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).

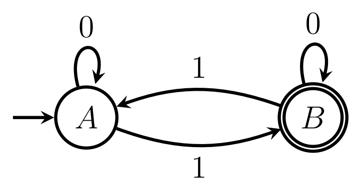
   Решение:

   Утверждение неверно во всех случаях. Более того, неверно даже то, что автомат будет минимальным с точностью до удаления недостижимых вершин. (в примере недостижимых вершин в произведении нет).

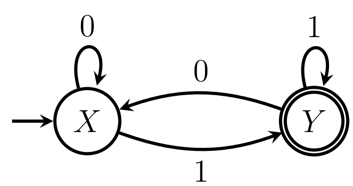
   Рассмотрим два языка над алфавитом $\{0, 1\}$. Пусть L_1 — слова с нечетным количеством единиц, L_2 — слова, заканчивающиеся на 1.

   Соответствующие минимальные автоматы:

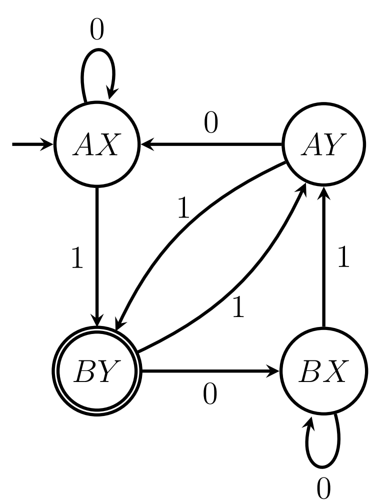
   Для L_1:

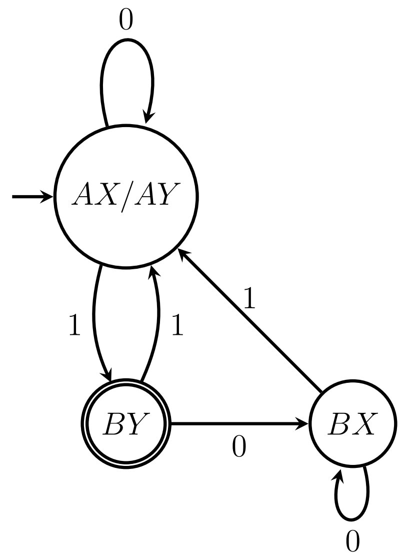


   Для L_2:

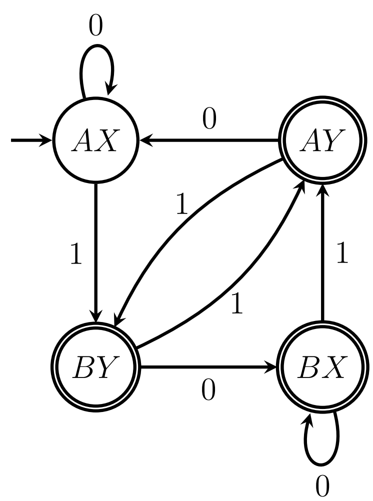


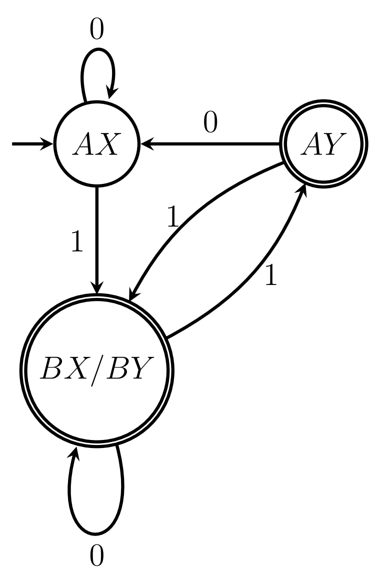
   Автоматы для $L_1 \cap L_2$: автомат-произведение и минимальный соответственно:



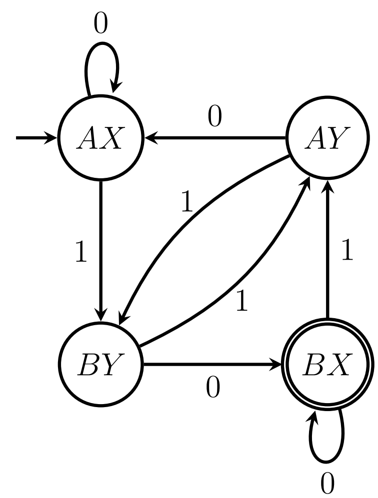


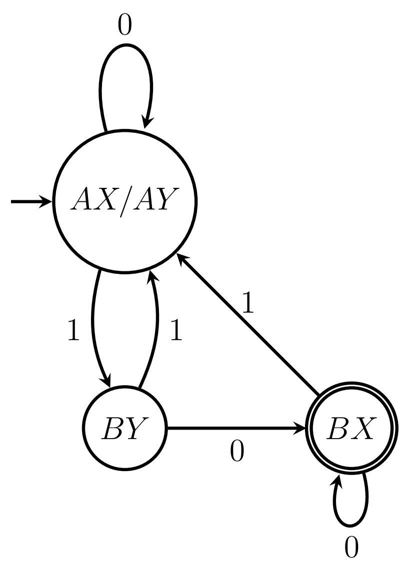
Автоматы для $L_1 \cup L_2$: автомат-произведение и минимальный соответственно:





Автоматы для $L_1 \setminus L_2$: автомат-произведение и минимальный соответственно:





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

Решение:

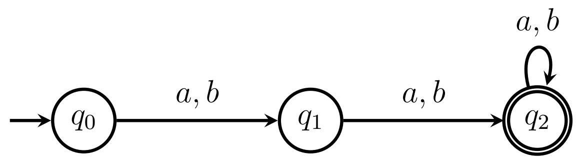
Заметим, что это регулярное выражение принимает ровно те слова, в которых хотя бы 2 символа (a или b).

Доказательство.

Пусть слово принимается выражением. Тогда хотя бы один символ соответствует первой скобке, и еще хотя бы один — последней.

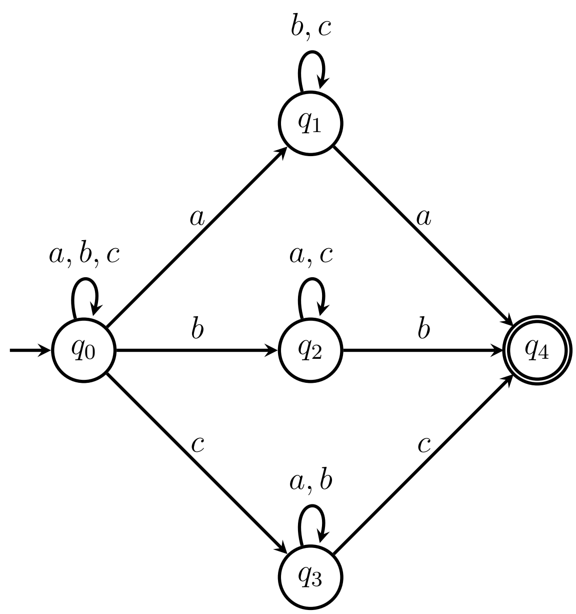
Пусть слово длины хотя бы 2. Тогда сопоставим первый символ первой скобке, второй скобке — ε, третьей — все остальные символы.

Тогда можем построить очевидный автомат, который подходит под все три пункта (да, мне лень набирать три разных):



Про минимальность тоже понятно: q_2 принимает все слова, q_1 — все непустые, q_0 — все длины хотя бы 2.

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



Решение:

Кажется, что необязательно применять пошаговый алгоритм и рисовать миллион промежуточных автоматов... Методом внимательного взгляда можем получить выражение:

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

Если посмотреть еще внимательнее, становится ясно, что этот автомат (и выражение) принимают ровно те непустые слова, в которых последний символ не является единственным таким символом в слове.

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

Решение:

Вспомним, что автоматные языки — это то же, что регулярные.

Пусть наш язык регулярный, тогда зафиксируем n из леммы о накачке. Рассмотрим слово $w = \omega\omega^r$, где $\omega = 0^{n-1}1$. Это слово длины $2n$, лежащее в языке. Возьмем разбиение из леммы $w = xyz, |xy| \leq n, |y| \geq 1$.

Если y содержит единицу, то в слове $xyyz$ 3 единицы, и оно не может быть четным палиндромом.

Иначе, y состоит из одних нулей. Но тогда, если бы было верно $xyyz = \omega_0\omega_0^r$, ω_0 должно было бы содержать ровно одну единицу (т. к. в $xyyz$ их две), но тогда (поскольку единицы все еще стоят рядом) ω_0^r было бы короче, чем ω_0.

В обоих случаях получили противоречие, а значит, язык не регулярный.

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

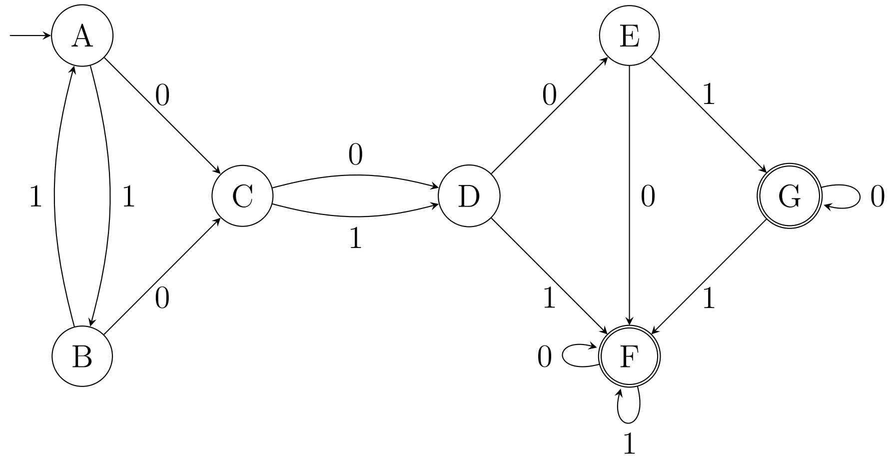
   Решение:

   Пусть язык регулярный, снова зафиксируем n из леммы о накачке. Рассмотрим слово $w = b^n aa(ba)^n$. Это слово длины $2n + 2$, лежащее в языке. Возьмем разбиение из леммы $w = xyz, |xy| \leq n, |y| \geq 1$. Заметим, что $y = b^p, p \geq 1$. Но тогда слово $xz = b^{n-p}aa(ba)^n$ также должно лежать в языке. Поскольку aa входит в слово лишь один раз, возможен только случай $u = b^{n-p}, v = (ba)^n$, но тогда $n - p = |u|_b < |v|_a = n$.

   Получили противоречие, а значит, язык не регулярный.

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: