные:

(a) Недетерминированный конечный автомат

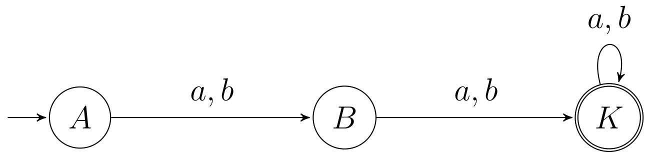


(b) Недетерминированный конечный автомат без ε-переходов

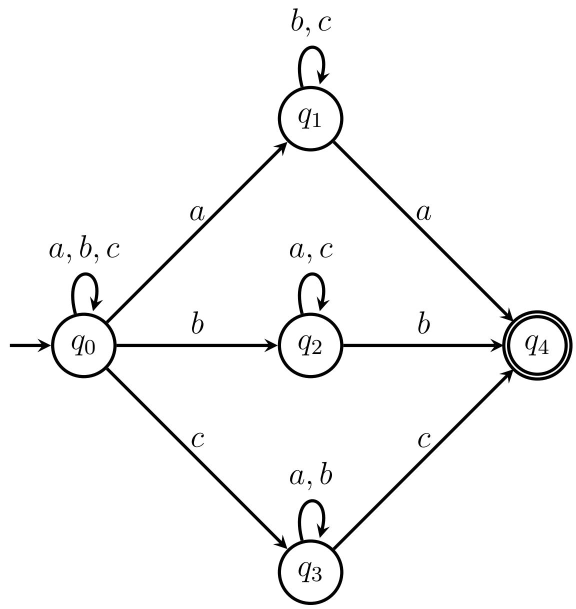


(c) Минимальный полный детерминированный конечный автомат
Можно присмотреться на определение регулярного языка или на предыдущий пункт

и понять, что это просто все такие слова, в которых хотя бы 2 символа (то есть можно просто убрать вершину C и два ромбика, который сверху и снизу) и получается:



3. Построить регулярное выражение, распознающее тот же язык, что и автомат:



Решение: $(a|b|c)^*((a(b|c)^*a) \mid (b(a|c)^*b) \mid (c(a|b)^*c))$

4. Определить, является ли автоматным язык $\{\omega\omega^r \mid \omega \in \{0,1\}^*\}$. Если является — построить автомат, иначе — доказать.

   Решение: Обозначим этот язык за L и предположим, что он автоматный. Возьмем n из леммы о накачке и $w = 01^n001^n0$. Длина этого слова не меньше n и оно лежит в L, поэтому из леммы верно, что $\exists x, y, z \in \Sigma^* : \ xyz = w, |y| \neq \varepsilon, \ |xy| \leq n$, т.ч. $\forall k \ xy^kz \in L$. Возможны два случая:

   (a) $x = \varepsilon$ (пустое слово), тогда в y будет ровно 1 ноль, так как есть ограничение на длину, а в z их тогда будет 3. Возьмем $k = 2$ и получим, что в xy^kz будет нечетное количество нулей (5), поэтому это слово точно не будет в L.

   (b) $x \neq \varepsilon$. Возьмем $k = 3$. Заметим, что второй ноль в w по построению должен быть в z, поэтому он будет после xy^k (в xy^k только один 0 и он в x, так как оно не пусто). Чтобы xy^kz было в L как минимум нужно, чтобы в левой и правой половине слова было по 2 нуля, но такое невозможно, потому что второй ноль стоял на позиции $n+2$ в w, а в таком слове на $n + 2 + 2|y|$, а середина слова была между $n + 2$ и $n + 3$, а стала между $n + 2 + |y|$ и $n + 3 + |y|$, то есть левее второго нуля. Поэтому такой случай тоже приводит к противоречию.

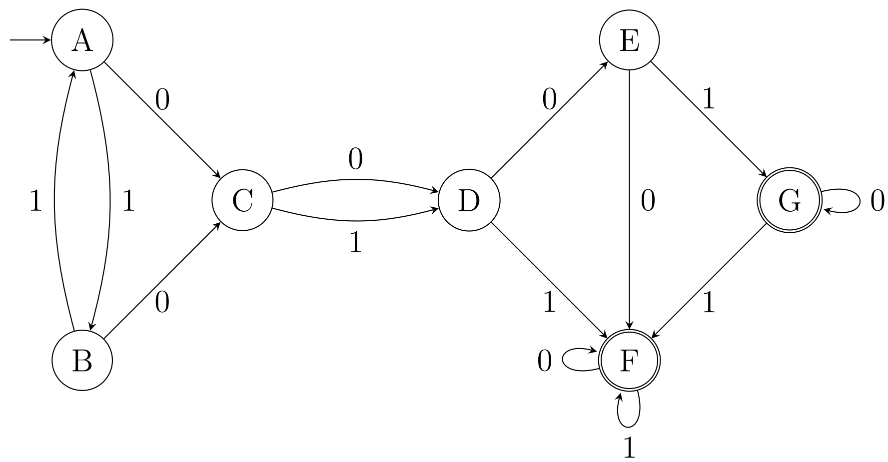
   Язык не автоматный.

5. Определить, является ли автоматным язык $\{uaav \mid u, v \in \{a, b\}^*, |u|_b \geq |v|_a\}$. Если является — построить автомат, иначе — доказать.

   Решение: Не является. Докажем это также через лемму о накачке: Пусть наш язык L автоматный. Берем n из леммы и $w = b^n aa(ba)^n$, оно лежит в L. Но если мы возьмем $k = 0$, то, поскольку y по условию непуст, до aa станет $< n$ нулей (в строке xy^kz возможно только одно вхождение aa для любых x, y, z), а после aa будет n символов a. Получили противоречие с тем, что $\forall k\ xy^kz \in L$.

Пример применения алгоритма минимизации

Минимизируем данный автомат:



Автомат полный, в нем нет недостижимых вершин — продолжаем.

Строим обратное δ отображение.

δ^{-1}	0	1
A	—	B
B	—	A
C	A B	—
D	C	C
E	D	—
F	E F	D F G
G	G	E

Отмечаем в таблице и добавляем в очередь пары состояний, различаемых словом ε: все пары, один элемент которых — терминальное состояние, а второй — не терминальное состояние. Для данного автомата это пары

$(A, F), (B, F), (C, F), (D, F), (E, F), (A, G), (B, G), (C, G), (D, G), (E, G)$

Дальше итерируем процесс определения неэквивалентных состояний, пока очередь не оказывается пуста.

(A, F) не дает нам новых неэквивалентных пар. Для (B, F) находится 2 пары: $(A, D), (A, G)$. Первая пара не отмечена в таблице — отмечаем и добавляем в очередь. Вторая пара уже отмечена в таблице, значит, ничего делать не надо. Переходим к следующей паре из очереди. Итерируем дальше, пока очередь не опустошится.

Результирующая таблица (заполнен только треугольник, потому что остальное симметрично) и порядок добавления пар в очередь.

	A	B	C	D	E	F	G
A							
B							
C	✓	✓					
D	✓	✓	✓				
E	✓	✓	✓	✓			
F	✓	✓	✓	✓	✓		
G	✓	✓	✓	✓	✓		

Очередь:
$(A, F), (B, F), (C, F), (D, F), (E, F), (A, G), (B, G), (C, G), (D, G), (E, G),$
$(B, D), (A, D), (A, E), (B, E), (C, E), (C, D), (D, E), (A, C), (B, C))$

В таблице выделились классы эквивалентных вершин: $\{A, B\}, \{C\}, \{D\}, \{E\}, \{F, G\}$. Остается только нарисовать результирующий автомат с вершинами-классами. Переходы добавляются тогда, когда из какого-нибудь состояния первого класса есть переход в какое-нибудь состояние второго класса. Минимизированный автомат: