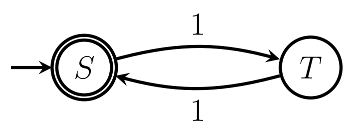
Формальные языки

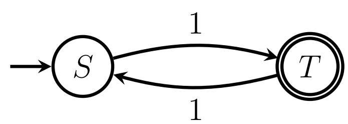
домашнее задание до 23:59 05.03

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

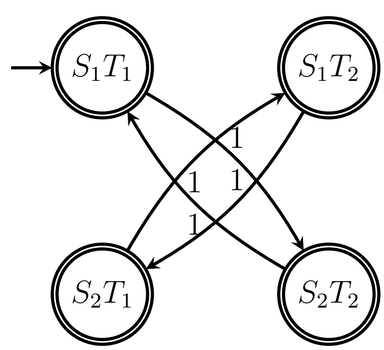
   (a) Для объединения.
   Возьмем в качестве первого языка $L_1 = 1^{2n}$ и в качестве второго $L_2 = 1^{2n+1}$
   Min ДКА для L_1:

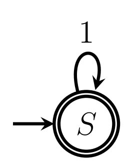
   Min ДКА для L_2:

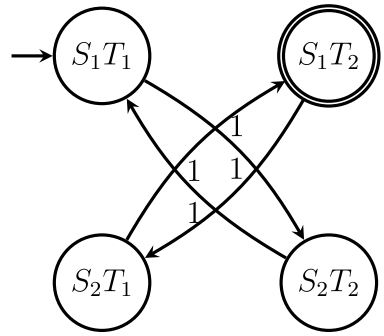
   Их произведение для объединения:

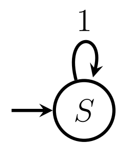
   Min ДКА для объединения:

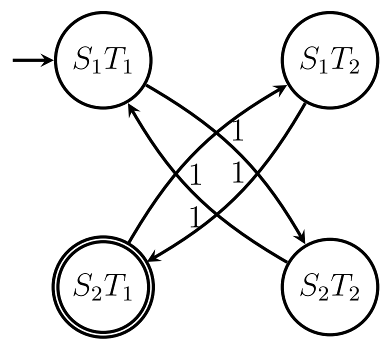
   (b) Для пересечения:
   Возьмем такие же L_1 и L_2.
   Их произведение для пересечения:



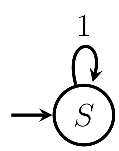
Min ДКА для пересечения:



(c) Для разности:
Возьмем произведение двух автоматов для языка $L = 1^{2n+1}$.
Их произведение:



Min ДКА для разности:

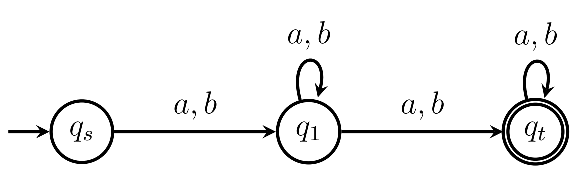


2. Для регулярного выражения:

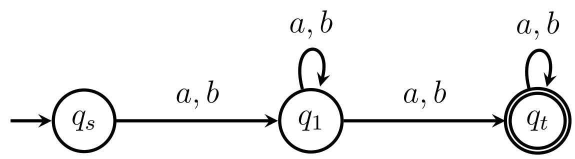
$$(a \mid b)^+(aa \mid bb \mid abab \mid baba)^*(a \mid b)^+$$

Построить эквивалентные:
Решение
Эта регулярка эквивалетна $(a \mid b)^+(a \mid b)^+$.

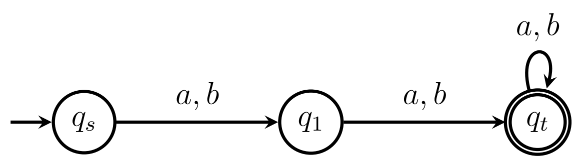
(a) Недетерминированный конечный автомат
Решение



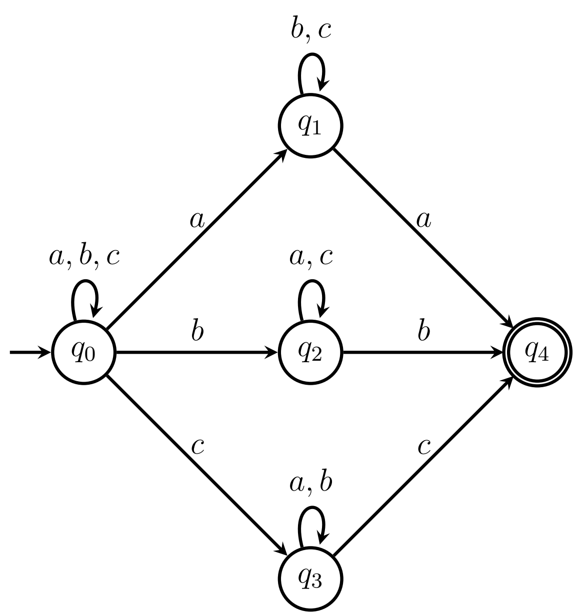
(b) Недетерминированный конечный автомат без ε-переходов
Решение



(c) Минимальный полный детерминированный конечный автомат
Решение



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



Решение

$$(a \mid b \mid c)^*((a(b \mid c)^*a) \mid (b(a \mid c)^*b) \mid (c(a \mid b)^*c))$$

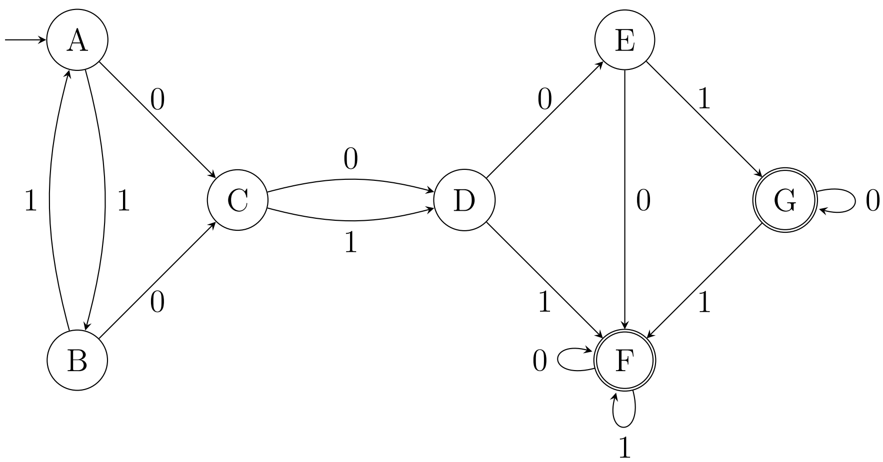
4. Определить, является ли автоматным язык $\{\omega\omega^r \mid \omega \in \{0,1\}^*\}$. Если является — построить автомат, иначе — доказать.
Решение
Пусть L - автоматный. Тогда для него верна лемма о накачке.
Возьмем n из леммы и возьмем слово $\alpha = (1)^n 00(1)^n$.
Мы знаем что есть слова $u, v, w \in L$, такие что $|uv| \leq n, |v| \geq 1$
Тогда $|v| = \{1\}^+$. Рассмотрим слово $uvvw = (1)^{n+|v|}00(1)^n$, оно должно лежать в языке по лемме.

Но, с другой стороны, это слово не является палиндромом.
Противоречие.

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
Пусть L - автоматный. Тогда для него верна лемма о накачке.
Возьмем n из леммы и возьмем слово $\alpha = (b)^n aa(ba)^n$.
Мы знаем что есть слова $u, v, w \in L$, такие что $|uv| \leq n, |v| \geq 1$
Тогда $v \in \{b\}^+$. Рассмотрим $uw = (b)^{n-|v|}aa(ba)^n$, оно должно лежать в языке по лемме.
Но, с другой стороны, его нельзя представить в виде двух слов U и V, что $uw = UaaV,\ |U|_b \geq |U|_a$, так как $(n - |v| = |U|_b < |U|_a = n)$.
Противоречие.

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: